الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة عمّار ثليجي بالأغواط
University of Amar Telidji - Laghouat



كلية التكنولوجيا
Faculty of Technology
قسم الالكتروتقني
Department of Electrical Engineering

MASTER Dissertation

Presented by:
CHERAIF Souad

Domain: Science and Technology
Field: Electrical Engineering
Option: Automation and Systems

Theme

Modeling & Intelligent Control of a Multi-DoF Robotic Arm

Jury members:

Full Name	**Grade**	**Quality**
Mrs. ABDEREZZAK Hadjer	M.C.A	Chair
Mr. RAHMANI Belkacem	M.C.A	Examiner
Mr. MOKRANI Lakhdar	Prof.	Supervisor
Mrs. BENLARBI Keltoum	M.C.B	Co-supervisor

Academic Year
2025/2026

ملخص.

تُعنى هذه المذكرة بالنمذجة الديناميكية والتحكم في تتبع عالي السرعة لمسار ذراع آلي ذي 6 درجات حرية (PUMA 560) تحت ضغط ديناميكي شديد وعدم استقرار وسيطي. وحدات التحكم *PID* التقليدية، كونها لا تأخذ بعين الاعتبار بطبيعتها البنيوية الديناميكيات غير الخطية والتفاعلات الحركية بين مختلف أجزاء الذراع الآلي، تُظهر أخطاء تتبع كبرى (16.8 مم) وتشبعًا في عزوم المحركات تحت تأثير حمولة غير منمذجة مقدارها 5 كغم.

للتغلب على هذه القيود، تم البحث في استراتيجيتين متعارضتين: التحكم الضبابي لممداني الذي لا يتطلب وجود نموذج ديناميكي للذراع الآلي والذي يوفر متانة ممتازة في التحكم لكنه يفتقر إلى الدقة المطلوبة، بينما التحكم بعزم الدوران المحسوب القائم على النموذج الديناميكي يتميز بدقة مثالية (بخطأ تتبع قدره 0.05 مم فقط) لكنه يفشل فشلا ذريعا في حالة التغيرات الوسيطية غير المنمذجة.

المساهمة الرئيسية لهذا العمل هو اقتراح بنية تحكم هجينة تدمج بشكل تآزري الإلغاء غير الخطي الدقيق المستخلص من النموذج الديناميكي مع تعويض ذكي للتغيرات الوسيطية العشوائية من قبل المعدل الضبابي. في حالة مسارات حلزونية عالية التردد وحمولة غير منمذجة مقدارها 5 كغم، يزيل المتحكم الهجين المقترح تمامًا الانحراف المساري سيما في النظام الدائم كما يحيد بشكل ملفت اهتزازات عزوم المحركات، محققًا خطأ تموضع رائع لا يتجاوز (0.1 مم).

تعتبر نتائج هذا العمل دليلاً قاطعا على أن دقة تتبع المسار المطلقة في البيئات الصناعية المتميزة عموما بتغيرات وسيطية غير متوقعة تتطلب الدمج التآزري بين النمذجة الفيزيائية والذكاء الاصطناعي.

كلمات مفتاحية : ذراع الروبوت PUMA 560 ذو 6 درجات حرية؛ تتبع المسار؛ التحكم بعزم الدوران المحسوب؛ التحكم الضبابي؛ التحكم الهجين؛ التغيرات الوسيطية غير المتوقعة؛ الديناميكيات غير الخطية؛ الصناعة من الجيل الرابع (4.0)؛ المناورات عالية التسارع؛ التتبع السريع للمسارات المعقدة.

Abstract :

This dissertation addresses the dynamic modeling and high-speed trajectory tracking control of a *6-DOF* PUMA 560 robotic manipulator under severe dynamic stress and parametric uncertainty. Classical *PID* controllers, being structurally unaware of nonlinear multi-body dynamics, exhibit serious tracking distortion (16.8 mm error) and actuator saturation under an unmodeled 5 kg payload. To overcome these limitations, two opposing strategies are investigated: Model-free *Mamdani Fuzzy Logic Control* (*FLC*) offers robustness but lacks deterministic precision, while model-based *Computed Torque Control* (*CTC*) delivers perfect nominal precision (0.05 mm error) but fails catastrophically under parametric variations. The main contribution is a hybrid *CTC-FLC* architecture that synergistically integrates exact nonlinear cancellation from the dynamic model with intelligent uncertainty rejection from the fuzzy compensator. Under sustained high-frequency helical maneuvers and an unmodeled 5 kg payload, the hybrid controller entirely eliminates steady-state spatial shift and actuator chattering, achieving a remarkable mean *Cartesian* error of only 0.1 mm. This work provides empirical evidence that absolute spatial accuracy in unpredictable industrial environments requires the synergistic fusion of deterministic physical modeling and artificial intelligence.

Key words: *6-DOF robot arm; PUMA 560; Trajectory tracking; Computed Torque Control; Fuzzy Logic Control; Hybrid control; Parametric uncertainty; Nonlinear dynamics; Industry 4.0; Aggressive maneuvering, Complex trajectory fast tracking .*

Résumé :

Ce mémoire traite de la modélisation dynamique et du contrôle de suivi de trajectoire à haute vitesse d'un manipulateur robotique à 6 degrés de liberté (PUMA 560) soumis à des contraintes dynamiques sévères et à des incertitudes paramétriques. Les régulateurs *PID* classiques, qui ne tiennent pas intrinsèquement compte des dynamiques non linéaires et des couplages entre les différentes parties du bras robotique, présentent des erreurs de poursuite importantes (16.8 mm) ainsi qu'une saturation des actionneurs en présence d'une charge utile non modélisée de 5 kg. Pour surmonter ces limitations, deux stratégies opposées ont été étudiées : Un régulateur flou de type *Mamdani* indépendant du modèle dynamique du manipulateur offre de la robustesse mais manque de précision déterministe, tandis que la commande par couple calculé basée sur le modèle dynamique du robot, assure une précision nominale parfaite (avec une erreur de 0.05 mm) mais échoue catastrophiquement sous les variations paramétriques. La principale contribution est une architecture de commande hybride (couple calculé – floue) qui intègre de manière synergétique la compensation exacte des efforts non linéaires issus du modèle dynamique avec le rejet intelligent des incertitudes provenant du compensateur flou. Sous des manœuvres hélicoïdales soutenues à haute fréquence et une charge utile non modélisée de 5 kg, le contrôleur hybride élimine entièrement l'erreur de poursuite spatiale en régime permanent et le broutement de couple des actionneurs, atteignant une erreur cartésienne moyenne remarquable de seulement (0.1 mm). Ce travail démontre que la précision absolue dans les environnements industriels imprévisibles nécessite la fusion synergétique de la modélisation physique déterministe et de l'intelligence artificielle.

Mots clés*: Bras robotique PUMA 560 à 6 degrés de liberté ; Poursuite de trajectoire ; Commande par couple calculé ; Commande floue ; Commande hybride ; Incertitude paramétrique ; Dynamique non linéaire ; Industrie 4.0 ; Manœuvres agressives ; Suivi rapide de trajectoire complexe.*

Dedication

To the ones who believed in me long before I learned to believe in myself. I dedicate this thesis to my family, whose boundless love and quiet sacrifices have guided me through every obstacle. You are the true architects of this achievement, and this success is as much yours as it is mine.

This memoir is also dedicated to the relentless pursuit of innovation in the field of automation and intelligent systems. May the findings of this research spark curiosity and empower others to continue pushing the boundaries of what is possible.

To the friends, mentors, and everyone who walked this demanding path with me — thank you for being my constant source of light, motivation, and strength. Your presence made all the difference.

SOUAD

Acknowledgment

First and foremost, I express my deepest gratitude to God Almighty for granting me the faith, perseverance, and strength to complete this work.

I extend my sincere appreciation to my supervisor, ***Prof. MOKRANI Lakhdar****, for his invaluable guidance, patience, and dedication throughout the preparation of this dissertation. His insightful feedback and unwavering support were instrumental in shaping this research, and for that, I am profoundly grateful.*

My earnest thanks also go to my co-supervisor, ***Dr. BENLARBI Keltoum****, for her commitment and expertise. Her meticulous guidance, particularly in the modeling aspects of this research and the structural review of our writing, was absolutely indispensable to its success.*

I am deeply honored by the time and effort invested by the members of the examination committee. My sincere thanks go to ***Dr. ABDEREZZAK Hadjer*** *for graciously accepting to preside over the jury, and to our examiner,* ***Dr. RAHMANI Belkacem****, for his thoughtful evaluation and constructive feedback on my work.*

I am truly grateful to my family and friends for their constant support, motivation, and understanding throughout this demanding period. Your presence and belief in me made this achievement possible.

Finally, I would like to acknowledge all the professors and staff of the Faculty of Technology, specifically within the Electrical Engineering Department at Amar Telidji University of Laghouat, for their continuous support and encouragement throughout my academic journey. Your contributions have greatly enriched my learning experience.

Symbols & Acronyms List

Symbols & Acronyms List

- ***Symbols***

Symbol	***Signification***
q	*Joint position vector*
$\dot{q}$	*Joint velocity vector*
$\ddot{q}$	*Joint acceleration vector*
τ	*Applied joint torque vector*
τ_i	*Torque applied to joint 'i'*
$M(q)$	*Inertia matrix*
$C(q,\dot{q})$	*Coriolis and centrifugal matrix*
$G(q)$	*Gravity vector*
$F(\dot{q})$	*Friction torque vector*
τ_{ext}	*External disturbance torque*
K	*Total kinetic energy*
P	*Total potential energy*
L	*Lagrangian function*
$J(q)$	*Jacobian matrix*
w	*Manipulability measure*
T	*Homogeneous transformation matrix*
a_i	*Link length*
α_i	*Link twist*
d_i	*Joint offset*
θ_i	*Joint angle*
x	*State vector*
u	*Control input vector*
q_d	*Desired joint position vector*

$e(t)$	*Tracking error*
$\dot{e}(t)$	*Tracking error derivative*
K_p	*Proportional gain* (*of PID / CTC*)
K_i	*Integral gain*
K_d	*Derivative gain* (*of PID / CTC*)
v	*End-effector velocity* (*Twist*)
R	*Rotation matrix*
p	*Position vector*
t	*Time*
dt	*Simulation time step*
k_1, k_2, k_3 & k_4	*Runge-Kutta intermediate slopes*
β_i	*Quintic polynomial coefficients* (i=0,1,2, …,5)
$A_{1,2,3}$	*Joint* (*1,2, 3*) *oscillation amplitude* (*Case of helical trajectory*)
w	*Hann window envelope of the helical trajectory*
ω	*Angular frequency of the helical trajectory*
m	*Link mass*
Δm	*Payload mass variation*
F_v	*Viscous friction coefficient*
F_c	*Coulomb friction coefficient*
$\Delta\tau$	*Incremental control torque*
$\Delta\tau_{FLC\text{-}PD}$	*Fuzzy PD torque increment*
$\mu(x)$	*Membership function*
$\mu(e)$	*Membership degree of error*
$\mu(\dot{e})$	*Membership degree of error derivative*
$\mu(\Delta\tau)$	*Membership Degree of Output Torque*
K_e	*Error scaling factor*
K_{de}	*Error derivative scaling factor*
K_{PD}, K_{PI}	*Output scaling factors*
μ_{Rk}	*output membership function of the* k^{th} *rule*

q_d	*Desired joint position*
$\dot{q}_d$	*Desired joint velocity*
$\ddot{q}_d$	*Desired joint acceleration*
K_v	*Derivative gain matrix of CTC*
τ_{CTC}	*Computed torque control input*
$\hat{M}(q)$	*Estimated inertia matrix*
$\hat{C}(q,\dot{q})$	*Estimated Coriolis / Centrifugal matrix*
$\hat{G}(q)$	*Estimated gravity vector*
$\hat{F}(\dot{q})$	*Estimated friction vector*
NB	*Negative Big*
NS	*Negative Small*
ZE	*Zero*
PS	*Positive Small*
PB	*Positive Big*

- ***Acronyms***

Acronym	***Signification***
AI	*Artificial Intelligence*
ANN	*Artificial Neural Network*
PID	*Proportional-Integral-Derivative*
F-CTC	*Fuzzy Computed Torque Control*
CTC	*Computed Torque Control*
RK4	*Fourth-Order Runge-Kutta*
D-H	*Denavit-Hartenberg*
DGM	*Direct Geometric Model*
IK	*Inverse Kinematics*
DoF	*Degree of Freedom*
SISO	*Single Input Single Output*
MIMO	*Multi Input Multi Output*
ODE	*Ordinary Differential Equation*

SCARA	*Selective Compliance Assembly Robot Arm*
UR	*Universal Robots*
ISO	*International Organization for Standardization*
Cobot	*Collaborative Robot*
F/T	*Force/Torque*
PUMA	*Programmable Universal Machine for Assembly*
FLC	*Fuzzy Logic Controller*
FLC-PID	*Fuzzy Logic PID Controller*
PD-FLC	*Proportional-Derivative Fuzzy Logic Controller*
MF	*Membership Function*
COG	*Center Of Gravity (Defuzzification)*
CTC	*Computed Torque Control*
CTC-FLC	*Hybrid Computed Torque–Fuzzy Logic Controller*

Figures List

Figures List

Figure	Page
Chapter I State of the Art of Robotic Manipulators: Architectures, Modeling, and Control Approaches	
Figure 1.1 *Pioneering industrial manipulators of the early 1960s*	**6**
Figure 1.2 *Foundational manipulators of the assembly era*	**7**
Figure 1.3 *The original SCARA robot (1979), birth of selective compliance*	**8**
Figure 1.4 *Examples of collaborative robots*	**10**
Figure 1.5 *Kinematic structure of a standard 6-DoF articulated robotic arm*	**12**
Figure 1.6 *SCARA robotic manipulator configuration*	**12**
Figure 1.7 *Delta parallel robotic manipulator architecture*	**13**
Figure 1.8 *Example of revolute and prismatic joints*	**13**
Figure 1.9 *Robot kinematic chains and coordinate frame assignment*	**15**
Figure 1.10 *Block diagram of a model-based Computed Torque Control architecture*	**18**
Figure 1.11 *Architecture of a Fuzzy Logic Control system*	**20**
Figure 1.12 *Common shapes of MFs used in Fuzzy Logic Control*	**21**
Figure 1.13 *Universal collaborative 6-DoF robot arm UR10e with integrated joint torque sensing*	**24**
Figure 1.14 *KUKA KR 6 R900 AGILUS, a high-speed 6-DoF industrial manipulator*	**25**
Figure 1.15 *ABB IRB 6700 heavy-duty 6-DoF manipulator for automotive welding applications*	**25**
Figure 1.16 *FANUC LR Mate 200iD compact 6-DoF robot for precision assembly and cleanroom applications*	**26**
Figure 1.17 *Yaskawa Motoman HC10DT collaborative robot with full-body torque*	**27**
Figure 1.18 *Franka Emika Panda research-5TFgrade 7-DoF manipulator with full joint torque sensing*	**28**

Chapter II Kinematic and Dynamic Modeling of Robotic Manipulators	
Figure 2.1 *Structural block diagram representation of the nonlinear rigid-body dynamics of an n-DoF robot manipulator*	**35**
Figure 2.2 *A single rigid link rotating about a fixed base joint in xOy plane*	**36**
Figure 2.3 *Physical layout and mechanical structure of the PUMA 560 manipulator*	**39**
Figure 2.4 *D-H coordinate frame assignment (X, Y, Z axes) for the PUMA 560*	**39**
Chapter III Classical PID and Model-Based Control of the PUMA 560 Robot Arm	
Figure 3.1 *Joint space trajectories – PID / Test I*	**50**
Figure 3.2 *Joint position tracking errors – PID / Test I*	**50**
Figure 3.3 *Joint torques – PID / Test I*	**51**
Figure 3.4 *3D end-effector trajectory – PID / Test I*	**52**
Figure 3.5 *Euclidean position error – PID / Test I*	**52**
Figure 3.6 *Joint space trajectories – PID / Test II*	**53**
Figure 3.7 *Joint position tracking errors – PID / Test II*	**54**
Figure 3.8 *Actuator control torques – PID / Test II*	**55**
Figure 3.9 *3D end-effector trajectory – PID / Test II*	**56**
Figure 3.10 *End-effector Cartesian position error – PID / Test II*	**56**
Figure 3.11 *Joint space trajectories – PID / Test III*	**57**
Figure 3.12 *Joint position tracking errors – PID / Test III*	**58**
Figure 3.13 *Actuator control torques – PID / Test III*	**59**
Figure 3.14 *3D end-effector trajectory – PID / Test III*	**60**
Figure 3.15 *End-effector Cartesian position error – PID / Test III*	**60**

Chapter IV Advanced Non-Linear and Intelligent Control of the PUMA 560 Robot Arm

Figure 4.1 *Architecture of the Mamdani PD-FLC*	**64**
Figure 4.2 *Membership functions of the FLC inputs/output*	**65**
Figure 4.3 *Block diagram of the FLC-PID*	**66**
Figure 4.4 *Joint trajectories – FLC / Test I*	**67**
Figure 4.5 *Tracking error evolution – FLC / Test I*	**68**
Figure 4.6 *Joint torques – FLC / Test I*	**69**
Figure 4.7 *3D end-effector trajectory – FLC / Test I*	**70**
Figure 4.8 *Euclidean position error – FLC / Test I*	**70**
Figure 4.9 *Joint trajectories – FLC / Test II*	**71**
Figure 4.10 *Tracking error evolution – FLC / Test II*	**72**
Figure 4.11 *Joint torques – FLC / Test II*	**72**
Figure 4.12 *3D End-Effector Trajectory – FLC / Test II*	**73**
Figure 4.13 *Euclidean position error – FLC / Test II*	**74**
Figure 4.14 *Joint trajectories – FLC / Test III*	**74**
Figure 4.15 *Tracking error evolution – FLC / Test III*	**75**
Figure 4.16 *Joint torques – FLC / Test III*	**76**
Figure 4.17 *3D end-effector trajectory – FLC / Test III*	**76**
Figure 4.18 *Euclidean position error – FLC / Test III*	**77**
Figure 4.19 *Joint trajectories – CTC / Test I*	**79**
Figure 4.20 *Tracking error evolution – CTC / Test I*	**80**
Figure 4.21 *Joint torques – CTC / Test I*	**81**
Figure 4.22 *3D end-effector trajectory – CTC / Test I*	**81**
Figure 4.23 *Euclidean position error – CTC / Test I*	**82**
Figure 4.24 *Joint trajectories – CTC / Test II*	**83**

Figure 4.25 *Tracking error evolution – CTC / Test II*	**83**
Figure 4.26 *Joint torques – CTC / Test II*	**84**
Figure 4.27 *3D end-effector trajectory – CTC / Test II*	**85**
Figure 4.28 *Euclidean position error – CTC / Test II*	**85**
Figure 4.29 *Joint trajectories – CTC / Test III*	**86**
Figure 4.30 *Tracking error evolution – CTC / Test III*	**87**
Figure 4.31 *Joint torques – CTC / Test III*	**87**
Figure 4.32 *3D end-effector trajectory – CTC / Test III*	**88**
Figure 4.33 *Euclidean position error – CTC / Test III*	**88**
Figure 4.34 *Joint trajectories – CTC-FLC / Test I*	**91**
Figure 4.35 *Tracking error evolution – CTC-FLC / Test I*	**91**
Figure 4.36 *Joint torques – CTC-FLC / Test I*	**92**
Figure 4.37 *3D end-effector trajectory – CTC-FLC / Test I*	**92**
Figure 4.38 *Euclidean position error – CTC-FLC / Test I*	**93**
Figure 4.39 *Joint trajectories – CTC-FLC / Test II*	**94**
Figure 4.40 *Tracking error evolution – CTC-FLC / Test II*	**94**
Figure 4.41 *Joint torques – CTC-FLC / Test II*	**95**
Figure 4.42 *3D end-effector trajectory – CTC-FLC / Test II*	**95**
Figure 4.43 *Euclidean position error – CTC-FLC / Test II*	**96**
Figure 4.44 *Joint trajectories – CTC-FLC / Test III*	**97**
Figure 4.45 *Tracking error evolution – CTC-FLC / Test III*	**97**
Figure 4.46 *Joint torques – CTC-FLC / Test III*	**98**
Figure 4.47 *3D end-effector trajectory – CTC-FLC / Test III*	**98**
Figure 4.48 *Euclidean position error – CTC-FLC / Test III*	**99**
Figure 4.49 *Macroscopic comparison of mean Cartesian errors across all evaluated control architectures*	**101**

Tables List

Tables List

Table	Page
Chapter II Kinematic and Dynamic Modeling of Robotic Manipulators	
Table 2.1 Standard D-H Parameters for the PUMA 560	40
Chapter IV Advanced Non-Linear and Intelligent Control of the PUMA 560 Robot Arm	
Table 4.1 Anti-diagonal inference matrix of a PD Fuzzy controller	65
Table 4.2 Synthesis of mean and maximum Cartesian errors across all test scenarios	101
Appendix	
Table A.1 Denavit-Hartenberg (D-H) parameters	113
Table A.2 Dynamic parameters (Mass, Base Inertia, and Friction)	113
Table A.3 PID controller gains across all test scenarios	114
Table A.4 CTC gains across all test scenarios	114
Table A.5 FLC scaling factors across all test scenario	115
Table A.6 Hybrid CTC-FLC scaling factors across all test scenarios	115
Table A.7 Initial and final joint positions for the reference trajectory	116
Table A.8 Numerical values of the quintic polynomial coefficients ($t_f = 6.0\ s$)	116
Table A.9 Numerical parameters of the 3-loop helical trajectory	117

Content

Title *Page*

General Introduction *1*

Chapter I State of the Art of Robotic Manipulators: Architectures, Modeling, and Control Approaches

I.1 INTRODUCTION 5

I.2 HISTORICAL EVOLUTION OF THE ROBOTIC ARM 5

I.2.1 INDUSTRIAL ORIGINS 6

I.2.2 THE AGE OF ASSEMBLY 6

I.2.3 STANDARDIZATION AND MODULARITY 8

I.2.4 THE COLLABORATIVE ERA 9

I.3 MECHANICAL AND MECHATRONIC DESIGN OF MANIPULATORS 11

I.3.1 KINEMATIC CONFIGURATIONS 11

I.3.2 JOINT AND LINK DESIGN 13

I.3.3 INTEGRATED SENSING 14

I.4 MODELING THE MANIPULATOR DYNAMICS 14

I.4.1 THE RIGID BODY DYNAMICS MODEL 14

I.4.2 FRICTION MODELING 15

I.4.3 ADDRESSING JOINT FLEXIBILITY 16

I.5 CONTROL ARCHITECTURES FOR MANIPULATORS 16

I.5.1 INDEPENDENT JOINT CONTROL 17

I.5.2 Computed Torque Control 17

I.5.3 Advanced Non-Linear Control 18

I.5.4 Comparative Analysis of Control Architectures 19

I.6 Intelligent Control Strategies for Manipulators 19

I.6.1 Neural Network and Learning-Based Control 19

I.6.2 Fuzzy Logic Control 20

I.7 Research Gap and Contribution Positioning 22

I.7.1 Comparative Analysis of Control Methodologies 22

I.7.2 Persistent Challenges in Arm Control Addressed by Fuzzy Logic 22

I.7.3 The Research Gap and Work Positioning 22

I.8 Overview of Multi-DOF Robotic Manipulators 23

I.8.1 Universal Robot UR10e 23

I.8.2 KUKA KR 6 R900 AGILUS Industrial Robot 24

I.8.3 ABB IRB 6700 Industrial Manipulator 25

I.8.4 FANUC LR Mate 200iD Multi-Application Robot 26

I.8.5 Yaskawa Motoman HC10DT Dual-Arm Cobot 26

I.8.6 Franka Emika Panda Torque-Controlled Robot 27

I.9 Comparative Analysis and Classification of Modern 6-DoF Robots 28

I.9.1 Classification by Application Domain 28

I.9.2 Emerging Trends 29

I.10 Conclusion 30

Chapter II Kinematic and Dynamic Modeling of Robotic Manipulators

II .1 INTRODUCTION 32

II.2 GENERAL KINEMATIC MODELING OF AN N-DOF MANIPULATOR 32

II.2.1 COORDINATE FRAMES AND SPATIAL TRANSFORMATIONS 32

II.2.2 THE DENAVIT–HARTENBERG (D-H) CONVENTION 33

II.2.3 FORWARD KINEMATICS FORMULATION 33

II.2.4 INVERSE KINEMATICS PROBLEM 33

II.2.5 DIFFERENTIAL KINEMATICS AND THE JACOBIAN MATRIX 34

II.3 GENERAL DYNAMIC MODELING OF AN N-DOF MANIPULATOR 34

II.3.1 KINETIC AND POTENTIAL ENERGY FORMULATION 34

II.3.2 EULER-LAGRANGE EQUATIONS OF MOTION 35

II.3.3 STANDARD MATRIX REPRESENTATION 35

II.4 PEDAGOGICAL CASE STUDY: 1-DOF REVOLUTE MANIPULATOR 36

II.4.1 KINEMATICS AND JACOBIAN SUB-MATRIX DERIVATION 36

II.4.2 LAGRANGIAN DYNAMICS AND ENERGY FORMULATION 37

II.5 KINEMATIC MODELING OF THE 6-DOF PUMA 560 MANIPULATOR 38

II.5.1 ACADEMIC JUSTIFICATION FOR THE BENCHMARK SELECTION 38

II.5.2 DESCRIPTION OF THE 6-DOF ARCHITECTURE 39

II.5.3 ASSIGNMENT OF D-H COORDINATE FRAME 39

II.5.4 DERIVATION OF THE D-H PARAMETERS TABLE 40

II.5.5 DERIVATION OF THE KINEMATIC MATRICES 40

II.6 FORMULATION OF THE 6-DOF DYNAMIC EQUATIONS 41

II.6.1 INERTIA AND CORIOLIS DYNAMIC COUPLING 41

II.6.2 INCORPORATING UNMODELED DYNAMICS AND JOINT FRICTION MODELS 41

II .7 CONCLUSION 42

Chapter III Classical PID and Model-Based Control of the PUMA 560 Robot Arm

III.1 INTRODUCTION 44

III.2 STATE-SPACE REPRESENTATION OF THE MANIPULATOR DYNAMICS 45

III.3 FORMULATION OF THE CONTROL PROBLEM 46

III.3.1 DECENTRALIZED TRACKING ERROR FORMULATIONS 46

III.3.2 STRUCTURE OF THE PID CONTROLLER 46

III.3.3 MODEL-BASED GRAVITY FEED-FORWARD COMPENSATION 46

III.4 NUMERICAL SIMULATION AND ALGORITHMIC FLOW 47

III.4.1 TRAJECTORY GENERATION: QUINTIC POLYNOMIAL / 3-LOOP HELICAL PATH 47

III.4.2. FOURTH-ORDER RUNGE-KUTTA INTEGRATION SOLVER 48

III.5 THE THREE SIMULATION SCENARIOS AND RESULTS ANALYSIS 49

III.5.1 FIRST TEST 'TRACKING PERFORMANCE UNDER BASELINE NOMINAL CONDITIONS' 49

III.5.2 SECOND TEST 'HIGH-DYNAMIC TRACKING OF A 3D HELICAL TRAJECTORY' 53

III.5.3 THIRD TEST 'ROBUSTNESS UNDER UNMODELED PAYLOAD DURING HIGH-DYNAMIC SPATIAL TRACKING' 57

III.6 CONCLUSION 61

Chapter IV Advanced Non-Linear and Intelligent Control of the PUMA 560 Robot Arm

IV.1 INTRODUCTION 63

IV.2 FUZZY LOGIC PID CONTROL 63

IV.2.1 THEORY AND MATHEMATICAL ARCHITECTURE OF THE MAMDANI FUZZY PID 64

IV.2.2 SIMULATION RESULTS AND DISCUSSION 66

IV.2.2.1 TEST I 'TRACKING PERFORMANCE UNDER BASELINE NOMINAL CONDITIONS' 67

IV.2.2.2 TEST II 'HIGH-DYNAMIC HELICAL TRAJECTORY TRACKING' 71

IV.2.2.3 TEST III 'ROBUSTNESS UNDER UNMODELED PAYLOAD DURING HIGH-DYNAMIC SPATIAL TRACKING' 74

IV.3 COMPUTED TORQUE CONTROL 77

IV.3.1 THEORY AND ARCHITECTURE 77

V.3.2 SIMULATION RESULTS AND DISCUSSION 79

IV.3.2.1 TEST I 'TRACKING PERFORMANCE UNDER BASELINE NOMINAL CONDITIONS' 79

IV.3.2.2 TEST II 'HIGH-DYNAMIC HELICAL TRAJECTORY TRACKING' 82

IV.3.2.3 TEST III 'ROBUSTNESS UNDER UNMODELED PAYLOAD DURING HIGH-DYNAMIC SPATIAL TRACKING' 86

IV.4 HYBRID COMPUTED TORQUE - FUZZY LOGIC CONTROL 89

IV.4.1 THEORY AND ARCHITECTURE 89

IV.4.2 SIMULATION RESULTS AND DISCUSSION 90

IV. 4.2.1 TEST I 'TRACKING PERFORMANCE UNDER BASELINE NOMINAL CONDITIONS' 90

IV.4.2.2 TEST II 'HIGH-DYNAMIC HELICAL TRAJECTORY TRACKING' 93

IV.4.2.3 TEST III 'ROBUSTNESS UNDER UNMODELED PAYLOAD DURING HIGH-DYNAMIC SPATIAL TRACKING' 96

IV.5 COMPARATIVE ANALYSIS 99

IV.5.1 CLASSICAL PID VERSUS MAMDANI FLC 99

IV.5.2 BASIC CTC VERSUS HYBRID CTC-FLC 100

IV.5.3 MACROSCOPIC DATA SYNTHESIS 100

IV.6 CONCLUSION 102

General Conclusion 103

Bibliography 106

Appendix 111

General Introduction

General Introduction

The emergence and ongoing development of industrial robotic manipulators have revolutionized the world of contemporary manufacturing, laying the foundation for automated assembly, precision machining, and smart production lines. They are designed as multi-degree-of-freedom (*DOF*) spatial mechanisms which are intended to emulate, and often surpass, the skills and toughness of the human arm. However, as the guiding principles of Industry 4.0 push industrial demands to exponentially higher operating speeds, heavier payloads and microscopic precision, the control of these manipulators has become a formidable engineering challenge. The multi-link robotic arm is an inherently highly coupled multi-variable and severely non-linear dynamical system in the physical domain. When the manipulator is forced to perform rapid, high-frequency spatial maneuvers, its structural links introduce massive state-dependent inertia variations. These are compounded by disturbance centrifugal, *Coriolis*, and gravitational forces that constantly and actively attempt to push the end-effector off its desired geometric path.

Industrial automation has been based on decentralized linear feedback control architectures for decades. The Proportional-Integral-Derivative (*PID*) controller is the dominant controller. These model-free loops are computationally simple and operationally reliable for slow, quasi-static, point-to-point positioning, but are structurally blind to the underlying multi-body physics of the robotic plant. Consequently, these classical architectures critically degrade for dynamic stress, high-speed continuous contouring, or unmodeled parametric uncertainties (such as picking up an unknown payload). They show extreme distortion of the spatial trajectories, destructive mechanical chattering on the actuators and persistent steady state tracking errors. Control techniques must evolve beyond static linear feedback and move towards model-based, dynamically aware and intelligent compensation strategies in order to bridge the gap between theoretical mathematical precision and the unpredictable demands of real-world industrial applications.

To tackle these engineering challenges, this dissertation aims at systematically modeling, critically analyzing and considerably improving the high-speed trajectory tracking performance of a *6-DOF* robotic manipulator. This work takes the Unimation PUMA 560 as a validated academic benchmark to explore the evolution of robotic control methods, from classical linear feedback, over deterministic inverse dynamics, to *AI* techniques.

This work concludes by synthesizing, simulating and verifying a highly robust hybrid control architecture for guaranteeing absolute spatial accuracy while exhibiting superior robustness with respect to large parametric uncertainties.

To systematically reach these goals, the dissertation is organized to follow four sequential and interrelated chapters:

- ***Chapter I*** is dedicated to the state of the art in robotic manipulation. It presents the fundamental background of the study by examining the physical structures of the industrial manipulators, their operational relevance and a comprehensive review of the classical linear and modern non-linear control strategies employed in academia and industry.

- ***Chapter II*** lays the mathematical foundation needed for advanced control synthesis. It presents the exact spatial mapping of the manipulator through the forward and inverse kinematics with the *Denavit-Hartenberg* (*D-H*) convention. Also, the exact highly coupled non-linear equations of motion are derived by the *Euler-Lagrange* energy formulation, which explicitly show the dynamic complexities of the plant.

- ***Chapter III*** offers a critical evaluation of the industrial baseline. This chapter implements a decentralized *PID* control architecture on the PUMA 560 dynamic model, and empirically demonstrates the physical limitations, energy inefficiencies and critical tracking failures of linear systems when subjected to high speed maneuvers and dynamic stress.

- ***Chapter IV*** is the main contribution of this work. In this chapter, advanced non-linear control algorithms are introduced and benchmarked, namely the model-free *Mamdani Fuzzy Logic Control* (*FLC*) and the model-based *Computed Torque Control* (*CTC*). Finally, the architecture of a hybrid *CTC-FLC* is proposed and validated. This chapter shows by means of severe stress testing the ultimate capability of the architecture to synergistically integrate the absolute mathematical determinism of exact feedback linearization and the heuristic, uncertainty-rejecting intelligence of fuzzy logic, realizing a complete absence of tracking errors in the presence of drastic payload variations.

The dissertation terminates with a general conclusion that recapitulates the key findings of each chapter and outlines promising perspectives for future research.

Chapter I

State of the Art of Robotic Manipulators: Architectures, Modeling, and Control Approaches

I.1 INTRODUCTION

The field of robotics has transcended its initial boundaries of simple automation to become a cornerstone of the modern industrial landscape. A robotic manipulator is a fundamental mechatronic system engineered to replicate, and in many cases exceed, the functional capabilities of the human upper limb, with its primary purpose being the spatial manipulation of objects through a sequence of coordinated motions [1], [2].

Designed to perform tasks requiring agility, precision and high payload capacity, the manipulator serves as the physical agent that translates digital control algorithms into meaningful work within a three-dimensional operational space, from heavy-duty industrial welding to delicate micro-assembly. Ultimately, a robotic manipulator is not merely a mechanical assembly of links and joints; it is the physical concretization of complex mathematical algorithms.

The true elegance of modern robotic engineering lies in harmonizing the rigid laws of *Newtonian* physics with the adaptable, cognitive nature of computational intelligence, a transition from blind, repetitive automation to sensory-rich, interactive manipulation that is the driving force behind contemporary control theory.

However, the synthesis of control for robotic arms remains one of the most challenging problems in engineering due to several intrinsic factors:

- the highly non-linear dynamics, where the relationship between joint torques and motion is governed by complex differential equations that change with the robot's configuration [3];
- the dynamic coupling in multi-DoF systems, where motion in one joint induces inertial forces, *Coriolis*, and centripetal effects on all other joints;
- and the presence of uncertainty and singularities, including unmodeled friction, payload variations, and kinematic singularities that pose a risk to stability.

This chapter establishes the fundamental context of the dissertation by providing a focused review of the evolution, physical constitution, mathematical representation, and control methodologies exclusive to multi-DoF robotic arms. To this end, the chapter is organized as follows: it first traces the historical evolution of manipulators, then details their mechatronic design, followed by a rigorous presentation of mathematical modeling, after which the main control methodologies are reviewed, and finally, the research gap addressed in this thesis is highlighted.

I.2 HISTORICAL EVOLUTION OF THE ROBOTIC ARM

Building upon the fundamental architectures and challenges outlined previously, understanding the present capabilities of robotic manipulators requires a brief examination of their technological development. The evolution of the robotic arm is not only a chronicle of improved hardware, but a story of paradigm shifts driven by industrial demand, economic pressure, and advances in control theory. From their origins as heavy, hydraulic-powered machines confined to safety cages, through the standardization brought by assembly lines and modular designs, to the current era of intelligent, human-collaborative systems, each phase has fundamentally redefined what a manipulator is and what it can do.

This section traces this four-stage trajectory, highlighting how the physical design and control priorities of each era have converged to form the state of the art we analyze today.

I.2.1 Industrial Origins

The origins of robotic manipulators can be traced back to the early 1960s with the deployment of the '*Unimate*' and '*Versatran*' systems, marking the first time a programmable, electrically powered manipulator was integrated into a production line (see figure 1.1). These early machines focused purely on positional repeatability rather than adaptability, executing simple '*pick-and-place*' and '*spot-welding*' operations with hydraulic actuation and rudimentary limit-switch or drum sequencer control [1].

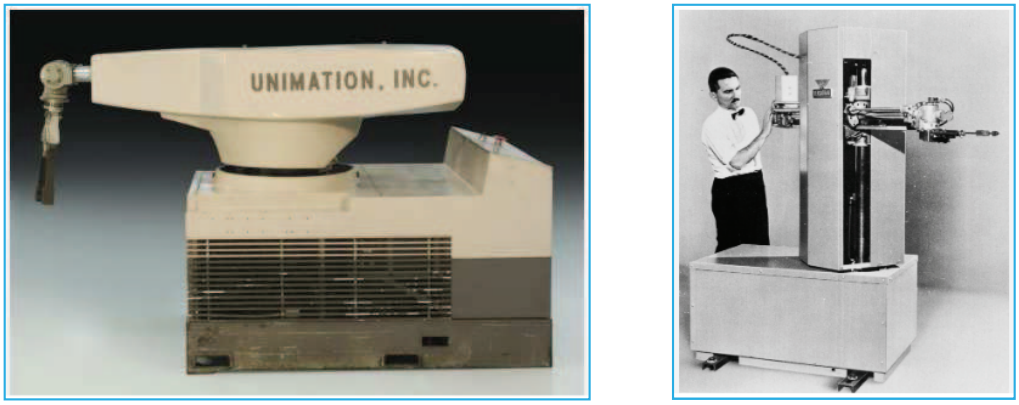


Figure 1.1 *Pioneering industrial manipulators of the early 1960s* [1]
a) Unimate (*1961*), *the first programmable robotic arm b) Versatran* (*1962*), *the first commercially available hydraulic manipulator*

Mechanical robustness was prioritized over sensory integration; as a result, these manipulators operated as blind, open-loop systems with no real-time feedback from force, torque, or vision sensors. Consequently, dynamic coupling effects, such as inertial interactions between joints or velocity-dependent *Coriolis* and centripetal forces, were not explicitly modeled in the control loop. Instead, controllers relied on joint-independent *PID* regulation, often accompanied by slow cycle times and noticeable overshoot or vibration during acceleration. Despite these limitations, the Unimate established the foundational concept of a reprogrammable manipulator, proving that a general-purpose arm could replace hard automation for dangerous, repetitive tasks and automotive assembly. This era set the technical trajectory for subsequent generations by demonstrating industrial viability, even as it highlighted the need for better dynamic modeling and sensory feedback.

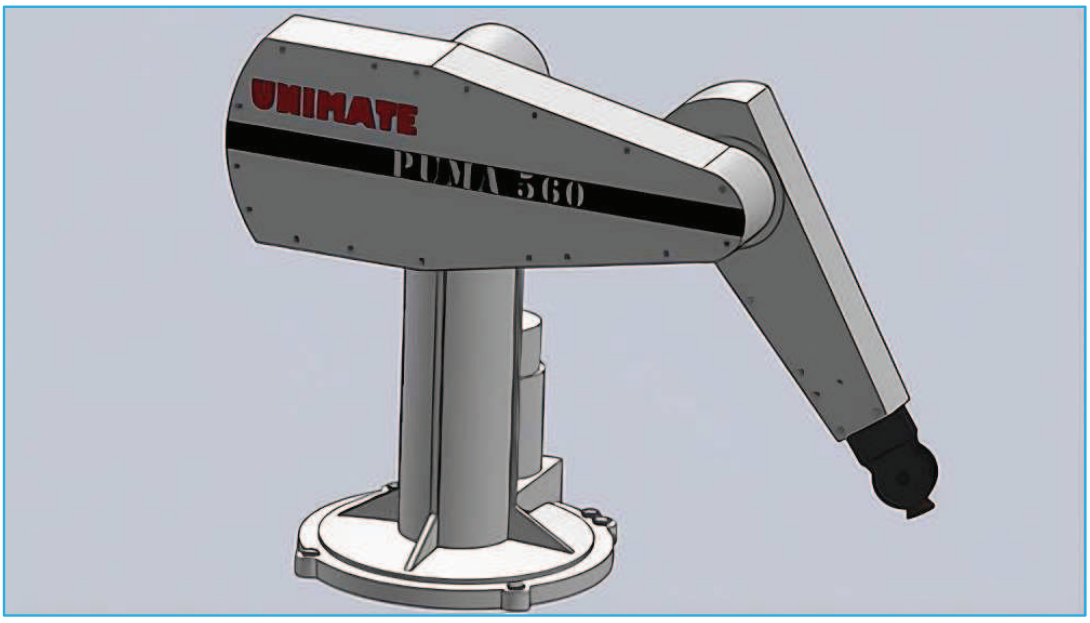
I.2.2 The Age of Assembly

The 1970s marked a pivotal stage with the emergence of the '*Stanford Arm*' and the *PUMA* (Programmable Universal Machine for Assembly), fundamentally shifting the role of manipulators from heavy industrial tasks to precision assembly operations [1]. Developed by Victor Scheinman in 1969, the '*Stanford Arm*' was an all-electric, six-axis articulated robot designed to permit a closed-form kinematic solution, enabling accurate spatial path following under computer control (see figure 1.2.a).

This breakthrough directly led to the *PUMA*, commercialized by Unimation in 1977 with support from General Motors, which retained the six-axis articulated architecture and added a spherical wrist configuration ideal for delicate assembly tasks such as inserting light bulbs or electronic components (see figure 1.2.b)). This transition highlighted the need for absolute accuracy and smooth trajectory planning, as assembly demanded not just positional repeatability but continuous, vibration-free motion through intermediate points.



a) Stanford arm (1969), all-electric, six-axis articulated prototype



b) The Unimate PUMA 560 (1977), a foundational 6-DoF industrial robotic arm

Figure 1.2 *Foundational manipulators of the assembly era* [1]

To meet these demands, the transition to assembly tasks required motion to be mathematically defined to ensure C^2 continuity (position, velocity, and acceleration) using quintic polynomial splines [3]:

$$q(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \beta_4 t^4 + \beta_5 t^5 \qquad (1.1)$$

From an engineering perspective, this historical transition represented a monumental leap. Early designers shifted their focus from merely increasing motor torque to optimizing control loop bandwidth and reducing structural inertia.

The story of how robots have evolved is essentially an ongoing effort to make the robot's real movement match its programmed path as closely as possible. This effort pushed engineers to move away from stiff, purely mechanical designs toward more adaptable systems that use sensors and feedback.

I.2.3 Standardization and Modularity

The maturation of the robotics industry led to the convergence of diverse mechanical experiments into standardized kinematic configurations. The 1980s saw the emergence of the Articulated Manipulator and the SCARA (Selective Compliance Assembly Robot Arm) as dominant industry standards [2]. The SCARA was invented by Professor Hiroshi Makino of Yamanashi University, Japan, who conceived the idea in December 1977, just two months after witnessing the SIGMA assembly robot presented at the 7th International Symposium on Industrial Robots in Tokyo (see figure 1.3). The first prototype was built in 1978, followed by a second improved prototype in 1980. Recognizing its commercial potential, Makino organized the SCARA Robot Consortium, bringing together Yamanashi University and thirteen Japanese companies for a three-year collaboration from April 1978 to March 1981. By 1981, multiple industrial partners began marketing their own versions, and by 1982, IBM had licensed the design to sell as the IBM 7535, the first international recognition of the SCARA robot.

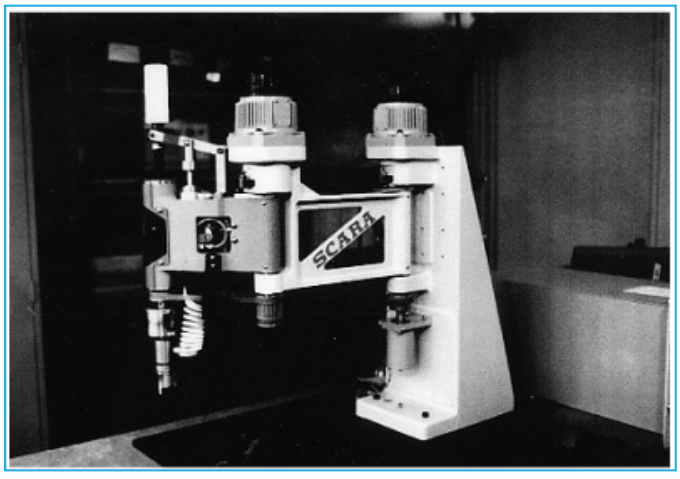


Figure 1.3 *The original SCARA robot (1979), birth of selective compliance* [*W.1*]

The defining feature of the SCARA was its ‘selective compliance’: high rigidity in the vertical (Z) direction for precise insertion tasks, yet intentional flexibility in the horizontal plane to accommodate minor misalignments during assembly.

This unique mechanical characteristic, inspired by the *Japanese* folding screen (Byobu), directly addressed the needs of automated assembly lines where parts must be mated without jamming or damaging delicate components. By the late 1980s, SCARA robots had become a global standard, particularly in electronics and watch assembly, where Seiko Epson successfully deployed them to assemble over 100 watch models on a single automated line, reducing workforce by 40 workers and cutting costs by over 60%.

Concurrent with standardization, the concept of modularity emerged as a parallel trend. Unlike fixed-configuration industrial robots, modular robots are built from interchangeable mechanical units (links, joints, actuators, and sensors) that can be rapidly reconfigured into different kinematic arrangements. This approach, pioneered in the late 1980s by researchers such as Fukuda with the CEBOT (Cellular Robot) system, allowed a single set of modules to serve multiple applications, reducing development time and capital costs.

