



U

P

T

**“CONTRIBUTIONS TO NONLINEAR CONTROL OF  
PARALLEL KINEMATIC MACHINES: FROM  
DESIGN TO REAL-EXPERIMENTS”**

Por

**JONATAN MARTÍN ESCORCIA HERNÁNDEZ**

Tesis sometida como requisito parcial para obtener el grado de:

**DOCTOR EN OPTOMECASTRÓNICA**

En la

**UNIVERSIDAD POLITÉCNICA DE TULANCINGO**

Tulancingo de Bravo, Hidalgo, México

Noviembre 2020

Directores de Tesis:

**Dr. Ahmed Chemori**

LIRMM - CNRS, Université de Montpellier

**Dr. Hipólito Aguilar Sierra**

Universidad Politécnica de Tulancingo

©UPT 2020

Derechos reservados

El autor otorga a UPT el permiso de reproducir  
y distribuir copias de esta tesis en su totalidad  
o en partes.









U

P

T

**“CONTRIBUTIONS TO NONLINEAR CONTROL OF  
PARALLEL KINEMATIC MACHINES: FROM  
DESIGN TO REAL-EXPERIMENTS”**

**Tesis de Doctorado en  
Optomecatrónica**

Por:

**JONATAN MARTÍN ESCORCIA HERNÁNDEZ**

Maestro en Automatización y Control

Universidad Politécnica de Tulancingo 2017-2020

Becario de CONACYT, CVU no. 648169

Apoyos recibidos en las convocatorias, 291250, 291276 y 291197

Directores de Tesis:

**Dr. Ahmed Chemori**

**Dr. Hipólito Aguilar Sierra**

Tulancingo de Bravo, Hidalgo, México, noviembre 2020





---

# Contents

<b>Contents</b>	<b>IV</b>
<b>List of Figures</b>	<b>VIII</b>
<b>List of Tables</b>	<b>IX</b>
<b>List of Acronyms</b>	<b>XI</b>
<b>List of Variables</b>	<b>XIII</b>
<b>Acknowledgments</b>	<b>XIX</b>
<b>General Introduction</b>	<b>XXI</b>
<b>I Context, Problem Formulation and State of the Art</b>	<b>1</b>
<b>1 Context and Problem Formulation</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Classification of Robotic Manipulators . . . . .	4
1.2.1 Serial Manipulators . . . . .	4
1.2.2 Parallel Kinematic Manipulators . . . . .	4
1.2.3 Serial versus Parallel Manipulators . . . . .	4
1.2.4 Hybrid Manipulators . . . . .	7
1.3 Overview of Parallel Kinematic Machines . . . . .	8
1.3.1 Historical review of PKMs . . . . .	8
1.3.2 Applications Developed by Parallel Kinematic Machines . . . . .	14
1.4 Control Problem Formulation . . . . .	20
1.4.1 Challenges in Control for PKMs . . . . .	21
1.5 Objectives of the Thesis . . . . .	22
1.6 Main Contributions of the Thesis . . . . .	23

1.6.1	Kinematic and Dynamic models for a New RA-PKM called SPIDER4	23
1.6.2	Control solutions for PKMs Based-on RISE Control . . . . .	24
1.7	Conclusion . . . . .	25
<b>2</b>	<b>State of the Art</b>	<b>27</b>
2.1	Introduction . . . . .	27
2.2	Dynamic Modeling of Parallel Kinematic Machines . . . . .	28
2.2.1	Dynamic modeling approaches for PKM . . . . .	28
2.2.2	Dynamic Modeling Representation . . . . .	33
2.3	Overview of Motion Controllers for PKMs . . . . .	35
2.4	Non-model-based controllers . . . . .	35
2.4.1	Non-model-based-non-adaptive controllers . . . . .	37
2.4.2	Non-model-based-adaptive Controllers . . . . .	38
2.5	Model-Based-Controllers . . . . .	39
2.5.1	Model-based-non-adaptive Controllers . . . . .	39
2.5.2	Model-based-adaptive Controllers . . . . .	41
2.6	Conclusion . . . . .	46
<b>II</b>	<b>Modeling and Proposed Control Solutions</b>	<b>47</b>
<b>3</b>	<b>Description and Modeling of Experimental Platforms</b>	<b>49</b>
3.1	Introduction . . . . .	49
3.2	Redundant PKM (SPIDER4) . . . . .	50
3.2.1	Description and Modeling of SPIDER4 . . . . .	50
3.2.2	Inverse Kinematic Model of SPIDER4 RA-PKM . . . . .	50
3.2.3	Forward Kinematic Model of SPIDER4 RA-PKM . . . . .	55
3.2.4	Velocity Relationship and Jacobian Analysis of SPIDER4 RA-PKM . . . . .	57
3.2.5	Inverse Dynamic Model of the delta-like positioning mechanism of SPIDER4 . . . . .	58
3.2.6	Inverse Dynamic Model of the Wrist . . . . .	61
3.2.7	Inverse Dynamic Model of SPIDER4 RA-PKM . . . . .	62
3.3	Non-Redundant PKM (Delta) . . . . .	63
3.3.1	Description and Modeling of Delta robot . . . . .	63
3.3.2	Inverse Kinematic Model of Delta PKM . . . . .	64
3.3.3	Forward Kinematic Model of Delta PKM . . . . .	66
3.3.4	Velocity Relationship and Jacobian Analysis of Delta PKM . . . . .	68
3.3.5	Inverse Dynamic Model of Delta PKM . . . . .	69
3.4	Conclusion . . . . .	70
<b>4</b>	<b>Proposed Control Solutions</b>	<b>73</b>
4.1	Introduction . . . . .	73
4.2	Background on RISE-Based Controllers . . . . .	74

4.2.1	Original RISE Control . . . . .	74
4.3	Background on Artificial Neural Networks . . . . .	77
4.3.1	ANNs general model . . . . .	79
4.3.2	The activation function . . . . .	79
4.3.3	ANNs architectures . . . . .	79
4.3.4	Application of ANNs to robotics control . . . . .	81
4.4	Control Solution 1: A RISE Controller with Nominal Feedforward . . . . .	81
4.4.1	Motivation . . . . .	81
4.4.2	Proposed Control law . . . . .	82
4.5	Control Solution 2: A RISE Feedforward controller with Adaptive Feedback Gains . . . . .	85
4.5.1	Motivation . . . . .	85
4.5.2	Proposed Control Law . . . . .	86
4.6	The actuation redundancy on SPIDER4 . . . . .	89
4.7	Control Solution 3: A RISE controller with Adaptive Feedforward Compensation Based-on B-Spline Neural Networks . . . . .	90
4.7.1	Motivation . . . . .	90
4.7.2	Controller Design . . . . .	91
4.8	Conclusion . . . . .	95
<b>III Real-time Experiments and Numerical Simulations</b>		<b>97</b>
<b>5</b>	<b>Experimental Results on SPIDER4 RA-PKM</b>	<b>99</b>
5.1	Introduction . . . . .	99
5.2	Description of the Experimental Setup, and Implementation Issues . . . . .	100
5.2.1	Description of Experimental Setup . . . . .	100
5.2.2	Reference Trajectory Generation . . . . .	101
5.2.3	Performance evaluation criteria . . . . .	108
5.2.4	Tuning Gains procedure . . . . .	108
5.3	Experimental results of control solution 1 . . . . .	109
5.4	Experimental results of control solution 2 . . . . .	112
5.4.1	Trajectory evaluation at low speed . . . . .	113
5.4.2	Trajectory evaluation at medium speed . . . . .	113
5.4.3	Trajectory evaluation at high speed . . . . .	117
5.5	Conclusion . . . . .	120
<b>6</b>	<b>Simulation and Experimental Results on Delta PKM</b>	<b>125</b>
6.1	Introduction . . . . .	125
6.2	Description of the Experimental Setup, and Implementation Issues . . . . .	126
6.2.1	Description of experimental Setup . . . . .	126
6.2.2	Trajectory Generation for Real-time Experiments . . . . .	128

---

6.2.3	Trajectory Generation for Numerical Simulations . . . . .	129
6.2.4	Performance evaluation criteria . . . . .	133
6.2.5	Tuning gains procedure . . . . .	133
6.3	Experimental Results with a PID Controller . . . . .	134
6.4	Simulation results of control solution 3 . . . . .	137
6.4.1	Scenario 1: Nominal case . . . . .	137
6.4.2	Scenario 2: Changes in the payload . . . . .	137
6.4.3	Scenario 3: Changes in the speed . . . . .	139
6.4.4	Comparison of BSNN compensation against nominal feedforward . . .	146
6.5	Conclusion . . . . .	148
<b>General Conclusion</b>		<b>149</b>
<b>A Proof of lemma 1</b>		<b>153</b>
<b>B Trajectory points for SPIDER4</b>		
<b>Real-time experiments</b>		<b>155</b>
B.1	Trajectory points for Scenario 1 . . . . .	155
B.2	Trajectory points for scenario 2 . . . . .	156
<b>References</b>		<b>168</b>

---

## List of Figures

1.1	Exemplification of a Serial manipulator with its kinematic configuration. . . . .	5
1.2	Exemplification of a PKM with its kinematic configuration. . . . .	5
1.3	Hybrid manipulator with serial carrier and parallel wrist . . . . .	8
1.4	Hybrid manipulator with parallel carrier and serial wrist . . . . .	9
1.5	Spatial Platform Patented by James E. Gwinnett [Gwinnett, 1931] . . . . .	9
1.6	Willard LV Pollard Parallel mechanism [Pollard, 1940] . . . . .	10
1.7	Gough Platform [Gough and Whitehall, 1962] . . . . .	11
1.8	Stewart Platform . . . . .	11
1.9	Flight simulator of Klaus Cappel [Cappel, 1971] . . . . .	12
1.10	Delta PKM of R. Clavel. Patent No. 4,976,582 [Clavel, 1990] . . . . .	13
1.11	Timeline of PKM based on the Delta Robot . . . . .	15
1.12	Examples of PKMs for P&P Tasks . . . . .	16
1.13	Examples of PKMs for Machining Operations . . . . .	16
1.14	Examples of PKMs for coordinate measuring . . . . .	17
1.15	Examples of PKMs for Medical Applications . . . . .	18
1.16	EcoRobotics system developed for agriculture applications . . . . .	18
1.17	Examples of Motion Simulators . . . . .	19
1.18	Examples of 3D printers based on PKMs . . . . .	20
1.19	Examples of haptic devices based on PKMs . . . . .	21
1.20	Illustration of the concept of actuation redundancy in PKMs . . . . .	23
2.1	General kinematic representation of a PKM . . . . .	29
2.2	Illustration of modeling simplification for PKMs with delta-like architecture [Nabat, 2007] . . . . .	33
2.3	Classification of implemented control schemes on Parallel Kinematic Machines	36
2.4	Block diagram comparison of standard CTC (blue) and ACTC (green) . . . . .	43
2.5	Block diagram of L1 adaptive controller for PKM (blue), plus adaptive feedforward compensation (green) . . . . .	45
3.1	The SPIDER4 RA-PKM and its kinematics . . . . .	51


3.2	Enclosure dimensions of the working area of SPIDER4 . . . . .	51
3.3	CAD View of SPIDER4 with its main components description . . . . .	52
3.4	Kinematics illustration of the SPIDER4 RA-PKM . . . . .	53
3.5	Illustration of the four virtual spheres intersection used to find ${}^o\mathbf{N}_N$ position	56
3.6	Illustration of dynamic model simplifications applied to SPIDER4 RA-PKM	59
3.7	Details of the SPIDER4 wrist . . . . .	61
3.8	The Delta PKM and its kinematic configuration . . . . .	64
3.9	Kinematics illustration of the Delta PKM . . . . .	65
3.10	Illustration of the three virtual spheres intersection used to find ${}^o\mathbf{X}$ position	67
4.1	Sketch of a biological neuron . . . . .	78
4.2	Standard artificial neuron model . . . . .	79
4.3	Common activation functions for ANNs . . . . .	80
4.4	Block-diagram of the proposed control solution 1 for the parallel device of SPIDER4 RA-PKM . . . . .	90
4.5	Block-diagram of the proposed control solution 2 for the parallel device of SPIDER4 RA-PKM . . . . .	91
4.6	Representation of the proposed control solution 3 for Delta PKM . . . . .	92
4.7	Distribution of the proposed activation functions of order 3 for the respective inputs . . . . .	93
4.8	Diagram of the proposed BSNN used as a compensation term for each kinematic chain of the Delta PKM . . . . .	94
5.1	Overview of SPIDER4 RA-PKM in its work environment . . . . .	101
5.2	Illustration of experimental setup of the SPIDER4 . . . . .	102
5.3	Interpolation examples using G1 and G2 instructions . . . . .	104
5.4	Evolution of the desired trajectories in Cartesian Space versus time . . . . .	104
5.5	Desired trajectory of the nacelle in Cartesian space versus time . . . . .	105
5.6	Scheme of desired piece with dimensions in mm and degrees . . . . .	106
5.7	Desired trajectory for the machining task . . . . .	106
5.8	Evolution of the desired trajectories versus time for different values of F . . .	107
5.9	Evolution of the tracking errors in Cartesian Space versus time . . . . .	110
5.10	Evolution of the tracking errors in joint space versus time . . . . .	110
5.11	Evolution of the torques generated by RISE controller with nominal feedforward versus time . . . . .	111
5.12	Evolution of the different performance indexes for Cartesian and joint space with respect to time . . . . .	112
5.13	Evolution of tracking errors in joint space respect to time for F=1200 mm/min	114
5.14	Evolution of tracking errors in joint space respect to time for F=1200 mm/min zoomed view . . . . .	114
5.15	Evolution of Ks gains respect to time for F=1200 mm/min . . . . .	115
5.16	Evolution of $\Lambda$ gains respect to time for F=1200 mm/min . . . . .	115
5.17	Evolution of the control inputs generated by RISE FF AG controller respect to time for F=1200 mm/min . . . . .	116

5.18	Evolution of tracking errors in joint space respect to time for F=2400 mm/min	117
5.19	Evolution of tracking errors in joint space respect to time for F=2400 mm/min zoomed view . . . . .	118
5.20	Evolution of Ks gains respect to time for F=2400 mm/min . . . . .	118
5.21	Evolution of $\Lambda$ gains respect to time for F=2400 mm/min . . . . .	119
5.22	Evolution of the control inputs generated by RISE FF AG controller respect to time for F=2400 mm/min . . . . .	119
5.23	Evolution of tracking errors in joint space respect to time for F=24000 mm/min	120
5.24	Evolution of tracking errors in joint space respect to time for F=24000 mm/min zoomed view . . . . .	121
5.25	Evolution of Ks gains respect to time for F=24000 mm/min . . . . .	121
5.26	Evolution of $\Lambda$ gains respect to time for F=24000 mm/min . . . . .	122
5.27	Evolution of the control inputs generated by RISE FF AG controller respect to time for F=24000 mm/min . . . . .	122
5.28	Machining results using RISE FF AF with different forward speeds . . . . .	123
6.1	View of Delta PKM prototype of Polytechnic University of Tulancingo . . .	127
6.2	Encoders connection diagram . . . . .	127
6.3	Power stage diagram of one servomotor of Delta PKM prototype of Polytechnic University of Tulancingo . . . . .	128
6.4	Isometric View of trajectory used for Real-time experiments . . . . .	129
6.5	Desired trajectories with respect to time generated in LabVIEW . . . . .	130
6.6	Desired 3D trajectory for a P&P Task. The lines in red color corresponds to trajectory portions where the Delta PKM is moving with a payload and the blue lines are the corresponding portions without payload . . . . .	131
6.7	Evolution of the desired trajectories in Cartesian space versus time for scenarios 1 and 2 . . . . .	132
6.8	Desired spiral trajectory in the plane $(x,y)$ for the Scenario 3 . . . . .	133
6.9	Tracking trajectory comparison in Cartesian and joint Space with respect to time	135
6.10	Tracking error evolution in Cartesian and joint space with respect to time . .	136
6.11	Controller signals . . . . .	136
6.12	Evolution of the tracking errors versus time in Cartesian and joint space for scenario 1 . . . . .	138
6.13	Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time for scenario 1 . . . . .	138
6.14	Evolution of the BSNNs' weights of scenario 1 . . . . .	139
6.15	Evolution of the tracking errors versus time in Cartesian and joint scenario 2	140
6.16	Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time scenario 2 . . . . .	140
6.17	Evolution of the BSNNs' weights of scenario 2 . . . . .	141
6.18	Evolution of the tracking errors versus time in Cartesian and joint space corresponding to low speed for scenario 3 . . . . .	142



---

6.19	Evolution of the tracking errors versus time in Cartesian and joint space corresponding to medium speed for scenario 3 . . . . .	142
6.20	Evolution of the tracking errors versus time in Cartesian and joint space corresponding to high speed for scenario 3 . . . . .	143
6.21	Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time that corresponds to low speed scenario 3 . . . . .	143
6.22	Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time that corresponds to medium speed scenario 3 . . . . .	144
6.23	Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time that corresponds to medium high scenario 3 . . . . .	145
6.24	Evolution of the BSNNs' weights for scenario 3 when the Delta PKM is subjected to changes in the speed . . . . .	145
6.25	Degradation graphs of RMSE at different speeds for Cartesian and joint space in scenario 3 . . . . .	146
6.26	Performance comparison between RISE feedforward and RISE BSNN scenario 1	147
6.27	Performance comparison between RISE feedforward and RISE BSNN scenario 2	147



---

## List of Tables

1.1	Comparison Proprieties Between Serial and Parallel Manipulators . . . . .	7
3.1	Summary of the kinematic parameters of SPIDER4 . . . . .	55
3.2	Summary of the dynamic parameters of SPIDER4 . . . . .	63
3.3	Summary of the kinematic parameters of Delta PKM . . . . .	66
3.4	Summary of the dynamic parameters of Delta PKM . . . . .	70
4.1	Knot-points Vectors' distribution . . . . .	93
5.1	Common Alphanumeric Address Codes . . . . .	103
5.2	Controllers Gains . . . . .	109
5.3	Controllers Performance Evaluation . . . . .	111
5.4	Controllers performance Evaluation Using Performance Indexes . . . . .	112
5.5	Controllers performance evaluation using RMSE for joint space tracking errors (Deg) . . . . .	120
5.6	Controllers performance evaluation using RMSE for Cartesian space tracking errors (cm) . . . . .	123
5.7	Enhancement of the proposed RISE FW AG respect to standard RISE and RISE FW using the RMSE for tracking errors in Cartesian and joint space . . . . .	124
6.1	Controller parameters used in the Delta PKM . . . . .	134
6.2	Controllers performance evaluation for scenario 1 and 2 . . . . .	141
6.3	Controllers performance evaluation for scenario 3 . . . . .	144
B.1	List of commands used for trajectory generation for scenario 1 . . . . .	155
B.2	List of commands used for trajectory generation for scenario 2 . . . . .	156





---

## List of Acronyms

- ACTC** Adaptive Computed Torque Control
- ADRC** Active Disturbance Rejection Control
- AG** Adaptive Gains
- AM** Additive Manufacturing
- AMNN** Associative Memory Neural Network
- ANN** Artificial Neural Network
- BNN** Biological Neural Network
- BSNN** B-Spline Neural Network
- CAD** Computer Aided Design
- CMAC** Cerebellar Model Articulation Controller
- CNC** Computer Numerical Control
- CTC** Computed Torque Control
- DC** Direct Current
- DCAL** Desired Compensation Adaptive Control
- DOF** Degrees of Freedom
- EIA** Electronics Industry Association
- ESO** Extend State Observer
- FKM** Forward Kinematic Model

- HOSMC** Higher Order Sliding Mode Control
- IDC** Inverse Dynamics Control
- IDM** Inverse Dynamic Model
- IGBT** Insulated Gate Bipolar Transistor
- IKM** Inverse Kinematic Model
- ISAE** Integral of the Absolute Square Error
- ISIS** Intelligent Surgical Instruments & Systems
- ITAE** Integral Time multiplied by the Absolute Error
- ITASE** Integral Time multiplied by the Absolute Square Error
- LIRMM** Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
- PID** Proportional Integral Derivative
- PKM** Parallel Kinematic Machine, or Parallel Kinematic Manipulator
- P&P** Pick & Place
- RA** Redundantly Actuated
- RBF** Radial Basis Function
- RISE** Robust Integral of the Sign Error
- RMSE** Root Mean Square Error
- SMC** Sliding Mode Control
- TLO** Tool Length Offset
- TSMC** Terminal Sliding Mode Control
- VI** Virtual Instrument

## List of Variables

The terminology presented in this list includes the variables and constants used in this thesis contributions, except those presented in state of the art.

$O - x_o, y_o, z_o$	Fixed reference frame
$N - x_n, y_n, z_n$	Nacelle reference frame
$\mathbf{Q}, \dot{\mathbf{Q}}, \ddot{\mathbf{Q}}$	Generalized coordinates vector of SPIDER4 and its derivatives
$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$	Joint variables vector of Delta and SPIDER4 positioning device
$\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d$	Desired joint variables vector of Delta and SPIDER4 positioning device
$q_i$	$i^{th}$ actuated joint
$\boldsymbol{\xi} = [\phi \ \psi]^T$	Joint variables of the SPIDER4 wrist mechanism
$\mathbf{R}_Z, \mathbf{R}_Y$	Rotational matrices over $z$ and $y$ axes
$\alpha_i$ base	Angular position of the $i^{th}$ motor located on the fixed base
${}^o\mathbf{S}_S$	Spindle position vector
${}^o\mathbf{N}_N, {}^o\dot{\mathbf{N}}_N, {}^o\ddot{\mathbf{N}}_N$	Cartesian positions, velocities and accelerations of the nacelle with respect to the fixed reference frame
${}^o\mathbf{X}, {}^o\dot{\mathbf{X}}, {}^o\ddot{\mathbf{X}}$	Cartesian positions, velocities and accelerations of the traveling plate with respect to the fixed reference frame

---

${}^o\mathbf{A}_i$	Position of the motor of each rear-arm with respect to the fixed reference frame
${}^o\mathbf{A}_i - x_i, y_i, z_i$	Auxiliary reference frame attached at ${}^o\mathbf{A}_i$
${}^o\mathbf{B}_i$	Position equivalent to the end of each rear-arm
${}^o\mathbf{C}_i$	Position of the $i^{th}$ passive joint connecting each forearm to the traveling plate or nacelle
${}^o\mathbf{A}_{i-x_i, y_i, z_i}$	Auxiliary frame located at ${}^o\mathbf{A}_i$
${}^i\mathbf{x}_i, {}^i\mathbf{y}_i$	Auxiliary positioning vectors measured with respect to the auxiliary frame
$L$	Rear-arm length
$L_c$	Distance to center of mass of $L$
$l$	Forearm length
$R_b$	Fixed base radius
$R_p$	Radius of the traveling plate or nacelle
$S_{offset}$	Distance between $\psi$ and ${}^o\mathbf{S}_S$
$A_{offset}$	Distance between ${}^o\mathbf{N}_N$ and $\psi$
${}^o\mathbf{B}'_i$	Virtual spheres center position
${}^N\mathbf{N}_i$	Position vector of the passive joints of the nacelle with respect to the nacelle center
${}^X\mathbf{X}_i$	Position vector of the passive joints of the traveling plate with respect to the traveling plate center
${}^o\mathbf{N}_{aux}$	Total distance between the nacelle center and the spindle tip
$\mathbf{J}$	Jacobian matrix
$\mathbf{J}_x$	Cartesian Jacobian matrix
$\mathbf{J}_q$	Joint Jacobian matrix
$\mathbf{J}_m$	Inverse Jacobian matrix
$\mathbf{H}$	Pseudo-inverse Jacobian matrix
$\mathbf{\Gamma}$	Torque vector

---

$\Gamma^*$	Regularized torque vector
$\Gamma_{act}$	Produced torque vector due to actuators, rear-arms and a half mass of the forearms
$\Gamma_{na}$	Produced torque vector due to the mass of the nacelle and a half mass of the forearms
$\Gamma_{Gna}$	Produced torque vector due to the gravity acceleration over the nacelle and a half mass of the forearms
$\Gamma_{Fna}$	Produced torque vector due to the inertial forces over the nacelle and a half mass of the forearms
$\Gamma_{Gra}$	Produced torque vector due to the gravity acceleration over the rear-arms and a half mass of the forearms
$\Gamma_{Fra}$	Produced torque vector due to the inertial forces over the rear-arms and a half mass of the forearms
${}^N\mathbf{W}_W$	Kinematic vector of the wrist with respect to the nacelle center
$\mathbf{K}_w$	Kinetic energy of the wrist
$\mathbf{U}_w$	Potential energy of the wrist
$\Gamma_w$	Torque vector of the wrist actuators
$\Gamma_T$	Torque vector of the whole actuators of SPIDER4
$\Gamma_{ra}$	Rear-arms' torque vector
$\Gamma_{fa}$	Forearms' torque vector
$\Gamma_{tp}$	Traveling plate torque vector
$\mathbf{M}_t$	Total mass matrix of the nacelle
$\mathbf{I}_t$	Total inertia matrix of the SPIDER4 rear-arms
$\mathbf{M}_{fra}$	Matrix of moments produced by gravity on the rear-arms and a half the mass of the forearms
$\mathbf{I}_{act}$	Inertia matrix of the actuators
$\mathbf{I}_{ra}$	Inertia matrix of the rear-arms
$\mathbf{I}_{fa}$	Inertia matrix of the forearms
$\mathbf{M}_{ra}$	Mass matrix of the rear-arms



---

$\mathbf{M}_{fa}$	Mass matrix of the forearms
$\mathbf{M}_{tp}$	Mass matrix of the traveling plate
$\mathbf{M}_{nfa}$	Mass matrix of half the mass of the forearms
$g$	Gravity acceleration
$\mathbf{G}$	Gravity acceleration vector
$m_{na}$	Nacelle mass
$m_{tp}$	Traveling plate mass
$m_{namotors}$	Mass of the three motors located on the Nacelle
$m_{ra}$	Rear-arm mass
$m_{fa}$	Forearm mass
$I_{arm}$	Rear-arm inertia
$I_{act}$	Motor inertia
$m_s$	Mass of the spindle motor
$m_\psi$	Motor mass that regulates movement in $\psi$
$m_\phi$	Motor mass that regulates movement in $\phi$
$\mathbf{M}(\mathbf{q})$	Inertia matrix of SPIDER4 delta-like positioning device and Delta robot
$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$	Coriolis/Centripetal forces matrix of SPIDER4 delta-like positioning device and Delta robot
$\mathbf{G}(\mathbf{q})$	Gravity forces vector of of SPIDER4 delta-like positioning device and Delta robot
$\mathbf{M}_w(\boldsymbol{\xi})$	Inertia matrix of the wrist mechanism
$\mathbf{C}_w(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})$	Coriolis/Centripetal forces matrix of the wrist mechanism
$\mathbf{G}_w(\boldsymbol{\xi})$	Gravity forces vector of the wrist mechanism
$\mathbf{M}(\mathbf{Q})$	Inertia matrix of SPIDER4
$\mathbf{C}(\mathbf{Q}, \dot{\mathbf{Q}})$	Coriolis/Centripetal forces matrix of SPIDER4
$\mathbf{G}(\mathbf{Q})$	Gravity forces vector of of SPIDER4

---

$\mathbf{e}_1, \mathbf{e}_q$	Joint tracking errors
$\mathbf{e}_2, \mathbf{r}$	Combined joint tracking errors
$\mathbf{z}$	Vector containing the joint tracking errors and combined joint tracking errors
$\mathbf{X}(t)$	Vector of system states
$\mathbf{M}(\mathbf{X}), \mathbf{F}(\mathbf{X})$	Uncertain nonlinear functions
$\mathbf{U}(t)$	Vector of control inputs
$\mathbf{\Gamma}_{RISE}$	RISE control signals
$\mathbf{\Gamma}_{ARISE}$	Adaptive RISE control signals
$\mathbf{\Gamma}_{FW}$	Nominal feedforward
$\tilde{\mathbf{N}}$	Vector containing model uncertainties
$\mathbf{K}_s, \mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \boldsymbol{\beta}$	RISE feedback control gains
$\mathbf{K}_s(t), \mathbf{\Lambda}_2(t),$	Time-varying adaptive RISE feedback control gains control signals
$\mathbf{K}_2, \mathbf{K}_{sM},$	$\mathbf{K}_s(t)$ bounds
$\mathbf{K}_3, \mathbf{\Lambda}_{2M},$	$\mathbf{\Lambda}_2(t)$ bounds
$\mathbf{y},$	Dystem states evaluated in the Lyapunov candidate function
$\mathbf{V}(\mathbf{y}, t), \dot{\mathbf{V}}(\mathbf{y}, t)$	Lyapunov candidate function and its derivative
$\mathbf{R}_H$	Projector operator
$\hat{\sigma}_i$	$i^{th}$ BSNN estimation output
$\hat{\boldsymbol{\Sigma}}$	Vector containing all BSNN output estimations
$S_K^j(u)$	B-Spline function
$\mathbf{a}_i$	$i^{th}$ Basis B-Spline output
$\mathbf{w}_i$	$i^{th}$ Weight vector
$\gamma$	Learning rate
$RMSE_C$	RMSE in Cartesian space
$RMSE_J$	RMSE in Joint space

---

$ITAE_C, ITSAE_C, IASE_C$	Performance indexes in Cartesian space
$ITAE_J, ITSAE_J, IASE_J$	Performance indexes in Joint space
$\mathbf{K}_p, \mathbf{K}_d, \mathbf{K}_i$	PID control gains
$\mathbf{V}(t),$	Vector of control voltage
$K_a,$	torque constant
$R_a,$	Armature resistance



---

## Acknowledgments

I first want to thank God for the lives of my parents Rosalba and Martín Escorcia, since they have given me their sustain at all times. Without this, it would have been difficult for me to complete this project. I want to express my sincere thanks to my external advisor Dr. Ahmed Chemori for invaluable support during my time as a doctoral student; I thank him for his advice, his patience, sharing his knowledge and experiences, and for allowing me to carry out research stays at the LIRMM, which helped me have a much greater scope in my research and allowed me to grow both professionally and personally. I am grateful to my internal advisor, Dr. Hipólito Aguilar Sierra, for his dedication in advising me and giving me ideas to considerably enrich this work. This work has been the result of a close collaboration between the Polytechnic University of Tulancingo and the LIRMM, for which I have to thank many people that I met in these institutions who gave me their unconditional trust and support. On behalf of my university, I thank my career coordinator, Dr. Carina Toxqui Quilt, Dr. José Humberto Arroyo Núñez, and Dr. Alfonso Padilla Vivanco for the support they gave me during these four years at the university. I have had the good fortune to share a workspace within the university with Raúl Castro, Gloria León, Daniel Santillán, Laura Alvarado, Maricela Delgadillo, Mario Arreola; I thank my friends from the university for sharing pleasant moments during this stage of my life. During my stay in France, I met many people who helped me personally and in the workplace. I want to sincerely thank Pierre-Elie Herve from Tecnalía for his advice and patience when teaching me to operate the SPIDER4 robot. Their help was undoubtedly vital in conducting this research. I thank the LIRMM staff for allowing me to carry out two research stays, which were of great importance for obtaining the results present in this thesis. I want to thank my friends that I made during my visit in France: Josué and Alin Rivera, JR, Hussein Saied, Rihab, Afef, Fedy Forero, Maxence Blond, Vinayak, Auwal, and Némie Oberdisse for spending enjoyable moments inside and outside of work, and for giving me help and advice. Finally, I thank the National Council of Science and Technology of Mexico (CONACYT) for the financial support received during my doctoral studies.





---

## General Introduction

Until before the industrial revolution in the 18th century, certain activities such as manufacturing processes, agricultural production, and means of transportation, among others, were carried out entirely manually with rudimentary tools or with the help of animals. With the invention of the first machines, these processes became more and more streamlined, which led to greater productivity and improved product quality, reducing time and costs. An essential part of the development of all industries has been the automation of their processes. As its name indicates, automation aims to automatically carry out each process to increase the production and quality of products. An automatic system makes use of various mechanical, electrical, electronic, hydraulic, and pneumatic elements to be able to carry out a specific task. Robots manipulators are devices that integrate several of the elements already mentioned in automation systems. A robot manipulator can be defined as an autonomous or semiautonomous reprogrammable mechatronic device capable of being used in multiple tasks such as welding, packaging, assembly, and quality control. There are two main types of robot manipulator architectures: serial robots and parallel robots. Serial robots consist of an open kinematic chain powered by individual actuators located on the joints. There exist multiple configurations of this type of manipulator; some of them are inspired by the human arm, such as the robotic arms with six degrees of freedom. These robots have several advantages, such as a large volume of work and great dexterity. However, this type of architecture is not very suitable for moving heavy loads at high speeds due to the high vibrations and bending produced, which results in a lack of precision. The other type of robot manipulators' are parallel robots, also known as Parallel Kinematic Machines (PKMs). Unlike the first type, these manipulators consist of multiple kinematic chains joined together in an element called the moving base or traveling plate. This kind of kinematic configuration allows the manipulator's final element to have more support points, which translates into greater stiffness, high speed capabilities, and improved precision compared to the serial ones. Thanks to these advantages, PKMs are gaining more space in applications where precision and high speed are of vital importance. Nevertheless, the multiple kinematic chains also restrict the manipulator's reach, which means that the workspace is considerably less than that of serial robots. Until now, PKMs have been used in tasks such as Pick and Place (P&P), remote surgery, precise positioning, and even high-speed machining. However, PKMs are known for their complex

nonlinear dynamics, the abundance of parametric and nonparametric uncertainties, and in some cases, the redundant actuation. These are considered as challenges when proposing control laws that efficiently regulate their positioning. Due to the dynamic complexity of parallel robots, obtaining a dynamic model using classical methods will result in multiple equations that require many computational resources for their resolution. Dynamic models with multiple and complex equations are inadequate to be implemented in advanced control schemes, so in recent decades methodologies have been proposed to obtain simpler dynamic models suitable for real-time implementations. Inevitably, by having a simplified dynamic model, the presence of structured uncertainties will become more noticeable. These kinds of uncertainties are due to unknown dynamic parameters, disturbances, modeling simplifications, and parameter variation over time. Considering the problems described above, it is necessary to design advanced control schemes that guarantee optimal tracking trajectory under the desired speed requirements, and with the best possible precision even in the presence of some or all of the issues described before. This research's goals can be summarized in the development of the kinematic and dynamic models of a new PKM prototype designed to perform machining operations and the proposal of multiple control schemes that improve the performance of parallel robots under various operating conditions. The mathematical models of the prototype, as well as the proposed controllers, have been validated through real-time experiments and numerical simulations. The results obtained support the validity of the proposals made in this thesis.

## Publications of the author

### Published in international journals

- [IJ1] **J.M. Escorcía-Hernández**, H. Aguilar-Sierra, O. Aguilar-Mejía, A. Chemori, and J.H. Arroyo-Nuñez “A New Adaptive RISE Feedforward Approach based on Associative Memory Neural Networks for the Control of PKMs”, *Journal of Intelligent & Robotic Systems*, 2020.
  
- [IJ2] **J.M. Escorcía-Hernández**, A. Chemori, H. Aguilar-Sierra, and J.A. Monroy-Anieva, “A New Solution for Machining with RA-PKMs: Modelling, Control and Experiments”, *Mechanism and Machine Theory*, volume 150, 2020, 103864.

### Published in international conferences

- [IC1] **J. M. Escorcía-Hernandez**, H. Aguilar-Sierra, O. Aguilar-Mejía, A. Chemori, and J. H. Arroyo-Nuñez, “An Intelligent Compensation Trough B-Spline Neural Network for a Delta Parallel Robot”, *The 6th International Conference on Control, Decision and Information Technologies - CoDiT'19*, Paris, France, 2019.

## Submitted to international journals

- [SJ1] **J.M. Escorcía-Hernández**, A. Chemori, and H. Aguilar-Sierra, “A Novel Adaptive Gains RISE Controller For RA-PKMs: Design and Real-Time Experiments”, *Robotics and Computer-Integrated Manufacturing*, **under review**.

## Organization of the thesis

The contents of this thesis are organized into three parts in the following form:

**Part 1** is formed by two chapters dedicated to introducing the reader to the context and state of the art on dynamics and control of PKMs.

- **Chapter 1** presents a general context of robotics emphasizing parallel robots, starting from their historical development to their potential applications. Finally, this chapter exposes in detail the main objectives of the thesis, together with the research contributions to the subject of PKMs.
- **Chapter 2** describes the methodologies reported in the literature to compute the dynamic model of PKMs. The chapter also provides a survey and classification of existing control strategies implemented to regulate the motion of parallel robots.

**Part 2** integrates the thesis contributions as the modeling of a new PKM prototype and the proposed control solutions. The contributions are organized in the following two chapters.

- **Chapter 3** is dedicated to describing and developing kinematic and dynamic models for two experimental platforms. Such experimental platforms are classified as non-redundant and redundant PKMs.
- **Chapter 4** establishes the control solutions designed to deal with specific tasks developed by parallel robots. The design procedure for these control solutions is described in detail. Moreover, the chapter includes a method to solve the problem of internal forces generation of redundantly actuated PKM.

**Part 3** addresses the validation of the proposed dynamic models and control schemes.

- **Chapter 5** presents the obtained experimental results of the redundant parallel robot applying the proposed models and control solutions. The results are summarized in tables and graphs.
- **Chapter 6** details the acquired results of one of the proposed control solutions for the non-redundant PKMs validated via numerical simulations.

Lastly, all the achievements made during this research are summarized in the general conclusion, where some perspectives on future work are also presented.



# Part I

## Context, Problem Formulation and State of the Art



---

# Context and Problem Formulation

*The use of Parallel Kinematic Machines in different kinds of industries has been increased considerably in the last decades due to the advantages obtained by their closed-loop construction in terms of higher stiffness, lower inertia, and structural stability. This chapter introduces the fundamental aspects related to them, as well as their control problem. The objectives of this thesis, together with the main contributions of this research, are also addressed in this chapter.*

## Contents

---

<b>1.1</b>	<b>Introduction . . . . .</b>	<b>3</b>
<b>1.2</b>	<b>Classification of Robotic Manipulators . . . . .</b>	<b>4</b>
<b>1.3</b>	<b>Overview of Parallel Kinematic Machines . . . . .</b>	<b>8</b>
<b>1.4</b>	<b>Control Problem Formulation . . . . .</b>	<b>20</b>
<b>1.5</b>	<b>Objectives of the Thesis . . . . .</b>	<b>22</b>
<b>1.6</b>	<b>Main Contributions of the Thesis . . . . .</b>	<b>23</b>
<b>1.7</b>	<b>Conclusion . . . . .</b>	<b>25</b>

---

## 1.1 Introduction

Robotics are mechatronic systems formed by mechanical, electrical, and electronic elements controlled by commands from a computer. Nowadays, they are presented in seas, the sky, schools, factories, hospitals, and homes doing productive or leisure activities. According to The Robotics Institute of America *A robot is a re-programmable multi-functional manipulator designed to move materials, parts tools, or specialized devices through variable programmed motions for the performance of a variety of tasks* [Tsai, 1999]. Robotic Manipulators are a branch of robotics that carry out tasks developed by human arms to handle and position objects.

Currently, robotic manipulators play a decisive role in industrial automation, preserving quantity and quality in production lines. They can be reconfigured to perform different automated tasks, such as spray painting, welding, material handling, or component assembling.

## 1.2 Classification of Robotic Manipulators

In several works reported in the literature, robotics manipulators are classified according to their kinematic configuration. There are three kinematic configurations for robotic manipulators, which are serial, parallel, and hybrid. Each kinematic architecture has certain advantages over the others in their mechanical construction that makes them attractive for carrying out specific tasks; let us expound each one of them.

### 1.2.1 Serial Manipulators

Serial robots are the most extended manipulators in industrial applications. They have an open-loop kinematic chain formed by several links connected to each other by prismatic or rotational joints. Usually, serial manipulators have actuators in all joints. The end-effector in a Serial Manipulator is located at the end of the kinematic chain. One example of a serial manipulator is the robotic arm imitating the human arm parts (shoulder, elbow, and wrist). Figure 1.1a shows a six Degrees of Freedom (DOF) serial robotic arm; its kinematic configuration is illustrated in Figure 1.1b, where the gray boxes with a letter R denote the active rotational joints.

### 1.2.2 Parallel Kinematic Manipulators

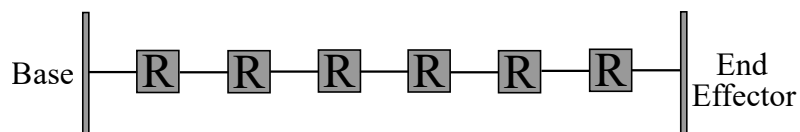
Parallel Kinematic Manipulators (PKM), also known as Parallel Robots or Parallel Kinematic Machines, are robotic devices conceived as a solution for the limitations presented in Serial Manipulators in terms of stiffness and precision at high accelerations. According to [Merlet, 2006], *a generalized Parallel Manipulator is a closed kinematic loop mechanism, whose end-effector is connected to the base through at least two independent kinematic chains*. Figure 1.2a shows a 6-DOF Hexapod being a typical example of a PKM. The manipulator's kinematic configuration is appreciated in Figure 1.2b, where the gray boxes with a letter P represent active prismatic joints, whereas the white boxes with a letter U symbolize passive universal joints.

### 1.2.3 Serial versus Parallel Manipulators

Serial manipulators and PKM have advantages and disadvantages due to their kinematic construction. These dates are detailed in this subsection [Krut, 2003].



(a) Robotic arm MH225 From Yaskawa corporation

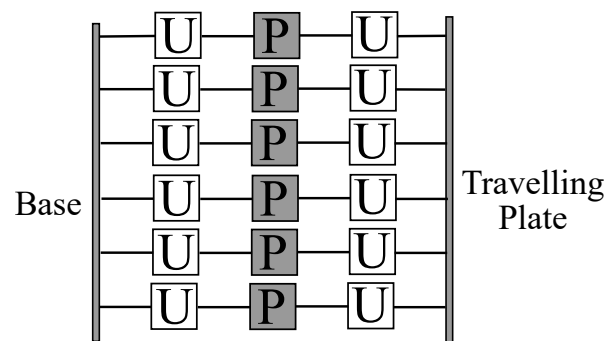


(b) Robotic arm kinematic chain diagram

Figure 1.1: Exemplification of a Serial manipulator with its kinematic configuration.



(a) Hexapod H-850 From PI Motion USA



(b) Hexapod chain diagram

Figure 1.2: Exemplification of a PKM with its kinematic configuration.

### Advantages of Serial Manipulators

The main desired features of serial robots are:

- Large workspace volume
- Good Dexterity proprieties

### Drawbacks of Serial Manipulators

In contrast to the above, the main inconveniences of Serial manipulators are listed below [Krut, 2003], [Germain, 2013].

- Low stiffness caused by their open-loop kinematic structure
- High inertia production due to the masses distributed over the entire kinematic chain (actuators, components transmission)
- Cumulative positioning errors due to the layout of the series segments.
- Fatigue and wear of the links ensuring the circulation of information between the sensors and the control. In terms of safety, this is essential since a transmission error can have disastrous consequences on the robot's movements.

### Advantages of Parallel Kinematic Manipulators

The closed-loop construction in PKMs entails the following advantages with respect to their serial counterparts [Germain, 2013], [Krut, 2003], [Merlet, 2006].

- High mechanical stiffness since the payload is taken up by all the kinematic chains
- High load capacity
- Low moving mass
- Possibility to perform high dynamic movements (high accelerations)
- They have the possibility of positioning the actuators directly on the fixed base or very close to it; this feature has the following positive consequences:
  - Wide choice of motors and gear-heads since their mass has no influence on the inertia of the moving robot.
  - Significant simplification of connection problems between motors, sensors and controller (simpler and more reliable wiring).
  - Easy cooling of the actuators, and therefore reduction of precision problems due to expansion and high potential power.

Table 1.1: Comparison Proprieties Between Serial and Parallel Manipulators

Proprieties	Serial Manipulator	Parallel Manipulator
Stiffens	Low	High
Workspace	Large	Small
Actuator's Location	In the Joints	In the base
Inertia forces and stiffness	High and less respectively	Less and high respectively
Payload/weight ratio	Low	High
Inverse kinematics	Difficult and complex	Straightforward and unique
Forward Kinematics	Straightforward and unique	Difficult and complex
Singularities	Inverse kinematic	Inverse, forward and combined

### Drawbacks of Parallel Kinematic Manipulators

Compared to serial robots, the disadvantages of parallel robots can be summarized as follows:

- Limited working volume compared to the total volume of the mechanism
- Forward Kinematic Models (FKM) sometimes are difficult to determine.
- Presence of singularities which lead to a loss of control of the mobile structure, or even to a deterioration of the mechanics. This is the most critical point when designing a machine with parallel architecture.
- The use of a large number of passive links induces wobbles, making the behavior of the robot difficult to model. However, they must be mastered to define dangerous zones near the singularities and to improve the precision of the robot.

In accordance with [Bennehar, 2015], [Saied, 2019], and the points mentioned above, one can summarize the following proprieties of Serial and Parallel Manipulators in Table 1.1.

#### 1.2.4 Hybrid Manipulators

A Hybrid Manipulator can be defined as a juxtaposition of two mechanisms where one of them is formed a Serial structure, and the other one is Parallel. This kind of architecture can be divided into two branches:

- Serial positioning device with parallel wrist: This configuration associates a serial structure with a parallel structure in the following form. The serial mechanism is responsible for positioning, while the parallel mechanism is used for the end-element orientation. In this category of hybrid manipulators, we can mention the Sprint Z3 of DS Technology developed for aerospace applications [Chen et al., 2014]. The carrier mechanism uses two prismatic actuators to move the wrist on the plane  $x,y$ . The last one provides two rotational motions and one translational along  $z$ ; the manipulator is shown in Figure 1.3.

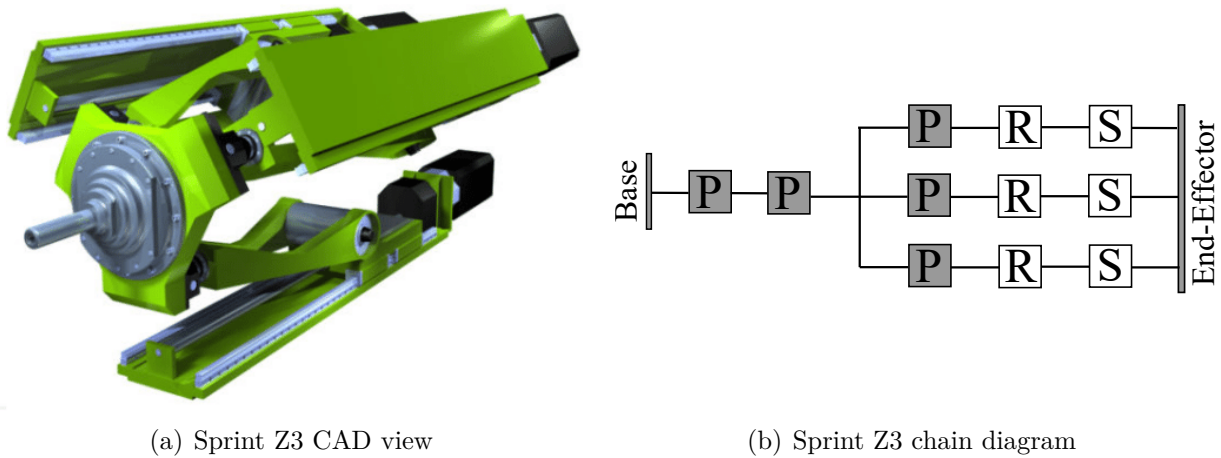


Figure 1.3: Hybrid manipulator with serial carrier and parallel wrist

- Parallel positioning device with serial wrist: In this case, the carrier mechanism is the Parallel Manipulator structure coupled with a Serial wrist responsible to orient the end-element. An example of this configuration is the Tricept 845 having 3T-2R DOF (see Figure 1.4) ; where three Universal-Prismatic-Spherical kinematic chains perform the three translational movements as long as a serial wrist mechanism performs the two rotational movements; the manipulator utilizes a passive kinematic chain (the U-P chain) to constrain the platform of the parallel structure [Neumann, 2006].