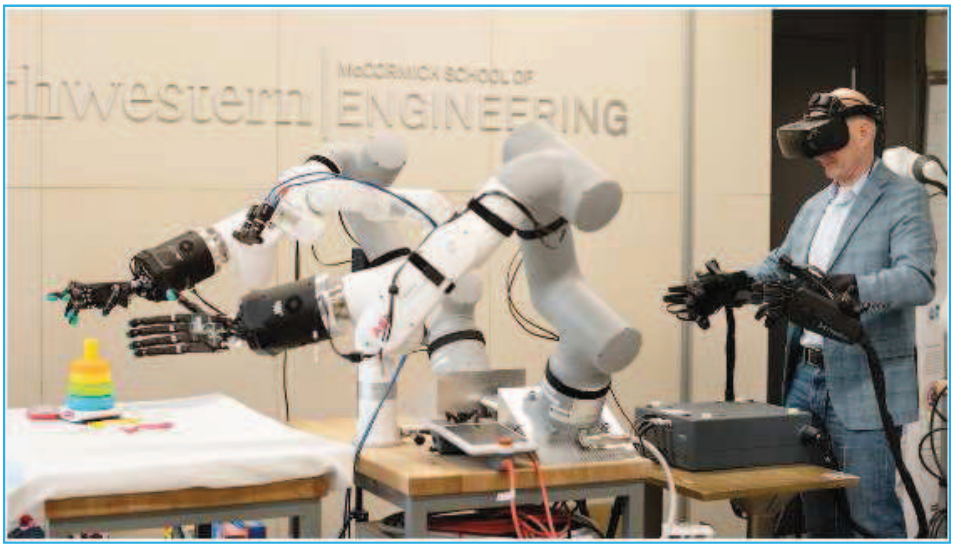
Modularity enabled rapid reconfiguration of links and joints, although it imposed a significant challenge on control theory to handle varying dynamic parameters such as inertia, mass distribution, and natural frequencies that change whenever modules are added, removed, or rearranged [1]. Generalized kinematic modeling methods based on *Denavit-Hartenberg* notation were developed to automatically derive the kinematics of arbitrary module assemblies, yet the problem of real-time adaptive control for reconfigurable manipulators remains an active research area.

On the whole, the 1980s established two complementary strategies: standardization, which fixed the mechanical architecture to optimize control performance for specific tasks (such as SCARA for assembly), and modularity, which prioritized flexibility at the cost of increased control complexity.

I.2.4 THE COLLABORATIVE ERA

The contemporary phase of robotic evolution represents a fundamental shift from keeping robots inside safety cages to allowing them to work directly alongside humans, giving birth to the Collaborative Robot (Cobot). Unlike traditional industrial robots that are heavy, fast, and dangerous, cobots are engineered for intrinsic safety through lightweight structures and built-in torque sensors, which naturally motivates the use of intelligent, soft-computing control systems.

The concept of the cobot originated in the mid-1990s at Northwestern University (see figure 1.4.a)), where Professors *J. Edward Colgate* and *Michael Peshkin*, granted from *General Motors*, introduced the idea of ‘collaborative manipulation’. Their early prototypes were actually passive devices without motors, designed only to guide human movement and reduce physical strain rather than replace the worker. The commercial breakthrough came in the late 2000s when advances in lightweight materials and sensors made practical cobots possible. In 2008, the Danish company Universal Robots released the UR5, the first commercially successful cobot, followed by the larger UR10 (see figure 1.4.b)) in 2012 and the tabletop UR3 in 2015. Other notable examples include KUKA's LBR IIWA (see figure 1.4.c)) and Rethink Robotics' Baxter, both introduced around 2012. All these early cobots shared a common safety principle called Power and Force Limiting: by restricting joint torques and using lightweight materials, the robot cannot physically injure a human upon accidental contact.



a) *Example of cobots from Northwestern University* [*W.2*]



b) *Universal* UR10 *Cobot* [*W.3*]



c) A KUKA *Cobot* [*W.4*]

Figure 1.4 *Examples of collaborative robots*

As cobots became more common, formal safety standards were developed. In 2016, ISO/TS 15066 was issued, providing detailed safety requirements for collaborative applications. This document defines four modes of safe human-robot collaboration: safety-rated monitored stop (the robot stops when a human enters the workspace), hand guiding (the human physically moves the robot to teach it), speed and separation monitoring (the robot slows down as the human approaches), and power and force limiting (the robot cannot exert harmful forces).

The control architectures of cobots have since evolved through three generations. The first generation (2008–2015) relied exclusively on power and force limiting, which restricted payloads to 3-5 kilograms and made cobots suitable only for light tasks. The second generation (2015-2021) introduced integrated force-torque sensors and vision systems, enabling multiple collaborative modes to be combined. The third generation (2021 – present) is defined by software intelligence: embedded *AI* allows cobots to adapt grip force, avoid obstacles, and perform real-time quality inspection without separate sensors.

This shift has encouraged the use of soft-computing control methods, such as fuzzy logic and neural networks, which can learn system behavior without requiring precise mathematical models. In summary, the collaborative era has democratized access to automation for small and medium-sized enterprises while pushing control theory toward adaptive, learning-based approaches capable of handling the unpredictability of shared workspaces.

I.3 MECHANICAL AND MECHATRONIC DESIGN OF MANIPULATORS

The structural topology of a manipulator defines its workspace geometry, while the actuation system determines its dynamic bandwidth. This section examines three core design areas. First, it presents the main kinematic configurations (articulated, SCARA, and Delta arms) and introduces Yoshikawa's manipulability measure for evaluating dexterity and singularity avoidance. Second, it discusses joint and link design, including harmonic drives and the challenge of joint flexibility. Third, it addresses integrated sensing technologies, such as encoders and force-torque sensors, which enable closed-loop control for contact-rich tasks.

I.3.1 KINEMATIC CONFIGURATIONS

Before selecting a specific topological configuration, engineers must evaluate the robot's kinematic dexterity. This is mathematically quantified using Yoshikawa's manipulability measure (w) [8], which evaluates the manipulator's ability to change its end-effector position and orientation. It is derived from the manipulator's *Jacobian* matrix $J(q)$ of joints position q:

$$w = \sqrt{\det\left(J(q)J^T(q)\right)} \qquad (1.2)$$

Physically, this scalar value acts as a distance metric from kinematic singularities. A higher value of w indicates that the robotic arm possesses a greater volume of feasible velocity vectors, allowing it to move freely without mechanical lock-ups. It is a critical parameter when designing the workspace of SCARA robots.

The arrangement of joints in space, known as the kinematic topology, directly determines the robot's manipulability and workspace shape. The three most common configurations used in industry are articulated arms, SCARA robots, and Delta parallel arms.

a) Articulated arms (Anthropomorphic manipulators)

These manipulators typically feature six degrees of freedom (6-DoF) and are designed to replicate the structure and mobility of the human arm (see figure 1.5). Their kinematic architecture provides a large and versatile workspace relative to the robot's physical size, making them suitable for a wide range of industrial tasks such as welding, painting, and material handling. However, this superior workspace comes at the cost of significant mathematical complexity in kinematic and dynamic modeling, due to the coupled, non-linear nature of their joint interactions [2], [4].

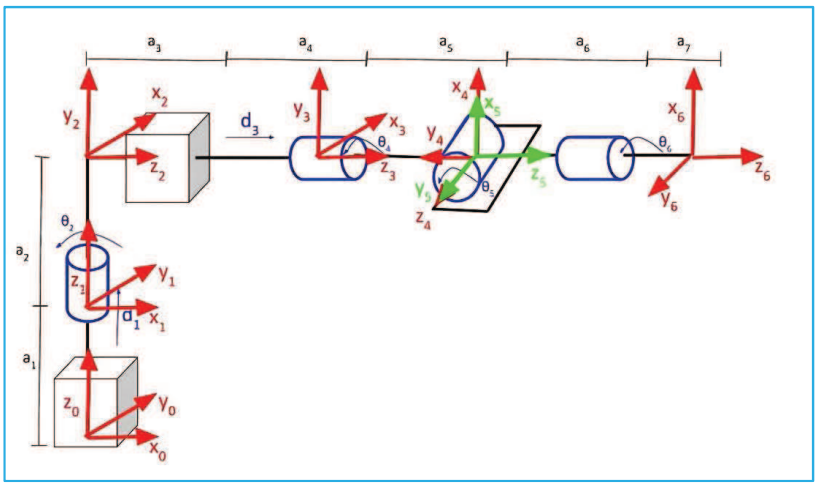


Figure 1.5 *Kinematic structure of a standard 6-DoF articulated robotic arm* [*W.5*]

b) SCARA (Selective Compliance Assembly Robot Arm)

The SCARA manipulator (see figure 1.6) is characterized by its unique selective compliance. It remains compliant (flexible) in the horizontal plane while maintaining extreme stiffness in the vertical axis. This mechanical property allows the robot to accommodate minor misalignments when inserting pins into holes or assembling components, yet resist vertical forces during pressing or insertion tasks. As a result, the SCARA is particularly well-suited for high-speed pick-and-place assembly operations [2].

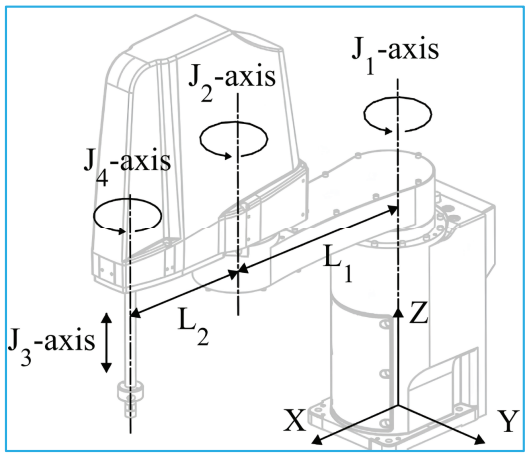


Figure 1.6 *SCARA robotic manipulator configuration* [*W.6*]

c) Delta and parallel arms

Unlike serial manipulators, Delta and other parallel arms utilize a closed-kinematic-chain structure, where multiple independent kinematic links connect the fixed base directly to the moving platform (see figure 1.7). By mounting the heavy actuators on the fixed base rather than on moving links, the moving mass of the structure is significantly reduced. This design enables the robot to achieve very high accelerations and fast cycle times, making parallel arms ideal for high-speed pick-and-place, packaging, and sorting applications [2].

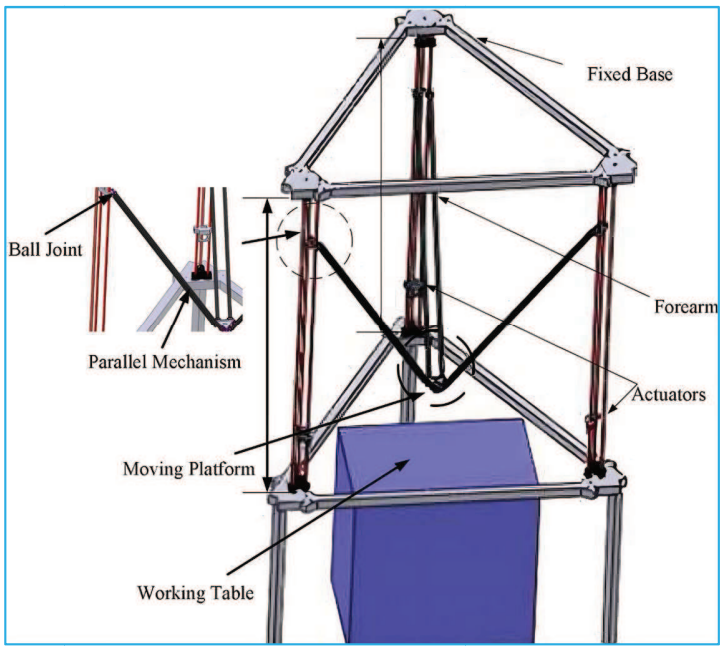


Figure 1.7 *Delta parallel robotic manipulator architecture* [*W.7*]

I.3.2 JOINT AND LINK DESIGN

The mechanical degrees of freedom of a robotic manipulator are achieved primarily through two types of joints: revolute joints, which provide rotational motion about an axis, and prismatic joints, which provide linear sliding motion (see figure 1.8). These joints are connected by rigid links that define the geometric structure of the arm. To transmit motion from the actuators to the joints with high precision, drive systems such as harmonic drives are commonly employed. Harmonic drives offer significant advantages, including compact size, high gear reduction ratios, and near-zero clearance, which eliminates unwanted play between mechanical components. However, they also introduce joint flexibility due to the elastic deformation of their internal ring. This flexibility, if ignored in the control design, can lead to oscillations, reduced tracking accuracy, and even instability under high-speed or high-load conditions. Therefore, advanced control strategies must explicitly account for joint elasticity to ensure stable and precise motion [1], [3].

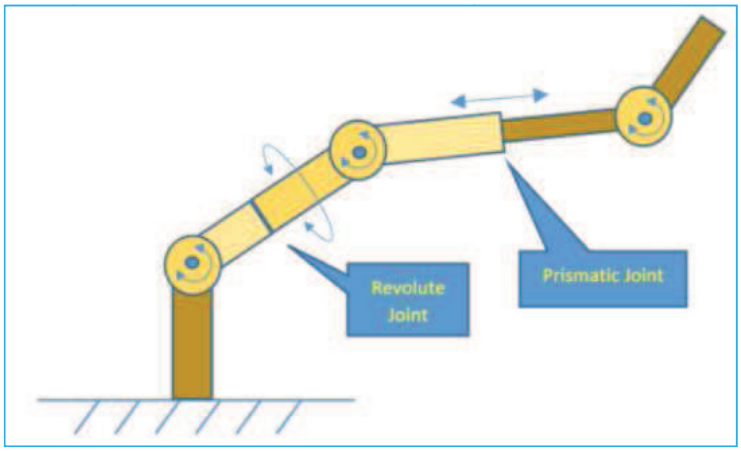


Figure 1.8 *Example of revolute and prismatic joints* [*W.8*]

I.3.3 INTEGRATED SENSING

Closed-loop control of robotic manipulators depends critically on accurate and timely feedback from various sensors mounted on the robot. High-resolution encoders, typically located at each joint, provide precise measurements of joint positions and velocities, forming the backbone of most feedback control schemes. However, position feedback alone is insufficient when the robot interacts physically with its environment, such as during sanding, polishing, or assembly tasks. To address this limitation, wrist-mounted force/torque (*F*/*T*) sensors are used to measure the contact forces and moments exerted by the end-effector. These sensors effectively bridge the gap between the physical mechanics of the robot and the digital control algorithms running in the computer. The measured force information enables advanced interaction control strategies to adjust the robot's motion in response to external forces, allowing the manipulator to comply with its surroundings during contact-rich tasks.

I.4 MODELING THE MANIPULATOR DYNAMICS

Mathematical modeling constitutes the core of robotic control theory. To design an effective controller (whether classical, intelligent, or adaptive), it is essential to first establish rigorous kinematic and dynamic models that describe how the robot moves and responds to forces [5].

Kinematics deals with the geometric relationships of motion (position, velocity, and acceleration) without considering the forces that cause them, while dynamics explicitly accounts for the forces and torques required to produce a desired motion. This section presents the fundamental mathematical tools for modeling rigid-body manipulators. It begins with the *Lagrange-Euler* formulation, which derives the standard dynamic equation relating joint torques to motion. It then addresses parameter identification and friction modeling, including the basic *Coulomb*-viscous model and the more advanced *LuGre* model for low-velocity behavior. Finally, it discusses how to model joint flexibility, a critical issue introduced by harmonic drives and other high-ratio transmissions [3]. Together, these models provide the mathematical foundation upon which all control strategies reviewed later in this chapter are built.

I.4.1 THE RIGID BODY DYNAMICS MODEL

The dynamic behavior of an *n*-link robotic manipulator (see figure 1.9) is systematically derived using the *Lagrange-Euler* formulation [1], [3]. The *Lagrangian* functional *L* is defined as the difference between total kinetic energy *K* and total potential energy *P*:

$$L(q,\dot{q}) = K(q,\dot{q}) - P(q) \qquad (1.3)$$

Where the kinetic energy is expressed as:

$$K(q,\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} \qquad (1.4)$$

And the potential energy is expressed as:

$$P(q) = \sum_{i=0}^{n} m_i\, gh(q_i) \qquad (1.5)$$

Here is the *Euler-Lagrange* equation to apply to the *Lagrangian* functional:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i \; for \; i = 1, \dots, n \qquad (1.6)$$

Substituting the kinetic and potential energy expressions yields the standard robot dynamic model:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_{ext} = \tau \qquad (1.7)$$

Where:

- $M(q) \in R^{n \times n}$ is the symmetric, positive-definite inertia matrix;
- $C(q, \dot{q}) \in R^{n \times n}$ is the *Coriolis* and centripetal matrix, where elements C_{ij} are derived using *Christoffel* symbols [3]:

$$C_{ij}(q,\dot{q}) = \sum_{k=1}^{n} \frac{1}{2}\left(\frac{dM_{ij}}{dq_k} + \frac{dM_{ik}}{dq_j} - \frac{dM_{jk}}{dq_i}\right)\dot{q}_k \qquad (1.8)$$

- $G(q) \in R^n$ is the gravity torque vector;
- $F(\dot{q})$ and τ_{ext} are the friction and external disturbances torques;
- $\tau \in R^n$ is the vector of applied joint torques (actuator torques).

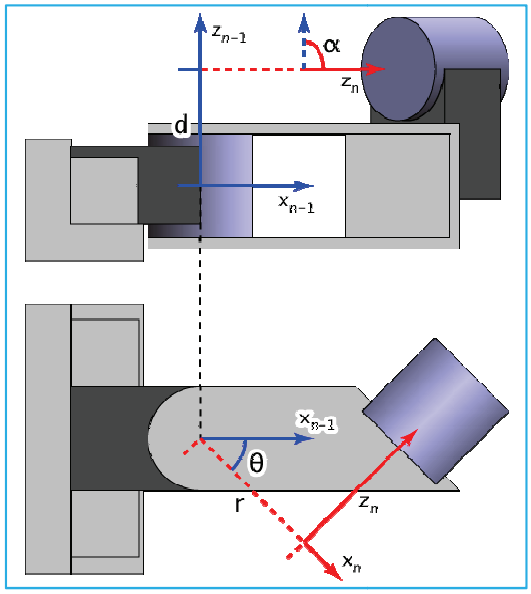


Figure 1.9 *Robot kinematic chains and coordinate frame assignment* [*W.9*]

I.4.2 Friction Modeling

To achieve high-precision control, unstructured uncertainties such as friction must be mathematically modeled. The basic friction model combines *Coulomb* and viscous terms:

$$F(\dot{q}) = F_v\dot{q} + F_c sgn(\dot{q}) \qquad (1.9)$$

Where F_v is the viscous friction coefficient, F_c is the *Coulomb* friction level, and $\dot{q}$ is the angular velocity.

However, this simple model fails to capture low-velocity friction phenomena such as the *Stribeck* effect (friction decreasing with increasing velocity) and pre-sliding displacement. The *LuGre* dynamic model [4] addresses these by introducing an internal state z, which represents the average deflection of elastic bristles between two surfaces:

$$\dot{z} = \dot{q} - \frac{k|\dot{q}|}{g(\dot{q})}z \qquad (1.10)$$

The function $g(\dot{q})$ models the *Stribeck* effect, it is defined as:

$$g(\dot{q}) = F_c + (F_s - F_c)e^{-(\dot{q}/v_s)^2} + \sigma_3\dot{q}$$

Where F_s is the static friction level, and v_s is the *Stribeck* velocity. The total friction torque predicted by the *LuGre* model is then given by [6]:

$$F_{Lugre} = \sigma_0 z + \sigma_1\dot{z} + \sigma_3\dot{q} \qquad (1.11)$$

Here, σ_0 is the bristle stiffness, σ_1 is the bristle damping coefficient, and σ_3 is the viscous friction coefficient (equivalent to F_v in the basic model).

The parameters of these friction models can be identified using standard system identification techniques, such as least-squares curve fitting from constant-velocity experiments or observer-based methods using torque measurements.

I.4.3 Addressing Joint Flexibility

High-ratio transmissions (such as harmonic drives, long shafts) introduce significant joint elasticity. A common approach models the joint as a torsional spring with stiffness K_s, leading to the two-mass flexible joint model [3]. Let q be the link position and θ the motor position (after gear reduction). The link-side dynamics are:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = K_s\,(q - \theta) \qquad (1.12)$$

The right-hand side represents the elastic torque transmitted from the motor to the link.

Elsewhere, the motor-side dynamics is:

$$J_m\ddot{\theta} + B_m\,\dot{\theta} - K_s(q - \theta) = \tau \qquad (1.13)$$

Here, J_m is the motor inertia (reflected through the gearbox), B_m is the motor viscous damping coefficient, and τ is the motor torque input. The term $K_s\,(\theta - q)$ is the spring torque acting on the motor (equal and opposite to that on the link). For improved accuracy, a damping term $B_s(\dot{\theta} - \dot{q})$ can be added, but is omitted here for brevity.

I.5 Control Architectures for Manipulators

Robotic manipulators present a challenging control problem due to their nonlinear, coupled, and uncertain dynamics. Control architectures address these challenges with varying degrees of model exploitation and robustness. This section reviews three representative approaches:

- Independent Joint Control, which ignores coupling for computational efficiency;
- Model-based Computed Torque Control (*CTC*), which linearizes the full dynamics at the cost of model dependence;
- And advanced nonlinear methods (Sliding Mode and Adaptive Control), which trade smoothness or complexity for robustness.

A comparative synthesis highlights the inherent trade-offs that motivate intelligent control frameworks.

I.5.1 INDEPENDENT JOINT CONTROL

The simplest and most widely implemented industrial control scheme is Independent Joint Control (*IJC*), typically realized with a *PID* (Proportional-Integral-Derivative) controller. The tracking error as:

$$e(t) = q_d(t) - q(t)$$

Where $q_d(t)$ is the desired trajectory and $q(t)$ is the actual joint position.

The independent joint control law for each actuator is:

$$\tau(t) = K_p\, e(t) + K_i \int_0^t e(t)dt + K_d \dot{e}(t) \qquad (1.14)$$

Here, $\tau(t)$ is the commanded torque to the motor, while K_p, K_i, and K_d are the proportional, integral, and derivative gains of the controller, respectively.

The primary advantages of *IJC* are computational efficiency, simplicity of tuning (each joint tuned independently), and the absence of a system model. However, because this method completely ignores dynamic coupling between joints, specifically the configuration-dependent inertia matrix $M(q)$, *Coriolis*/centripetal terms $C(q, \dot{q})$, and gravity $G(q)$, tracking errors become significant under high-speed or high-acceleration maneuvers [1]. Nevertheless, *IJC* remains effective for low-speed operations, highly geared transmissions (where coupling is mechanically attenuated), and tasks with relaxed precision requirements.

I.5.2 COMPUTED TORQUE CONTROL

Computed Torque Control (*CTC*) is a model-based technique that globally linearizes the nonlinear robot dynamics [3]. Unlike *IJC*, which ignores coupling effects, *CTC* explicitly compensates for the inertia, *Coriolis*, *gravity*, and friction terms using the system model (see figure 1.10). The control law is formulated as:

$$\tau = M(q)u + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) \quad (1.15)$$

Where u is the auxiliary control input representing the desired closed-loop dynamics, it is given by:

$$u = \ddot{q_d} + K_d\dot{e} + K_p e \qquad (1.16)$$

Here, e is the tracking error, and K_p, K_d are positive definite gain matrices (or positive scalars for single joints).

Substituting the torque into the robot dynamics yields the linear error equation:

$$\ddot{e} + K_d\dot{e} - K_p e = 0 \qquad (1.17)$$

The error dynamics in (1.17) is linear and exponentially stable for any positive K_p, K_d. In the ideal case of perfect model knowledge ($\widehat{M}$=M, $\hat{C}$=C … etc.), *CTC* achieves asymptotic tracking. However, under parametric uncertainty, such as mismatch between the estimated inertia matrix $\widehat{M}$ and the true M, the cancellation is imperfect. Residual nonlinearities reappear, and tracking performance degrades significantly [1]. This sensitivity to model fidelity is the primary limitation of standard *CTC*, motivating robust and adaptive extensions.

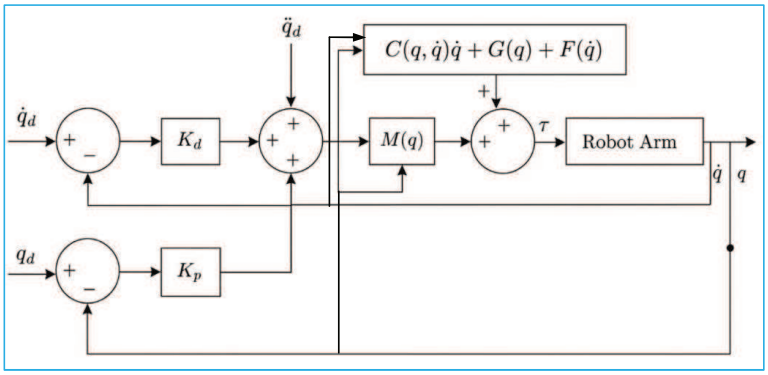


Figure 1.10 *Block diagram of a model-based Computed Torque Control architecture*

I.5.3 ADVANCED NON-LINEAR CONTROL

When model uncertainties are too large for *CTC* to maintain acceptable performance, advanced nonlinear methods offer robust alternatives. Two representative approaches are Sliding Mode Control (*SMC*) and Adaptive Control.

SMC handles bounded uncertainties by forcing the system trajectory onto a user-defined sliding surface and maintaining it there thereafter [7]. For a robot manipulator, the sliding surface is defined as:

$$S = \dot{e} + \lambda e \quad (\lambda > 0) \qquad (1.18)$$

Where λ determines the surface bandwidth. Once on the surface (S=0), the error dynamics satisfy $\dot{e}$=−λe, converging exponentially to zero.

The *SMC* control law consists of an equivalent control term τ_{eq} and a discontinuous switching term:

$$\tau = \tau_{eq} - Ksgn(S) \quad (\lambda > 0) \qquad (1.19)$$

The switching gain K must exceed the magnitude of the model uncertainties to satisfy the reaching condition ($S\dot{S}$ <0). While highly robust, the discontinuous function 'sgn(S)' causes chattering (high-frequency oscillations in the control signal) [7]. It excites unmodeled dynamics, increases actuator wear, and it is undesirable for mechanical systems. A common mitigation replaces sgn(S) with a saturation function sat(S/Φ) inside a thin boundary layer, trading asymptotic convergence for convergence to a small residual set.

As an alternative, adaptive control continuously estimates unknown parameters (such as payload mass, friction coefficients … etc.) online. A typical adaptation law updates the estimated parameters $\hat{\theta}$ according to:

$$\dot{\hat{\theta}} = -\Gamma Y^T(q, \dot{q}, \dot{q_r}, \ddot{q_r})S \quad (\lambda > 0) \qquad (1.20)$$

Where Γ>0 is the adaptation gain matrix, Y is the regression matrix, and S is a filtered tracking error and q_r is reference trajectory that includes error feedback.

Unlike *SMC*, adaptive control eliminates chattering and does not require knowledge of uncertainty bounds. However, it requires the uncertain parameters to appear linearly in the dynamics, a restriction that does not apply to *SMC*.

I.5.4 Comparative Analysis of Control Architectures

Each control architecture presents distinct trade-offs. Independent joint *PID* is computationally simple and easily tuned, but ignores dynamic coupling, leading to large tracking errors under high-speed, multi-DoF motions. *CTC* theoretically linearizes the full nonlinear dynamics, achieving perfect tracking with an exact model, but it becomes fragile under parameter mismatch. *SMC* offers robustness to bounded uncertainties through discontinuous switching, but introduces chattering that can excite unmodeled dynamics. Adaptive control continuously estimates unknown parameters online, eliminating chattering without requiring uncertainty bounds, at the cost of higher complexity and the linear-in parameters assumption.

These limitations of classical and strictly model-based methods (fragility, chattering, or dependence on exact models) motivate the exploration of intelligent control frameworks. Techniques such as fuzzy logic, neural networks, and learning-based control offer the ability to approximate nonlinearities, learn from data, and adapt to unstructured environments, addressing the shortcomings of analytical approaches while paving the way toward autonomous and high-performance manipulation.

I.6 Intelligent Control Strategies for Manipulators

The analytical control architectures reviewed previously (*PID*, *CTC*, *SMC* and Adaptive Control) each suffer from inherent limitations: coupling neglect, model fragility, chattering, or restrictive linear parameterization. Intelligent control strategies address these shortcomings by approximating nonlinearities, learning from data, and encoding expert knowledge without requiring exact plant models. This section examines three prominent intelligent approaches: Neural networks and learning-based control and Fuzzy Logic Control (*FLC*).

I.6.1 Neural Network and Learning-Based Control

Neural Networks (*NNs*) serve as universal function approximators, capable of representing arbitrary nonlinearities to arbitrary precision given sufficient neurons. A single-hidden-layer *NN* output y_{nn} is defined as:

$$y_{nn} = W^T \Phi(V^T x) \qquad (1.21)$$

Where $x \in \mathrm{R}^n$ is the input vector (such as e, $\dot{e}$), V is the input-to-hidden weight matrix, $\Phi(\cdot)$ is a nonlinear activation function (such as *sigmoid*, *tanh* ...), and W is the hidden-to-output weight matrix. Online weight tuning algorithms update W and V using gradient descent or *Lyapunov*-based laws to minimize tracking error.

Reinforcement Learning (*RL*) extends this approach by learning optimal control policies directly from interaction data, typically using a reward signal rather than supervised error. While powerful for complex, high-dimensional tasks, *RL* remains computationally intensive, sample-inefficient, and often lacks stability guarantees.

Generally, the '*black-box*' nature of *NNs* makes them internally difficult to understand, hard to verify and fix. They also suffer from memorization, poor optimization, and sensitivity to settings.

I.6.2 FUZZY LOGIC CONTROL

Fuzzy Logic, introduced by *Lotfi Zadeh* in 1965 [8], translates linguistic rules into mathematical mappings. Unlike *Boolean* logic where a variable is strictly true or false (0 or 1), fuzzy logic introduces the concept of partial truth, a variable can belong to multiple sets with degrees of membership between 0 and 1. This makes it highly suitable for robotic control, where dynamic transitions are continuous and nonlinear. The standard architecture of an *FLC* involves three fundamental stages: fuzzifier, inference engine and defuzzifier (see figure 1.11).

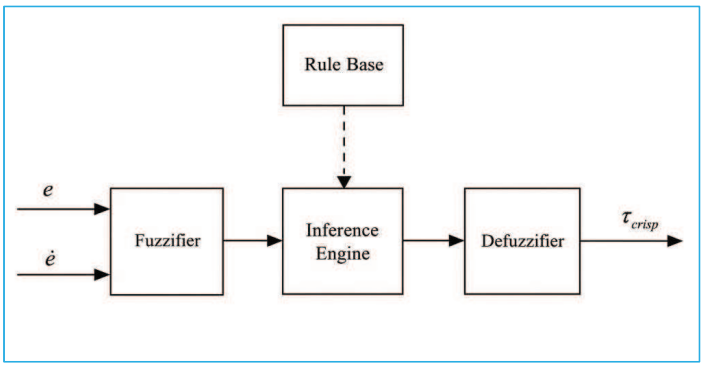


Figure 1.11 *Architecture of a Fuzzy Logic Control system* [*W.10*]

a) Fuzzification Interface

This stage maps crisp physical inputs (such as joint position error e and velocity error $\dot{e}$) into fuzzy sets. A fuzzy set A in a universe of discourse X is defined by a membership function $\mu_A(x)$ that assigns a degree of membership between 0 and 1. In robotics literature, the most commonly utilized membership functions (see figure 1.12) are:

1. *Triangular Membership Function* (*MF*)**,** favored for its computational simplicity in real-time control and expressed as follows:

$$\mu_{tri}(x, a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \qquad (1.22)$$

2. *Trapezoidal MF,* which is often chosen for its computational efficiency and piecewise linearity, provides a flat region of full membership and smooth transitions. It is critical to avoid sudden torque spikes in actuators while maintaining simple real-time computation. It is defined as:

$$\mu_{trap}(x, a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{d-x}{d-c}\right), 0\right) \qquad (1.23)$$

3. *Gaussian MF,* which is often chosen to provide smooth control surfaces and is critical to avoid sudden torque spikes in the actuators. It is

$$\mu_{Gauss}(x, c, \sigma) = e^{-\frac{(x-b)^2}{2\sigma^2}} \qquad (1.24)$$

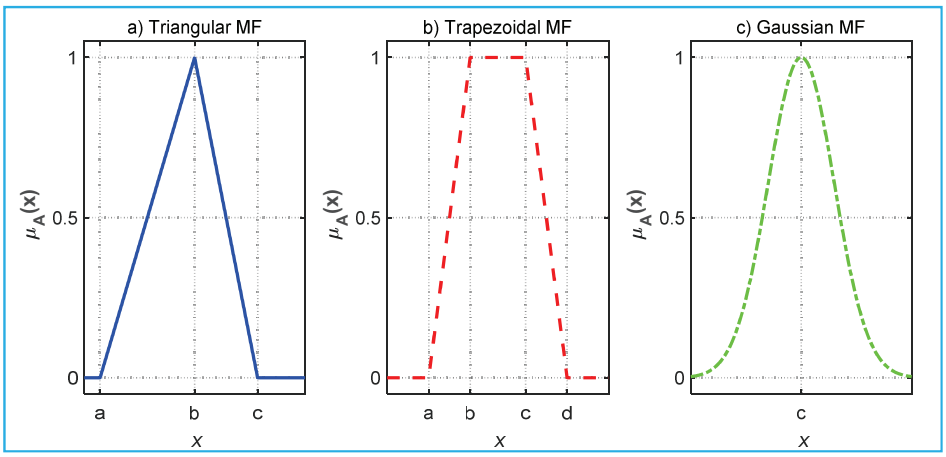


Figure 1.12 *Common shapes of MFs used in Fuzzy Logic Control* [*W.11*]

b) Rule Base and Inference Engine

The core of the *FLC* is its rule base, which encodes expert knowledge into linguistic *IF-THEN* rules, bypassing the need for an exact mathematical plant model. A typical structural rule $R^{(l)}$ for a manipulator is:

$$R^{(l)}: IF\ e\ is\ A_1^{(l)}\ \text{and}\ \dot{e}\ is\ A_2^{(l)}\ \text{THEN}\ \tau\ \text{is}\ B^{(l)} \qquad (1.25)$$

The inference engine processes these rules. The *Mamdani* inference mechanism is widely adopted in intelligent manipulation, where the firing strength (weight) $w^{(l)}$ of a rule is typically computed using the intersection (min) operator:

$$w^{(l)} = \min\left(\mu_{A_2^{(l)}}(e), \mu_{A_2^{(l)}}(\dot{e})\right) \qquad (1.26)$$

Alternatively, the product operator ($w^{(l)} = \mu_{A_2^{(l)}}(e) \times \mu_{A_2^{(l)}}(\dot{e})$) can be used, which preserves more information from both inputs at the cost of increased computational effort.

c) Defuzzification Interface

Since robotic actuators require a crisp numerical torque value τ_{crisp} rather than a fuzzy set, defuzzification is necessary. Among various techniques, the Center of Gravity (*CoG*) method is the standard for achieving precise and smooth control actions. In its practical discrete form:

$$\tau_{crisp} = \frac{\sum_{i=1}^{N} \tau_i \times \mu_{out}(\tau_i)}{\sum_{i=1}^{N} \mu_{out}(\tau_i)} \qquad (1.27)$$

This equation calculates the weighted average or the exact '*center of mass*' of all suggested control actions. For the joint's actuator (motor), this means it will not receive erratic or jumping voltage signals. Instead, it receives a single, stable and compromised torque value that smoothly resolves all overlapping expert rules.

d) Handling MIMO Complexity via Fuzzy Logic

For a multi-*DoF* manipulator, the control problem is highly coupled, making it a strict Multiple-Input Multiple-Output (*MIMO*) system. Fuzzy logic natively accommodates this complexity without requiring real-time inversion of large inertia matrices. It allows engineers to design either:

- **Decoupled rule bases** where each joint controlled independently (simpler, but ignores coupling), or
- **Interconnected rule bases** where the positional error of one joint (such as the shoulder) influences the corrective torque applied to another (such as the elbow). This mimics the coordinated neuromuscular control of the human arm, offering a high degree of robustness against unstructured payload variations that would typically destabilize classical algorithms.

I.7 RESEARCH GAP AND CONTRIBUTION POSITIONING

The previous sections have presented the principal architectures, kinematic and dynamic models, and control strategies applied to multi-DoF robotic manipulators. Classical approaches such as *PID* and *CTC* offer well-understood advantages but also exhibit clear limitations under dynamic uncertainties and model inaccuracies. More advanced methods, including neural networks and fuzzy logic, introduce promising capabilities such as learning and interpretability, however they bring their own implementation challenges. This section synthesizes these observations, provides a comparative analysis of control methodologies, identifies persistent limitations in fuzzy-based approaches, and formally states the research gap that motivates the hybrid intelligent control framework proposed in this dissertation.

I.7.1 COMPARATIVE ANALYSIS OF CONTROL METHODOLOGIES

Classical *PID* controller remains dominant in industrial applications due to its structural simplicity and ease of tuning. However, its performance degrades significantly under high-speed dynamics, varying payloads, and unmodeled disturbances. In contrast, model-based methods such as *CTC* offer high theoretical precision but are not robust, as they rely on accurate dynamic models and are sensitive to parameter uncertainties and measurement noise. Neural network-based controllers have demonstrated strong approximation capabilities, but they often lack algorithmic interpretability, making validation and fault diagnosis difficult. Within this spectrum, Fuzzy Logic (*FL*) occupies a unique position: it provides transparency and human-like reasoning, enabling heuristic knowledge to be embedded directly into the control law [8].

I.7.2 PERSISTENT CHALLENGES IN ARM CONTROL ADDRESSED BY FUZZY LOGIC

As a universal approximator, *FL* is theoretically capable of modeling any nonlinear function. Nevertheless, conventional *FLCs* face two major practical issues. First, rule extraction and membership function tuning rely heavily on human experts, rendering the design process subjective, time-consuming, and difficult to generalize. Second, when applied to multi-degree-of-freedom manipulators, the rule base of the *FLC* expands exponentially with the number of inputs, leading to increased computational load and sparse rule activation.

I.7.3 THE RESEARCH GAP AND WORK POSITIONING

CTC delivers high-precision tracking for multi-DoF robotic arms by leveraging the system's dynamic model. However, it is inherently fragile: its performance degrades sharply under modeling errors, parameter uncertainties, and external disturbances. Conversely, simple *PID* control is robust but lacks the precision required for high-speed or variable-payload trajectories.

On the other hand, *FLC* occupies an intermediate and advantageous position. Unlike neural networks or adaptive control methods, which demand extensive training data, complex architectures, and offer little interpretability, *FLC* remains conceptually simple, transparent, and easy to implement. Furthermore, prior studies have shown that a well-designed *FLC* can outperform classical *PID*, particularly under moderate nonlinearities and uncertainties.

Nevertheless, existing *FLC* implementations for robotic arms suffer from two limitations:

- Manual tuning of rules and membership functions is subjective and time-consuming;
- More critically, pure *FLC* alone does not fully exploit the high-performance potential of model-based compensation.

This dissertation investigates whether the robustness of *FLC* can compensate for the fragility of *CTC*. Four controllers are implemented on a PUMA 560 robot arm: PID (baseline), CTC (high performance but sensitive), FLC-PID (FL outperforms PID with simplicity), and the proposed Fuzzy-*CTC* hybrid. In this hybrid controller, *FL* acts as an adaptive compensator that reduces *CTC's* sensitivity to modeling errors while preserving its tracking precision. Through comparative simulation, this work try to demonstrate if the Fuzzy-*CTC* bridges the gap between simple robust control and fragile high-performance control, offering an effective and interpretable solution for multi-DoF robotic arms.

I.8 Overview of Multi-DOF Robotic Manipulators

The technological landscape of robotic manipulators has evolved considerably beyond the classical systems reviewed in earlier sections. Modern Multi-DoF industrial and collaborative robots integrate advanced sensing, embedded real-time controllers, and high-precision transmission systems that far surpass the performance envelopes of their predecessors. This section provides a focused survey of the most representative state-of-the-art of Multi-DoF robots, categorized by their mechanical architecture, control interface, and intended application domain [1], [2], [9].

I.8.1 Universal Robot UR10e

The Universal Robot UR10e (see figure 1.13) is an anthropomorphic 6-DoF collaborative robot (cobot) designed to operate safely alongside human workers without the need for protective fencing, certified under ISO/TS 15066. Its kinematic chain consists exclusively of six revolute joints, each driven by a brushless DC servo motor coupled to a high-ratio harmonic drive transmission, achieving near-zero backlash. The joint arrangement follows a modified spherical wrist configuration with a payload of 12.5 kg and a maximum reach of 1300 mm. A defining feature of the UR10e platform is its integrated force-torque sensing in each joint, enabling compliant motion and safe human-robot interaction. The robot’s embedded controller runs a real-time Linux-based operating system with a cycle time of 2 ms, allowing responsive closed-loop trajectory control [9], [10].

From a control perspective, the UR10e exposes a standardized Universal Robots Description Format (URDF) model, and its open SDK (URScript and ROS 2 driver) has made it a benchmark platform for academic research on advanced torque control, impedance control, and shared autonomy frameworks.

Its collaborative certification means that the control system must guarantee that contact forces remain bounded by safety thresholds at all times, providing a strict real-world test for the type of intelligent controllers investigated in this work. Key performance specifications include a positional repeatability of ±0.05 mm, a maximum tool speed of 1000 mm/s, and a power supply of 200–240 V AC [9], [10].



Figure 1.13 *Universal collaborative 6-DoF robot arm UR10e with integrated joint torque sensing* [*W.3*]

I.8.2 KUKA KR 6 R900 AGILUS INDUSTRIAL ROBOT

The KUKA KR 6 R900 AGILUS (see figure 1.14) belongs to KUKA's AGILUS series, engineered specifically for high-speed, high-precision small-parts assembly and handling tasks in electronics manufacturing and medical device production. This fully articulated 6-DoF robot achieves a maximum path velocity of 7750 mm/s with a positional repeatability of ±0.03 mm, among the tightest tolerances in its payload class. Its compact structure, with a total weight of only 52 kg and a 6 kg payload capacity at a 900 mm reach, makes it ideal for dense cell layouts. The wrist design (joints 4, 5, 6) employs a bevel gear arrangement that enables a continuous 360° rotation on axis 6, which is critical for complex tool-orientation tasks such as dispensing and precision screwdriving [9], [11].

The KUKA AGILUS is operated via the KUKA Sunrise OS and KRL (KUKA Robot Language) and can be integrated with ROS-Industrial for research applications. Its 'KUKA.SafeOperation' and 'KUKA.SafeRangeMonitoring' software modules allow configuring configurable safety zones, enabling semi-collaborative deployment. From a control standpoint, this robot demonstrates the challenge of managing high inertial variation at the wrist joints during rapid acceleration phases, a classical scenario where model-based methods (such as Computed Torque Control reviewed previously) struggle under parameter uncertainty, motivating the hybrid intelligent controllers addressed in this work [1], [11].



Figure 1.14 *KUKA KR 6 R900 AGILUS, a high-speed 6-DoF industrial manipulator* [*W.12*]

I.8.3 ABB IRB 6700 INDUSTRIAL MANIPULATOR

The ABB IRB 6700 (see figure 1.15) series represents the top of heavy-duty 6-DoF industrial manipulation, designed for automotive body-in-white welding, material handling, and machine tending at payloads ranging from 150 kg to 300 kg at reaches of up to 3200 mm. ABB's signature LeanID (Lean Integrated Dressing) cable routing integrates all wiring for welding equipment and process tools through the hollow upper arm and wrist, minimizing cable wear and enabling continuous rotation at axis 6. The robot is mounted on ABB's IRC5 controller platform, which features a proprietary Advanced Robot Control (ARC) algorithm implementing dynamic model feedforward compensation in real time. The 'QuickMove' and 'TrueMove' motion optimization algorithms embedded in the IRC5 controller ensure that the cartesian Tool-Center-Point (TCP) path closely tracks the programmed trajectory even at high speeds, achieving a path deviation below 0.1 mm under nominal load [2], [9], [12].



Figure 1.15 *ABB IRB 6700 heavy-duty 6-DoF manipulator for automotive welding applications* [*W.13*]

I.8.4 FANUC LR MATE 200iD MULTI-APPLICATION ROBOT

FANUC's LR Mate 200iD (see figure 1.16) is a miniature articulated 6-DoF robot widely deployed in electronics assembly, machine tending, and laboratory automation. It features a 7 kg payload, a 717 mm reach, and a repeatability of ±0.02 mm. The LR Mate series is notable for its IP67 rating (full dust and water entering protection) and an available cleanroom variant certified to ISO Class 5, making it suitable for pharmaceutical and semiconductor manufacturing environments. FANUC's proprietary R-30iB Plus controller, which governs all LR Mate models, incorporates an advanced Learning Vibration Control (*LVC*) algorithm that identifies and suppresses structural resonance modes online, resulting in significantly reduced cycle times compared to conventional bang-bang trajectory profiles [2], [9], [13].



Figure 1.16 *FANUC LR Mate 200iD compact 6-DoF robot for precision assembly and cleanroom applications* [*W.14*]

I.8.5 YASKAWA MOTOMAN HC10DT DUAL-ARM COBOT

The Yaskawa Motoman HC10DT is a 7-DoF (redundant) collaborative arm (see figure 1.17) with a 10 kg payload and a 1200 mm reach, designed for tasks requiring maximum configurational flexibility in confined spaces. Although technically 7-DoF, it is commonly analyzed and compared in the 6-DoF literature because its kinematic redundancy adds one additional degree of freedom on top of a standard 6-DoF task-space requirement. Each of its seven joints integrates a torque sensor capable of detecting contact forces as low as 0.1 N, enabling extremely sensitive human-robot interaction.

The HC10DT's Dual Check Safety (DCS) function implements configurable software-defined safety zones at a dedicated safety-rated CPU level, an approach that has become the industrial benchmark for collaborative robot safety since its adoption by IEC 62061 and ISO 13849-1 standards [9], [14].



Figure 1.17 *Yaskawa Motoman HC10DT collaborative robot with full-body torque sensing* [*W.15*]

I.8.6 FRANKA EMIKA PANDA TORQUE-CONTROLLED ROBOT

The Franka Emika Panda (now succeeded by Franka Research 3) has emerged as the standard platform for academic and industrial research in torque-controlled manipulation, see figure 1.18. With 7 revolute joints (one additional degree of redundancy beyond a standard 6-DoF manipulator), each equipped with a high-resolution joint torque sensor (1 kHz sampling rate), the Panda offers full torque-control transparency. Its Franka Control Interface (FCI) provides a real-time, 1 kHz control loop via a direct Ethernet connection, allowing researchers to implement custom control laws, including the fuzzy logic architectures central to this work, at the joint torque level, bypassing the proprietary controller.

The robot's 3 kg payload and 855 mm reach, position it as a research and light-assembly tool, while its *libfranka* C++ library and ROS 2 *franka_ros* integration have enabled a vast body of published work on impedance control, learning from demonstration, and adaptive control [9], [15].



Figure 1.18 *Franka Emika Panda research-grade 7-DoF manipulator with full joint torque sensing* [*W.16*]

I.9 COMPARATIVE ANALYSIS AND CLASSIFICATION OF MODERN 6-DOF ROBOTS

This section provides a comparative analysis and classification of modern 6-DoF robots. By categorizing these manipulators according to application domain (industrial high-speed, collaborative, and research platforms) and emerging technological approaches, we establish the operational contexts in which different control strategies are typically deployed.

It highlights precisely why a hybrid fuzzy-CTC approach, as will be seen in this dissertation, occupies a critical niche. While industrial robots prioritize model-based feedforward for high-speed precision, cobots demand compliant intelligence that pure model-based methods struggle to provide, and research platforms enable the validation of novel adaptive frameworks. Thus, the following classification serves to situate the control problem that the Fuzzy-CTC hybrid seeks to solve: delivering CTC-level accuracy where models are accurate, while leveraging FLC robustness where they are not.

I.9.1 CLASSIFICATION BY APPLICATION DOMAIN

Modern 6-DoF manipulators can be categorized along three primary axes: payload class, control architecture, and intended operational context. The classification below reflects current market and research trends [9]-[12].

***a*) *Industrial High-Speed Robots* (*Payload < 20 kg*)**

This category includes the KUKA AGILUS and FANUC LR Mate series. These robots operate within protective fencing, are optimized for cycle time minimization, and use position-controlled servo drives with proprietary path planning algorithms. Control challenges are primarily related to resonance suppression and high-velocity *Coriolis* compensation, where classical *PID* controller tuned per-joint reaches its limits and model-based feedforward compensation becomes essential [1], [3].

***b*) *Collaborative Robots* (*Cobots, Payload < 35 kg*)**

Universal Robots (*UR*) series and Yaskawa HC series belong to this category. They simultaneously achieve trajectory accuracy and contact compliance, presenting a fundamental trade-off: high stiffness for precision conflicts with the soft compliance required for human safety. This dilemma is precisely where intelligent control methods, such as the fuzzy controller investigated in this work, will show its advantage over purely model-based approaches [7], [8].

c*) *Research and Education Platforms

The Franka Emika Panda and similar platforms (such as Kinova Gen3) occupy the research category. They expose full torque-level control, provide open software interfaces, and are physically safe for laboratory deployment. These robots have become the primary testbeds for validating novel control frameworks, including reinforcement learning policies, adaptive fuzzy controllers, and model predictive control strategies. The widespread URDF (Unified Robot Description Format) model availability for these robots also facilitates realistic simulation before physical deployment [2], [15].

I.9.2 EMERGING TRENDS

The frontier of modern 6-DoF robotics is no defined solely by mechanical performance metrics. The integration of embedded AI inference engines directly within the robot controller marks a real revolution. NVIDIA's robotics platform and ABB's Omniverse Digital Twin framework illustrate this trend, enabling real-time simulation of the robot's complete dynamic model running in parallel with physical execution [9]. This 'shadow model' approach allows the controller to anticipate disturbances, pre-compute corrective torques, and validate planned trajectories before physically committing to them.

Furthermore, Deep Reinforcement Learning (*DRL*) policies trained in simulation (sim-to-real transfer) are increasingly deployed on production robots such as Boston Dynamics Spot and 1X Technologies Neo, demonstrating that the boundary between industrial and research robotics is rapidly dissolving [9], [16].

The Digital Twin concept, where a high-fidelity physics-based model of the robot runs synchronously with the physical system, provides a natural deployment environment for the *FLCs* investigated in this dissertation. Instead of tuning membership functions on a physical robot (which risks hardware damage and consumes experimental time), the *FLC* can be trained, evaluated, and optimized entirely within the digital twin environment, with sim-to-real transfer validated by the real-time synchronization data. This approach aligns directly with the research gap identified in this work: the need for autonomous adaptivity that removes the dependence on manual, expert-driven *FLC* tuning [8], [9].

I.10 CONCLUSION

This chapter reviewed the state of the art in robotic manipulators: their architectures, modeling methods, and control approaches. We classified modern 6-DoF robots into three families: industrial high-speed robots, collaborative cobots, and research platforms. We also noted emerging trends like embedded AI and digital twins that are reshaping controller development.

The key finding is clear: There is no control strategy fully solves the problem at hand and is simultaneously simple, high performing, robust, and not onerous. Classical *PID* struggles with nonlinearities. Computed Torque Control (*CTC*) offers high precision but is fragile and model-dependent. Fuzzy Logic Control (*FLC*) provides robustness but suffers from manual, expert-driven tuning. This gap between fragile performance and robust but onerous control defines the research opportunity.

Therefore, this chapter positions the objective of the dissertation: to investigate whether a hybrid Fuzzy-*CTC* controller can close the gap. Using a PUMA 560 robot arm, we will compare four controllers *PID*, *CTC*, *FLC-PID*, and the proposed Fuzzy-*CTC*, through comparative simulation, aiming to demonstrate that fuzzy logic can act as an adaptive compensator, reducing *CTC's* sensitivity to modeling errors while preserving its tracking precision.

Chapter II
Kinematic and Dynamic Modeling of Robotic Manipulators

II.1 INTRODUCTION

This chapter presents the fundamental mathematical models required to describe the motion and forces of robotic manipulators. In the study of robotic systems, two interconnected concepts are essential: kinematics and dynamics. Kinematics focuses on the geometric aspects of motion (such as position, velocity, and acceleration) without considering the forces or torques that produce that motion [17]. Dynamics, in contrast, addresses the relationship between the forces acting upon the manipulator and the resulting motion of its links and joints [18].

To design effective controllers, simulate robot behavior, or predict trajectory tracking performance, both kinematic and dynamic models must be established [19]. The kinematic model is typically divided into direct (forward) kinematics, which determines the end-effector pose given the joint coordinates, and inverse kinematics, which calculates the joint coordinates required to achieve a desired end-effector pose [1]. The dynamic model, on the other hand, is often derived using either *Newton-Euler* or *Lagrangian* formulations. The *Newton-Euler* method yields a recursive, computationally efficient model for real-time control, whereas the *Lagrangian* approach provides a closed-form representation that highlights the structure of the inertia, *Coriolis*, centripetal, and gravitational terms [5].

Thus, the objective of this chapter is to provide a step-by-step derivation of both the kinematic and dynamic equations for a serial robotic manipulator. The notation and conventions introduced here follow the standard *Denavit-Hartenberg* parametrization [2], and the resulting models will be used as the basis for the simulation and control strategies developed in subsequent chapters.

By the end of this chapter, after working through an illustrative 1-DOF modeling example to build physical intuition, the complete 6-DOF dynamic model of the PUMA 560 industrial manipulator will be obtained.

II.2 GENERAL KINEMATIC MODELING OF AN N-DOF MANIPULATOR

Kinematics is the science of motion that treats the subject without regard to the forces causing it [1]. In robotics, it focuses on the analytical mapping between the joint space and the operational space through nonlinear coordinate transformations. For an open-chain serial manipulator comprising *n* rigid links connected by *n* joints, the spatial configuration is completely defined by the generalized joint coordinate vector:

$$q = [q_1, q_2, \dots, q_n]^T \in \mathbb{R}^n \qquad (2.1)$$

II.2.1 COORDINATE FRAMES AND SPATIAL TRANSFORMATIONS

A robotic manipulator is fundamentally a sequence of rigid links connected by joints that provide either rotational (revolute) or translational (prismatic) degrees of freedom. To systematically describe the relative position and orientation between adjacent links, a dedicated Cartesian coordinate frame $\{i\}$ is rigidly attached to each link L_i, where $\{0\}$ denotes the fixed inertial base frame. The relative pose of frame $\{i\}$ with respect to frame $\{i\text{-}1\}$ is mathematically represented by a homogeneous *Transformation* matrix $T_i^{i-1} \in SE(3)$ [1]:

$$T_i^{i-1} = \begin{bmatrix} R_i^{i-1} & P_i^{i-1} \\ 0_{1\times3} & 1 \end{bmatrix} \quad (2.2)$$

Where $R_i^{i-1} \in SO(3)$ is a 3×3 orthonormal rotation matrix describing the spatial orientation, and $P_i^{i-1} \in \mathbb{R}^3$ is a 3×1 position vector defining the origin of frame $\{i\}$ relative to frame $\{i\text{-}1\}$.

By invoking the principle of chain-multiplication, the total forward kinematics mapping defining the absolute pose of the end-effector frame $\{n\}$ relative to the fixed base frame $\{0\}$ is computed as:

$$T_n^0 = T_1^0 \times T_2^1 \times \ldots \times T_n^{n-1} \quad (2.3)$$

II.2.2 The Denavit–Hartenberg (*D-H*) Convention

To standardize frame attachment, the *Denavit–Hartenberg* (*D-H*) convention restricts the relative pose of adjacent links to four geometric parameters defined relative to the joint axes [5]:

1. ***Link length*** **(a_i):** Distance from $\hat{Z}_{i-1}$ to $\hat{Z}_i$ along the $\hat{X}_i$ axis.
2. ***Link twist*** **(α_i):** Angle from $\hat{Z}_{i-1}$ to $\hat{Z}_i$ about the $\hat{X}_i$ axis.
3. ***Joint offset*** **(d_i):** Distance from $\hat{X}_{i-1}$ to $\hat{X}_i$ along $\hat{Z}_{i-1}$ axis.
4. ***Joint angle*** **(θ_i):** Angle from $\hat{X}_{i-1}$ to $\hat{X}_i$ about the $\hat{Z}_{i-1}$ axis.

The homogeneous transformation matrix T_i^{i-1} is expressed as the product of four basic transformations [1], [5]:

$$T_i^{i-1} = Rot_z(\theta_i) \times Trans_z(d_i) \times Trans_x(a_i) \times Rot_x(\alpha_i) \quad (2.4)$$

By executing this matrix multiplication, the final explicit symbolic matrix is obtained:

$$T_i^{i-1} = \begin{bmatrix} cos\theta_i & -\sin\theta_i\, cos\alpha_i & \sin\theta_i\, sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i\, cos\alpha_i & -cos\theta_i\, sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

II.2.3 Forward Kinematics Formulation

The forward kinematics problem computes the exact *Cartesian* position and spatial orientation of the end-effector based on a given set of joint variables q. This is achieved analytically by sequentially multiplying the individual *D-H* homogeneous transformation matrices via the chain multiplication equation [18]:

$$T_n^0 = T_1^0(q_1) \times T_1^2(q_2) \times \ldots \times T_n^{n-1}(q_n) \quad (2.6)$$

This resulting 4×4 matrix explicitly provides the rotation matrix R_n^0 and the position vector P_n^0 of the end-effector center point relative to the global base frame.

II.2.4 Inverse Kinematics Problem

Conversely, the Inverse Kinematics (IK) problem aims to determine the required joint variables q necessary to drive the end-effector to a desired *Cartesian* target pose. Solving this problem typically involves choosing between analytical geometric approaches (which offer precise closed-form solutions for specific architectures, such as those satisfying *Pieper's* principle) and numerical iterative methods required for more geometrically complex systems [18].

The *I K* problem presents significant mathematical challenges due to the highly non-linear trigonometric equations inherent in multi-DOF systems and the potential existence of multiple, or even infinite, geometric solutions for a single target pose [19].

II.2.5 Differential Kinematics and the Jacobian Matrix

While direct kinematics deals with static positioning, differential kinematics addresses the mapping of joint velocities $\dot{q}$ to the *Cartesian* linear and angular velocities (denoted by the twist vector *v*) of the end-effector. This fundamental mapping is formulated using the manipulator's analytical *Jacobian* matrix $J(q) \in \mathbb{R}^{6\times n}$ [19]:

$$v = \begin{bmatrix} \dot{P} \\ w \end{bmatrix} = J(q)\dot{q} \qquad (2.7)$$

To increase mathematical rigor, the geometric formulation of the *Jacobian* matrix is systematically derived column by column. For an *n*-DOF manipulator, the i^{th} column of the *Jacobian* corresponding to a revolute joint '*i*' is analytically expressed using cross-products as [1]:

$$J_i = \begin{bmatrix} z_{i-1} \times (p_n - p_{i-1}) \\ z_{i-1} \end{bmatrix} \qquad (2.8)$$

Where z_{i-1} is the axis of rotation and p_{i-1} is the position vector of the joint. This analytical formulation is strictly required not only for velocity control but also for the identification of kinematic singularities. A singularity occurs when the manipulator reaches a configuration where the *Jacobian* matrix loses full rank ($det\,(J\,(q))$ = 0 for a 6×6 matrix). In such configurations, the manipulator physically loses one or more degrees of mobility, and infinite joint velocities would theoretically be required to maintain certain *Cartesian* trajectories [19].

II.3 General Dynamic Modeling of an n-DOF Manipulator

This section presents the fundamental energy-based approach to derive the dynamic equations of motion for an *n*-DOF robotic manipulator. The *Lagrangian* formulation provides a systematic and compact framework for obtaining the joint torques or forces required to produce a desired motion. The derivation is divided into two parts: first, the formulation of kinetic and potential energies as functions of the joint variables and velocities; second, the application of the *Euler-Lagrange* equations to yield the complete dynamic model.

II.3.1 Kinetic and Potential Energy Formulation

The dynamic derivation strictly relies on energy principles. For an *n*-DOF manipulator, the total kinetic energy (*K*) is formulated as the sum of the translational and rotational kinetic energy contributions of all individual links. Mathematically, it is explicitly expressed as a quadratic function of the joint velocities $\dot{q}$, governed by the manipulator's inertia matrix *M*(*q*) [1], [18]:

$$K(q,\dot{q}) = \frac{1}{2}\ \dot{q}^T M(q)\dot{q} \qquad (2.9)$$

Conversely, the total potential energy (*P*) is strictly a function of the joint positions *q*, derived from the gravitational forces acting on the centers of mass of the respective links [19]:

$$P(q) = \sum_{i=0}^{n} m_i\, gh(q_i) \qquad (2.10)$$

The *Lagrangian* functional L is defined as the difference between total kinetic energy K and total potential energy P:

$$L(q, \dot{q}) = K(q, \dot{q}) - P(q) \qquad (2.11)$$

II.3.2 Euler-Lagrange Equations of Motion

The dynamic behavior of an n-link robotic manipulator is systematically derived using the *Lagrange-Euler* formulation. Consequently, the derivation of the required i^{th} joint torques τ_i is achieved as follows differential equations [1]:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i \qquad (2.12)$$

II.3.3 Standard Matrix Representation

Structuring the generalized dynamics yields the standard robotic formulation [18], [19], see figure (2.1):

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_{ext} = \tau \qquad (2.13)$$

In this definition:

- $M(q)$ represents the $n \times n$ Inertia Matrix;
- $C(q, \dot{q})$ represents the $n \times n$ *Coriolis* and Centripetal matrix. Its elements mathematically capture the velocity coupling effects between moving links;
- $G(q)$ represents the $n \times 1$ Gravity vector;
- $F(\dot{q})$ and τ_{ext} are the friction and external disturbances torques;
- $\tau \in \mathbb{R}^n$ is the vector of applied joint torques (actuator torques).

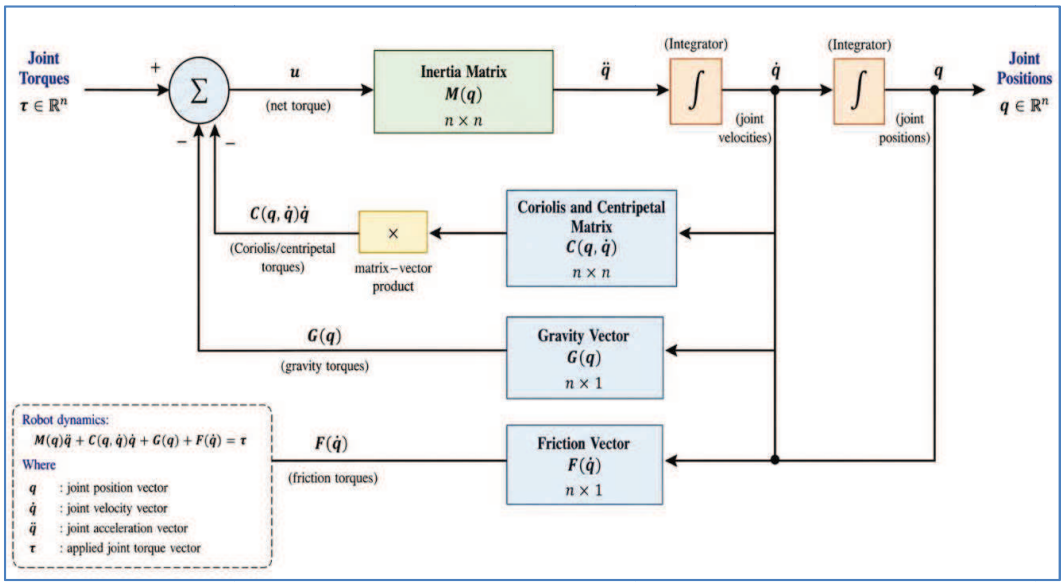


Figure 2.1 *Structural block diagram representation of the nonlinear rigid-body dynamics of an n-DoF robot manipulator*

Analytically evaluating these dynamic matrices in real-time is highly demanding. Analytical formulations may lead to high computational complexity depending on the formulation approach (explicit symbolic or recursive) [1]. For a high-DOF industrial robot, this imposes a significant computational demand.

II.4 PEDAGOGICAL CASE STUDY: 1-DOF REVOLUTE MANIPULATOR

Before confronting the dense trigonometric complexity and multi-variable dynamic coupling inherent to a 6-DOF industrial architecture, it is analytically instructive to evaluate a reduced-order mechanical system. By isolating a single-degree-of-freedom (1-DOF) revolute manipulator operating in a vertical plane, the fundamental algebraic structures of the rigid-body dynamics (specifically the inertia matrix and the gravitational vector) can be transparently observed without getting lost in complicated matrix calculations [1].

Let us consider a single rigid link of total length L_1 and uniformly distributed mass m_1,see figure (2.2). The link rotates about a fixed base joint in the *Cartesian x-y* plane, subject to a uniform gravitational field g acting along the negative y-axis. The center of mass 'C' is located at a radial distance r_1 from the axis of rotation. The generalized joint coordinate defining the system's configuration is the angular displacement $q_1 = \theta$.

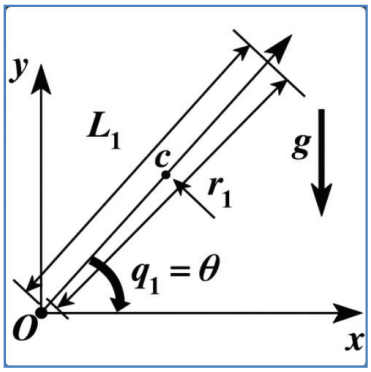


Figure 2.2 *A single rigid link rotating about a fixed base joint in xOy plane*

II.4.1 KINEMATICS AND JACOBIAN SUB-MATRIX DERIVATION

The forward kinematic mapping determines the precise spatial coordinates (x_c, y_c) of the link's center of mass as a function of the joint variable θ. Formulated geometrically, these position equations are :

$$\begin{cases} x_c = \cos(\theta) \\ y_c = \sin(\theta) \end{cases} \qquad (2.14)$$

To establish the differential kinematics, the position equations are differentiated with respect to time to extract the linear velocity mapping of the center of mass:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} = \begin{bmatrix} -r_1 \sin(\theta) \\ r_1 \cos(\theta) \end{bmatrix} \dot{\theta} \qquad (2.15)$$

This relationship is canonically represented using the analytical Jacobian matrix:

$$v_c = J_v(\theta)\dot{\theta} \qquad (2.16)$$

Where $J_v(\theta) = [-r_1 \sin(\theta) \;\; r_1 \cos(\theta)]^T \in \mathbb{R}^{2\times 1}$ represents the linear velocity *Jacobian* sub-matrix, which explicitly projects the joint-space angular velocity $\dot{\theta}$ into the *Cartesian* operational velocity vector v_c [2].

II.4.2 Lagrangian Dynamics and Energy Formulation

The explicit dynamic equation of motion is systematically derived using the energy-based *Euler-Lagrange* formulation [1], [2]. The *Lagrangian* state function $L(q, \dot{q})$ is defined as the difference between the system's total kinetic energy K and its total potential energy P:

$$L = K - P \tag{2.17}$$

The total kinetic energy K of the rigid link is the sum of the translational kinetic energy of its center of mass 'c' and the rotational kinetic energy about this center:

$$K = \frac{1}{2} m_1 \|v_c\|^2 + \frac{1}{2} I_{zz} \dot{\theta}^2 \tag{2.18}$$

Where I_{zz} is the scalar mass moment of inertia of the link about its centroidal z-axis. Substituting the squared velocity magnitude $\|v_c\|^2 = \dot{x}_c^2 + \dot{y}_c^2 = r_1^2 \dot{\theta}^2$ into the kinetic energy equation yields:

$$K = \frac{1}{2}(I_{zz} + m_1 r_1^2)\dot{\theta}^2 = \frac{1}{2} M_k \dot{\theta}^2 \tag{2.19}$$

In this formulation, the scalar constant $M_k = I_{zz} + m_1 r_1^2$ represents the strictly positive, effective rotational inertia of the link about the joint axis.

The potential energy P, resulting entirely from the gravitational field, is a function of the vertical elevation y_c of the center of the mass:

$$P = m_1 g r_1 \sin(\theta) \tag{2.20}$$

Constructing the complete *Lagrangian* function gives:

$$L(\theta, \dot{\theta}) = \frac{1}{2} M_k \dot{\theta}^2 - m_1 g r_1 \sin(\theta) \tag{2.21}$$

Now let us apply the fundamental *Euler-Lagrange* differential equation for a non-conservative system subject to an external actuator torque τ :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i} = \tau \tag{2.22}$$

And evaluate the required partial derivatives:

$$\frac{\partial L}{\partial \dot{\theta}} = M_k \dot{\theta} \Longrightarrow \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = M_k \ddot{\theta}$$

$$\frac{\partial L}{\partial \theta} = -\, m_1 g r_1 \cos(\theta)$$

Substituting these terms back into the differential equation yields the explicit, non-linear dynamic model for the 1-DOF system:

$$M_k \ddot{\theta} + m_1 g r_1 \cos(\theta) = \tau \tag{2.23}$$

By casting equation (2.23) into the canonical state-space matrix representation (2.13), the individual dynamic structural matrices are explicitly identified as:

- $\boldsymbol{M(q) = M_k}$: The inertia matrix (a scalar constant in this basic configuration);
- $\boldsymbol{C(q, \dot{q}) = 0}$: The *Coriolis* and centripetal matrix is zero, demonstrating that velocity-coupling effects strictly require a minimum of two degrees of freedom to manifest;
- $\boldsymbol{G(q) = m_1 g r_1 \cos(\theta)}$: The non-linear gravity vector.
- In this illustrative example, the friction term $F(\dot{q})$ and external disturbance τ_{ext} are not considered for simplicity reasons, as they are omitted from (2.23).

Note that this pedagogical proof of concept clearly establishes that even an isolated, single-joint mechanical system exhibits severe sinusoidal non-linearity ($G(q)$). This dynamic complexity scales exponentially as additional serial links and velocity-dependent *Coriolis* forces are integrated, providing a mathematically rigorous transition into the 6-DOF architecture.

II.5 KINEMATIC MODELING OF THE 6-DOF PUMA 560 MANIPULATOR

Having established the foundational kinematic and dynamic formulations through a simplified 1-DOF pedagogical model, this section extends the analytical framework to the primary engineering plant of this dissertation 'The 6-DOF Unimation PUMA 560 industrial manipulator' [17], [19].

II.5.1 ACADEMIC JUSTIFICATION FOR THE BENCHMARK SELECTION

While modern collaborative robots (Cobots) and next-generation multi-axis platforms represent the contemporary state of the art in industrial environments, deploying them as computational testbeds for novel intelligent control architectures introduces a severe methodological limitation. The complete structural, inertial, and frictional parameters of modern commercial manipulators are strictly proprietary and protected as closed-source intellectual property by their respective manufacturers.

To rigorously validate the trajectory-tracking precision, robustness, and adaptive capabilities of the proposed Hybrid Fuzzy-*CTC* framework under explicit parametric variations (such as unmodeled payload disturbances), a mathematically transparent and fully identified dynamic model is an absolute necessity. Utilizing a closed-source platform would introduce unquantifiable modeling ambiguities, making it analytically impossible to isolate whether tracking deviations stem from the limitations of the proposed control algorithm or from implicit structural mismatch in the proprietary physics engine.

Consequently, the Unimation PUMA 560 is formally selected as the central academic benchmark for this investigation. Its physical, inertial, and frictional parameters have been exhaustively identified, experimentally verified, and published in standard open-source robotics literature, most notably by *Armstrong et al.* [17] and *Corke* [19]. Utilizing this standardized, open-architecture platform ensures that the baseline plant dynamics are completely known, guaranteeing that the performance evaluation of the intelligent fuzzy supervisor remains objective, replicable, and mathematically rigorous.

II.5.2 Description of the 6-DOF Architecture

The PUMA 560 features a purely anthropomorphic (articulated) architecture consisting of six revolute joints (RRRRRR) [2]. It structurally mimics the human arm, comprising a shoulder, an elbow, and a wrist, see figure (2.3). The workspace is roughly spherical, making it highly adaptable for complex spatial orientations.

Crucially, the axes of the last three joints intersect at a single common point to form a 'spherical wrist', a geometric design choice that physically and mathematically decouples the end-effector positioning problem from the orientation problem [19].



Figure 2.3 *Physical layout and mechanical structure of the PUMA 560 manipulator* [2]

II.5.3 Assignment of D-H Coordinate Frame

To systematically describe the spatial geometry of the PUMA 560 manipulator, coordinate frames are assigned to each link (see figure 2.4) according to the *Denavit-Hartenberg* (*DH*) convention using the following rules.

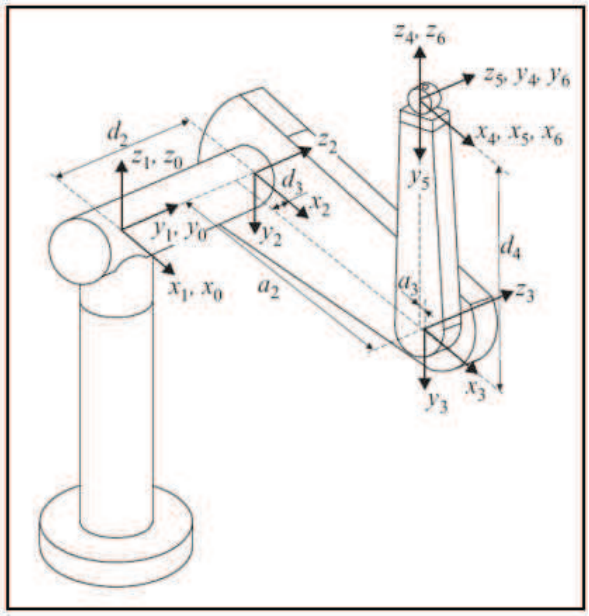


Figure 2.4 *D-H coordinate frame assignment* (*X, Y, Z axes*) *for the PUMA 560* [*W.17*]

Following the *Denavit-Hartenberg* convention [1], coordinate frames are attached sequentially from the stationary base (Frame 0) to the end-effector (Frame 6):

- ***Z-Axis assignment:*** Since all joints are revolute, each Z_{i-1} axis is strictly aligned along the physical axis of rotation for joint *i*.
- ***Frame origins:*** The origins of frames 4, 5, and 6 are forced to intersect at the wrist center to satisfy the spherical wrist condition. This kinematic property decouples position from orientation, significantly simplifying the inverse kinematics solution.

II.5.4 Derivation of the D-H Parameters Table

Based on the specific frame assignments and the standard *D-H* convention, the 6×4 geometric parameters table for the modeled robot is extracted as follows [6]:

Table 2.1 *Standard D-H Parameters for the PUMA 560*

Link *i*	α_i (°)	a_i (*m*)	d_i (*m*)	$q_i = \theta_i$ (Variables)
1	0	0	0	$q_1=\theta_1$
2	$\pi/2$	0	0	$q_2=\theta_2$
3	0	0.4318	0.14909	$q_3=\theta_3$
4	$-\pi/2$	0.0203	0.43307	$q_4=\theta_4$
5	$\pi/2$	0	0	$q_5=\theta_5$
6	$-\pi/2$	0	0	$q_6=\theta_6$

II.5.5 Derivation of the Kinematic Matrices

To accurately model the spatial displacement and velocity relationship of a robotic manipulator, it is essential to derive both the forward kinematic transformation and the associated *Jacobian* matrix.

For a structured manipulator such as the PUMA 560, the forward kinematics are established by sequentially multiplying the six specific *D-H* homogeneous transformation matrices, denoted as $T^{\,i}_{i-1}(q_i)$, resulting in the composite matrix $T_6^0(q)$ which expresses the end-effector pose as a function of the joint variables θ_i. Owing to the all-revolute architecture of the PUMA 560, the elements of $T_6^0(q)$ consist entirely of nonlinear trigonometric functions of these joint variables.

Building upon this positional model, the formulation of the 6×6 *Jacobian* matrix $J(q)$ provides the analytical mapping from joint velocities to end-effector *Cartesian* velocities. For each revolute joint *i*, the corresponding column of the *Jacobian* is computed using the standard revolute contribution formula, which incorporates the rotational axis vector Z_{i-1} and the positional vectors P_{i-1} and P_6, thereby capturing both the angular and linear velocity contributions of every joint to the tool center point motion.

b) Forward Kinematics Transformation: The specific homogeneous transformation matrix $T_6^0(q)$ representing the end-effector pose is computed by sequentially multiplying the six specific *D-H* matrices:

$$T_6^0(q) = T_1^0(q_1) \times T_2^1(q_2) \times \dots T_6^5(q_6) \qquad (2.24)$$

The position and orientation elements of $T_6^0(q)$ are entirely composed of non-linear trigonometric functions (sines and cosines) of the joint variables θ_i.

b) Jacobian Matrix Formulation: The specific 6×6 *Jacobian* matrix $J(q)$ is formulated to map joint velocities to *Cartesian* velocities. Because the PUMA 560 is an all-revolute structure, all six columns are analytically computed using the revolute joint contribution formula [19]:

$$J_i = \begin{bmatrix} Z_{i-1}(P_6 P_{i-1}) \\ Z_{i-1} \end{bmatrix} \qquad (2.25)$$

Where Z_{i-1} is the projection of the rotation axis of joint i, P_{i-1} is the position vector of the origin of frame i-1, and P_6 is the absolute position vector of the end-effector tool center point.

II.6 Formulation of the 6-DOF Dynamic Equations

The explicit dynamic equations of motion governing the PUMA 560 are systematically derived using energy-based *Euler-Lagrange* formulation, resulting in a highly nonlinear dynamic model. This model has been experimentally identified and thoroughly documented in foundational literature by *Armstrong et al.* [17]. The specific dynamic behavior of the robot is extracted directly from these *Euler-Lagrange* equations, as comprehensively presented in their work.

II.6.1 Inertia and Coriolis Dynamic Coupling

The resulting 6×6 rigid-body inertia matrix $M(q)$ is inherently coupled, symmetric, and positive-definite, with its entries exhibiting strong centripetal dependence on the instantaneous joint positions. Given that all six joints are revolute, the *Coriolis* and centripetal matrix $C(q, \dot{q})$, derived from *Christoffel* symbols of the first kind, introduces pronounced nonlinear velocity coupling terms.

Consequently, during high-speed operation, the rotation of any single joint generates substantial centripetal and gyroscopic torque disturbances on neighboring structural links. This dynamic interference poses significant challenges for classical decoupled linear joint controllers, which struggle to compensate for such cross-coupled effects.

II.6.2 Incorporating Unmodeled Dynamics and Joint Friction Models

To ensure that the mathematical model accurately replicates real-world physical behavior for the upcoming control synthesis in next chapters, unmodeled non-linear dynamics must be injected into the ideal rigid-body equations. Specifically, localized joint friction models encompassing both viscous friction and *Coulomb* friction, are explicitly integrated into the plant physics equation (2.13) by putting:

$$F(\dot{q}) = F_v \dot{q} + F_c sgn(\dot{q}) \qquad (2.26)$$

Where F_v is the viscous friction (proportional to joint velocity) coefficient, F_c is the *Coulomb* friction (a constant resistive force acting against the instantaneous direction of motion) magnitude, and $sgn(\dot{q})$ denotes the sign function of the velocity $\dot{q}$.

II.7 CONCLUSION

This chapter has systematically established the structural mathematical foundation for the kinematics and multi-body dynamics of the 6-DOF Unimation PUMA 560 industrial manipulator. Beginning with generalized geometric principles, the direct kinematic mapping $T_6^0(q)$ and the differential velocity *Jacobian* matrix $J(q)$ were formally derived under the standard *Denavit-Hartenberg* convention.

A two-stage dynamic modeling strategy was deployed: a transparent 1-DOF case study was first evaluated to isolate and build clear physical intuition regarding independent inertial and gravitational terms, which was subsequently expanded into the complete six-joint matrix formulation via the *Euler-Lagrange* energy framework. To closely mirror physical industrial behavior, the ideal rigid-body equations of motion were augmented with viscous-plus-*Coulomb* joint friction model based on validated open-source datasets.

The complete non-linear vector ordinary differential equation compiled at the conclusion of this chapter encapsulates the true physical behavior of the plant, including severe state-dependent inertia variations, non-linear velocity cross-couplings, and friction disturbances.

This high-fidelity dynamic model serves as the plant to be controlled in the following chapters, where four control strategies are implemented and compared in simulation on the PUMA 560 manipulator: classical *PID*, Computed Torque Control (*CTC*), and intelligent controllers comprising Fuzzy *PID* and Fuzzy-*CTC*, each evaluated for their ability to compensate for the modeled nonlinearities and disturbances.

Chapter III

Classical PID and Model-Based Control of the PUMA 560 Robot Arm

III.1 INTRODUCTION

The development of high-performance control algorithms for robotic manipulators relies fundamentally on translating complex physical rigid-body dynamics into highly precise algorithmic execution loops. As shown in the previous chapter, through the analytical derivations, the governing multi-body equations of motion for a six-degree-of-freedom (6-DOF) articulated manipulator, such as the Unimation PUMA 560, exhibit severe transcendental non-linearities, configuration-dependent inertia variations, and multi-variable *Coriolis*, centrifugal, and gravitational cross-couplings. When such a manipulator operates at high operational speeds or undergoes rapid spatial reversals, the dynamic coupling between adjacent links induces massive transient torque disturbances across the entire structural chain.

Consequently, establishing a robust joint control framework is an absolute necessity to guarantee trajectory tracking precision, preserve closed-loop system stability, and mitigate the detrimental effects of structural cross-coupling [3], [20]. In contemporary industrial automation environments, decentralized independent joint control serves as the universal baseline architecture. This classical strategy simplifies the highly coupled, multi-variable Multi-Input Multi-Output (*MIMO*) robotic plant by approximating it as a collection of isolated, decoupled Single-Input Single-Output (*SISO*) mechanical channels. Each localized joint is subsequently regulated by a dedicated Proportional-Integral-Derivative (*PID*) feedback loop [20]. To push the linear independent joint control framework to its absolute physical limits and minimize tracking errors under distinct operational environments, the feedback controller gains (K_p, K_i, K_d) in this work are meticulously adjusted scenario-by-scenario. Furthermore, these localized loops are augmented with model-based gravity feed-forward compensation. This latter utilizes the nominal geometric parameters derived in the kinematic model to compute and cancel out the static, configuration-dependent weight of the linkages in real-time, thereby removing workload from the highly tuned linear feedback loops [21].

To systematically evaluate the performance ceilings, disturbance rejection boundaries, and ultimate structural limits of this manually tuned classical framework, this chapter establishes a modular and rigorous evaluation pathway. The logical flow of this investigation is structured as follows:

- ***State-space reformulation:*** First, the highly coupled second-order vector ordinary differential equations (*ODEs*) representing the physical plant physics are converted into a canonical twelfth-dimensional, first-order state-space representation, establishing the definitive mathematical framework required for deterministic numerical execution [21].
- ***Controller synthesis:*** Next, the independent joint *PID* control law is formally synthesized, incorporating the scenario-specific gain tuned parameters, and coupled with the analytical model-based gravity feed-forward compensation loop [20].
- ***Simulation engine setup:*** The implementation of the numerical simulation engine is detailed, showcasing the integration of a trajectory planner with a highly stable fourth-order *Runge-Kutta* (*RK4*) integration solver to execute the closed-loop system dynamics [3].
- ***Progressive scenario evaluation:*** Finally, the controller is assessed through three progressively demanding simulation test cases.

The first evaluates trajectory tracking under nominal, ideal conditions using a 5th-order polynomial trajectory. The second introduces a continuous 3-loop 3D helical (spiral) path, constructed by superimposing high-frequency sinusoidal oscillations onto Joints (1, 2 and 3). The third adds a sudden unmodeled dynamic payload step to this complex spatial movement, testing robustness under combined aggressive path tracking and unexpected disturbance. Rather than utilizing a single rigid set of parameters, the controller gains are uniquely re-tuned for each specific scenario to evaluate its tracking performance under different operating conditions

By comprehensively documenting that even with meticulous manual gain tuning for each operating condition, a fixed-structure linear controller inevitably faces clear tracking performance limitations when confronted with unmodeled multi-body physics, this chapter establishes a rigorous baseline benchmark. This empirical evidence provides the direct scientific justification for the synthesis of the advanced, automated, and real-time adaptive hybrid Fuzzy Computed-Torque Control (*F-CTC*) architecture designed in the next chapter of this dissertation [7], [8].

III.2 State-Space Representation of the Manipulator Dynamics

To facilitate numerical simulation and algorithmic controller synthesis, the highly coupled, second-order vector ordinary differential equation (*ODE*) modeling the robot's rigid-body physics must be mathematically transformed [2]. Standard numerical integration solvers (such as the *Runge-Kutta* methods utilized in the simulation engine) are inherently designed to process systems of first-order differential equations [22]. Therefore, the second-order manipulator dynamics derived in the previous chapter must be cast into a canonical first-order state-space system [3].

The transformation begins by defining a twelve-dimensional state vector $x \in \mathbb{R}^{12}$. This vector is constructed by concatenating the generalized joint position vector $q \in \mathbb{R}^6$ and the joint velocity vector $\dot{q} \in \mathbb{R}^6$:

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = [q_1 \ \dots q_6 \ \dot{q}_1 \dots \dot{q}_6]^T \tag{3.1}$$

By defining the generalized control input vector as the vector of applied actuator torques $u = \tau \in \mathbb{R}^6$, the complete system dynamics can be explicitly isolated for the joint accelerations $\ddot{q}$. This algebraic manipulation yields an affine-in-input, highly non-linear state-space plant model governed by the following system of first-order equations [3]:

$$\dot{x}_1 = \dot{q} \tag{3.2a}$$

$$\dot{x}_2 = \ddot{q} = M^{-1}(q)(\tau - C(q,\dot{q}) - G(q) - F_v\dot{q} - F_c sgn(\dot{q})) \tag{3.2b}$$

This state-space architecture serves as the foundational physical model that is directly embedded inside the computational simulation engine.

A critical mathematical necessity for the viability of this state-space model is the guaranteed existence of the inverse inertia matrix $M^{-1}(q)$ at every discrete time step [4]. Because the physical mass properties of the PUMA 560 rigid linkages are strictly positive and geometrically bounded, the 6×6 inertia matrix $M(q)$ is fundamentally symmetric and strictly positive-definite across all physically realizable joint configurations [2].

From a computational control perspective, this positive-definite property is paramount. It mathematically guarantees that $M^{-1}(q)$ exists globally, its determinant is strictly non-zero ($det(M)=0$), and its matrix norm remains bounded throughout the entire accessible workspace [3].

Consequently, the state derivatives $\dot{x}_2$ can be continuously evaluated without risking computational singularities or infinite outputs. This structure guarantees a smooth and highly precise numerical integration of the dynamic plant [22].

III.3 Formulation of the Control Problem

This section formulates the control problem for the robotic manipulator under study. The control architecture is structured around three key components. First, the decentralized tracking error formulations are presented, defining how individual joint errors are computed and managed independently. Next, the structure of the *PID* controller is described, including the unique gain retuning applied for each test scenario. Finally, the model-based gravity feed-forward compensation is detailed, which enhances tracking performance by anticipating gravitational torques across the manipulator's configuration space.

III.3.1 Decentralized Tracking Error Formulations

The foundational objective of the closed-loop control system is to force the manipulator's actual generalized joint position vector $q(t)\epsilon\mathbb{R}^6$ to continuously and precisely track a time-varying desired reference trajectory $q_d(t)\epsilon\mathbb{R}^6$. To quantify this tracking performance, the instantaneous localized joint tracking error vector $e(t)\ \epsilon\mathbb{R}^6$ and the joint velocity error vector $\dot{e}(t)\ \epsilon\mathbb{R}^6$ are defined mathematically as [2]:

$$e(t) = q_d(t) - q(t)\ ,\ \dot{e}(t) = \dot{q}_d(t) - \dot{q}(t) \qquad (3.3)$$

The controller must generate an input torque vector τ that drives the tracking error $e(t)$ to zero asymptotically as $t\rightarrow\infty$. This control law must actively counteract the inherent dynamics of the manipulator, defined by the state-space equation derived previously:

$$M(q)\ddot{q} = \tau - C(q,\dot{q}) - G(q) - F_v\dot{q} - F_c sgn(\dot{q}) \qquad (3.4)$$

III.3.2 Structure of the PID Controller

In this approach, decentralized control is utilized; the highly coupled multi-DOF manipulator is treated as a set of independent single-input single-output linear systems. The control law is formulated as:

$$\tau = K_p e(t) + K_i \int_0^t e(t)dt + K_d\ \dot{e}(t) \qquad (3.5)$$

Where K_p, K_i, and K_d are positive-definite diagonal gain matrices representing the proportional, integral, and derivative gains, respectively.

III.3.3 Model-Based Gravity Feed-forward Compensation

While the independent joint *PID* controller is highly effective for localized error regulation, industrial manipulators are constantly subjected to severe, configuration-dependent gravitational fields [1].

Relying purely on the linear integral loop (K_i) to compensate for the massive weight of the rigid structural links requires excessively high integral gains, which fundamentally degrades the phase margin and frequently induces closed-loop instability or steady-state limit cycles.

To resolve this limitation, the classical linear architecture is augmented with a feed-forward structural loop designed to explicitly decouple the static gravity vector $\hat{G}(q)$. The final applied actuator torque vector τ is synthesized as a hybrid configuration [2]:

$$\tau_{total} = \tau_{PID} + G(q) \qquad (3.6)$$

By utilizing the known nominal kinematic and mass parameters of the manipulator to compute $\hat{G}(q)$ in real-time, this model-based feed-forward term dynamically injects the exact torque required to hold the arm against gravity at any given posture [1]. This effectively *weight-cancels* the physical plant from the perspective of the feedback controller, alleviating the linear integral loop (K_i) from compensating for static structural loads. Consequently, the *PID* loop is free to dedicate its entire control effort toward mitigating dynamic, transient positioning offsets, thereby enhancing steady-state accuracy without risking closed-loop instability.

III.4 Numerical Simulation and Algorithmic Flow

The practical simulation of the control system follows a rigorous algorithmic flow to solve the robot's physics. The script executes the following modular steps during each time step '*dt*':

III.4.1 Trajectory Generation: Quintic Polynomial / 3-Loop Helical Path

To ensure the manipulator executes fluid, industrially viable motions without inducing mechanical shock to the joints, the control system is driven by two distinct reference trajectories, each serving a specific evaluation purpose [1].

The simulation engine employs a quintic (fifth-order) polynomial reference generation algorithm for smooth, nominal joint-space path planning [3]. The positional profile for each joint *i* as a function of time *t* is defined as:

$$q_{di}(t) = \beta_{0i} + \beta_{1i}t + \beta_{2i}t^2 + \beta_{3i}t^3 + \beta_{4i}t^4 + \beta_{5i}t^5 \qquad (3.7)$$

The primary engineering advantage of a fifth-order polynomial lies in its capacity to satisfy six explicit boundary conditions: initial and final positions, velocities, and accelerations [4]. By enforcing zero velocity and zero acceleration at departure and arrival points, the resulting profile is strictly twice-differentiable [3], preventing step changes in acceleration and eliminating high-frequency torque chattering.

For more aggressive performance assessment, a second trajectory is introduced: a continuous 3-loop *3D* helical (spiral) path. This is constructed by superimposing high-frequency sinusoidal and cosinusoidal oscillations onto joints (1, 2 and 3), modulated by a *Hann* window envelope $w(t) = \frac{1}{2}(1 - \cos(2\pi t/t_f))$ to ensure zero velocity and acceleration at start and end points. The resulting joint trajectories are:

$$q_{d1}(t) = q_{d1}^{base}(t) + A_1 w(t)\sin(\omega t) \quad (3.7.a)$$
$$q_{d2}(t) = q_{d2}^{base}(t) + A_2 w(t)\cos(\omega t) \quad (3.7.b)$$
$$q_{d3}(t) = q_{d3}^{base}(t) + A_3 w(t)\sin(\omega t) \quad (3.7.c)$$
$$q_{dj}(t) = q_{dj}^{base}(t)\ (j = 4,5,6) \quad (3.7.d)$$

With $\omega=6\pi/t_f$ for exactly three loops, and base trajectory $q_{di}^{base}(t)$ as the quintic polynomial baseline. The resulting motion rigorously tests tracking fidelity under complex, continuous spatial demands.

Thus, the quintic polynomial evaluates baseline tracking under smooth, ideal conditions, while the 3-loop helical trajectory challenges the controller with highly aggressive, continuous *3D* path following.

III.4.2 Fourth-Order Runge-Kutta Integration Solver

The multi-body non-linear state-space equations derived in Section III.2 must be numerically integrated to simulate the continuous-time physics of the robot within a discrete computational environment like *MATLAB* [22]. Because the dynamics of the PUMA 560 are highly coupled and exhibit stiff mathematical characteristics, standard single-step solvers such as *Euler* integration suffer from severe cumulative truncation drift, leading to rapidly diverging and unstable simulations.