## 1.3 Overview of Parallel Kinematic Machines

Having explained the concepts related to robotic manipulators in a general form. Let us now focus on the subject of PKMs, starting with their historical development and continuing with their potential applications.

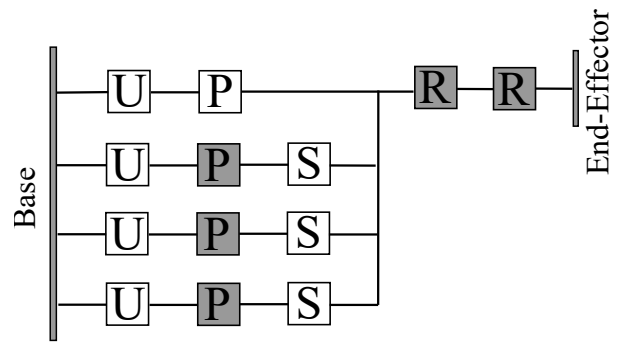
### 1.3.1 Historical review of PKMs

According to [Bonev, 2003], in the twentieth century, James E. Gwinnett designed a moving platform for the entertainment industry. The patent for this system was requested in 1928. Unfortunately, the designed mechanism was never built [Gwinnett, 1931] (see Figure 1.5). After in 1938, Willard LV Pollard designed the first industrial parallel robot, intended for spray painting operations. The design consists of a system of five degrees of freedom, composed of three link sets. Each set is formed by a proximal and a distal arm joined together through universal joints. Three actuators mounted on the base commanded the tool's position, while its orientation is provided by the other two actuators located on the base transmitting the movement to the tool using flexible cables, the parallel robot of Willard LV Pollard is shown on Figure 1.6 [Pollard, 1940]. In the year of 1947, Dr. Eric Gough designed the first octahedral hexapod type platform with variable length sides. Dr. Gough called it the universal testing





(a) Tricept general view



(b) Tricept chain diagram

Figure 1.4: Hybrid manipulator with parallel carrier and serial wrist

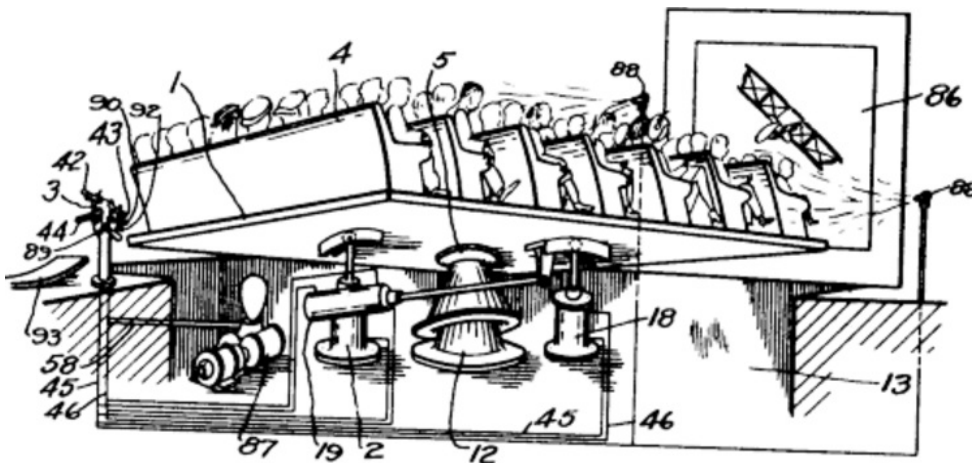


Figure 1.5: Spatial Platform Patented by James E. Gwinnett [Gwinnett, 1931]

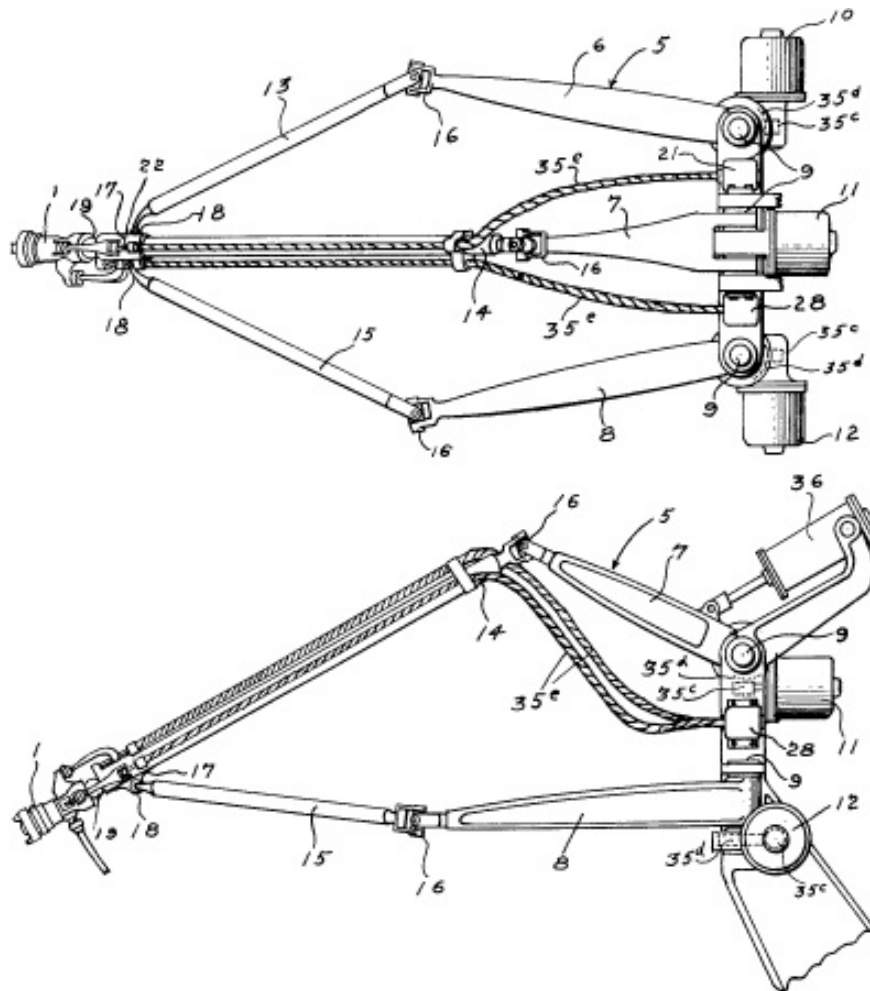


Figure 1.6: Willard LV Pollard Parallel mechanism [Pollard, 1940]



Figure 1.7: Gough Platform [Gough and Whitehall, 1962]

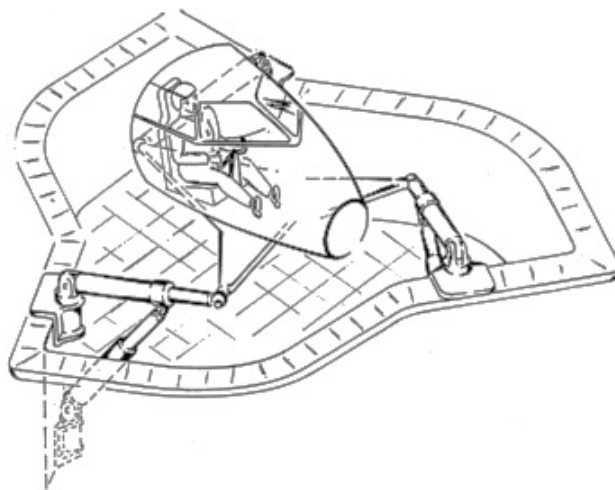


Figure 1.8: Stewart Platform

machine or universal platform. This system was invented to respond to the problems of air-landing loads; that is, the Gough platform tried to simulate an airplane's landing process. This machine was used to check the tires of the Dunlop house under loads applied on different axes. The Gough platform has been one of the mechanisms that have achieved greater recognition in parallel robotics. Figure presents 1.7 an example of the Gough platform [Gough and Whitehall, 1962]. In 1965 Stewart published an article in which he designs a 6-DOF platform intended to be used as a flight simulator. His design was different for the proposed octahedral hexapod by Gough. Ironically, the Gough platform is often referred to as the Stewart platform. Figure 1.8 depicts a sketch of the Stewart platform. [Stewart, 1965]. Nevertheless, Stewart is not considered the creator of the first flight simulators based-on PKM. Indeed, in the '60s, the US engineer Klaus Cappel designed and built the first functional flight simulator. His design was based on an octahedral hexapod that reached the same

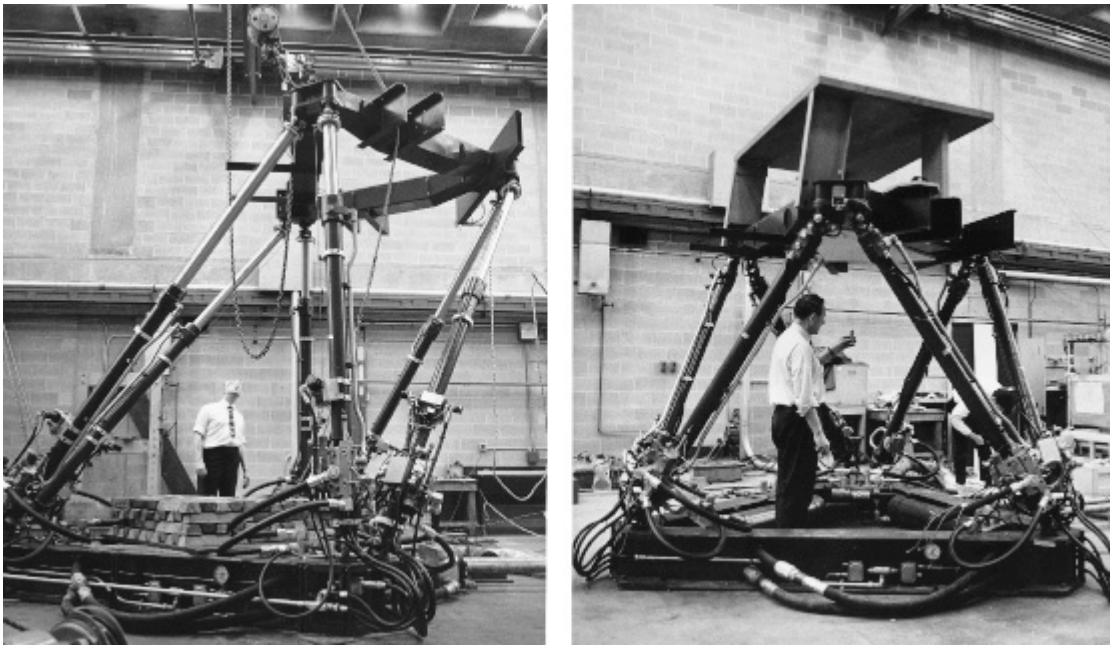


Figure 1.9: Flight simulator of Klaus Cappel [Cappel, 1971]

kinematic arrangement as the one proposed years ago by Dr. Gough. The Cappel patent was filed in 1964; however, he was unaware of the invention of Dr. Gough and Stewart's paper, which was not yet published. Figure 1.9 shows the first flight simulator built by Ing. Klaus Cappel [Cappel, 1971]. About twenty years passed without any significant progress in the subject of PKMs until, in the '80s, when Raymond Clavel, professor of the École Polytechnique Fédérale de Lausanne (EPFL), introduces the 3T-1R-DOF Delta PKM designed for P&P tasks [Clavel, 1990]. The Delta robot is mainly composed of three Revolute-Spherical-Spherical (RSS) kinematic chains making use of mechanisms based on parallelograms restraining the orientation of the traveling plate entirely, resulting in only translational movements, and an interdependent Rotational-Universal-Prismatic-Universal (RUPU) chain performs the rotational movement of the Delta robot. This PKM can reach high accelerations (Up to 50 G) thanks to its lightweight mechanism. The Delta is one of the most successful PKMs in history ever with many modifications around the world. In 1999 Dr. Clavel was awarded the "Gloden Robot Award" (sponsored by ABB) for his innovative creation. Figure 1.10 shows the original design of the Delta PKM [Taghirad, 2013].

### Variations of the Delta PKM along the time

Several modifications have been made to the original prototype of R. Clavel. In [Pierrot et al., 1991], F. Pierrot extend the concept of the Delta PKM from 3-DOF to 6-DOF, leading to the creation of the "Hexa" PKM. Six rotational actuators located on the fixed base perform the robot's movements, and unlike the Delta, the Hexa PKM can regulate the orientation of its traveling plate. In order to eliminate the RUPU chain presented on the Delta robot, some mechanical solutions based on articulated traveling plates have been proposed. The first

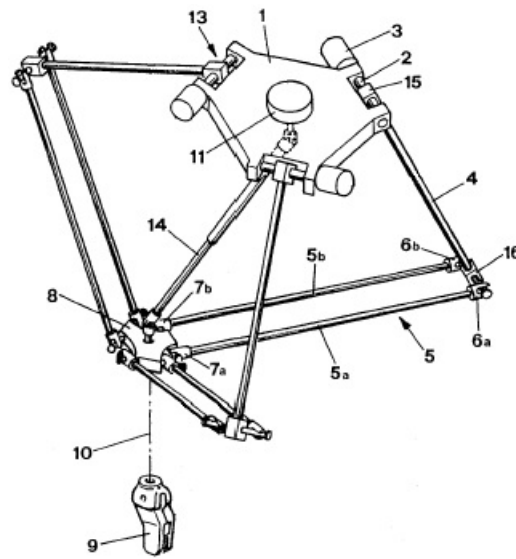


Figure 1.10: Delta PKM of R. Clavel. Patent No. 4,976,582 [Clavel, 1990]

prototype was called "H4," consisting of four kinematic chains connected to an articulate traveling plate composed of three pieces [Company and Pierrot, 1999]. After in [Krut, 2003], the traveling of H4 was customized, adding a gear mechanism to improve its performance. Notwithstanding, this PKM had some drawbacks such as the abundance of singularities and possible internal collisions, so a careful selection of the positions had to be necessary to avoid them. To overcome the issues presented on H4, a new family of PKMs called I4 was proposed in [Krut et al., 2003]. The main difference between I4 PKMs and H4 is the replacement of pivot joints by prismatic ones and gears by rack-and-pinion mechanisms. This family of PKMs had two variants I4L and I4R, being actuated by linear and rotational actuators, respectively. However, the drawback of an articulate traveling plate using prismatic joints is its short life if the PKM is frequently operated at High-speed. To overcome this limiting, in [Nabat et al., 2005], a new PKM called Par4 was developed with a new concept of articulated traveling plate consisting of four main parts linked by two rotational joints. Two mechanisms, one using a gear assembly and other with pulleys and belts, were proposed to amplified the produced rotational movement. Nowadays, Par4 is commercialized by Adept Technologies under Quattro's name, considered until now the fastest industrial PKM in the world [Bennehar, 2015]. A modified version of Par4 called Heli4 was presented in [Nabat, 2007]. The difference between this PKM and Par4 is the traveling plate composed of two platforms connected through a helical screw that produces the rotation around the vertical axis. Penta Robotics commercializes Heli4 under the name of Veloce [Shayya, 2015]. It has been proposed solutions to produce motions in a vertical plane with very high-speed for instance, in [Baradat et al., 2009], a 2-DOF PKMs named Par2 was proposed for very high-speed motion on the  $x, z$  plane. This PKM is formed by two sets of active and passive kinematic chains; these last ones have the function to restrict the movement in only one plane. It has been reported that this PKM is able to reach high accelerations up to 40G [Pierrot et al., 2009]. In [Germain et al., 2011], a novel 2-DOF PKM



concept called IRSBOT-2 was presented, constructing by only two active kinematic chains. Therefore, it is less subject to uncontrolled parasitic effects that may produce the passive kinematic chains of Par2. There have been reported in the literature other prominent PKMs which make use of a Redundantly Actuated (RA) configuration. They are R4 with 3-DOF and four actuators, destined for applications that requiring extreme accelerations [Corbel et al., 2010], SPIDER4, being the first delta-like PKM destined for machining operations, and T3KR. This last one PKM shares several similarities with Par4; however, the rotational movement is performed by an independent actuator located on the traveling plate. Figure 1.11 presents a timeline with all the examples presented in this part.

### 1.3.2 Applications Developed by Parallel Kinematic Machines

Thanks to their main features as improved stiffness, high dynamics, and potentially improved accuracy, PKMs have gained several places in industries where such features play a crucial role. Therefore, let us explain some potential applications developed by them.

#### P&P Tasks

The concept of P&P Place Tasks is basically to pick an object from an initial position and place it to another position. This action is essential in assembly lines or packaging production lines. Lightweight PKMs as the delta-like manipulators are the best candidates to develop high-speed P&P operations thanks to their improved stiffness and high dynamic response. The original Delta PKM developed by R. Clavel was conceived for this purpose used in a chocolate packaging line in a first instance and for manipulating very light objects. As was showed before, the original Delta robot has been modified to satisfy the greatest P&P exigences, such as high positioning precision under extreme high-speed demand, improved orientation capabilities, higher payload capacity, and an extended operational workspace. Figure 1.12 shows a food line assembly with Delta PKMs of the Swiss company Demaurex, and Adept Quattro being the fastest commercialized PKM.

#### Machining Operations

Machining is the broad term used to describe the removal of material from a workpiece and is one of the most critical manufacturing processes. Machining operations can be applied to metallic and non-metallic materials such as polymers, wood, ceramics, composites, and other materials. The most common machining operations are milling, turning, and drilling. These operations require high precision in positioning the cutting tool and the desired cutting path. Hence the machine tools should satisfy these requirements. PKMs, due to their structure conformed by the closed-loop kinematic chain mechanism, have some key advantages over their serial counterparts in terms of accuracy, stiffness, and moving masses and inertias. They are relatively more unaffected in mechanical construction than the existing conventional machining tools [Zhang, 2009] being thus more suitable for machining operations. However, the main drawbacks of PKMs lie in their small workspace and extensive singular configurations [Merlet, 2006] [Park and Kim, 1999]. Some notable examples of PKMs for

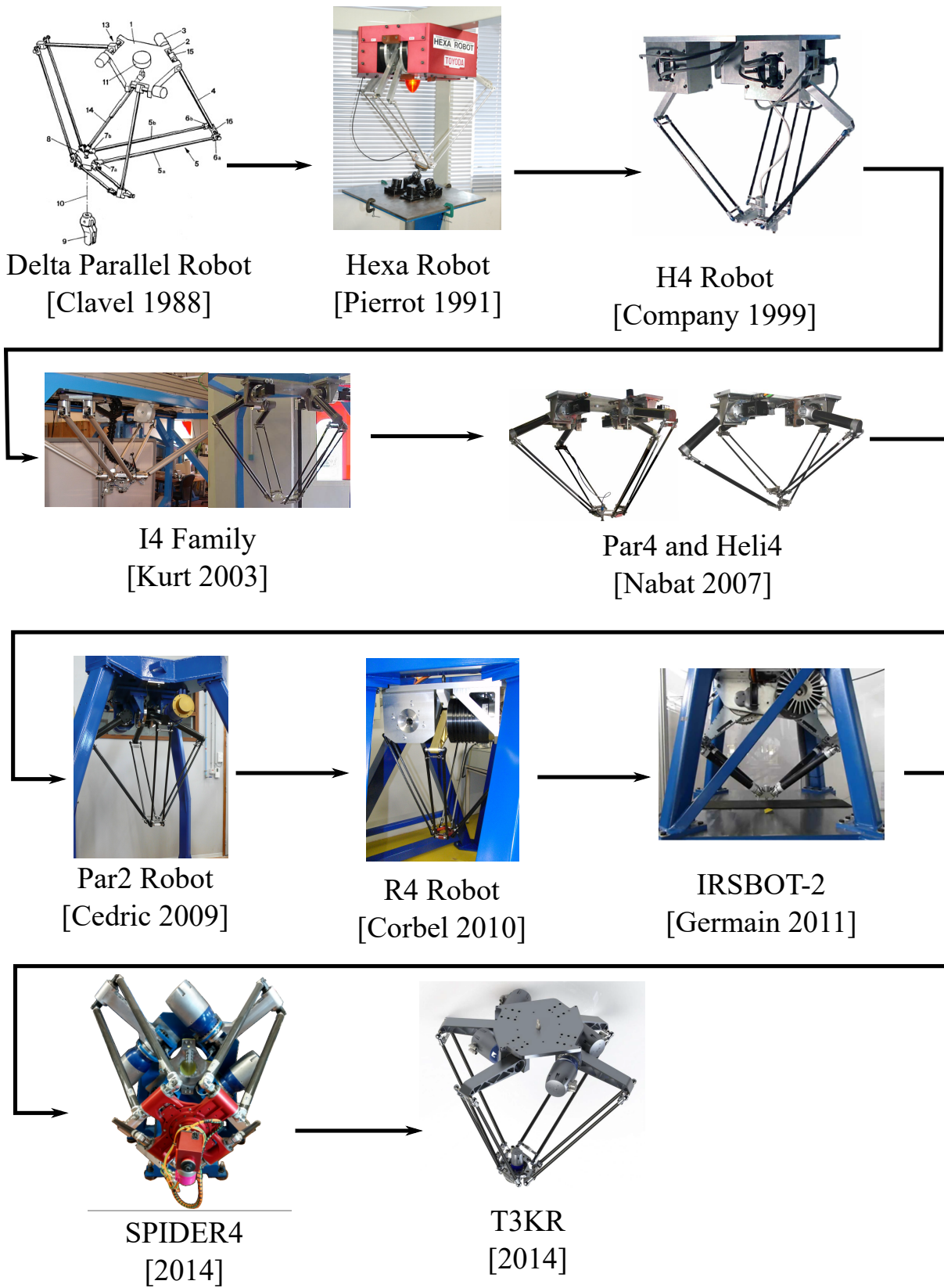


Figure 1.11: Timeline of PKM based on the Delta Robot

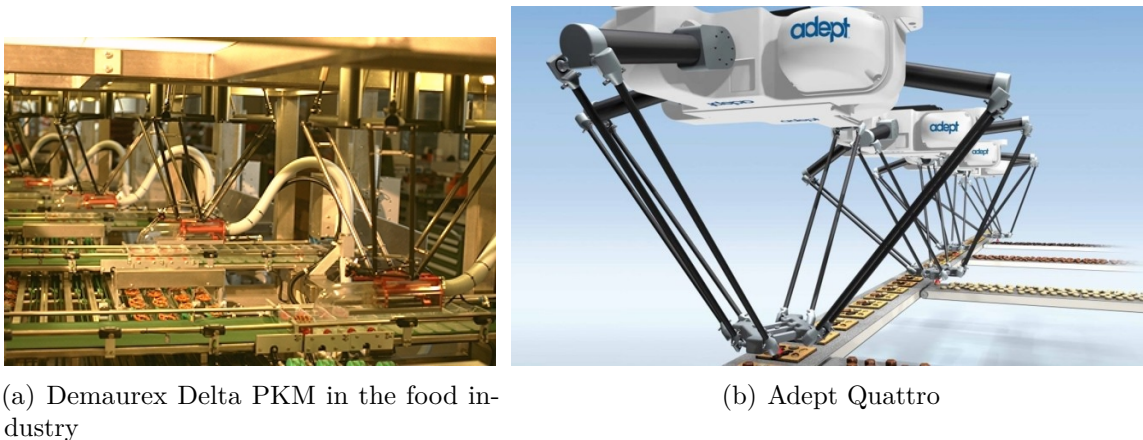


Figure 1.12: Examples of PKMs for P&amp;P Tasks

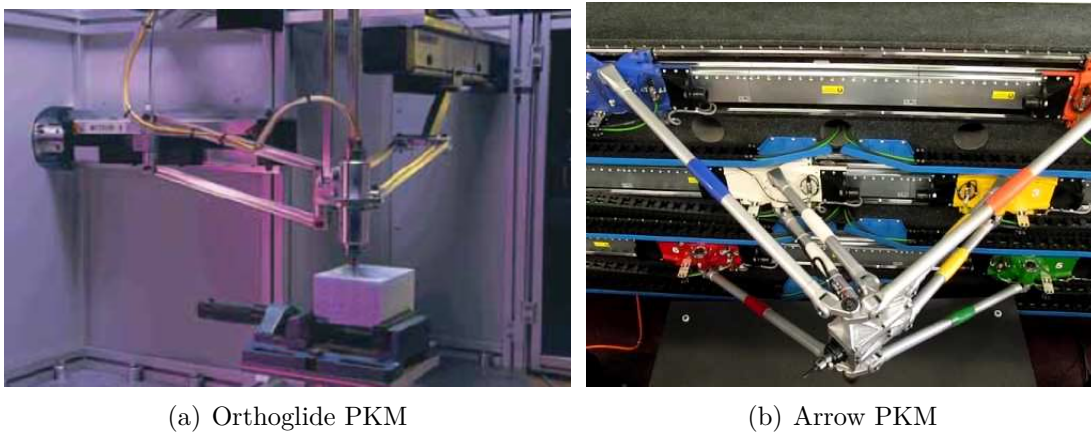


Figure 1.13: Examples of PKMs for Machining Operations

machining are: the Variax Hexacenter developed by Giddings & Levis company with 6-DOFs based on the Gough platform [Merlet, 2006], Orthoglide PKM, having 3 translational DOFs with a fixed orientation [Chablat and Wenger, 2007], ARROW PKM which has 5-DOFs (3T-2R) [Shayya, 2015], HexaM of Toyoda with 6-DOFs (3T-3R) [Pierrot and Shibukawa, 1999], P800/P200 from Metrom, Okuma Cosmo Center and The Octahedral Hexapod from Ingersoll with 5-DOFs [Merlet, 2006]. Figure 1.13 illustrates two of these examples.

### Coordinate Measuring Machines

Measuring is another potential application that PKMs can execute successfully due to the accuracy provide of their closed-loop kinematic construction, also increasing the structural stability of the machine. Traditionally, serial devices have performed this operation. However, errors caused by geometric offsets and deformations by load effects, and a serial structure's low stiffness may affect the precision of the measurements [Zhuang and Wang, 1997]. Some examples of Coordinate measuring machines are the KIM-75 form the Russian company Lopic



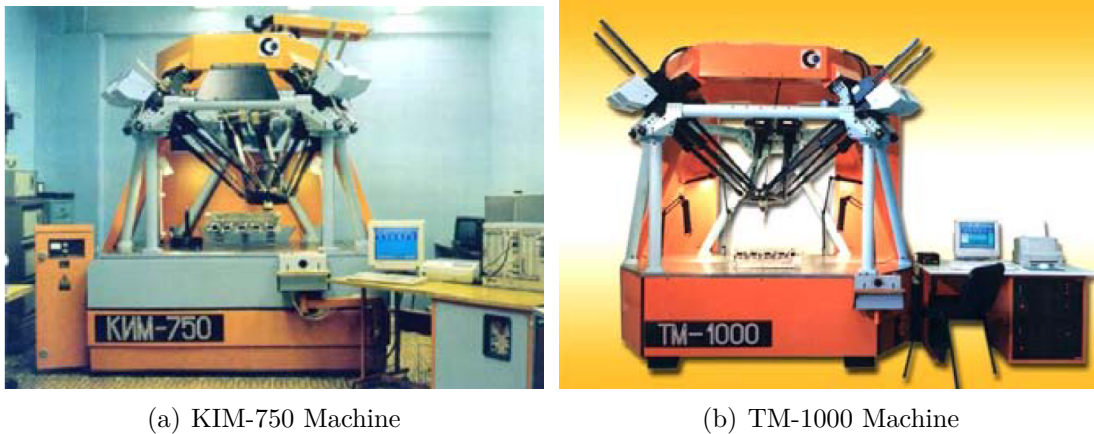


Figure 1.14: Examples of PKMs for coordinate measuring

(Figure 1.14a), or the TM 1000 machine (Figure 1.14b) [Angeles et al., 2005], [Merlet, 2006].

### Medical Applications

Thanks to the closed-loop architecture, some PKMs as Gough/Stewart platforms or delta-like manipulators have been integrated successfully in this area. PKMs present desired features that make them attractive solutions in the field of medicine. For instance, robotic-assisted surgery requires a device that takes up as little space as possible, guaranteeing precise positioning of the surgical instruments [Simaan, 1999]. In [Wapler et al., 2003], a 6-DOF hexapod was proposed as a universal platform for different types of surgical instruments, making emphasis on neuro endoscopy ensuring accuracy over  $20\mu m$ . The surgiscope is a tool-holder device developed by the Intelligent Surgical Instruments & Systems company (ISIS). This is a delta-like manipulator dedicated to microscope applications in neurosurgery and contains a microscope on the traveling plate whose mass is approximately 70 kg [Briot et al., 2007]. Moreover, the applications of the Delta PKM in the area of medicine are not only limited to the positioning of objects. In [Li and Xu, 2007], a delta-like manipulator was proposed for cardiopulmonary resuscitation (CPR), making chest compression in the patient. The mentioned examples for medical applications are shown in Figure 1.15.

### Agriculture Applications

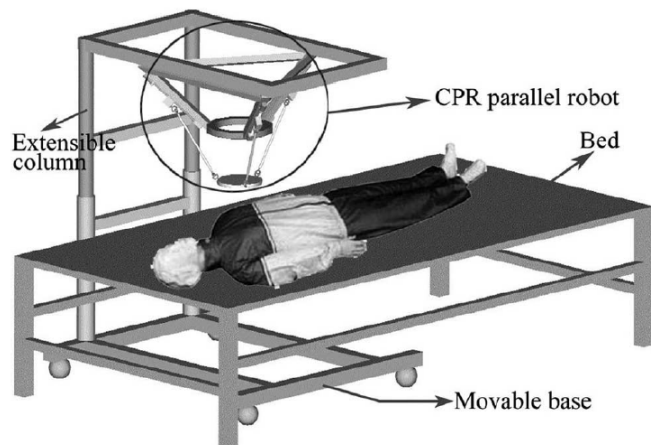
The agriculture industry is an important area where robotics are making headway due to the hard work that cultivating and preserving crops represents. Such operations make use of chemical or mechanical processes to destruct the weeds [Ovchinnikov et al., 2020]. EcoRobotix is an autonomous robotic system designing for weeding. It consists of a mobile vehicle with two Delta robots integrating a vision system (see Figure 1.16). The cameras identify weeds, and the Delta PKMs applied a microdose of herbicide to kill them without hurting any of the crops. The system is powered with solar energy, and the machine status can be consulted through an application for smartphones [EcoRobotix, 2011]. This invention helps considerably to reduce the use of herbicides and fertilizers, which can harm when used in excess.



(a) SurgiScope PKM from ISIS



(b) Surgical PKM from Physics Instruments



(c) Delta PKM applied for CPR

Figure 1.15: Examples of PKMs for Medical Applications



Figure 1.16: EcoRobotics system developed for agriculture applications

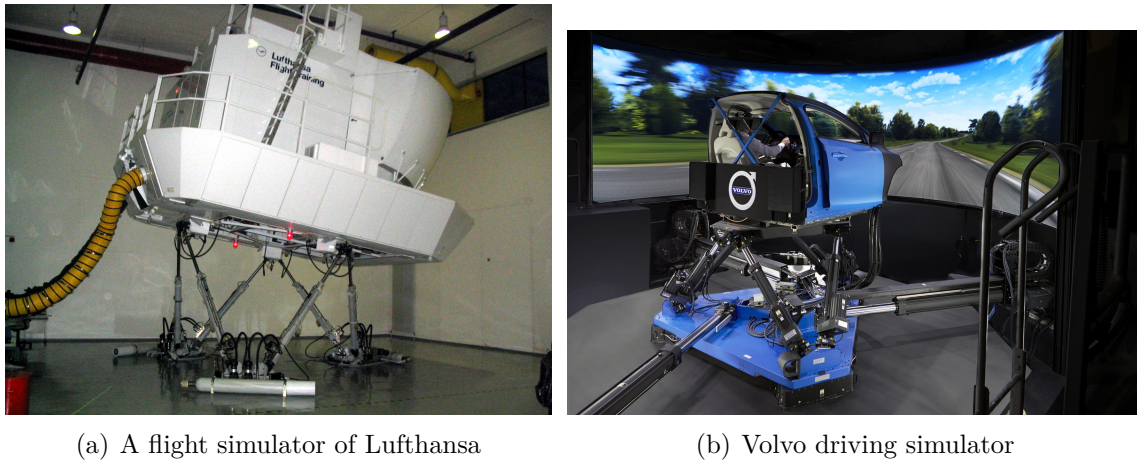


Figure 1.17: Examples of Motion Simulators

### Motion Simulators

Motion simulators are some of the most relevant applications developed by PKMs. Devices used for this purpose must require 6-DOF motions, enough capacity to support heavy loads, fast and accurate movements, and high stiffness. For these mentioned reasons, the majority of them are based on the Gough/Stewart's and Cappel's platform architecture. In the aerospace industry, they play an essential role in aircrafts' design to prevent fatal accidents [Asif, 2012]. In addition to being used as flight simulators, motion simulators have been used as car, ship, and space simulators [Merlet, 2006]. Figure 1.17 presents some examples of motion simulators.

### 3D Printers

Three-dimensional (3D) printing, also known as Additive Manufacturing (AM), is the construction of a three-dimensional object from a Computer-Aided Design (CAD) model. These devices build products adding the material layer by layer rather than by subtracting material from a piece of material as machining tools. In recent years, 3D printers have gained popularity since they facilities the manufacturing of low-cost pieces reducing wasted material. Nowadays, this technology is used to manufactured automobile components, aircraft components, Custom hearing aids, Custom orthodontics, among others [Campbell et al., 2011]. The Delta robot with linear actuators is widely used for 3D printing due to the parallelograms bars keeping the traveling plate perfectly horizontally, and the linear actuators provide more rigidity resulting in useful quality end-products [Campbell et al., 2011]. Figure 1.18 presents some commercial 3D printers based on the linear Delta PKM.

### Haptic Devices

Haptic is the science that deals with the study of the sense of touch. Haptic devices intended that the user can feel and interact indirectly with an external environment through physical manipulation of the device. These kinds of instruments are mainly present in virtual and



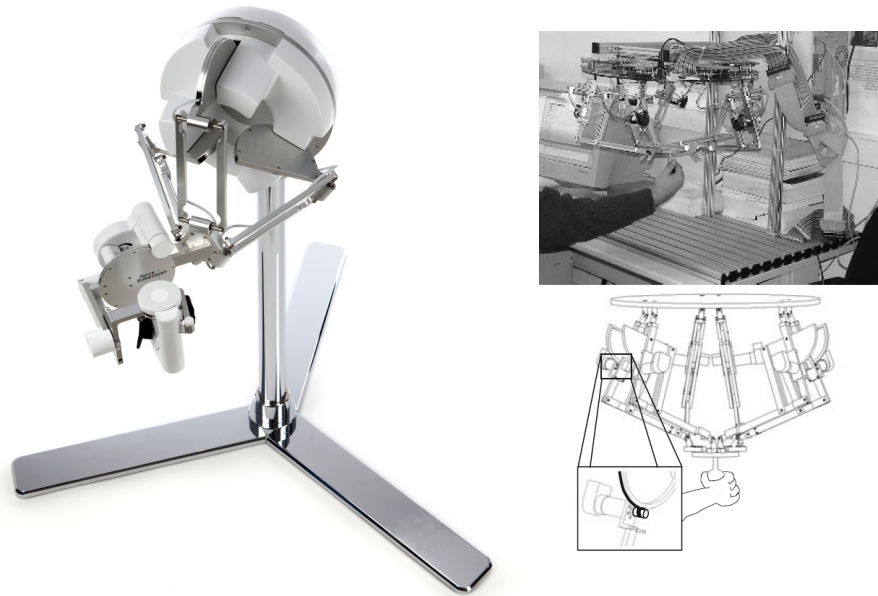
Figure 1.18: Examples of 3D printers based on PKMs

augmented reality, gaming controllers, and remotely operated surgery. According to [Birglen, 2003], some mechanical features such as low friction, low inertia, high dynamic range, and high stiffness are essential for high performance in haptic devices. PKMs are well-known for having those mentioned properties, so they are appropriate candidates to be used as sophisticated haptic interfaces. There have been proposed haptic devices based on the Gough/Stewart platform and the Delta robot. For instance, the sigma.7 model of the Swiss company Force dimension, which is a 7DOF haptic interface based on a 3-DOF Delta PKM [ForceDimension, 2010], or the prototype proposed in [Sabater et al., 2007], being a kind of 6DOF Gough/Stewart actuated through cable-driven pantographs instead of linear actuators. Figure 1.20 shows the mentioned examples.

## 1.4 Control Problem Formulation

Many factors converge in order to PKMs successfully develop the previously mentioned applications. These factors range from mechanical design, modeling to trajectory planning and control design. In the literature, it has been reported works relating to the enhancement of such aspects in order to increase the capabilities of Parallel robots. Control is a vital research topic in PKMs since control laws determine the generalized forces/torques that the actuators must develop in order to guarantee the execution of the commanded task while satisfying given transient and steady-state requirements [Siciliano et al., 2010]. Following this research line, the general purpose of this thesis is to propose and validate advanced control solutions to improve the overall performance of PKMs under different operating conditions.





(a) Sigma.7 from Force Dimension

(b) Prototype based on the Gough/Stewart platform

Figure 1.19: Examples of haptic devices based on PKMs

### 1.4.1 Challenges in Control for PKMs

Parallel robots are complex nonlinear systems, so controlling these kinds of systems results in a significantly challenging task. From the control point of view, the following considerations must be taken into account to design control schemes for PKMs. [Chemori, 2017].

#### Highly Nonlinear dynamics

The closed-kinematic-structure, together with the presence of several passive joints, make PKMs complex nonlinear systems. Their kinematic configuration leads to coupling dynamics, so the actuators' movements must be synchronized to each other not to harm the whole performance of the PKM. According to [Natal et al., 2015], the effect of nonlinearities in PKMs increases considerably at high speeds/accelerations operative conditions. In this context, high-accelerations yield to mechanical vibrations that may produce mechanical damages and decreasing the precision. Thus, the use of decentralized linear controllers does not guarantee the safety and stability of PKMs in such operative conditions. For these reasons, advanced nonlinear control techniques should be developed in order to minimize the nonlinearities' effects while keeping the demands of high precision under high-acceleration requirements.

### Uncertainties unstructured and structured

Uncertainties are some of the reasons why robots behave unexpectedly in the real environment concerning the expected behavior obtained through numerical simulations. As is explained in [Saied, 2019], and [Bennehar, 2015]. Uncertainties are the differences between a calculated dynamic model and the real system. They are classified as unstructured and structured. The first type includes geometric manufacturing errors, non-modeled phenomena, dynamic model simplifications (i.e., neglect the actuators' dynamics or friction), sensors' noise, and wear of mechanical elements. Whereas, the second type comprises inaccurate knowledge of dynamic parameters (e.g., masses and inertia), and dynamic parameters variations as the payload in a P&P application, or contact forces in machining operations. Advanced control strategies should be considered to deal with any uncertainty ensuring good dynamic performance and high precision in the developed task.

### Actuation Redundancy

In robotics, actuation redundancy means that the system has more actuators than DOF [Mueller, 2011]. Actuation redundancy has been used to improve kinematic capabilities in PKMs, adding one more kinematic chain with its respective actuator. This configuration presents the following advantages:

- Enhanced dexterity
- Singularity avoidance in the operational workspace
- Collision avoidance
- Backlash avoidance
- It improves the load distributions as long as reduces the actuators' power consumption
- Improved stiffness and therefore higher accuracy

However, actuation redundancy represents a significant challenge in the control of PKM, since it leads to the generation of internal forces that may create pre-stress in the mechanism without operational motions and can damage the robot's mechanical structure. According to [Hufnagel and Muller, 2012], the internal forces can be produced by non-synchronized independent control of the actuators, geometric imperfections, and measurement errors. In the literature, actuation redundancy solutions based on the pseudo-inverse Jacobian matrix have been proposed to eliminate those antagonistic forces. The concept of actuation redundancy is illustrated in Figure 1.2 through two 3-DOF PKMs.

## 1.5 Objectives of the Thesis

The main objective of the present research consists of proposing and validating through numerical simulations and real-time experimentation, advanced control schemes to enhance

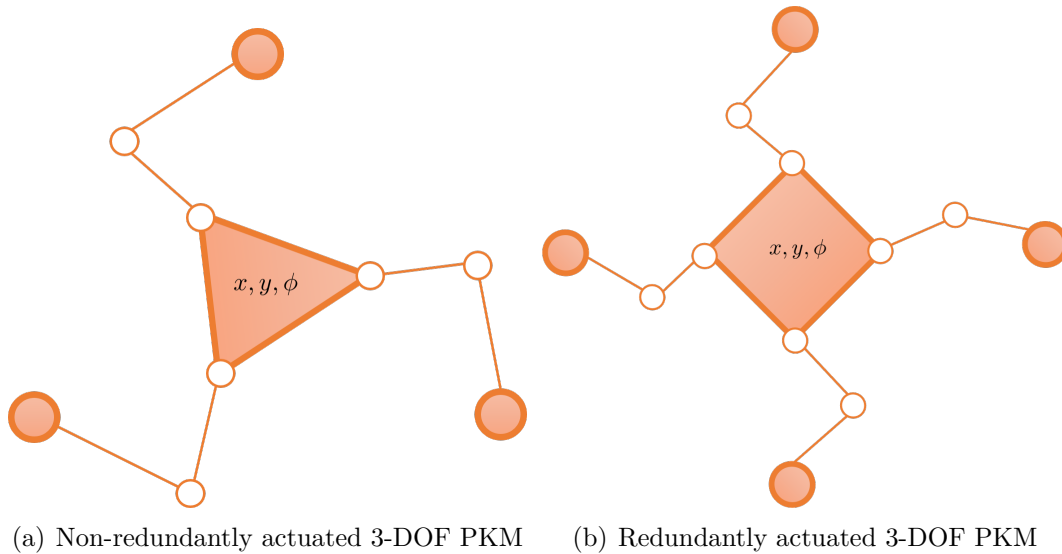


Figure 1.20: Illustration of the concept of actuation redundancy in PKMs

the tracking performance of parallel robots operating in a wide of work environments. Such control solutions are validated in two PKMs prototypes with different application purposes. The first one corresponds to a redundantly actuated PKM designed for machining operations called SPIDER4. The last one is a non-redundantly actuated 3-DOF Delta PKM designed for P&P applications. For the Delta robot, many modeling approaches have been reported in the literature. However, until this investigation, no information related to the modeling of SPIDER4 had been previously reported. Therefore, the kinematic and models corresponding to SPIDER4 must be developed before to validate such control schemes on this platform.

## 1.6 Main Contributions of the Thesis

In order to fulfill the objectives set forth above, the following contributions were developed in this thesis research.

### 1.6.1 Kinematic and Dynamic models for a New RA-PKM called SPIDER4

With the purpose of testing advanced control schemes on SPIDER4, it was primordially necessary to develop its kinematics and dynamics models. Mechanically, SPIDER4 is formed by a redundant actuated parallel device responsible for the positioning and serial wrist mechanism that performs the machining tool orientation. The Inverse Kinematic Model (IKM) of the SPIDER4 was formulated through the loop closure method commonly used in PKM, and a geometrical analysis for the serial wrist. In contrast, the Forward Kinematic Model (FKM) was obtained through a virtual spheres algorithm in virtue that the traveling plate of SPIDER4 always remains in the same orientation. The Inverse Dynamic Model (IDM) of SPIDER4

proposed in this work was formulated considering some modeling simplifications with the purpose of integrating it as a part of the proposed control solutions. Such simplifications aim to reduce the computation time in real-time implementations.

### **1.6.2 Control solutions for PKMs Based-on RISE Control**

Robust Integral of the Sign Error (RISE) is a robust control technique that is becoming popular in robotics control since it can compensate additive system disturbances and parametric uncertainties guaranteeing asymptotic stability. Considering these features presented by the RISE control. This controller is selected as the feedback controller base for the following control solutions.

#### **Control Solution 1: A RISE Controller with Nominal Feedforward**

This control solution has the objective to validate the effectiveness of the proposed inverse dynamic model of SPIDER4. To this end, the obtained dynamic equations were integrated as the feedforward term together with RISE control. This control solution was validated by real-time experimentation in a nominal scenario with respect to standard RISE control. Performance indexes were used to determine the enhancement produced by the addition of the obtained inverse dynamic model.

#### **Control Solution 2: A RISE Feedforward Controller with Adaptive Feedback Gains**

Since SPIDER4 is conceived to perform machining operations, contact forces will inevitably affect the robot's performance in a milling or drilling operation. In order to hold as small as possible the tracking errors under this operative condition, adaptive feedback gains for RISE feedforward are proposed. If the values of the tracking errors in joint space increase, adaptation laws will make the values of certain gains of the RISE controller increase to keep the values of the tracking errors as small as possible without overestimating the controller gains that may lead to producing undesirable noise. This control solution was validated through a real-time machining task. Its performance was compared with respect to Standard RISE and the previously mentioned RISE feedforward controller.

#### **Control Solution 3: A RISE Controller with Adaptive Feedforward Compensation Based-on B-Spline Neural Networks**

Because of the presence of structured uncertainties in the majority of work environments, the fusion of RISE control with Associative Memory Neural Networks (AMNN) is proposed to keep tracking errors as small as possible despite the presence of such issues. The proposed control solution makes use of a kind of AMNN called B-spline Neural Network (BSNN). The BSNN aims to online estimate the dynamics of the Delta PKM to integrated it into the control loop in order to cancel inherent nonlinearities and compensate for uncertainties. Artificial Neural Networks (ANN) are well known to be universal approximators, being employed widely in



nonlinear model identification. The reasons behind the selection of the BSNN for online estimation are explained in Chapter 4. This control contribution was validated by the Delta robot under different scenarios through numerical simulations in Chapter 5.