To ensure high numerical precision and deterministic stability, the simulation engine propagates the robot's state vector $x(t)$ over a discrete step time dt utilizing the classical fourth-order *Runge-Kutta* (*RK4*) numerical integration algorithm [22]. The *RK4* sequence calculates four intermediate derivative slopes (k_1, k_2, k_3, k_4) to accurately approximate the non-linear curvature of the state trajectory:

$$k_1 = f(x_t, u_t, t) \quad (3.8.a)$$
$$k_2 = f(x_t + \frac{dt}{2}k_1, u_t, t + \frac{dt}{2}) \quad (3.8.b)$$
$$k_3 = f(x_t + \frac{dt}{2}k_2, u_t, t + \frac{dt}{2}) \quad (3.8.c)$$
$$k_4 = f(x_t + dt\times k_3, u_t, t + dt) \quad (3.8.d)$$

The discrete future state of the manipulator is then computed as the weighted average of these four slopes:

$$x(t + dt) = x(t) + \frac{dt}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (3.9)$$

By evaluating the system dynamics at the midpoint and endpoint of each integration interval, the *RK4* algorithm effectively eliminates the lower-order truncation errors inherent in standard integration methods [22]. This robust solver loop ensures that the complex *Coriolis*, centrifugal, and inertial cross-couplings are propagated with extreme fidelity throughout the entire duration of the tracking simulation.

III.5 THE THREE SIMULATION SCENARIOS AND RESULTS ANALYSIS

To systematically evaluate the absolute tracking capabilities, disturbance rejection limits, and structural fragility of the classical independent joint architecture, the closed-loop system was subjected to three distinct and progressively demanding simulation test scenarios. These scenarios are:

- Nominal trajectory tracking under ideal conditions using a smooth quintic polynomial path;
- Aggressive 3-loop helical trajectory tracking to assess performance under complex, continuous spatial demands;
- And robustness evaluation under a sudden unmodeled dynamic payload step added to the helical path.

In each scenario, the *PID* controller gains (K_p, K_i, and K_d) were meticulously tuned and optimized to extract the maximum possible performance from the linear control law, ensuring a rigorously fair assessment of its fundamental limits.

III.5.1 FIRST TEST 'TRACKING PERFORMANCE UNDER BASELINE NOMINAL CONDITIONS'

In the initial scenario, the manipulator was simulated under its standard, baseline physical conditions. Unlike simplified and idealized models, this baseline simulation explicitly incorporates the inherent mechanical joint friction of the PUMA 560 arm to reflect realistic industrial operation. Both the velocity-proportional viscous damping ($F_v\,\dot{q}$) and the dry *Coulomb* friction ($F_c\,sgn(\dot{q})$) were fully active within the integration engine. However, the system operated without any external operational disturbances: the dynamic payload mass variation was set to zero ($\Delta m = 0$). The control objective was to track the smooth *quintic* polynomial reference trajectory while compensating for the arm's structural mass and internal frictional resistance.

Upon executing the simulation over the defined tracking window, the tuned *PID* controllers coupled with the gravity feed-forward loop successfully maintained strict adherence to the reference path. To thoroughly document this baseline performance, five standard diagnostic plots were generated and explained hereafter.

A direct visual analysis of figure 3.1 reveals the kinematic tracking profiles for all six joints of the PUMA 560 over the 6-second execution window. The results demonstrate that the actual joint positions (red dashed lines) successfully track the 5th-order reference trajectories (blue lines). As observed, the actual joint responses overlay practically the desired paths across all regional (Joints 1-3) and wrist axes (Joints 4-6). The joints transition smoothly from their initial starting configurations to their final target angles without any visible transient lag or oscillatory overshoot. This good convergence demonstrates the high effectiveness of the manually tuned proportional (K_p) and derivative (K_d) control gains, which successfully force the mechanical links to adhere to the dynamic reference while overcoming the baseline internal viscous and *Coulomb* friction.

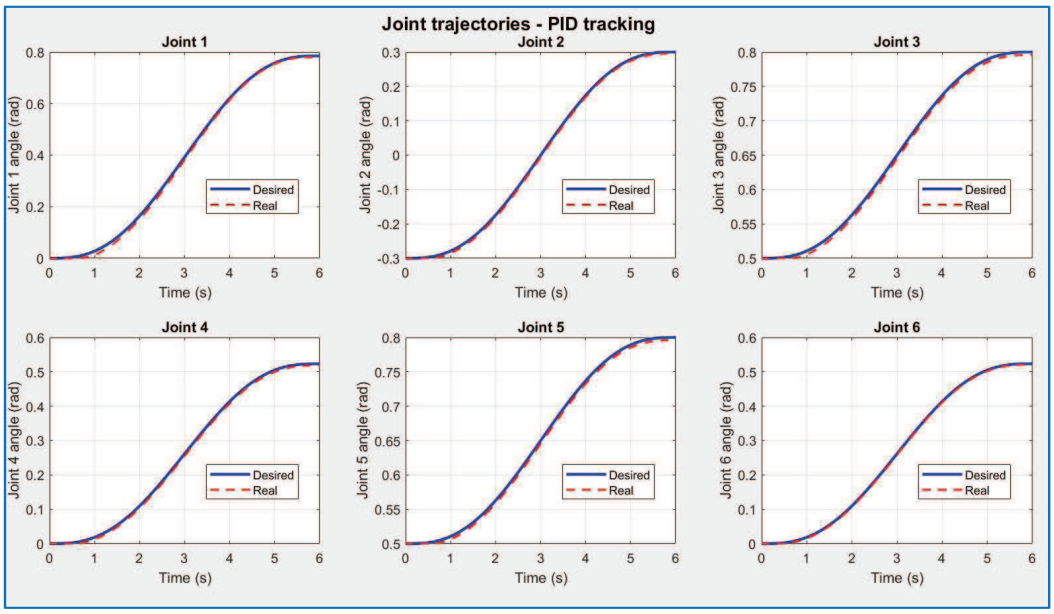


Figure 3.1 *Joint space trajectories – PID / Test I*

Moreover, a detailed examination of figure 3.2 illustrates the transient and steady-state tracking error profiles, $e_i(t)$, for all six independent joints. The black dashed lines indicate the predefined operational tolerance limit of (**±0.02 rad**). As shown by the solid green curves, the tracking error remains within a very tight tolerance (typically < **0.02 rad**), indicating that the *RK4* solver maintains high accuracy throughout the 6-second simulation. A minor transient peak is observable in Joint 1, reaching approximately (**0.015 rad**) during the initial acceleration phase (between t = 1 and t = 2 seconds). This localized deviation coincides with the period of maximum dynamic resistance caused by structural inertia and internal baseline friction. Nevertheless, the accumulated integral action (K_i) and the virtual damping provided by the derivative term (K_d) successfully suppress these dynamic ripples. The errors are efficiently driven toward a stable, near-zero steady state across all axes, proving the baseline robustness and adequate tuning of the decentralized *PID* control loops.

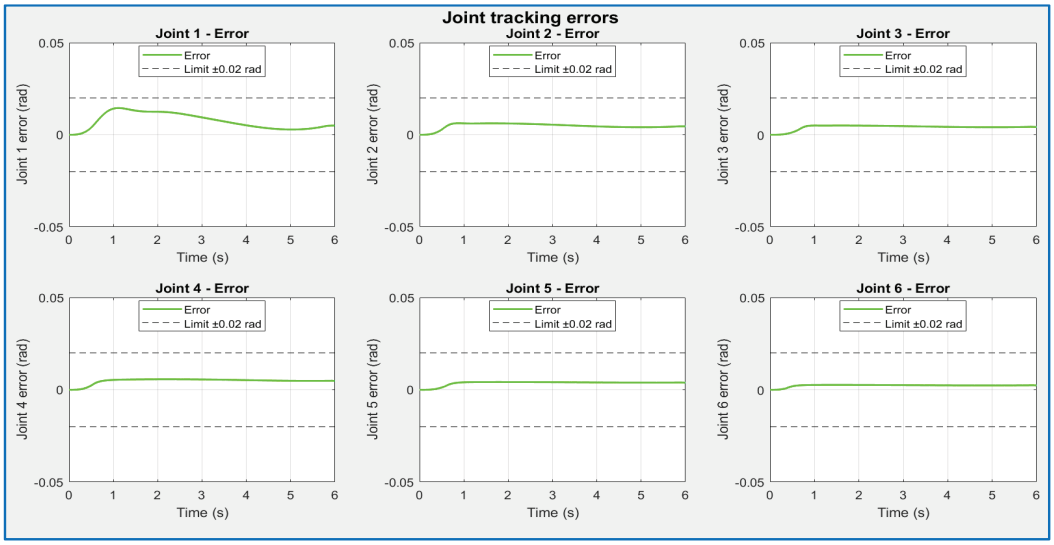


Figure 3.2 *Joint position tracking errors – PID / Test I*

Elsewhere, figure 3.3 presents the chronological evolution of the control effort, $\tau_i(t)$, demanded by the independent *PID* controllers for each of the six joint actuators. A comparative analysis of the subplots reveals a significant disparity in torque magnitudes across the kinematic chain, directly reflecting the manipulator's physical mass distribution. Joint 2 requires the maximum control effort, peaking at approximately (**74 N.m**), as it carries the dominant gravitational weight of the succeeding links. Conversely, the distant wrist joints (Joints 4, 5, and 6) demand minimal driving torque (well below **0.5 N.m**) due to their negligible regional inertia. Importantly, all commanded signals remain comfortably bounded within the predefined physical actuator saturation limits of ($\pm$**150 N.m**), confirming that the control system operates safely within its stable linear bandwidth. Furthermore, localized high-frequency torque chattering is distinctly observable at initial (t=0s) and terminal (t=6s) instants of the motion, particularly within the lighter wrist joints. This phenomenon is a direct physical manifestation of the non-linear *Coulomb* friction term ($F_c\ sgn(\dot{q})$ creating a discontinuous resistive barrier during zero-velocity crossings, which the linear *PID* controller aggressively, but imperfectly, attempts to compensate.

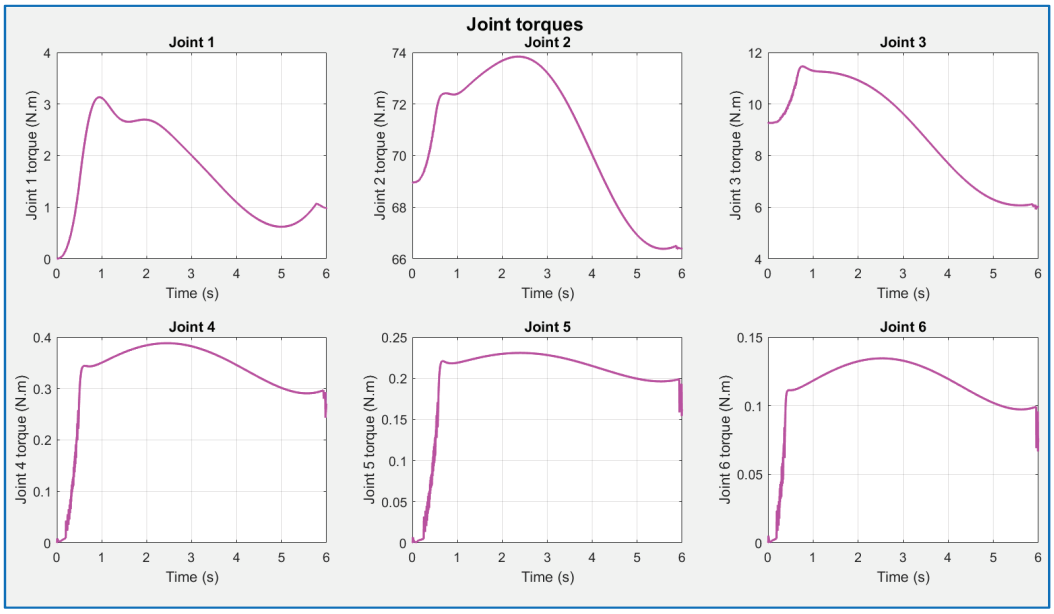


Figure 3.3 *Joint torques – PID / Test I*

Besides, figure 3.4 provides a comprehensive spatial visualization of the robot's kinematic execution within its three-dimensional workspace. The plot displays the tool center point navigating from the starting coordinates (marked by the green sphere) to the final target position (marked by the red sphere). A critical observation is the perfect visual overlay of the actual tracked path (dashed red line) onto the desired quintic polynomial path (solid blue line).

This good *Cartesian* matching confirms the highly synchronized multi-axis coordination achieved by the decentralized *PID* controllers. Furthermore, it directly validates the structural accuracy of the Direct Geometric Model (*DGM*) matrix transformations implemented in the simulation engine, ensuring that the isolated joint-space regulations successfully translate into precise macroscopic task-space execution without spatial deviations.

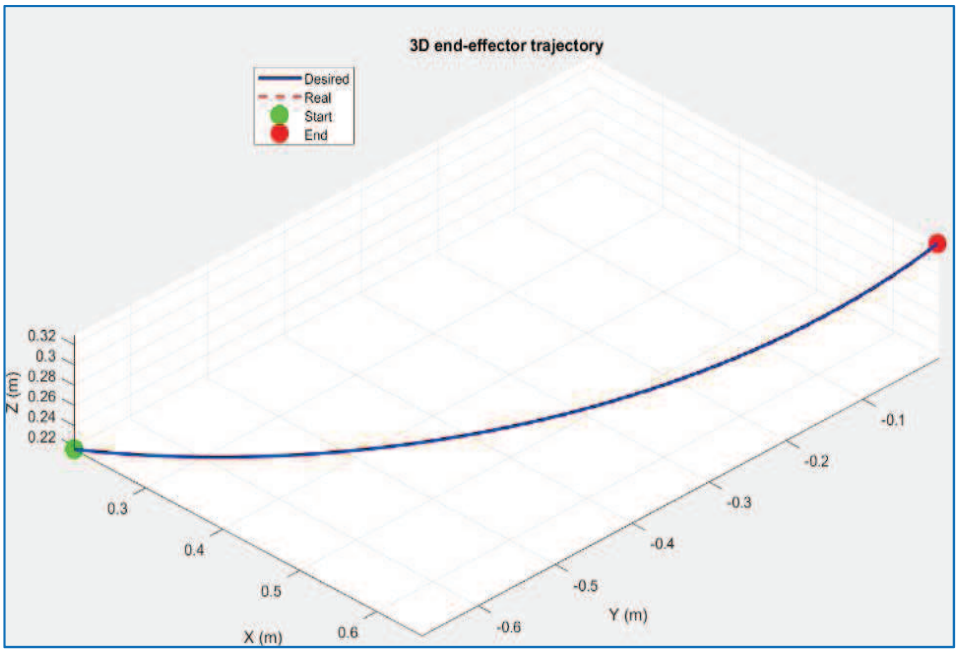


Figure 3.4 *3D end-effector trajectory – PID / Test I*

The final diagnostic plot (see figure 3.5) summarizes the overall macroscopic tracking quality by mapping the absolute *Euclidean Cartesian* position error of the end-effector. The error curve exhibits an initial transient rise, peaking at approximately **(14.2 mm)** around the 2-second mark. This peak strictly coincides with the maximum acceleration phase of the manipulator, where the controller must exert maximum effort to overcome static structural inertia and the breakaway *Coulomb* friction present in the baseline plant. As the manipulator reaches its peak velocity and subsequently decelerates, the integral action (K_i) aggressively compensates for these non-linearities, driving the deviation down. Across the entire trajectory, the system achieves a good mean *Cartesian* error of just **(8.5 mm)**. These quantitative results confirm that, under baseline conditions, the well-tuned classical *PID* architecture provides highly accurate spatial positioning suitable for standard industrial applications.

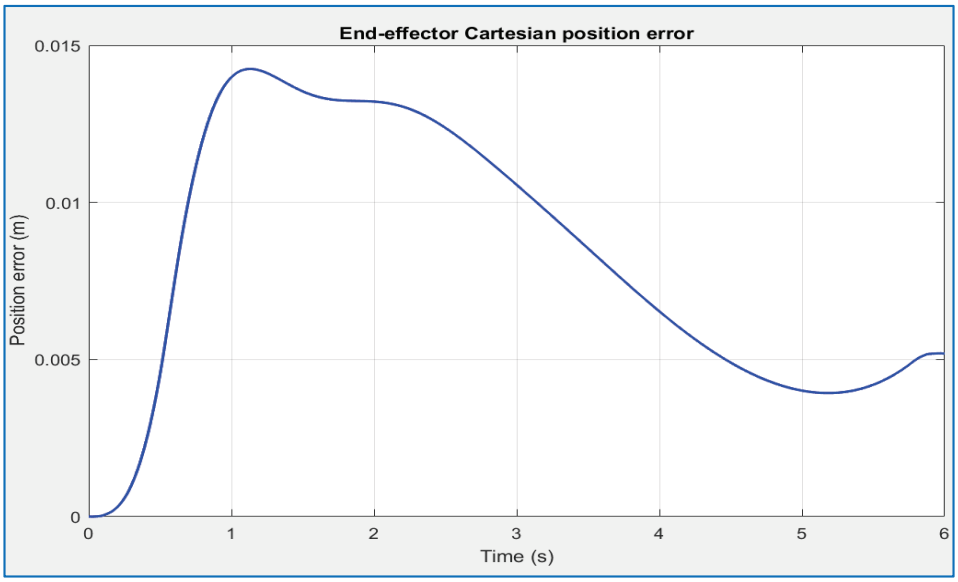


Figure 3.5 *Euclidean position error – PID / Test I*

III.5.2 Second Test 'High-Dynamic Tracking of a 3D Helical Trajectory'

Rather than testing simple mass variations, the second simulation scenario was designed to push the linear *PID* architecture to its absolute dynamic and bandwidth limits. In this test, the smooth point-to-point polynomial trajectory was replaced by a highly aggressive, continuous *3-loop 3D helical* (*spiral*) *path*. This complex spatial movement was mathematically constructed by superimposing high-frequency sinusoidal and cosinusoidal oscillations onto the base regional joints (Joints 1, 2, and 3), modulated by a *Hann* window envelope to ensure boundary continuity (see equations 3.7).

Executing this continuous oscillating motion induces massive, rapidly fluctuating *Coriolis* and centrifugal cross-coupling forces across the structural links. To combat these severe multi-body non-linearities and force the manipulator to track the spiral, the independent *PID* gains were aggressively amplified (for example $K_p = 1800$ and $K_d = 250$ for the base joint, see the appendix for more details).

A visual analysis of figure 3.6 immediately highlights the aggressive nature of the second test scenario. The kinematic reference profiles (solid blue lines) for the regional positioning links (Joints 1, 2, and 3) have been mathematically transformed into high-frequency, continuous oscillatory waves to generate the *3D* spatial loops. In contrast, the terminal wrist joints (Joints 4, 5, and 6) maintain standard quintic polynomial paths to preserve end-effector orientation.

Despite the severe dynamic stress and continuous velocity reversals demanded by the base joints, the actual measured joint positions (dashed red lines) successfully overlay the desired trajectories without diverging. This visual tracking success is the direct mathematical consequence of pushing the independent *PID* gains to their extreme maximum stable limits (see the appendix).

By operating at the absolute boundary of the linear controller's bandwidth, the virtual mechanical stiffness is momentarily sufficient to force the links along the complex path, though at the cost of immense, hidden dynamic strain on the actuators.

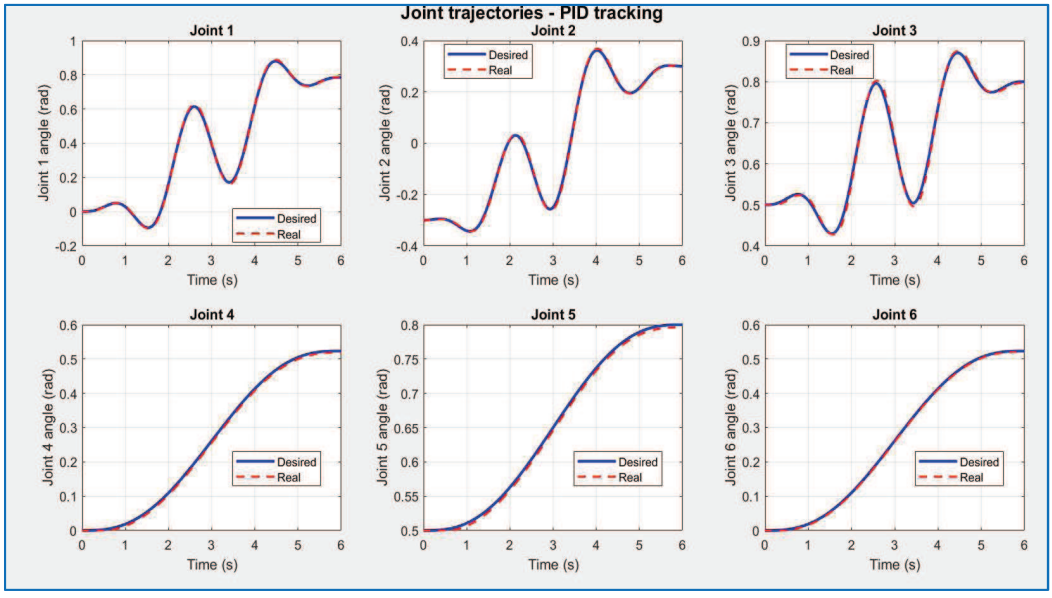


Figure 3.6 *Joint space trajectories – PID / Test II*

A detailed examination of figure 3.7 reveals the fundamental struggle of the linear control architecture when subjected to highly dynamic non-linearities. The tracking error profiles exhibit a marked distinction between the regional positioning links (Joints 1, 2, and 3) and the distal wrist links (Joints 4, 5, and 6). Because the wrist joints execute standard, smooth polynomial motions, their errors quickly settle into a stable steady-state.

Conversely, the error curves for the base and shoulder joints exhibit pronounced sinusoidal ripples that perfectly mirror the frequency of the *3D* spatial loops. This cyclic oscillation occurs because the integral term (K_i) is structurally incapable of achieving a zero steady-state error when the reference velocity is continuously reversing. The linear *PID* controller is forced to continuously '*hunt*' the dynamic reference, reacting to the rapidly fluctuating *Coriolis* and centrifugal cross-coupling forces rather than anticipating them.

However, it is crucial to note that despite this severe dynamic stress, the extreme virtual stiffness provided by the maximized proportional gains successfully constrains the peak transient deviations strictly within the (**±0.02 rad**) tolerance boundaries, preventing total system divergence.

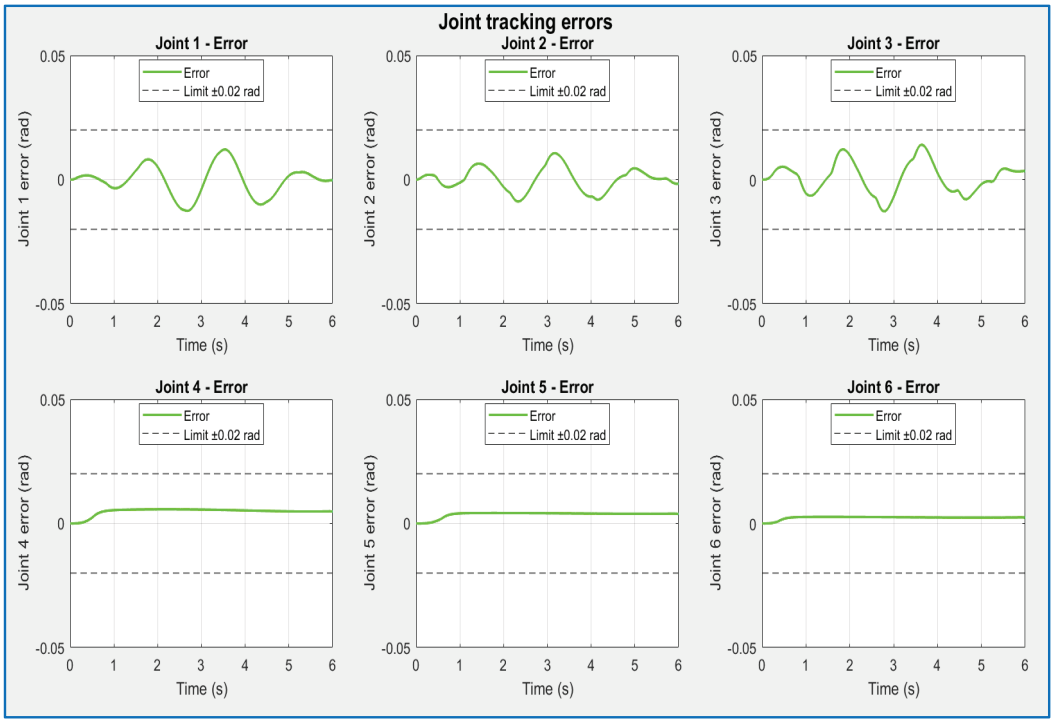


Figure 3.7 *Joint position tracking errors – PID / Test II*

Figure 3.8 provides a physical evidence of the mechanical strain imposed on the manipulator during the high-dynamic helical test. The torque command curves, $\tau_i(t)$, reveal a severe difference between the operation of the base and the wrist. Because joints 4, 5, and 6 are executing standard polynomial paths, their torque profiles remain smooth and require minimal effort. In contrast, the regional actuators (Joints 1, 2, and 3) exhibit massive and high-frequency torque oscillations. Driven by the extreme controller gains (isuch as K_p = 1800 of joint 1 and K_p = 1200 of joint 2, see the appendix), the control signals aggressively and constantly reverse direction to fight the induced centrifugal momentum and *Coriolis* forces of the swinging robotic arm.

Joint 2 experiences the most severe dynamic loading, fluctuating violently and peaking near (**90 N·m**), while joint 1 exhibits continuous torque reversals between (**-30 N·m** and **+30 N·m**). While these extreme torque commands successfully force the arm to track the path, this operational state risks rapid mechanical wear, overheating, and potential damage to physical servo drives and gearboxes in a real-world scenario.

This clearly demonstrates the energetic inefficiency and physical limitations of utilizing over-tuned, fixed-gain linear controllers for complex spatial tasks.

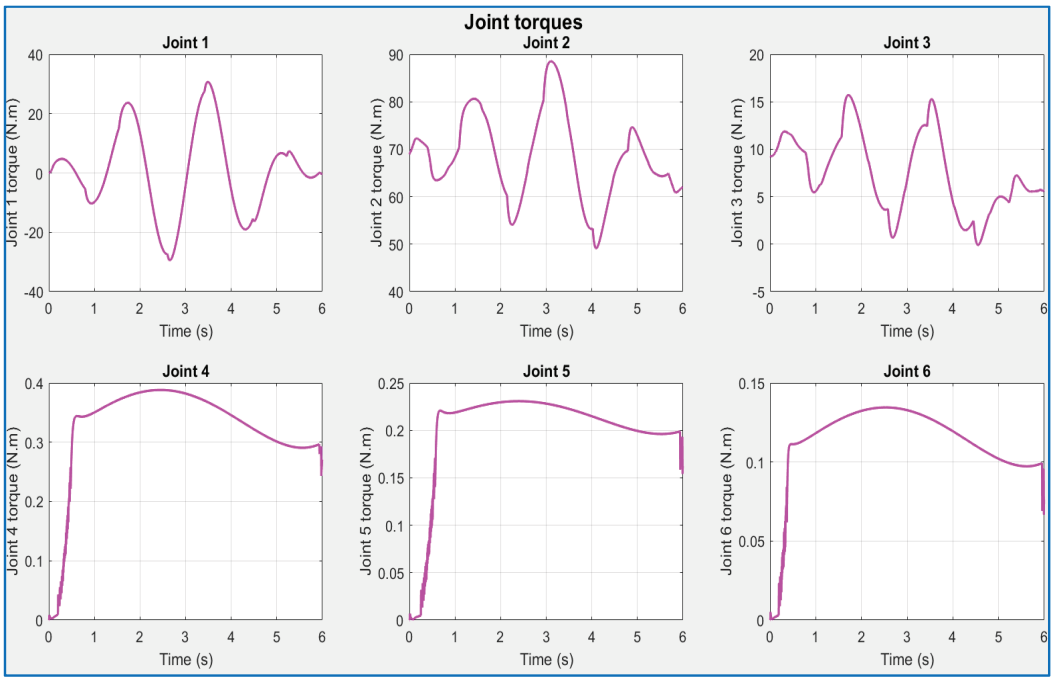


Figure 3.8 *Actuator control torques – PID / Test II*

Furthermore, figure 3.9 provides a satisfying spatial visualization of the extreme dynamic task. The tool center point is commanded to navigate through a continuous, three-dimensional 3-loop spiral from the green starting coordinate to the red destination marker.

While the overall geometric shape is successfully executed, proving that the extreme proportional gains prevent system instability, a close visual inspection reveals distinct *Cartesian* deviations.

At the apogees (sharpest curves) of each helical loop, the actual trajectory (dashed red line) noticeably separates from the desired path (solid blue line).

This macroscopic spatial separation is the direct physical manifestation of the localized joint errors observed in figure 3.7. It visually confirms that while tuning a decentralized linear controller to its extreme limits can force a manipulator through a complex path, it fundamentally fails to achieve perfect spatial synchronization due to the uncompensated, highly coupled *Coriolis* and centrifugal forces.

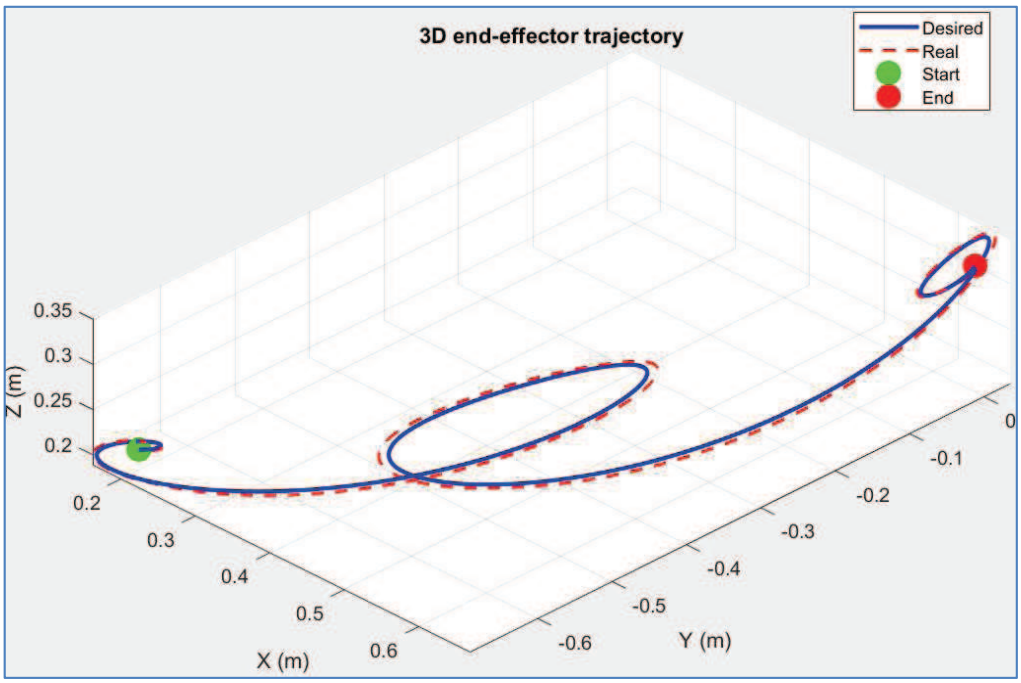


Figure 3.9 *3D end-effector trajectory – PID / Test II*

The final diagnostic plot (see figure 3.10) quantitatively summarizes the macroscopic impact of the helical trajectory by measuring the absolute *Euclidean Cartesian* error over time. Unlike the baseline nominal test where the error smoothly converges, this error profile exhibits continuous, severe cyclic peaks. These recurring error spikes directly correspond to the moments when the manipulator is executing the sharpest curves of the three spatial loops, where centrifugal acceleration and dynamic cross-coupling are at their absolute maximum. The highest positional deviation occurs near t=3.4 s, reaching a peak *Cartesian* error of approximately **(14.5 mm)**. The inability of this curve to settle asymptotically, despite the aggressive K_p and K_d gain amplification, mathematically quantifies the fundamental limits of classical model-free feedback. It definitively proves that extreme gain tuning alone cannot mathematically decouple or completely neutralize continuous, high-speed multi-body non-linearities, resulting in persistent spatial inaccuracies during complex industrial tasks.

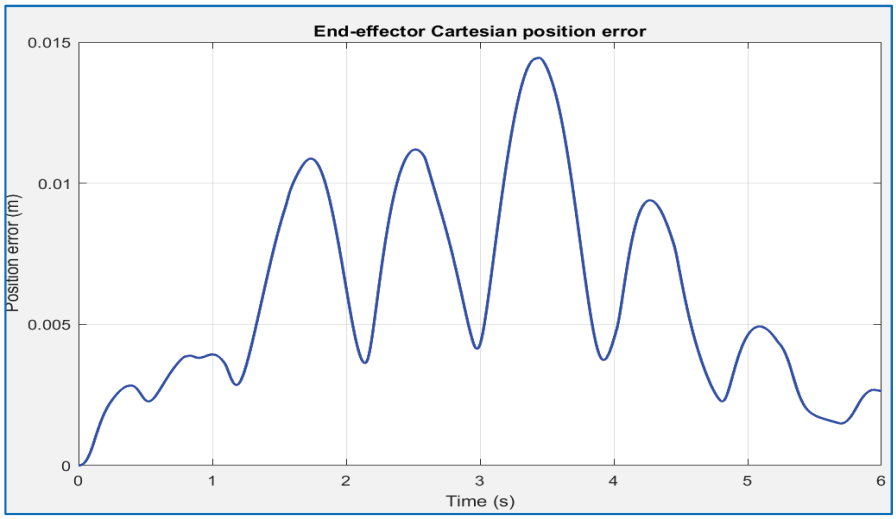


Figure 3.10 *End-effector Cartesian position error – PID / Test II*

III.5.3 Third Test 'Robustness Under Unmodeled Payload During High-Dynamic Spatial Tracking'

The third and final scenario represents the most severe operational stress test for the classical control architecture. It combines the highly dynamic, continuous 3-loop helical trajectory from of the second test case with a severe parametric uncertainty: an unmodeled payload. In this simulation, the physical plant (integration engine) is mathematically loaded with an additional 5kg mass at the end-effector (Δm = 5kg). However, the controller's internal gravity feed-forward loop, $\hat{G}(q)$, is deliberately strictly constrained to use the nominal, empty-load parameters.

This intentional model mismatch creates a critical vulnerability. The pre-calculated gravity compensation fails to cancel the new 5kg weight, instantly shifting the entire structural compensation load onto the linear *PID* feedback loops, specifically the integral action (K_i). To prevent complete system collapse under this combined inertial and gravitational stress, the controller gains were again tuned and pushed to large values (For example K_p = 3000, K_i = 200, K_d = 400 for joint 2, see appendix for details).

A visual analysis of figure 3.11 illustrates the joint kinematics operating under extreme combined dynamic stress. The control system is simultaneously fighting the high-frequency inertial forces of the 3-loop helical path and the uncompensated gravitational moment of the 5kg payload. While the massive proportional and derivative gains (For example K_p = 3000 and K_d = 400 for joint 2) forced the actual joint angles (dashed red lines) to follow the reference waves (blue lines), a distinct performance degradation is now clearly visible. Unlike the closer tracking observed in the second test, there is a perceptible macroscopic separation between the desired and actual trajectories, most notably within joint 2 and joint 3.

Because the internal gravity feed-forward loop is '*blind*' to the added mass, the red tracking lines visibly lag and shift below the blue reference curves, particularly during the upward anti-gravity acceleration phases of the sinusoidal loops. This visual deviation confirms that the fixed-gain linear controller is being physically overpowered by the parametric mismatch.

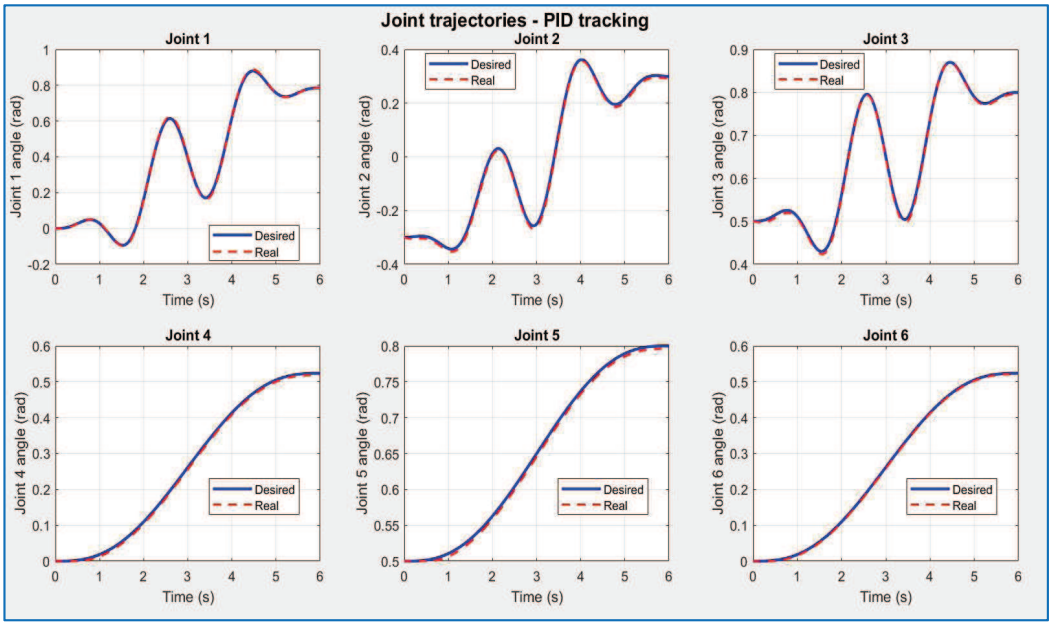


Figure 3.11 *Joint space trajectories – PID / Test III*

Moreover, a detailed examination of the tracking error profiles $e_i(t)$, see figure 3.12, exposes the severe limitations of the linear architecture when subjected to parametric mismatch. While the terminal wrist articulations (Joints 4, 5, and 6) maintain near-zero error, the primary load-bearing links (Joints 2 and 3) exhibit a critical dynamic phenomenon. Unlike the purely sinusoidal, symmetric errors observed in the second test case, the error curve for joint 2 is now distinctly asymmetric, shifted entirely into the positive region (oscillating between approximately **+0.005 rad** and **+0.015 rad**). Because the mathematical tracking error is defined as $e(t) = q_d(t) - q(t)$, a persistent positive error implies that the actual joint angle constantly lags below the desired reference.

Since the controller's internal gravity feed-forward loop is '*blind*' to the extra 5kg, it underestimates the necessary holding torque. Consequently, the linear integral term (K_i) is forced to continuously accumulate a permanent, non-zero steady-state error just to support the unmodeled weight. Although the extreme controller gains successfully prevent the error from violating the absolute (**±0.02 rad**) safety threshold, this persistent structural offset fundamentally degrades the absolute spatial precision of the manipulator.

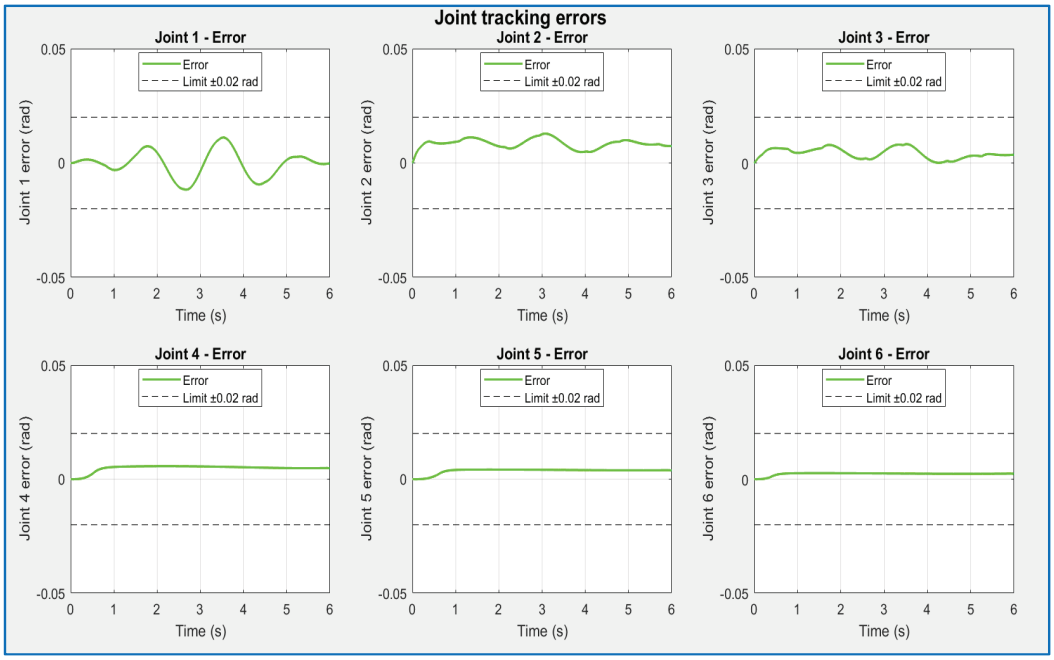


Figure 3.12 *Joint position tracking errors – PID / Test III*

Elsewhere, figure 3.13 provides critical, quantitative insight into the massive energetic cost of utilizing a linear controller under severe parametric mismatch. The torque command curves $\tau_i(t)$, expose the extreme mechanical constraint imposed primarily on the regional load-bearing joints. A direct comparison with the second test case reveals a radical upward shift in the continuous control effort for joint 2. Because the internal gravity feed-forward loop is unaware of the added 5 kg payload, it completely fails to provide the necessary baseline lifting torque.

Consequently, the integral action (K_i) of the *PID* controller must bear the entire structural load, forcing joint 2 to operate incessantly in a highly elevated torque band. The torque continuously oscillates between (**+70 N.m** and **+120 N.m**), peaking dangerously near (**120 N.m**). Joint 3 similarly experiences an amplified peak command approaching (**25 N.m**).

While these extreme efforts mathematically manage to drag the links along the helical path, this state of continuous, high-load oscillation is energetically impracticable. In a physical industrial setting, sustaining such extreme torque commands would inevitably induce rapid thermal overload in the servo motors, severe mechanical fatigue in the transmission gearboxes, and drastically reduce the operational lifespan of the manipulator.

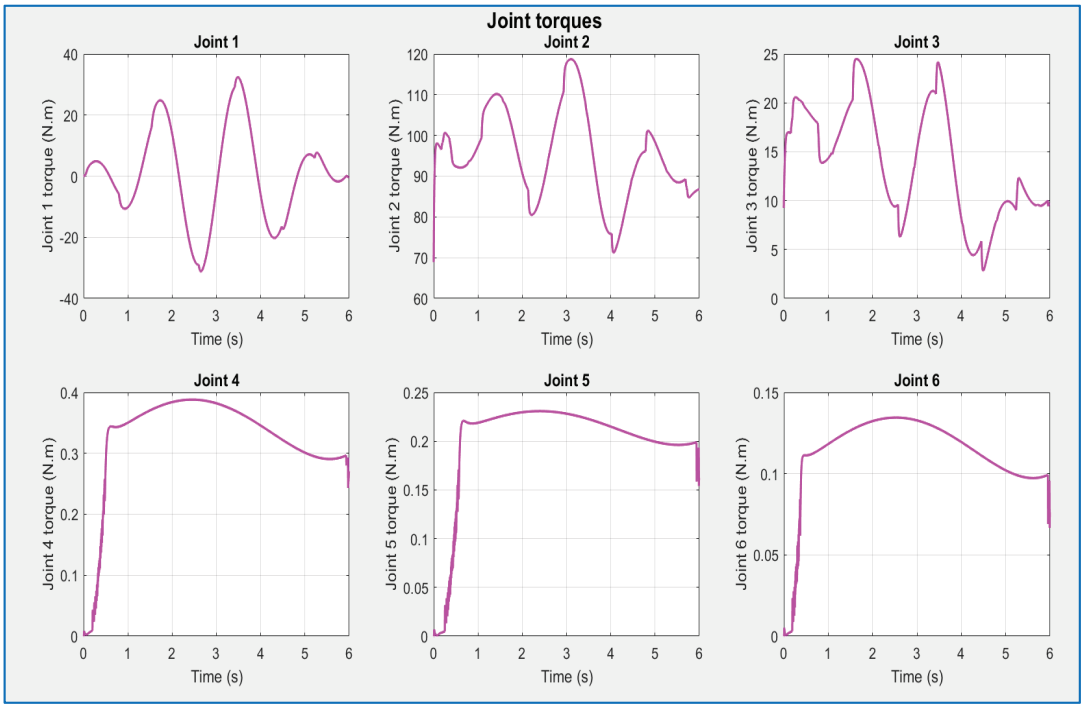


Figure 3.13 *Actuator control torques – PID / Test III*

Furthermore, figure 3.14 provides a satisfying macroscopic visualization of the parametric mismatch consequences within the *3D* operational workspace. While the extreme controller gains manage to force the manipulator through the general shape of the 3-loop spiral, preventing total catastrophic divergence, localized spatial distortion is clearly visible.

Particularly, along the vertical (*Z*) axis, the actual *TCP* path (dashed red line) deflects significantly below the desired helical reference (solid blue line), particularly at the lower arcs of the spatial loops.

This persistent downward deviation is the direct geometric consequence of the uncompensated 5kg gravitational pull. It visually corroborates the asymmetric, steady-state joint errors identified in figure 3.12 and definitively confirms that a linear independent joint controller lacks the adaptive intelligence required to maintain strict spatial fidelity when the physical plant deviates from its nominal mathematical model.