## 1.7 Conclusion

In this chapter, an overall framework of PKMs has been presented. The chapter introduces robotics concepts as fundamental definitions, the classification of the different kinematic structures of robotics manipulators (serial, parallel, and hybrid), making a comparison of the advantages and drawbacks between serial and parallel manipulators. The first investigations and prototypes of parallel robots began in the 20th century, standing out mainly the PKMs with Gough/Stewart architecture and those based on the Delta PKM. Emphatizing on the Delta robot, many modifications have been carried out to the original prototype in order to enhance its capabilities in terms of improved stiffens, larger workspace, and the ability to reach higher accelerations. For their desirable features compared to serial manipulators, PKMs have dabbled in different industries developing multiple applications where precision is essential. However, PKMs are known to be highly nonlinear systems integrating the presence of uncertainties and, in some cases, the redundant actuation; therefore, advanced control strategies must be provided to deal with the mentioned challenges guaranteeing good dynamic and tracking performance of PKMs in tasks executions. This thesis aims to develop model-based control strategies to improve the tracking performance of PKMs in the presence of such issues. In the next chapter, the state of the art of proposed control solutions for PKMs in the literature will be presented.






---

## State of the Art

*Since this thesis's aim relies on the modeling and control of PKM, this chapter introduces a brief description of the main formulations to compute the dynamic model of PKMs. These formulations are Newton-Euler, the principle of virtual works, and Euler-Lagrange. Moreover, it is explained a methodology simplification for PKMs with delta-like architecture. Regarding the control topic, a survey of some existing control techniques implemented on PKMs is presented. These control schemes are classified into main categories: non-model-based controllers and model-based controllers.*

### Contents

---

<b>2.1</b>	<b>Introduction</b>	<b>27</b>
<b>2.2</b>	<b>Dynamic Modeling of Parallel Kinematic Machines</b>	<b>28</b>
<b>2.3</b>	<b>Overview of Motion Controllers for PKMs</b>	<b>35</b>
<b>2.4</b>	<b>Non-model-based controllers</b>	<b>35</b>
<b>2.5</b>	<b>Model-Based-Controllers</b>	<b>39</b>
<b>2.6</b>	<b>Conclusion</b>	<b>46</b>

---

### 2.1 Introduction

As stated in Chapter 1, this investigation aims to propose new advanced control solutions of PKMs to fulfill the requirements needed to execute a tracking trajectory task satisfactorily. That is, the proposed controller must guaranteeing high precision under different operative conditions even in the presence of any uncertainty or perturbation that may affect the system performance. In order to test the proposed control techniques, it is necessary to have knowledge of the kinematic and dynamic models of the experimental platforms that will be used. For the case of the Delta PKM, both its kinematic and dynamic models have been

well reported in the literature. However, SPIDER4 is a new prototype, and therefore it is necessary to calculate both its kinematic and dynamic models before proceeding with the implementation of controllers in it. In this way, we establish the other main objective of this thesis, computing the kinematic and dynamic models for SPIDER4. Therefore, let us present an overview of the dynamic formulations for PKMs before describing some implemented control strategies on them.

## 2.2 Dynamic Modeling of Parallel Kinematic Machines

Dynamic modeling formulation is a crucial part of the study of parallel robots since it allows us to analyze the behavior of them under different operative conditions through numerical simulations. The results obtained from the simulations help researchers propose improvements to any PKM in terms of mechanical or control design to obtain a proper performance under specific operating requirements. Moreover, some advanced controller techniques require full or partial information about the dynamic equations of the PKM to be controlled to improve the tracking performance. Dynamic analysis of PKM is complex due to the closed-loop structure involving several kinematic constraints. In the literature, the most common ways to develop the dynamic motion equations for a PKM are based on the Newton-Euler, principle of virtual works and Lagrangian formulations.

### 2.2.1 Dynamic modeling approaches for PKM

Before proceeding with the explanation of the different formulations to obtain the equations of motion of a parallel robot, let us define the following nomenclatures for the analysis taking into consideration Figure 2.1.

- The fixed base of the PKM is attached to the fixed reference frame  $O - x_o, y_o, z_o$
- A reference frame is located on the moving platform denoted by  $p - x_p, y_p, z_p$  whose coordinates are in function of  $O - x_o, y_o, z_o$
- For any parallel robot the number of DOF will be represented by  $m$
- The joints (prismatic, revolute, universal, etc.) are located at the points  $A_{ij}$ , where the kinematic chain number is denoted by  $i = 1, \dots, n$ , and the joint number by  $j = 1, \dots, l$

#### Newton-Euler formulation

Newton-Euler (NE) is one of the most common formulations used to compute the dynamic model in PKMs. The methodology is based on the computation of the motion equations for each body element of the robot. Applying this formulation is necessary to derive linear and angular accelerations of links as well as the moving platform. However, it leads to a large number of equations resulting in poor computation efficiency [Tsai, 1999]. Considering the

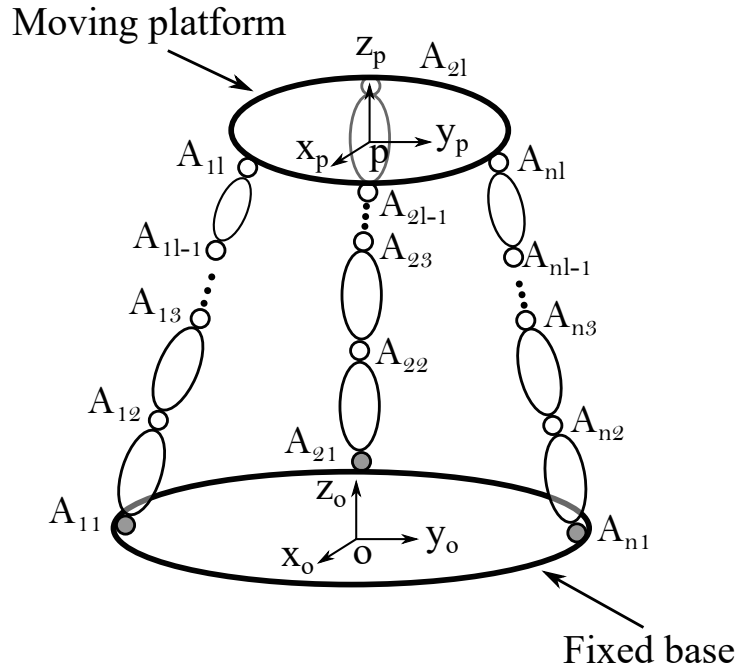


Figure 2.1: General kinematic representation of a PKM

diagram presented in Figure 2.1, we defined the NE equations for the link set of a PKM that may be expressed in the following form [Briot and Khalil, 2015]:

$$\begin{aligned} \sum \mathbf{f}_{ij} &= m_{ij}^c \ddot{\mathbf{x}}_{ij} \\ \sum {}^c \mathbf{n}_{ij} &= {}^c \mathbf{I}_{ij} \dot{\boldsymbol{\omega}}_{ij} + \boldsymbol{\omega}_{ij} \times ({}^c \mathbf{I}_{ij} \boldsymbol{\omega}_{ij}) \end{aligned} \quad (2.1)$$

where:

- $m_{ij}$  is the body mass
- $\sum \mathbf{f}_{ij}$  is the sum of external forces acting on each link
- $\sum {}^c \mathbf{n}_{ij}$  is the sum of external moments including gravity effects acting of the body center of mass
- ${}^c \ddot{\mathbf{x}}_{ij}$  is the acceleration to the center of mass of the  $ij$  link
- $\dot{\boldsymbol{\omega}}_{ij}, \boldsymbol{\omega}_{ij}$ , are the rotational acceleration and velocity of the  $ij$  link
- ${}^c \mathbf{I}_{ij}$  is the inertia matrix of the  $ij$  link expressed at its center of mass.

The NE equations describing the dynamics of the moving platform are expressed as:

$$\begin{aligned} \mathbf{f}_p &= m_p^c \ddot{\mathbf{x}}_p \\ {}^c \mathbf{n}_p &= {}^c \mathbf{I}_p \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times ({}^c \mathbf{I}_p \boldsymbol{\omega}_p) \end{aligned} \quad (2.2)$$

In which:

- $m_p$  is the mass of the moving platform
- ${}^c\mathbf{I}_p$  is the inertia matrix of the moving platform expressed at its center of mass
- $\ddot{\mathbf{x}}_p$ , and  $\dot{\boldsymbol{\omega}}_p$  are the linear and rotational acceleration of the moving platform

One can mention some literature works where NE formulation has been used to compute Parallel robots' dynamics. For instance, in [Borchert et al., 2015], the Newton-Euler equations were employed to obtain the dynamic model of a 3-DOF Delta PKM. Also, in [Harib and Srinivasan, 2003], the Newton-Euler method was applied to get the inverse dynamic model of a Stewart platform.

### Principle of virtual works

This principle establishes that the virtual work produced by external forces and moments corresponding to any set of virtual displacement of a body or system of bodies is zero. Unlike the NE formulation, this principle does not imply the unknown contains forces and moments, resulting in less computational effort. The principle of virtual works applied to a system of  $N$  bodies results in the following equation [Codourey, 1998]:

$$\delta W = \sum (m_i \ddot{\mathbf{x}}_i - \mathbf{f}_i) \cdot \delta \mathbf{x}_i + \sum ({}^c\mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times ({}^c\mathbf{I}_i \boldsymbol{\omega}_i) - {}^c\mathbf{n}_i) \cdot \delta \boldsymbol{\theta}_i = 0 \quad (2.3)$$

Where  $\delta \mathbf{x}_i$ , and  $\delta \boldsymbol{\theta}_i$  represent the virtual linear and rotational displacement of the  $i$  body, respectively. The previous equation can be adapted to a PKM with  $n$  number of limbs and a moving platform as follows [Taghirad, 2013]:

$$\delta W = \delta \mathbf{q}^T \boldsymbol{\Gamma} + \delta \mathbf{X}_p^T \hat{\mathbf{F}}_p + \sum \delta \mathbf{X}_i^T \hat{\mathbf{F}}_i = 0 \quad (2.4)$$

Where:

- $\delta \mathbf{X}_p$ , and  $\delta \mathbf{X}_i$  represents the virtual displacements of the moving platform and limbs center of masses, respectively
- $\delta \mathbf{q}$  denotes the virtual displacements of the joint variables
- $\boldsymbol{\Gamma}$  is the vector of the actuator forces
- $\hat{\mathbf{F}}_p$  and  $\hat{\mathbf{F}}_i$  represent the difference between the generalized inertial forces/moments, and the generalized external forces/moments acting on the moving platform center of mass, and each limb center of mass, respectively, whose components are the following:

$$\hat{\mathbf{F}}_p = \begin{bmatrix} m_p \ddot{\mathbf{x}}_p - \mathbf{f}_p \\ \mathbf{I}_p \dot{\boldsymbol{\omega}}_p + \boldsymbol{\omega}_p \times (\mathbf{I}_p \boldsymbol{\omega}_p) - \mathbf{n}_p \end{bmatrix} \quad \hat{\mathbf{F}}_i = \begin{bmatrix} m_i^c \ddot{\mathbf{x}}_i - \mathbf{f}_i \\ {}^c\mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times ({}^c\mathbf{I}_i \boldsymbol{\omega}_i) - {}^c\mathbf{n}_i \end{bmatrix}$$

In which  $m_p$  is the moving platform mass,  $\mathbf{I}_p$  is the inertia moment of the moving platform, and  $\mathbf{f}_p$  and  $\mathbf{n}_p$  represent the external forces and moments acting of the moving platform. The virtual displacements of the joint variables and the limbs are related to the virtual displacement of the moving platform by means the following relations:

$$\delta \mathbf{q} = \mathbf{J} \delta \mathbf{X}_p \quad \delta \mathbf{X}_i = \mathbf{J}_i \delta \mathbf{X}_p \quad (2.5)$$

Where  $\mathbf{J}$  is the Jacobian matrix of the manipulator, and  $\mathbf{J}_i$  is the limb Jacobian matrix. Substituting (2.5) in (2.4) it yields:

$$\delta \mathbf{X}_p^T (\mathbf{J}^T \boldsymbol{\Gamma} + \hat{\mathbf{F}}_p + \sum \mathbf{J}_i^T \hat{\mathbf{F}}_i) = 0 \quad (2.6)$$

Finally, canceling  $\delta \mathbf{X}_p^T$  from the previous expression and rearranging terms. The dynamic formulation of a PKM using the virtual works principle can be stated as follows:

$$\boldsymbol{\Gamma} = -\mathbf{J}^{-T} (\hat{\mathbf{F}}_p + \sum \mathbf{J}_i^T \hat{\mathbf{F}}_i) \quad (2.7)$$

The virtual work formulation has been extensively used in dynamic formulation of PKM as can be seen in [Codourey, 1996], [Zhao and Gao, 2009], [Zhao et al., 2005], and [Li and Xu, 2005], among others works.

### Euler-Lagrange formulation

Euler-Lagrange formulation describes the motion system based on the kinetic and potential energy of the whole system. This formulation is suitable for dynamic analysis of multibody systems where the forces and moments of each body interact with each other. The Lagrangian function can be expressed in the following form:

$$\mathcal{L} = K - U \quad (2.8)$$

In which  $K$  and  $U$  represent the total kinetic and potential energy of the system, respectively. According to [Taghirad, 2013], the Lagrange motion equations can be divided into two kinds first kind and second kind. The simplest representation of them is the second kind, which is expressed as:

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{q}_j} \right) - \frac{\partial \mathcal{L}}{\partial q_j} = Q_j, \quad j = 1, 2, \dots, m \quad (2.9)$$

Where  $q_j$  is the generalized coordinate,  $Q_j$  represents the generalized force, and  $m$  expresses the system's independent generalized coordinates. The second kind of Lagrangian motion equation is suitable for manipulators with open-loop configuration. However, the closed-loop construction of PKMs introduces a set of constraint equations that must be introduced in the motion equations leading the first kind of Lagrange motion equations, expressed in the next equation:

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{q}_j} \right) - \frac{\partial \mathcal{L}}{\partial q_j} = Q_j + \sum_{i=1}^k \lambda_i \frac{\partial h_i}{\partial q_j}, \quad j = 1, 2, \dots, n \quad (2.10)$$

Where  $k$  is the number of constraints,  $\lambda_i$  is the Lagrange multiplier,  $n$  represents the number of redundant coordinates exceeding the number of DOF  $m$  by  $k$ , which is  $n = m + k$ , and  $h_i$  is the constrain function. In order to compute the generalized forces  $Q_j$ , it is necessary to split up (2.10) in two sets [Tsai, 1999]. The first one contains the Lagrange multipliers as the unknown variables, and the second one is the generalized forces as the unknown variables. The first set can be written as:

$$\sum_{i=1}^k \lambda_i \frac{\partial h_i}{\partial q_j} = \frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{q}_j} \right) - \frac{\partial \mathcal{L}}{\partial q_j} - \hat{Q}_j, \quad j = 1, 2, \dots, k \quad (2.11)$$

Being  $\hat{Q}_j$  a generalized external force that is supposed to be known. Once the Lagrange multipliers are found, the forces and/or torques of the actuators can be obtained through the following expression:

$$Q_j = \frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{q}_j} \right) - \frac{\partial \mathcal{L}}{\partial q_j} - \sum_{i=1}^k \lambda_i \frac{\partial h_i}{\partial q_j}, \quad j = k + 1, \dots, n \quad (2.12)$$

Some related works where Euler-Lagrange formulation has been used to obtain the dynamic model equations in Parallel robots are [Ahmadi et al., 2008], [Park et al., 2013], and [Geng et al., 1992], among others.

### Simplified method for PKMs with delta-like structure

The computation of a suitable dynamic model for real-time implementation of PKMs represents a significant challenge due to the large number of obtained equations, which can become a problem due to the large number of computing resources that can be consumed by the computer. Parallel robots with delta-like structure have linkages formed by two elements, the first one is the rear-arm formed by a single piece, and the second one is the forearm, which is constituted by two parallelogram bars. Obtaining the dynamic model of this type of robots using only any of the three classic methodologies presented above would lead to multiple real-time computing problems. For example, when using the NE formulation, the forces and moments of all the robot elements must be considered, resulting in a large number of equations that demand significant computational time to solve them. Alternatively, if one chooses to use the principle of virtual works, the calculation of the inverse link-Jacobian matrices will represent a laborious computation process. For PKMs based on the Delta PKM, it has been proposed some hypothesis simplifications in [Nabat, 2007] to obtain a suitable dynamic model for real-time implementation. The hypothesis simplifications are listed below:

- Since obtaining an accurate frictional model for PKMs, the frictional forces dry and viscous are omitted in the analysis.
- The rotational inertia of the forearms is neglected. Nevertheless, its mass is divided into two equivalent parts; one part is added to the rear-arm mass, and the other part is joined to the traveling plate mass. This simplification is justified if the forearms' mass is smaller than the other components of the robot.

Concerning the second simplification, the ratio between the mass of the forearms and the mass of the rear-arms must be sufficiently small (that is less or equal than 0.3 equaling to a 2% error with respect to a complete dynamic model), as it is established on the analysis developed for PKM with delta-like architecture in [Nabat, 2007] Figure 2.2 illustrates the concept such model simplifications. The resulting equation of motion considering the above simplifications will have the following form:

$$\mathbf{\Gamma}(t) = \mathbf{\Gamma}_{act} + \mathbf{\Gamma}_p \quad (2.13)$$

Where  $\mathbf{\Gamma}_{act} \in \mathbb{R}^n$  involves the forces/torque contributions of the actuators and the rear-arms plus a half mass of the forearms, whereas that  $\mathbf{\Gamma}_p \in \mathbb{R}^n$  represents the forces/torque



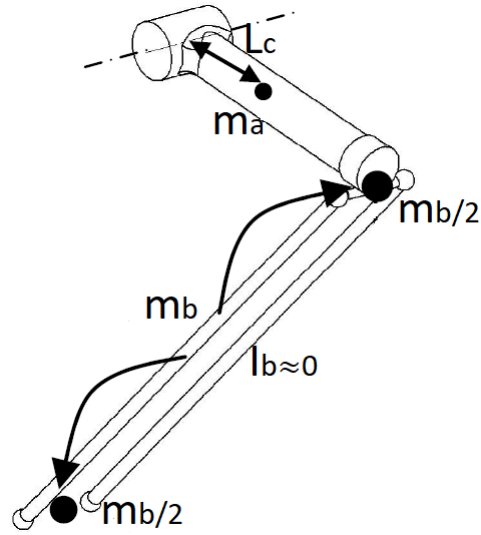


Figure 2.2: Illustration of modeling simplification for PKMs with delta-like architecture [Nabat, 2007]

contribution of the moving platform plus the other half mass of the forearm. This formulation will be considered in Chapter 3 for the development of the dynamic models of the experimental platforms.

## 2.2.2 Dynamic Modeling Representation

The dynamic study in robotic manipulators deals with the relationship between the generalized forces and torques of a robotic system with its corresponding accelerations, velocities, and position of the joints and end-effector of the manipulator. The dynamic analysis can be divided into two branches [Staicu, 2019]:

- *Forward Dynamics*: It can be defined as given the set generalized forces and torques of the robot, compute the resulting motion of the end-effector and joints as a function of time. The Forward dynamic formulation expressed for the Cartesian space variables can be represented as follows:

$$\ddot{\mathbf{X}} = f(\mathbf{F}, \dot{\mathbf{X}}, \mathbf{X}) \quad (2.14)$$

where  $\mathbf{X} \in \mathbb{R}^m$  is the vector of the Cartesian space coordinates and  $\mathbf{F} \in \mathbb{R}^m$  is the vector of forces/torques applied on the moving platform. The representation of the Forward dynamics in joint space is presented in the following form:

$$\ddot{\mathbf{q}} = f(\mathbf{\Gamma}, \dot{\mathbf{q}}, \mathbf{q}) \quad (2.15)$$

where  $\mathbf{q} \in \mathbb{R}^n$  is the vector of joint space coordinates, and  $\mathbf{\Gamma} \in \mathbb{R}^n$  is the vector of actuated joint force/torque.

- *Inverse Dynamics*: It can be defined as given a set of trajectories for the end-effector and actuated joints in the function of time, find the set of forces/torques that produce the motion. This sentence may be represented in the following form for Cartesian and joint space coordinates as follows:

$$\mathbf{F} = f(\ddot{\mathbf{X}}, \dot{\mathbf{X}}, \mathbf{X}) \quad (2.16)$$

$$\mathbf{\Gamma} = f(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) \quad (2.17)$$

The forces/torques applied to the end-effector and those produced by the actuated joints are related by means the following expression:

$$\mathbf{F} = \mathbf{J}^T \mathbf{\Gamma} \quad (2.18)$$

Being  $\mathbf{J} \in \mathbb{R}^{m \times n}$  the Jacobian matrix of the manipulator. The inverse dynamic modeling plays a vital role in real-time model-based controllers. Therefore, we will explain in more detail expressions (2.16)-(2.17).

### Representation of the Inverse Dynamic Model in joint space

The IDM of a robotic system with  $m$  DOF and  $n$  actuators can be represented for the joint coordinates as follows [Kelly et al., 2006]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{\Gamma}(t) \quad (2.19)$$

In which

- $\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q} \in \mathbb{R}^n$  The vectors of acceleration, velocity, and position of the robot in joint space
- $\mathbf{\Gamma} \in \mathbb{R}^n$  is the input torque vector
- $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$  is the inertia matrix
- $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$  is the Centripetal/Coriolis forces matrix
- $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^n$  is the vector of produced forces due to gravity acceleration

This representation is the most common representation of the IDM for control applications because most PKMs do not have sensors to measure the position of the traveling plate directly with respect to the fixed coordinates system.

### Representation of the Inverse Dynamic Model in Cartesian space

The IDM can be expressed in Cartesian coordinates by means the following relationships based-on the Jacobian matrix.

$$\dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{X}} \quad \ddot{\mathbf{q}} = \mathbf{J}\ddot{\mathbf{X}} + \dot{\mathbf{J}}\dot{\mathbf{X}} \quad (2.20)$$

Substituting (2.20) and (2.18) in (2.19), the IDM in Cartesian space can be written as [Taghirad, 2013]:

$$\mathbf{M}_X \ddot{\mathbf{X}} + \mathbf{C}_X \dot{\mathbf{X}} + \mathbf{G}_X = \mathbf{J}^T \boldsymbol{\Gamma} \quad (2.21)$$

Where

- $\ddot{\mathbf{X}}, \dot{\mathbf{X}}, \mathbf{X} \in \mathbb{R}^m$  are the acceleration, velocity and position vector of the moving platform of the PKM
- $\mathbf{M}_X = \mathbf{J}^T \mathbf{M}(\mathbf{q}) \mathbf{J}$  is the inertia matrix expressed in Cartesian space
- $\mathbf{C}_X = \mathbf{J}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{J}} + \mathbf{J}^T \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{J}$  is the Centripetal/Coriolis matrix expressed in Cartesian space
- $\mathbf{G}_X = \mathbf{J}^T \mathbf{G}(\mathbf{q})$  is the vector of produced forces due to gravity acceleration expressed in Cartesian space

## 2.3 Overview of Motion Controllers for PKMs

In the literature, several control approaches have been proposed for motion control of parallel robots. Many of them have been taken from control schemes implemented in serial robots since they share similarities in the dynamic model's mathematical representation. Control solutions for robotics can be classified into two primary sorts: non-model-based and model-based controllers [Saied et al., 2019b]. The first type is the easiest to implement since they do not require any knowledge about the controlled system dynamics. In contrast, the second type of controllers makes use of the whole or some portions of the inverse dynamic model. In most cases, the obtained tracking performance using model-based controllers outcomes produced by a non-model-based controller [Cheng et al., 2003]. Other possible classifications of control laws of robotics are in non-adaptive and adaptive controllers [Bennehar, 2015]. Figure 2.3 presents a classification of control strategies for PKMs with some controllers examples existing in the literature. Such examples will be briefly described in the next subsections of this chapter except for those based on the RISE control, which will be described in greater detail in Chapter 4.

## 2.4 Non-model-based controllers

As its name says, these control schemes do not need information about the dynamics of the system to be controlled. All the necessary information lies in the measurement of the system's states (positions and velocities). Non-model-based controllers can be divided into two categories: Non-adaptive and adaptive controllers, which will be discussed in greater detail in the following subsections.

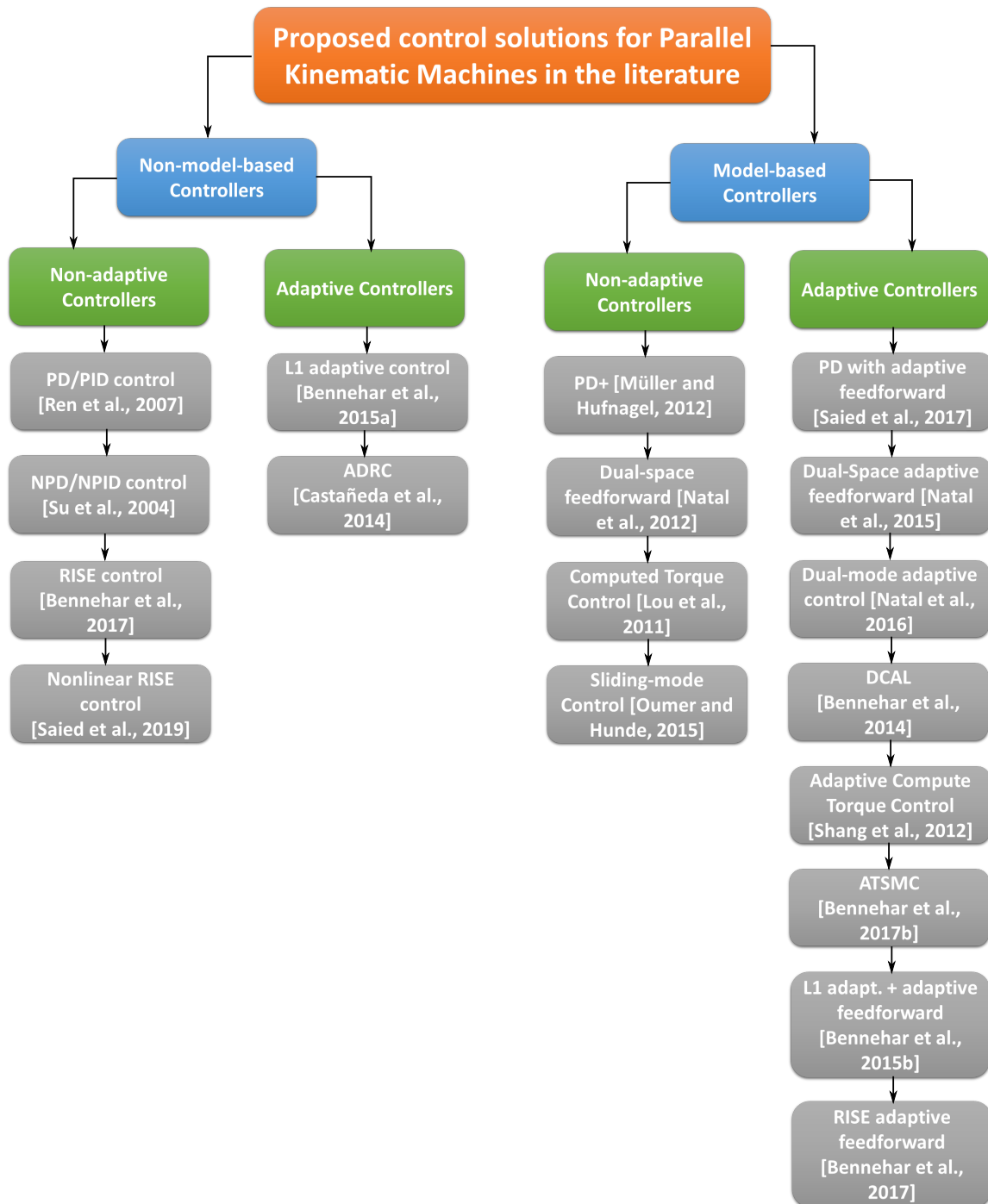


Figure 2.3: Classification of implemented control schemes on Parallel Kinematic Machines

### 2.4.1 Non-model-based-non-adaptive controllers

The implementation of these control schemes on parallel robots is generally easy and straightforward since they do not require any prior knowledge about the manipulator dynamics. The data providing from the robot states are processed in a feedback control loop, which sends the control commands to the robot actuators. Most of them have fixed feedback gains providing relatively good performance for not very demanding tasks. However, to increase these controllers' robustness against external disturbances, it has been proposed to replace the fixed gains by nonlinear time-varying gains. So let us start with the description of them.

#### PD/PID Controllers

The Proportional-Integral-Derivative (PID) controller [Ziegler et al., 1942] is the most extensively used controller in the industry even to these days, thanks to its simplicity and relatively good performance. However, the performance of this controller decreases considerably if the system is subjected to high accelerations. In the literature, PID control has been implemented in parallel robots in both joint space [Ren et al., 2007] and Cartesian space [Natal et al., 2015]. For joint space, the equation of the PID controller can be written as:

$$\mathbf{\Gamma}(t) = \mathbf{K}_p \mathbf{e}_q(t) + \mathbf{K}_d \frac{d}{dt} \mathbf{e}_q(t) + \mathbf{K}_i \int_0^t \mathbf{e}_q(\tau) d\tau \quad (2.22)$$

Where  $\mathbf{K}_p$ ,  $\mathbf{K}_d$ , and  $\mathbf{K}_i \in \mathbb{R}^{n \times n}$  are positive definite diagonal gains matrices for the proportional, derivative, and integral actions, respectively. The tracking error in joint space  $\mathbf{e}_q(t) = \mathbf{q}_d(t) - \mathbf{q}(t)$  is defined as the the difference between the desired trajectory in joint space  $\mathbf{q}_d(t) \in \mathbb{R}^n$ , and the measured one  $\mathbf{q}(t) \in \mathbb{R}^n$ . (2.22) represents the most common way to implement the PID control on PKMs since most of them only dispose of encoders to estimate the joint position of the actuators. Nevertheless, for those PKMs disposing of sensors that allow them to estimate the position of the moving platform directly, it is recommended to implement the PID controller in Cartesian space. Its expression is slightly different from the previous one.

$$\mathbf{F}(t) = \mathbf{K}_p \mathbf{e}_X(t) + \mathbf{K}_d \frac{d}{dt} \mathbf{e}_X(t) + \mathbf{K}_i \int_0^t \mathbf{e}_X(\tau) d\tau \quad (2.23)$$

Where  $\mathbf{e}_X(t) = \mathbf{X}_d(t) - \mathbf{X}(t)$  denotes the tracking error in Cartesian space, being  $\mathbf{X}_d(t) \in \mathbb{R}^m$ , the desired trajectory, and  $\mathbf{X}(t) \in \mathbb{R}^m$  the measured one,  $\mathbf{K}_p$ ,  $\mathbf{K}_d$ , and  $\mathbf{K}_i \in \mathbb{R}^{m \times m}$  are the controller gains . Since (2.23) computes  $\mathbf{F}(t)$  instead of  $\mathbf{\Gamma}(t)$ , the control signals for the actuators must be computed trough the Jacobian matrix  $\mathbf{J} \in \mathbb{R}^{m \times n}$ , or the pseudoinverse Jacobian matrix  $\mathbf{H} \in \mathbb{R}^{m \times n}$  if the PKM is non-redundantly or redundantly actuated. These relationships are indicated in (2.17) and (2.18).

#### Nonlinear PD/PID Controllers

Nonlinear PD/PID controllers arise from the need to improve the response and tracking performance of standard PID controllers with fixed gains. The nonlinear time-varying gains endow the controller with the ability to reject external disturbances. Furthermore,

this controller is less sensitive than its counterpart with fixed gains [Seraji, 1998]. The mathematical expression of the Nonlinear PID controller applied to a robotic manipulator can be written in terms of the joint space variables as:

$$\mathbf{\Gamma}(t) = \mathbf{K}_p(\cdot)\mathbf{e}_q(t) + \mathbf{K}_d(\cdot)\frac{d}{dt}\mathbf{e}_q(t) + \mathbf{K}_i(\cdot)\int_0^t \mathbf{e}_q(\tau)d\tau \quad (2.24)$$

Where  $\mathbf{K}_p(\cdot)$ ,  $\mathbf{K}_d(\cdot)$ , and  $\mathbf{K}_i(\cdot) \in \mathbb{R}^{n \times n}$  are the nonlinear time-varying feedback gains for the proportional, derivative, and integral actions, respectively. The controller gains are automatically adjusted according to a series of rules based on the measurement of the error tracking signal. If the tracking error is large, the gains' values will increase their values, making a quick correction action. Once the error is diminished, the feedback gains are automatically reduced to avoid excessive oscillations and overshoots. Several methodologies have been proposed in the literature to adjust the nonlinear time-varying feedback gains [Ouyang et al., 2002], [Armstrong and Wade, 2000], [Seraji, 1998], among others. One can mention an example application in parallel robots. For instance, in [Su et al., 2004], a nonlinear PID controller in joint space was proposed to regulate the positioning of a 6-DOF PKM.

## 2.4.2 Non-model-based-adaptive Controllers

Controller schemes of this do not consider any information on the system dynamics. However, They exploit the information available from the system states to online-estimate nonlinearities, uncertainties, and in some cases, the system dynamic behavior. If the adaptation law is designed to estimate the system's uncertainties and nonlinearities, then the controller will take action to reduce the effects produced by them. On the other hand, if the adaptation law is designed to estimate the system's dynamics, this will be integrated as a compensation term in the controller scheme. Some control schemes that belong to this category are:

### $\mathcal{L}_1$ Adaptive Control

$\mathcal{L}_1$  adaptive control is a recent control strategy published for the first time in [Cao and Hovakimyan, 2006a] and [Cao and Hovakimyan, 2006b]. This control technique is inspired in the MRAC (Model Reference Adaptive Control) but decoupling the feedback control loop from the adaptive estimation loop. Moreover, this controller guarantees the controller signals to stay in the low-frequency range. In [Bennehar et al., 2015a], the first implementation of the  $\mathcal{L}_1$  control in a PKM is reported. Two independent terms compose the controller equation, a fixed state-feedback term characterizing the evolution of the transient response of the PKM and an adaptive term that compensates for the nonlinearities of the system, as follows:

$$\mathbf{\Gamma}(t) = \mathbf{A}_m \mathbf{r}(t) + \mathbf{\Gamma}_{ad}(t) \quad (2.25)$$

Where  $\mathbf{A}_m \in \mathbb{R}^{n \times n}$  is the Hurwitz matrix,  $\mathbf{r}(t) \in \mathbb{R}^n$  is the combined tracking error in joint space  $\mathbf{r}(t) = \dot{\mathbf{e}}_q(t) + \mathbf{\Lambda} \mathbf{e}_q(t)$  where  $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$  is a positive definite gain matrix. The adaptive control term  $\mathbf{\Gamma}_{ad}(t)$  is the output of the following equation:

$$\mathbf{\Gamma}_{ad}(s) = \mathbf{C}(s)\hat{\boldsymbol{\eta}}(s) \quad (2.26)$$

In which,  $\hat{\boldsymbol{\eta}}(s)$  is the Laplace transform of  $\hat{\boldsymbol{\eta}}(t) = \hat{\boldsymbol{\theta}}(t)\|\mathbf{r}(t)\|_{\mathcal{L}_\infty} + \hat{\boldsymbol{\zeta}}(t)$ , and  $\mathbf{C}(s)$  is a low-pass filter.  $\hat{\boldsymbol{\theta}}(t)$ , and  $\hat{\boldsymbol{\zeta}}(t)$  are nonlinear estimation functions for the nonlinearities and disturbances of the system, respectively. The procedure to estimate such nonlinear functions is detailed in [Bennehar, 2015], [Bennehar et al., 2016a], and [Hovakimyan and Cao, 2010].

### Active Disturbance Rejection Control

Similar to the  $\mathcal{L}_1$  adaptive controller, the Active Disturbance Rejection Control (ADCR) does not require any knowledge about the dynamics of the system to be controlled. This controller on-line estimates nonlinearities and parametric uncertainties for a subsequent proper cancellation of them. The estimation is frequently obtained through Extend State Observers (ESOs) being a kind of high gain observers [Gao, 2006]. To understand the logic of this control law in broad strokes, the dynamic model of a robot must be expressed as follows:

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}\boldsymbol{\Gamma} - \mathbf{M}(\mathbf{q})^{-1}[\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})] \quad (2.27)$$

The previous model can be represented in state variables as follows:

$$\begin{aligned} \dot{\mathbf{x}}_a(t) &= \mathbf{x}_b(t) \\ \dot{\mathbf{x}}_b(t) &= f(\dot{\mathbf{x}}_T(t)) + g(\mathbf{x}_a(t))\boldsymbol{\Gamma}(t) + \chi(\mathbf{x}_T(t), t) \end{aligned} \quad (2.28)$$

Where  $\mathbf{x}_T \in \mathbb{R}^{2n}$  is the robotic system state vector that is  $\mathbf{x}_T = [\mathbf{x}_a \ \mathbf{x}_b]^T$ . Being  $\mathbf{x}_a = \mathbf{q}$ , and  $\mathbf{x}_b = \dot{\mathbf{q}}$ , the term  $\chi(\mathbf{x}_T, t)$  considers the presence of uncertainties and non-modeled sections. With the system structure presented in (2.28) it is proceeded to design the high gain observer resulting in a control law with the following structure.

$$\boldsymbol{\Gamma}(t) = -g(\mathbf{x}_a)^{-1}(\mathcal{H}^T \hat{\mathbf{x}}_T(t) + \hat{\boldsymbol{\Lambda}}(t)\hat{\mathbf{x}}_T(t)) \quad (2.29)$$

In which,  $\mathcal{H} \in \mathbb{R}^{2n}$  is a linear constant gain matrix, and  $\hat{\boldsymbol{\Lambda}} \in \mathbb{R}^{2n \times n}$  is a matrix containing the time-varying parameters which are adjusted by an adaptive rule based on the least mean square error. ADCRC was implemented successfully to a 3-DOF Delta PKM in [Castañeda et al., 2014].

## 2.5 Model-Based-Controllers

Control schemes belonging to this category incorporate complete or partial knowledge of the dynamics of the system to reduce the effects of nonlinearities, enhancing the overall performance of the robotic system. In most of the cases reported in the literature, the performance obtained through these control schemes far exceeds that of non-model-based controllers [Saied et al., 2019b].

### 2.5.1 Model-based-non-adaptive Controllers

As the name of this subcategory indicates, the estimated dynamic parameters of the robot always remain fixed, which has certain advantages and disadvantages. On the one hand, if exist

precise knowledge of such parameters, the performance obtained will be satisfactory. However, if such parameters are poorly estimated, instead of improving the robot's performance, it will worsen. Some examples of this type of control schemes are:

### Computed Torque Control

Computed Torque Control (CTC), also known in some literature as Inverse Dynamics Control (IDC) [Taghirad, 2013], is a control technique that has been widely applied in the control of robotics. It is a non-adaptive control technique that allows us to obtain a linear closed-loop equation in terms of the state variables. This control technique may provide an excellent tracking performance if the system parameters are precisely known. The control equation applied to any manipulator in joint space can be established as follows [An et al., 1987]:

$$\Gamma(t) = \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d + \mathbf{K}_p \mathbf{e}_q(t) + \mathbf{K}_d \dot{\mathbf{e}}_q(t)) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \quad (2.30)$$

where  $\mathbf{K}_p, \mathbf{K}_d \in \mathbb{R}^{n \times n}$  are positive definite diagonal gains matrices for the proportional, and derivative actions respectively.  $\mathbf{e}_q(t) \in \mathbb{R}^n$ , and  $\dot{\mathbf{e}}_q(t) \in \mathbb{R}^n$  are the tracking errors or the joint positions and velocities, respectively. Substituting this control law in the inverse dynamic equation in joint space for a manipulator (2.19), leads the following closed-loop system dynamics:

$$\ddot{\mathbf{e}}_q(t) + \mathbf{K}_d \dot{\mathbf{e}}_q(t) + \mathbf{K}_p \mathbf{e}_q(t) = \mathbf{0} \quad (2.31)$$

In which  $\ddot{\mathbf{e}}_q$  represents the acceleration error in joint space. It is worth to note that the previous equation is valid only if all dynamic parameters are perfectly known. Moreover, as can be seen in (2.28), if the controller gains are chosen correctly, the tracking error can converge quickly towards zero with suitable steady-state performance. However, this control technique is computationally intensive, and the use of the measured variables as joint positions and velocities yields noise that may harm the system performance. In [Lou et al., 2011], the CTC was used to regulate the positioning of a 3-DOF PKM called Orthopod.

### Augmented PD Control

This control law is also known as PD+. It is composed of a PD feedback controller plus the inverse dynamic model equation of the robot evaluated with the desired trajectories and the measured ones. One key difference between it and the CTC is that the inertia matrix is outside the position and velocity feedback loops yielding a faster computation process [Paden and Panja, 1988]. The controller equation is relatively simple, and it can be expressed as:

$$\Gamma(t) = \mathbf{K}_p \mathbf{e}_q(t) + \mathbf{K}_d \dot{\mathbf{e}}_q(t) + \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{G}(\mathbf{q}) \quad (2.32)$$

where  $\mathbf{K}_p, \mathbf{K}_d \in \mathbb{R}^{n \times n}$  are positive definite diagonal gains matrices for the proportional, and derivative actions respectively.  $\mathbf{e}_q(t) \in \mathbb{R}^n$ , and  $\dot{\mathbf{e}}_q(t) \in \mathbb{R}^n$  are the tracking errors or the joint positions and velocities, respectively. The feedback PD controller guarantees asymptotic stability, and the model-based part cancels the effect of some nonlinearities. However, to this end, the dynamic parameters used in the equation must be relatively close to the real ones. The original PD+ has been modified in different ways to improve its features. For instance,



in [Müller and Hufnagel, 2012], a PD+ controller in redundant coordinates was proposed as an alternative to coordinate switching and applied to the control of redundantly actuated PKM.

### PD Computed Feedforward Control

This controller consists of a PD controller plus the nominal feedforward terms. Its expression is similar to the previous controller. However, the whole inverse dynamic model is evaluated with the desire trajectories instead of the measured ones since the sensors used to measure the actual joint variables may carry out the noise, which can impair robot performance. PD feedforward is one of the most used control techniques in robotic due to its simplicity, and it can provide asymptotic stability. The expression is given as follows [Santibañez and Kelly, 2001]:

$$\Gamma(t) = \mathbf{K}_p \mathbf{e}_q(t) + \mathbf{K}_d \dot{\mathbf{e}}_q(t) + \mathbf{M}(\mathbf{q}_d) \ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d) \dot{\mathbf{q}}_d + \mathbf{G}(\mathbf{q}_d) \quad (2.33)$$

where  $\mathbf{K}_p, \mathbf{K}_d \in \mathbb{R}^{n \times n}$  are positive definite diagonal gains matrices for the proportional, and derivative actions respectively.  $\mathbf{e}_q(t) \in \mathbb{R}^n$ , and  $\dot{\mathbf{e}}_q(t) \in \mathbb{R}^n$  are the tracking errors or the joint positions and velocities, respectively. In [Sartori-Natal et al., 2012], a modified version of the standard PD feedforward called “Dual-Space Control” was validated experimentally in an RA-PKM called R4. The difference between this control scheme from the standard PD feedforward is that the dynamic model part is formulated in both coordinate systems, Cartesian and joint space, and not only in one as happens with the original one formulation.

### Higher Order Sliding Mode Control

Sliding Mode Controllers (SMCs) are robust nonlinear control techniques appropriate for nonlinear uncertain systems. The main feature of these control schemes is the use of a discontinuous function with a high control gain leading finite-time convergence and robustness of the closed-loop system. However, these controllers have an inconvenience regarding the produced undesirable high-frequency oscillations called chattering [Utkin and Lee, 2006]. This unwanted phenomenon is too dangerous for the system actuators. Therefore, to overcome that issue, Higher-Order Sliding Mode Controllers (HOSMCs) have been proposed. Many methodologies based on HOSMCs have been proposed for the control of PKMs. For instance, in [Schindele and Aschemann, 2010], a trajectory control based on a HOSMC was implemented on a PKM with 2-DOF driven by pneumatic muscle actuators. In [Kumar and Bandyopadhyay, 2012], a HOSMC was proposed for the stabilization of a Stewart-platform. The velocities of such robots were estimated through an observer based on the super-twisting algorithm. Numerical simulations were performed to know the effectiveness of the controller.

## 2.5.2 Model-based-adaptive Controllers

Model-adaptive-based controllers emerge because, in reality, the dynamic parameters of a manipulator (as masses or inertia) are challenging to know with precision, and most of the time, some of such parameters vary according to the assigned task. Many of the control schemes belonging to this subcategory are improvements of those model-based with fixed

parameters. Therefore, the dynamic model representation must be rewritten, decoupling the dynamic parameters from known linear and nonlinear functions of the dynamic model. Considering (2.19), IDM is reformulated as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\Phi \quad (2.34)$$

In which  $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbb{R}^{n \times r}$  is known as the regressor matrix involving all linear and nonlinear functions of the model dynamics, and  $\Phi \in \mathbb{R}^r$  is the vector parameters, where the superscript  $r$  indicates the number of parameters. The right side of (2.34) will be considered in the description of the following control solutions.

### Adaptive Computed Torque Control

This extension of the classical CTC presented before arises because, in most cases, the dynamic parameters are not well known, or they are time-varying. Consequently, the presented nonlinearities of the system are not entirely canceled by the controller. Therefore, an estimator algorithm should be considered in order to estimate in real-time such unknown parameters. Taking into account (2.30), we can express the closed-loop system for Adaptive Computed Torque Control (ACTC) as [Craig et al., 1987]:

$$\ddot{\mathbf{e}}_q(t) = -\mathbf{K}_p \mathbf{e}_q(t) - \mathbf{K}_d \dot{\mathbf{e}}_q(t) + \hat{\mathbf{M}}^{-1}(\mathbf{q})\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\mathbf{e}_\Phi \quad (2.35)$$

In which,  $\hat{\mathbf{M}} \in \mathbb{R}^{n \times n}$  is the estimation of the manipulator inertia matrix,  $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$  is the regressor matrix evaluated with the measured joint variables, and  $\mathbf{e}_\Phi$  is the vector of parameter errors which is given as  $\mathbf{e}_\Phi = \Phi - \hat{\Phi}$  being  $\Phi$  the parameter vector, and  $\hat{\Phi}$  its estimated vector. The adaptation law is formulated using the following Lyapunov function candidate:

$$V(\boldsymbol{\varepsilon}_q, \mathbf{e}_\Phi) = \boldsymbol{\varepsilon}_q^T \mathbf{P} \boldsymbol{\varepsilon}_q + \mathbf{e}_\Phi^T \boldsymbol{\Xi}^{-1} \mathbf{e}_\Phi \quad (2.36)$$

Where  $\boldsymbol{\varepsilon}_q = [\dot{\mathbf{e}}_q \ \mathbf{e}_q]^T$ ,  $\mathbf{P}$  is a positive-definite matrix designed to satisfy the Lyapunov equation, and  $\boldsymbol{\Xi}$  is a positive-definite constant matrix defined for the adaptation law, which can be expressed as:

$$\dot{\hat{\Phi}} = \boldsymbol{\Xi} \mathbf{Y}^T \hat{\mathbf{M}}^{-1} \mathbf{r}(t) \quad (2.37)$$

Being  $\mathbf{r}(t)$  the combined tracking error in joint space  $\mathbf{r}(t) = \dot{\mathbf{e}}_q(t) + \boldsymbol{\Lambda} \mathbf{e}_q(t)$ , and  $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$  is a positive definite gain matrix. However, like its non-adaptive counterpart, ACTC has the drawback of using measured velocities and accelerations, which are tedious to estimate. This control scheme was applied in [Shang et al., 2012] to a RA 2-DOF planar parallel robot, the obtained performance was compared to a classical CTC. The results showed a considerable tracking performance improvement of ACTC compared to CTC. Figure 2.4 illustrates the control scheme differences between CTC and ACTC.