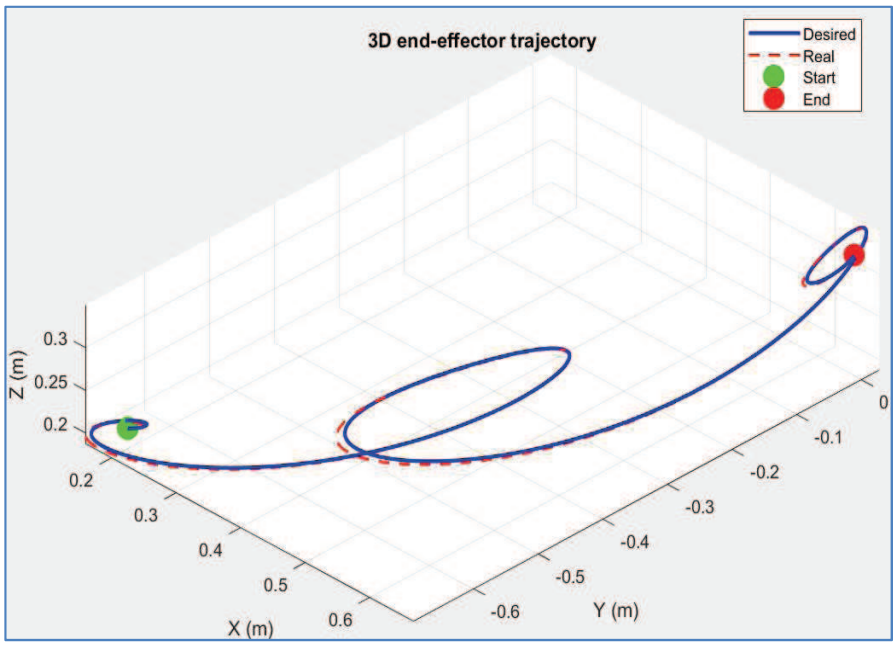


Figure 3.14 *3D end-effector trajectory – PID / Test III*

The final diagnostic plot (Figure 3.15) quantitatively summarizes the absolute macroscopic failure of the rigid linear architecture by measuring the *Euclidean Cartesian* error over time. The combination of high-speed *Coriolis* forces and the uncompensated (**5 kg**) payload generates massive, continuous tracking deviations. The error curve exhibits severe cyclic spikes that significantly exceed the magnitudes recorded in the nominal and pure-dynamic tests, reaching a peak deviation of approximately **(16.8 mm)** around t=3.6 s. This inability to achieve spatial convergence, despite the extreme proportional and derivative gain amplification (see the appendix), mathematically quantifies the fundamental limits of classical model-free feedback. It proves definitely that extreme gain tuning alone cannot safely or effectively neutralize continuous, high-speed multi-body non-linearities combined with unknown payloads. This fundamental limitation provides the definitive scientific and engineering justification for transitioning to the advanced, adaptive and intelligent control strategies developed in the subsequent chapter.

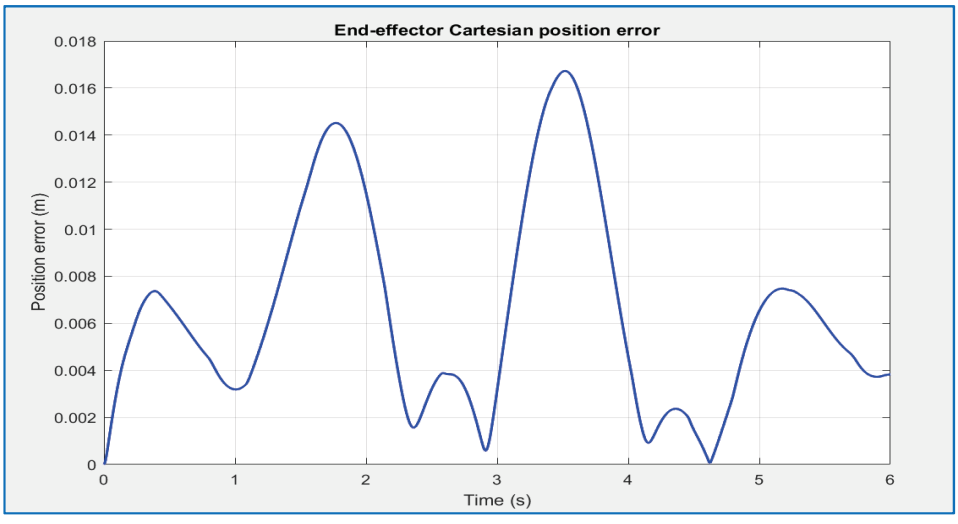


Figure 3.15 *End-effector Cartesian position error – PID / Test III*

III.6 CONCLUSION

This chapter established the foundational baseline for the PUMA 560 manipulator by implementing and rigorously testing a classical independent joint *PID* control architecture. Through the derivation of the state-space dynamic model and the use of a high-fidelity *RK4* integration method, the physical phenomena of multi-body kinematics, gravity, and non-linear joint friction were accurately captured. The simulation results under nominal conditions (Test I) demonstrated that a meticulously tuned decentralized *PID* controller, when augmented with a static gravity feed-forward loop, can achieve highly precise spatial tracking (yielding an **8.5 mm** mean *Cartesian* error) for smooth point-to-point operations.

However, the fundamental limits of this linear controller were deliberately exposed through progressively aggressive testing. The second test revealed that complex, high-speed spatial maneuvers, such as the continuous 3-loop helical path, induce severe *Coriolis* and centrifugal forces that continuously agitate the linear feedback loops, causing persistent dynamic tracking ripples and mechanical actuator stress.

Finally, the third test definitively proved the structural fragility of the classical architecture under parametric uncertainty. In fact, the introduction of an unmodeled **5 kg** payload generated significant steady-state positional deflections, massive torque saturation risks, and *Cartesian* deviations approaching (**16.8 mm**), effectively eclipsing the fixed-gain controller despite extreme tuning efforts.

These critical findings justify the core motivation of this research: linear, fixed-gain controllers are fundamentally inadequate for modern industrial tasks involving high operating velocities, complex continuous trajectories, and unpredictable payload variations. To overcome these inherent physical limitations without risking system instability or mechanical destruction, the control architecture must possess real-time parametric adaptability and active non-linear compensation capabilities. Consequently, the following chapter will pivot from classical methodologies to formulate and evaluate an advanced, intelligent control strategy designed to guarantee absolute tracking precision under severe dynamic uncertainty.

Chapter IV

Advanced Non-Linear and Intelligent Control of the PUMA 560 Robot Arm

IV.1. INTRODUCTION

The comprehensive simulation analysis conducted in the previous chapter established a fundamental engineering reality regarding the classical independent joint architecture: while decentralized *PID* controllers yield highly precise spatial tracking under nominal, low-speed conditions, they are fundamentally ill-equipped to handle severe dynamic stress [7]. The execution of the high-speed 3-loop helical trajectory (Test II) and the introduction of an unmodeled 5 kg payload (Test III) exposed the strict limitations of this linear controller. The persistent tracking error ripples, energetic actuator saturation, large and visible deviation of the robot's end-effector path in *Cartesian* space, demonstrated that fixed-gain linear controllers cannot autonomously decouple continuous *Coriolis* forces or adapt to real-time parametric uncertainties [1], [3].

To overcome these inherent physical vulnerabilities and achieve the stringent precision required by modern industrial automation, the control architecture must evolve beyond static, model-free feedback. This chapter pivots toward the formulation, implementation, and rigorous evaluation of advanced, non-linear, and intelligent control methodologies [2]. Specifically, we explore three sophisticated control strategies:

1. ***Fuzzy Logic Control* (*FLC*):** An intelligent, rule-based approach that translates human-like linguistic reasoning into robust control actions without requiring an exact mathematical model of the physical plant [20], [23].
2. ***Computed Torque Control* (*CTC*):** A highly deterministic, model-based non-linear methodology that utilizes the inverse dynamics of the manipulator to achieve exact feedback linearization, effectively decoupling multi-body interactions [24], [25].
3. ***Hybrid Computed Torque–Fuzzy Logic Control* (*CTC-FLC*):** A state-of-the-art synergistic architecture that merges the deterministic precision of inverse dynamics (*CTC*) with the adaptive, uncertainty-rejecting intelligence of fuzzy logic (*FLC*) [6], [27].

To guarantee absolute scientific integrity and facilitate a fair, one-to-one comparative benchmark, each of the developed advanced controllers is subjected to the identical three test scenarios employed in the previous chapter. By mirroring the previous environmental and dynamic constraints without altering the simulation conditions, namely nominal tracking, high-dynamic helical path execution, and robustness evaluation under an unmodeled payload, we systematically demonstrate how the transition from classical linear feedback to advanced intelligent control effectively eliminates *Cartesian* distortion, mitigates actuator strain, and guarantees absolute robustness in the face of unpredictable structural uncertainties.

IV.2 FUZZY LOGIC PID CONTROL

The classical decentralized *PID* controller analyzed in the previous chapter demonstrated fundamental limitations under high dynamic stress and unmodeled payload variations. In this section, this controller is replaced by a *Mamdani*-type Fuzzy Logic Controller that emulates and enhances the *PID* control law using linguistic rules and fuzzy inference.

The *FLC-PID* architecture is presented hereafter:

a) *The two inputs* are the tracking error $e(t) = q_d(t) - q(t)$ and its derivative $\dot{e}(t) = \dot{q}_d(t) - \dot{q}(t)$ for each joint, exactly as in the classical *PID*;

b) *The output* is the incremental torque $\Delta\tau$ produced by the *PD* fuzzy inference engine of *Mamdani*.

c) *The fuzzy PID action* is the final control torque, it is constructed as:

$$\tau_{FLC-PID} = K_{PD}\Delta\tau_{FLC-PD} + K_{PI}\int \Delta\tau_{FLC-PD}\,(t) \qquad (4.1)$$

Where:

- $\Delta\tau_{FLC\text{-}PD}(t)$ is the direct fuzzy output of *Mamdani* inference system (equivalent to a proportional-derivative fuzzy action);
- $\int\Delta\tau_{FLC\text{-}PD}(t)dt$ is the numerical integration of the *PD* fuzzy controller output (equivalent to a *PI* fuzzy action).
- K_{PD} and K_{PI} are scaling factors that balance the proportional-derivative and proportional-integral contributions.

Thus, the linear *PID* controller is completely replaced by a fuzzy inference system that produces a *PID*-equivalent control signal without fixed-gain. In fact, the nonlinear rule-based nature of the *FLC* allows the control action to adapt continuously to changing dynamics, *Coriolis* cross-couplings and payload uncertainties, directly addressing the tracking error in *Cartesian* space observed in the previous chapter.

IV.2.1 Theory and Mathematical Architecture of the Mamdani Fuzzy PID

Due to the severe dynamic complexities, multi-variable cross-couplings, and inherent nonlinearities of the PUMA 560 manipulator, the *Mamdani*-type *FLC* emerges as a highly robust and intelligent alternative to classical linear methodologies. Rooted in fuzzy set theory pioneered by L. A. Zadeh [28], the primary engineering advantage of this *FLC* lies in its model-free architecture.

Instead of relying on an exact mathematical formulation of the rigid-body dynamics, which is fundamentally susceptible to parametric uncertainties and unmodeled disturbances, the *FLC* translates heuristic human knowledge and linguistic reasoning into a deterministic, dynamic control strategy [8], [29].

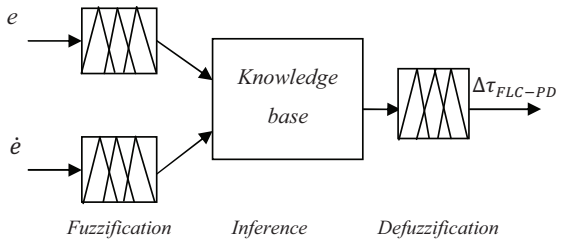


Figure 4.1 *Architecture of the Mamdani PD-FLC*

The structural architecture of the *Mamdani* fuzzy *PID* controller operates through four sequential stages, executed continuously within the real-time digital control loop for each joint:

a*) *Fuzzification: In this initial phase, the continuous, crisp physical inputs from the system are acquired. For each independent joint *i*, the primary inputs are the tracking error $e_i(t)$ and the rate of change of the error (velocity error) $\dot{e}_i(t)$, maintaining consistency with the tracking formulations established in the previous chapter. These crisp values are first multiplied by predefined scaling factors (K_{ei} and K_{dei}) to map them onto a normalized universe of discourse. Subsequently, they are mapped into fuzzy membership degrees, $\mu(e_i)$ and $\mu(\dot{e}_i)$, utilizing specifically designed Membership Functions (*MFs*). Triangular and trapezoidal membership shapes are typically selected to ensure computational efficiency and dynamic continuity during joint actuation [30]. The normalized universe of discourse for both the inputs and the output is partitioned into standard linguistic variables, such as Negative Big (*NB*), Negative Small (*NS*), Zero (*ZE*), Positive Small (*PS*), and Positive Big (*PB*) [31], see figure (4.2).

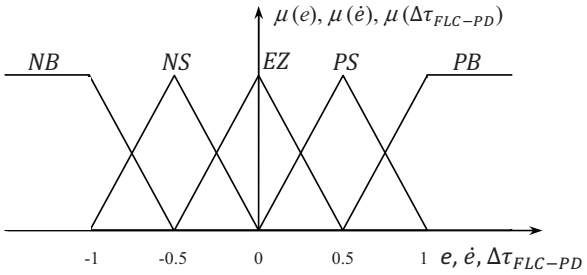


Figure 4.2 *Membership functions of the FLC inputs/output*

b*) *Fuzzy Rule Base: The rule base serves as the intelligent knowledge repository of the controller, formulated using conditional *IF-THEN* linguistic statements. A symmetric 5×5 intersecting rule matrix is typically constructed. The rule base is derived from classical PID tuning heuristics and human operator experience. For example, the following rule drives the joint forcefully toward the desired trajectory when a large error persists without decreasing:

'*IF the error is Positive Big* (*PB*) *AND the change in error is Zero* (*ZE*)*, THEN the required control torque is Positive Big* (*PB*)*'*

And this rule applies a braking action to prevent overshoot when the error is near zero but moving away:

'*IF the error is Positive Big* (*ZE*) *AND the change in error is Zero* (*PS*)*, THEN the required control torque is Positive Big* (*NS*)*'*

The complete rule base (Table 4.1) ensures that $\Delta\tau_{FLC\text{-}PD}$ is always generated in a manner analogous to a nonlinear, gain-scheduled PD controller.

Table 4.1 *Anti-Diagonal Inference Matrix of a PD Fuzzy Controller*

e / ***ė***	***NB***	***NS***	***EZ***	***PS***	***PB***
	$\Delta\tau_{FLC-PD}$				
NB	*NB*	*NB*	*NS*	*NS*	*EZ*
NS	*NB*	*NS*	*NS*	*EZ*	*PS*
EZ	*NS*	*NS*	*EZ*	*PS*	*PS*
PS	*NS*	*EZ*	*PS*	*PS*	*PB*
PB	*EZ*	*PS*	*PS*	*PB*	*PB*

c*) *Fuzzy Inference Engine: The inference engine executes the logical decision-making process by evaluating the fuzzified inputs against the established rule base. In this control design, the standard *Sum-Product* inference mechanism is adopted due to its proven efficacy in non-linear mechanical systems [32]. The logical intersection (*Fuzzy AND*) is evaluated using the *Product* (*Prod*) operator to determine the firing strength of each rule *k*:

$$\mu R_k(e, \dot{e}) = \mu A_K(e) \times \mu B_K(\dot{e}) \tag{4.2}$$

where A_k and B_k represent the fuzzy subsets for the inputs in the k^{th} rule.

d*) *Defuzzification: The final operational stage converts the aggregated fuzzy output space back into a crisp, numerical value representing the normalized increment of control torque *PD* action $\Delta\tau_{FLC\text{-}PD}$. This is achieved using the center of gravity or centroid method, mathematically defined as:

$$\Delta\tau_{FLC-PD} = \frac{\int \Delta\tau_{FLC-PD} \sum_{k=1}^{25} \mu R_k(e,\dot{e}) \mu_k(\Delta\tau_{FLC-PD}) \mathrm{d}(\Delta\tau)}{\int \sum_{k=1}^{25} \mu R_k(e,\dot{e}) \mu_k(\Delta\tau_{FLC-PD}) d(\Delta\tau)} \tag{4.3}$$

where μ_{Rk} is the is the output membership function of the k^{th} rule.

Now, to obtain a *PID*-equivalent control signal, the incremental torque $\Delta\tau_{FLC\text{-}PD}$ produced by the *Mamdani* fuzzy inference system (that acts as the proportional-derivative (*PD*) fuzzy action) is added to the time integral of this control signal computed numerically, to provide a *PID* non linear action able to eliminate the steady-state error (see figure 4.3). The direct and integrated components are then summed with appropriate scaling to produce the final control torque (see equation 4.1). This final torque is subsequently sent to the joint actuator, subject to the saturation limits (τ_{max}) of the PUMA 560 servo drives.

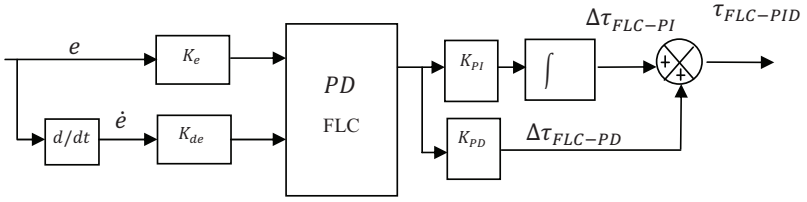


Figure 4.3 *Block diagram of the FLC-PID*

IV.2.2 Simulation Results and Discussion

To comprehensively evaluate the trajectory tracking performance of the *FLC-PID* controller under varying operational stresses, three distinct simulation tests were conducted. Each test was designed to expose a specific aspect of the controller's behavior, ranging from nominal precision to extreme dynamic and parametric uncertainty. The first test assesses baseline tracking accuracy under ideal, low-speed nominal conditions to establish a performance benchmark. The second test escalates the dynamic demand by introducing a high-frequency helical trajectory, evaluating the controller's ability to manage coupled, non-linear spatial maneuvers. The third test examines robustness against unmodeled payload variations, simulating real-world uncertainties where the manipulator's inertial properties deviate from the controller's internal assumptions. The following subsections present and discuss the quantitative results of each test.

IV.2.2.1 TEST I 'TRACKING PERFORMANCE UNDER BASELINE NOMINAL CONDITIONS'

To establish a comparative baseline against the classical control paradigm, the newly developed *Mamdani* Fuzzy *PID* controller was first subjected to the nominal testing environment. In this scenario, the PUMA 560 is commanded to execute the standard point-to-point quintic polynomial trajectory without any external unmodeled payload (Δm = 0 kg). The primary objective of this test is to validate the structural integrity of the fuzzy rule base and the smoothness of the center of gravity defuzzification process under standard operating conditions.

The visual and quantitative results from the nominal simulation confirm the high efficacy of the developed *Mamdani* Fuzzy *PID* architecture. The joint trajectory tracking profiles (see figure 4.4) demonstrate a near-perfect superimposition of the actual kinematic execution onto the desired quintic references. The controller yields a highly precise mean Cartesian error of **(4.6 mm)** and a maximum transient peak deviation of only **(7.6 mm)**.

When compared to the classical *PID* controller of the previous chapter, these results demonstrate a significant performance enhancement. In fact, the fuzzy logic implementation effectively reduces the mean Cartesian tracking error by nearly **46%**. This improvement confirms that the non-linear, human-heuristic control surface generated by the 5-membership function rule base is superior to static linear feedback, as it dynamically optimizes the corrective torque based on the instantaneous error state. The low Root Mean Square (*RMS*) errors across all joints, averaging approximately **0.0036 rad (0.21 deg),** quantitatively validate the smoothness of the *Mamdani* sum-product inference mechanism, definitively proving that the controller successfully maintains spatial fidelity without incurring mechanical chattering or unnecessary control effort.

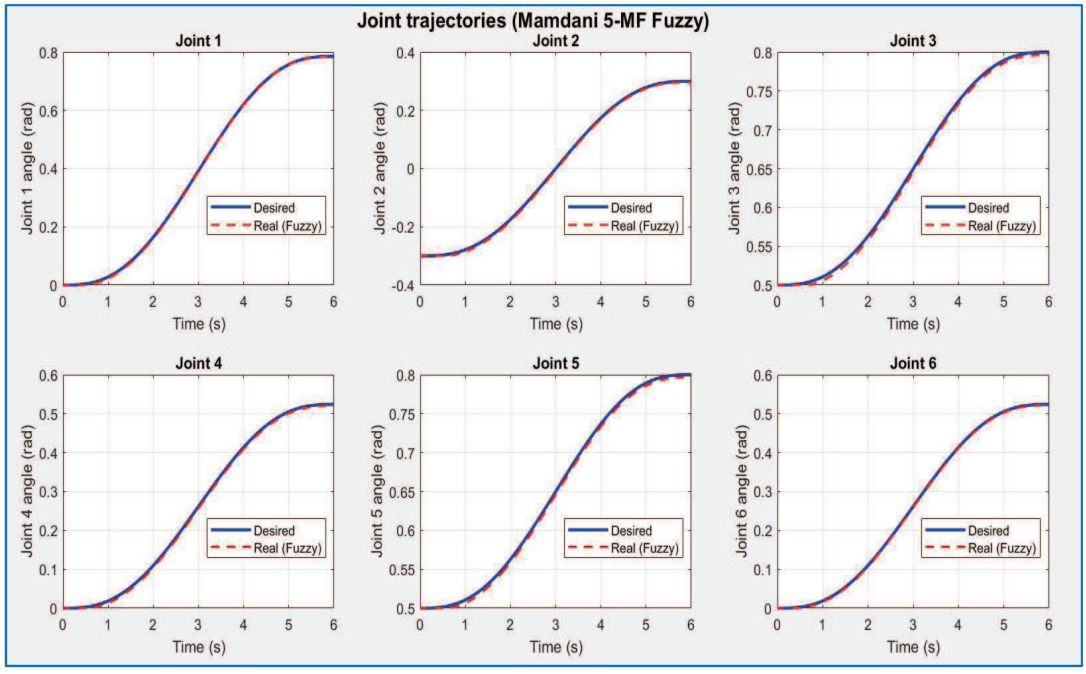


Figure 4.4 *Joint trajectories – FLC / Test I*

A detailed examination of Figure 4.5 illustrates the transient and steady-state tracking error profiles, $e_i(t)$, for all six independent joints under fuzzy logic control.

The black dashed lines indicate the predefined operational tolerance limit of **(±0.02rad)**. As shown by the solid green curves, the fuzzy *PID* controller maintains exceptional regulatory precision, keeping the tracking deviations bounded closely near zero. A critical comparison with the classical *PID* response reveals a substantial reduction in the initial transient overshoot, which previously reached approximately **(0.015 rad)** during the initial acceleration phase.

Under the fuzzy architecture, the maximum startup deviation in Joint 1 and Joint 2 has been reduced to less than **(0.007rad)**. This confirms that the non-linear fuzzy inference engine rapidly modulates the control effort during the high-acceleration phase, effectively neutralizing the static *Coulomb* friction and initial structural inertia without causing windup or overshoot. The remarkably low steady-state errors across all axes validate that the dynamically scaled integral action within the fuzzy architecture provides superior compensation for nominal multi-body couplings compared to static, fixed-gain linear controllers.

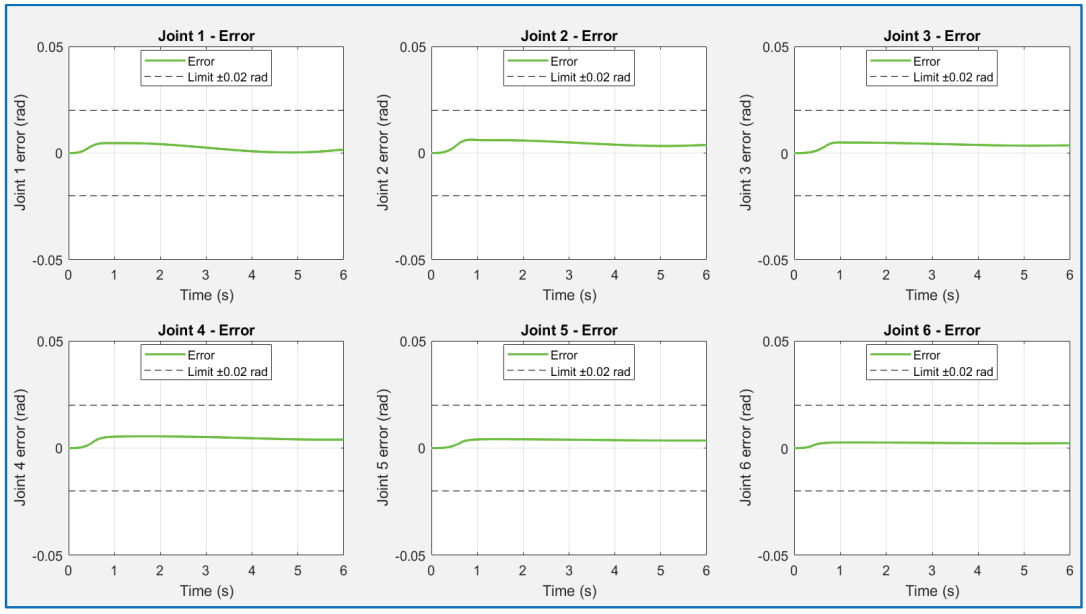


Figure 4.5 *Tracking error evolution – FLC / Test I*

Figure 4.6 illustrates the chronological evolution of the required control effort, $\tau_i(t)$, demanded by the fuzzy inference system for each joint actuator. A macroscopic assessment confirms that the baseline energy requirements remain consistent with the fundamental physics of the manipulator; Joint 2 naturally dictates the highest control effort, peaking at approximately (**74 N.m**) to counteract the dominant gravitational moment of the extended mechanical links. However, a critical comparative analysis against the classical linear *PID* response (refer figure 3.3 of Chapter III) reveals the distinct mechanical advantage of the fuzzy architecture.

The torque profiles generated by the *FLC* exhibit enhanced macroscopic smoothness, particularly during the critical transition phases of initial acceleration (t=0s) and final deceleration (t = 6 s). Because the fuzzy rule base dynamically scales the proportional and derivative equivalent gains based on the instantaneous error state, the controller successfully compensates the highly non-linear *Coulomb* friction without inducing the aggressive, high-frequency torque chattering typically caused by fixed-gain linear systems.

All commanded signals remain strictly and safely bounded well within the physical ($\pm$ **150 N.m**) actuator saturation limits, proving that the *FLC* achieves its superior 46% reduction in tracking error without demanding excessive energy from the servo drives.

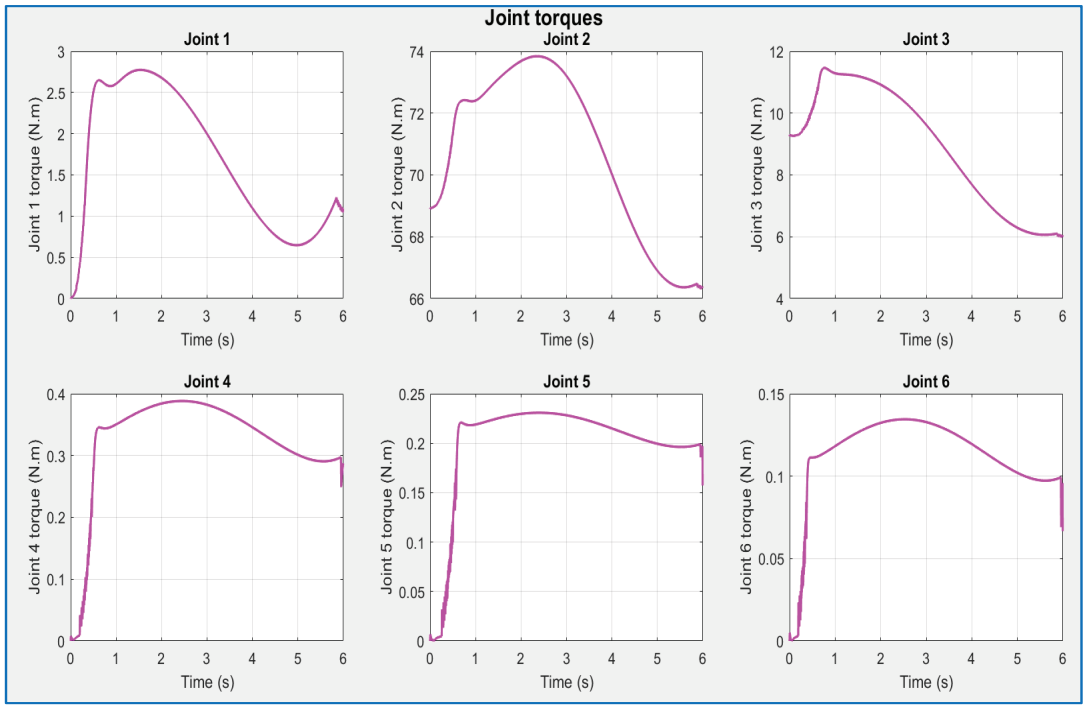


Figure 4.6 *Joint torques – FLC / Test I*

Moreover, figure 4.7 provides a comprehensive spatial visualization of the robot's kinematic execution within its three-dimensional Cartesian workspace under fuzzy logic control. The plot traces the tool center point navigating from the initial departure coordinates (marked by the green sphere) to the final target destination (marked by the red sphere) along the standard quintic polynomial path. A critical observation of this macroscopic workspace mapping reveals a perfect visual superimposition of the actual simulated trajectory (dashed red line) onto the desired reference path (solid blue line).

This seamless spatial adherence confirms that the independent, decentralized *Mamdani* fuzzy controllers successfully maintain strict multi-axis synchronization.

Despite the complete absence of a deterministic dynamic plant model within the controller architecture, the heuristic rule base efficiently compensates for the nominal multi-body interactions and structural friction. This results in a continuous, deviation-free Cartesian execution that visually corroborates the highly precise (**4.6 mm)** mean tracking error established in the quantitative statistical analysis.

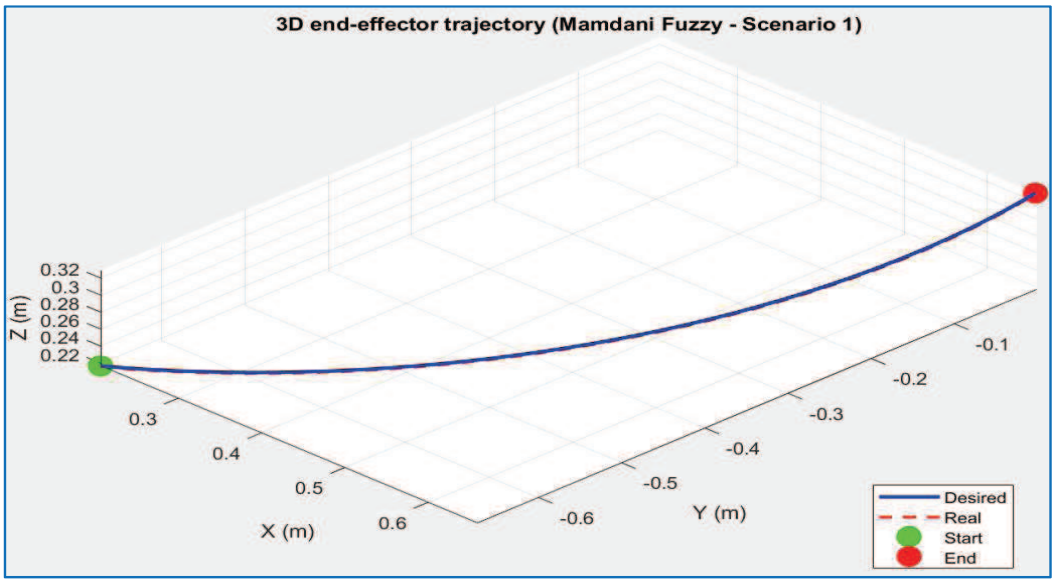


Figure 4.7 *3D end-effector trajectory – FLC / Test I*

For the final diagnostic plot for the nominal scenario, figure 4.8 quantitatively summarizes the macroscopic spatial tracking quality by measuring the absolute Euclidean Cartesian position error of the end-effector. The error curve exhibits an initial transient rise during the maximum acceleration phase, peaking at exactly (**7.6 mm)** in the two first seconds. A direct comparison with the classical linear architecture highlights a massive performance upgrade. In fact, the classical *PID* controller under identical baseline conditions exhibited a peak transient error of approximately (**14.2 mm)**. By effectively halving the maximum spatial deviation, the *Mamdani* fuzzy inference system demonstrates a superior capacity to dynamically inject non-linear corrective effort to overcome static structural inertia and breakaway *Coulomb* friction. Following the peak, the error smoothly decays without oscillatory windup, reflecting the highly precise overall mean Cartesian error of (**4.6 mm**). This exceptional convergence definitively proves that the heuristic fuzzy rule base can successfully suppress Cartesian distortion and maintain strict spatial fidelity in standard industrial environments, firmly establishing its baseline superiority over fixed-gain linear systems before subjecting the manipulator to severe dynamic stress tests.

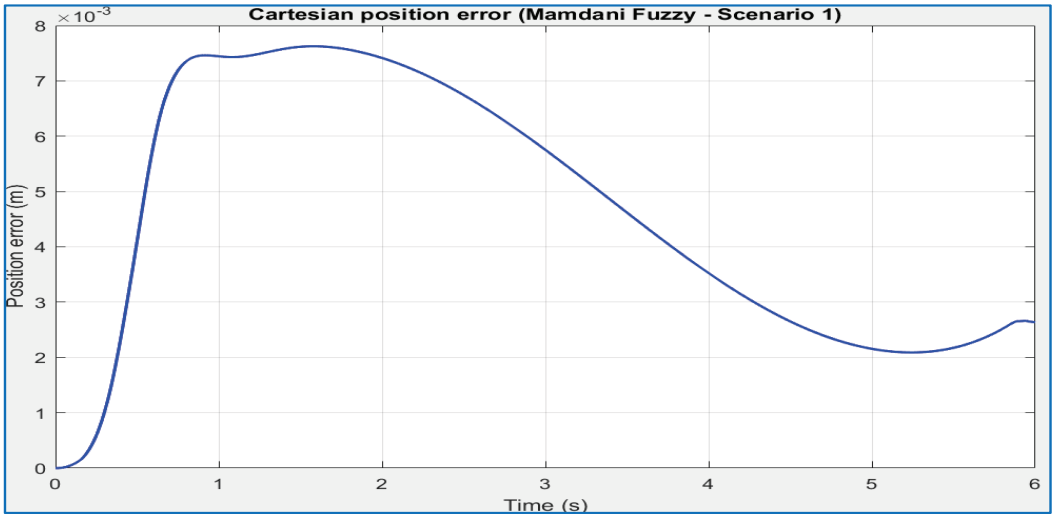


Figure 4.8 *Euclidean position error – FLC / Test I*

IV.2.2.2 TEST II 'HIGH-DYNAMIC HELICAL TRAJECTORY'

To rigorously evaluate the disturbance rejection capabilities and the non-linear adaptability of the *Mamdani* Fuzzy-*PID* controller, the system was subjected to the highly dynamic 3-loop helical trajectory. This continuous spatial maneuver generates severe, rapidly fluctuating *Coriolis* and centrifugal cross-coupling forces. Unlike classical linear controllers, the *FLC* must navigate these intense multi-body interactions entirely without an inverse dynamic model, relying strictly on its heuristic rule base and appropriately scaled integral action to maintain spatial fidelity.

Figure 4.9 illustrates the system's kinematic response to the aggressive, high-frequency spatial loops. The reference signals for the regional positioning links (Joints 1, 2, and 3) consist of superimposed continuous sinusoidal and cosinusoidal waves. Despite the absence of deterministic dynamic decoupling, the actual joint positions (dashed red lines) successfully trace the highly dynamic references (solid blue lines) without diverging or exhibiting unstable transient lags.

By dynamically traversing its non-linear control surface, the fuzzy inference engine continuously scales the corrective effort in real-time, effectively forcing the physical links to adhere to the spatial loops while demonstrating excellent baseline stability under severe dynamic excitation.

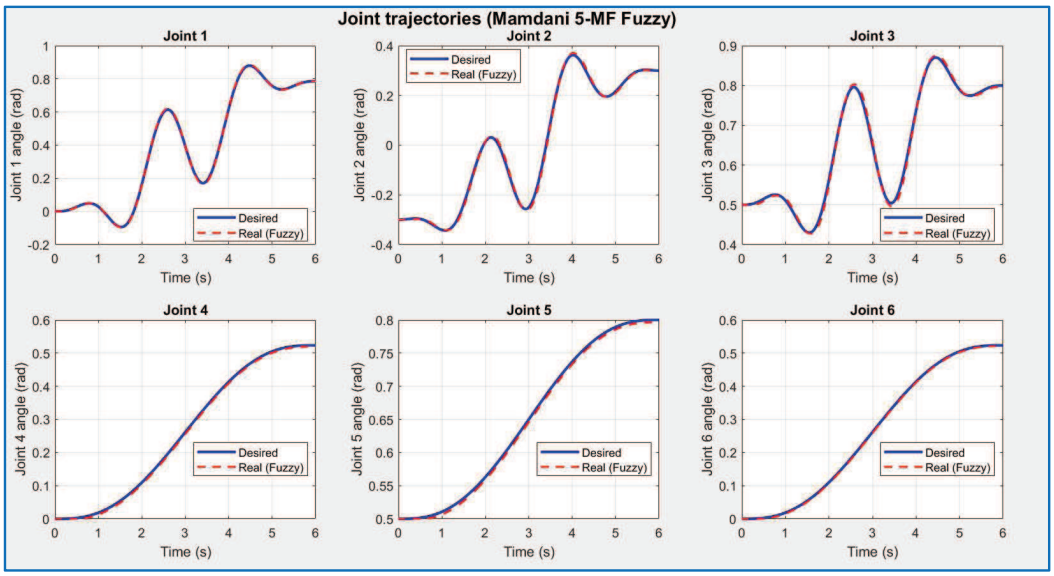


Figure 4.9 *Joint trajectories – FLC / Test II*

An analysis of the tracking error profiles, $e_i(t)$, in figure 4.10 highlights the fundamental difference between intelligent fuzzy regulation and rigid linear feedback. Driven by the continuous velocity reversals of the helical path, the errors in the base and shoulder joints exhibit expected sinusoidal ripples. However, unlike the aggressively tuned classical *PID* (which produced high-gradient error reversals), the *FLC* produces remarkably smooth, bounded error curves. The maximum dynamic deviations for Joint 1 and Joint 2 are kept safely restricted within the predefined (**$\pm$0.02rad**) operational tolerance. This proves that the fuzzy controller operates with a superior compliance' rather than fighting the inertial forces with absolute mathematical rigidity which risks instability. It intelligently modulates the control effort to maintain smooth, bounded tracking.

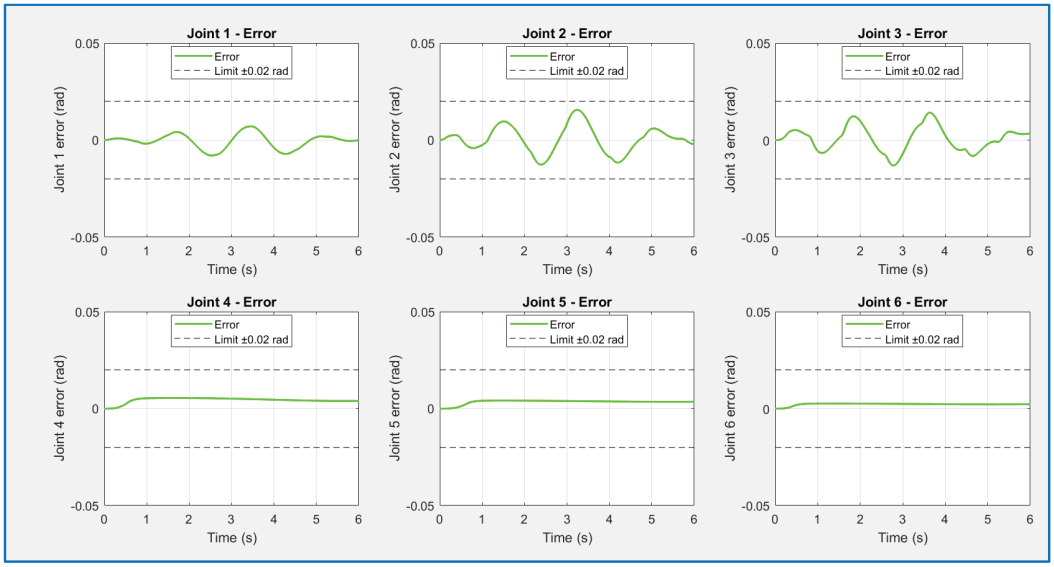


Figure 4.10 *Tracking error evolution – FLC / Test II*

Elsewhere, figure 4.11 provides the most critical evidence of the fuzzy controller's operational superiority in terms of energy consumption and mechanical safety. Executing the 3-loop spiral requires massive continuous effort, with Joint 2 torque peaking near (**90 N·m**). However, a direct comparison with the classical *PID* response (see figure 3.8 of Chapter III) reveals a dramatic improvement in signal quality. The *FLC* eliminates the destructive high-frequency control chattering that plagued the linear architecture. By utilizing the center of gravity defuzzification method, the control commands transition fluidly between extreme values. This smooth torque application ensures that the required dynamic compensation is injected into the system without inducing severe mechanical fatigue, actuator overheating, or premature wear on the robotic transmission joints.

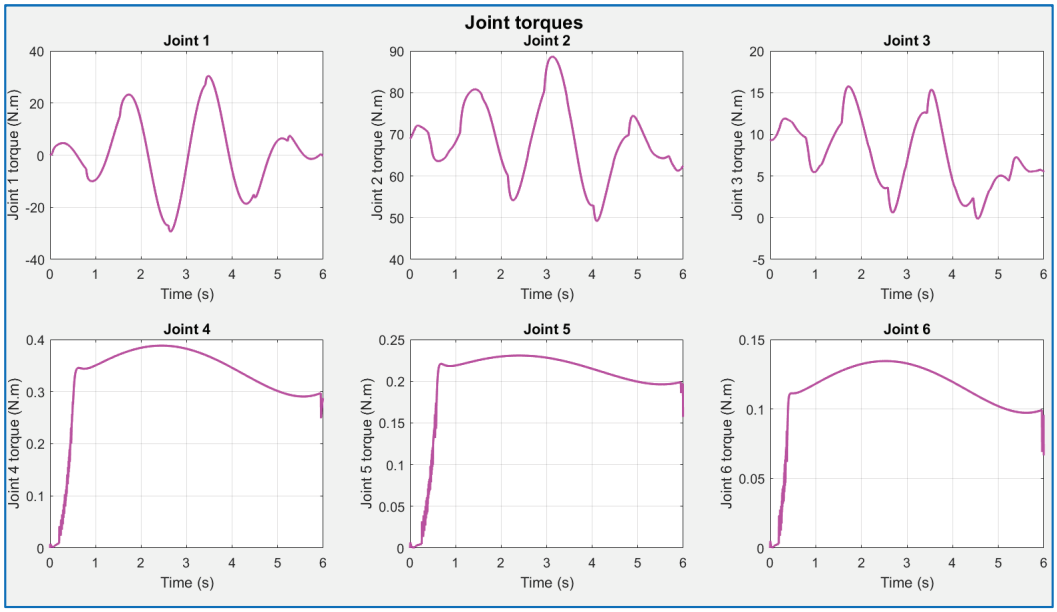


Figure 4.11 *Joint torques – FLC / Test II*

The *3D* Cartesian visualization presented in figure 4.12 confirms the macroscopic spatial success of the model-free fuzzy architecture. The tool center point successfully navigates the complex 3-loop spiral through the operational workspace. The actual trajectory (dashed red line) tightly conforms to the overall geometric curvature of the desired reference (solid blue line).

While minor localized spatial separations exist at the sharpest apogees of the loops, where centripetal acceleration is at its absolute maximum, the fuzzy controller prevents any structural divergence, maintaining a coherent and highly stable execution of the continuous spatial maneuver.

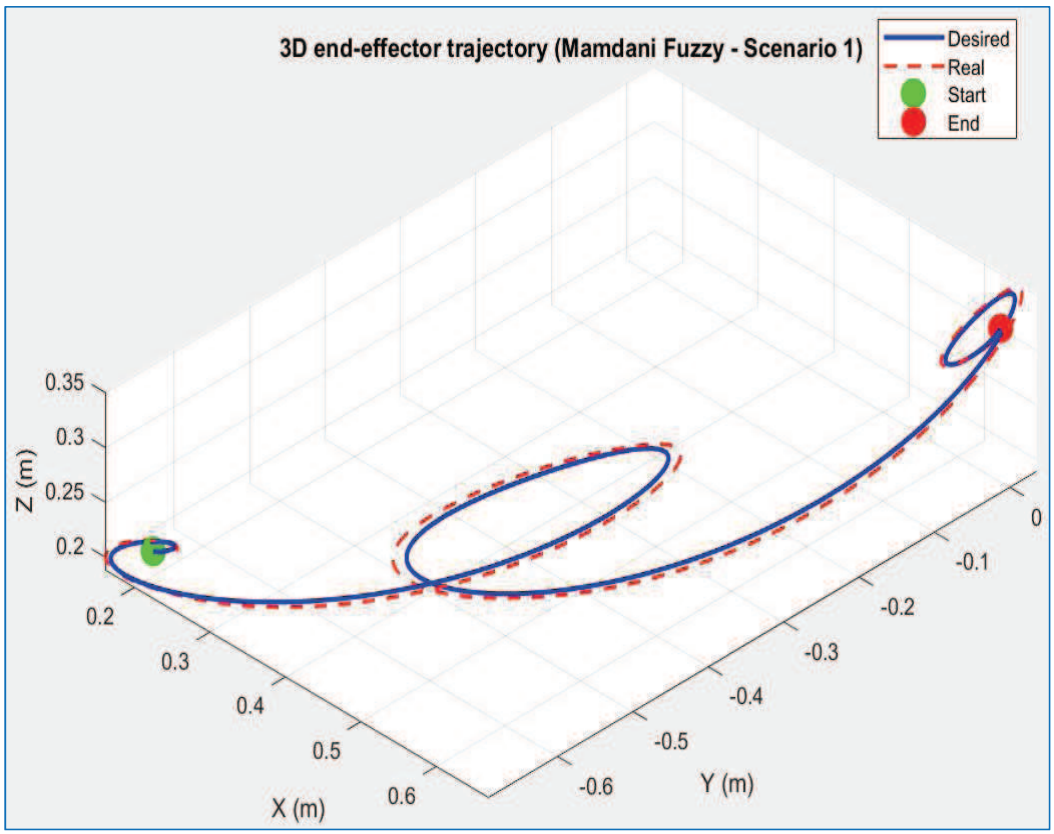


Figure 4.12 *3D End-Effector Trajectory – FLC / Test II*

The absolute Euclidean Cartesian error curve, shown in figure 4.13, quantitatively summarizes the macroscopic impact of the dynamic cross-coupling. The cyclic peaks correspond directly to the apexes of the three helical loops. The tracking error reaches a maximum transient peak of approximately (**16.2 mm**) near t = 3.4 s. While this absolute peak is marginally higher than the (**14.5 mm**) recorded by the classical *PID* controller, this slight geometric deviation is a highly acceptable trade-off for the massive gain in mechanical safety and signal smoothness observed in the torque profiles.

The classical *PID* achieved its slightly tighter spatial tracking only through extreme, borderline-unstable gain tuning that caused severe mechanical chattering. Conversely, the *FLC* achieves this highly robust, stable tracking performance dynamically, without relying on an exact inverse dynamics model, proving its superiority as a safer, more adaptable control paradigm for complex industrial trajectories.