### PD with Adaptive Feedforward

PD control with adaptive feedforward is one of the simplest adaptive controllers-based that can be implemented on robotics. Two parts form the controller. On one hand, the PD feedback

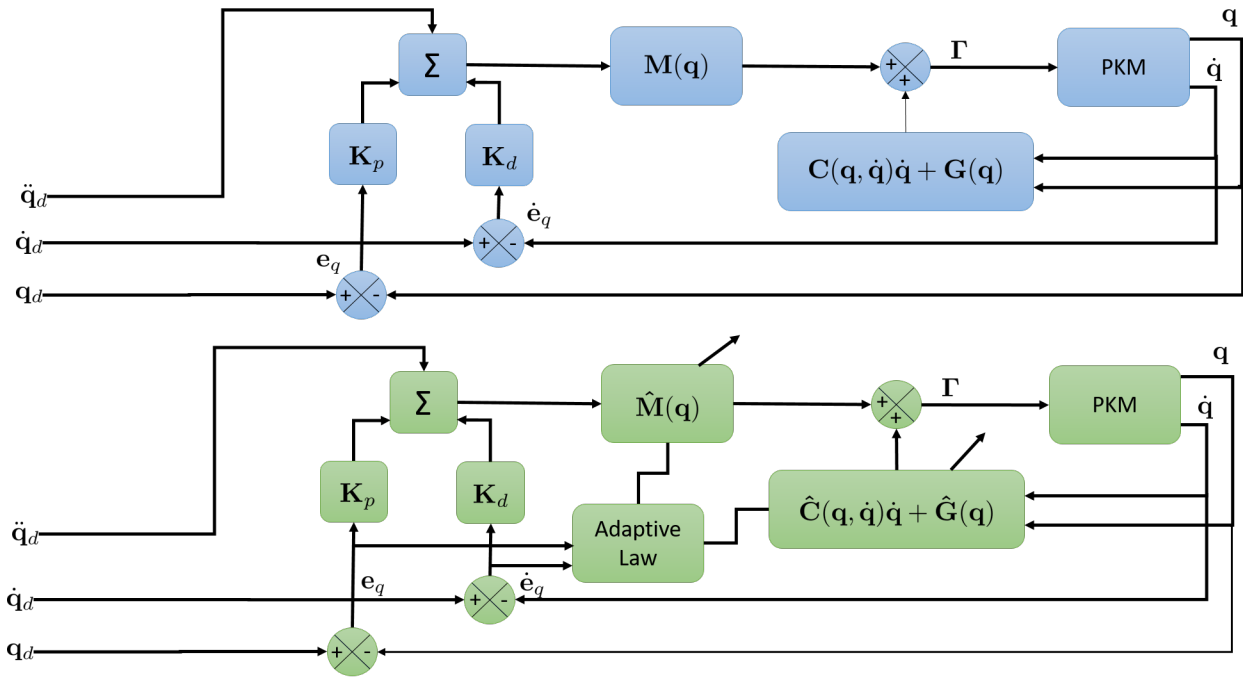


Figure 2.4: Block diagram comparison of standard CTC (blue) and ACTC (green)

controller keeps the system stability, and by the other hand, the adaptive feedforward term intends to cancel nonlinearities and reduce the effect produced by parametric uncertainties. For robotics, the mathematical expression of this controller can be defined as [Honegger et al., 1997]:

$$\Gamma(t) = \mathbf{K}_p \mathbf{e}_q(t) + \mathbf{K}_d \dot{\mathbf{e}}_q(t) + \mathbf{Y}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d) \hat{\Phi} \quad (2.38)$$

Where  $\mathbf{K}_p$  and  $\mathbf{K}_d \in \mathbb{R}^{n \times n}$  are positive-definite diagonal gains matrices for the proportional and derivative action, respectively,  $\hat{\Phi} \in \mathbb{R}^r$  is the vector of unknown dynamic parameters being  $r$  the number of its elements and  $\mathbf{Y}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d) \in \mathbb{R}^{n \times r}$ , is the regressor matrix. The unknown dynamic parameters are estimated according the following adaptation rule:

$$\dot{\hat{\Phi}} = \mathbf{K} \mathbf{Y}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)^T (\mathbf{K}_p \mathbf{e}_q(t) + \mathbf{K}_d \dot{\mathbf{e}}_q(t)) \quad (2.39)$$

Where  $\mathbf{K} \in \mathbb{R}^{r \times r}$  is a positive-definite diagonal matrix that must be carefully adjusted to obtain a good estimate of the parameters while maintaining the controller's stability. This control scheme was implemented in real-time to regulate the positioning of a 4-DOF PKM called Veloce. The performance obtained was compared to that produced by a PID controller. The results showed a vast superiority of the PD adaptive feedforward controller over the PID controller [Saied et al., 2019b].

### Desired Compensation Adaptive Control

Desired Compensation Adaptive Control (DCAL) was initially reported in [Sadegh and Horowitz, 1990], this model-based control scheme is formed by three parts, a decentralized

PD controller, an adaptive feedforward term, and an auxiliary nonlinear compensation term. The controller equation can be written in the following form [Bennehar et al., 2016b]:

$$\mathbf{\Gamma}(t) = \mathbf{K}_p \mathbf{e}_q(t) + \mathbf{K}_v \mathbf{r}(t) + \mathbf{Y}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d) \hat{\mathbf{\Phi}} + \vartheta \|\mathbf{e}_q(t)\|^2 \mathbf{r}(t) \quad (2.40)$$

Where  $\mathbf{K}_p, \mathbf{K}_v \in \mathbb{R}^{n \times n}$  are positive definite diagonal gains matrices,  $\mathbf{r}(t) \in \mathbb{R}^n$  is the combined tracking error in joint space formulated as  $\mathbf{r}(t) = \dot{\mathbf{e}}_q(t) + \mathbf{\Lambda} \mathbf{e}_q(t)$ . The parameter estimation vector can be formulated by the following expression:

$$\dot{\hat{\mathbf{\Phi}}} = \mathbf{K} \mathbf{Y}^T(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d) \mathbf{r}(t) \quad (2.41)$$

Being  $\mathbf{K} \in \mathbb{R}^{n \times n}$  a positive-definite adaptation gain matrix. One can highlight the following advantages of the DCAL controller: Reduced computing time thanks to using the desired joint trajectories instead of the measured joint variables in the regressor matrix and the parameters adaptation rule, noise effect reduction, and compensation for parametric uncertainties and nonlinearities. In [Bennehar et al., 2014a], DCAL control was applied to an RA-PKM called dual-V. In the same work, the fixed gains for the feedback controller part were substituted by time-varying nonlinear ones in order to improve the overall performance of the parallel robot.

### Dual-mode Adaptive Control

This control scheme was implemented on Par2 PKM, it consists of three terms: an adaptive feedforward term, a smooth variable structure term, and a stabilizing term. Dual-mode adaptive control equation can be expressed as follows [Natal et al., 2016]:

$$\mathbf{\Gamma}(t) = \mathbf{K} \mathbf{r}(t) + \bar{d} \frac{\alpha_s \mathbf{r}(t)}{\|\alpha_s \mathbf{r}(t)\| + 1} + \mathbf{Y}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d) \hat{\mathbf{a}} \quad (2.42)$$

Where  $\mathbf{K} \in \mathbb{R}^{n \times n}$  is a positive-definite gain matrix,  $\alpha$ , and  $\bar{d}$  are positive constants.  $\hat{\mathbf{a}}$  represents an estimation of the unknown parameters of the system  $\hat{\mathbf{a}}_{nom}$ . The adaptation rule is given as follows:

$$\dot{\hat{\mathbf{\Phi}}} = -\gamma \text{Proj}(\mathbf{r}(t)^T \mathbf{Y}, \hat{\mathbf{\Phi}}) = -\sigma_r \hat{\mathbf{\Phi}} - \gamma \mathbf{r}(t)^T \quad (2.43)$$

In which,  $\hat{\mathbf{\Phi}}$  is the difference between the currently estimated values and the nominal values of the parameters,  $\gamma$  is the adaptation gain constant, and  $\sigma_r$  is a variable adjusted as follows:

$$\sigma_r = \begin{cases} 0 & \text{if } \|\hat{\mathbf{\Phi}}\| < M_{\Phi} \quad \text{or} \quad \sigma_{eq} < 0 \\ \sigma_{eq} & \text{if } \|\hat{\mathbf{\Phi}}\| \geq M_{\Phi} \quad \text{or} \quad \sigma_{eq} \geq 0 \end{cases} \quad (2.44)$$

$$\sigma_{eq} = -\frac{\gamma \mathbf{r}(t)^T \mathbf{Y} \hat{\mathbf{\Phi}}}{\|\hat{\mathbf{\Phi}}\|^2} \quad (2.45)$$

Where  $M_{\Phi}$  is the maximum possible value of the inertia matrix, and  $\|\hat{\mathbf{\Phi}}\|^2$  is the 2-norm of  $\hat{\mathbf{\Phi}}$ . According to [Natal et al., 2009], the adaptation law of Dual-Mode adaptive control provides some advantages concerning other robust control algorithms as the generation of continuous control signals, improvement in system robustness, less sensitivity to measurement noises. Moreover, for large tracking errors, the controller behaves like a sliding mode controller generating exponential convergence to a residual domain arbitrarily small, and to smaller errors, it behaves as a parametric adaptation law.

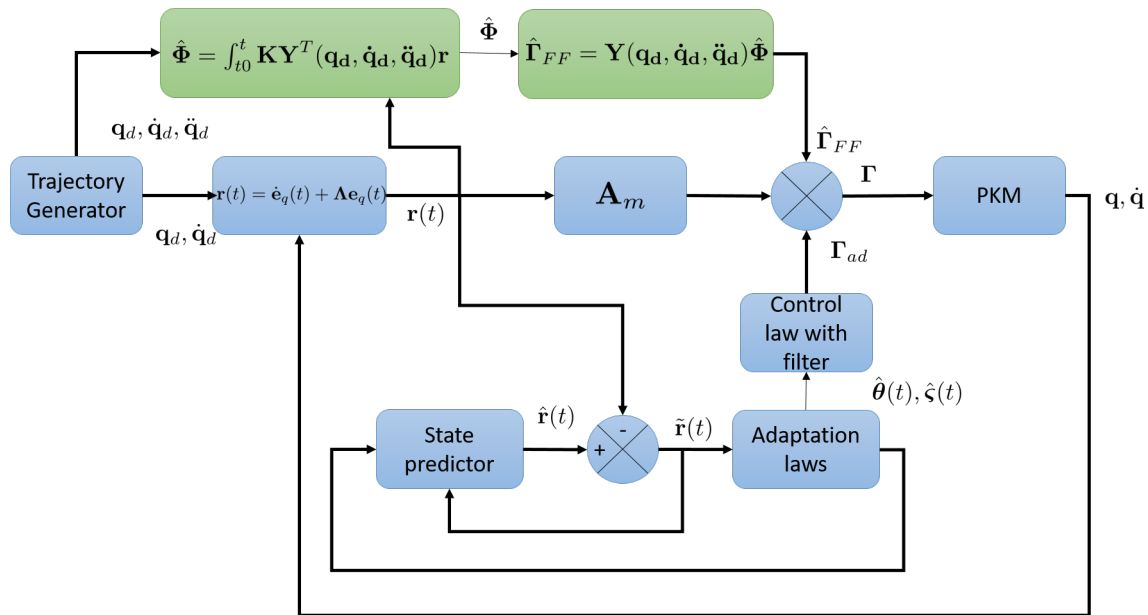


Figure 2.5: Block diagram of L1 adaptive controller for PKM (blue), plus adaptive feedforward compensation (green)

### $\mathcal{L}_1$ Adaptive Control with adaptive Feedforward

Although the original  $\mathcal{L}_1$  adaptive controller does not require knowledge of the dynamics of the system, there are still nonlinearities and uncertainties in which the adaptive compensation term  $\Gamma_{ad}(t)$  has not been able to eliminate. In order to improve the capabilities of the original  $\mathcal{L}_1$  adaptive controller, in [Bennehar et al., 2015b], it was proposed to integrate a model-based adaptive feedforward term used to diminish further the effect of such nonlinearities and uncertainties of the system. Thus, (2.25) is extended as:

$$\Gamma(t) = \mathbf{A}_m \mathbf{r}(t) + \Gamma_{ad}(t) + \mathbf{Y}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d) \hat{\Phi} \quad (2.46)$$

Where  $\mathbf{Y}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d) \in \mathbb{R}^{n \times r}$ , is the regressor matrix, and  $\hat{\Phi} \in \mathbb{R}^r$ , is the vector of estimated dynamic parameters whose elements are computed using the adaptation rule presented in (2.37). This control law was validated experimentally on the RA-4-DOF Arrow PKM. The obtained results showed an improvement in the trajectory tracking of more than 78% compared to the original  $\mathcal{L}_1$  adaptive controller. Figure 2.5 presents the control diagram of standard  $\mathcal{L}_1$  adaptive controller in color blue plus the adaptive-model-based term proposed in [Bennehar et al., 2015b] in green color.

### Adaptive Terminal Sliding Mode Control

Terminal Sliding Mode Controllers (TSMCs) are a kind of HOSMCs with finite-time convergence, maintaining the robustness of conventional SMCs. The first designs of TSMCs had a singularity issue stemming from the design of the sliding manifold [Venkataraman and Gulati, 1992]. Nevertheless, in some research works, it has been proposed solutions to deal

with such a problem re-designing the sliding manifold [Feng et al., 2002]. Considering the features providing by TSMCs and adaptive controllers. In [Bennehar et al., 2017], a novel adaptive TSMC was proposed for motion control of PKMs where an adaptation loop on-line estimates the unknown, uncertain, and time-varying parameters of the model-based control loop. The control law can be expressed as:

$$\Gamma(t) = \mathbf{Y}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d) \hat{\Phi} - \mathbf{K}_1 \mathbf{s} - \mathbf{K}_2 |\mathbf{s}|^\rho \text{sgn}(\mathbf{s}) \quad (2.47)$$

In which,  $\mathbf{Y}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d) \in \mathbb{R}^{n \times r}$ , is the regressor matrix evaluated with the desired joint trajectories, and  $\hat{\Phi} \in \mathbb{R}^r$ , is the vector of estimated dynamic parameters,  $\mathbf{K}_1, \mathbf{K}_2 \in \mathbb{R}^{n \times n}$  are positive-definite diagonal gain matrices, and  $\rho$  is an auxiliary control term. The sliding manifold is designed as  $\mathbf{s} = \mathbf{e}_q + \beta |\dot{\mathbf{e}}_q|^\gamma \text{sgn}(\dot{\mathbf{e}}_q)$ , being  $0 < \gamma < 2$  and  $\beta > 0$  control design parameters. The unknown parameters are estimated according the following adaptation rule

$$\hat{\Phi} = -\Xi(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)^T (\mathbf{K}_3 \mathbf{s} + \mathbf{K}_4 |\mathbf{s}|^\rho \text{sgn}(\mathbf{s})) \quad (2.48)$$

Where  $\Xi \in \mathbb{R}^{r \times r}$  is a positive-definite diagonal gain matrix known as the adaptation gain matrix, and  $\mathbf{K}_3, \mathbf{K}_4 \in \mathbb{R}^{n \times n}$  are design matrices for the adaptation law. This adaptive TSMC was validated with real-time experiments on a PKM called Veloce in [Bennehar et al., 2017]. The obtained tracking performance was compared to the produced by nominal TSMC, obtaining an improvement of more than 50%.

## 2.6 Conclusion

This chapter presented a summary of the three main formulations to derive the motion equations for any parallel robot, including a simplified formulation appearing in the literature for PKMs with a delta-like structure that makes use of NE equations and the principle of virtual works. This formulation aims to generate a suitable inverse dynamic model for real-time implementations in the controller loop. Moreover, this chapter presented an overview of existing control solutions already implemented on PKMs. Such control solutions were classified into two principal categories Non-model-based and model-based controllers. Both categories are also divided into two branches non-adaptive and adaptive controllers. One can highlight that non-model-based non-adaptive controllers are easy and straightforward for real-time implementation. However, according to the literature, their performance is poor when the robot is required to operate at high speeds/accelerations and uncertainties present due to the operating environment. On the other hand, model-based-adaptive controllers may present the best tracking performance compared to other control categories mentioned before due to their very elaborate design. Nevertheless, the tuning of their gains is not easy, owing to the presence of multiple control elements.

**Part II**

**Modeling and Proposed Control  
Solutions**





## Description and Modeling of Experimental Platforms

*In this chapter, the analysis concerning kinematic and dynamic analysis for two PKMs with delta-like architecture is introduced. The robots are the SPIDER4 RA-PKM with 3T-2R-DOF and a Delta PKM with 3T-DOF. The first one is a redundantly actuated PKM, as long as the last one is a fully non-redundant PKM. The inverse kinematic modeling for both manipulators is obtained through the closed-loop equation being formulated by a geometric analysis of the kinematic chains. The forward kinematic model for both PKMs is obtained solving the intersection of virtual spheres whose radii are related with the forearms' length. The Jacobian matrices are derived for the geometric analysis of the PKMs. Finally, the IDM for both PKMs is presented, where some hypothesis simplifications were considered.*

### Contents

<b>3.1</b>	<b>Introduction</b>	<b>49</b>
<b>3.2</b>	<b>Redundant PKM (SPIDER4)</b>	<b>50</b>
<b>3.3</b>	<b>Non-Redundant PKM (Delta)</b>	<b>63</b>
<b>3.4</b>	<b>Conclusion</b>	<b>70</b>

### 3.1 Introduction

Modeling is a crucial part of the study of any robotics system since it allows us to know the behavior of the mechanism and how the different mechanical parameters affect the system's overall performance. Additionally, it helps in the design of control schemes to precisely regulate the robot movement. The kinematic analysis studies the geometry of a robot's motion, but without considering the forces or torques that produced the motion [Taghirad, 2013]. The mostly kinematic models in PKMs have developed from geometric analysis of the kinematic chains resulting in an expression involving the joint and Cartesian variables called the closed-loop equation [Merlet, 2006]. Obtaining the IKM in PKM is a relatively easy and

straightforward task, unlike serial manipulators. However, obtaining the FKM is a complex task for the majority of PKMs configurations because it involves highly coupled nonlinear equations with a not unique solution. The aim of this chapter is the development of the different kinematic and dynamic models for two PKMs. One is a RA-PKM called SPIDER4, and the other robot is a non-RA-PKM (the Delta robot). As was stated in Chapter 1, one of the main thesis objectives is the modeling of SPIDER4, so we will first focus on developing the kinematic and dynamic models of this platform, then we will present the modeling equations for the 3-DOF Delta PKM.

## 3.2 Redundant PKM (SPIDER4)

### 3.2.1 Description and Modeling of SPIDER4

SPIDER4 is a novel RA-PKM designed and manufactured in collaboration between The Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM), and TecNALIA company. This machine is conceived for CNC machining tasks of resin materials. It has 5-DOF: 3 translations denoted by  $(x, y, z)$  and 2 rotations denoted by  $(\phi, \psi)$ . The translational motion is performed by the parallel structure consisting of four kinematic chains, each one formed by one arm and one forearm linked by a passive universal joint. Each arm is connected directly to an electric motor located in the fixed base, and the forearm is connected to the moving platform through two parallel bars, in a PKM Delta-like architecture. The rotational motions are generated by an independent serial wrist mechanism placed on the nacelle, which is actuated through two electric motors. The machining spindle is placed at the end of this mechanism. Figure 3.1 shows SPIDER4 RA-PKM and its kinematic configuration diagram, where the gray boxes represent the active rotational joints, and the white boxes represent the passive universal joints. To develop a machining task, SPIDER4 makes use of a tooling plate to fix the raw material. The material's maximum dimensions must be a cylinder of 700mm diameter and 300mm depth. The machine and its tooling plate are inside an enclosure for operator safety; its dimensions are 3.9x2.5x2.4 m (LxWxH), as Figure 3.2 illustrates. It is worth to mention that SPIDER4 can be equipped with linear encoders to measure directly the position of the nacelle from the origin of the fixed base. However, for the control tasks developed in this research, such sensors will not be considered; only the rotary encoders integrated with the motors will be used. The different elements of composing SPIDER4 are described in detail in Figure 3.3.

### 3.2.2 Inverse Kinematic Model of SPIDER4 RA-PKM

Let us start with the development of the IKM for SPIDER4, which consists of finding the generalized coordinates vector  $\mathbf{Q} = [q_1 \ q_2 \ q_3 \ q_4 \ \phi \ \psi]^T \in \mathbb{R}^{6 \times 1}$  given the spindle position in the fixed reference frame  $O - x_o, y_o, z_o$  as  ${}^o\mathbf{S}_S = [x \ y \ z \ \phi \ \psi]^T \in \mathbb{R}^{5 \times 1}$ . It is worth mentioning that the variables  $\phi, \psi$  are the same for operational and joint spaces; in addition,

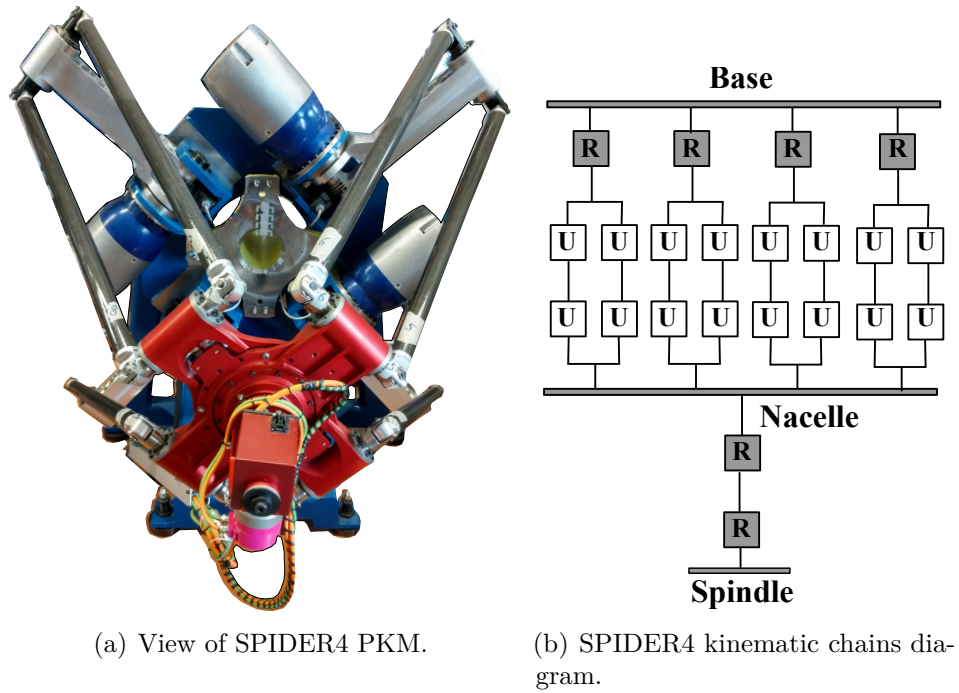


Figure 3.1: The SPIDER4 RA-PKM and its kinematics

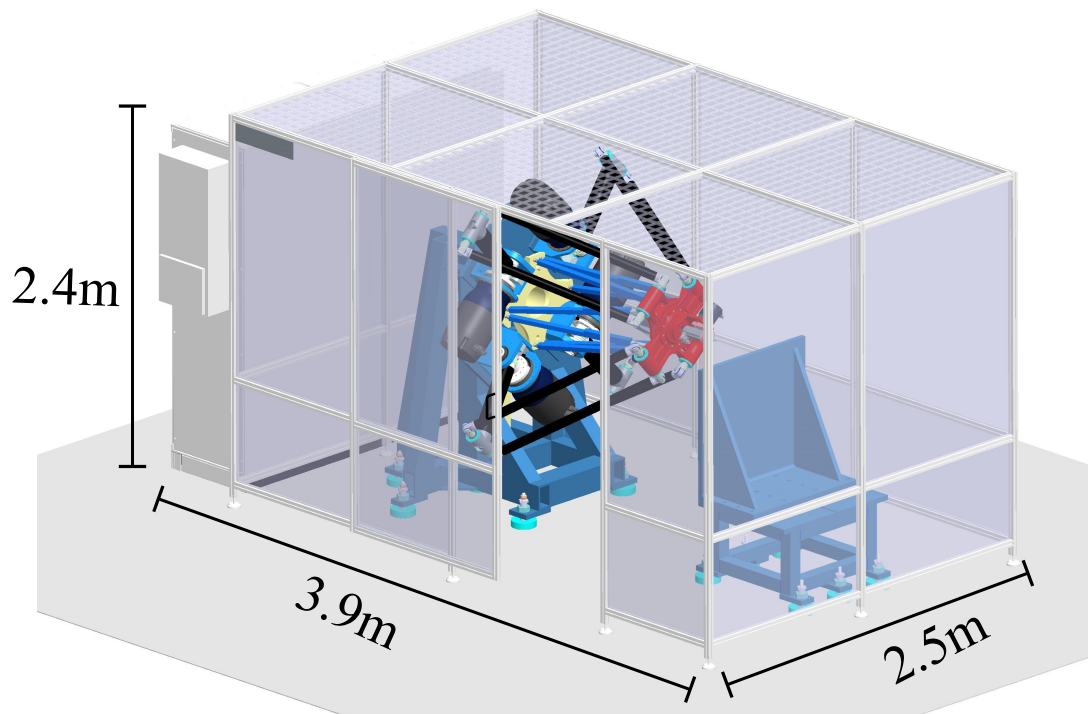


Figure 3.2: Enclosure dimensions of the working area of SPIDER4

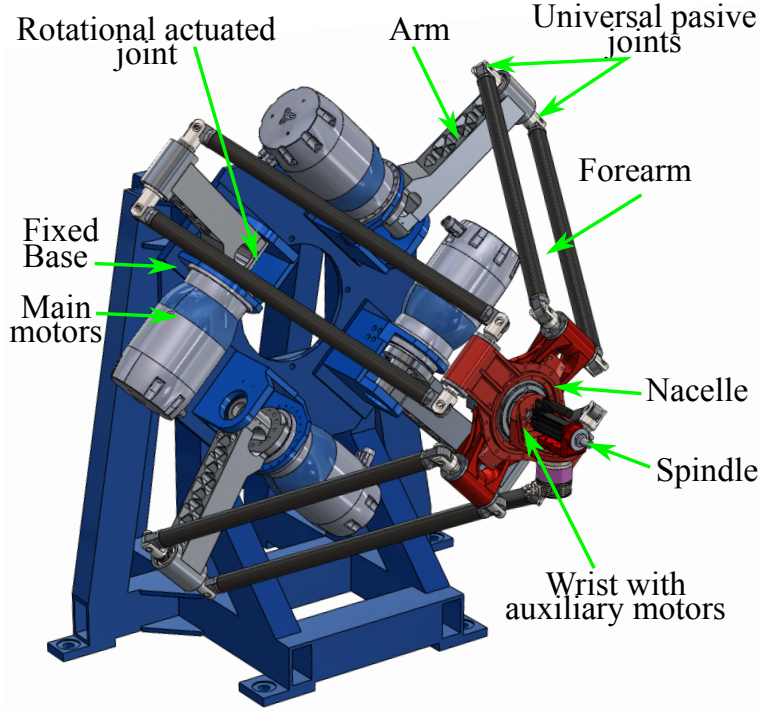


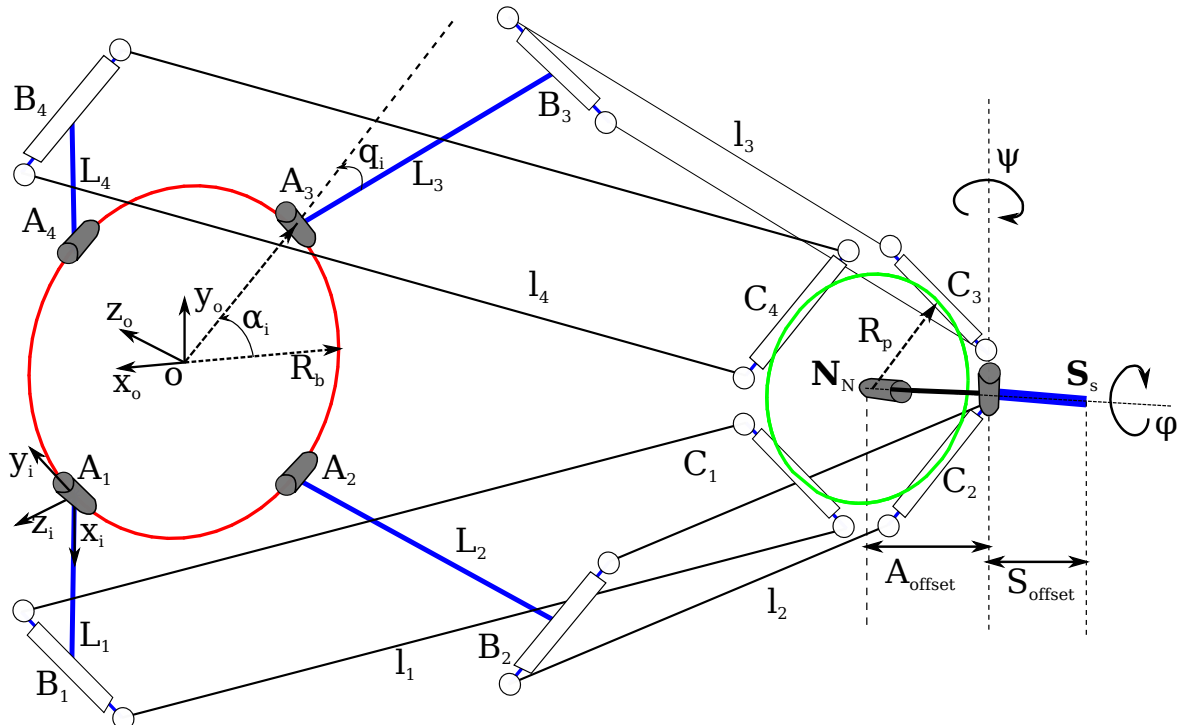
Figure 3.3: CAD View of SPIDER4 with its main components description

it is possible to define a vector which only involves the SPIDER4 parallel structure joint variables  $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T \in \mathbb{R}^{4 \times 1}$ . SPIDER4 shares several similarities with the Delta PKM; in fact, their kinematic composition for both PKMs is composed of one rear-arm and one forearm formed by two parallel bars. Nevertheless, SPIDER4 has a wrist mechanism on the nacelle. Therefore, before to formulate the kinematic equations of the delta-like positioning device, it is indispensable to develop the kinematic relations of the wrist to get the position of the nacelle denoted as  ${}^o\mathbf{N}_N = [x_n \ y_n \ z_n]^T \in \mathbb{R}^{3 \times 1}$ . Considering the kinematic diagram of SPIDER4 depicted in Figure 3.4, we establish the following rotation matrices to derive the kinematics relationships of the wrist mechanism.

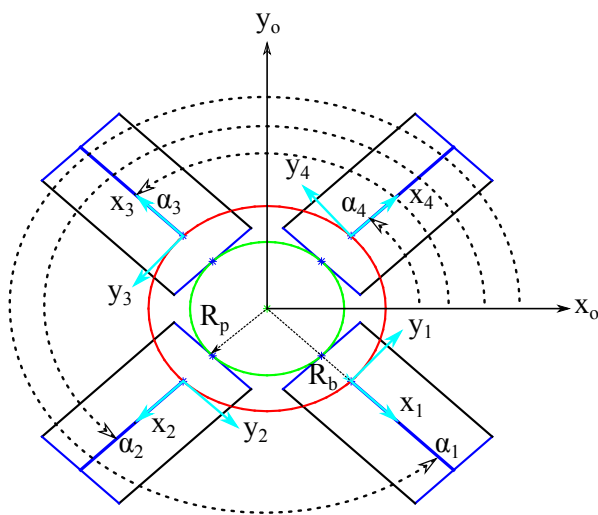
$$\mathbf{R}_Z = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R}_Y = \begin{bmatrix} \cos(\psi) & 0 & \sin(\psi) \\ 0 & 1 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) \end{bmatrix} \quad (3.1)$$

The above expressions involve the wrist mechanism variables  $(\phi, \psi)$ , consequently  ${}^o\mathbf{N}_N$  can be defined as follows:

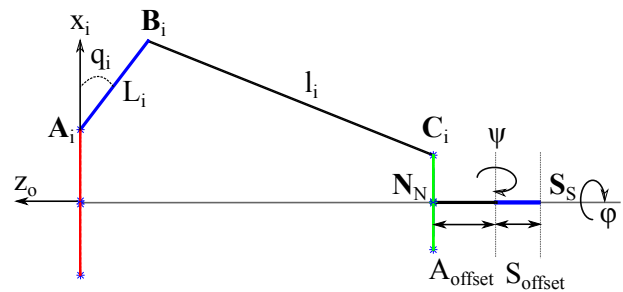
$${}^o\mathbf{N}_N = \begin{bmatrix} x \\ y \\ z + A_{offset} \end{bmatrix} + \mathbf{R}_Z \mathbf{R}_Y \begin{bmatrix} 0 \\ 0 \\ S_{offset} \end{bmatrix} \quad (3.2)$$



(a) Isometric view



(b) Front view



(c) Lateral view

Figure 3.4: Kinematics illustration of the SPIDER4 RA-PKM

Reducing the equation (3.2)  ${}^o\mathbf{N}_N$  results in:

$${}^o\mathbf{N}_N = \begin{bmatrix} x + S_{offset} \cos(\phi) \sin(\psi) \\ y + S_{offset} \sin(\phi) \sin(\psi) \\ z + S_{offset} \cos(\psi) + A_{offset} \end{bmatrix} \quad (3.3)$$

where  $S_{offset}$  is the distance between the  $\psi$  joint axis and the tip of the spindle, and  $A_{offset}$  is the distance between the center of the nacelle and the  $\psi$  joint axis. Once  ${}^o\mathbf{N}_N$  is computed, we can set the active and passive joints locations with respect to the Cartesian-fixed frame  $O - x_o, y_o, z_o$ . The actuated joints located in the points  $\mathbf{A}_i, \forall i = 1..4$ , are expressed respecting the fixed reference frame by the following vector:

$${}^o\mathbf{A}_i = R_b \begin{bmatrix} \cos(\alpha_i) & \sin(\alpha_i) & 0 \end{bmatrix}^T \quad (3.4)$$

Where  $R_b$  is the fixed-base radius. The four main actuators are placed on the fixed base with the following angles with respect to the fixed Cartesian frame  $\boldsymbol{\alpha} = \left[ \frac{7\pi}{4} \quad \frac{5\pi}{4} \quad \frac{3\pi}{4} \quad \frac{\pi}{4} \right]^T$ . The universal passive joints are located at the points  $\mathbf{B}_i$  and  $\mathbf{C}_i$  whose coordinates are expressed in the fixed reference frame  $O - x_o, y_o, z_o$  as follows:

$${}^o\mathbf{B}_i = {}^o\mathbf{A}_i + L \begin{bmatrix} \cos(\alpha_i) \cos(q_i) & \sin(\alpha_i) \cos(q_i) & -\sin(q_i) \end{bmatrix}^T \quad (3.5)$$

$${}^o\mathbf{C}_i = \begin{bmatrix} R_p \cos(\alpha_i) + x_n & R_p \sin(\alpha_i) + y_n & z_n \end{bmatrix}^T \quad (3.6)$$

being  $L$  the arm length and  $R_p$  the nacelle radius. For further analysis an auxiliary frame located at  ${}^o\mathbf{A}_i - x_i, y_i, z_i$  is defined, where two auxiliary vectors  ${}^i\mathbf{x}_i$  and  ${}^i\mathbf{y}_i$  are defined as:

$${}^i\mathbf{x}_i = \begin{bmatrix} \cos(\alpha_i) & \sin(\alpha_i) & 0 \end{bmatrix}^T \quad (3.7)$$

$${}^i\mathbf{y}_i = \begin{bmatrix} -\sin(\alpha_i) & \cos(\alpha_i) & 0 \end{bmatrix}^T \quad (3.8)$$

The expression relating the operational nacelle variables  ${}^o\mathbf{N}_N$  with the parallel structure joint variables  $\mathbf{q}$  is called the close-loop equation, which can be defined as:

$$\|{}^o\mathbf{C}_i - {}^o\mathbf{B}_i\|^2 = l_i^2 \quad (3.9)$$

According to [Pierrot, 1991], [Clavel, 1991] the mathematical expression (3.9) can be rewritten as follows in order to obtain the values of the joint variables  $\mathbf{q}$ .

$$D_i \sin(q_i) + E_i \cos(q_i) + F_i = 0 \quad \forall i = 1, 2, 3, 4 \quad (3.10)$$

where  $D_i = 2L_i({}^o\mathbf{A}_i^o\mathbf{C}_i \cdot \mathbf{z}_o)$ ,  $E_i = 2L_i({}^o\mathbf{A}_i^o\mathbf{C}_i \cdot {}^i\mathbf{x}_i)$ , and  $F_i = l_i^2 - L_i^2 - \|{}^o\mathbf{A}_i^o\mathbf{C}_i\|^2$ . The values of  $q_i$  are obtained by solving for (3.10), resulting in the following expression:

$$q_i = \arctan \left( \frac{-D_i \pm \sqrt{\Delta_i}}{F_i - E_i} \right) \quad (3.11)$$

Being  $\Delta_i = D_i^2 + E_i^2 - F_i^2$ . The kinematic parameters of SPIDER4 as lengths of the limbs, the radius of the fixed base, and nacelle were obtained from the CAD model provided by Tecnalia Company. Thereby, the kinematic parameters of SPIDER4 are known with enough precision. These parameters are presented in Table 3.1.

Table 3.1: Summary of the kinematic parameters of SPIDER4

Parameter	Description	Value
$L$	Rear-arm length	0.535 $m$
$l$	Forearm length	1.100 $m$
$R_b$	Fixed base radius	0.4 $m$
$R_p$	Nacelle radius	0.260 $m$
$S_{offset}$	Distance between $\psi$ and ${}^o\mathbf{S}_S$	1.135 $m$
$A_{offset}$	Distance between ${}^o\mathbf{N}_N$ and $\psi$	0.198375 $m$

### 3.2.3 Forward Kinematic Model of SPIDER4 RA-PKM

Generally, obtaining the FKM in PKMs is a complex task since it requires to solve multiple coupled nonlinear algebraic equations yielding multiple solutions. Notwithstanding, for PKMs with delta-like architecture, the FKM solution is relatively straightforward owing to the configuration based-on parallelograms restricting the orientation of the traveling plate allowing linear translation motion only [Taghirad, 2013]. The FKM solution for this kind of configuration is finding the intersection of  $n$ th virtual spheres whose radii are equal to the forearms' length [Williams II, 2016]. For SPIDER4, the FKM is computed in two steps; the first is to find  ${}^o\mathbf{N}_N \in \mathbb{R}^{3 \times 1}$  when  $\mathbf{q} \in \mathbb{R}^{4 \times 1}$  is given; in other words, we have to compute the FKM of the delta-like positioning device. The second step lies in computing the spindle position  ${}^o\mathbf{S}_S$  once  ${}^o\mathbf{N}_N$  has been found. Figure 3.5 illustrates the intersection algorithm of four virtual spheres to obtain the FKM of SPIDER4. According to this figure, one can see that this method leads to two solutions; however, it is easy to choose the appropriate solution for the FKM taking into account the robot's geometry. The virtual spheres centers for SPIDER4 are established by means the following kinematic expression:

$${}^o\mathbf{B}'_i = {}^o\mathbf{B}_i - {}^N\mathbf{N}_i \quad \forall i = 1, \dots, 4 \quad (3.12)$$

Where

$${}^N\mathbf{N}_i = [R_p \cos(\alpha_i) \quad R_p \sin(\alpha_i) \quad 0]^T \quad (3.13)$$

The equations of the four virtual spheres to obtain the FKM of the SPIDER4 delta-like positioning device are given as follows:

$$\begin{aligned} x_n^2 + y_n^2 + z_n^2 - 2x_1x_n - 2y_1y_n - 2z_1z_n + w_1 &= l^2 \\ x_n^2 + y_n^2 + z_n^2 - 2x_2x_n - 2y_2y_n - 2z_2z_n + w_2 &= l^2 \\ x_n^2 + y_n^2 + z_n^2 - 2x_3x_n - 2y_3y_n - 2z_3z_n + w_3 &= l^2 \\ x_n^2 + y_n^2 + z_n^2 - 2x_4x_n - 2y_4y_n - 2z_4z_n + w_4 &= l^2 \end{aligned} \quad (3.14)$$

In which,  $x_i, y_i, z_i \quad \forall i = 1, \dots, 4$ , represent the coordinates of the center of each virtual sphere, and  $w_i = x_i^2 + y_i^2 + z_i^2 \quad \forall i = 1, \dots, 4$ . The solution of the equations' system (3.14) that corresponds to the nacelle position is given as follows:

$$x_n = d_2z_n + e_2 \quad (3.15)$$



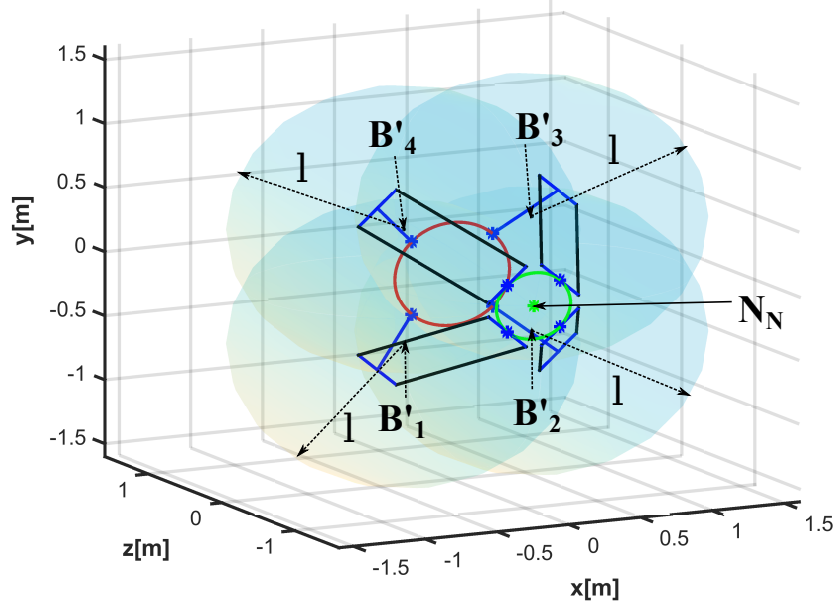


Figure 3.5: Illustration of the four virtual spheres intersection used to find  ${}^o\mathbf{N}_N$  position

$$y_n = d_1 z_n + e_1 \quad (3.16)$$

$$z_n = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (3.17)$$

Where the auxiliary constants that are involved in expressions (3.15)-(3.17) are defined as follows:

$$\begin{aligned} a_1 &= \frac{y_1 - y_3}{x_3 - x_1} & b_1 &= \frac{z_1 - z_3}{x_3 - x_1} & c_1 &= \frac{w_3 - w_1}{2(x_3 - x_1)} & d_1 &= \frac{b_2 - b_1}{a_1 - a_2} & e_1 &= \frac{c_2 - c_1}{a_1 - a_2} \\ a_2 &= \frac{y_2 - y_3}{x_3 - x_1} & b_2 &= \frac{z_2 - z_3}{x_3 - x_2} & c_2 &= \frac{w_3 - w_2}{2(x_3 - x_2)} & d_2 &= a_1 d_1 + b_1 & e_2 &= a_1 e_1 + c_1 \end{aligned}$$

$$\begin{aligned} a &= d_1^2 + d_2^2 + 1 \\ b &= 2[(e_1 - y_4)d_1 + (e_2 - x_4)d_2 - z_4] \\ c &= e_1^2 + e_2^2 - 2(e_1 y_4 + e_2 x_4) + w_4 - l^2 \end{aligned}$$

Once the positions of the nacelle  $(x_n, y_n, z_n)$  are found, an auxiliary vector is defined in order to find the spindle position; the auxiliary vector involves the variables of the wrist mechanism  $(\phi, \psi)$ , and is expressed as:

$${}^o\mathbf{N}_{aux} = {}^o\mathbf{N}_N - \begin{bmatrix} S_{offset} \cos(\phi) \sin(\psi) \\ S_{offset} \sin(\phi) \sin(\psi) \\ S_{offset} \cos(\psi) + A_{offset} \end{bmatrix} \quad (3.18)$$

Finally, the spindle position is defined through the following expression:

$${}^o\mathbf{S}_S = [{}^o\mathbf{N}_{aux}^T \quad \phi \quad \psi]^T \quad (3.19)$$



### 3.2.4 Velocity Relationship and Jacobian Analysis of SPIDER4 RA-PKM

The Jacobian matrices for SPIDER4 are generated, taking into account only the parallel structure mechanism since the wrist mechanism's velocities are the same for the Cartesian and joint space. The Jacobian matrices allow us to map the actuated joints' velocities based on the nacelle's velocities. Having in mind that  $\|{}^o\mathbf{C}_i - {}^o\mathbf{B}_i\|^2 = BC_{ix}^2 + BC_{iy}^2 + BC_{iz}^2$  equation (3.9) can be rewrite as

$$\begin{aligned} x_n^2 - 2Lx \cos \alpha_i \cos q_i + L^2 \cos^2 \alpha_i \cos^2 q_i - 2aL \cos \alpha_i \cos q_i + a^2 + \\ y_n^2 - 2Ly \sin \alpha_i \cos q_i + L^2 \sin^2 \alpha_i \cos^2 q_i - 2bL \sin \alpha_i \cos q_i + b^2 + \\ z_n^2 + 2Lz \sin q_i + L^2 \sin^2 q_i = l \end{aligned} \quad (3.20)$$

where  $a = R_p \cos \alpha_i - R_b \cos \alpha_i$  and  $b = R_p \sin \alpha_i - R_b \sin \alpha_i$ . The derivative of (3.20) with respect to time yields the general equation of velocities' relationship expressed in this form:

$$\begin{aligned} ((R_p - R_b) \cos \alpha_i + x - L \cos \alpha_i \cos q_i) \dot{x} + ((R_p - R_b) \sin \alpha_i + y - L \sin \alpha_i \cos q_i) \dot{y} + \\ (z + L \sin q_i) \dot{z} = -L((x \cos \alpha_i + y \sin \alpha_i + R_p - R_b) \sin q_i + z \cos q_i) \dot{q}_i \quad \forall i = 1, \dots, 4 \end{aligned} \quad (3.21)$$