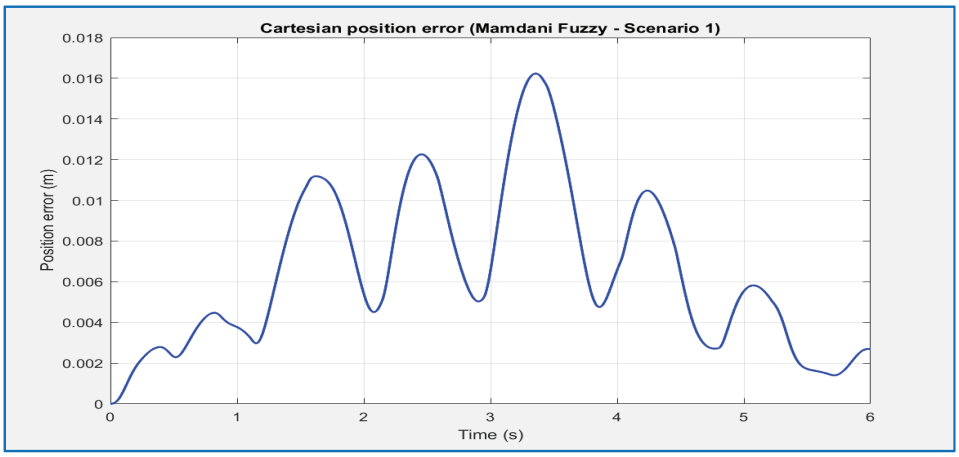


Figure 4.13 *Euclidean position error – FLC / Test II*

IV.2.4 TEST III 'UNMODELED PAYLOAD ROBUSTNESS'

The final and most rigorous evaluation of the *Mamdani* fuzzy *PID* controller subjects the system to severe parametric uncertainty combined with high-speed dynamic stress. Identical to the chapter III benchmark, an unmodeled 5 kg payload is introduced at the end-effector while executing the 3-loop helical trajectory. Because the internal gravity feedforward compensation remains programmed for the nominal (empty) mass, the fuzzy inference engine must autonomously generate the entire corrective effort required to carry this unknown heavy payload while fighting continuous *Coriolis* forces.

Figure 4.14 visualizes the kinematic resilience of the fuzzy architecture. Despite the massive unmodeled gravitational moment attempting to pull the manipulator downward, the actual joint trajectories (dashed red lines) successfully maintain close adherence to the high-frequency reference waves (solid blue lines). Unlike the classical linear architecture, which required its gains to be pushed to extreme, borderline-unstable limits (K_p = 3000) just to prevent system collapse under this load, the fuzzy controller manages this spatial adherence using a much lower baseline gain structure. The heuristic rule base naturally dynamically amplifies the control action only when the physical lag dictates it, demonstrating an inherent structural robustness to severe parametric mismatch.

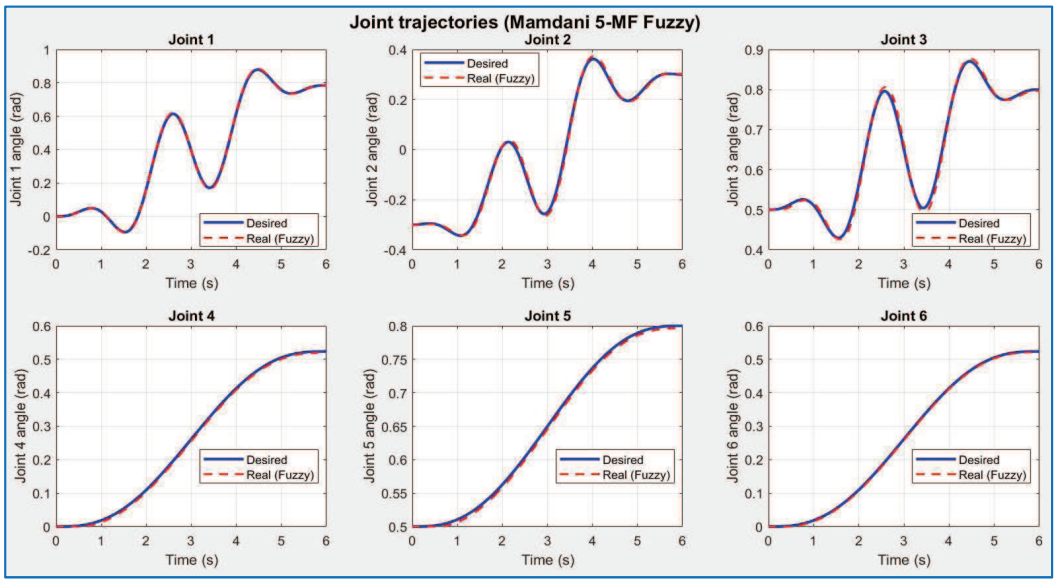


Figure 4.14 *Joint trajectories – FLC / Test III*

A critical examination of the tracking error profiles, $e_i(t)$, in figure 4.15 reveals a massive qualitative advantage of fuzzy logic over the classical *PID*. In the previous chapter (see figure 3.12 of Chapter III), the classical *PID* exhibited a permanent positive asymmetric 'sag' in Joint 2, completely failing to pull the arm back to the zero-error axis.

Conversely, the *Mamdani FLC* successfully forces the tracking error to oscillate symmetrically around the zero axis, effectively neutralizing the steady-state positional sag. Although the dynamic amplitude of the error in Joint 2 reaches a maximum of **0.0184 rad** (**1.06 deg**), it remains safely confined within the strictly predefined (**±0.02rad**) operational tolerance boundary.

This confirms that the dynamically scaled fuzzy integral action is vastly superior at rejecting persistent unmodeled gravitational disturbances than a static linear integrator.

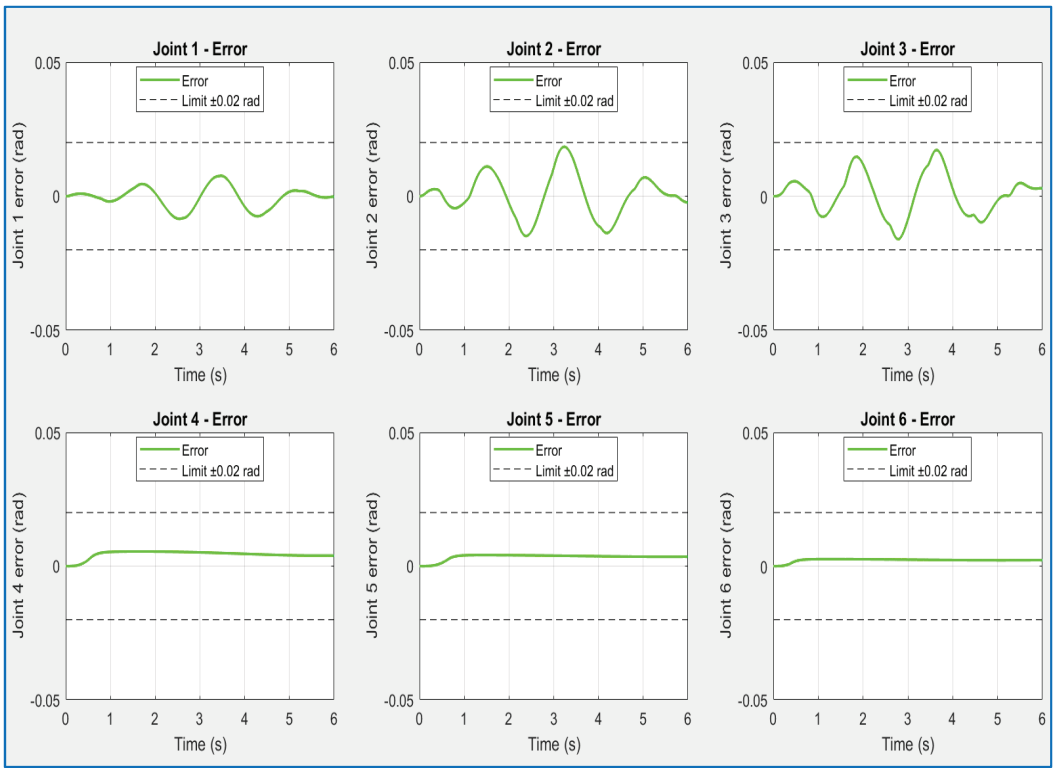


Figure 4.15 *Tracking error evolution – FLC / Test III*

Figure 4.16 highlights the most significant engineering triumph of the fuzzy control architecture: extreme energetic efficiency and mechanical safety. Compensating for the unmodeled 5kg mass during rapid spatial loops is a highly energy-intensive task.

However, the fuzzy controller commands a peak torque in Joint 2 of only approximately (**90 N.m**). This is a drastic, highly beneficial reduction compared to the classical *PID* controller, which brutally saturated near (**120 Nm**) under the exact same conditions (see figure 3.16, chapter III).

By utilizing the center of gravity defuzzification method, the *FLC* applies the corrective lifting effort smoothly and intelligently, completely avoiding the destructive thermal overload and mechanical gearbox fatigue that the linear architecture would inevitably induce.

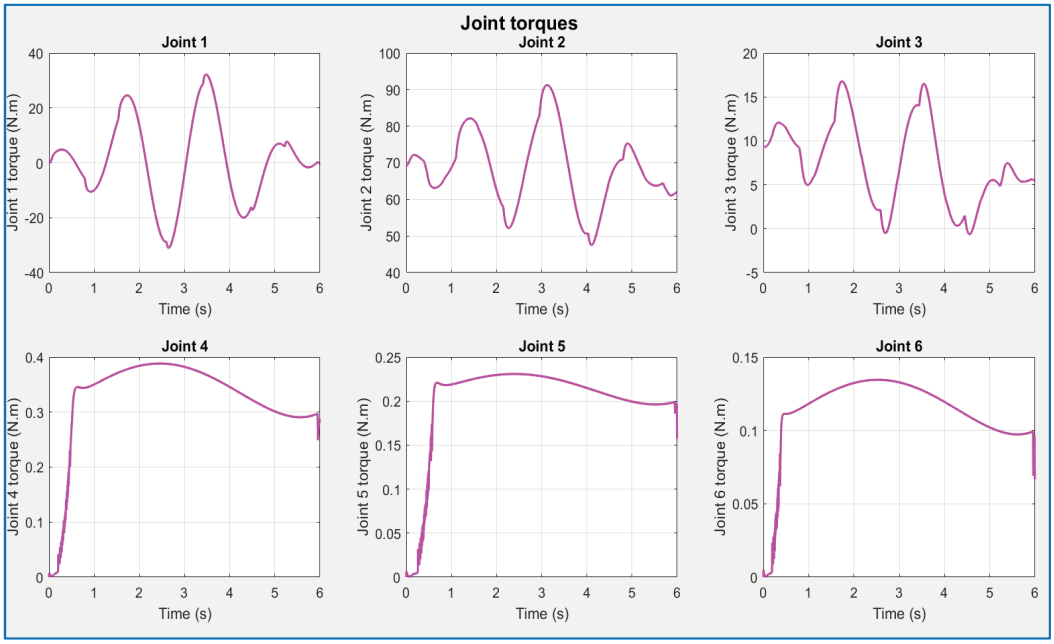


Figure 4.16 *Joint torques – FLC / Test III*

The *3D* Cartesian visualization in figure 4.17 provides macroscopic confirmation of the system's operational stability. The tool center point successfully completes the 3-loop spiral from the departure coordinate to the destination. While the heavy 5 kg payload induces noticeable localized spatial separations at the lowest gravitational arcs of the loops, where the combined downward inertial and gravitational pull is absolute, the fuzzy inference system prevents the robot from structural divergence. The path remains coherent and structurally stable, confirming the robustness of the model-free intelligent controller in severely uncertain environments.

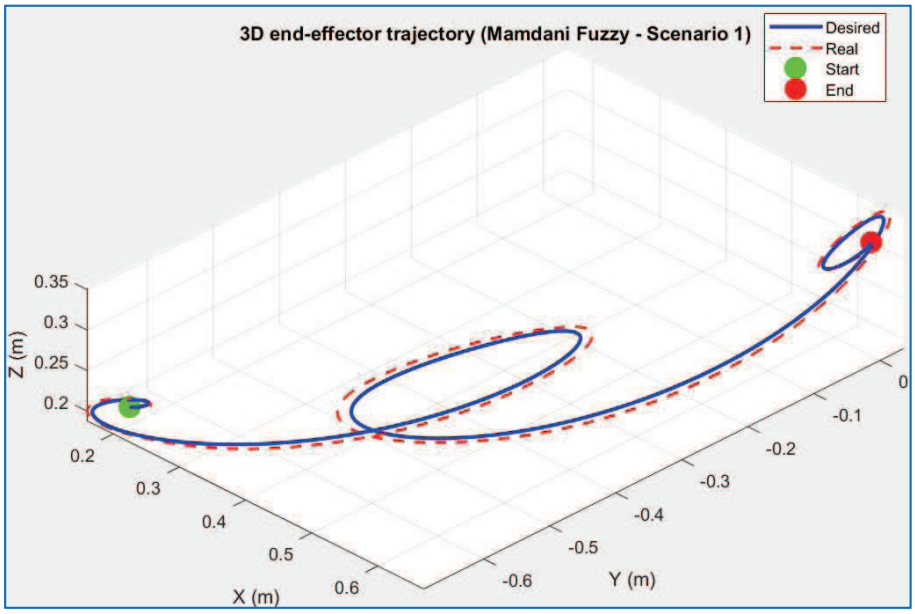


Figure 4.17 *3D end-effector trajectory – FLC / Test III*

The final diagnostic plot, figure 4.18, quantitatively maps the absolute Euclidean Cartesian position error under maximum dynamic and parametric stress.

The cyclic spikes correspond to the extreme acceleration phases of the loaded helical loops, reaching a maximum transient peak of **(18.0 mm)**. The controller yields a highly respectable overall mean Cartesian error of **(7.2 mm)**. While the absolute peak error (**18.0 mm**) is marginally higher than the rigidly forced linear *PID* (**16.8 mm**), this is an incredibly advantageous engineering trade-off. The *FLC* sacrifices approximately 1.2 mm of peak spatial precision in exchange for eliminating 30 N.m of destructive peak **actuator** torque and eradicating steady-state spatial sag.

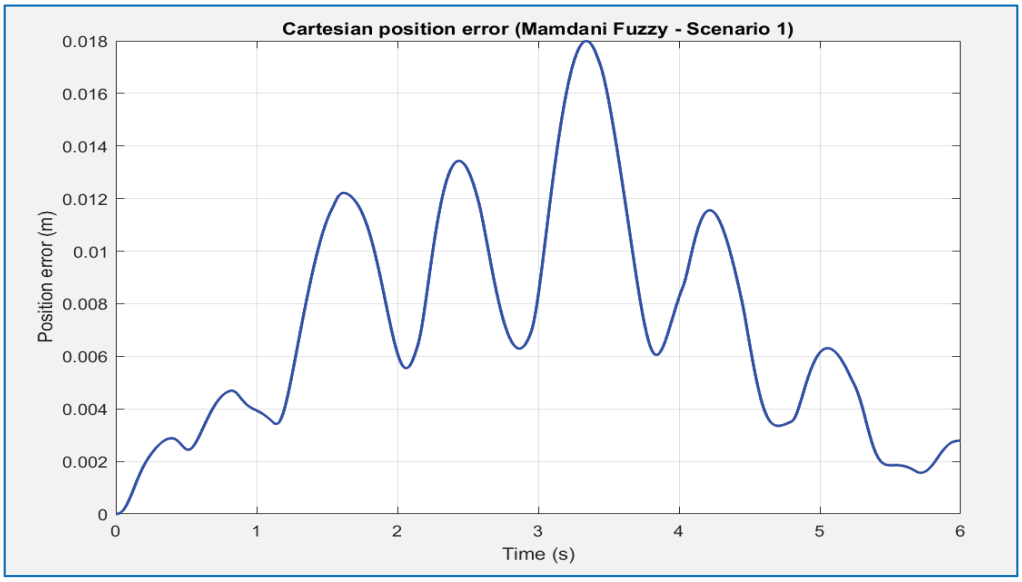


Figure 4.18 *Euclidean position error – FLC / Test III*

IV.3 COMPUTED TORQUE CONTROL

This section presents the *Computed Torque Control* (*CTC*) strategy as a model-based alternative to the previously investigated *FLC-PID* architecture. The first subsection outlines the theoretical foundation and architectural framework of *CTC*, which leverages the exact inverse dynamics of the PUMA 560 manipulator to achieve linearization and decoupling of the highly non-linear robotic system. The second subsection presents the simulation results and discussion, evaluating *CTC's* tracking performance under the same test scenarios, nominal conditions, high-dynamic helical trajectories, and unmodeled payload variations, to establish a direct comparative baseline against the intelligent control approach.

IV.3.1 THEORY AND ARCHITECTURE

While Fuzzy Logic Control (*FLC*) provides robust model-free adaptability, achieving absolute, deterministic tracking precision in high-speed robotic applications often requires explicitly neutralizing the manipulator's non-linearities. The Computed Torque Control (*CTC*) technique, fundamentally grounded in the theory of exact feedback linearization, represents one of the most powerful non-linear, model-based control strategies in modern robotics [33], [34]. Unlike linear *PID* controllers that attempt to overpower dynamic cross-couplings through high error-driven gains, *CTC* actively utilizes the rigid-body inverse dynamic model of the robot arm (PUMA 560 in this dissertation) to mathematically decouple the interconnected mechanical channels and cancel out the non-linear *Coriolis*, centrifugal, and gravitational forces in real-time [35].

Let us recall the standard dynamic equation of an *n*-degree-of-freedom rigid manipulator is given by:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau \quad (4.4)$$

Where $M(q)$ is the inertia matrix, $C(q, \dot{q})$ represents the *Coriolis* and *centrifugal* matrix, $G(q)$ is the gravity vector, $F(\dot{q})$ defines joint friction, and τ is the applied torque vector.

The *CTC* architecture achieves exact feedback linearization by formulating a control law composed of two distinct parts: a non-linear inner model-based cancellation loop, and a linear outer error-regulation loop [36]. The total commanded torque τ_{CTC} is structured as follows:

$$\tau_{CTC} = \hat{M}(q)\,u + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q}) \quad (4.5)$$

Here, the terms denoted with a circumflex ($\hat{M}$, $\hat{C}$, $\hat{G}$, $\hat{F}$) represent the controller's internal computational estimates of the physical plant parameters. The variable u represents the auxiliary linear control input, designed to drive the tracking error to zero. For a standard trajectory tracking problem, u is typically structured as a Proportional-Derivative (*PD*) feedback loop acting on the desired joint accelerations:

$$u = \ddot{q}d + K_d\,\dot{e} + K_p\,e \quad (4.6)$$

Where $e(t) = q_d(t) - q(t)$ is the joint position tracking error, and K_p, K_d are positive-definite diagonal gain matrices.

By substituting the *CTC* control law into the fundamental dynamic equation and assuming perfect parametric modeling where the estimated matrices perfectly match the physical plant ($\hat{M} = M$, $\hat{C} = C$, $\hat{G} = G$, $\hat{F} = F$), the highly complex, non-linear multi-body system elegantly reduces to a set of *n* decoupled, linear double integrators:

$$\ddot{e} + K_d\,\dot{e} + K_p\,e = 0 \quad (4.7)$$

This resulting closed-loop error dynamics equation represents a linear, second-order ordinary differential equation for each independent joint.

By selecting appropriate positive gains for K_p and K_d (often chosen to achieve critical damping), asymptotic convergence of the tracking error to zero ($e(t) \rightarrow 0$ as $t \rightarrow \infty$) is mathematically guaranteed, regardless of the trajectory's operational speed or spatial complexity [37], [38].

While *CTC* provides unparalleled theoretical precision, its fundamental architecture relies absolutely on the exactness of the mathematical model [39]. In practical industrial environments, parametric uncertainties, such as unmodeled structural friction or, critically, the introduction of unknown end-effector payloads (as evaluated in Test III), create a mismatch between the controller's estimates ($\hat{M}$, $\hat{G}$) and the actual plant dynamics (M, G). Under such stress, the exact feedback linearization fails, transforming the mathematically guaranteed zero-error dynamics into a perturbed system susceptible to steady-state Cartesian sag and potential instability [40].

IV.3.2 Simulation Results and Discussion

The performance of the pure *Computed Torque Control* (*CTC*) scheme was evaluated through three simulation tests, conducted in the same manner as those applied to the *FLC-PID* controller. Each test focuses on a different operational condition. The first test examines tracking accuracy under nominal, low-speed motion to validate the correct implementation of the inverse dynamics model. The second test imposes a high-frequency helical trajectory to assess the controller's response under aggressive, coupled motion. The third test introduces an unmodeled payload variation to reveal the sensitivity of model-based control to parametric uncertainties. The following subsections present and discuss the results obtained from each test.

IV.3.2.1 Test I 'Tracking Performance Under Baseline Nominal Conditions'

To establish a definitive performance baseline, the Computed Torque Control (*CTC*) architecture was first evaluated under nominal operating conditions. In this scenario, the PUMA 560 executed the standard 5th-order quintic polynomial trajectory without any unmodeled external payloads (Δm = 0 kg). Because the controller's internal computational model perfectly matches the physical plant parameters in this specific test ($\hat{M}=M$, $\hat{C}=C$, $\hat{G}=G$), the expected outcome is exact feedback linearization, mathematically guaranteeing asymptotic convergence of the tracking error.

Figure 4.19 visually confirms the absolute determinism of the *CTC* architecture under ideal parametric conditions. The actual joint positions (dashed red lines) are perfectly superimposed over the desired reference trajectories (solid blue lines) across all six degrees of freedom.

Unlike the classical *PID* controller, which exhibited minor initial transient lags while fighting structural inertia, the *CTC* explicitly computes and injects the exact acceleration required via its inverse dynamic model. Consequently, the mechanical links follow the polynomial path with mathematically perfect geometric adherence from the very first computational step.

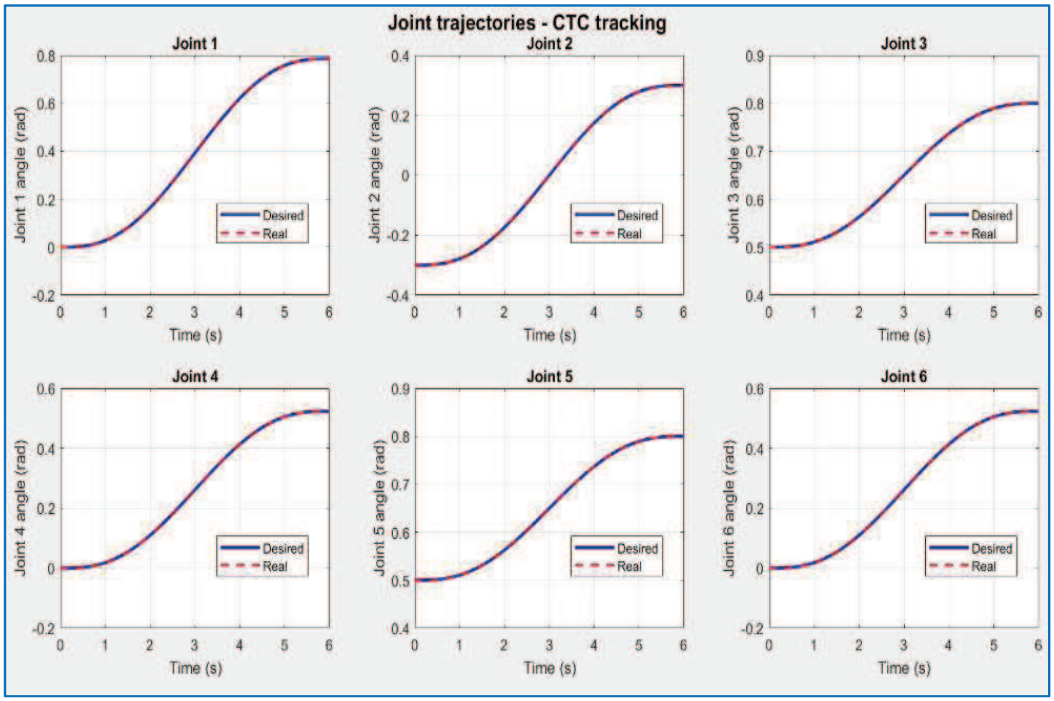


Figure 4.19 *Joint trajectories – CTC / Test I*

Elsewhere, the joint position tracking error profiles, $e_i(t)$, presented in figure 4.20 serve as a physical proof of successful exact feedback linearization. The error curves are virtually indistinguishable from the absolute zero axis, maintaining flatlines well inside the strict (**±0.02 rad**) tolerance limits.

Moreover, the statistical extraction validates this visual perfection, showing a maximum joint *RMS* error of merely **2×10^{-4} rad** (**0.01 degrees**) in the remote joints. By canceling the non-linear terms through inverse dynamics, the outer Proportional-Derivative (*PD*) loop with (K_p = 400, K_v = 40) operates on a decoupled, idealized double-integrator system, allowing it to drive transient deviations to zero almost instantly.

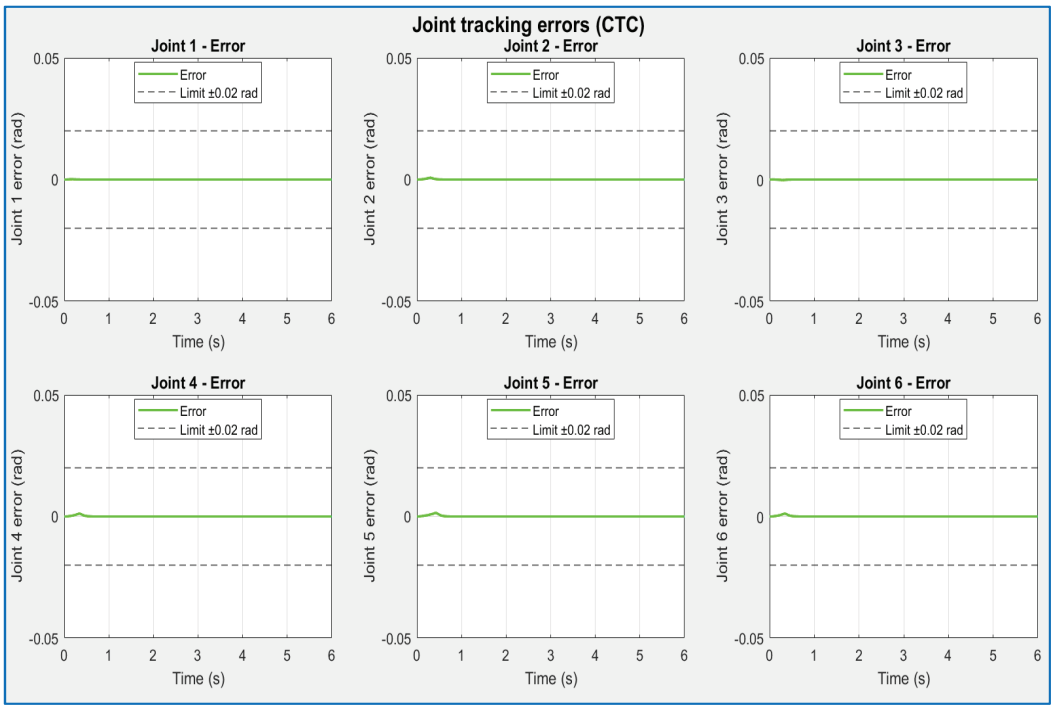


Figure 4.20 *Tracking error evolution – CTC / Test I*

An analysis of the actuator control torques in figure 4.21 highlights how the *CTC* optimizes energy consumption. While the macroscopic torque curves mirror the baseline gravitational requirements of the manipulator (with Joint 2 peaking smoothly near (**74 Nm**)), the critical feature is the absence of high-frequency control chattering. The only visible transients are microscopic spikes occurring strictly at t= 0s.

These localized initial spikes represent the exact, instantaneous calculated torque ($\widehat{F}_c \text{sgn}(\dot{q})$) required to overcome the static *Coulomb* friction barrier. After the starting is achieved, the model-based feedforward cancellation governs the motion fluidly, protecting the servo motors from destructive oscillations while delivering unparalleled kinematic accuracy.

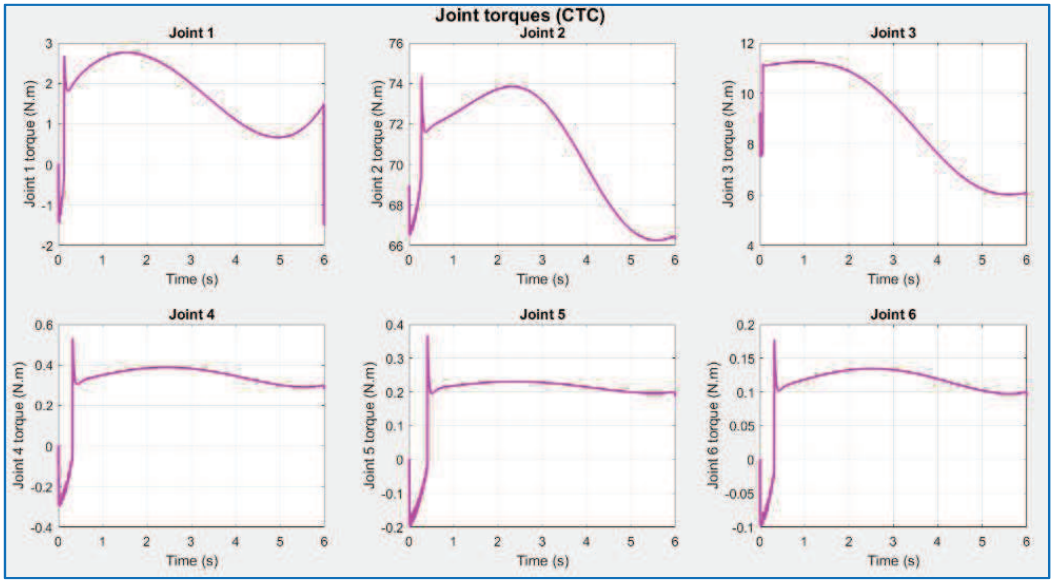


Figure 4.21 *Joint torques – CTC / Test I*

Figure 4.22 maps the spatial execution of the end-effector within the *3D* Cartesian workspace. The macroscopic visualization of the path corroborates the localized joint-level perfection. The actual tool center point trajectory completely eclipses the programmed reference curve from the green departure coordinate to the red destination sphere.

This geometric flawless execution proves that the multi-variable cross-coupling, which typically distorts spatial paths in classical decentralized control, has been entirely neutralized mathematically before the mechanical motion even occurs.

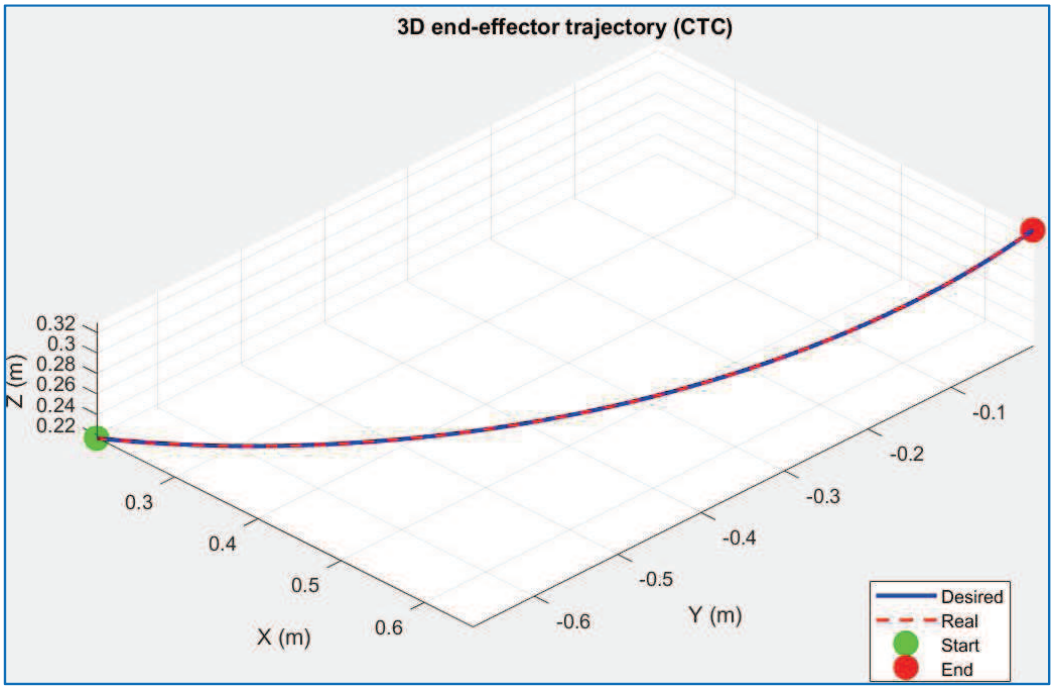


Figure 4.22 *3D end-effector trajectory – CTC / Test I*

The absolute Euclidean Cartesian error profile in figure 4.23 represents the ultimate achievement of the deterministic model-based strategy over classical model-free feedback.

The transient error exhibits a negligible initial peak of exactly **(0.6 mm)** at (***t* ≈ 0.3 s**) during maximum initial acceleration. Most impressively, the curve subsequently drops and asymptotically locks onto absolute zero, yielding an overall Mean Cartesian Error of exactly '**zero**'. When compared to the nominal mean errors of the highly tuned *PID* (**8.5 mm**) and the intelligent *FLC* (**4.6 mm**), the *CTC* achieves a perfect level of absolute spatial convergence. This establishes the Computed Torque Controller as the definitive standard for trajectory tracking provided that the dynamic model of the rigid-body system remains perfectly known and undisturbed.

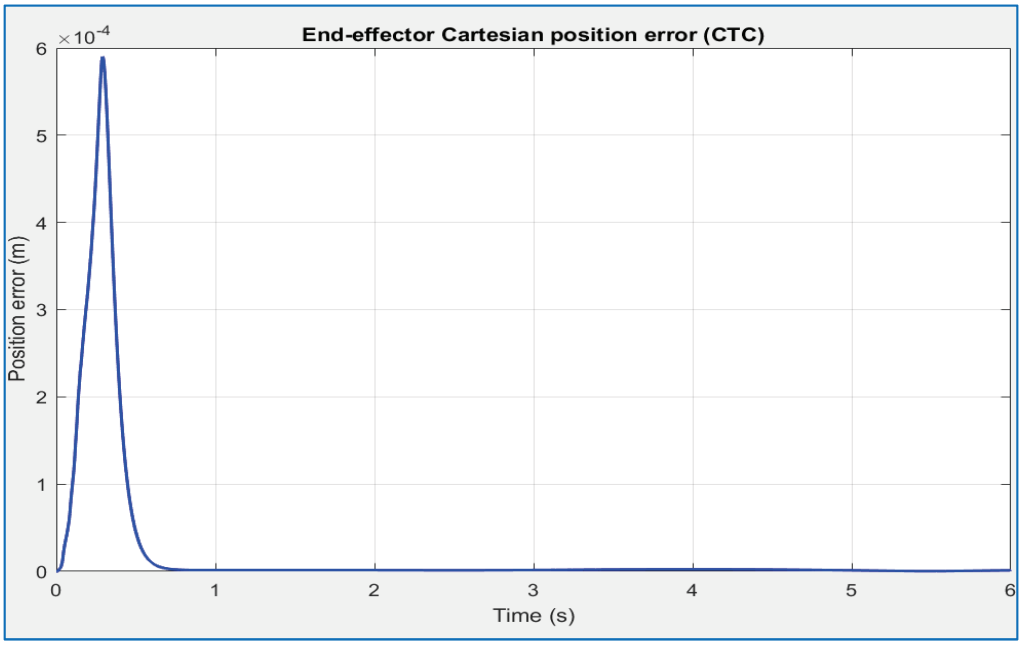


Figure 4.23 *Euclidean position error – CTC / Test I*

IV.3.2.2 TEST II 'HIGH-DYNAMIC HELICAL TRAJECTORY'

To evaluate the exact feedback linearization capabilities of the *CTC* architecture under severe dynamic stress, the system was commanded to execute the high-speed 3-loop helical trajectory. This spatial maneuver forces the manipulator into continuous velocity reversals, generating massive *Coriolis* and centrifugal cross-coupling forces.

Unlike classical *PID* or model-free fuzzy logic, which attempt to suppress these disturbances reactively through feedback loops, the *CTC* algorithm explicitly calculates and cancels these non-linearities predictively utilizing its internal dynamic model.

Figure 4.24 demonstrates absolute kinematic adherence. The highly dynamic sinusoidal and cosinusoidal reference signals of the regional base and shoulder links are tracked perfectly by the actual joint positions (dashed red lines).

Despite the intense multi-body dynamic coupling induced by the continuous spatial loops, the deterministic cancellation of the non-linear inertia and *Coriolis* matrices effectively isolates each joint. This mathematical decoupling allows the localized linear Proportional-Derivative (*PD*) controllers to govern the system precisely, resulting in perfect trajectory superimposition without any visible transient lag.

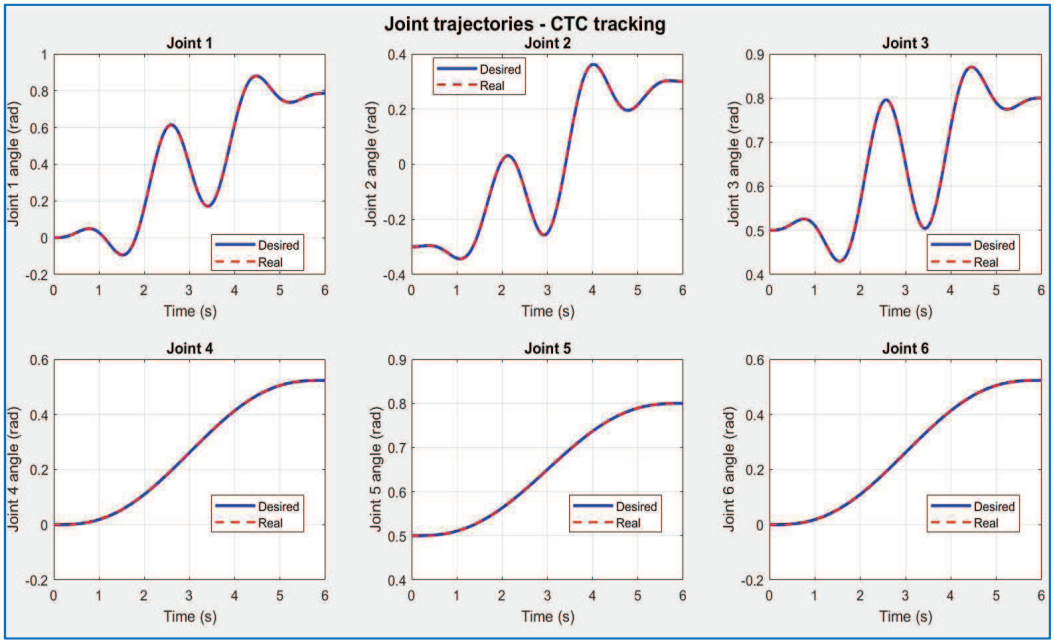


Figure 4.24 *Joint trajectories – CTC / Test II*

A critical comparative examination of the tracking error profiles, e_i (t), in figure 4.25 exposes the unparalleled superiority of model-based control under ideal parametric conditions. In previous evaluations, both the classical *PID* and the *Mamdani FLC* exhibited pronounced sinusoidal error ripples driven by the uncompensated helical frequencies.

Conversely, the *CTC* completely flattens these dynamic disturbances. The error curves remain virtually indistinguishable from the absolute zero axis, recording a maximum joint *RMS* error of merely (**2.36×10^{-4} rad**) in Joint 5. This flatline response proves that the inverse dynamics model successfully neutralized the severe *Coriolis* and centrifugal forces before they could manifest as physical tracking deviations.

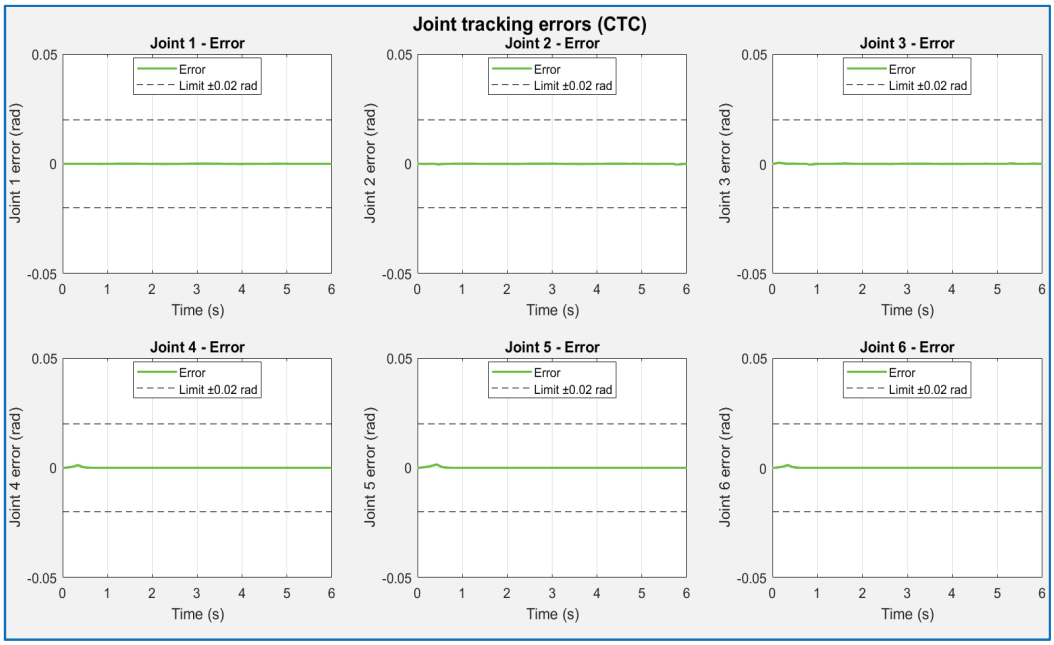


Figure 4.25 *Tracking error evolution – CTC / Test II*

Figure 4.26 provides deep insight into the energetic mechanisms of the exact feedback linearization. To maintain perfect spatial fidelity during the high-speed loops, the *CTC* commands rapid, high-amplitude torque reversals, particularly in the structural base links. Joint 1 continuously sweeps between (**-30 N.m** and **+30 N.m**), while Joint 2 peaks near (**88 N.m**).

However, unlike the dangerous, high-frequency mechanical chattering generated by the aggressively tuned *PID* controller under identical conditions, these *CTC* torque profiles are structurally deterministic, smooth, and fluid.

The control effort represents the exact mathematical physical requirement to drive the inertia tensor through space, guaranteeing dynamic precision without inducing destructive vibratory fatigue in the servo gearboxes.

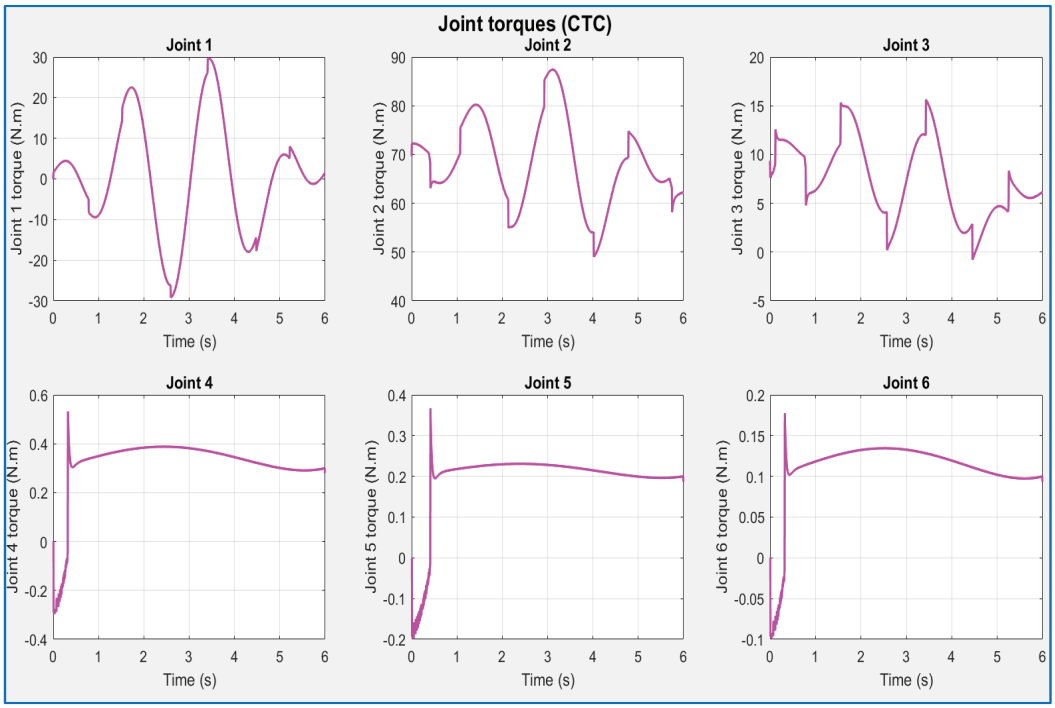


Figure 4.26 *Joint torques – CTC / Test II*

Elsewhere, the macroscopic spatial visualization in figure 4.27 confirms the absolute success of the multi-variable decoupling strategy. The actual tool center point path easily navigates the 3-loop spiral, completely eclipsing the programmed blue reference curve.

In contrast to the classical independent joint architecture, which suffered from distinct spatial separations at the extreme apogees of the loops due to centripetal acceleration, the tool center point path exhibits no geometric distortion. The spatial execution is structurally coherent and perfectly synchronized across all *3D* axes.

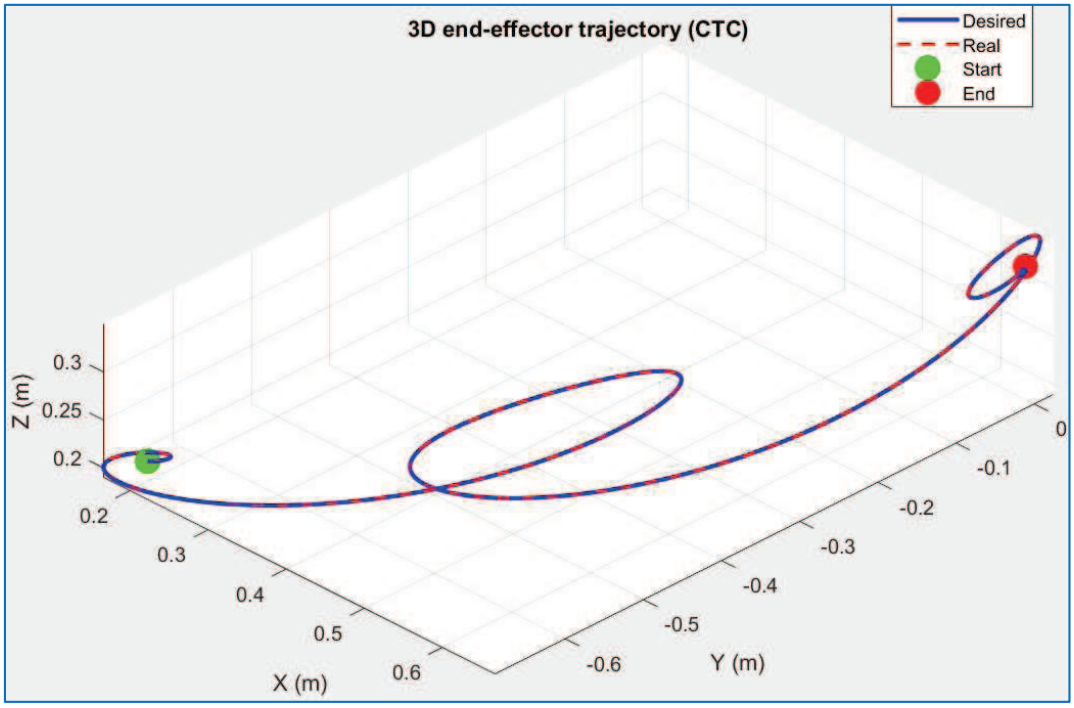


Figure 4.27 *3D end-effector trajectory – CTC / Test II*

The absolute *Euclidean Cartesian* error curve presented in figure 4.28 represents a definitive engineering triumph for the Computed Torque Control technique, provided the dynamic model is perfectly known. During a maneuver that forced the highly tuned *PID* controller to deviate by nearly (**14.5 mm**), and the intelligent *FLC* by (**16.2 mm**), the *CTC* restricts the maximum transient peak error to a microscopic value of **(0.268 mm)**. Furthermore, the overall mean Cartesian error across the entire 6-second complex trajectory is a tiny value of **(0.051 mm)**. By leveraging the exact inverse dynamics of the PUMA 560, the *CTC* effectively transforms a highly complex, non-linear mechanical challenge into a trivial linear regulation problem, resulting in an order-of-magnitude decrease in absolute positional deviation.

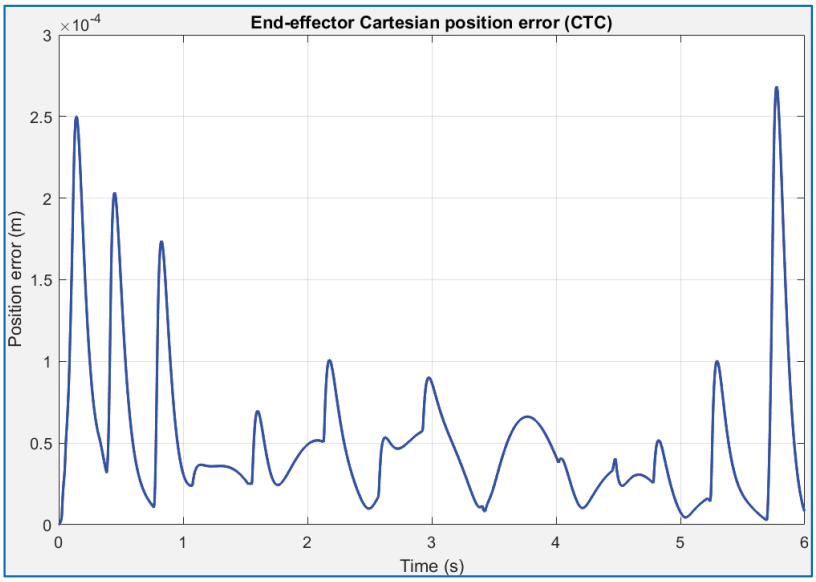


Figure 4.28 *Euclidean position error – CTC / Test II*

IV.3.2.3 TEST III 'ROBUSTNESS UNDER UNMODELED PAYLOAD DURING HIGH-DYNAMIC SPATIAL TRACKING'

The final evaluation of the *CTC* architecture exposes its fundamental theoretical vulnerability: absolute dependence on precise parametric modeling. In this rigorous test, the highly dynamic 3-loop helical trajectory is executed while an unmodeled 5 kg payload is rigidly attached to the end-effector. Critically, the controller's internal inverse dynamics algorithm ($\widehat{M}$, $\hat{C}$, $\hat{G}$,) remains completely blind to this mass variation, operating entirely on the nominal empty-load parameters. This deliberate mismatch destroys the exact feedback linearization assumption ($\widehat{M} \neq M, \hat{G} \neq G$), forcing the outer linear Proportional-Derivative (*PD*) loop to single-handedly fight the massive uncompensated inertial and gravitational disturbances.

Figure 4.29 displays the kinematic response of the parametrically mismatched *CTC* system. While the high proportional gains (K_p=400) manage to keep the actual joint trajectories (dashed red lines) in the general vicinity of the desired high-frequency reference waves (solid blue lines) preventing catastrophic divergence, the pristine mathematical superimposition observed in 'Test II' has visibly deteriorated. The localized separation, particularly in the base and shoulder joints, indicates that the feedforward cancellation has failed, and the system has reverted to behaving like a standard, heavily perturbed linear feedback loop.

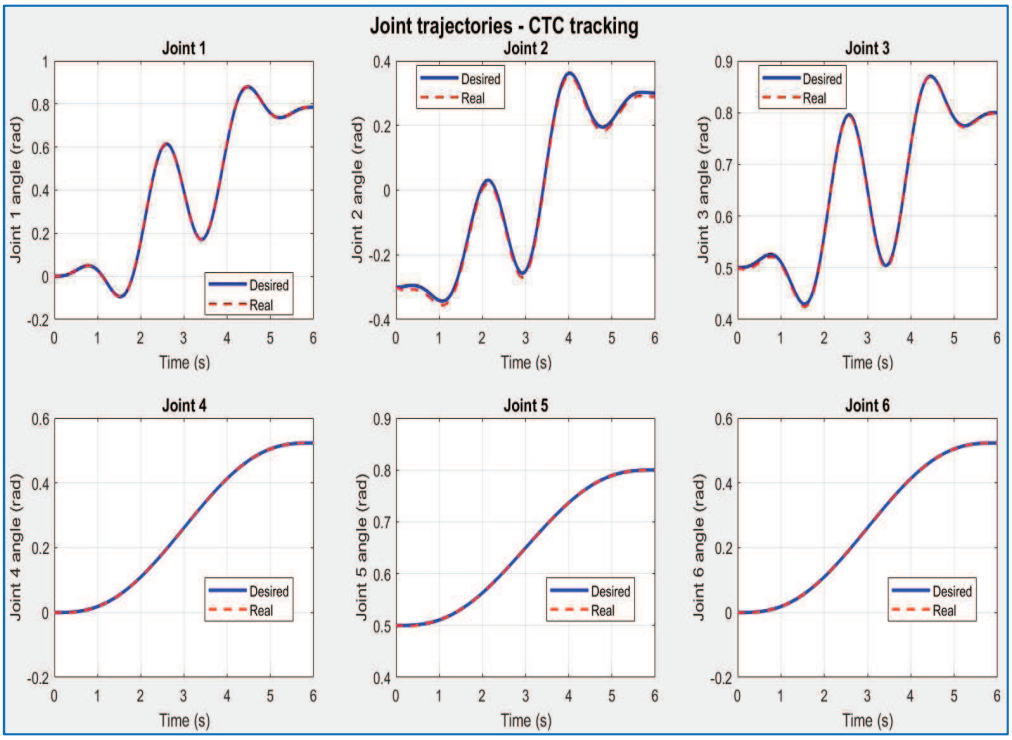


Figure 4.29 *Joint trajectories – CTC / Test III*

Figure 4.30 presents the definitive proof of the *CTC*'s architectural drawback. In 'Test II', the tracking errors were absolutely zero. Under the 5 kg payload, the error for Joint 2 exhibits a severe, permanent positive steady-state shift, with an *RMS* error of (**0.0120 rad**). This steady-state 'sag' is a direct mathematical consequence of the controller's structure. Because the computed gravity vector $\hat{G}$ drastically underestimates the actual gravitational pull *G*, consequently there is a constant torque deficit.

Since the standard *CTC* utilizes an outer *PD* loop lacking integral action (K_i=0), the only way the controller can generate the missing holding torque is by maintaining a permanent, non-zero position error $e(t)$ to be multiplied by the proportional gain K_p. This explicitly proves that standard *CTC* cannot achieve zero steady-state error under persistent parametric uncertainty.

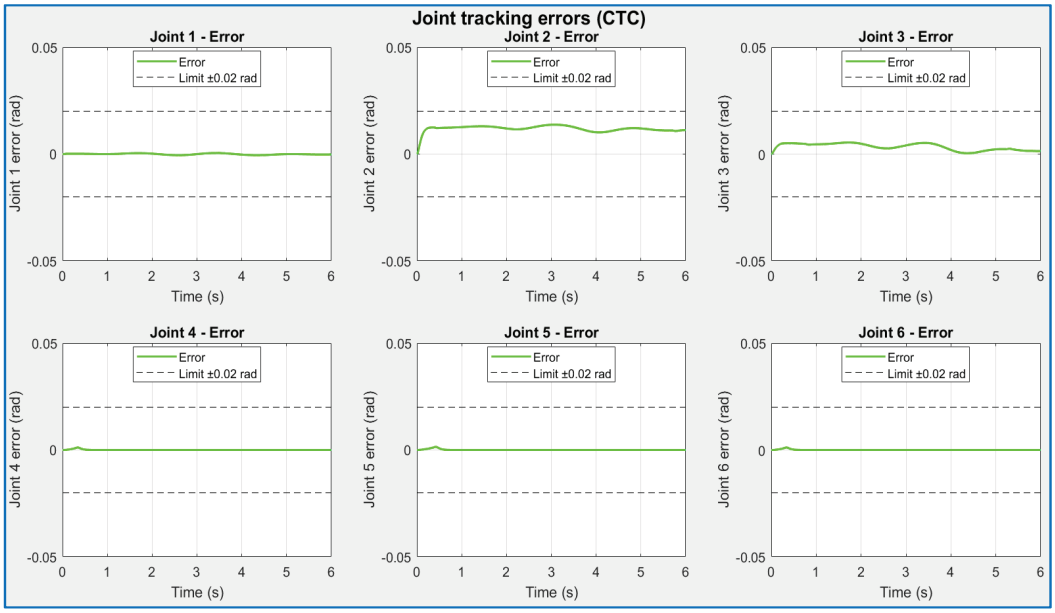


Figure 4.30 *Tracking error evolution – CTC / Test III*

The actuator effort visualized in figure 4.31 highlights the intense strain shifted onto the feedback loop due to the failure of the feedforward model. With the inverse dynamics underestimating the structural load, the *PD* loop severely compensates, driving the torque command for Joint 2 to peak dangerously near (**120 N.m**) during the upward helical loops.

While the *CTC* retains its characteristic signal smoothness, avoiding the high-frequency chattering seen in the purely classical *PID*, it still suffers from extreme energy demands and near-saturation operation, severely limiting the manipulator's operational safety margins.

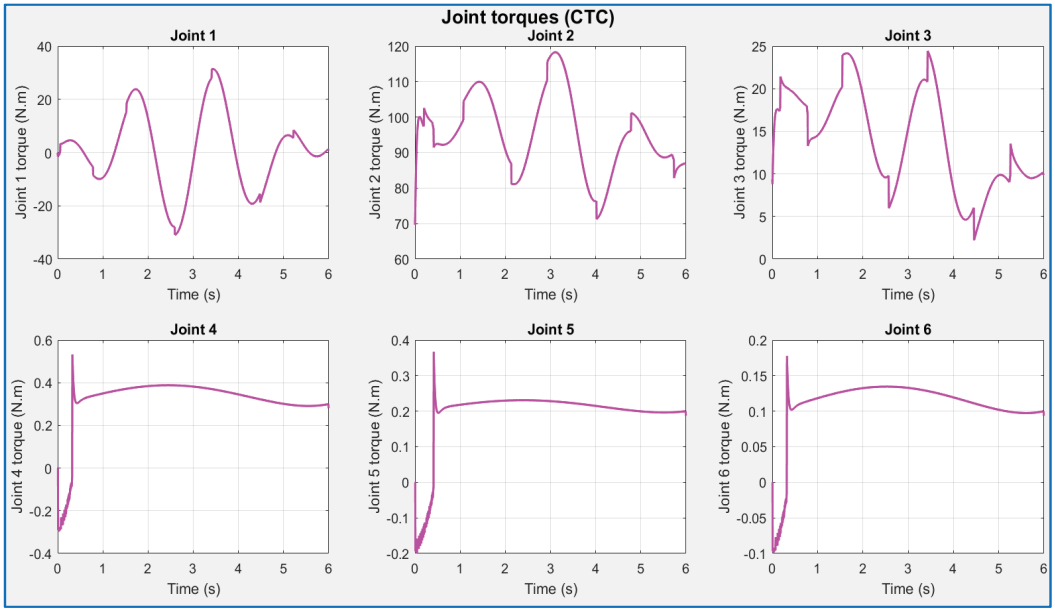


Figure 4.31 *Joint torques – CTC / Test III*

The macroscopic spatial visualization in figure 4.32 confirms the geometric consequences of the uncompensated mass. The heavy payload forcefully pulls the tool center point downward, causing the actual execution path (dashed red line) to shift below the desired blue reference curve, particularly at the lowest Z-axis coordinates of the spatial loops. The exact feedback linearization has completely degraded, re-introducing the multi-variable cross-coupling and Cartesian distortion that the *CTC* was explicitly designed to eliminate

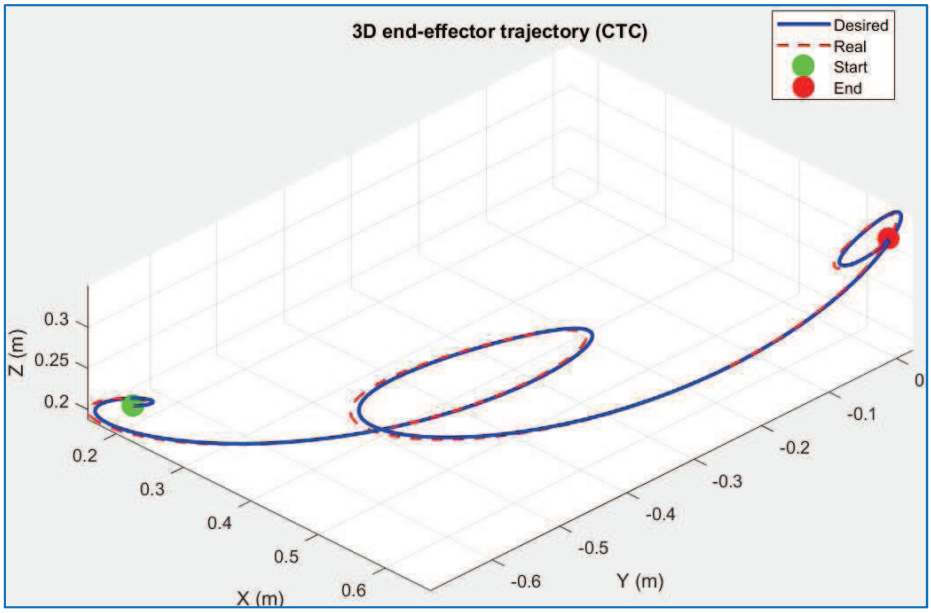


Figure 4.32 *3D end-effector trajectory – CTC / Test III*

The Euclidean Cartesian error profile in figure 4.33 quantitatively shows the *CTC* technique limitations. The perfect zero-error tracking achieved in the nominal state has entirely vanished. The Cartesian deviation rapidly spikes and continuously oscillates, reaching a maximum transient peak of (**9.0 mm**) and yielding a mean operational error of (**7.5 mm**). While a value of (**9.0 mm**) represents a tighter absolute peak tolerance compared to the severely oscillations given by the classical *PID* (**16.8 mm**) and the intelligent *FLC* (**18.0 mm**). In this case, the *CTC* fundamentally fails its primary objective: mathematical precision.

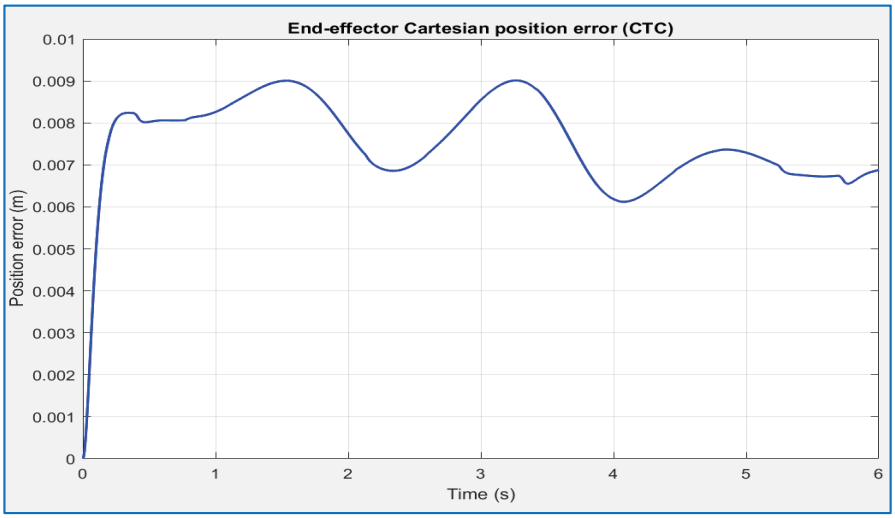


Figure 4.33 *Euclidean position error – CTC / Test III*

IV.4 HYBRID COMPUTED TORQUE - FUZZY LOGIC CONTROL

This section presents the hybrid *Computed Torque - Fuzzy Logic Controller* (*CTC-FLC*) as a synergistic solution that combines the strengths of both previous approaches. The first subsection describes the theoretical foundation and architectural design of the hybrid scheme, where the *CTC* component handles nominal dynamics cancellation while the *FLC* compensates for residual uncertainties and unmodeled disturbances. The second subsection presents the simulation results and discussion, evaluating the hybrid controller under the same three test conditions applied to the pure *FLC-PID* and pure *CTC* controllers. The following subsections detail the architecture and performance of the proposed hybrid scheme.

IV.4.1 THEORY AND ARCHITECTURE

The rigorous benchmarking conducted in the preceding sections established a fundamental engineering dichotomy: Computed Torque Control (*CTC*) delivers a perfect tracking when the rigid-body dynamics are exactly known, but it structurally fails under robustness and adaptability against unmodeled payloads and it lacks the absolute mathematical determinism required to achieve zero steady-state error during high-speed, complex multi-variable maneuvers [41].

To overcome the individual limitations of both control techniques, this section formulates a synergistic architecture: the Hybrid Computed Torque - Fuzzy Logic Controller (*CTC-FLC*). The philosophy of this hybrid approach is to strictly divide the computational cost. The inverse dynamics model is utilized to explicitly cancel all known non-linearities (nominal feedforward linearization), while an intelligent fuzzy inference system acts in parallel as a robust supervisory compensator to dynamically neutralize all unknown uncertainties and external disturbances (heuristic feedback compensation) [42],[43].

Mathematically, the true dynamic equation of the perturbed PUMA 560 manipulator under a payload variation can be expressed by separating the nominal (known) model parameters ($\widehat{M}$, $\hat{C}$, $\hat{G}$,) from the parametric uncertainties (ΔM, ΔC, ΔG) and unmodeled friction (F):

$$(\widehat{M}(q) + \Delta \mathrm{M}(q))\ddot{q} + (\hat{C}(q,\dot{q}) + \Delta \mathrm{C}(q,\dot{q}))\dot{q} + (\hat{G}(q) + \Delta \mathrm{G}(q)) + F(\dot{q}) = \tau \quad (4.8)$$

By grouping all modeling errors, external payloads, and friction terms into a single lumped disturbance vector $d(t)$, the actual plant dynamics simplify to:

$$\widehat{M}(q)\ddot{q} + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) + d(t) = \tau \quad (4.9)$$

Where $d(t) = \Delta M(q))\ddot{q} + \Delta C(q,\dot{q})\dot{q} + \Delta G + F(\dot{q})$. In the previous 'Test III', the classical *CTC* failed because it assumed $d(t) = 0$.

To aggressively force the trajectory tracking error to zero despite the presence of $d(t)$, the Hybrid *CTC-FLC* control law is defined as the superposition of two distinct torque commands:

$$\tau = \tau_{CTC} + \tau_{FLC} \quad (4.10)$$

a*) *The Nominal Model-Based Component **(τ_{CTC}):** This component executes the exact feedback linearization based only on the baseline mathematical model of the robot:

$$\tau_{CTC} = \widehat{M}(q)(\ddot{q}d + K_d\,\dot{e} + K_p\,e) + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) \quad (4.11)$$

When this torque is injected into the physical plant, the closed-loop error dynamics become:

$$\ddot{e} + K_d\,\dot{e} + K_p\,e = \widehat{M}^{-1}(q)\,(d - \tau_{FLC}) \quad (4.12)$$

b) The Intelligent Fuzzy Compensator **(τ_{FLC}):** To guarantee asymptotic stability and restore perfect tracking, the right side of the error equation must be forced to zero.
Therefore, the absolute requirement is that the auxiliary control signal must dynamically approximate and cancel the lumped disturbance: $\tau_{FLC} \approx d(t)$.

Because $d(t)$ is highly non-linear, unpredictable, and analytically inaccessible in real-time, classical linear integrators cannot effectively track it without inducing severe windup [44]. Instead, a *Mamdani* fuzzy inference system is deployed as a universal non-linear approximator. Utilizing the position error e_i(t) and velocity error $\dot{e}_i(t)$ as inputs, the *FLC* continuously navigates its heuristic rule base to generate the precise corrective torque: τ_{FLC} required to neutralize the unmodeled 5 kg payload and dynamic friction.

The structural smart of this hybrid architecture lies in the fact that the *FLC* is no longer tasked with driving the massive inertia of the entire robot, that task is handled deterministically by the *CTC*. The fuzzy logic is strictly reserved for managing the localized residual error $d(t)$ [45]. This reduces significantly the load on the rule base, allowing the fuzzy membership functions to be tuned much more aggressively around the zero-error axis without risking the high-frequency control chattering, the inherent limitation that conventional feedback loops suffer from.

IV.4.2 Simulation Results and Discussion

The performance of the proposed hybrid *Computed Torque - Fuzzy Logic Controller* (*CTC-FLC*) was evaluated through three simulation tests, conducted in the same manner as those applied to the pure *FLC-PID* and pure *CTC* controllers. Each test targets a different operational condition. The first test examines tracking accuracy under nominal, low-speed motion to verify the hybrid scheme's baseline precision. The second test imposes a high-frequency helical trajectory to assess the controller's ability to handle aggressive, coupled motion. The third test introduces an unmodeled payload variation to demonstrate the hybrid architecture's robustness against parametric uncertainties. The following subsections present and discuss the results obtained from each test.

IV. 4.2.1 Test I 'Tracking Performance Under Baseline Nominal Conditions'

Before subjecting the newly synthesized Hybrid Computed Torque - Fuzzy Logic Controller (*CTC-FLC*) to extreme parametric stress, it must first be validated under baseline nominal conditions. In this scenario, the PUMA 560 executes the smooth quintic polynomial trajectory with no unmodeled payload (Δm = 0 kg). The primary engineering objective of this test is to verify that the injection of the fuzzy auxiliary control signal (τ_{FLC}) does not destabilize the exact feedback linearization achieved by the nominal *CTC* model. Under ideal conditions where $d(t) \approx 0$, the fuzzy compensator should remain largely disabled, allowing the deterministic inverse dynamics to govern the motion smoothly.

Figure 4.34 visually confirms the baseline kinematic stability of the hybrid architecture. The actual joint positions (dashed red lines) execute a perfect superimposition over the desired quintic reference profiles (solid blue lines). The integration of the *Mamdani* fuzzy inference system into the acceleration control loop operates in perfect harmony with the dynamic feedforward model. The system transitions smoothly from the initial departure to the final target coordinates without any induced transient lag, proving that the aggressive fuzzy scaling factors (such as $Ke_1 = 800$ for Joint 1 acceleration) do not inject unnecessary operational stiffness when the system is unperturbed.

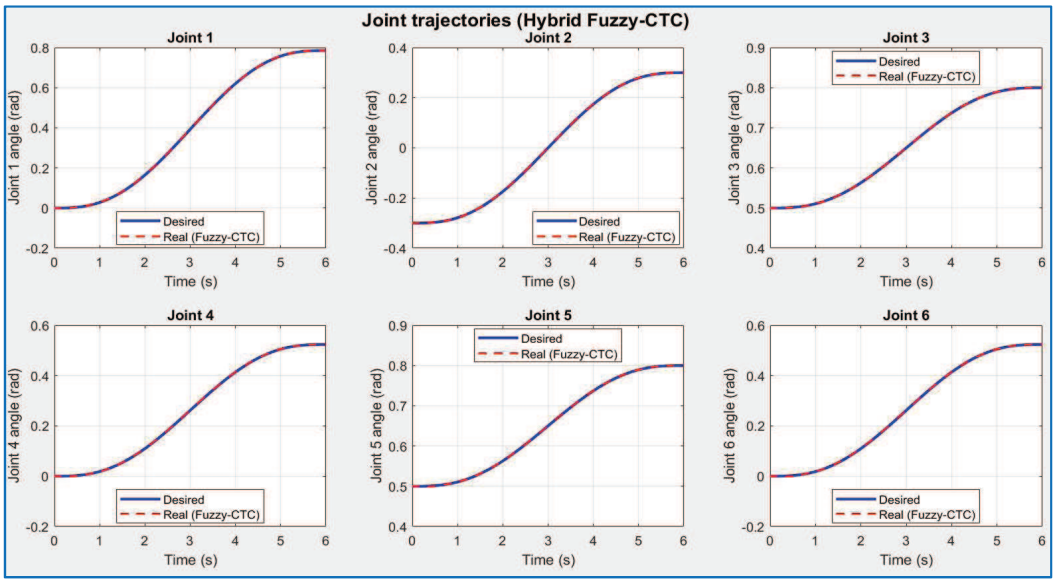


Figure 4.34 *Joint trajectories – CTC-FLC / Test I*

The chronological tracking error profiles $e_i(t)$, presented in figure 4.35 underscore the extreme precision of the hybrid approach. The error curves essentially flatline along the absolute zero axis, maintaining absolute compliance well within the strict (**±0.02 rad**) tolerance thresholds. The statistical extraction corroborates this, with maximum transient deviations restricted to just a tiny amount of **1.2×10^{-3} rad** (**0.07 degrees**) in Joint 1. This verifies that the fuzzy rule base, designed to act as a secondary stabilizing mechanism, successfully processes the microscopic residual errors.

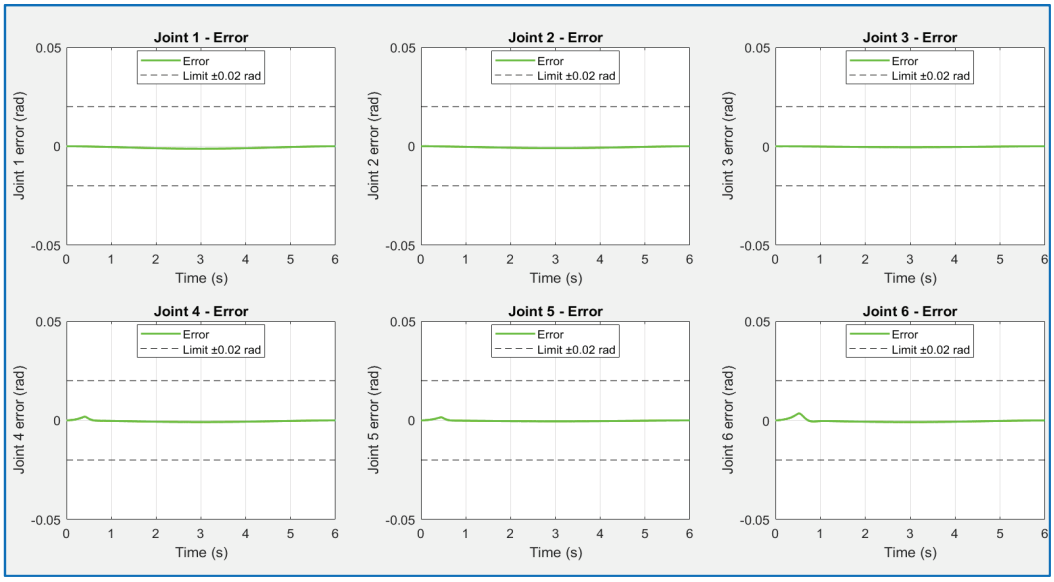


Figure 4.35 *Tracking error evolution – CTC-FLC / Test I*

Figure 4.36 provides a critical insight into the energetic synergy between the model-based and model-free components. The macroscopic torque profiles remain conveniently continuous and fluid, mirroring the baseline gravitational and inertial requirements of the mechanical plant (with Joint 2 peaking nominally near **74 N.m**). The absence of high-frequency chattering is a real success; it proves that the center of gravity defuzzification method smoothly translates the fuzzified kinematic errors into a continuous auxiliary acceleration command (τ_{FLC}). This protects the servo drives from vibratory degradation while preserving the exact dynamic cancellation properties of the basic *CTC*.

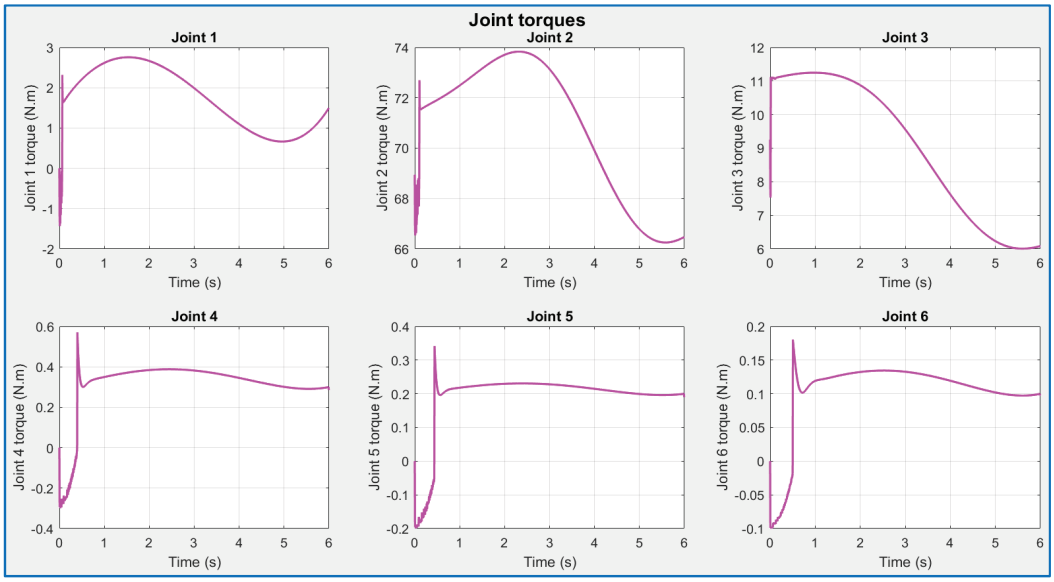


Figure 4.36 *Joint torques – CTC-FLC / Test I*

Elsewhere, the spatial execution of the end-effector mapped in figure 4.37 visually shows the success of the multi-variable tracking. The actual tool center point path perfectly eclipses the programmed reference curve within the *3D* Cartesian workspace. Because the *CTC* element successfully eliminates the non-linear cross-coupling and the *FLC* element dynamically tightens the residual feedback regulation, the end-effector maintains absolute spatial integrity from the green start coordinate to the red destination sphere.

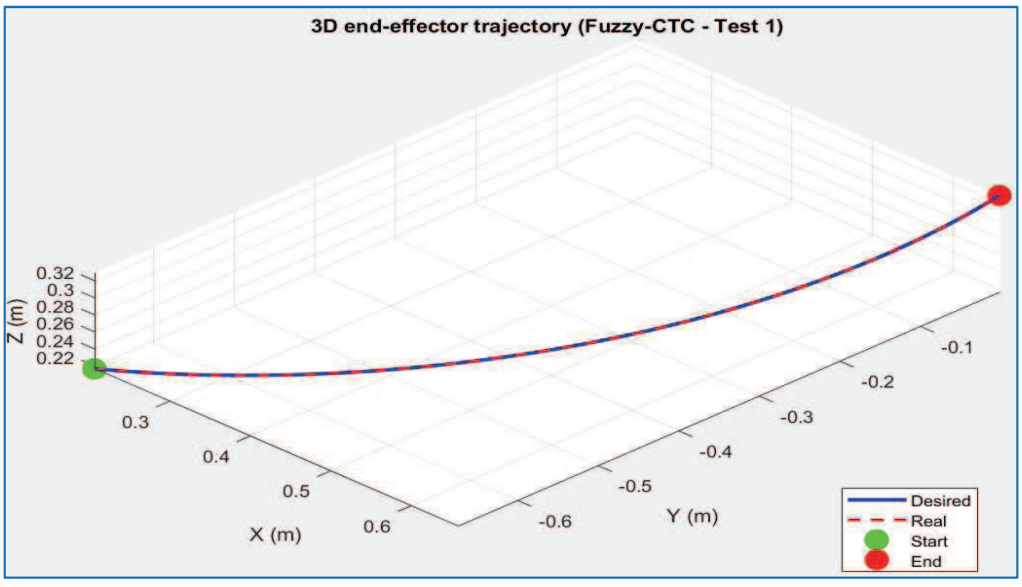


Figure 4.37 *3D end-effector trajectory – CTC-FLC / Test I*

The ultimate quantitative verification of the hybrid system's baseline perfection is captured in the absolute *Euclidean Cartesian* error curve (see figure 4.38). The maximum transient spatial deviation is restricted to a microscopic peak of (**0.1 mm**) during the initial acceleration phase, after which the curve rapidly drops and asymptotically adheres to zero. The resulting mean *Cartesian* error across the entire trajectory is almost zero (**0.0 mm**). This result fundamentally proves that the hybrid *CTC-FLC* architecture retains 100% of the absolute mathematical precision of a pure *CTC* system under ideal conditions, establishing a perfect foundation before introducing the severe uncertainties of the final evaluations.

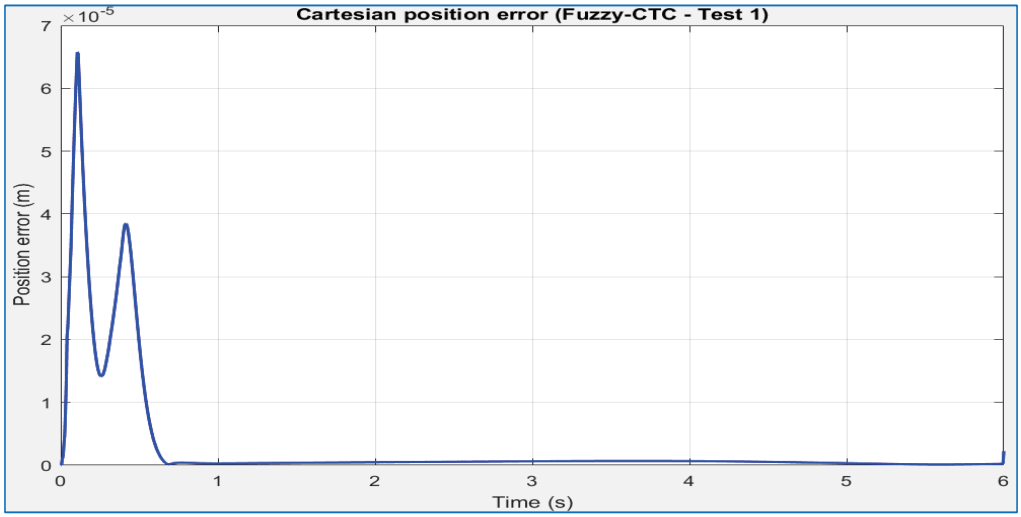


Figure 4.38 *Euclidean position error – CTC-FLC / Test I*

IV.4.2.2 TEST II 'HIGH-DYNAMIC HELICAL TRAJECTORY'

To evaluate the dynamic resilience of the hybrid architecture, the system was subjected to the highly aggressive 3-loop helical trajectory. This maneuver forces the spatial links into continuous, high-frequency velocity reversals, generating massive *Coriolis* and centrifugal cross-coupling forces. In this nominal state (without an external unmodeled payload), the exact inverse dynamics of the *CTC* component are expected to perfectly decouple the system, while the *FLC* operates in a supervisory capacity, ensuring that its non-linear inference mechanism does not inadvertently inject disruptive chattering into the perfectly linearized acceleration loop.

The kinematic response visualized in figure 4.39 demonstrates absolute multi-variable synchronization. The highly oscillatory sinusoidal and cosinusoidal desired reference signals for the base and shoulder links are tracked perfectly by the physical joints.

The dashed red execution curves practically eclipse the solid blue reference paths. This visual perfection confirms that the model-based feedforward component ($\widehat{M}$, $\hat{C}$, $\hat{G}$,) successfully neutralizes the massive inertial dynamics, allowing the fuzzy auxiliary input (τ_{FLC}) to process only microscopic residual errors without disrupting the macroscopic kinematic flow.

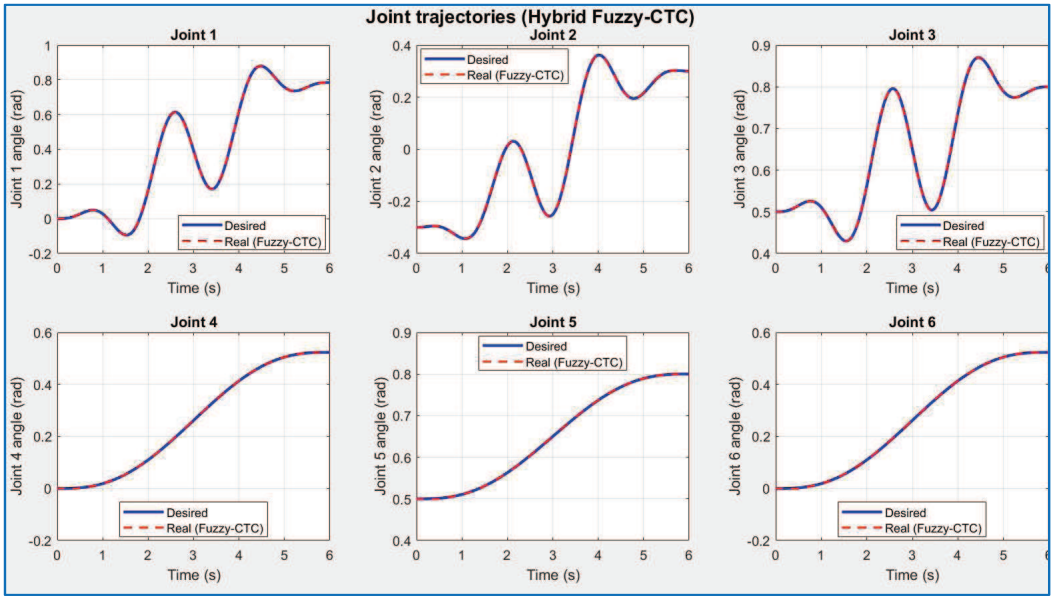


Figure 4.39 *Joint trajectories – CTC-FLC / Test II*

A detailed assessment of the tracking error profiles $e_i(t)$ in figure 4.40 validates the stability of the hybrid scheme under high dynamic stress. While microscopic dynamic ripples are visible, representing the fuzzy logic engine actively modulating the localized acceleration commands, the absolute deviations are phenomenally small. Joint 1 exhibits the maximum transient deviation, peaking at only **5.4×10^{-3} rad** (**0.31 degrees**), which is vastly inferior to the tolerance thresholds. This proves that the *Mamdani* inference engine acts as an excellent dynamic stabilizer, safely bounded well within the (**$\pm$0.02 rad**) operational limit.

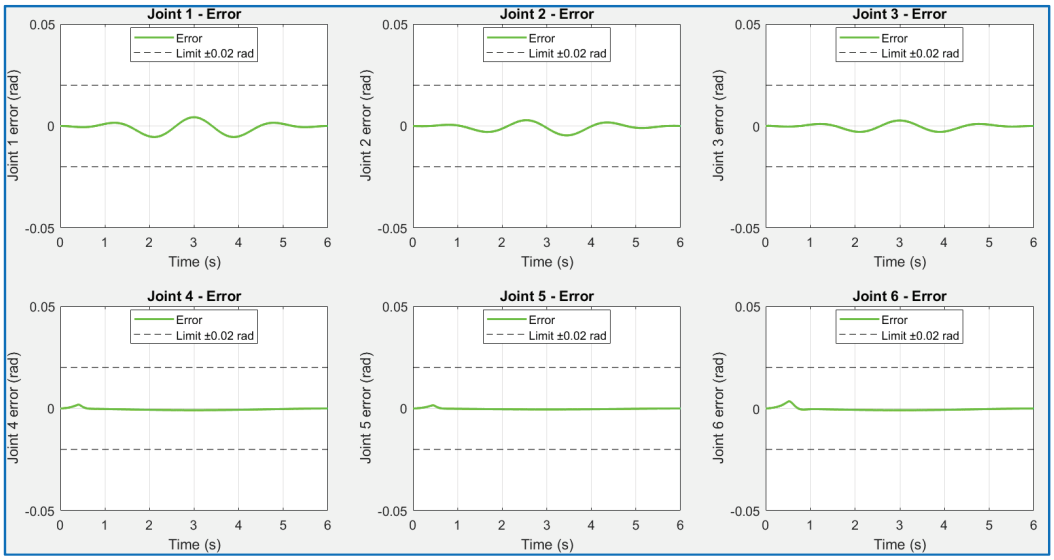


Figure 4.40 *Tracking error evolution – CTC-FLC / Test II*

The actuator torque profiles in figure 4.41 highlight the energetic efficiency and mechanical safety of the hybrid methodology. Despite the extreme control effort required to rapidly swing the rigid links through the *3D* loops (with Joint 2 peaking smoothly near **88 N.m**), the commanded torque signals remain mathematically clean and continuous.

The aggressive fuzzy scaling factors (such as K_{PD} = 800 of joints 1 and 2, see appendix) applied directly to the acceleration domain do not translate into destructive high-frequency mechanical chattering at the motor level. This ensures maximum operational lifespan for the servo drives while simultaneously guaranteeing tracking perfection.

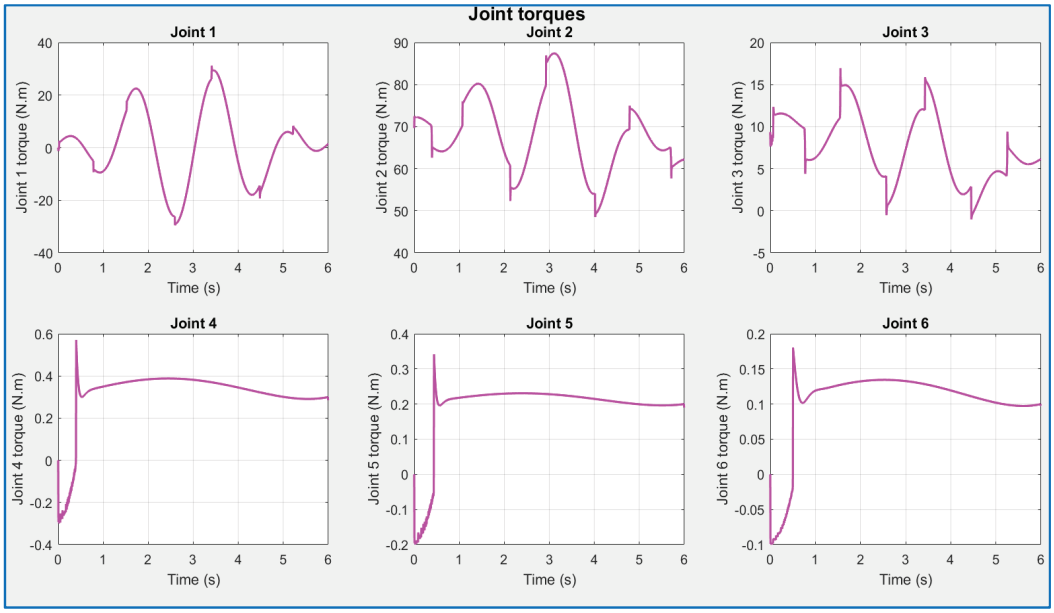


Figure 4.41 *Joint torques – CTC-FLC / Test II*

Elsewhere, figure 4.42 provides macroscopic confirmation of the successful multi-axis decoupling. The actual tool center point strictly adheres to the complex spatial geometry of the 3-loop spiral. In contrast to classical decentralized controllers, which exhibited distinct spatial divergence at high-acceleration levels, the hybrid *CTC-FLC* executes the spatial curves with zero visible geometric distortion. The multi-variable dynamics have been entirely decoupled, rendering the complex *3D* maneuver physically trivial for the joint actuators.