The terms of the previous expression are regrouped with respect to the velocities in Cartesian space and the velocities in joint space in order to obtain the Jacobian matrices  $\mathbf{J}_x \in \mathbb{R}^{4 \times 3}$ , and  $\mathbf{J}_q \in \mathbb{R}^{4 \times 4}$ . These matrices are represented as:

$$\mathbf{J}_x = \begin{bmatrix} R \cos \alpha_1 + x_n - L \cos \alpha_1 \cos q_1 & R \sin \alpha_1 + y_n - L \sin \alpha_1 \cos q_1 & z_n + L \sin q_1 \\ R \cos \alpha_2 + x_n - L \cos \alpha_2 \cos q_2 & R \sin \alpha_2 + y_n - L \sin \alpha_2 \cos q_2 & z_n + L \sin q_2 \\ R \cos \alpha_3 + x_n - L \cos \alpha_3 \cos q_3 & R \sin \alpha_3 + y_n - L \sin \alpha_3 \cos q_3 & z_n + L \sin q_3 \\ R \cos \alpha_4 + x_n - L \cos \alpha_4 \cos q_3 & R \sin \alpha_4 + y_n - L \sin \alpha_4 \cos q_4 & z_n + L \sin q_4 \end{bmatrix} \quad (3.22)$$

$$\mathbf{J}_q = \begin{bmatrix} Jq_{11} & 0 & 0 & 0 \\ 0 & Jq_{22} & 0 & 0 \\ 0 & 0 & Jq_{33} & 0 \\ 0 & 0 & 0 & Jq_{44} \end{bmatrix} \quad (3.23)$$

Where

$$\begin{aligned} Jq_{11} &= -L((x_n \cos \alpha_1 + y_n \sin \alpha_1 + R) \sin q_1 + z_n \cos q_1) \\ Jq_{22} &= -L((x_n \cos \alpha_2 + y_n \sin \alpha_2 + R) \sin q_2 + z_n \cos q_2) \\ Jq_{33} &= -L((x_n \cos \alpha_3 + y_n \sin \alpha_3 + R) \sin q_3 + z_n \cos q_3) \\ Jq_{44} &= -L((x_n \cos \alpha_4 + y_n \sin \alpha_4 + R) \sin q_4 + z_n \cos q_4) \end{aligned}$$

Therefore the general Jacobian matrix of a PKM may be written as:

$$\mathbf{J} = \mathbf{J}_q^{-1} \mathbf{J}_x \in \mathbb{R}^{4 \times 3} \quad (3.24)$$

The joint velocities of the SPIDER4 RA-PKM  $\dot{\mathbf{q}} \in \mathbb{R}^{4 \times 1}$  can be computed from the Cartesian velocities of the nacelle  $\dot{\mathbf{N}} \in \mathbb{R}^{3 \times 1}$  through the general Jacobian matrix  $\mathbf{J} \in \mathbb{R}^{4 \times 3}$  in this way

$$\dot{\mathbf{q}} = \mathbf{J}^o \dot{\mathbf{N}}_N \quad (3.25)$$

However, to find the Cartesian velocities of the nacelle in function of the joint velocities it so necessary to compute the pseudoinverse Jacobian matrix  $\mathbf{H}$  since  $\mathbf{J}$  is a non-square matrix. The pseudoinverse Jacobian matrix is defined as

$$\mathbf{H} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \in \mathbb{R}^{3 \times 4} \quad (3.26)$$

Thus, the Cartesian velocities of the parallel structure of SPIDER4 are computed through the following expression:

$${}^o\dot{\mathbf{N}}_N = \mathbf{H}\dot{\mathbf{q}} \quad (3.27)$$

### 3.2.5 Inverse Dynamic Model of the delta-like positioning mechanism of SPIDER4

The IDM of the SPIDER4 delta-like positioning device is obtained using the modeling simplifications described in Chapter 2. The methodology presented in [Nabat, 2007] summarizes the dynamic model for a PKM with delta-like architecture in the following form:

$$\mathbf{\Gamma}(t) = \mathbf{\Gamma}_{act} + \mathbf{\Gamma}_{na} \quad (3.28)$$

Where  $\mathbf{\Gamma}_{act} \in \mathbb{R}^n$  are the forces/torque contributions of the actuators and the rear-arms plus a half mass of the forearms, whereas that  $\mathbf{\Gamma}_{na} \in \mathbb{R}^n$  represents the forces/torque contribution of the nacelle plus the other half masses of the forearms. The IDM for the parallel structure of SPIDER4 has notable differences compared to the Delta robot concerning to the gravity effect over the kinematic chains since the kinematic chains of SPIDER4 are oriented horizontally as long as those of Delta are oriented vertically. Figure 3.6 illustrates the modeling simplifications applied to SPIDER. Let us start with the computation of  $\mathbf{\Gamma}_{na}$ . Considering that the nacelle performing only translation movements, it would be possible to avoid the calculation of moments over it. Therefore, we can describe the nacelle dynamics by NE formulation as follows:

$$\mathbf{M}_t({}^o\ddot{\mathbf{N}}_N - \mathbf{G}) = 0 \quad (3.29)$$

In which,  ${}^o\ddot{\mathbf{N}}_N \in \mathbb{R}^{3 \times 1}$  denotes the Cartesian acceleration on the nacelle, and  $\mathbf{G} \in \mathbb{R}^{3 \times 1}$  is the gravity acceleration vector denoted by  $\mathbf{G} = [0 \ g \ 0]^T$  with  $g = 9.81 \text{ m/s}^2$ .  $\mathbf{M}_t$  is the total mass matrix of the nacelle and the elements that are over it. This matrix is constituted as  $\mathbf{M}_t = \text{diag}\{[m_t \ m_t \ m_t]\} \in \mathbb{R}^{3 \times 3}$ , whose elements are given by:

$$m_t = m_{na} + m_{namotors} + 4 \frac{m_{fa}}{2} \quad (3.30)$$

Where  $m_{na}$  is the nacelle mass,  $m_{fa}$  is the forearm mass composed by the two parallelogram bars mass,  $m_{namotors}$  is the mass of the three motors located at the nacelle and their coupling parts. The motors located at the nacelle include: the motor generating the movement around  $\phi$ , the motor producing the movement over  $\psi$ , and the spindle motor. It is worth to note that gravity acceleration is along the  $y$ -axis instead of  $z$ -axis due to the orientation of the parallel structure of the robot. Therefore, the gravitational force acting on the nacelle can be expressed as follows:

$$\mathbf{G}_{na} = -\mathbf{M}_t \mathbf{G} \quad (3.31)$$

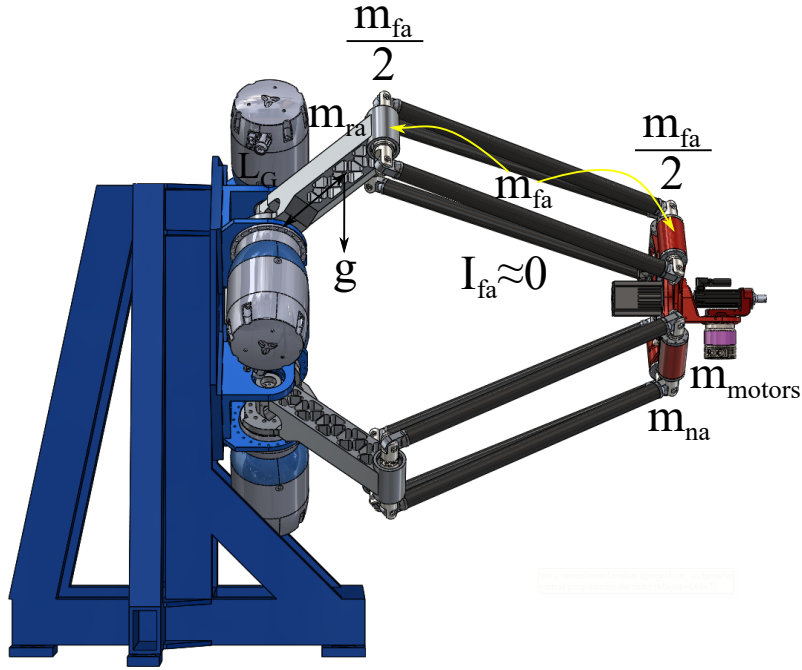


Figure 3.6: Illustration of dynamic model simplifications applied to SPIDER4 RA-PKM

The inertial forces acting on the nacelle owing to the Cartesian acceleration  ${}^o\ddot{\mathbf{N}}_N \in \mathbb{R}^{3 \times 1}$  are expressed by:

$$\mathbf{F}_{na} = \mathbf{M}_t^o \ddot{\mathbf{N}}_N \quad (3.32)$$

The contributions of  $\mathbf{G}_{na}$  and  $\mathbf{F}_{na}$  in joint space can be calculated using the pseudoinverse Jacobian matrix  $\mathbf{H}(\mathbf{q}, {}^o\mathbf{N}_N) \in \mathbb{R}^{3 \times 4}$ . These contributions are expressed through the following relationship:

$$\mathbf{\Gamma}_{G_{na}} = -\mathbf{H}^T \mathbf{M}_t \mathbf{G} \quad (3.33)$$

$$\mathbf{\Gamma}_{F_{na}} = \mathbf{H}^T \mathbf{M}_t^o \ddot{\mathbf{N}}_N \quad (3.34)$$

Therefore,  $\mathbf{\Gamma}_{na}$  is given as follows:

$$\mathbf{\Gamma}_{na} = \mathbf{\Gamma}_{G_{na}} + \mathbf{\Gamma}_{F_{na}} \quad (3.35)$$

Having the dynamic equations for the nacelle, the next step consists to identify the dynamics of the acting mechanism which is formed by the set of actuators, rear-arms, and forearms. Similarly, as the nacelle, it should compute the torques produced by joint acceleration and gravity acceleration. The torques produced by joint acceleration on the kinematic chains  $\mathbf{\Gamma}_{F_{ra}} \in \mathbb{R}^{4 \times 1}$  are expressed in this way.

$$\mathbf{\Gamma}_{F_{ra}} = \mathbf{I}_t \ddot{\mathbf{q}} \quad (3.36)$$

where  $\mathbf{I}_t \in \mathbb{R}^{4 \times 4}$  is a diagonal matrix whose elements are formed by:

$$I_t = I_{act} + I_{ra} + \frac{L_i^2 m_{fa}}{2} \quad (3.37)$$

where  $I_{act}$ ,  $I_{ra}$  represents the actuators inertia and the rear-arms inertia, respectively. The term  $\frac{L_i^2 m_{fa}}{2}$  corresponds to the inertial contribution of the forearms using the second modeling simplification where,  $L$  is the rear-arm's length. In order to obtain the torques produced by the gravity acceleration acting on the kinematic chains, we have to take into account the orientation of the manipulator since the gravity force affects differently each one of the kinematic chains. The gravity effect over the limbs of SPIDER4 is directly related to its inclination around  $z$ -axis with a fixed angular orientation  $\alpha_i$  as is appreciated in Figure 3.4. We may define the vector of torques produced by the gravitational forces acting on the kinematic chains set in this form:

$$\mathbf{\Gamma}_{G_{ra}} = -\mathbf{M}_{fra} g \cos(\mathbf{q}) \quad (3.38)$$

Where  $\cos(\mathbf{q}) = [\cos(q_1) \quad \cos(q_2) \quad \cos(q_3) \quad \cos(q_4)]^T$  the matrix  $\mathbf{M}_{fra} \in \mathbb{R}^{4 \times 4}$  is composed as follows:

$$\mathbf{M}_{fra} = \begin{bmatrix} m_{fra} \sin(\alpha_1) & 0 & 0 & 0 \\ 0 & m_{fra} \sin(\alpha_2) & 0 & 0 \\ 0 & 0 & m_{fra} \sin(\alpha_3) & 0 \\ 0 & 0 & 0 & m_{fra} \sin(\alpha_4) \end{bmatrix} \quad (3.39)$$

In which,  $m_{fra} = m_{ra} L_c + \frac{m_{fa} L}{2}$ ,  $L_c$  is the distance to the center of mass of each rear-arm. In this form  $\mathbf{\Gamma}_{act}$  is established as:

$$\mathbf{\Gamma}_{act} = \mathbf{\Gamma}_{Fra} + \mathbf{\Gamma}_{G_{ra}} \quad (3.40)$$

Substituting (3.38) and (3.35) into (3.28), the following expression is obtained:

$$\mathbf{\Gamma} + \mathbf{\Gamma}_{G_{na}} + \mathbf{\Gamma}_{G_{ra}} = \mathbf{\Gamma}_{F_{na}} + \mathbf{\Gamma}_{F_{ra}} \quad (3.41)$$

Equation (3.41) is a representation of the IDM combining Cartesian and joint space variables. However, in order to express the dynamic model of the delta-like positioning device of SPIDER4 only in the joint space, we have to use the following relationship, based on the pseudoinverse Jacobian matrix and its derivative:

$${}^o\ddot{\mathbf{N}}_N = \mathbf{H}\ddot{\mathbf{q}} + \dot{\mathbf{H}}\dot{\mathbf{q}} \quad (3.42)$$

By rearranging the terms, the simplified IDM of the delta-like positioning mechanism of SPIDER4 is expressed terms of the redundant joint coordinates ( $\mathbf{q}$ ) as follows:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{\Gamma}(t) \quad (3.43)$$

with:

- $\mathbf{M}(\mathbf{q}) = \mathbf{I}_t + \mathbf{H}^T \mathbf{M}_t \mathbf{H}$
- $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{H}^T \mathbf{M}_t \dot{\mathbf{H}}$
- $\mathbf{G}(\mathbf{q}) = -\mathbf{\Gamma}_{G_{na}} - \mathbf{\Gamma}_{G_{ra}}$

where  $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{4 \times 4}$  denotes the inertia matrix of the delta-like positioning device of SPIDER4,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{4 \times 4}$  is the Coriolis/Centripetal forces matrix satisfying the skew-symmetry property;  $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{4 \times 1}$  is the vector of forces and torques due to gravity being continuous and bounded.

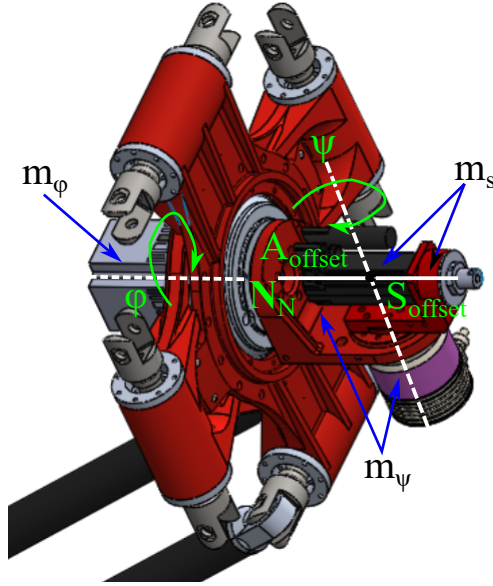


Figure 3.7: Details of the SPIDER4 wrist

### 3.2.6 Inverse Dynamic Model of the Wrist

In the previous subsection, a simplified IDM for the delta-like positioning device of SPIDER4 was derived. Notwithstanding, to derive the IDM of the whole manipulator, it is essential to obtain the IDM of the wrist. The wrist is a serial mechanism formed by two links actuated by two rotational joints. The IDM of this part of the robot is computed using Euler-Lagrange formulation. Thus, we need first to obtain the forward kinematic model of the wrist, which is obtained considering the equation (3.3). Figure 3.7 illustrates the dynamic and kinematic parameters involved in this analysis. The forward kinematic model of the wrist is expressed in the nacelle reference frame  $N - x_n, y_n, z_n$  as follows:

$${}^N \mathbf{W}_W = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} -S_{offset} \cos(\phi) \sin(\psi) \\ -S_{offset} \sin(\phi) \sin(\psi) \\ -S_{offset} \cos(\psi) + A_{offset} \end{bmatrix} \quad (3.44)$$

The Euler-Lagrange formulation needs the calculation of the kinetic and potential energy of the wrist. Developing the derivative of (3.44) the kinetic energy of the wrist can be expressed as:

$$\mathbf{K}_w = \frac{1}{2} m_s^N \dot{\mathbf{W}}_W^{TN} \dot{\mathbf{W}}_W + \frac{1}{2} I_{m\phi} \dot{\phi} \quad (3.45)$$

where  $m_s$  is the mass of the spindle motor and its coupling parts and  $I_{m\phi}$  is the rotational inertia produced by the mass of the second motor on the nacelle and its coupling parts. The Potential energy of the wrist can be expressed in the following form, considering that the gravity acceleration is over  $y$ -axis.

$$\mathbf{U}_w = -m_s g S_{offset} \sin(\phi) \sin(\psi) \quad (3.46)$$

Hence, the Lagrangian function is established in function of  $\boldsymbol{\xi} = [\phi \ \psi]^T$ , as follows [Siciliano et al., 2010]:

$$\mathcal{L}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \mathbf{K}_w(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) - \mathbf{U}_w(\boldsymbol{\xi}) \quad (3.47)$$

The Lagrangian function (3.47) is evaluated in the Euler-Lagrange dynamic equation, in order to acquire the IDM of the wrist.

$$\frac{d}{dt} \left[ \frac{\partial \mathcal{L}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})}{\partial \dot{\boldsymbol{\xi}}} \right] - \left[ \frac{\partial \mathcal{L}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})}{\partial \boldsymbol{\xi}} \right] = \boldsymbol{\Gamma}_w \quad (3.48)$$

Being  $\boldsymbol{\Gamma}_w \in \mathbb{R}^{2 \times 1}$  the torque vector of the wrist actuators. Developing the equation (3.48); the IDM of the wrist is formulated in the function of its joint coordinates as follows:

$$\mathbf{M}_w(\boldsymbol{\xi})\ddot{\boldsymbol{\xi}} + \mathbf{C}_w(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})\dot{\boldsymbol{\xi}} + \mathbf{G}_w(\boldsymbol{\xi}) = \boldsymbol{\Gamma}_w \quad (3.49)$$

being

$$\mathbf{M}_w(\boldsymbol{\xi}) = \begin{bmatrix} m_s S_{offset}^2 \sin \psi + I_{m\phi} & 0 \\ 0 & m_s S_{offset}^2 \end{bmatrix} \quad \mathbf{C}_w(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \begin{bmatrix} 0.5 m_s S_{offset} \cos \psi \dot{\phi} & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{G}_w(\boldsymbol{\xi}) = \begin{bmatrix} -m_s g S_{offset} \sin \psi \cos \phi \\ -m_s g S_{offset} \sin \phi \cos \psi \end{bmatrix}$$

where  $\mathbf{M}_w(\boldsymbol{\xi}) \in \mathbb{R}^{2 \times 2}$  denotes the inertia matrix of the wrist,  $\mathbf{C}_w(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) \in \mathbb{R}^{2 \times 2}$  is the Coriolis/Centripetal forces of the wrist matrix. and  $\mathbf{G}_w(\boldsymbol{\xi}) \in \mathbb{R}^{2 \times 1}$  is the vector of gravity forces for the wrist.

### 3.2.7 Inverse Dynamic Model of SPIDER4 RA-PKM

The whole inverse dynamic model of SPIDER4 can be expressed in function of the generalized coordinates, as a result of the fusion of the IDM of the positioning delta-like device (3.43), and the inverse dynamic model of the wrist (3.49), yielding:

$$\mathbf{M}(\mathbf{Q})\ddot{\mathbf{Q}} + \mathbf{C}(\mathbf{Q}, \dot{\mathbf{Q}})\dot{\mathbf{Q}} + \mathbf{G}(\mathbf{Q}) = \boldsymbol{\Gamma}_T(t) \quad (3.50)$$

being:

$$\mathbf{M}(\mathbf{Q}) = \begin{bmatrix} \mathbf{M}(\mathbf{q})_{[4 \times 4]} & \mathbf{0}_{[2 \times 4]} \\ \mathbf{0}_{[2 \times 4]} & \mathbf{M}_w(\boldsymbol{\xi})_{[2 \times 2]} \end{bmatrix} \quad \mathbf{C}(\mathbf{Q}, \dot{\mathbf{Q}}) = \begin{bmatrix} \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})_{[4 \times 4]} & \mathbf{0}_{[2 \times 4]} \\ \mathbf{0}_{[2 \times 4]} & \mathbf{C}_w(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})_{[2 \times 2]} \end{bmatrix}$$

$$\mathbf{G}(\mathbf{Q}) = \begin{bmatrix} \mathbf{G}(\mathbf{q})_{[4 \times 1]} \\ \mathbf{G}_w(\boldsymbol{\xi})_{[2 \times 1]} \end{bmatrix} \quad \boldsymbol{\Gamma}_T = \begin{bmatrix} \boldsymbol{\Gamma}_{[4 \times 1]} \\ \boldsymbol{\Gamma}_w_{[2 \times 1]} \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} \mathbf{q}_{[4 \times 1]} \\ \boldsymbol{\xi}_{[2 \times 1]} \end{bmatrix}$$

where  $\mathbf{M}(\mathbf{Q}) \in \mathbb{R}^{6 \times 6}$  denotes the inertia matrix of SPIDER4,  $\mathbf{C}(\mathbf{Q}, \dot{\mathbf{Q}}) \in \mathbb{R}^{6 \times 6}$  is the Coriolis/Centripetal forces matrix of SPIDER4,  $\mathbf{G}(\mathbf{Q}) \in \mathbb{R}^{6 \times 1}$  is the vector of gravity forces and,  $\boldsymbol{\Gamma}_T \in \mathbb{R}^{6 \times 1}$  is the torque vector of the whole actuators of SPIDER4.

The dynamic parameters SPIDER4 as the masses of the rear-arms, forearms, the nacelle,

Table 3.2: Summary of the dynamic parameters of SPIDER4

Parameter	Description	Value
$m_{na}$	Nacelle mass	22.76 <i>Kg</i>
$m_{namotors}$	Mass of the three motors located on the Nacelle	19.5 <i>Kg</i>
$m_{ra}$	Rear-arm mass	17.6 <i>Kg</i>
$m_{fa}$	Forearm mass	4.64 <i>Kg</i>
$I_{arm}$	Rear-arm inertia	1.69 <i>Kgm</i> <sup>2</sup>
$I_{act}$	Inertia of one of the principal actuators	0.00223 <i>Kgm</i> <sup>2</sup>
$m_s$	Mass of the spindle motor	3.2 <i>Kg</i>
$m_\psi$	Motor mass that regulates movement in $\psi$	5.1 <i>Kg</i>
$m_\phi$	Motor mass that regulates movement in $\phi$	11.2 <i>Kg</i>

coupling parts, and rear-arms inertia were calculated through material assignation functionality of SolidWorks. The other dynamic parameters as the inertia of the motors and their masses were obtained from the datasheets of the manufacturers of the actuators. The values of dynamical parameters are given in Table (3.2).

**NOTE:** The IDM introduced in (3.43) is taken for subsequent control development in Chapter 4 instead of (3.50) owing to the delta-like, and the wrist mechanisms are controlled independently. The robot's software allows us to modify only the control scheme for the delta-like mechanism; however, for the wrist, that is not possible. In the next section, the control scheme for the delta-like positioning will be presented.

## 3.3 Non-Redundant PKM (Delta)

### 3.3.1 Description and Modeling of Delta robot

Delta PKM is a 3-DOF PKM designed to perform high-speed P&P operations. It has been developed by Reymond Clavel [Clavel, 1991]. The Delta PKM consists of three kinematic chains formed by one rear-arm and one forearm linked together by means passive spherical joints. The forearms of the Delta robot are constituted by two parallel bars whose purpose is to restrict the traveling-plate orientation. The kinematic chains are responsible for transferring the movement generated by the three located on the fixed base to the traveling-plate. Figure 3.8 presents a Delta PKM prototype, which is in the process of development at the Polytechnic University of Tulancingo. The Figure also shows the kinematic configuration of this manipulator where the gray boxes represent the active rotational joints, and the white boxes represent the passive spherical joints.

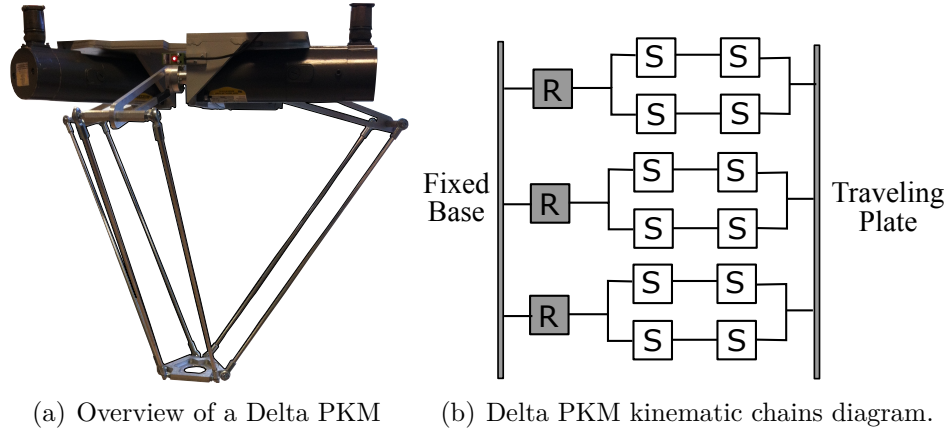


Figure 3.8: The Delta PKM and its kinematic configuration

### 3.3.2 Inverse Kinematic Model of Delta PKM

The IKM of a Delta PKM consists to find the corresponding joint positions of the three actuated joints  $\mathbf{q} = [q_1 \ q_2 \ q_3]^T \in \mathbb{R}^{3 \times 1}$  given the position of the traveling-plate expressed in the fixed reference frame  $O - x_o, y_o, z_o$  as  $\mathbf{X} = [x \ y \ z]^T \in \mathbb{R}^{3 \times 1}$ . The IKM of the Delta PKM can be found by means the following equation [Pierrot et al., 1991].

$$\|{}^o\mathbf{B}_i {}^o\mathbf{C}_i\|^2 = l_i^2 \quad (3.51)$$

This equation is called the closed-loop equation and it involves all geometric variables of the Delta PKM. The IKM of a Delta PKM consists to find the corresponding joint positions of the three actuated joints  $\mathbf{q} = [q_1 \ q_2 \ q_3]^T \in \mathbb{R}^{3 \times 1}$  given the position of the traveling-plate expressed in the fixed reference frame  $O - x_o, y_o, z_o$  as  $\mathbf{X} = [x \ y \ z]^T \in \mathbb{R}^{3 \times 1}$ . Making use of Figure 3.9 the following equations are established in order to obtain the IKM of Delta PKM.

$${}^o\mathbf{A}_i = R_b \begin{bmatrix} \cos(\alpha_i) & \sin(\alpha_i) & 0 \end{bmatrix}^T \quad (3.52)$$

where  $\mathbf{A}_i, \forall i = 1..3$  represents the location of the three actuated joints expressed in the fixed reference frame.  $R_b$  is the fixed-base radius, the actuated joints are placed with the following angles  $\boldsymbol{\alpha} = \left[ \frac{3\pi}{2} \ \frac{\pi}{6} \ \frac{5\pi}{6} \right]^T$ .

The points  $\mathbf{B}_i$  and  $\mathbf{C}_i$  whose coordinates are expressed in the fixed reference frame  $O - x_o, y_o, z_o$  are defined as follows:

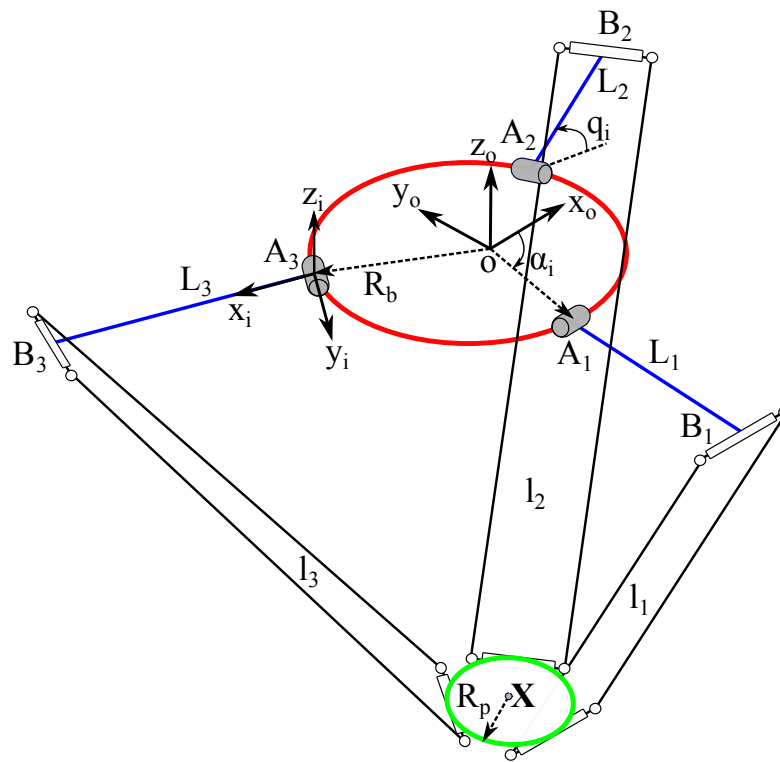
$${}^o\mathbf{B}_i = {}^o\mathbf{A}_i + L \begin{bmatrix} \cos(\alpha_i) \cos(q_i) & \sin(\alpha_i) \cos(q_i) & -\sin(q_i) \end{bmatrix}^T \quad (3.53)$$

$${}^o\mathbf{C}_i = \begin{bmatrix} R_p \cos(\alpha_i) + x & R_p \sin(\alpha_i) + y & z \end{bmatrix}^T \quad (3.54)$$

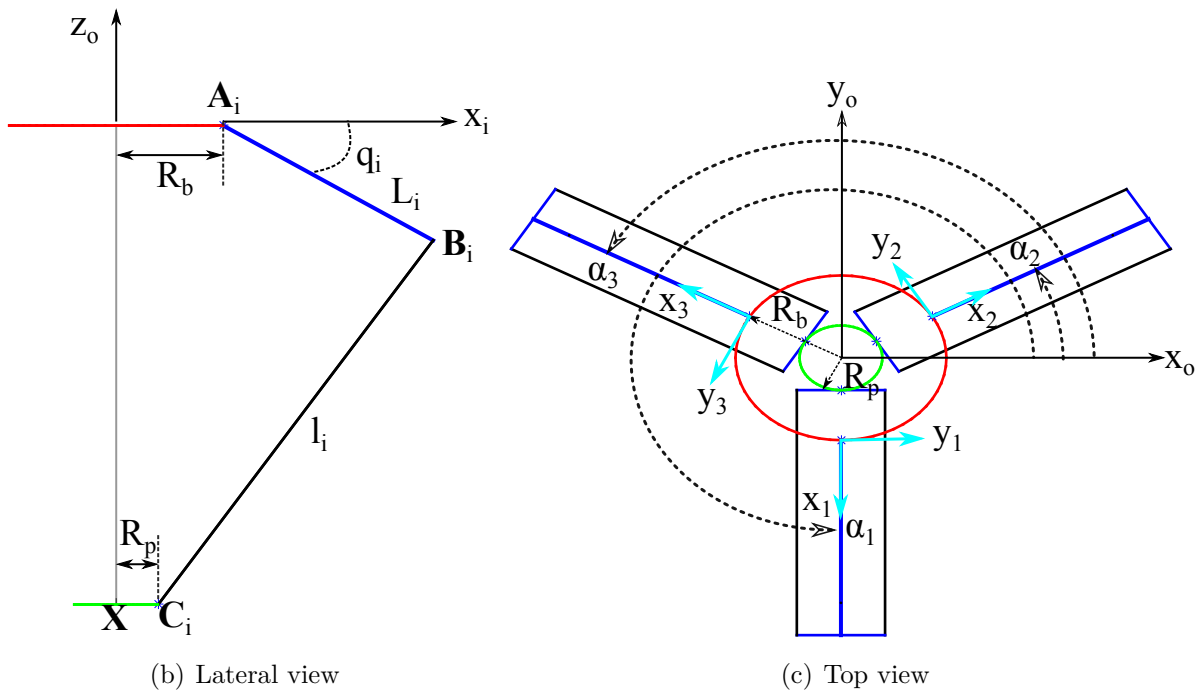
Being  $L$ , the rear-arm length, and  $R_p$  is the traveling-plate radius. For further analysis, an auxiliary frame located at  ${}^o\mathbf{A}_i - x_i, y_i, z_i$  is defined, where the auxiliary vectors  ${}^i\mathbf{x}_i$  and  ${}^i\mathbf{y}_i$  are defined as:

$${}^i\mathbf{x}_i = \begin{bmatrix} \cos(\alpha_i) & \sin(\alpha_i) & 0 \end{bmatrix}^T \quad (3.55)$$





(a) Isometric view



(b) Lateral view

(c) Top view

Figure 3.9: Kinematics illustration of the Delta PKM

Table 3.3: Summary of the kinematic parameters of Delta PKM

Parameter	Description	Value
$L$	Rear-arm length	0.3 $m$
$l$	Forearm length	0.624 $m$
$R_b$	Fixed base radius	0.1267 $m$
$R_p$	Traveling plate radius	0.0497 $m$

$${}^i\mathbf{y}_i = \begin{bmatrix} -\sin(\alpha_i) & \cos(\alpha_i) & 0 \end{bmatrix}^T \quad (3.56)$$

Having defined all the equations that involve the closed-loop equation, expression (3.51) is re-write in the following form to obtain the values of  $q_i$ .

$$D_i \sin(q_i) + E_i \cos(q_i) + F_i = 0 \quad \forall i = 1, 2, 3 \quad (3.57)$$

where  $D_i = 2L_i({}^o\mathbf{A}_i^o \mathbf{C}_i \cdot \mathbf{z}_o)$ ,  $E_i = 2L_i({}^o\mathbf{A}_i^o \mathbf{C}_i \cdot {}^i\mathbf{x}_i)$ , and  $F_i = l_i^2 - L_i^2 - \|{}^o\mathbf{A}_i^o \mathbf{C}_i\|^2$ . Solving (3.57) the values of  $q_i$  can be computed by means of the following expression:

$$q_i = \arctan \left( \frac{-D_i \pm \sqrt{\Delta_i}}{F_i - E_i} \right) \quad (3.58)$$

Being  $\Delta_i = D_i^2 + E_i^2 - F_i^2$ . The mechanism of the Delta PKM of the Polytechnic University of Tulancingo, including all the coupling parts, was designed and assembled in SolidWorks Software. The kinematic parameters were obtained from the CAD model lengths of the limbs, the radius of the fixed base, and the traveling plate. Thereby, the kinematic parameters of Delta are known with precision. Such parameters are presented in Table 3.3.

### 3.3.3 Forward Kinematic Model of Delta PKM

Similar to SPIDER4 RA PKM, to find the traveling plate position, it is necessary to establish a set of nonlinear algebraic equations describing the intersection of virtual spheres. Figure 3.10 describes graphically the FKM solution applied to Delta PKM. Emphasizing the 3-DOF Delta PKM, its FKM consists of finding the corresponding traveling plate position  $\mathbf{X} = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^{3 \times 1}$  expressed in the fixed reference frame  $O - x_o, y_o, z_o$  given the joint positions of the three actuated joints  $\mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T \in \mathbb{R}^{3 \times 1}$ . Figure 3.10 illustrates the intersection algorithm of three virtual spheres to obtain the FKM of Delta Robot. A crucial part of applying this method is to establish the virtual sphere centers. To do that, the points  $\mathbf{B}_i$  must be a move to the center of the traveling plate  $\mathbf{X}$ , which is unknown. Considering the kinematic relationships of a PKM with delta-like architecture, the virtual spheres' centers are established by following kinematic expression.

$${}^o\mathbf{B}'_i = {}^o\mathbf{B}_i - {}^X\mathbf{X}_i \quad (3.59)$$

where  $\mathbf{X}$  is defined in the traveling plate reference frame  $X - x_x, y_x, z_x$  as:

$${}^X\mathbf{X}_i = \begin{bmatrix} R_p \cos(\alpha_i) & R_p \sin(\alpha_i) & 0 \end{bmatrix}^T \quad (3.60)$$

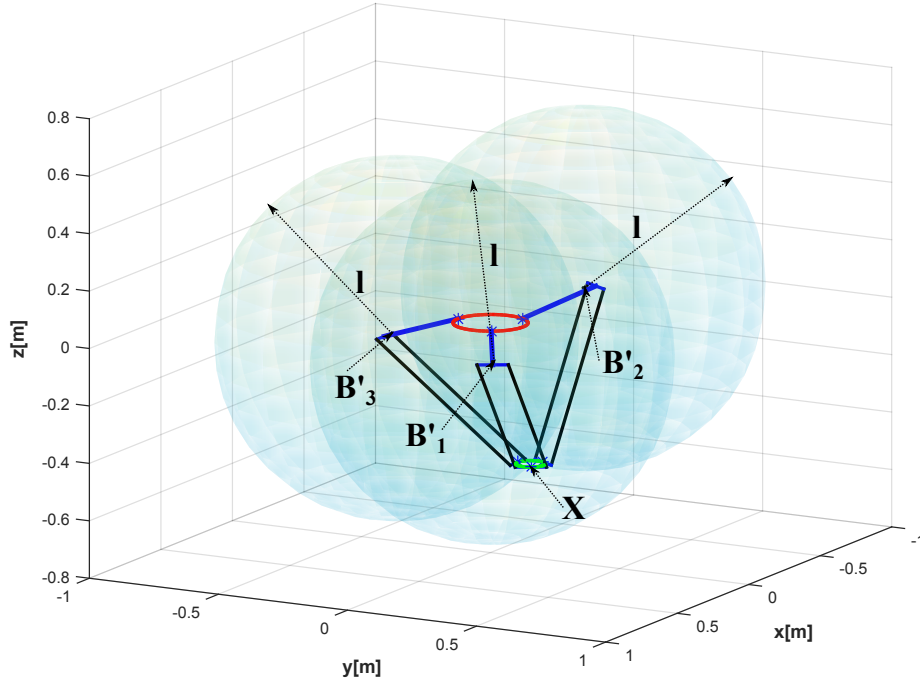


Figure 3.10: Illustration of the three virtual spheres intersection used to find  ${}^o\mathbf{X}$  position

Thus, the sphere equation may be defined as follows:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = l_i^2 \quad (3.61)$$

where  $x_i$ ,  $y_i$ , and  $z_i$  represents the center coordinates of the sphere. Expression (3.59) can be expanded by obtaining the following equation:

$${}^o\mathbf{B}'_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} (L \cos(q_i) + R_b - R_p) \cos(\alpha_i) \\ (L \cos(q_i) + R_b - R_p) \sin(\alpha_i) \\ -L \sin(q_i) \end{bmatrix} \quad (3.62)$$

For the 3-DOF Delta PKM (3.61) is extended to obtain the following equations' system:

$$\begin{aligned} x^2 + y^2 + z^2 - 2x_1x - 2y_1y - 2z_1z + w_1 &= l^2 \\ x^2 + y^2 + z^2 - 2x_2x - 2y_2y - 2z_2z + w_2 &= l^2 \\ x^2 + y^2 + z^2 - 2x_3x - 2y_3y - 2z_3z + w_3 &= l^2 \end{aligned} \quad (3.63)$$

In which,  $w_i = x_i^2 + y_i^2 + z_i^2 \quad \forall i = 1, 2, 3$ . The solution of the previous equations' can be expressed as follows:

$$x = d_2z + e_2 \quad (3.64)$$

$$y = d_1z + e_1 \quad (3.65)$$

$$z = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (3.66)$$

Where the auxiliary constants needed in (3.64)-(3.66) are defined as:

$$\begin{aligned} a_1 &= \frac{y_1 - y_2}{x_2 - x_1} & b_1 &= \frac{z_1 - z_2}{x_2 - x_1} & c_1 &= \frac{w_2 - w_1}{2(x_2 - x_1)} & d_1 &= \frac{b_2 - b_1}{a_1 - a_2} & e_1 &= \frac{c_2 - c_1}{a_1 - a_2} \\ a_2 &= \frac{y_1 - y_3}{x_3 - x_1} & b_2 &= \frac{z_1 - z_3}{x_3 - x_1} & c_2 &= \frac{w_3 - w_1}{2(x_3 - x_1)} & d_2 &= a_1 d_1 + b_1 & e_2 &= a_1 e_1 + c_1 \end{aligned}$$

$$\begin{aligned} a &= d_1^2 + d_2^2 + 1 \\ b &= 2[(e_1 - y_1)d_1 + (e_2 - x_1)d_2 - z_1] \\ c &= e_1^2 + e_2^2 - 2(e_1 y_1 + e_2 x_1) + w_1 - l^2 \end{aligned}$$

### 3.3.4 Velocity Relationship and Jacobian Analysis of Delta PKM

The Jacobian matrices allow to map the velocities of the actuated joints based-on the velocities of the traveling plate or the nacelle. Having in mind that  $\|{}^o\mathbf{B}_i^o\mathbf{C}_i\|^2 = BC_{ix}^2 + BC_{iy}^2 + BC_{iz}^2$  equation (3.51) can be rewrite as

$$\begin{aligned} x^2 - 2Lx \cos \alpha_i \cos q_i + L^2 \cos^2 \alpha_i \cos^2 q_i - 2aL \cos \alpha_i \cos q_i + a^2 + \\ y^2 - 2Ly \sin \alpha_i \cos q_i + L^2 \sin^2 \alpha_i \cos^2 q_i - 2bL \sin \alpha_i \cos q_i + b^2 + \\ z^2 + 2Lz \sin q_i + L^2 \sin^2 q_i = l \end{aligned} \quad (3.67)$$

where  $a = R_p \cos \alpha_i - R_b \cos \alpha_i$  and  $b = R_p \sin \alpha_i - R_b \sin \alpha_i$ . The derivative of (3.67) with respect to time yields the general equation of velocities' relationship expressed in this form:

$$\begin{aligned} ((R_p - R_b) \cos \alpha_i + x - L \cos \alpha_i \cos q_i)\dot{x} + ((R_p - R_b) \sin \alpha_i + y - L \sin \alpha_i \cos q_i)\dot{y} + \\ (z + L \sin q_i)\dot{z} = -L((x \cos \alpha_i + y \sin \alpha_i + R_p - R_b) \sin q_i + z \cos q_i)\dot{q}_i \quad \forall i = 1, \dots, n \end{aligned} \quad (3.68)$$

The terms of the previous expression are regrouped with respect to the velocities in Cartesian space and the velocities in joint space in order to obtain the Jacobian matrices  $\mathbf{J}_x \in \mathbb{R}^{3 \times 3}$ , and  $\mathbf{J}_q \in \mathbb{R}^{3 \times 3}$ . Therefore the general Jacobian matrix of a robot manipulator may be written as:

$$\mathbf{J} = \mathbf{J}_q^{-1} \mathbf{J}_x \in \mathbb{R}^{3 \times 3} \quad (3.69)$$

Considering (3.68) the Jacobian matrices for the Delta PKM are expressed as follows:

$$\mathbf{J}_x = \begin{bmatrix} R \cos \alpha_1 + x - L \cos \alpha_1 \cos q_1 & R \sin \alpha_1 + y - L \sin \alpha_1 \cos q_1 & z + L \sin q_1 \\ R \cos \alpha_2 + x - L \cos \alpha_2 \cos q_2 & R \sin \alpha_2 + y - L \sin \alpha_2 \cos q_2 & z + L \sin q_2 \\ R \cos \alpha_3 + x - L \cos \alpha_3 \cos q_3 & R \sin \alpha_3 + y - L \sin \alpha_3 \cos q_3 & z + L \sin q_3 \end{bmatrix} \quad (3.70)$$

$$\mathbf{J}_q = \begin{bmatrix} J_{q11} & 0 & 0 \\ 0 & J_{q22} & 0 \\ 0 & 0 & J_{q33} \end{bmatrix} \quad (3.71)$$

where  $R = R_p - R_b$  and

$$\begin{aligned} Jq_{11} &= -L((x \cos \alpha_1 + y \sin \alpha_1 + R) \sin q_1 + z \cos q_1) \\ Jq_{22} &= -L((x \cos \alpha_2 + y \sin \alpha_2 + R) \sin q_2 + z \cos q_2) \\ Jq_{33} &= -L((x \cos \alpha_3 + y \sin \alpha_3 + R) \sin q_3 + z \cos q_3) \end{aligned}$$

The joint velocities of the Delta PKM  $\dot{\mathbf{q}} \in \mathbb{R}^{3 \times 1}$  may be computed from the Cartesian velocities  $\dot{\mathbf{X}} \in \mathbb{R}^{3 \times 1}$  through the general Jacobian matrix  $\mathbf{J} \in \mathbb{R}^{3 \times 3}$  in this way

$$\dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{X}} \quad (3.72)$$

To obtain the Cartesian velocities in function of the joint velocities (3.72) is reformulated as:

$$\dot{\mathbf{X}} = \mathbf{J}_m \dot{\mathbf{q}} \quad (3.73)$$

being  $\mathbf{J}_m$  the inverse matrix of  $\mathbf{J}$

### 3.3.5 Inverse Dynamic Model of Delta PKM

The simplified IDM of the 3-DOF Delta PKM is obtained considering the hypothesis simplifications described in Chapter 2, where all kinds of friction are neglected, and the forearms' mass is distributed equivalently between the rear-arms and the traveling plate. Therefore, equation (2.13) can be re-written as:

$$\mathbf{\Gamma}(t) = \mathbf{\Gamma}_{ra}(t) + \mathbf{\Gamma}_{fa}(t) + \mathbf{\Gamma}_{tp}(t) \quad (3.74)$$

where  $\mathbf{\Gamma}_{ra}(t)$  represents the vector of torques produced by the rear-arms,  $\mathbf{\Gamma}_{fa}(t)$  are the torques generated by the forearms, and  $\mathbf{\Gamma}_{tp}(t)$  are the torques produced by the traveling plate or nacelle. The rear-arms torques are computed through the following equation:

$$\mathbf{\Gamma}_{ra}(\mathbf{t}) = \mathbf{I}_{act}\ddot{\mathbf{q}} + \mathbf{I}_{ra}\ddot{\mathbf{q}} - \mathbf{M}_{rag}L_c \cos(\mathbf{q}) \quad (3.75)$$

where  $\mathbf{I}_{act} = \text{diag}([I_{act}]) \in \mathbb{R}^{m \times m}$ , represents the matrix inertia of the actuators and the gear-head,  $\mathbf{I}_{ra} = \text{diag}([I_{ra}]) \in \mathbb{R}^{3 \times 3}$  is the inertia matrix of the rear-arms',  $\cos(\mathbf{q})$  is a vector of  $3 \times 1$ , representing the cosine of each angle  $q_i \forall i = 1, \dots, 3$ ,  $\mathbf{M}_{ra} = \text{diag}([M_{ra}]) \in \mathbb{R}^{3 \times 3}$  is the mass matrix of the rear-arms', and  $L_c$  is the distance from the rotational axis of the rear-arm to its gravity center. The contributions of the inertia and torque (or effort) due to the spatial parallelogram are given by means the following expression:

$$\mathbf{\Gamma}_{fa}(t) = \mathbf{I}_{fa}\ddot{\mathbf{q}} - \mathbf{M}_{fag}L_c \cos(\mathbf{q}) + \mathbf{J}_m^T \mathbf{M}_{nfa}(\ddot{\mathbf{X}} + \mathbf{G}) \quad (3.76)$$

Being  $\mathbf{I}_{ra} = \text{diag}([L^2 \frac{m_{fa}}{2}]) \in \mathbb{R}^{3 \times 3}$ , and  $\mathbf{M}_{fa} = \text{diag}([\frac{m_{fa}}{2}]) \in \mathbb{R}^{3 \times 3}$ , where  $m_{fa}$  is the mass of a forearm considering the two parallel bars.  $\mathbf{M}_{nfa} = \text{diag}([n \frac{m_{fa}}{2}])$  is a matrix of  $3 \times 3$ , as long as  $\mathbf{J}_m$  is the pseudo inverse Jacobian matrix with a dimension of  $3 \times 3$ , and  $\mathbf{G} = [0 \ 0 \ g]$ . The torque contributions given by the traveling are denoted by the following expression

$$\mathbf{\Gamma}_{tp}(t) = \mathbf{J}_m^T \mathbf{M}_{tp}(\ddot{\mathbf{X}} - \mathbf{G}) \quad (3.77)$$

Table 3.4: Summary of the dynamic parameters of Delta PKM

Parameter	Description	Value
$m_{tp}$	Traveling Plate Mass	0.19 Kg
$m_{ra}$	Rear-arm Mass	0.29 Kg
$m_{fa}$	Forearm Mass	0.28 Kg
$I_{ra}$	Rear-arm Inertia	0.0213 Kg $m^2$
$I_{act}$	Motor Inertia	$3.8 \times 10^{-6}$ Kg $m^2$

Being  $\mathbf{M}_{tp} = \text{diag}([m_{tp}]) \in \mathbb{R}^{3 \times 3}$ , where  $m_{tp}$  is the mass of the traveling plate. Substituting equations (3.75)-(3.77) into (3.74), we obtain the following expression for the Delta PKM dynamic model.

$$\mathbf{\Gamma} = (\mathbf{I}_{act} + \mathbf{I}_{ra} + \mathbf{I}_{fa})\ddot{\mathbf{q}} - (\mathbf{M}_{rag}L_G + \mathbf{M}_{fag}L)\cos(\mathbf{q}) + \mathbf{J}_m^T(\mathbf{M}_{tp} + \mathbf{M}_{nfa})(\ddot{\mathbf{X}} - \mathbf{G}) \quad (3.78)$$

To express entirely the IDM in the joint variables we should derive (3.73), obtaining:

$$\ddot{\mathbf{X}} = \mathbf{J}_m\ddot{\mathbf{q}} + \dot{\mathbf{J}}_m\dot{\mathbf{q}} \quad (3.79)$$

Equation (3.79) is substituted into (3.78), the resulted expression is reorganize in the following form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{\Gamma} \quad (3.80)$$

The inertia matrix  $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{3 \times 3}$  is expressed as:

$$\mathbf{M}(\mathbf{q}) = \mathbf{I}_t + \mathbf{J}_m^T\mathbf{M}_t\mathbf{J}_m \quad (3.81)$$

where  $\mathbf{I}_t = \mathbf{I}_{act} + \mathbf{I}_{ra} + \mathbf{I}_{fa} \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{M}_t = \mathbf{M}_{tp} + \mathbf{M}_{nfa} \in \mathbb{R}^{3 \times 3}$ . The matrix of Coriolis and centrifugal forces  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{3 \times 3}$  is denoted as:

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_m^T\mathbf{M}_t\dot{\mathbf{J}}_m \quad (3.82)$$

Finally, the gravity forces vector  $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{3 \times 1}$  is conformed by:

$$\mathbf{G}(\mathbf{q}) = -\mathbf{J}_m^T\mathbf{M}_t\mathbf{G} - (\mathbf{M}_{rag}L_c + \mathbf{M}_{fag}L)\cos(\mathbf{q}) \quad (3.83)$$

The dynamic parameters of Delta PKM as the masses of the rear-arms, forearms, the traveling plate, coupling parts, and rear-arms inertia were calculated through the material assignation functionality of SolidWorks. The other dynamic parameters as the inertia of the motors and their masses were obtained from the datasheets of the manufacturers of the actuators. The values of such dynamical parameters are provided in Table 3.4.

## 3.4 Conclusion

In this chapter, two PKMs were introduced, the new solution for machining called SPIDER4, which was built between LIRMM and Tecnalía, and a Delta PKM designed and built at the

---

Polytechnic University of Tulancingo intended to be used in P&P tasks. The kinematic and dynamic models of both platforms were developed. For the SPIDER4 positioning device and the Delta PKM, the closed-loop method was used to obtain the IKM. Besides, it was employed transformations matrices to compute the IKM of the wrist of SPIDER4. The IKM of the delta-like positioning devices and the wrist were coupled to get the IKM of the whole SPIDER4. The FKM for both PKMs has been calculated through a virtual spheres intersection algorithm, which provides the nacelle and traveling-plate position when the robot motors' joint position is known. The relationship of velocities was achieved thanks to the Jacobian matrices. These matrices were derived from the closed-loop equation established in the kinematic analysis to obtaining the IKMs. Two significant simplifications were considered in the IDM development for the delta-like positioning device of SPIDER4 and the Delta PKM. The first one is the forearms' inertia is despite, and their mass is divided into two parts. One part is added to the end of the rear-arm, and the other part is added to the traveling plate or nacelle. The second simplification neglects the frictional forces in the joints of the PKMs since it is complicated to obtain an accurate friction model in nonlinear systems as these kinds of PKMs. Moreover, in the case of SPIDER4, it was calculated the IDM of the wrist using Euler-Lagrange formulation. Both IDMs of the wrist and the positioning part were joined to get the whole IDM of SPIDER4. Nevertheless, we will consider the IDM of the delta-like device for control purposes since we can not modify the control law of the wrist mechanism.








---

## Proposed Control Solutions

*In this chapter, the proposed control solutions for SPIDER<sub>4</sub> RA-PKM and the Delta PKM developed at the Polytechnic University of Tulancingo will be described in detail. For SPIDER<sub>4</sub>, the control solutions are a RISE controller with nominal feedforward and a RISE feedforward controller with adaptive feedback gains. Finally, for Delta, we propose a RISE control with an adaptive feedforward compensation term based-on B-Spline neural networks.*

### Contents

---

<b>4.1</b>	<b>Introduction</b>	<b>73</b>
<b>4.2</b>	<b>Background on RISE-Based Controllers</b>	<b>74</b>
<b>4.3</b>	<b>Background on Artificial Neural Networks</b>	<b>77</b>
<b>4.4</b>	<b>Control Solution 1: A RISE Controller with Nominal Feedforward</b>	<b>81</b>
<b>4.5</b>	<b>Control Solution 2: A RISE Feedforward controller with Adaptive Feedback Gains</b>	<b>85</b>
<b>4.6</b>	<b>The actuation redundancy on SPIDER<sub>4</sub></b>	<b>89</b>
<b>4.7</b>	<b>Control Solution 3: A RISE controller with Adaptive Feedforward Compensation Based-on B-Spline Neural Networks</b>	<b>90</b>
<b>4.8</b>	<b>Conclusion</b>	<b>95</b>

---

### 4.1 Introduction

Control of PKMs is an important subject that is still in development because, in each decade, new architectures of these types of manipulators emerge to be applied in one or more specific tasks. As detailed in Chapter 1, parallel robots, due to their kinematic construction, have desirable features such as improved stiffness, equitable distribution of the payload in its set of linkages, high repeatability, potential accuracy, and, in the case of RA-PKMs, the

singularity avoidance, and improved dynamics. Notwithstanding, the control of these devices represent significant challenges due to the presence of different types of uncertainties, the highly nonlinear dynamics, and the issue of internal forces generation on RA-PKMs. To overcome these issues, advanced nonlinear control techniques must be implemented to obtain the highest potential that PKMs can offer. Aware of the importance of implementing advanced control schemes for Parallel robots, in this Chapter, we will describe in detail three proposed control solutions for PKMs. The first two control solutions will be applied on SPIDER4, as long as the third one will be implemented on Delta PKM. The proposed controllers aim to keep the joint tracking error as minimum possible under the environment in which the robot is intended to operate. The first control solution aims to validate the proposed IDM of SPIDER4 experimentally, the second one is conceived to deal with the problem machining in soft materials, and the third one is proposed to keep precision on P&P tasks. Although these control proposals were thought to satisfy machining and P&P tasks, it is also possible to implement these controllers in any other type of robot manipulator. Before describing the proposed control schemes in detail, let us present a background of control techniques used as the basis of our proposals.

## 4.2 Background on RISE-Based Controllers

RISE control is a kind of continuous feedback controller that may ensure semi-global asymptotic stability even in the presence of a large class of uncertainties and nonlinearities [Xian et al., 2004]. All real nonlinear systems are well known for the presence of non-modeled phenomena as friction dynamics, or the inaccurate knowledge of its dynamic parameters. Indeed, in robotic manipulators, some of such dynamic parameters are time-varying, such as the payload. Therefore, based on limited assumptions of the system to be controlled as the equation system structure, the controller can partially compensate for such nonlinearities and uncertainties with an appropriate gain selection. For the mentioned advantages, RISE control has already been tested on PKMs. In [Bennehar et al., 2018], the first implementation of the original RISE control was reported on a PKM, a 3-DOF Delta robot. Moreover, in the mentioned work, an adaptive feedforward-based term was added to original RISE control to enhance the overall performance of the robot. A similar RISE control with adaptive feedforward was implemented on a 3-DOF RA-PKM called Dual-V [Bennehar et al., 2014b], the main difference between this control scheme and the previous one lies in the adaptation law. Another interesting extension of RISE control was presented in [Saied et al., 2019a], where some of the constant feedback gains were replaced by nonlinear time-varying ones endowing the controller with better robustness against external perturbations.

### 4.2.1 Original RISE Control

Lets us consider the following  $n$ th order MIMO nonlinear system

$$\mathbf{M}(\mathbf{X})\mathbf{X}^n + \mathbf{F}(\mathbf{X}) = \mathbf{U}(t) \quad (4.1)$$

In which,  $\mathbf{X}(t) = [\mathbf{x}(t) \ \dot{\mathbf{x}}(t) \ \dots, \ \mathbf{x}^{n-1}(t)]^T \in \mathbb{R}^n$  represents the system states,  $\mathbf{U}(t) \in \mathbb{R}^n$  denotes the control input, and  $\mathbf{M}(\mathbf{X}) \in \mathbb{R}^{n \times n}$ ,  $\mathbf{F}(\mathbf{X}) \in \mathbb{R}^n$  are uncertain nonlinear functions. Based on the system states one can define the output tracking error as

$$\mathbf{e}_1 = \mathbf{x}_d - \mathbf{x} \quad (4.2)$$

Where  $\mathbf{x}_d \in \mathbb{R}^n$  represents the desired trajectory. The control objective is to guarantee  $\mathbf{e}_1(t) \rightarrow 0$  as long as  $t \rightarrow \infty$

**Property 4.2.1** *The nonlinear functions  $\mathbf{M}(\cdot)$ ,  $\mathbf{F}(\cdot)$  are second-order differentiable and bounded.*

**Property 4.2.2** *The uncertain nonlinear function  $\mathbf{M}(\cdot) \in \mathbb{R}^{n \times n}$  is a symmetric positive-definite matrix satisfying the following inequality  $\forall \boldsymbol{\zeta} \in \mathbb{R}^n$  :*

$$\underline{m} \|\boldsymbol{\zeta}\|^2 \leq \boldsymbol{\zeta}^T \mathbf{M}(\cdot) \boldsymbol{\zeta} \leq \overline{m}(\mathbf{x}) \|\boldsymbol{\zeta}\|^2 \quad (4.3)$$

Where  $\underline{m}$  is a positive constant,  $\overline{m}(\mathbf{x})$  is positive non-decreasing function, and  $\|\cdot\|$  represents the standard Euclidean norm.

**Property 4.2.3** *The desired trajectory  $\mathbf{x}_d$  is continuously differentiable with respect to time until the  $(n+2)$  derivative.*

### Control Law

Lets us consider the following auxiliary error signals before to introduce the control law:

$$\begin{aligned} \mathbf{e}_2(t) &= \dot{\mathbf{e}}_1(t) + \mathbf{e}_1(t) \\ \mathbf{e}_3(t) &= \dot{\mathbf{e}}_2(t) + \mathbf{e}_2(t) + \mathbf{e}_1(t) \\ \mathbf{e}_4(t) &= \dot{\mathbf{e}}_3(t) + \mathbf{e}_3(t) + \mathbf{e}_2(t) \\ &\vdots \\ \mathbf{e}_n(t) &= \dot{\mathbf{e}}_{n-1}(t) + \mathbf{e}_{n-1}(t) + \mathbf{e}_{n-2}(t) \end{aligned} \quad (4.4)$$

According to the stability analysis presented in [Xian et al., 2004], the following control law is established in order to reach the control objective:

$$\mathbf{U}(t) = (\mathbf{K}_s + \mathbf{I})\mathbf{e}_n(t) - (\mathbf{K}_s + \mathbf{I})\mathbf{e}_n(0) + \int_0^t [(\mathbf{K}_s + \mathbf{I})\boldsymbol{\Lambda}\mathbf{e}_n(\tau) + \boldsymbol{\beta}\text{sgn}(\mathbf{e}_n(\tau))]d\tau \quad (4.5)$$

Where  $\mathbf{K}_s$ ,  $\boldsymbol{\Lambda}$ ,  $\boldsymbol{\beta} \in \mathbb{R}^{n \times n}$  are positive-definite control gain matrices, and  $\mathbf{I} \in \mathbb{R}^{n \times n}$  represents the identity matrix.

### Application of Standard RISE control to PKMs

For a PKM with  $m$ -DOF and  $n$  actuators, equation (4.1) may expressed as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{\Gamma}(t) \quad (4.6)$$

Where  $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{\Gamma}_d$ , being  $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$  the vector containing the friction effects, and  $\mathbf{\Gamma}_d \in \mathbb{R}^n$  a bounded vector of disturbances. The joint tracking error  $\mathbf{e}_1(t) \in \mathbb{R}^n$  is defined as:

$$\mathbf{e}_1(t) = \mathbf{q}_d(t) - \mathbf{q}(t) \quad (4.7)$$

The following filtered tracking errors are defined to assist in the further stability analysis [Patre et al., 2006]:

$$\mathbf{e}_2(t) = \dot{\mathbf{e}}_1(t) + \mathbf{\Lambda}_1\mathbf{e}_1(t) \quad (4.8)$$

$$\mathbf{r}(t) = \dot{\mathbf{e}}_2(t) + \mathbf{\Lambda}_2\mathbf{e}_2(t) \quad (4.9)$$

Where  $\mathbf{\Lambda}_1$  and  $\mathbf{\Lambda}_2 \in \mathbb{R}^{n \times n}$ , are two positive-definite diagonal gain matrices. The open-loop tracking error system can be obtained multiplying (4.9) by  $\mathbf{M}(\mathbf{q})$  yielding the following expression:

$$\mathbf{M}(\mathbf{q})\mathbf{r} = \mathbf{M}(\mathbf{q})(\dot{\mathbf{e}}_2 + \mathbf{\Lambda}_2\mathbf{e}_2) \quad (4.10)$$

Making use of (4.9) and (4.8), the closed-loop tracking error system is expressed as:

$$\mathbf{M}(\mathbf{q})\mathbf{r} = \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d + \mathbf{\Lambda}_1\dot{\mathbf{e}}_1 + \mathbf{\Lambda}_2\mathbf{e}_2) - \mathbf{\Gamma}_{RISE}(t) \quad (4.11)$$

In which,  $\mathbf{\Gamma}_{RISE}(t) = \mathbf{\Gamma}(t)$ . Equation (4.5) can be re-formulated to be applied in robotic systems as follows:

$$\mathbf{\Gamma}_{RISE}(t) = (\mathbf{K}_s + \mathbf{I})\mathbf{e}_2(t) - (\mathbf{K}_s + \mathbf{I})\mathbf{e}_2(0) + \int_0^t [(\mathbf{K}_s + \mathbf{I})\mathbf{\Lambda}_2\mathbf{e}_2(\tau) + \beta \text{sgn}(\mathbf{e}_2(\tau))]d\tau \quad (4.12)$$

To facilitate the stability analysis of Standard RISE control, one can develop the time derivative of (4.11) obtaining the following equation:

$$\begin{aligned} \mathbf{M}(\mathbf{q})\dot{\mathbf{r}} = & -\dot{\mathbf{M}}(\mathbf{q})(\mathbf{r} - \ddot{\mathbf{q}}_d) + \dot{\mathbf{F}}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{\Lambda}_1(\mathbf{M}(\mathbf{q})\dot{\mathbf{e}}_1 + \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{e}}_1) \\ & \mathbf{\Lambda}_2(\mathbf{M}(\mathbf{q})\dot{\mathbf{e}}_2 + \dot{\mathbf{M}}(\mathbf{q})\mathbf{e}_2) - \dot{\mathbf{\Gamma}}_{RISE} \end{aligned} \quad (4.13)$$

The previous equation is re-written in the following form as is stated in [Xian et al., 2004]

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{r}} = -\frac{1}{2}\dot{\mathbf{M}}(\mathbf{q})\mathbf{r} + \mathbf{N}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{r}, t) - \mathbf{e}_2(t) - \dot{\mathbf{\Gamma}}_{RISE}(t) \quad (4.14)$$

Where  $\mathbf{N}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{r}, t) \in \mathbb{R}^n$  is a nonlinear auxiliary function containing the uncertain terms related to the robot's dynamic model whose elements are defined as follows:

$$\begin{aligned} \mathbf{N}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{r}, t) = & \dot{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_d + \dot{\mathbf{F}}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{e}_2(t) - \frac{1}{2}\dot{\mathbf{M}}(\mathbf{q})\mathbf{r} \\ & \mathbf{\Lambda}_1(\mathbf{M}(\mathbf{q})\dot{\mathbf{e}}_1 + \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{e}}_1) + \mathbf{\Lambda}_2(\mathbf{M}(\mathbf{q})\dot{\mathbf{e}}_2 + \dot{\mathbf{M}}(\mathbf{q})\mathbf{e}_2) \end{aligned} \quad (4.15)$$

The time derivative of Standard RISE control is given as:

$$\dot{\Gamma}_{RISE} = (\mathbf{K}_s + \mathbf{I})\mathbf{r} + \beta \text{sgn}(\mathbf{e}_2) \quad (4.16)$$

Let us define the following auxiliary function  $\mathbf{N}_d(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d, t) \in \mathbb{R}^n$  used to facilitate the subsequent stability analysis of the controller.

$$\mathbf{N}_d(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d, t) = \mathbf{M}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \dot{\mathbf{M}}(\mathbf{q}_d)\dot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \dot{\mathbf{C}}(\mathbf{q}_d, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \dot{\mathbf{G}}(\mathbf{q}_d) + \dot{\mathbf{f}}(\mathbf{q}_d, \dot{\mathbf{q}}_d) \quad (4.17)$$

Adding and subtracting  $\mathbf{N}_d(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d, t)$  to the right-hand side of the closed-loop error system equation (4.14), the following mathematical expression is obtained:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{r}} = -\frac{1}{2}\dot{\mathbf{M}}(\mathbf{q})\mathbf{r} + \mathbf{N}_d + \tilde{\mathbf{N}} - \mathbf{e}_2(t) - \dot{\Gamma}_{RISE}(t) \quad (4.18)$$

Where

$$\tilde{\mathbf{N}}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{r}, t) = \mathbf{N}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{r}, t) - \mathbf{N}_d(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d, t) \quad (4.19)$$

The vector  $\tilde{\mathbf{N}}$  can be upper bounded as follows using the Mean Value Theorem:

$$\|\tilde{\mathbf{N}}\| \leq \rho(\|\mathbf{z}\|)\|\mathbf{z}\| \quad (4.20)$$

In which,  $\|\cdot\|$  represents the Euclidean norm, and  $\mathbf{z}(t) \in \mathbb{R}^{3n}$  is an error vector defined as:

$$\mathbf{z}(t) = [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{r}]^T \quad (4.21)$$

The stability analysis for this control scheme was proved in [Xian et al., 2004], achieving the control objective presented at the beginning of this section. The mentioned work will be considered for the analysis of the proposed control solutions detailed in further sections.

## 4.3 Background on Artificial Neural Networks

Artificial Neural Networks (ANN) are computational networks that intend to imitate the behavior of Biological Neural Networks (BNN) [Graupe, 2013]. The human brain is composed of an extensive network of neurons. A neuron is a specialized biological cell that receives and transmits information to other neurons with the help of some electrical and chemical change [Sharma et al., 2012]. The task that performs an individual neuron is simple; however, when they constitute a big network, they can perform quickly and precisely complex tasks such as image recognition and speech [Zou et al., 2008]. An individual neuron consist of the cell body, also called soma, an axon, and dendrites. The information that the neuron receives is processed in the nucleus, which is inside the soma; such information is collected from neighboring neurons through the dendrites. The neuron sends signals at spikes of electrical activity through a long thin stand known as an axon, and an axon splits these signals through synapse and sends it to the other neurons [Sharma et al., 2012]. Figure 4.1 presents a sketch of a biological neuron. Mostly ANNs have three indispensable parts: the nodes, network topology, and the learning rule. Nodes are the equivalent to neurons on a BNN. They include

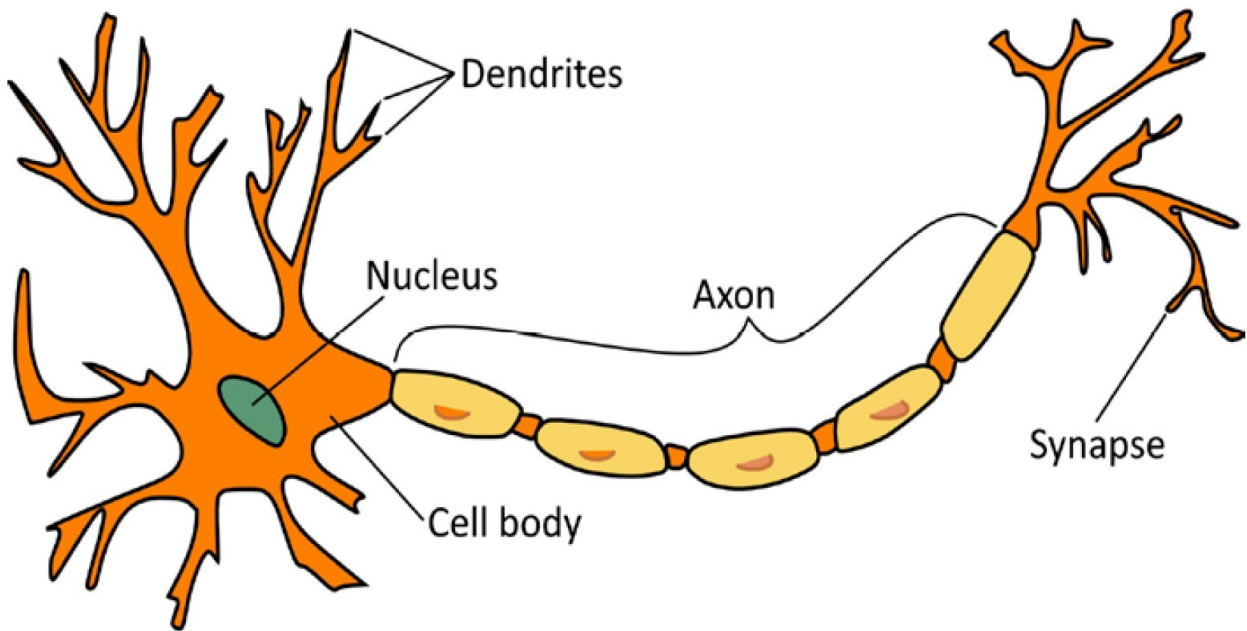


Figure 4.1: Sketch of a biological neuron

the activation function, which processes the entering information on the node inputs. The respective weights associate transmits the information to the input and output of the node. The network topology determines how many nodes will be involved in the network, their distribution, the kind of activation function, and how they will be connected between them. The learning rule determines how the associate weights will be initialized and adjusted. ANNs have been applied extensively in several robotics fields such as pattern recognition, learning, signal processing, modeling identification, and control [Jiang et al., 2017]. Some advantages of the use of ANNs are listed below [Jain et al., 1996]:

- Adaptability
- Learning ability
- Massive parallelism
- Distributed representation and computation
- Inherent contextual formation processing
- Fault tolerance
- Low energy consumption

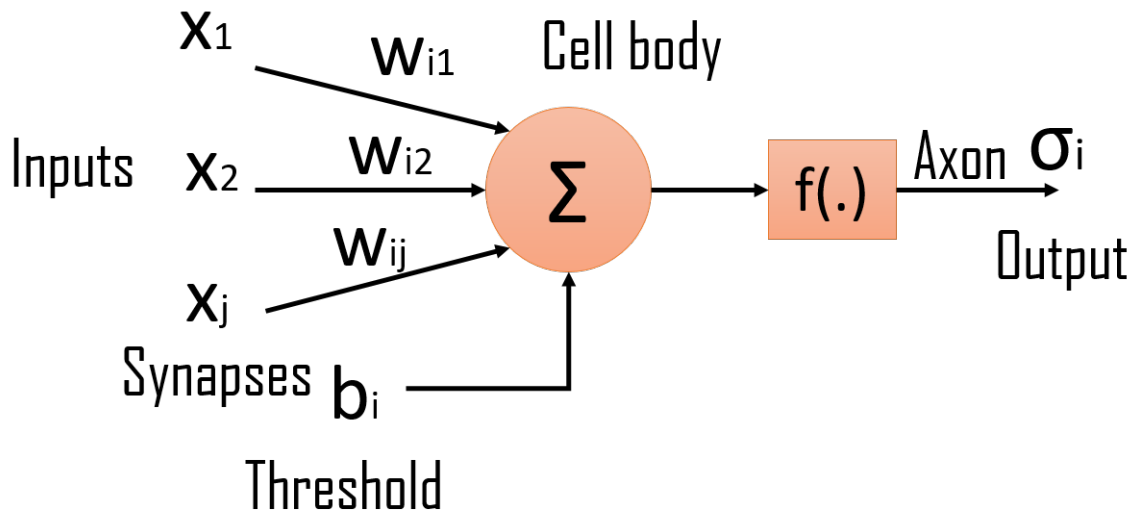


Figure 4.2: Standard artificial neuron model

### 4.3.1 ANNs general model

In many cases the processing nodes of any ANN have a response in the following form:

$$\sigma_i = f \left( \sum_k w_{ij} x_j + b_i \right) \quad (4.22)$$

Where  $\sigma_i$  represents the output of the  $i^{th}$ ,  $b_i$  is a threshold,  $f(.)$  is the activation function which depends of the output of other nodes or external inputs represented by  $x_j$ , and the involved weights  $w_{ij}$ . Figure 4.2 represents the model of an artificial neuron.

### 4.3.2 The activation function

Activation functions are mathematical equations playing a crucial part in each node since they determine the ANN's output. Furthermore, the accuracy, computation efficiency, and the convergence speed of the ANN are strictly related to the activation function. Usually, activation functions are nonlinear functions helping to normalize the output of each node to a range between 1 and 0 or between -1 and 1. Activation functions must be computationally efficient because, in an ANN, hundreds or thousands of nodes work together to achieve a specific task. The most common used activation functions are: Unit step, sign, piecewise linear, sigmoid, hyperbolic tangent, Gaussian, among others (see Figure 4.3).

### 4.3.3 ANNs architectures

The ANNs architectures can be according the flow information as follows:

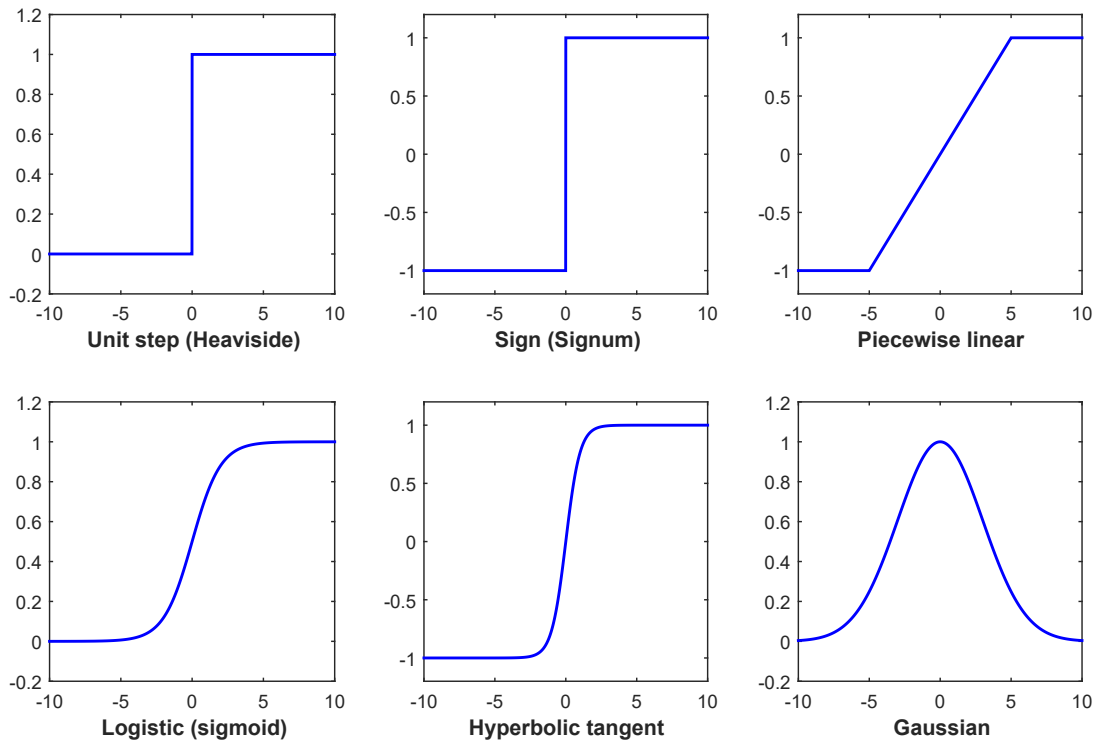


Figure 4.3: Common activation functions for ANNs

- Feedforward ANNs, in which the flow of information only goes forward
- Feedback ANNs, in which loops occur because of feedback connections.

Some examples of ANNs divided into these categories are: Single-layer perceptron, multilayer perceptron, and Radial Basis Function (RBF) for feedforward ANNs, and Hopfield network, competitive networks as feedback ANNs [Jain et al., 1996]. In addition to the classification mentioned above, ANNs architectures can be divided into the following two categories. The first one is the multi-layer ANN (with one or more hidden layers of nodes). This configuration increases the computation complexity since the information travels bidirectionally between the hidden layers of the neural network, besides they entail a considerable computational cost requiring long training time [Deng et al., 2010]. The second one is the single-layer ANN (with only one layer of nodes), for instance, the Associative Memory Neural Network (AMNN). This kind of ANN requires less computational processes due to its single layer of neurons. These kinds of ANNs assume the principle of local generalization, implying that for a specific input, just a portion of the ANN will be involved; thus, the computational effort is reduced. Moreover, their activation functions are linear respect to the adaptable weights so, straightforward instantaneous learning rules can be used to update their adjusted weights [dos Santos Coelho and Pessoa, 2009]. Some examples AMNNs are the Cerebellar Model Articulation Controller (CMAC) and the BSNN.



### 4.3.4 Application of ANNs to robotics control

There have been reported some recent advances in the field of robotics control using ANNs; we can mention some examples of their application in multi-layer and single-layer ANNs. In the branch of multi-layer-based ANNs, a nonlinear adaptive controller was proposed to regulate the trajectory tracking of a Cable-driven robot in [Asl and Janabi-Sharifi, 2017]; the mentioned controller can compensate for parametric and non-parametric uncertainties of the nonlinear robot dynamics; the weights are updated through projection operators. On the other hand, for control solutions based on single-layer ANNs, we can stand out the proposed modification of a CMAC reported in [Razmi and Macnab, 2020], which was proposed to find optimum weight values to outstrip nonlinearities like gravity. This algorithm freezes a set of adaptive weights in a feedforward-like component in the CMAC. When the feedforward component has been established, the algorithm starts to learn another set of weights which contribute to feedback-like terms in the CMAC and these weights get frozen when they no longer reduce a cost-functional. This control solution based in the CMAC ANN was validated with numerical simulations to a two-link flexible-joint robot. Another example is the output feedback controller with a feedforward term based on the RBF ANN reported in [Asl et al., 2018], which was proposed to compensate for uncertainties in the dynamic model of a robotic exoskeleton. This advanced control solution requires only position information for the RBF inputs.

Considering the presented background of RISE control and the fundamental concepts of ANNs, we can proceed to introduce the proposed control solutions for SPIDER4 RA-PKM and Delta PKM.

## 4.4 Control Solution 1: A RISE Controller with Nominal Feedforward

### 4.4.1 Motivation

Feedforward control is an open-loop scheme that compensates the system dynamics without needing information about the system states as the tracking error. For such reason, it requires precise knowledge about the dynamics of the system to be controlled. However, in reality, it is practically impossible to have an accurate knowledge of the system parameters. Besides, the abundance of uncertainties makes inappropriate the implementation of a feedforward control alone because the tracking error will always be incremented owing to this control scheme does not offer any corrective action. Thus, it is necessary to integrate a feedback term guaranteeing that the tracking error goes to zero. In the literature, it has been demonstrated in several works that the addition of a feedforward term to a feedback controller can significantly enhance the overall performance over simple feedback control. So, we propose to combine the Standard RISE control as a robust feedback control to a nominal feedforward term. The main objective of this control solution is to demonstrate the effectiveness of the proposed IDM for SPIDER4 with its computed dynamic parameters. Considering the previous advantage mentioned about

the addition of a feedforward term to a feedback controller, we can say that if the IDM of SPIDER4 and its parameters are sufficiently well approximated to the real dynamics, we will obtain better performance compared to RISE control alone. However, in the opposite case, the resulting performance will be worse.

#### 4.4.2 Proposed Control law

Considering the advantages presented by RISE control and taking into account the benefits generated by the addition of model-based control terms to the control law, the following controller is proposed:

$$\mathbf{\Gamma}(t) = \mathbf{\Gamma}_{RISE} + \mathbf{\Gamma}_{FW} \quad (4.23)$$

Where  $\mathbf{\Gamma}_{RISE} \in \mathbb{R}^n$  is the same controller equation described in (4.12), and the nominal feedforward term  $\mathbf{\Gamma}_{FW} \in \mathbb{R}^n$  is defined as follows

$$\mathbf{\Gamma}_{FW} = \mathbf{M}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \mathbf{G}(\mathbf{q}_d) \quad (4.24)$$

#### Controller Design

Substituting the proposed control law (4.23) into the dynamics of a PKM (4.6) the following closed-loop tracking error system is obtained

$$\mathbf{M}(\mathbf{q})\mathbf{r} = \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d + \mathbf{\Lambda}_1\dot{\mathbf{e}}_1 + \mathbf{\Lambda}_2\mathbf{e}_2) - \mathbf{\Gamma}_{RISE}(t) - \mathbf{\Gamma}_{FW}(t) \quad (4.25)$$

The first time-derivative of (4.25) results in the following expression:

$$\begin{aligned} \mathbf{M}(\mathbf{q})\dot{\mathbf{r}} = & -\dot{\mathbf{M}}(\mathbf{q})(\mathbf{r} - \ddot{\mathbf{q}}_d) + \dot{\mathbf{F}}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{\Lambda}_1(\mathbf{M}(\mathbf{q})\ddot{\mathbf{e}}_1 + \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{e}}_1) \\ & \mathbf{\Lambda}_2(\mathbf{M}(\mathbf{q})\dot{\mathbf{e}}_2 + \dot{\mathbf{M}}(\mathbf{q})\mathbf{e}_2) - \dot{\mathbf{\Gamma}}_{RISE} - \dot{\mathbf{\Gamma}}_{FW} \end{aligned} \quad (4.26)$$

The previous equation is reformulated in the same way as was presented in subsection 4.2.1.

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{r}} = -\frac{1}{2}\dot{\mathbf{M}}(\mathbf{q})\mathbf{r} + \mathbf{N}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{r}, t) - \mathbf{e}_2(t) - \dot{\mathbf{\Gamma}}_{RISE}(t) - \dot{\mathbf{\Gamma}}_{FW}(t) \quad (4.27)$$

Where  $\mathbf{N}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{r}, t) \in \mathbb{R}^n$  was defined in (4.15). To facilitate the stability analysis of this control solution one should reformulated (4.27) in the following form:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{r}} = -\frac{1}{2}\dot{\mathbf{M}}(\mathbf{q})\mathbf{r} + \mathbf{N}_d + \tilde{\mathbf{N}} - \mathbf{e}_2(t) - \dot{\mathbf{\Gamma}}_{RISE}(t) - \mathbf{N}_{FW} \quad (4.28)$$

Where  $\mathbf{N}_d \in \mathbb{R}^n$  and  $\tilde{\mathbf{N}} \in \mathbb{R}^n$  were already defined in (4.17) and (4.19), respectively. The variable  $\mathbf{N}_{FW} = \dot{\mathbf{\Gamma}}_{FW}(t)$  is introduced for the subsequent stability analysis.

### Stability Analysis

Before to present the stability analysis of the proposed control solution 1. Let us establish the following lemma which is a slightly modification of the Lemma 1 presented in [Xian et al., 2004].

**Lemma 1** *An auxiliary function called  $\mathbf{L}(t) \in \mathbb{R}^n$  is defined as:*

$$\mathbf{L}(t) = \mathbf{r}(\mathbf{N}_d(t) - \mathbf{N}_{FW}(t) - \beta \text{sgn}(\mathbf{e}_2)) \quad (4.29)$$

*If the controller gain  $\beta$  is chosen to satisfy the following inequity:*

$$\beta > \|\mathbf{N}_d(t)\|_{\mathcal{L}\infty} - \|\mathbf{N}_{FW}(t)\|_{\mathcal{L}\infty} + \frac{1}{\Lambda_2} \left( \|\dot{\mathbf{N}}_d(t)\|_{\mathcal{L}\infty} - \|\dot{\mathbf{N}}_{FW}(t)\|_{\mathcal{L}\infty} \right) \quad (4.30)$$

*Then*

$$\int_0^t \mathbf{L}(\tau) d\tau \leq \zeta_b \quad (4.31)$$

*In which  $\zeta_b$  is a positive constant defined as:*

$$\zeta_b = \beta \|\mathbf{e}_2(0)\| + \mathbf{e}_2(0)^T (\mathbf{N}_{FW}(0) - \mathbf{N}_d(0)) \quad (4.32)$$

**Proof 1** *See Appendix A*

**Theorem 1** *The tracking error in joint space  $\mathbf{e}_1$  of a PKM whose dynamics is given by (4.6) under the control law (4.23) converges asymptotically to zero as long as time goes to infinity if the gain  $K_s$  is chosen large enough, and the following design parameters are selected such that:*

$$\Lambda_1 > \frac{1}{2}, \quad \Lambda_2 > 1, \quad \beta > \|\mathbf{N}_d(t)\|_{\mathcal{L}\infty} - \|\mathbf{N}_{FW}(t)\|_{\mathcal{L}\infty} + \frac{1}{\Lambda_2} \left( \|\dot{\mathbf{N}}_d(t)\|_{\mathcal{L}\infty} - \|\dot{\mathbf{N}}_{FW}(t)\|_{\mathcal{L}\infty} \right)$$

**Proof 2** *Let  $\mathcal{D} \subset \mathbb{R}^{3n+1}$  a domain containing*

$$\mathbf{y}(t) = [\mathbf{z}^T(t) \quad \sqrt{\mathbf{P}(t)}]^T = \mathbf{0} \quad (4.33)$$

*Where  $\mathbf{P}(t) \in \mathbb{R}^n$  is a supplementary function introduced as:*

$$\mathbf{P}(t) = \beta \|\mathbf{e}_2(0)\| - \mathbf{e}_2(0)^T (\mathbf{N}_{FW}(0) - \mathbf{N}_d(0)) - \int_0^t \mathbf{L}(\tau) d\tau \quad (4.34)$$

*The time-derivative of (4.34) can be written as follows:*

$$\dot{\mathbf{P}}(t) = -\mathbf{r}^T(\mathbf{N}_d(t) - \mathbf{N}_{FW}(t) - \beta \text{sgn}(\mathbf{e}_2)) = -\mathbf{L}(t) \quad (4.35)$$

*One can note that  $\mathbf{P}(t) \geq 0, \forall t \geq 0$  taking into account (4.30) and (4.31)*

Now, Let us define the following Lyapunov candidate function  $\mathbf{V}(\mathbf{y}, t) : \mathcal{D} \times [0, \infty) \rightarrow \mathbb{R}$  being a continuously differentiable positive-definite function as:

$$\mathbf{V}(\mathbf{y}, t) = \frac{1}{2} \mathbf{r}^T \mathbf{M}(\mathbf{q}) \mathbf{r} + \mathbf{e}_1^T \mathbf{e}_1 + \frac{1}{2} \mathbf{e}_2^T \mathbf{e}_2 + \mathbf{P} \quad (4.36)$$

Equation (4.5) satisfies the following inequality

$$\lambda_1 \|\mathbf{y}\|^2 \leq \mathbf{V}(\mathbf{y}, t) \leq \lambda_2 (\|\mathbf{y}\|) \|\mathbf{y}\|^2 \quad (4.37)$$

In which

$$\lambda_1 = \frac{1}{2} \min\{1, \underline{m}\}, \quad \lambda_2(\|\mathbf{y}\|) = \max\{\frac{1}{2} \overline{m}(\|\mathbf{y}\|), 1\}$$

Computing the time-derivative of (4.36) leads to:

$$\dot{\mathbf{V}}(\mathbf{y}, t) = \frac{1}{2} \mathbf{r}^T \dot{\mathbf{M}}(\mathbf{q}) \mathbf{r} + \mathbf{r}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{r}} + 2\mathbf{e}_1^T \dot{\mathbf{e}}_1 + \mathbf{e}_2^T \dot{\mathbf{e}}_2 + \dot{\mathbf{P}} \quad (4.38)$$

Substituting (4.35), (4.27), (4.16) and considering (4.7)-(4.9), expression (4.38) can be rewritten as:

$$\begin{aligned} \dot{\mathbf{V}}(\mathbf{y}, t) = & \mathbf{r}^T \left( -\frac{1}{2} \dot{\mathbf{M}}(\mathbf{q}) \mathbf{r} + \mathbf{N}_d + \tilde{\mathbf{N}} - \mathbf{e}_2(t) - ((\mathbf{K}_s + \mathbf{I}) \mathbf{r} + \beta \text{sgn}(\mathbf{e}_2)) - \mathbf{N}_{Fw} \right) \\ & + 2\mathbf{e}_1^T (\mathbf{e}_2 - \mathbf{\Lambda}_1 \mathbf{e}_1) + \mathbf{e}_2^T (\mathbf{r} - \mathbf{\Lambda}_2 \mathbf{e}_2) - \mathbf{r}^T (\mathbf{N}_d - \mathbf{N}_{FW} - \beta \text{sgn}(\mathbf{e}_2)) \\ & + \frac{1}{2} \mathbf{r}^T \dot{\mathbf{M}}(\mathbf{q}) \mathbf{r} \end{aligned} \quad (4.39)$$

Since  $\mathbf{K}_s$ ,  $\mathbf{I}$ ,  $\mathbf{\Lambda}_1$ , and  $\mathbf{\Lambda}_2$ , are positive definite diagonal gains matrices., and all the elements of each diagonal have the same value. We can simplified  $\dot{\mathbf{V}}(\mathbf{y}, t)$  as:

$$\dot{\mathbf{V}}(\mathbf{y}, t) = \mathbf{r}^T \tilde{\mathbf{N}} - (K_s + 1) \mathbf{r}^T \mathbf{r} - 2\mathbf{\Lambda}_1 \mathbf{e}_1^T \mathbf{e}_1 - \mathbf{\Lambda}_2 \mathbf{e}_2^T \mathbf{e}_2 + 2\mathbf{e}_2^T \mathbf{e}_1 \quad (4.40)$$

Where the terms  $\mathbf{r}^T \mathbf{r}$ ,  $\mathbf{e}_1^T \mathbf{e}_1$ ,  $\mathbf{e}_2^T \mathbf{e}_2$ , and  $\mathbf{e}_2^T \mathbf{e}_1$  can be upper bounded as follows:

$$\mathbf{r}^T \mathbf{r} \leq \|\mathbf{r}\|^2, \quad \mathbf{e}_1^T \mathbf{e}_1 \leq \|\mathbf{e}_1\|^2, \quad \mathbf{e}_2^T \mathbf{e}_2 \leq \|\mathbf{e}_2\|^2, \quad \mathbf{e}_2^T \mathbf{e}_1 \leq \frac{1}{2} \|\mathbf{e}_1\|^2 + \frac{1}{2} \|\mathbf{e}_2\|^2 \quad (4.41)$$

Substituting the inequalities of (4.41) into (4.40) and rearranging terms,  $\dot{\mathbf{V}}(\mathbf{y}, t)$  can be upper bounded as:

$$\dot{\mathbf{V}}(\mathbf{y}, t) \leq \mathbf{r}^T \tilde{\mathbf{N}} + \|\mathbf{e}_1\|^2 + \|\mathbf{e}_2\|^2 - (K_s + 1) \|\mathbf{r}\|^2 + 2\mathbf{\Lambda}_1 \|\mathbf{e}_1\|^2 - \mathbf{\Lambda}_2 \|\mathbf{e}_2\|^2 \quad (4.42)$$

Taking into account (4.20), the previous equation can be written as:

$$\dot{\mathbf{V}}(\mathbf{y}, t) \leq \|\mathbf{r}\| \rho(\|\mathbf{z}\|) \|\mathbf{z}\| + \|\mathbf{e}_1\|^2 + \|\mathbf{e}_2\|^2 - (K_s + 1) \|\mathbf{r}\|^2 + 2\mathbf{\Lambda}_1 \|\mathbf{e}_1\|^2 - \mathbf{\Lambda}_2 \|\mathbf{e}_2\|^2 \quad (4.43)$$

Rearranging terms, equation (4.43) can be expressed as follows:

$$\dot{\mathbf{V}}(\mathbf{y}, t) \leq \lambda_3 \|\mathbf{z}\|^2 - \left( K_s \|\mathbf{r}\|^2 - \rho(\|\mathbf{z}\|) \|\mathbf{z}\| \|\mathbf{r}\| \right) \quad (4.44)$$

Where  $\lambda_3 = \min\{2\Lambda_1 - 1, \Lambda_2 - 1, 1\}$ , and  $\rho(\|\mathbf{z}\|) \in \mathbb{R}^n$  is a positive globally invertible non-decreasing function [Patre et al., 2006]. From  $\lambda_3$  one can see that  $\Lambda_1$  and  $\Lambda_2$  must be chosen according the following inequalities:

$$\Lambda_1 > \frac{1}{2}, \quad \Lambda_2 > 1 \quad (4.45)$$

By developing the squares of the last term of (4.44), such expression is re-written as:

$$\dot{\mathbf{V}}(\mathbf{y}, t) \leq - \left( \lambda_3 - \frac{\rho^2(\|\mathbf{z}\|)}{4K_s} \right) = c\|\mathbf{z}\|^2 \quad (4.46)$$

In which,  $c\|\mathbf{z}\|^2$  is a continuous positive semi-definite function being it defined on the following region

$$\mathcal{D} = \left\{ \mathbf{y} \in \mathbb{R}^{3n+1} \mid \|\mathbf{y}\| \leq \rho^{-1} \left( 2\sqrt{\lambda_3 K_s} \right) \right\} \quad (4.47)$$

The following subset of  $\mathcal{D}$  can be defined by:

$$\mathcal{S} = \left\{ \mathbf{y}(t) \in \mathcal{D} \mid c\|\mathbf{z}\|^2 < \lambda_3 \left( \rho^{-1} \left( 2\sqrt{\lambda_3 K_s} \right) \right)^2 \right\} \quad (4.48)$$

According to [Bennehar et al., 2018]  $c\|\mathbf{z}\|^2$  is consistently continuous in  $\mathcal{D}$ . So, one can conclude that:

$$c\|\mathbf{z}\|^2 \rightarrow 0 \quad \text{as } t \rightarrow \infty \quad \forall \mathbf{y}(0) \in \mathcal{S} \quad (4.49)$$

Therefore

$$\|\mathbf{e}_1\| \rightarrow 0 \quad \text{as } t \rightarrow \infty \quad \forall \mathbf{y}(0) \in \mathcal{S} \quad (4.50)$$

And here the stability proof of this controller is concluded.