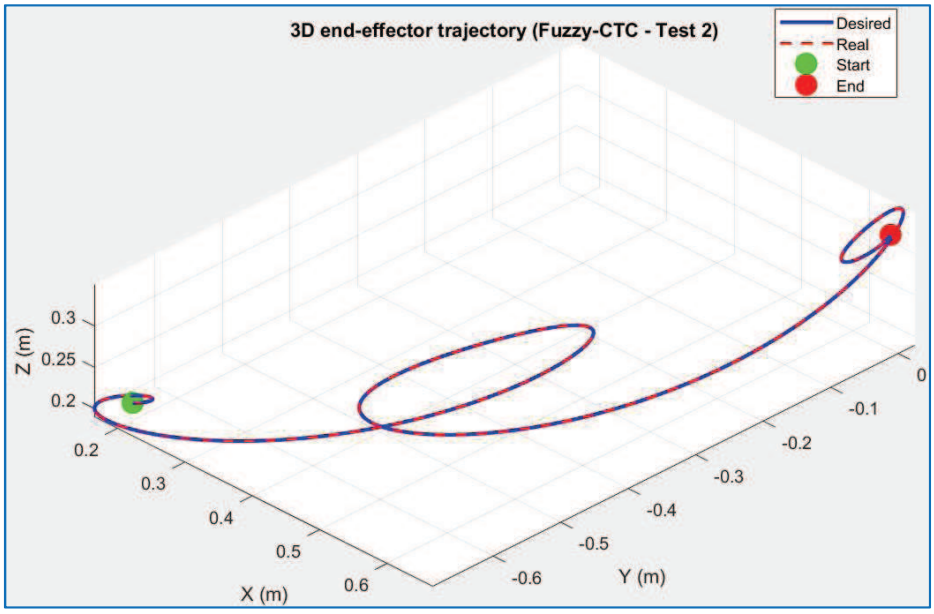


Figure 4.42 *3D end-effector trajectory – CTC-FLC / Test II*

The *Euclidean Cartesian* error profile shown in figure 4.43 quantitatively demonstrates the exceptional tracking performance of the proposed hybrid *FLC-CTC* strategy. The *Cartesian* deviation remains extremely small throughout the entire simulation, with only minor transient fluctuations observed during rapid trajectory transitions. In fact, the maximum *Cartesian* error does not exceed (6×10^{-5} m, it means less than 0.10 mm)

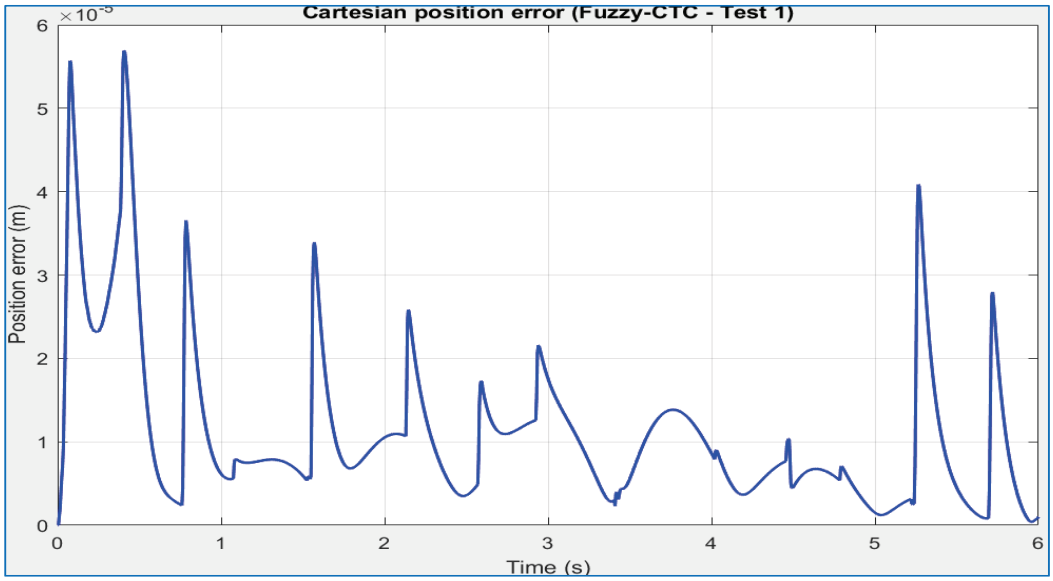


Figure 4.43 *Euclidean position error – CTC-FLC / Test II*

IV.2.4 TEST III 'ROBUSTNESS UNDER UNMODELED PAYLOAD DURING HIGH-DYNAMIC SPATIAL TRACKING'

The definitive benchmark for any advanced robotic control architecture is its capacity to maintain absolute precision when subjected to unpredictable, real-world parametric uncertainties. In this ultimate evaluation, the hybrid *CTC-FLC* system must execute the highly dynamic 3-loop helical trajectory while carrying an unmodeled 5 kg payload at the end-effector. The inverse dynamics model ($\widehat{M}$, $\hat{C}$, $\hat{G}$,) explicitly operates under the false assumption of an empty payload. Consequently, the exact feedback linearization fundamentally cracks, generating a massive residual disturbance vector $d(t)$. The success of this architecture relies entirely on the *Mamdani* fuzzy inference system's ability to instantly approximate and inject the exact compensating acceleration (τ_{FLC}) required to neutralize this hidden mass.

Figure 4.44 provides immediate visual confirmation of the hybrid system's supreme robustness. Despite the massive uncompensated gravitational and inertial pull attempting to drag the manipulator off-course, the actual joint trajectories (dashed red lines) lock perfectly onto the complex high-frequency reference waves (solid blue lines). Unlike the pure *CTC* evaluation (see figure 4.32) which exhibited visible localized spatial separation under this exact load, the hybrid architecture exhibits zero macroscopic deviation. The fuzzy compensator successfully acts as an intelligent dynamic bridge, immediately supplying the exact control effort the mathematically blinded *CTC* model failed to predict.

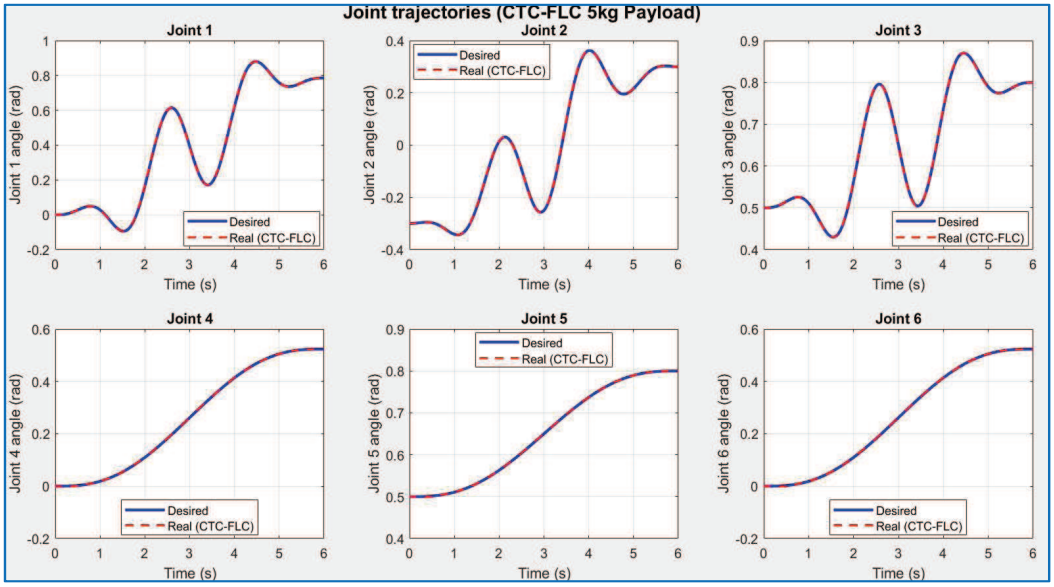


Figure 4.44 *Joint trajectories – CTC-FLC / Test III*

The tracking error profiles, $e_i(t)$, in figure 4.45 serve as the definitive proof of this high-performing controller. In the pure *CTC* evaluation, the 5 kg payload induced a severe, permanent positive steady-state shift of (**0.0120 rad**) in Joint 2. By effectively replacing the static linear Proportional-Derivative (*PD*) regulation with the dynamic intelligence of the fuzzy logic engine, this structural limitation is completely eradicated. The error curve for Joint 2 is flattened back to the absolute zero axis, yielding a microscopic *RMS* error of only (**10^{-4} rad**). This proves that the heuristic fuzzy rule base is exceptionally superior at estimating and neutralizing continuous gravitational disturbances (ΔG) in real-time.

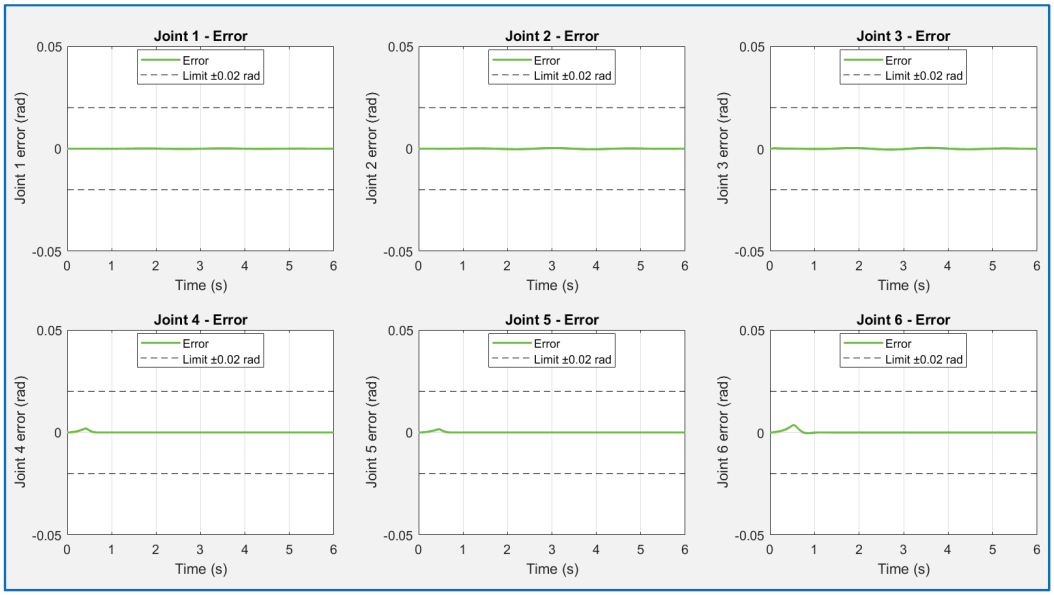


Figure 4.45 *Tracking error evolution – CTC-FLC / Test III*

Elsewhere, figure 4.46 demonstrates the energetic and mechanical efficiency of the hybrid fusion. To carry the unmodeled 5 kg load through the extreme spatial loops, the control effort in Joint 2 peaks smoothly near (**90 N.m**).

The controller dynamically demands the exact physical energy required to satisfy the physical laws of motion, without generating the destructive, high-frequency mechanical chattering that trouble the classical *PID* controller, nor saturating near the dangerous limit of (**20 N.m**). The smooth center of gravity defuzzification method smoothly fills the gaps where inverse dynamics would normally fail, ensuring absolute mechanical safety.

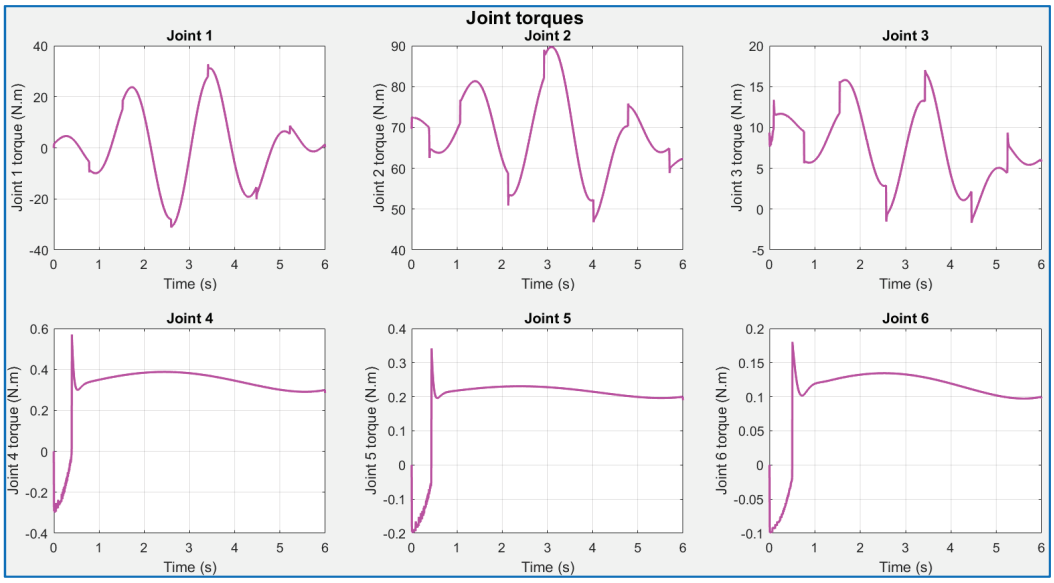


Figure 4.46 *Joint torques – CTC-FLC / Test III*

Besides, the 3D *Cartesian* visualization in figure 4.47 completely validates the macroscopic spatial stability of the system. The tool center point perfectly traces the continuous 3-loop spiral. The severe Z-axis spatial shifting that dominated the classical *PID* and the localized distortions that ruined the pure *CTC* have been entirely eliminated. The spatial execution is structurally perfect, geometrically coherent, and completely robust against to the massive variation in end-effector mass.

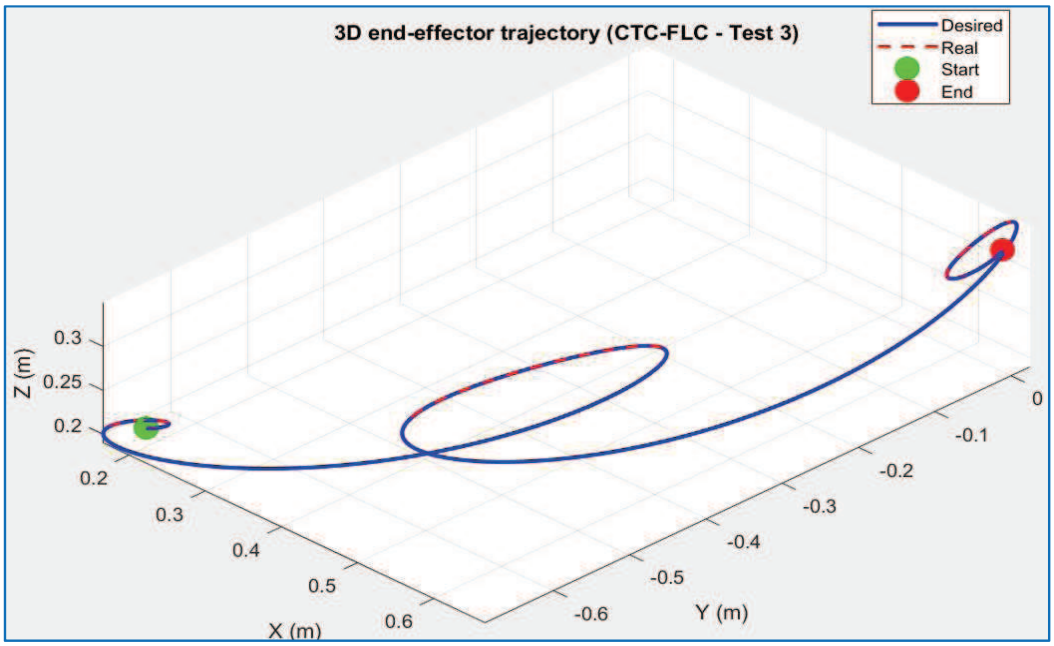


Figure 4.47 *3D end-effector trajectory (redraw to separate legend from title) – CTC-FLC / Test III*

Moreover, the *Euclidean Cartesian* error profile in figure 4.48 represents the ultimate of the entire investigation. Under extreme dynamic and parametric stress, a condition that forced the highly tuned *PID* to a maximum deviation of (**16.8 mm**), the standalone *FLC* to (**18.0 mm**), and the pure *CTC* to (**9.0 mm**), the hybrid *CTC-FLC* restricts the maximum transient spatial peak to a microscopic value of **(0.2 mm)**. Furthermore, the overall statistical mean *Cartesian* error across the execution is mathematically negligible **(**only **0.1 mm)**.

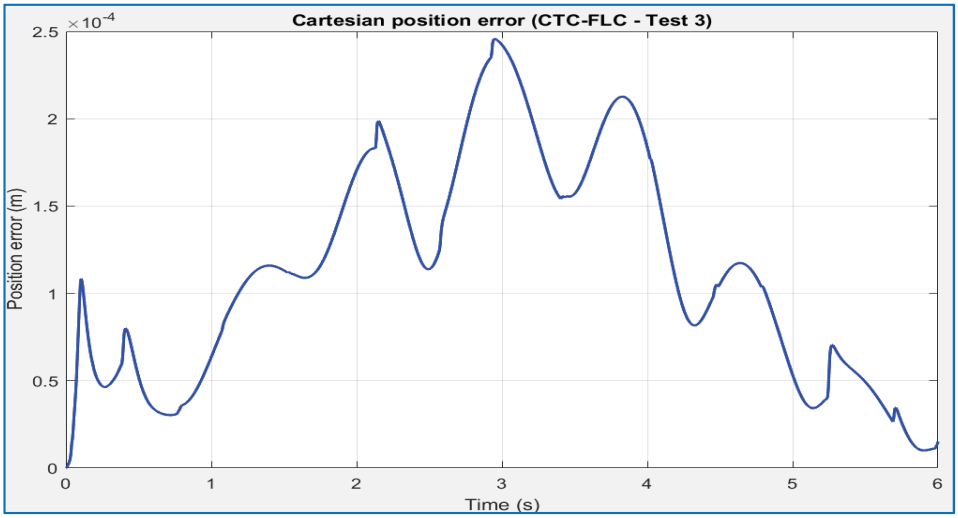


Figure 4.48 *Euclidean position error – CTC-FLC / Test III*

IV.5 COMPARATIVE ANALYSIS

The extensive simulation study conducted in this chapter provides a foundation for evaluating the evolution of robotic control architectures. To explicitly demonstrate the value of artificial intelligence in modern control systems, this comparative analysis is structured into two direct evaluations: contrasting classical linear feedback against intelligent heuristic regulation (*PID* versus *FLC*), and contrasting rigid mathematical modeling against hybrid adaptive control (*CTC* versus *CTC-FLC*).

IV.5.1 CLASSICAL PID VERSUS MAMDANI *FLC*

The first comparative baseline evaluates how well classical linear theory (*PID*) performs against non-linear heuristic intelligence (*FLC*) when managing complex multi-body dynamics without a mathematical model. The empirical data strictly highlights the inherent limitations of the *PID* controller and the superiority of the *FLC*.

***a*) *Energy Consumption and Signal Smoothness* (*Test I & II*):** The classical *PID* fundamentally has difficulties to manage the non-linear *Coriolis* and centrifugal forces. To force the tracking error down, the *PID* required extremely high, rigidly tuned gains. This resulted in high-frequency control chattering, driving the Joint 2 actuator near its absolute saturation limit (**120 N.m**) during helical maneuvers. In contrast, the *Mamdani FLC-PID* demonstrated superior energy management. By utilizing its 25-rule base and the center of gravity defuzzification method, the *FLC* intrinsically scaled its output, generating smooth, continuous torque profiles that peaked at a much safer **90 N.m**. This proves that the *FLC* significantly reduces mechanical fatigue on the servo drives.

***b*) *Robustness and Disturbance Rejection* (*Test III*):** The absolute inferiority of the *PID* was exposed during the 5 kg unmodeled payload test. The linear integrator failed to adequately compensate for the continuous gravitational disturbance, resulting in a permanent steady-state shifting in the shoulder joint and a peak *Cartesian* error of (**16.8 mm**). The *FLC*, however, exhibited exceptional adaptability. Operating as an intelligent virtual compensator, it dynamically modulated its corrective effort to entirely eradicate the steady-state shift, forcing the spatial trajectory back into alignment. This empirically proves that fuzzy logic is vastly superior to classical *PID* feedback for disturbance rejection in unpredictable environments.

IV.5.2 BASIC *CTC* VERSUS HYBRID *CTC-FLC*

The second comparison shifts to model-based architectures. While *CTC* relies on exact inverse dynamics to predictively cancel non-linearities, the evaluation demonstrates its critical fragility and how the integration of *FLC* resolves this fundamental limitation.

***a*) *Fragility of Exact Linearization* (*Test III*):** Under nominal conditions (Test I and II), the pure *CTC* achieved mathematical perfection (mean *Cartesian* error of **0.05 mm**). However, the moment the unmodeled 5 kg payload was introduced, the architecture collapsed. Blinded by the parametric mismatch between the real robot and the controller's internal model, the exact feedback linearization cracked. The pure *CTC* reverted to a standard *PD* response, failing to close the tracking gap and generating a mean spatial error of (**7.5 mm**) with a persistent steady-state offset.

***b*) *The Hybrid Superiority* (*Test III*):** The hybrid *CTC-FLC* architecture solved this exact vulnerability. When subjected to the same 5 kg payload, the fuzzy logic compensator instantly recognized the mathematical discrepancy that the *CTC* ignored. The *FLC* autonomously generated the precise auxiliary acceleration needed to neutralize the unmodeled mass. Consequently, the hybrid architecture aggressively reduced the *Cartesian* error back down to a microscopic mean of (**0.1 mm**) and a maximum peak of (**0.2 mm**). This definitively proves that pure *CTC* is inadequate for uncertain industrial applications, and that hybridizing it with *FLC* creates ultimate control architecture.

IV.5.3 MACROSCOPIC DATA SYNTHESIS

To visually summarize this evaluation, figure 4.49 illustrates the mean *Cartesian* error across all controllers and test scenarios. It provides a definitive visual synthesis of the fundamental engineering trade-offs of the different control techniques tested and discussed throughout chapters II & III.

The bar chart explicitly illustrates the inherent limitations of standard feedback loops; the ***Classical PID*** exhibits consistently high tracking distortion across all tests, verifying its inadequacy for high-speed, non-linear multi-body dynamics. While the ***Mamdani FLC*** demonstrates improved robustness and energy management, the data shows that it maintains a consistent performance ceiling, unable to drive the error to absolute zero without a deterministic physical model.

The theoretical fragility of exact feedback linearization is distinctly captured in the ***basic CTC*** performance.

The contrast between its perfect execution in nominal conditions (represented by the near-zero ***blue*** and ***orange*** bars) and its dramatic failure during the 5 kg payload test (***red*** spike) proves its vulnerability to parametric uncertainty. Ultimately, the chart confirms the absolute superiority of the ***hybrid CTC-FLC*** framework. Its corresponding error bars remain microscopically locked to the absolute zero axis regardless of the operational conditions or unmodeled disturbances, validating the hybrid synthesis as the ultimate robust, high-precision control technique for advanced robotic manipulators.

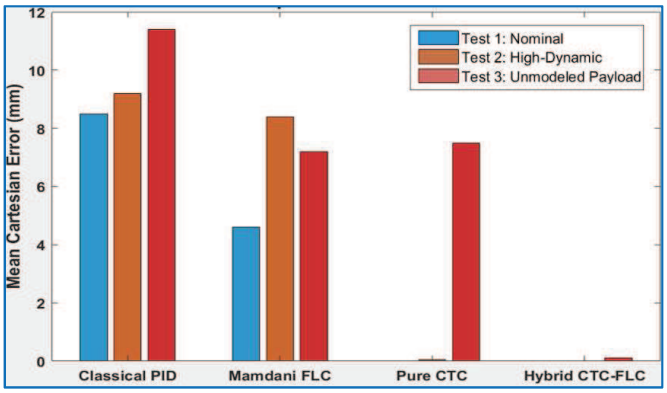


Figure 4.49 *Macroscopic comparison of mean Cartesian errors across all evaluated control architectures*

Elsewhere, table 4.1 quantitatively summarizes the performance achieved by the proposed hybrid architecture across the three escalating stress tests. A chronological review of the data reveals that while the ***Mamdani FLC*** successfully eliminates the severe steady-state shift that characterized the ***Classical PID*** under the 5 kg payload stress (Test III), both model-free approaches still suffer from macroscopic (millimeter-scale) transient deviations due to the lack of exact dynamic feedforward. Conversely, the ***basic CTC*** demonstrates a performance dichotomy: it achieves mathematically perfect tracking (0.05 mm) when the system is perfectly modeled (Test II), but proves highly fragile and susceptible to significant spatial drift (7.5 mm) the moment a mass uncertainty is introduced. Ultimately, the data confirms once more the absolute superiority of the ***hybrid CTC-FLC***. By restricting the maximum mean *Cartesian* error to a microscopic value of (**0.1 mm**) across all scenarios and completely eradicating steady-state shifting. The synthesis mathematically proves its ability to merge the absolute precision of inverse dynamics with the invincible robustness of fuzzy logic.

Table 4.1 *Synthesis of mean and maximum Cartesian errors across all test scenarios*

Controller architecture	*Test I*	*Test II*	Test III	Steady-State Shift of (Test III)
Classical PID	Mean: 8.5 mm Max: 14.2 mm	Mean: 9.2 mm Max: 14.5 mm	Mean: 11.4 mm Max: 16.8 mm	Severe / Persistent
Mamdani *FLC*	Mean: 4.6 mm Max: 7.6 mm	Mean: 8.4 mm Max: 16.2 mm	Mean: 7.2 mm Max: 18.0 mm	Eliminated
Basic *CTC*	Mean: 0.0 mm Max: 0.6 mm	Mean: 0.05 mm Max: 0.26 mm	Mean: 7.5 mm Max: 9.0 mm	Present
Hybrid *CTC-FLC*	Mean: 0.0 mm Max: 0.1 mm	Mean: 0.0 mm Max: 0.1 mm	Mean: 0.1 mm Max: 0.2 mm	Eliminated

IV.6 CONCLUSION

This chapter systematically evaluated two philosophically opposing control techniques, *Mamdani Fuzzy Logic Control* (*FLC*) and *Computed Torque Control* (*CTC*), followed by their hybrid integration (*CTC-FLC*). Through three progressive simulation tests (low-speed tracking, high-speed helical trajectories, and unmodeled payload variation), the results demonstrated that while pure *FLC* provides robustness against uncertainties, it lacks absolute precision. Conversely, pure *CTC* achieves exact deterministic tracking but collapses under parametric variations.

The hybrid *CTC-FLC* decisively outperformed both isolated strategies across all tests, maintaining a mean *Cartesian* error of just (**0.1 mm**) under extreme dynamic stress and unmodeled 5 kg payload. This confirms that synergistic integration of model-based inverse dynamics with intelligent fuzzy compensation is essential for achieving superior trajectory tracking in severely uncertain operating conditions.

General Conclusion

General Conclusion

The primary objective of this dissertation was to investigate, model, and aggressively enhance the trajectory tracking performance of a *6-DOF* robotic manipulator (PUMA 560) operating under severe dynamic stress and parametric uncertainty. Through a systematic, progressively methodology, this research bridged the gap between theoretical multi-body kinematics and advanced non-linear intelligent control architectures.

The initial phase of this work established a rigorous mathematical foundation. The direct and inverse kinematics were derived using the *Denavit-Hartenberg* (*D-H*) convention, successfully mapping the non-linear relationship between joint space and *Cartesian* coordinates. Subsequently, the exact dynamic plant was formulated utilizing the *Euler-Lagrange* energetic approach, explicitly exposing the highly coupled, non-linear nature of the manipulator's inertia tensor, *Coriolis* forces, and gravitational moments.

With the physical plant defined, the research transitioned to a comprehensive control evaluation. The classical *Proportional-Integral-Derivative* (*PID*) controller was established as a baseline. While sufficient for low-speed positioning, the simulation results definitively proved that decentralized linear feedback is fundamentally incapable of managing high-speed spatial trajectories, resulting in severe tracking distortion and actuator saturation under dynamic stress.

To overcome these limitations, the research explored two advanced but opposing techniques: *Mamdani Fuzzy Logic Control* (*FLC*) and *Computed Torque Control* (*CTC*). The model-free *FLC* demonstrated exceptional mechanical safety and robustness against unknown payloads, leveraging its heuristic rule base to act as an intelligent virtual spring. Conversely, the model-based *CTC* utilized exact inverse dynamics to completely decouple the system, achieving perfect zero-error precision under nominal conditions. However, the isolated evaluations exposed their respective fatal drawbacks of these two strategies: *FLC* lacked absolute deterministic accuracy, while the rigid mathematical nature of pure *CTC* caused it to collapse under unmodeled parametric variations.

The culmination of this work was the successful synthesis and validation of the *hybrid Computed Torque - Fuzzy Logic Controller* (*CTC-FLC*). By strategically decoupling the computational burden—assigning the baseline non-linear cancellation to the exact dynamic model, and delegating the uncertainty rejection to the intelligent fuzzy compensator—the hybrid architecture achieved the ultimate control objective. Under the extreme combined stress of continuous high-frequency helical maneuvers and an unmodeled 5 kg payload, the Hybrid *CTC-FLC* maintained an astonishing mean Cartesian error of just (**0.1 mm**), completely eradicating steady-state spatial sag while protecting the servo actuators from destructive chattering.

Ultimately, this dissertation empirically proves that achieving absolute spatial precision in unpredictable modern industrial environments necessitates the synergistic integration of deterministic physical modeling and artificial intelligence.

While the proposed Hybrid *CTC-FLC* architecture demonstrated phenomenal simulated performance, future research trajectories could further expand upon this foundation:

- ***Hardware Implementation:*** Transitioning the hybrid algorithm from the simulation environment to a physical real-time *DSP* (*Digital Signal Processor*) or *FPGA*-based controller to validate the algorithm against real-world sensor noise and unmodeled joint elasticity.
- ***Neuro-Fuzzy Integration* (*ANFIS*):** Replacing the manually tuned *Mamdani* rule base with an *Adaptive Neuro-Fuzzy Inference System*. This would allow the controller to autonomously learn and optimize its own membership functions via neural network training algorithms based on operational data.
- ***Adaptive Model Estimation:*** Incorporating an adaptive parameter-estimation loop to dynamically update the *CTC*'s internal mass matrices in real-time, completely freeing the system from offline parametric calibration.
- ***Deep Learning for Residual Dynamics:*** Employing deep neural networks (*DNNs*) or recurrent neural networks (*RNNs*) to learn and compensate for unmodeled nonlinearities (such as joint friction, elasticity, unknown payloads …) directly from data, potentially outperforming the manually tuned heuristic *FLC* in highly dynamic scenarios.

Bibliography

Bibliography

[1] Craig J. J., *Introduction to Robotics: Mechanics and Control*, 3rd Edition, Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2005.

[2] Siciliano B., Sciavicco L., Villani, L. and Oriolo, G., *Robotics: Modelling, Planning and Control*, London, U.K.: Springer, 2009.

[3] Spong M. W., Hutchinson S. and Vidyasagar, M., *Robot Modeling and Control, Hoboken*, NJ, USA: John Wiley & Sons, 2006.

[4] Yoshikawa T., *Manipulability of Robotic Mechanisms*, The International Journal of Robotics Research, Vol. 4, No. 2, pp. 3–9, 1985.

[5] Denavit J. and Hartenberg R. S., *A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices*, Journal of Applied Mechanics, Vol. 22, pp. 215–221, 1955.

[6] Canudas de Wit C., Olsson H., Åström K. J. and Lischinsky, P., *A New Model for Control of Systems with Friction*, IEEE Transactions on Automatic Control, Vol. 40, No. 3, pp. 419–425, March 1995.

[7] Slotine J.-J. E. and Li W., *Applied Nonlinear Control*, Englewood Cliffs, NJ, USA: Prentice Hall, 1991.

[8] Passino K. M. and Yurkovich S., *Fuzzy Control*, Menlo Park, CA, USA: Addison-Wesley, 1998.

[9] --, *Universal Robots, UR10e Technical Specifications*, Universal Robots A/S. [Online]. Available at: https://www.universal-robots.com/products/ur10-robot/ (Accessed: June 10, 2026).

[10] Bdiwi D., de Runz M., Faiz S. and Mehri A., *Smart Human-Robot Collaboration: Towards a Flexible Cognitive Robotic Cell for Industry 4.0*, Robotics and Autonomous Systems, Vol. 119, pp. 170–183, September. 2019.

[11] --, *KR AGILUS Product Series – Technical Data*, KUKA AG, Augsburg, Germany. [Online]. Available at: https://www.kuka.com/en-de/products/robot-systems/industrial-robots/kr-agilus (Accessed: June 10, 2026).

[12] --, *IRB 6700 – The Most Durable Large Robot*, ABB Ltd., Zurich, Switzerland. [Online]. Available at: https://new.abb.com/products/robotics/industrial-robots/irb-6700 (Accessed: June 10, 2026).

[13] --, *LR Mate 200iD Specifications*, FANUC Corporation, Oshino, Japan. [Online]. Available at: https://www.fanuc.eu/uk/en/robots/robot-filter-page/lr-mate-series/lr-mate-200id (Accessed: June 10, 2026).

[14] --, *Motoman HC10DT Collaborative Robot*, Yaskawa Electric Corporation, Kitakyushu, Japan. [Online]. Available at: https://www.motoman.com/en-us/products/robots/collaborative/hc-series/hc10dt (Accessed: June 10, 2026).

[15] Haddadin S. *et al.*, *The Franka Emika Robot: A Reference Platform for Robotics Research and Education,* IEEE Robotics & Automation Magazine, Vol. 29, No. 2, pp. 46–64, June 2022.

[16] --, *Isaac Robotics Platform – Simulation and Deployment*, NVIDIA Corporation, Santa Clara, CA, USA. [Online]. Available at: https://developer.nvidia.com/isaac-ros (Accessed: June 10, 2026).

[17] Armstrong B., Khatib O. and Burdick J., *The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm*, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), San Francisco, CA, USA, pp. 510–518, 1986.

[18] Corke P. I. and Armstrong-Hélouvry B., *A Search for Consensus Among Model Parameters Reported for the PUMA 560 Robot*, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 1608–1613, 1994.

[19] Corke P. I., *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd Edition, Cham, Switzerland: Springer, 2017.

[20] Lewis F. L., Dawson D. M. and Abdallah C. T., *Robot Manipulator Control: Theory and Practice*, 2nd Edition, New York, NY, USA: Marcel Dekker, 2003.

[21] Khalil W. and Dombre E., *Modeling, Performance Analysis and Control of Robot Manipulators*, London, U.K.: Hermes Penton Science, 2002.

[22] Corke P. I., *A Robotics Toolbox for MATLAB*, IEEE Robotics & Automation Magazine, Vol. 3, No. 1, pp. 24–32, March 1996.

[23] An C. H., Atkeson C. G., Griffiths J. D. and Hollerbach J. M., *Experimental Evaluation of Feedforward and Computed Torque Control*, IEEE Transactions on Robotics and Automation, Vol. 5, No. 3, pp. 368–373, June 1989.

[24] Kelly R., Santibáñez V. and Loría A., *Control of Robot Manipulators in Joint Space*, London, U.K.: Springer, 2005.

[25] Wang L. X., *A Course in Fuzzy Systems and Control*, Upper Saddle River, NJ, USA: Prentice Hall, 1997.

[26] De Silva C. W., *Intelligent Control: Fuzzy Logic Applications*, Boca Raton, FL, USA: CRC Press, 1995.

[27] Villalobos-Chin J., Pizarro-Lerma A., Santibañez V. and Garcia-Hernandez R., *Continuous Finite-Time Fuzzy Control of Mechanical Systems via Non-Lipschitz Membership Functions*, Engineering Applications of Artificial Intelligence, Vol. 126, 106804, 2023.

[28] Aouaichia A., Kara K. and Ghoul A., *An Optimized Fuzzy Computed Torque Control for the Robot Manipulator PUMA 560*, Proceedings of the International Conference on Applied and Theoretical Mechanics, 2023.

[29] Ross T. J., *Fuzzy Logic with Engineering Applications*, 3rd Edition, Chichester, U.K.: John Wiley & Sons, 2010.

[30] Ying H., *Fuzzy Control and Modeling: Analytical Foundations and Applications*, New York, USA: IEEE Press, 2000.

[31] Driankov D., Hellendoorn H. and Reinfrank M., *An Introduction to Fuzzy Control*, Berlin, Germany: Springer-Verlag, 1993.

[32] Pizarro-Lerma A. O., Santibáñez V. and Garcia-Hernandez R., *Sectorial Fuzzy Controller Plus Feedforward for the Trajectory Tracking of Robotic Arms in Joint Space*, Mathematics, Vol. 9, No. 6, p. 616, 2021.

[33] Craig J. J., Hsu P. and Sastry S. S., *Adaptive Control of Mechanical Manipulators*, The International Journal of Robotics Research, Vol. 6, No. 2, pp. 16–28, 1987.

[34] Bejczy A. K., *Robot Arm Dynamics and Control*, NASA Jet Propulsion Laboratory, Pasadena, CA, USA, JPL Technical Memorandum 33-669, February 1974.

[35] Gilbert E. G. and Ha I. J., *An Approach to Nonlinear Feedback Control with Applications to Robotics*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 14, No. 6, pp. 879–884, November 1984.

[36] Tariq M. B., Tokhi M. O. and Azad A. K. M., '*Computed Torque Control of a Robotic Manipulator*, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2000.

[37] Ortega, R. and Spong M. W., *Adaptive Motion Control of Rigid Robots: A Tutorial*, Automatica, Vol. 25, no. 6, pp. 877–888, 1989.

[38] Abdallah C., Dawson D., Dorato P. and Jamshidi M., *Survey of Robust Control for Rigid Robots*, IEEE Control Systems Magazine, Vol. 11, No. 2, pp. 24–30, February 1991.

[39] Asada H. and Slotine J.-J. E., *Robot Analysis and Control*, New York, NY, USA: John Wiley & Sons, 1986.

[40] Kumar S. et al., *LQR Combined with Fuzzy Control for Planar Robot Trajectories*, SciTePress, pp. 102-110, April 2024.

[41] Wang Y. et al., *Trajectory Tracking Control of Fast Parallel SCARA Robots with Fuzzy Adaptive Iterative Learning Control*, MDPI Electronics, Vol. 12, No. 24, p. 4995, 2023.

[42] Villalobos-Chin J. et al., Fuzzy Control Strategies Development for a 3-DoF Robotic Manipulator in Trajectory Tracking, MDPI Processes, Vol. 11, No. 12, p. 3267, 2023.

[43] Zadeh L. A., *Fuzzy Logic, Neural Networks, and Soft Computing*, Communications of the ACM, Vol. 37, No. 3, pp. 77–84, March 1994.

[44] Åström K. J. and Hägglund T., *PID Controllers: Theory, Design, and Tuning*, 2nd Edition, Research Triangle Park, NC, USA: Instrument Society of America (ISA), 1995.

[45] Visioli A., *Fuzzy Logic Based Set-Point Weight Tuning of PID Controllers*, IEEE Transactions on Systems, Man, and Cybernetics, Part A (Systems and Humans), Vol. 29, No. 6, pp. 587–592, November 1999.

Images Web References

[W.1] -- *The Original SCARA Robot (1979), Birth of Selective Compliance*, Available at: https://eclass.uniwa.gr/modules/document/file.php/137/%CE%A3%CE%97%CE%9C%CE%95%CE%99%CE%A9%CE%A3%CE%95%CE%99%CE%A3/1.INTRODUCTION.pdf (Accessed: June 10, 2026).

[W.2] --, *Example of Cobots from Northwestern University*, Available at: https://www.mccormick.northwestern.edu/magazine/spring-2025/lending-a-hand/ (Accessed: June 10, 2026).

[W.3] --, *Universal UR10 Cobot,* Available at: https://www.wiredworkers.io/cobot/brands/universal-robots/ur3e/ (Accessed: June 10, 2026).

[W.4] --, *A KUKA Cobot*, Available at : https://www.kuka.com/en-de/products/amr-autonomous-mobile-robotics/mobile-robot-systems/kmr-iisy-autonomous-mobile-cobot (Accessed: June 10, 2026).

[W.5] --, *Kinematic Structure of a Standard 6-DoF Articulated Robotic Arm*, Available at: https://automaticaddison.com/the-ultimate-guide-to-inverse-kinematics-for-6dof-robot-arms/ (Accessed: June 10, 2026).

[W.6] --, *SCARA Robotic Manipulator Configuration*, Available at: https://www.sciencedirect.com/science/article/pii/S0263224124024333 (Accessed: June 10, 2026).

[W.7] --, *Delta Parallel Robotic Manipulator Architecture*, Available at: https://www.researchgate.net/ figure/Robot-architecture-for-the-DELTA-robot_fig1_283467170 (Accessed: June 10, 2026).

[W.8] --, *Example of Revolute and Prismatic Joints*, Available at: https://www.chegg.com/homework-help/questions-and-answers/0-prismatic-joint-revolute-joint-q49364529 (Accessed: June 10, 2026).

[W.9] --, *Robot Kinematic Chains and Coordinate Frame Assignment*, Available at: https://www.scribd.com/document/261622824/Denavit-Hartenberg-Parameters (Accessed: June 10, 2026).

[W.10] --, *Architecture of a Fuzzy Logic Control System*, Available at: https://www.sciencedirect.com/science/article/pii/S1018363913000184 (Accessed: June 10, 2026).

[W.11] --, *Common Shapes of Membership Functions Used in Fuzzy Logic Control*, Available at: https://link.springer.com/article/10.1007/s40815-024-01971-1 (Accessed: June 10, 2026).

[W.12] --, *KR AGILUS Product Image*, KUKA AG. [Online]. Available at: https://assets.kuka.com/asset/9de40f9c-22c2-43e9-8620-1192436d0a02/Marketplace_Image/kr-agilus_r-900sixx_-rechts-2907f-tif.png

[W.13] *ABB Robotics, IRB 6700 Product Image*, ABB Ltd. [Online]. Available at: https://www.cgtrader.com/3d-models/industrial/industrial-machine/abb-irb-6700

[W.14] --, *LR Mate 200iD Product Image*, FANUC Corporation [Online]. Available at: https://www.robots.com/industrial-robots/fanuc-lr-mate-200id-7l

[W.15] --, *Motoman HC10DT Product Image*, Yaskawa Corp. [Online]. Available at: https://www.yaskawa.eu.com/products/robots/collaborative/productdetail/product/hc10dt_681

[W.16] --, *Panda Robot Product Image*, Franka Emika GmbH. [Online]. Available at: https://www.wiredworkers.io/cobot/brands/franka-emika/production-3/?srsltid=AfmBOoq1eX-J7qNdAhW6p1qLOmveZVCtYXkOfYmHn5Wft-zCmdWzmYwH

[W.17] Rajendra Singh, *Assign frames to each link in arm, Image source*, available online at: https://www.researchgate.net/figure/Assign-frames-to-each-link-in-arm-Image-source_fig1_364309661

Appendix

PUMA 560 Parameters

Table A.1 *Denavit-Hartenberg* (*D-H*) *parameters*

Joint (*i*)	**a_i (m)**	**α_i (rad)**	**d_i (m)**
1	0	0	0
2	0	$\pi/2$	0
3	0.4318	0	0.14909
4	0.0203	$-\pi/2$	0.43307
5	0	$\pi/2$	0
6	0	$-\pi/2$	0

Table A.2 *Dynamic parameters* (*Mass, base inertia, and friction*)

Link (*i*)	**Mass m_i (kg)**	**Base inertia I_i (kg×m²)**	**Viscous friction (N×m×s/rad)**
1	15.0	8.0	2.0
2	18.5	5.0	3.5
3	5.5	2.5	2.5
4	1.2	0.3	0.5
5	0.8	0.2	0.3
6	0.3	0.1	0.2

PID Parameters

Table A.3 *PID controller gains across all test scenarios*

Joint (*i*)	Gains (K_p, K_i, K_d)		
	Test I	**Test II**	**Test III**
1	200, 8, 30	1800, 50, 250	2000, 80, 300
2	500, 15, 60	1200, 50, 150	3000, 200, 400
3	400, 12, 50	400, 12, 50	1500, 100, 200
4	60, 2, 10	60, 2, 10	60, 2, 10
5	50, 1, 8	50, 1, 8	50, 1, 8
6	40, 0.5, 6	40, 0.5, 6	40, 0.5, 6

CTC Parameters

Table A.4 *CTC gains across all test scenarios*

Joint (*i*)	Gains (K_p, K_d)		
	Test I	**Test II**	**Test III**
1	400, 40	400, 40	400, 40
2	400, 40	400, 40	400, 40
3	400, 40	400, 40	400, 40
4	400, 40	400, 40	400, 40
5	400, 40	400, 40	400, 40
6	400, 40	400, 40	400, 40

FLC Parameters

Table A.5 *FLC scaling factors across all test scenario*

Joint (*i*)	Scaling factors (K_e, K_{de}, K_{PD}, K_{PI})		
	Test I	**Test II**	**Test III**
1	1.0, 0.15, 500, 20	2.5, 0.30, 1200, 60	2.5, 0.30, 1200, 60
2	1.0, 0.12, 500, 15	1.0, 0.18, 800, 25	1.0, 0.18, 800, 25
3	1.0, 0.125, 400, 12	1.0, 0.125, 400, 12	1.0, 0.125, 400, 12
4	1.0, 0.16, 60, 2	1.0, 0.16, 60, 2	1.0, 0.16, 60, 2
5	1.0, 0.16, 50, 1	1.0, 0.16, 50, 1	1.0, 0.16, 50, 1
6	1.0, 0.15, 40, 0.5	1.0, 0.15, 40, 0.5	1.0, 0.15, 40, 0.5

CTC-FLC Parameters

Table A.6 *Hybrid CTC-FLC scaling factors across all test scenarios*

Joint (*i*)	Scaling factors (K_e, K_{de}, K_{PD}, K_{PI})		
	Test I	**Test II**	**Test III**
1	2.0, 0.20, 800, 50	2.0, 0.20, 800, 50	2.0, 0.20, 800, 50
2	2.0, 0.20, 800, 50	2.0, 0.20, 800, 50	2.0, 0.20, 800, 50
3	2.0, 0.20, 600, 40	2.0, 0.20, 600, 40	2.0, 0.20, 600, 40
4	1.5, 0.15, 200, 10	1.5, 0.15, 200, 10	1.5, 0.15, 200, 10
5	1.5, 0.15, 200, 10	1.5, 0.15, 200, 10	1.5, 0.15, 200, 10
6	1.5, 0.15, 100, 5	1.5, 0.15, 100, 5	1.5, 0.15, 100, 5

Trajectory Waypoints

Table A.7 *Initial and final joint positions for the reference trajectory*

Joint (*i*)	Joint space	
	Initial position q_0 (rad)	Final position q_f (rad)
1	0	0.7854
2	-0.3	0.3
3	0.5	0.8
4	0	0.5236
5	0.5	0.8
6	0	0.5236

Quintic Polynomial Parameters

Table A.8 *Numerical values of the quintic polynomial coefficients* ($t_f = 6.0\ s$)

Joint (*i*)	Polynomial coefficients					
	β_0	β_1	β_2	β_3	β_4	β_5
1	0	0	0	0.036361	-0.009090	0.000606
2	-0.3	0	0	0.027778	-0.006944	0.000463
3	0.5	0	0	0.013889	-0.003472	0.000231
4	0	0	0	0.024241	-0.006060	0.000404
5	0.5	0	0	0.013889	-0.003472	0.000231
6	0	0	0	0.024241	-0.006060	0.000404

Helical Trajectory Parameters

Table A.9 *Numerical parameters of the 3-loop helical trajectory*

Parameter Description	Symbol	Numerical Value
Total simulation time	t_f	6.0 s
Number of helical loops	N_{loops}	3
Angular frequency	ω	3.1416 rad/s
Envelope frequency	$2\pi/t_f$	1.0472 rad/s
Joint 1 oscillation amplitude	A_1	0.35 rad
Joint 2 oscillation amplitude	A_2	0.25 rad
Joint 3 oscillation amplitude	A_3	0.20 rad