## 4.5 Control Solution 2: A RISE Feedforward controller with Adaptive Feedback Gains

### 4.5.1 Motivation

It is well known that for robotic systems developing tasks where the end-effector is in contact with a stiff environment such as grinding, machining, or assembly tasks, a specific interacting wrench is required to ensure accuracy. Therefore, force contact control may be required. According to [Taghirad, 2013], the problem of force control can be described as to derive the actuator forces to generate a specified desired wrench (force or torque) at the end-element of the manipulator, when the manipulator is performing its desired motion. Therefore, to implement a force control-loop with the positioning control, it is necessary to have wrench sensors on the end-element or directly on the actuators. SPIDER4 is an RA-PKM designed for machining applications such as milling or drilling in resin materials. Nevertheless, for hardware limitations, it is not possible to integrate any wrench sensor to SPIDER4 in order

to implement a force control-loop. As an alternative of the force control-loop, we propose to modify the already presented control solution 1 in the following form. The fixed gains  $\mathbf{K}_s$  and  $\mathbf{\Lambda}_2$  of the RISE control part will be modified as adaptive time-varying. The purpose of this action is because when the tracking errors start to increase due to contact forces on the material to be machined, corrective action will produce to reduce such tracking error values'. If tracking errors become big, then gains  $\mathbf{K}_s$  and  $\mathbf{\Lambda}_2$  also will increase their values. Nevertheless, when the tracking errors will at some fair value, the values for  $\mathbf{K}_s$  and  $\mathbf{\Lambda}_2$  will decrease, adjusting themselves to the best optimum value. The purpose of this modification of the gains is to improve the tracking performance when the robot is performing a machining task. With the above in mind, let us start with the description of the control solution 2.

### 4.5.2 Proposed Control Law

The proposed control law for machining solutions is expressed as follows:

$$\mathbf{\Gamma}(t) = \mathbf{\Gamma}_{ARISE} + \mathbf{\Gamma}_{FW} \quad (4.51)$$

Where  $\mathbf{\Gamma}_{FW} \in \mathbb{R}^n$  corresponds to the nominal feedforward term described in (4.24), and  $\mathbf{\Gamma}_{ARISE} \in \mathbb{R}^n$  is the RISE controller with adaptive feedback gains whose mathematical expression is given as follows:

$$\begin{aligned} \mathbf{\Gamma}_{ARISE}(t) = & (\mathbf{K}_s(t) + \mathbf{I})\mathbf{e}_2(t) - (\mathbf{K}_s(t) + \mathbf{I})\mathbf{e}_2(0) \\ & + \int_0^t [(\mathbf{K}_s(\tau) + \mathbf{I})\mathbf{\Lambda}_2(\tau)\mathbf{e}_2(\tau) + \beta \text{sgn}(\mathbf{e}_2(\tau))]d\tau \end{aligned} \quad (4.52)$$

Where the adaptive gains  $\mathbf{K}_s(t)$  and  $\mathbf{\alpha}_2(t)$  are adjusted by a modification concept of the adaptive gains presented in [Plestan et al., 2010] as follows:

$$\mathbf{K}_s(t) = \bar{\mathbf{K}}_s|\boldsymbol{\eta}| + \mathbf{K}_2 \quad (4.53)$$

$$\mathbf{\Lambda}_2(t) = \bar{\mathbf{\Lambda}}_2|\boldsymbol{\eta}| + \mathbf{K}_3 \quad (4.54)$$

In which,  $\bar{\mathbf{K}}_s$  and  $\bar{\mathbf{\Lambda}}_2 \in \mathbb{R}^{n \times n}$  are positive-definite constant diagonal matrices used in the adaptation process of the controller feedback gains, and  $\mathbf{K}_2, \mathbf{K}_3 \in \mathbb{R}^{n \times n}$  are other positive-definite constant diagonal matrices that establish the minimum possible value for each adaptive feedback gains.  $\boldsymbol{\eta} \in \mathbb{R}^n$  is a nonlinear function depending on the combined joint tracking error  $\mathbf{e}_2$ .

$$\dot{\boldsymbol{\eta}} = \tanh(\mathbf{e}_2) - \boldsymbol{\eta} \quad (4.55)$$

### Controller Design

Let us consider the dynamic model of a PKM presented on (4.6) together with the equations defining the control solution 2 (4.52)-(4.55). To express the closed-loop control system, it is necessary to reformulate the unmeasurable combined tracking error defined in (4.9) as:

$$\mathbf{r}(t) = \dot{\mathbf{e}}_2(t) + \mathbf{\Lambda}_2(t)\mathbf{e}_2(t) \quad (4.56)$$

Where  $\Lambda_2(t)$  was already defined in (4.54). Multiplying both sides of (4.56) and making use of (4.6) and (4.8), the following is obtained:

$$\begin{aligned} \mathbf{M}(\mathbf{q})\mathbf{r} = & \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{\Gamma}_d(t) \\ & + \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d + \Lambda_1\dot{\mathbf{e}}_1 + \Lambda_2(t)\mathbf{e}_2(t)) - \mathbf{\Gamma}_{ARISE}(t) - \mathbf{\Gamma}_{FW}(t) \end{aligned} \quad (4.57)$$

For stability analysis purposes one can compute the derivative with respect to time of (4.57), yielding the following expression:

$$\begin{aligned} \mathbf{M}(\mathbf{q})\dot{\mathbf{r}} = & -\dot{\mathbf{M}}(\mathbf{q})\mathbf{r} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + \dot{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \dot{\mathbf{G}}(\mathbf{q}) + \dot{\mathbf{f}}(\mathbf{q}, \dot{\mathbf{q}}) + \dot{\mathbf{\Gamma}}_d(t) + \mathbf{M}(\mathbf{q})\ddot{q}_d \\ & + \dot{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}}_d + \Lambda_1(\mathbf{M}(\mathbf{q})\ddot{\mathbf{e}}_1 + \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{e}}_1) + \mathbf{M}(\mathbf{q})(\Lambda_2(t)\dot{\mathbf{e}}_2(t) + \dot{\Lambda}_2(t)\mathbf{e}_2(t)) \\ & + \dot{\mathbf{M}}(\mathbf{q})\Lambda_2(t)\mathbf{e}_2(t) - \dot{\mathbf{\Gamma}}_{ARISE} - \dot{\mathbf{\Gamma}}_{FW} \end{aligned} \quad (4.58)$$

Where the time-derivative of  $\mathbf{\Gamma}_{ARISE}$  is expressed as:

$$\dot{\mathbf{\Gamma}}_{ARISE} = (\mathbf{K}_s(t) + \mathbf{I})\dot{\mathbf{e}}_2(t) + [(\mathbf{K}_s(t) + \mathbf{I})\Lambda_2(t) + \dot{\mathbf{K}}_s(t)]\mathbf{e}_2(t) + \beta\text{sgn}(\mathbf{e}_2(t)) \quad (4.59)$$

The closed-loop error system can be expressed in the following form:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{r}} = -\frac{1}{2}\dot{\mathbf{M}}(\mathbf{q})\mathbf{r} + \mathbf{N}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{r}, t) - \mathbf{e}_2(t) - \dot{\mathbf{\Gamma}}_{ARISE} - \dot{\mathbf{\Gamma}}_{FW} \quad (4.60)$$

Where the nonlinear term  $\mathbf{N}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{r}, t) \in \mathbb{R}^n$  is written as:

$$\begin{aligned} \mathbf{N}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{r}, t) = & \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + \dot{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \dot{\mathbf{G}}(\mathbf{q}) + \dot{\mathbf{f}}(\mathbf{q}, \dot{\mathbf{q}}) + \dot{\mathbf{\Gamma}}_d(t) + \mathbf{M}(\mathbf{q})\ddot{q}_d \\ & + \dot{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}}_d + \Lambda_1(\mathbf{M}(\mathbf{q})\ddot{\mathbf{e}}_1 + \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{e}}_1) + \mathbf{M}(\mathbf{q})(\Lambda_2(t)\dot{\mathbf{e}}_2(t) \\ & + \dot{\Lambda}_2(t)\mathbf{e}_2(t)) + \dot{\mathbf{M}}(\mathbf{q})\Lambda_2(t)\mathbf{e}_2(t) + \mathbf{e}_2(t) - \frac{1}{2}\dot{\mathbf{M}}(\mathbf{q})\mathbf{r} \end{aligned} \quad (4.61)$$

For stability analysis purposes (4.60) is re-written following the same procedure as in the control solution 1.

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{r}} = -\frac{1}{2}\dot{\mathbf{M}}(\mathbf{q})\mathbf{r} + \mathbf{N}_d + \tilde{\mathbf{N}} - \mathbf{e}_2(t) - \dot{\mathbf{\Gamma}}_{ARISE} - \mathbf{N}_{FW} \quad (4.62)$$

In which,  $\tilde{\mathbf{N}} = \mathbf{N} - \mathbf{N}_d$  begin  $\mathbf{N}_d$  defined in (4.17). Since the nominal feedforward term for this control solution is the same as the previous one controller, one can establish  $\mathbf{N}_{FW} = \dot{\mathbf{\Gamma}}_{FW}$ .

### Criterion for Adaptive Gains

Since the nonlinear functions  $\mathbf{K}_s(t)$  and  $\Lambda_2(t)$  are continuously differentiable, they can be bounded in the following form:

$$\mathbf{K}_2 \leq \mathbf{K}_s(t) \leq \mathbf{K}_{sM} \quad (4.63)$$

$$\mathbf{K}_3 \leq \Lambda_2(t) \leq \Lambda_{2M} \quad (4.64)$$

Where  $\mathbf{K}_{sM}$  and  $\Lambda_{2M} \in \mathbb{R}^{n \times n}$  represent the maximum admissible values that  $\mathbf{K}_s(t)$  and  $\Lambda_2(t)$  can obtain.

### Stability Analysis

**Theorem 2** *The tracking error in joint space  $\mathbf{e}_1$  of a PKM whose dynamics is given by (4.6) under the control law (4.51) converges asymptotically to zero as long as time goes to infinity if the following design parameters are selected such that:*

- i)  $\Lambda_1 > \frac{1}{2}$
- ii)  $\Lambda_{2M} > K_3$
- iii)  $\beta > \|\mathbf{N}_d(t)\|_{\mathcal{L}^\infty} - \|\mathbf{N}_{FW}(t)\|_{\mathcal{L}^\infty} + \frac{1}{\Lambda_{2M}} \left( \|\dot{\mathbf{N}}_d(t)\|_{\mathcal{L}^\infty} - \|\dot{\mathbf{N}}_{FW}(t)\|_{\mathcal{L}^\infty} \right)$

**Proof 3** *Considering the same auxiliary function  $\mathbf{L}(t)$  defined in (4.28), the following inequality is holding:*

$$\int_0^t \mathbf{L}(\tau) d\tau \leq \beta \|\mathbf{e}_2(0)\| + \mathbf{e}_2(0)^T (\mathbf{N}_{FW}(0) - \mathbf{N}_d(0)) \quad (4.65)$$

If  $\beta$  is now selected to satisfied the following condition:

$$\beta > \|\mathbf{N}_d(t)\|_{\mathcal{L}^\infty} - \|\mathbf{N}_{FW}(t)\|_{\mathcal{L}^\infty} + \frac{1}{\Lambda_{2M}} \left( \|\dot{\mathbf{N}}_d(t)\|_{\mathcal{L}^\infty} - \|\dot{\mathbf{N}}_{FW}(t)\|_{\mathcal{L}^\infty} \right) \quad (4.66)$$

Let us consider the Lyapunov function defined in (4.36)

$$\mathbf{V}(\mathbf{y}, t) = \frac{1}{2} \mathbf{r}^T \mathbf{M}(\mathbf{q}) \mathbf{r} + \mathbf{e}_1^T \mathbf{e}_1 + \frac{1}{2} \mathbf{e}_2^T \mathbf{e}_2 + \mathbf{P}$$

Developing the time-derivative of (4.36) and making use of (4.35), (4.60), (4.59), (4.56), and (4.8), the following equation results:

$$\dot{\mathbf{V}}(\mathbf{y}, t) = \mathbf{r}^T \tilde{\mathbf{N}} - (K_s(t) + 1) \mathbf{r}^T \mathbf{r} - \dot{K}_s(t) \mathbf{r}^T \mathbf{e}_2 + 2\mathbf{e}_1^T \mathbf{e}_2 - 2\Lambda_1 \mathbf{e}_1^T \mathbf{e}_1 - \Lambda_2(t) \mathbf{e}_2^T \mathbf{e}_2 \quad (4.67)$$

Considering the lower bounds for  $\mathbf{K}_s(t)$  and  $\Lambda_2(t)$  and the relationships presented in (4.41), expression (4.67) can be bounded as follows:

$$\begin{aligned} \dot{\mathbf{V}}(\mathbf{y}, t) \leq & \|\mathbf{r}\| \rho(\|\mathbf{z}\|) \|\mathbf{z}\| - (K_2 + 1) \|\mathbf{r}\|^2 - \frac{|K_{sdm}|}{2} \|\mathbf{r}\|^2 - \frac{|K_{sdm}|}{2} \|\mathbf{e}_2\|^2 \\ & + \|\mathbf{e}_1\|^2 + \|\mathbf{e}_2\|^2 - 2\Lambda_1 \|\mathbf{e}_1\|^2 - K_3 \|\mathbf{e}_2\|^2 \end{aligned} \quad (4.68)$$

Where  $K_{sdm}$  denotes a lower bound for  $\dot{\mathbf{K}}_s(t)$  as is explained in [Saied et al., 2019a]. The previous equation can be re-written in the following form:

$$\dot{\mathbf{V}}(\mathbf{y}, t) \leq -\lambda_3 \|\mathbf{z}\|^2 - \left( K_2 \|\mathbf{r}\|^2 - \rho(\|\mathbf{z}\|) \|\mathbf{r}\| \|\mathbf{z}\| \right) \quad (4.69)$$

In which,  $\mathbf{z}(t) \in \mathbb{R}^{3n}$  represents the vector containing the different tracking errors of the system described by equation (4.21), and  $\lambda_3 = \min\{\eta_1, \eta_2, \eta_3\}$ , where the constants  $\eta_1$ ,  $\eta_2$ , and  $\eta_3$  are established as:

$$\eta_1 = 2\Lambda_1 - 1, \quad \eta_2 = \frac{|K_{sdm}|}{2} + K_3 - 1, \quad \eta_3 = \frac{|K_{sdm}|}{2} + 1 \quad (4.70)$$



Form (4.70) one can see that  $\Lambda_1$  must be chosen such that  $\Lambda_1 > 1/2$ . Completing the squares for the second and last term of (4.69), the following expression is obtained:

$$\dot{\mathbf{V}}(\mathbf{y}, t) \leq \lambda_3 \|\mathbf{z}\|^2 - \frac{\rho^2(\mathbf{z}) \|\mathbf{z}\|^2}{4K_2} = -c \|\mathbf{z}\|^2 \quad (4.71)$$

In (4.71) the term  $c \|\mathbf{z}\|^2$  denotes a continuous positive semi-definite function which is specified in the following domain:

$$\mathcal{D} = \left\{ \mathbf{y} \in \mathbb{R}^{3n+1} \mid \|\mathbf{y}\| \leq \rho^{-1} \left( 2\sqrt{\lambda_3 K_2} \right) \right\} \quad (4.72)$$

The following subset of  $\mathcal{D}$  can be defined by:

$$\mathcal{S} = \left\{ \mathbf{y}(t) \in \mathcal{D} \mid c \|\mathbf{z}\|^2 < \lambda_3 \left( \rho^{-1} \left( 2\sqrt{\lambda_3 K_2} \right) \right)^2 \right\} \quad (4.73)$$

According to [Bennehar et al., 2018]  $c \|\mathbf{z}\|^2$  is consistently continuous in  $\mathcal{D}$ . So, one can conclude that:

$$c \|\mathbf{z}\|^2 \rightarrow 0 \quad \text{as } t \rightarrow \infty \quad \forall \mathbf{y}(0) \in \mathcal{S} \quad (4.74)$$

Therefore

$$\|\mathbf{e}_1\| \rightarrow 0 \quad \text{as } t \rightarrow \infty \quad \forall \mathbf{y}(0) \in \mathcal{S} \quad (4.75)$$

And here the stability proof of this controller is concluded.

## 4.6 The actuation redundancy on SPIDER4

In Chapter 1, it was explained the different challenges involved in the control of PKMs. One of them is the actuation redundancy that is present on Parallel robots having more actuators than DOF [Mueller, 2011]. Such is the case of the SPIDER4, which has 5-DOF (3T-2T), where the positioning device develops the 3T-DOF using four actuators located on the fixed base. We mentioned that robotics systems with this type of configuration have advantages compared with their non-redundant counterparts, such as higher accuracy and improved stiffness. Furthermore, actuation redundancy can also lead to singularity-free large workspaces. However, this configuration leads to the generation of internal forces that may create pre-stress in the mechanism without operational motions, and it can damage the mechanical structure of the robot. According to [Hufnagel and Muller, 2012], the internal forces can be caused by geometric uncertainties and may be amplified by the use of decentralized control techniques. This may lead to uncoordinated control of the individual actuators since such a control law does not account for kinematic constraints. In order to avoid such an issue, the use of a projection matrix can be used, which is based on the pseudo inverse jacobian matrix evaluated with the desired variables  $\mathbf{H}(\mathbf{q}_d, {}^o \mathbf{N}_{Nd}) \in \mathbb{R}^{m \times n}$ . The projection operator is defined by:

$$\mathbf{R}_H = (\mathbf{H}^+)^T \mathbf{H}^T \quad (4.76)$$

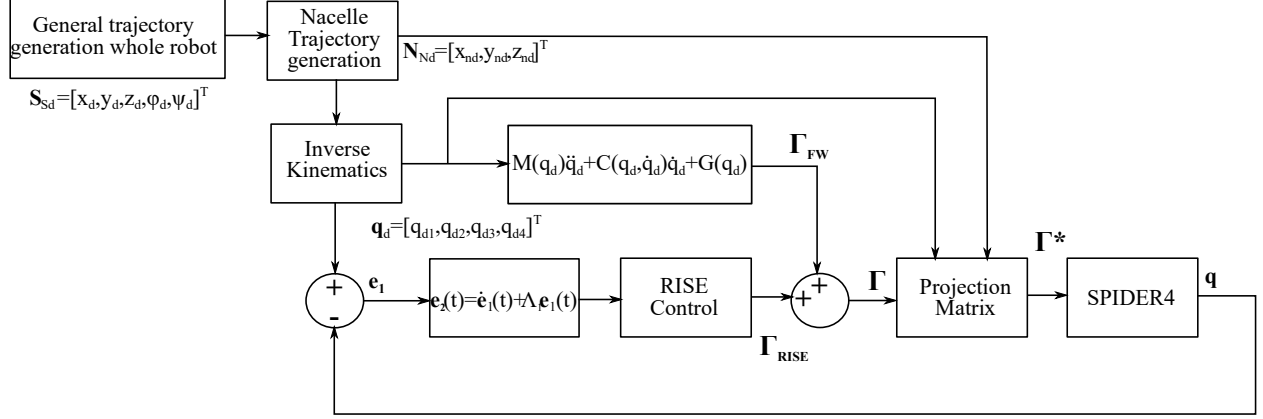


Figure 4.4: Block-diagram of the proposed control solution 1 for the parallel device of SPIDER4 RA-PKM

The projection matrix  $\mathbf{R}_H$  eliminates the control inputs in the null-space of  $\mathbf{H}^T$ . Hence, all control inputs applied to the manipulator have to be 'regularized' using this projection matrix as follows:

$$\mathbf{\Gamma}^* = \mathbf{R}_H \mathbf{\Gamma} \quad (4.77)$$

Where  $\mathbf{\Gamma}$  denotes the torques vector generated by the control scheme. Therefore, all torques generated by the presented control solutions 1, and 2 for SPIDER4, must be regularized before sending them to the manipulator. Figures 4.4 and 4.5 present block diagrams of the proposed control solution 1 and 2 including the projection operator.

## 4.7 Control Solution 3: A RISE controller with Adaptive Feedforward Compensation Based-on B-Spline Neural Networks

### 4.7.1 Motivation

The main objective of a Delta PKM is to perform high speed and high accuracy P&P operations with the smallest possible tracking error. To reach this objective, it is crucially essential to design a control scheme capable of keeping the precision under abrupt changes of mass and acceleration. To satisfy these demands, we propose integrating the RISE control algorithm with an adaptive feedforward compensation term. The main feature of RISE controller can ensure semi-global asymptotic stability in the presence of general uncertain disturbances [Xian et al., 2004]. It is well known in robotics control that the addition of a feedforward term can compensate the inherent nonlinearities and improve the system performance. However, sometimes, the dynamic model or dynamic parameters as masses and inertia are unknown or not measurable. Consequently, wrong parameter estimation or an inaccurate dynamic model can harm the efficiency of the control scheme instead of improving. ANNs are an attractive

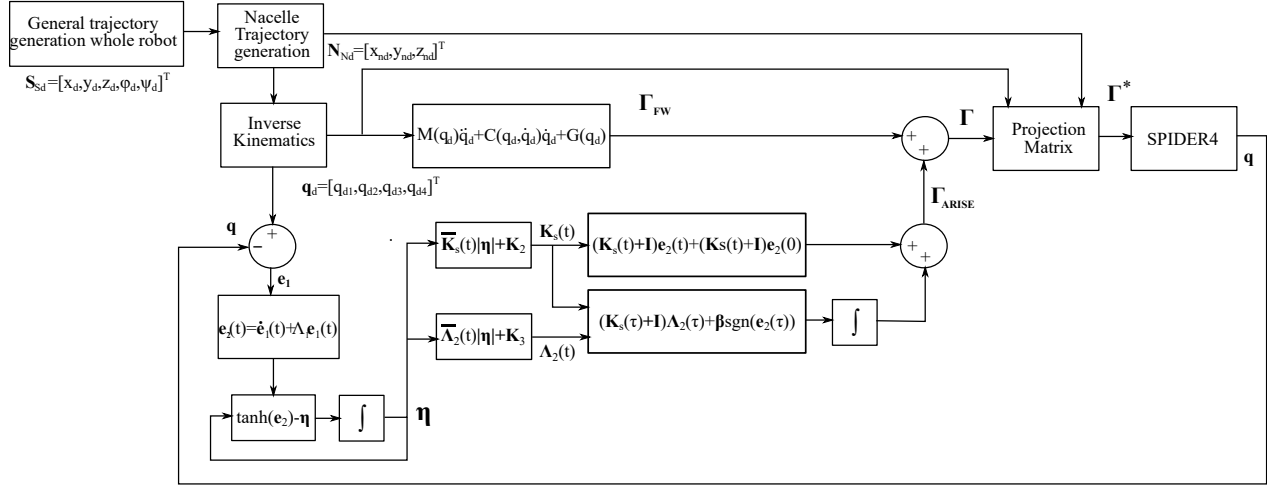


Figure 4.5: Block-diagram of the proposed control solution 2 for the parallel device of SPIDER4 RA-PKM

solution for nonlinear modeling systems due to their ability to identify unknown dynamic models through a set of inputs and outputs related to each other. BSNN is a kind of ANN formed by three parts: A lattice used to normalize the inputs, a single layer set of basis functions defined over the lattice, and the network output, which is a linear combination of the basis functions with the adjustable weights [Lopes et al., 2000]. This ANN is very suitable for nonlinear model identification in real-time due to its construction formed by only one hidden layer of basis functions avoiding large number calculus compared to any multilayer ANN. In this thesis, we employed BSNNs to approximate the inverse dynamics for each kinematic chain of the Delta PKM.

### 4.7.2 Controller Design

Having in mind the benefits of RISE control and ANN, we establish the following control scheme for the Delta PKM:

$$\Gamma = \Gamma_{RISE} + \hat{\Sigma}(q_d, \dot{q}_d, \ddot{q}_d, e_2) \quad (4.78)$$

where  $\Gamma_{RISE} \in \mathbb{R}^n$  corresponds to feedback RISE control (4.12), and the term  $\hat{\Sigma}(q_d, \dot{q}_d, \ddot{q}_d, e_2) \in \mathbb{R}^n$  is the intelligent compensation term vector based on BSNNs whose elements are defined as:

$$\hat{\Sigma} = [\hat{\sigma}_1 \quad \hat{\sigma}_2 \quad \hat{\sigma}_3]^T \quad (4.79)$$

where  $\hat{\sigma}_i \quad \forall i = 1, 2, \dots, n$  denotes the respective BSNN output used to approximate the dynamics of one kinematic chain of Delta PKM. In Figure 4.6 general overview of the proposed control technique applied to a Delta PKM is illustrated.

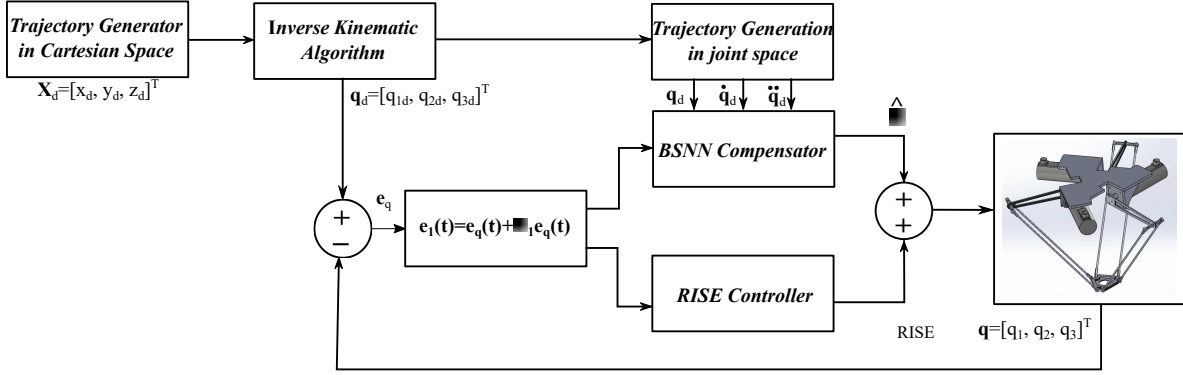


Figure 4.6: Representation of the proposed control solution 3 for Delta PKM

### Design of the feedforward term based on BSNNs

As it was mentioned above, the BSNNs aims to estimate on-line the dynamic behavior of the DPR to include it into the control loop as a feedforward compensation term. Let us remember the expression corresponding for the nominal feedforward term (4.24).

$$\mathbf{M}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \mathbf{G}(\mathbf{q}_d) = \mathbf{\Gamma}_{FW}$$

In this proposed control solution  $\mathbf{M} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{C} \in \mathbb{R}^{n \times n}$ , and  $\mathbf{G} \in \mathbb{R}^n$  are considered unknown. From (4.24), one can see that the Inertia, Centripetal/Coriolis matrices, and the gravity vector are evaluated with the desired trajectories  $\mathbf{q}_d$ ,  $\dot{\mathbf{q}}_d$ ,  $\ddot{\mathbf{q}}_d$  instead of the measured ones. Therefore, we set the trajectories values as the data input for the BSNNs. An important aspect of the design of each BSNN is to define the input space lattice formed by a set of  $n$  knot-vectors, one-knot vector for each input axis. Once the input data is established, the next step is to define the  $K$  order, shape, number, and distribution of the basis functions. The  $K$  order defines the shape of the basis functions, i.e., if  $K = 1$ , we obtain piecewise constant functions,  $K = 2$  leads to piecewise linear functions,  $K = 3$  generates piecewise quadratic functions and, when  $K = 4$  piecewise cubic functions are obtained. Selecting a higher-order for the functions result in a better approximation. The number of knots and the value of each one, as well as the interval between them, are set by prior knowledge of the selected BSNN inputs. Dynamics of Parallel Robots are highly and complex; thus, we selected basis functions of third-order to acquire an accurate approximation of the dynamics behavior without making a greater number of calculations as may occur with cubic functions. A knot-vector is defined for each input axis considering the extreme admissible values of the trajectories as the maximum and minimum values of the input vectors. For the axes where  $\mathbf{q}_d$  are the inputs the minimum and maximum values are from  $-1$  to  $1$  rad respectively,  $-10$  to  $10$  rad/s for  $\dot{\mathbf{q}}_d$  and  $-200$  to  $200$  rad/s<sup>2</sup> for  $\ddot{\mathbf{q}}_d$ . Once the input range is already defined for the input axes, the next step is to define the number and distribution of  $j$ -th knots of the vector. Each knot-vector is formed by 8 knot-points and they are distributed in groups of four elements to generate three b-spline functions that share some knot-points among them. We selected this configuration because it gives a good approximation of the system behavior, as being reported in the results section. In Figure 4.7, the distribution of the knot-points and B-spline functions for each input axis are depicted. The knot-points values for the input axes are given in Table 4.1.

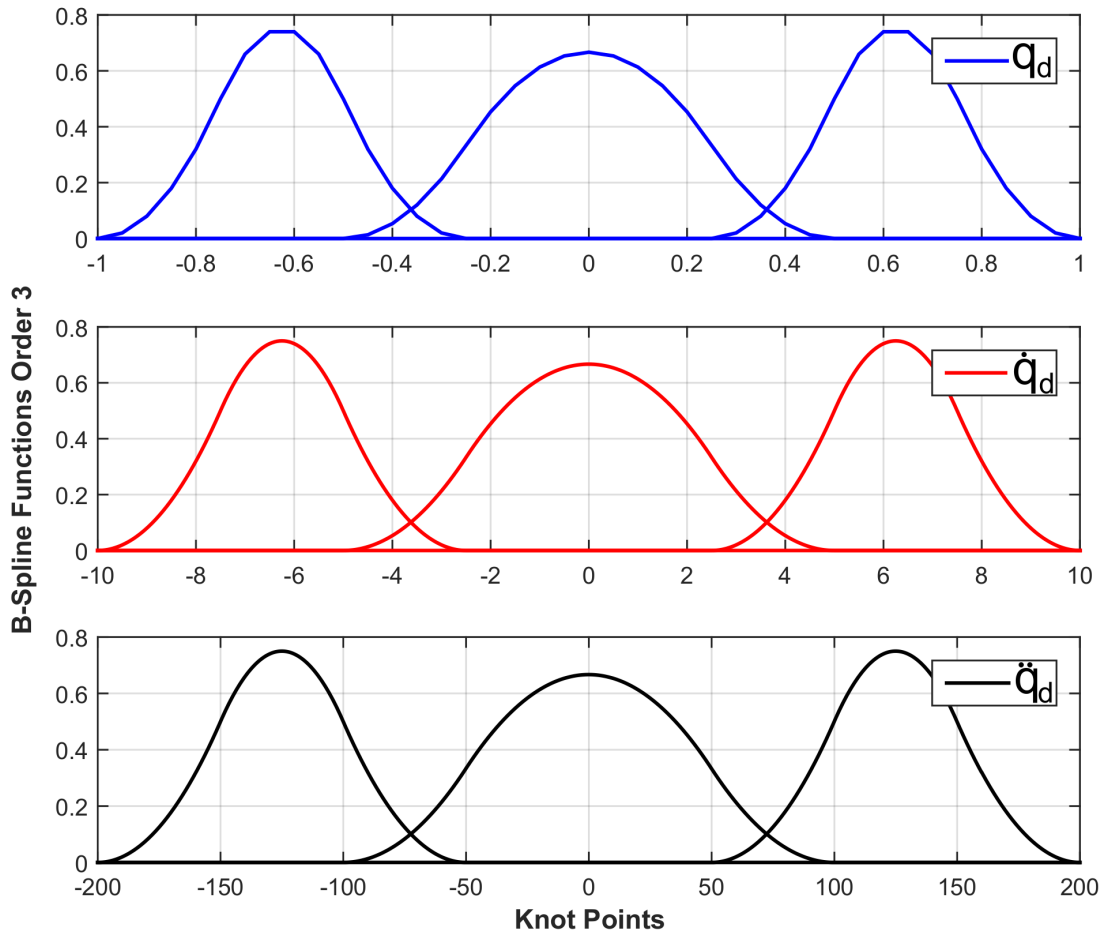


Figure 4.7: Distribution of the proposed activation functions of order 3 for the respective inputs

Table 4.1: Knot-points Vectors' distribution

Input	Knot-points Vector
$q_d$	$[-1 \quad -0.75 \quad -0.5 \quad -0.25]$
	$[-0.5 \quad -0.25 \quad 0.25 \quad 0.5]$
	$[0.25 \quad 0.5 \quad 0.75 \quad 1]$
$\dot{q}_d$	$[-10 \quad -7.5 \quad -5 \quad -2.5]$
	$[-5 \quad -2.5 \quad 2.5 \quad 5]$
	$[2.5 \quad 5 \quad 7.5 \quad 10]$
$\ddot{q}_d$	$[-200 \quad -150 \quad -100 \quad -50]$
	$[-100 \quad -50 \quad 50 \quad 100]$
	$[50 \quad 100 \quad 150 \quad 200]$

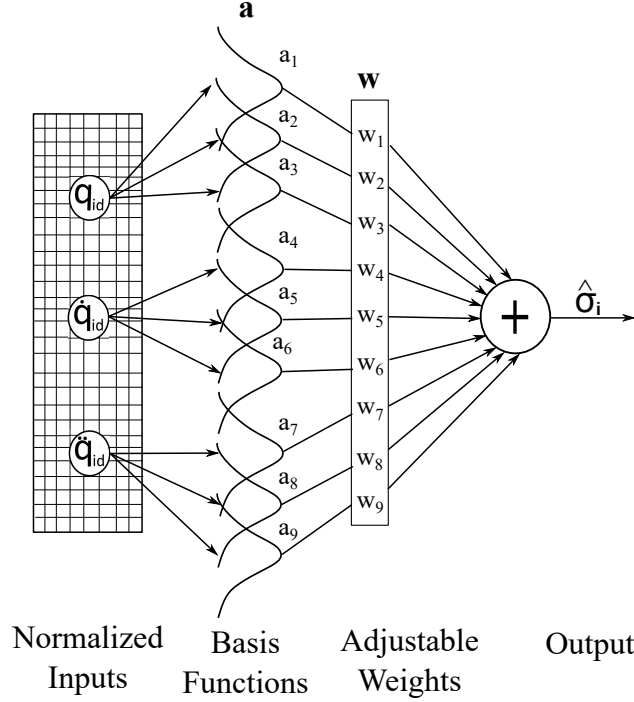


Figure 4.8: Diagram of the proposed BSNN used as a compensation term for each kinematic chain of the Delta PKM

### The B-Spline Function

We proceeded to present the expression of univariate B-Spline basis function, which is defined through the following recurrence relationship [Mirea, 2014]:

$$\begin{aligned}
 S_K^j(u) &= \left( \frac{u - \lambda_{j-K}}{\lambda_{j-1} - \lambda_{j-K}} \right) S_{K-1}^{j-1}(u) + \left( \frac{\lambda_j - u}{\lambda_j - \lambda_{j-K+1}} \right) S_{K-1}^j(u) \\
 S_1^j(u) &= \begin{cases} 1 & \text{if } u \in I_j \\ 0 & \text{other cases} \end{cases}
 \end{aligned} \tag{4.80}$$

where  $u$  corresponds to the input,  $\lambda_j$  is the  $j$ th knot point and  $I_j = [\lambda_{j-1}, \lambda_j)$  is the  $j$ th interval between two-knot points, and  $K$  is the order of the output function. The output of each one of the BSNN can be written as follows [Brown and Harris, 1994]:

$$\hat{\sigma}_i = \sum_{m=1}^P a_m w_m = \mathbf{a}_i^T \mathbf{w}_i \quad \forall i = 1, 2, 3 \tag{4.81}$$

where  $\mathbf{a}_i$  is a  $P$ -dimensional vector which contains the outputs of the BSNN basis functions and,  $\mathbf{w}_i$  is the weights vector. The diagram depicted in Figure 4.8 represents the BSNN configuration for the Delta PKM dynamic estimation.

### Learning Rule

An instantaneous training algorithm is used for the BSNN; this algorithm only adjusts the weights corresponding to the active basis functions. The instantaneous learning rule is formulated, minimizing an instantaneous estimation of a performance function of the Mean Square Error (MSE) of the output, and the parameters are updated using descending gradient rules. The MSE estimate is given by:

$$J_e(t) = (\hat{\sigma}(t) - \sigma(t))^2 \quad (4.82)$$

A variation of the standard descending gradient is the Normalized Least Mean Square (NLMS) algorithm employed for instantaneous training. We used this formulation as a learning rule because it uses few computational resources, which is essential for real-time implementation. The learning rule is given as follows [Brown and Harris, 1994]:

$$\mathbf{w}_i = \mathbf{w}_i(t-1) + \frac{\gamma \tilde{\sigma}_i(t)}{\|\mathbf{a}_i(t)\|_2^2} \mathbf{a}_i(t) \quad \forall i = 1, 2, 3 \quad (4.83)$$

where  $\gamma$  is the learning rate,  $\mathbf{a}_i$  is the vector that contains the output of the basis functions,  $\mathbf{W}_i$  is the adjustable weights vector, and  $\tilde{\sigma}_i(t) = \sigma_i(t) - \hat{\sigma}_i(t)$  is the BSNN output error. To do the on-line training of the BSNN, it is necessary an error signal that is the difference between the real variable and the estimated by the BSNN. However, in this case, the real value is not available since it is required to obtain through the BSNN. For this reason, it is consistent with using the measurement of the robot's position and comparing it with the values of the established desired trajectory to obtain an error signal. In this case,  $\tilde{\sigma}_i(t)$  is estimated using the composed tracking error  $\mathbf{e}_1$ , as illustrated in Figure 4.6.

## 4.8 Conclusion

In this chapter, the thesis contributions related to control for PKMs were presented. Three control solutions were detailed, emphasizing the purpose of why they were proposed based on the applications that the experimental platforms develop. The proposed control solutions were: A RISE control with nominal feedforward, a RISE feedforward with adaptive feedback gains, and a RISE control with adaptive feedforward compensation based-on B-Spline neural networks. The first controller was proposed to validate the effectiveness of the developed kinematic and dynamic for SPIDER4 RA-PKM. The second one was designed to improve the tracking performance of a PKM as SPIDER4 that develops tasks were contact forces are involved as machining; there are not any feedback sensors to measure such contact forces. Finally, the third one controller was conceived to estimate on-line the dynamics of the manipulator to be controlled and include such estimations as a feedforward compensation term. For SPIDER4, a method to deal with the problem of internal forces generation due to its redundantly actuated configuration was detailed. Moreover, the stability analysis for control solutions 1 and 2 was formulated. In the next thesis chapter, the real-time experimental results using control solutions 1 and 2 on SPIDER4 will be presented.





## **Part III**

# **Real-time Experiments and Numerical Simulations**





# Experimental Results on SPIDER4 RA-PKM

*In this chapter, the real-time experimental results of the control solutions 1 and 2 implemented on SPIDER4 RA-PKM are presented and discussed. The experimental setup and programming issues for the controllers are also explained in detail. Several graphs and performance indexes are used to demonstrate the effectiveness of the proposed control solutions under different scenarios.*

## Contents

<b>5.1</b>	<b>Introduction</b>	<b>99</b>
<b>5.2</b>	<b>Description of the Experimental Setup, and Implementation Issues</b>	<b>100</b>
<b>5.3</b>	<b>Experimental results of control solution 1</b>	<b>109</b>
<b>5.4</b>	<b>Experimental results of control solution 2</b>	<b>112</b>
<b>5.5</b>	<b>Conclusion</b>	<b>120</b>

## 5.1 Introduction

The proposed RISE with nominal feedforward and RISE feedforward with adaptive gains are validated via real-time experiments under different scenarios. The control solution 1 is only evaluated in a nominal scenario, which is free motion without any physical contact on the spindle tip because the objective of the first control solution is only to prove the veracity of the proposed IDM of SPIDER4. The obtained performance of RISE with nominal feedforward is evaluated against the produced of the standard RISE control. A machining scenario is used to validate the proposed control solution 2. The machining experiments are carried out on blocks of styrofoam, which is an excellent material to perform the first machining tests of SPIDER4. The machining scenario is performed under three forward speeds (low,

medium, and high) in order to see how the precision is affected when the same task is executed at different speeds. The RISE's obtained performance feedforward with adaptive gains is compared to that produced by Standard RISE and RISE with nominal feedforward controllers. Without further ado, let us start with the description of the experimental setup of SPIDER4 and its implementation issues.

## 5.2 Description of the Experimental Setup, and Implementation Issues

### 5.2.1 Description of Experimental Setup

Figure 5.1 presents an overview of SPIDER4 in its work cell, including the tooling plate used to fix the material block to be machined. SPIDER4 uses different actuators to perform the linear motions of the parallel positioning device and the wrist. SPIDER4. Four WITTESTEIN TPMA110S-022M-6KB1-220H-W6 are responsible for performing the movements on  $x$ ,  $y$ , and  $z$  axis, each actuator include a gearhead with a gear ratio of 1:22 providing a peak torque of 3100  $Nm$  and 189 rpm as maximum rotation speed. In order to measure the joint positions, these motors are equipped with multiturn absolute encoders. On the nacelle, three actuators perform the independent angular motions  $\phi$  and  $\psi$  as well as the spindle's movement. The motor responsible for the movement in the  $\phi$  axis is STOBBER EZH501USVC4P097 motor; this actuator can generate a peak torque of 200  $Nm$ . To perform the angular motion in the  $\psi$  axis, a HARMONIC DRIVE CHA-20A-30-H-M1024-B is used, this motor can provide a peak torque of 27  $Nm$ . Finally, the B&R 8JSA24.E4080D000-0 is used for the actuation of the spindle machining tool. This motor can provide a torque of 1.41  $Nm$  and a related speed of 8000 *r.p.m.* As was mentioned in Chapter 3, the implemented control solutions affect only the main motors located on the fixed base. However, it is worth mentioning the features of the other motors located on the nacelle. The joint velocities of SPIDER4 are not measured directly because the robot is not equipped with sensors to measure the joint velocities. Notwithstanding, they are calculated through numerical derivatives to form the joint positions' measurements with a sampled time of 0.0004 sec. The structure of each control scheme is created in Simulink software from MathWorks. The Simulink project contains the kinematic algorithms, Jacobian matrices, and the control scheme for SPIDER4. However, all the functionalities of SPIDER4, including the motion control, are programmed in a B&R Automation Studio project. Therefore, a library called B&R Automation studio target for Simulink is used to convert the Simulink code to C code and transfer it to the B&R Automation Studio Project. When the all changes have been carried out to the project, the next step is to compile and transfer them to the B&R Automation PC 910 with a sampling time of 0.4 msec, sending and receiving the control signals to the X20 system which in turn sends and receives signals to the inverted modules called ACOPOSmulti system regulating each SPIDER4 motor. The SPIDER4 project is executed through a Graphic User Interface (GUI) developed by Tecnalía Company and also programmed in B&R Automation Studio. The experimental setup for SPIDER4 with the main components is illustrate in Figure 5.2.

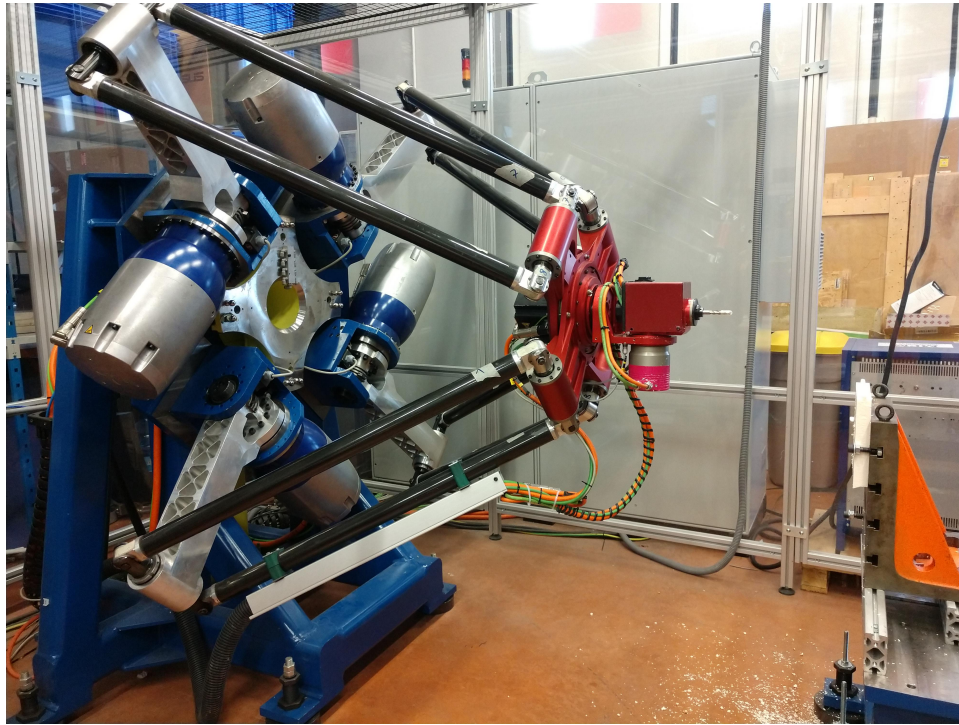


Figure 5.1: Overview of SPIDER4 RA-PKM in its work environment

### 5.2.2 Reference Trajectory Generation

To program a machining task on a CNC machine, the majority of them used the programming language developed by the Electronics Industry Association (EIA) in the 1960's named RS-274D but this programming language is better known as "G-code" or "G&M-code" because many of the instructions of this language starts with the letters G and M. The most common addressed codes are described for this programming language are given in Table 5.1. For SPIDER4, the desired trajectories for the tested controllers are generated using the CNC functionality of B&R AUTOMATION STUDIO. There exist many instructions to programming CNC machines using G and M functions. However, only the codes used in the presented experiments on SPIDER4 will be detailed.

- **M3**: Starts the spindle clockwise
- **M30**: Ends program
- **F**: Forward speed. For B&R Automation Studio the units are mm/min.
- **G1**: Linear motion. This command moves the tool in a straight line with the programmed forward speed. e.g., Suppose that the spindle of the robot is initially in X0 Y0, and one wants to move it to X10 Y15; the following line code is witted **G1 X10 Y15 F1000** (see Figure 5.3a).

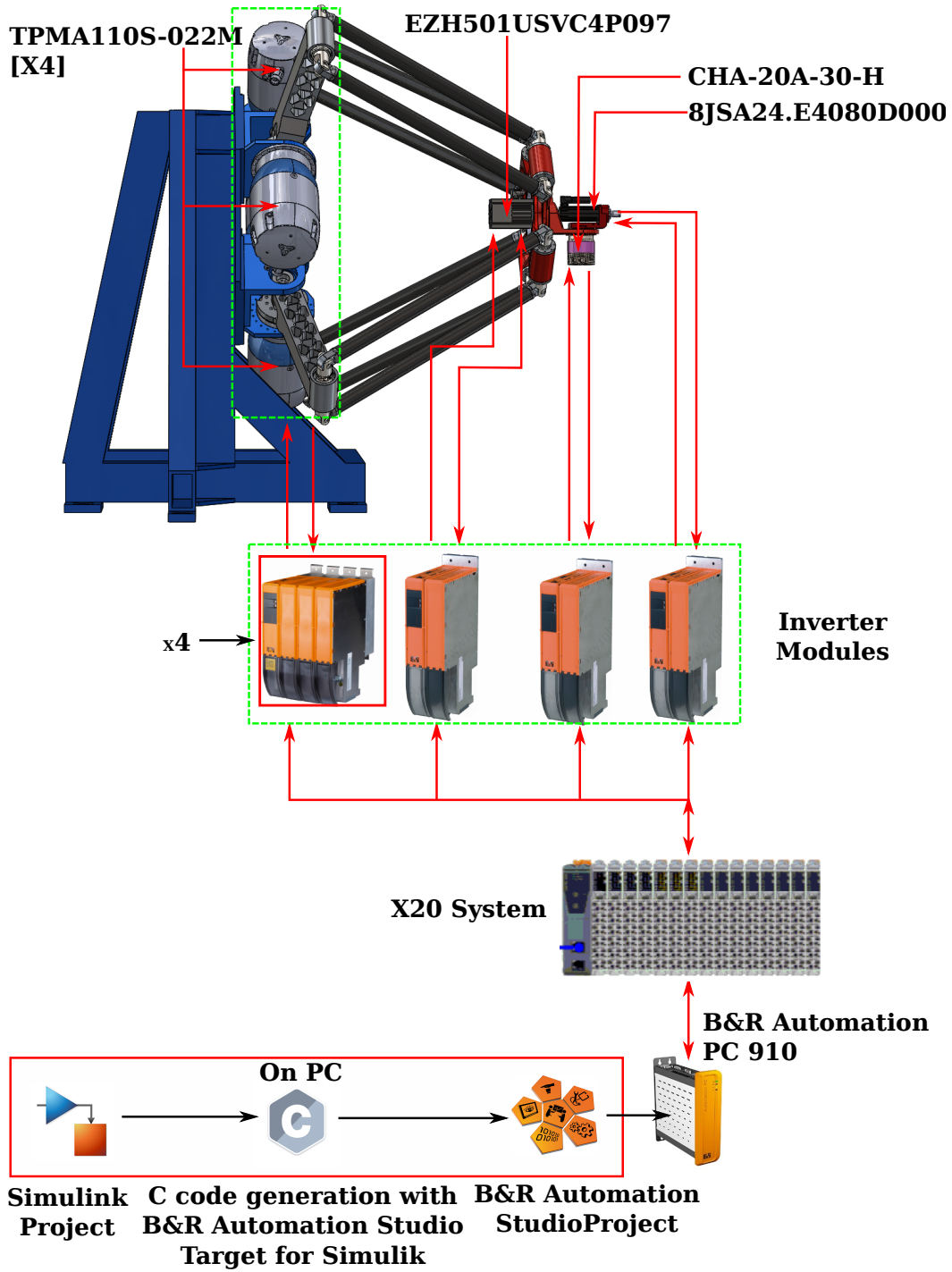


Figure 5.2: Illustration of experimental setup of the SPIDER4

Table 5.1: Common Alphanumeric Address Codes

Code	Meaning	Code	Meaning
A	Rotation about X-Axis	O	Program number
B	Rotation about Y-Axis	P	Dwell time
C	Rotation about Z-Axis	Q	Used in drill cycles
F	Forward Speed	R	Arc Radius
G	G-Code (preparatory code)	S	Spindle speed
H	Tool Length Offset (TLO)	T	Tool number
I	Arc center X-vector	X	X-coordinate
J	Arc center Y-vector	Y	Y-coordinate
K	Arc center Z-vector	Z	Z-coordinate
M	M-Code (Miscellaneous code)	N	Block Number

- **G2:** Circular interpolation clockwise. This command moves the tool in a circular arc with the programmed forward speed. There are two forms to use this instruction one is considering the I, J, and K instructions as follows. Suppose that initially, the spindle of the robot is at position X0 Y20 and it is desired to move it to point X20 Y0 forming a quarter circle whose center is at position X0 Y0 with a forward speed of 1000 mm/min. The programming line would be expressed as **G2 X20 Y0 I0 J-20 F1000**. Where I and J represents the offset on the X and Y axis, respectively (see Figure 5.3b for a better understanding). The other way to program G2 is by using the ratio R specification. The previous line code can be written as **G2 X20 Y0 R20 F1000**.
- **G3:** Circular interpolation counterclockwise. This command is basically the same as G2, but the interpolation is performed on the opposite side.

### Trajectory for Scenario 1

Scenario 1 consists of a free motion trajectory where the SPIDER4 spindle tip is not in contact with any material to be machined. This trajectory is used to validate the RISE control with nominal feedforward. Linear and circular interpolation are used to generate the path to follow. Moreover, the trajectory integrates angular displacements for  $\phi$  and  $\psi$  axes whose movements are controlled independently to the proposed controller. The established forward speed F for all trajectory is  $12000\text{mm}/\text{min}$ . Figure 5.4 depicts the trajectories evolution with respect to time and the tridimensional representation. The points sequence of this trajectory is in Appendix B. However, because of the proposed control solutions for SPIDER4 only cover the parallel mechanism without the wrist, it is necessary to provide the reference trajectories just for the nacelle. Nacelle trajectories in Cartesian space are computed evaluating (3.36) with the data provided in Figure 5.4, the resulting trajectories for the nacelle are shown in Figure 5.5.

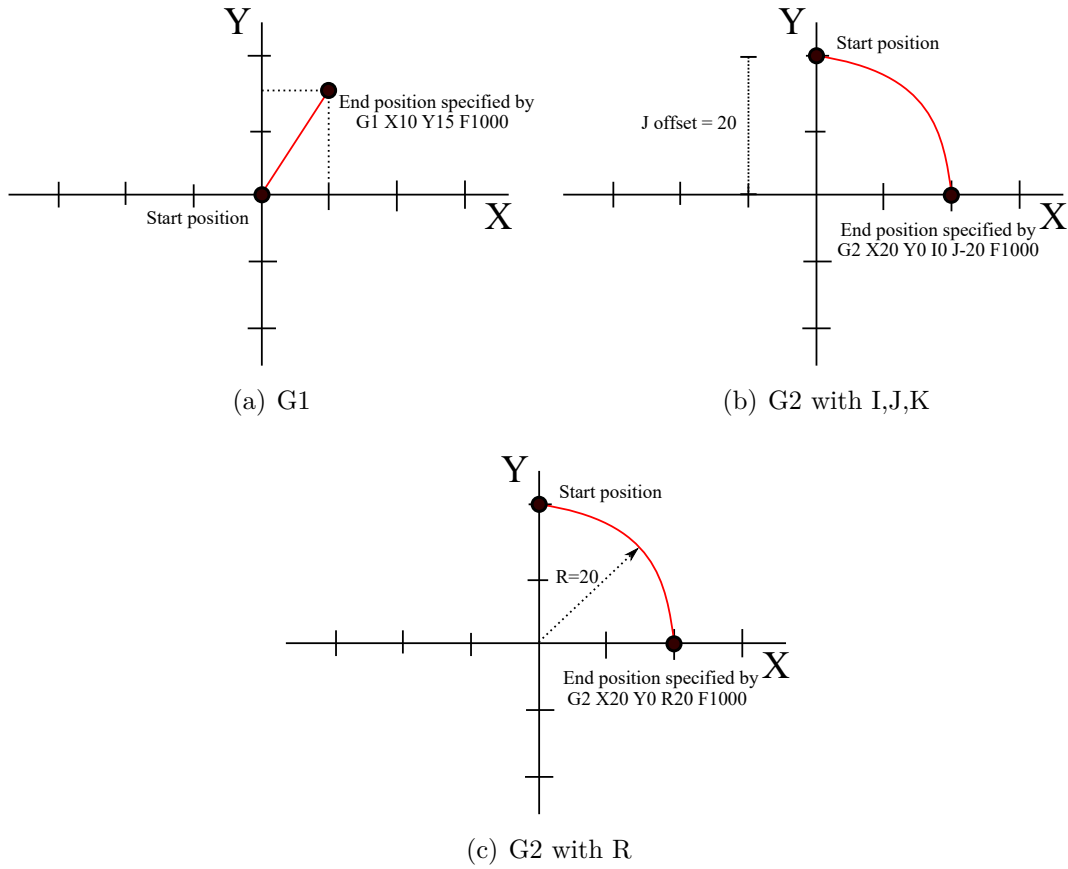


Figure 5.3: Interpolation examples using G1 and G2 instructions

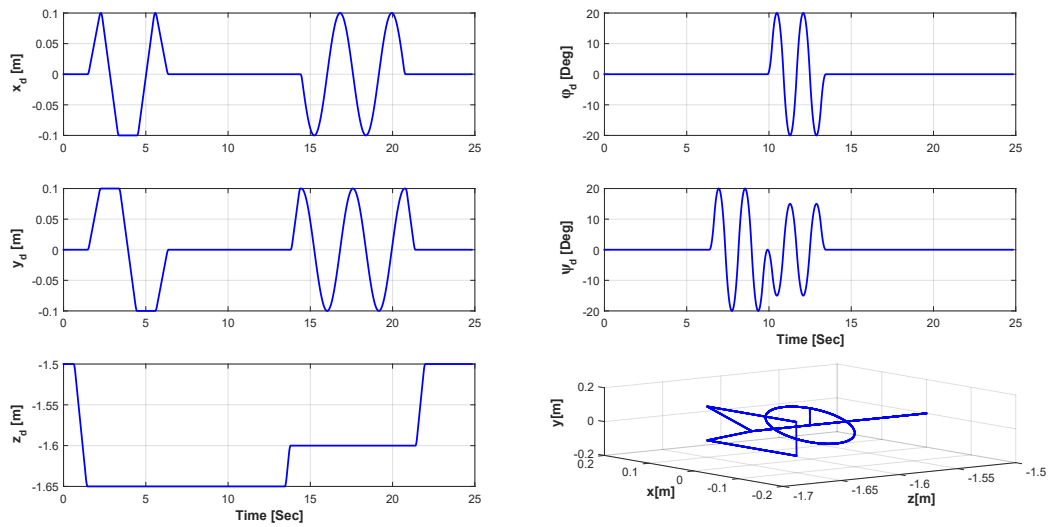


Figure 5.4: Evolution of the desired trajectories in Cartesian Space versus time



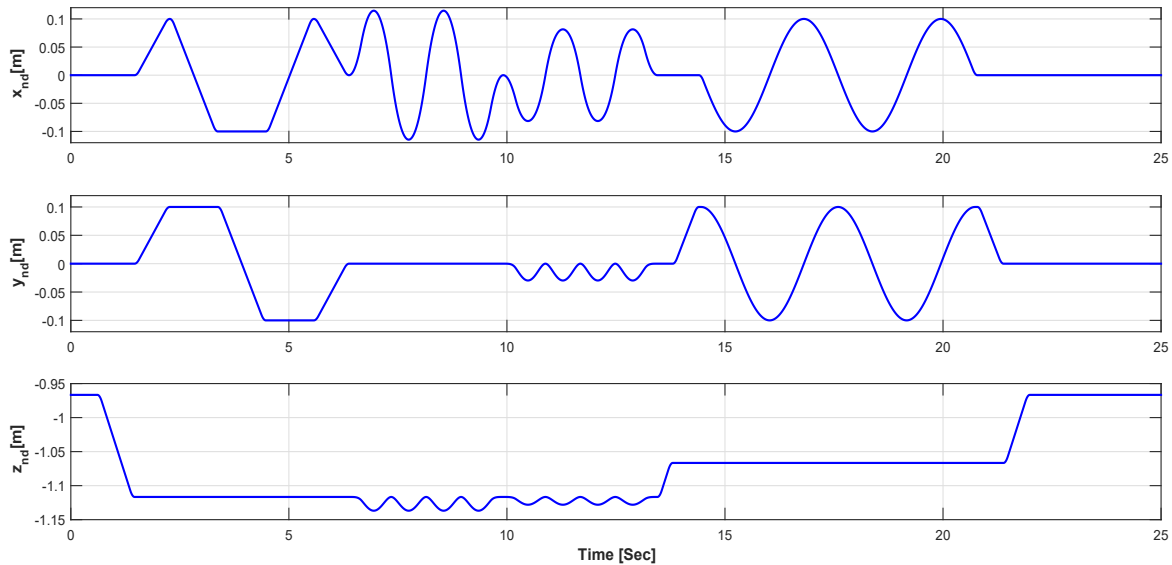


Figure 5.5: Desired trajectory of the nacelle in Cartesian space versus time

### Trajectory for Scenario 2

The trajectory generated for this scenario is used for the validation of the RISE feedforward with adaptive gains. Compared to the previous trajectory, this one is more complex due to a large number of operations it contains to generate different geometric shapes. This trajectory describes a milling machining process on a flat piece of styrofoam with a witness of one inch. The G-code for this milling machining task was firstly simulated in WinUnisoft software from Alecop for a safety implementation on SPIDER4. The cutting depth established for these experiments is 5mm, where the cutting speed established for the spindle S is 7000 rpm. One can modify the speed S from the G-code, but it is not possible to do it automatically by the established control law. Figure 5.6 depicts the flat piece to be machined with its dimensions. The trajectory generated by G code is shown in three dimensions in Figure 5.7, where the lines in blue describe the part of the trajectory where the robot is not cutting material while the lines in red represent the sections of the trajectory where the spindle is devastating the styrofoam. The Second scenario is executed at different forward speeds in order to see the impact the speed changes have on the controllers' performance. The speeds values are listed below.

- F=1200 mm/min (low speed)
- F=2400 mm/min (medium speed)
- F=1200 mm/min (high speed)

Figure 5.8 represents the desired trajectories in Cartesian space for  $x$ ,  $y$ , and  $z$  with respect to time executed at the specified forward speeds.

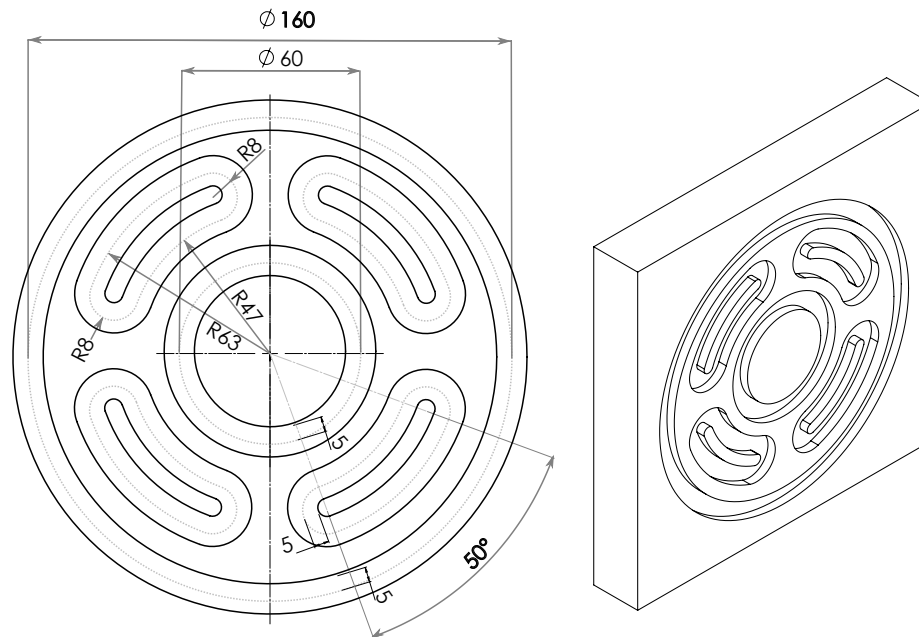


Figure 5.6: Scheme of desired piece with dimensions in mm and degrees

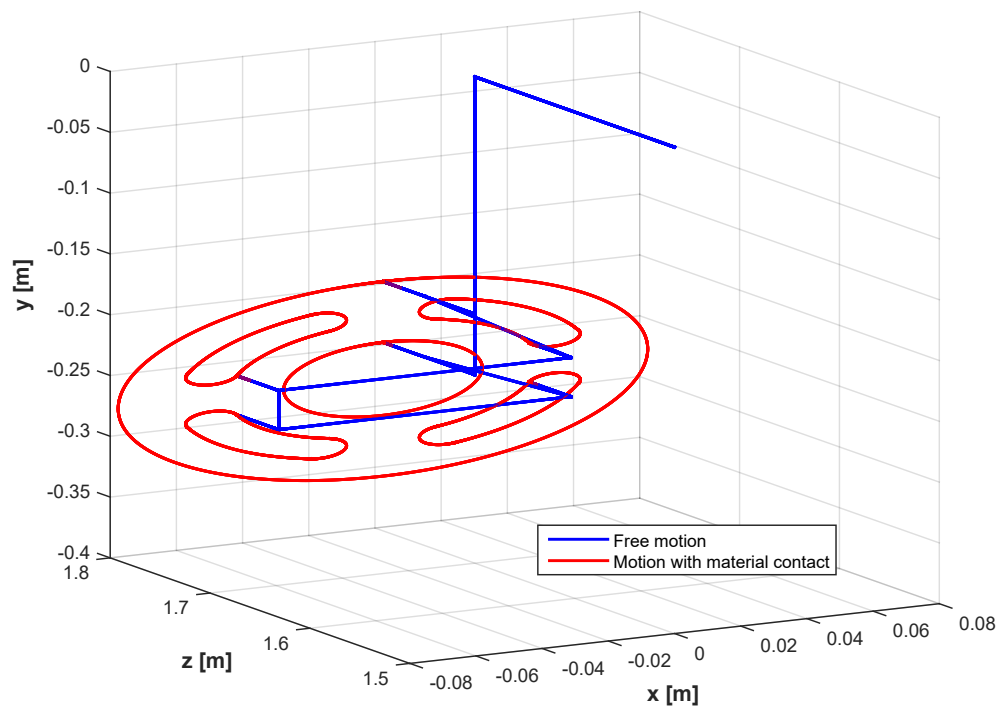


Figure 5.7: Desired trajectory for the machining task

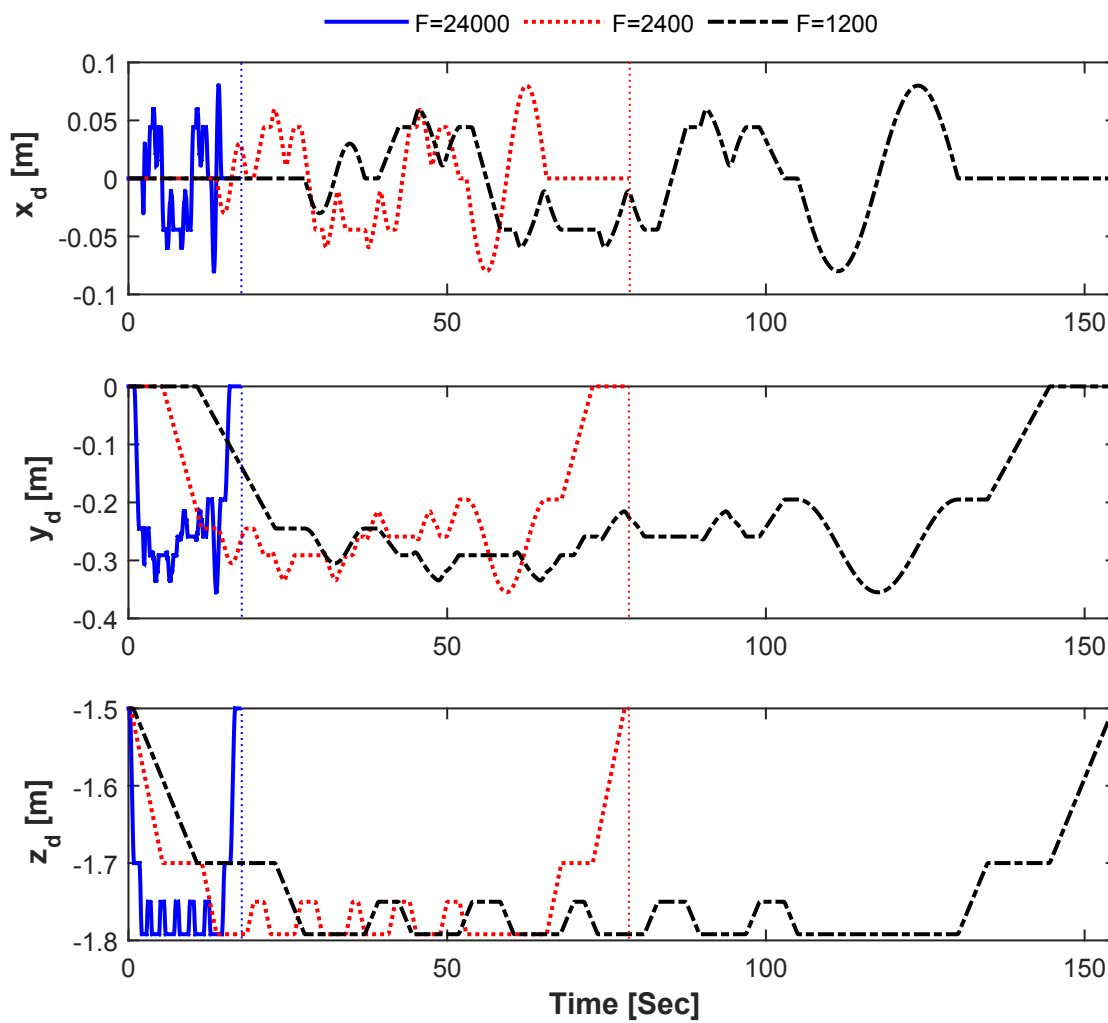


Figure 5.8: Evolution of the desired trajectories versus time for different values of  $F$

### 5.2.3 Performance evaluation criteria

In order to quantify the performance of the proposed control solutions 1 and 2, the Root Mean Square Error (RMSE) formula is used. It enables us to quantify the accuracy obtained by the tested controller in only one numerical data. The RMSEs expressions for SPIDER4 positioning device in Cartesian ( $RMSE_C$ ) and joint space ( $RMSE_J$ ) are given as follows:

$$RMSE_C = \sqrt{\frac{1}{N} \sum_{k=1}^N (e_{xn}^2(k) + e_{yn}^2(k) + e_{zn}^2(k))} \quad (5.1)$$

$$RMSE_J = \sqrt{\frac{1}{N} \sum_{k=1}^N (e_{11}^2(k) + e_{12}^2(k) + e_{13}^2(k) + e_{14}^2(k))} \quad (5.2)$$

where  $e_{xn}, e_{yn}, e_{zn}$  denote the Cartesian position tracking errors of the nacelle along the  $x, y, z$  axes, while  $e_{11}, e_{12}, e_{13}, e_{14}$  are the different joint space tracking errors.  $N$  is the number of samples. The following performance indexes will be used exclusively in the evaluation of the control solution 1 under the first scenario. These indexes are: The Integral Time multiplied by the Absolute Error (ITAE), the Integral Time multiplied by the Absolute Squared Error (ITASE), and the Integral of the Absolute Square Error (ISAE) [Rosales-Díaz et al., 2019], whose mathematical equations are given as follows for Cartesian space and joint space respectively.

$$\begin{aligned} ITAE_C &= \int t \cdot |e_{xn}(t) + e_{yn}(t) + e_{zn}(t)| dt \\ ITASE_C &= \int t \cdot |e_{xn}(t) + e_{yn}(t) + e_{zn}(t)|^2 dt \\ IASE_C &= \int |e_{xn}(t) + e_{yn}(t) + e_{zn}(t)|^2 dt \\ ITAE_J &= \int t \cdot |e_{11}(t) + e_{12}(t) + e_{13}(t) + e_{14}(t)| dt \\ ITASE_J &= \int t \cdot |e_{11}(t) + e_{12}(t) + e_{13}(t) + e_{14}(t)|^2 dt \\ IASE_J &= \int |e_{11}(t) + e_{12}(t) + e_{13}(t) + e_{14}(t)|^2 dt \end{aligned} \quad (5.3)$$

### 5.2.4 Tuning Gains procedure

For both controllers tested on SPIDER4, the gains were selected via trial and error method since there are phenomena such as noise, which can be considerably amplified if the gains are not well selected, damaging the system's performance. Unfortunately for real-time experimentation, it is difficult to perform a gain adjustment analytically when there is noise, so the best solution is to do it by trial and error until proper performance is achieved.

#### Tuning gains procedure for control solution 1

For the proposed RISE control with nominal feedforward and the Standard RISE, firstly, we established a value large enough for  $\Lambda_1$  and  $K_s$ , then we set a value of 1 for  $\Lambda_2$  subsequently,

Table 5.2: Controllers Gains

RISE/RISE FW	RISE FW AG
$\Lambda_1 = 110$	$\bar{\Lambda}_1 = 110$
$\Lambda_2 = 10$	$\bar{\Lambda}_2 = 3500$
$K_s = 33$	$\bar{K}_s = 3500$
$\beta = 8$	$\beta = 8$
	$K_2 = 33$
	$K_3 = 1$

that value is progressively increased until obtaining the minimum possible tracking error. If the tracking error starts to increase, we proceed to reduce the value of  $\Lambda_2$ . The value of the gain  $\beta$  increases the robustness of the controller. However, this value must be set very small at the beginning and gradually increase in order to avoid the phenomenon of chattering. Table 5.2 in the first column presents the values of the gains for the Standard RISE and RISE with nominal feedforward.

### Tuning gains procedure for control solution 2

The gains adjustment for this case is similar to the previous one. First, we establish a value large enough for  $\Lambda_1$ , which in this case, is equal to that of scenario 1 because it is the same gain. The gains  $K_2$  and  $K_3$  represent the minimum values that the gains  $K_s(t)$  and  $\Lambda_2(t)$  can take, respectively. The gains  $\bar{K}_s$ , and  $\bar{\Lambda}_2$  are adjusted according to the desired sensibility to changes of the tracking error in joint space in order to increase the output value of  $K_s(t)$  and  $\Lambda_2(t)$ . Finally, the value for  $\beta$  is keeping the same as in control solution 1. The gains values for the RISE feedforward controller with adaptive gains are presented in Table 5.2 in the second column.

## 5.3 Experimental results of control solution 1

The experimental results of the RISE controller with nominal feedforward are compared to standard RISE control in order to see the obtained enhancement when the IDM of the positioning device of SPIDER4 is used as the nominal feedforward term. The different tracking errors in Cartesian and joint space for the Standard RISE and RISE feedforward are depicted in Figures 5.9 and 5.10, respectively. It can be seen that the proposed RISE controller with nominal feedforward significantly reduces the overshoots with respect to the Standard RISE control, and even though both tracking errors for the whole trajectory are small. The tracking errors in the Cartesian space depicted in Figure 5.9 were calculated using the FKM because the controllers are in function of the joint variables since there is no sensor available to measure the nacelle position directly. The generated torques by the RISE feedforward controller are shown in Figure 5.11, where we can note that the steady-state values for the produced torques  $\Gamma_1$  and  $\Gamma_2$  are positive, while the torques  $\Gamma_3$ , and  $\Gamma_4$  are negative due to the orientation of the kinematic chains and the gravity effects which are different for the upper

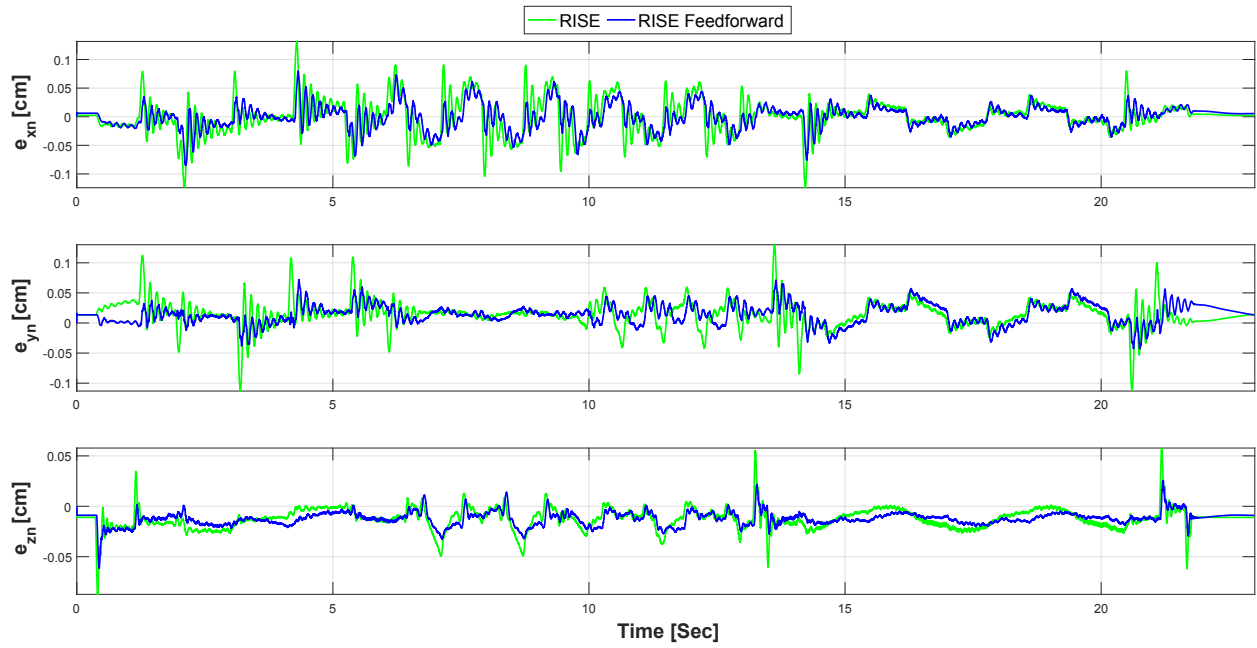


Figure 5.9: Evolution of the tracking errors in Cartesian Space versus time

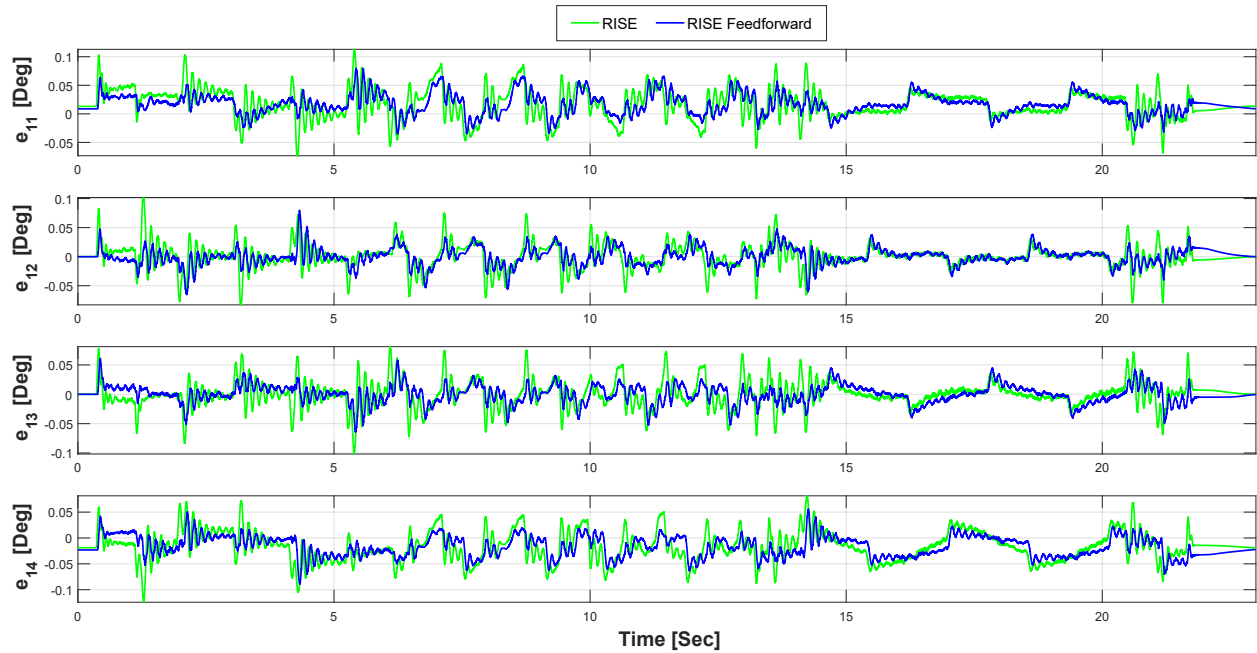


Figure 5.10: Evolution of the tracking errors in joint space versus time

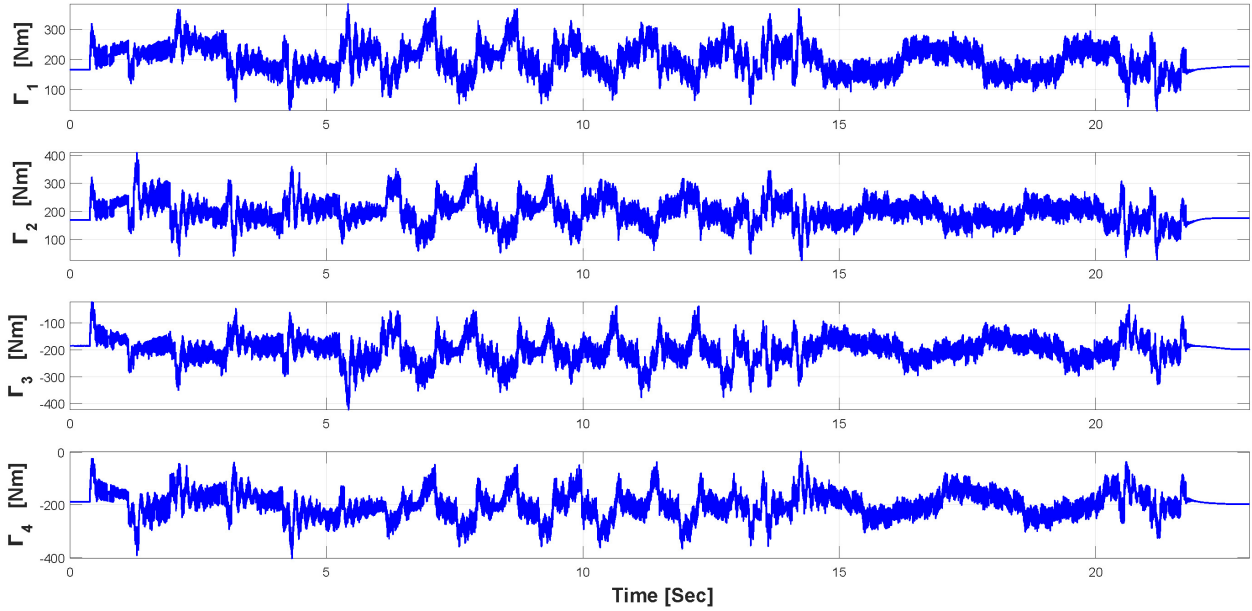


Figure 5.11: Evolution of the torques generated by RISE controller with nominal feedforward versus time

Table 5.3: Controllers Performance Evaluation

	$RMSE_C$ (cm)	Improvement Cartesian	$RMSE_J$ (Deg)	Improvement Joint
<b>RISE</b>	0.0432	0 %	0.0552	0 %
<b>RISE Feedforward</b>	0.0330	23.51 %	0.0420	23.95 %

and the lower chains. From Figure 5.11 also, we can note the presence of noise in the control signals. The noise phenomena came from the fact that measured encoder signals yield noise even if it is not very noticeable. Thereby, when these signals are processed in the control loop, the numerical derivative operations amplify the effect of the noise, resulting in noisy control signals. We did not use a filter at the output of the controller since it may affect the robustness of the controller, in addition to that the computational cost may also increase without significant improvement in the performance of the overall system. The computed RMSEs along the desired trajectories for the nacelle presented in Figure 5.5 are given in Table 5.3. The Table also presents the percentage of improvement of the control solution 1 with respect to the Standard RISE controller. As shown in Table 5.3, the RISE feedforward improvement with respect to the standard RISE controller is up to 23%. To reinforce the superiority of the proposed RISE control with nominal feedforward with respect to standard RISE, we evaluate the resulting tracking error for both controllers with the performance indexes given in equation (5.3), where the evaluation results are depicted in Table 5.3. The behaviors of each one of the performance indexes are shown in Figure 5.12. It is appreciated in the results for Cartesian space that the RISE feedforward control improves the standard RISE control by 36% considering  $ITAE_C$ , 62% if  $ITASE_C$  is taken into account, and 59%

Table 5.4: Controllers performance Evaluation Using Performance Indexes

Tracking Error	Performance Index	RISE	RISE Feedforward
Cartesian Space	$ITAE_C$	7.899	6.555
	$ITSAE_C$	0.418	0.227
	$IASE_C$	0.041	0.020
Joint Space	$ITAE_J$	8.048	5.076
	$ITSAE_J$	0.507	0.189
	$IASE_J$	0.047	0.019

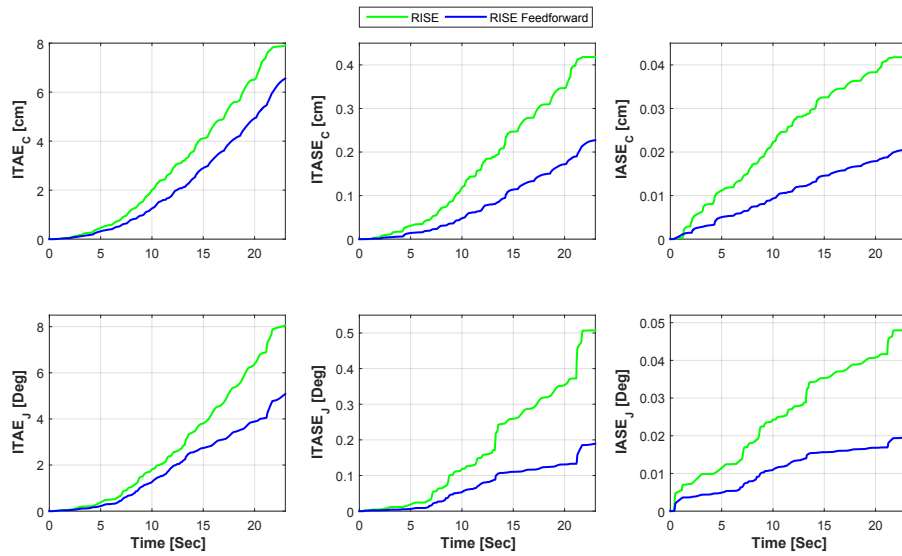


Figure 5.12: Evolution of the different performance indexes for Cartesian and joint space with respect to time

using  $IASE_C$ . The acquired enhancements in joint space are 17%, 62%, and 51% involving  $ITAE_J$ ,  $ITSAE_J$ , and  $IASE_J$ , respectively. The obtained results using these criteria confirm that the proposed RISE feedforward controller far exceeds the standard RISE controller as a result of the integration of our proposed IDM as a feedforward part in the control scheme together with the RISE feedback controller.

## 5.4 Experimental results of control solution 2

To present the experimental results of control solution 2, Standard RISE and RISE with nominal feedforward controllers are also tested with the same trajectory and execution speeds to view in more detail the impact of the RISE feedforward with adaptive feedback gains in the real-time experiments. The three controllers' results are compared with each other in graphs and performance tables to see their effectiveness in a machining task and with different forward speed changes. The name of the proposed controllers is simplified for presenting the



experimental results in graphs and tables as follows: RISE FW represents RISE control with nominal feedforward, and RISE FW AG is RISE feedforward with adaptive gains.

### 5.4.1 Trajectory evaluation at low speed

The comparison of the tracking errors in joint space with a forward speed  $F = 1200 \text{ mm/min}$  are depicted in Figure 5.13. One can see from the graph for motor 1 that the tracking error in parts is smaller with the RISE controller and in other parts with the RISE FW AG controller. For the graphs of motors 2 and 3, the tracking error is quite similar between the RISE FW and RISE FW AG controllers; however, several overshoots are observed in the standard RISE controller's tracking error. In the graph for motor 4, it can be seen quite well that the tracking error of RISE FW AG is the closest to zero, then that of RISE FW and finally the standard RISE. The variations observed in the behavior of the tracking errors of each of the motors may be due to the fact that although the dynamic parameters of each set of links were considered the same, in reality, there are slight variations in mass between them. Figure 5.14 is a zoom of the previous figure in the interval of 75 and 80 seconds to improve the appreciation of the different curves for  $F = 1200 \text{ mm/min}$ . The evolution of the adaptive gains  $\mathbf{K}_s$  and  $\mathbf{\Lambda}_2$  are shown in Figures 5.15 and 5.16, respectively. It can be seen that the minimum values of these gains are those previously established for  $\mathbf{K}_2$  and  $\mathbf{K}_3$ . The maximum peak values that  $\mathbf{K}_s$  and  $\mathbf{\Lambda}_2$  reach are 36 and 4, respectively. The torques generated from the RISE FW AG controller are illustrated in Figure 5.17. It can be seen that the values for motors 1 and 2 range between 100 and 30 Nm, while for motors 3 and 4, the values range from -100 to -350 Nm. These variations are due to the horizontal orientation of the SPIDER4 kinematic chains since gravity's acceleration affects the lower linkages (kinematic chains 1 and 2) differently than the upper linkages (kinematic chains 3 and 4). To give a concrete result of the controllers' performance under this operating condition, one must resort to the use of the RMSEs equations (5.1) and (5.2) such results are presented in Tables 5.5 and 5.6 for joint and Cartesian space, respectively. With the data provided in such Tables, one can compute the resulting improvement of the RISE FW AG with respect to RISE and RISE FW where the results are registered in Table 5.7. We can see that the improvement obtained from the RISE FW AG controller is 24% in the joint space, and 19% in the Cartesian space compare it to the standard RISE controller. Comparing RISE FW AG with respect to RISE FW, we get improvements of 17% and 15% expressed for joint and Cartesian space. Considering these results, we can note a significant enhancement in the system performance when RISE FW AG is used to regulate the motion of SPIDER4. The resulting machining piece of this experimentation using control solution 2 is presented in Figure 5.28a.

### 5.4.2 Trajectory evaluation at medium speed

Increasing the Forward speed  $F$  from 1200 to 2400 mm/min, the following results are obtained. In the graphs presented in Figure 5.18, it can be seen in the four graphs that the tracking error of the RISE FW AG controller is the one that remains closest to zero in most of the execution of the trajectory, then is the tracking error obtained by the RISE FW. However, the

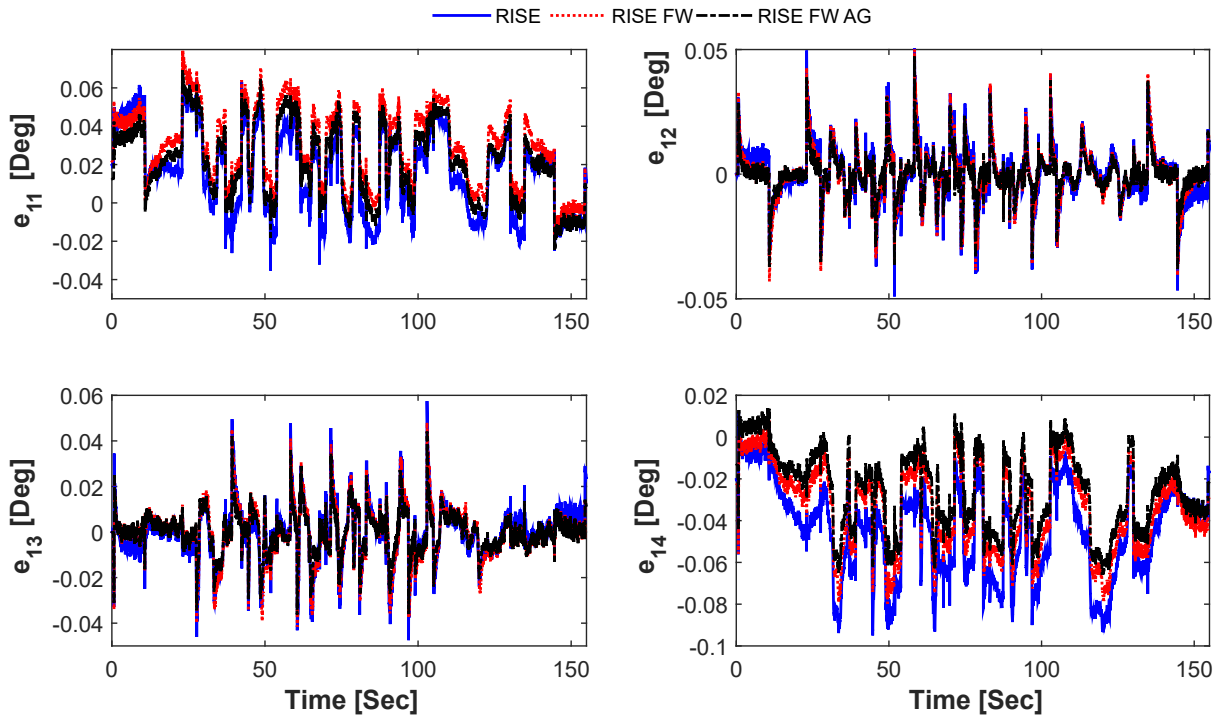


Figure 5.13: Evolution of tracking errors in joint space respect to time for  $F=1200$  mm/min

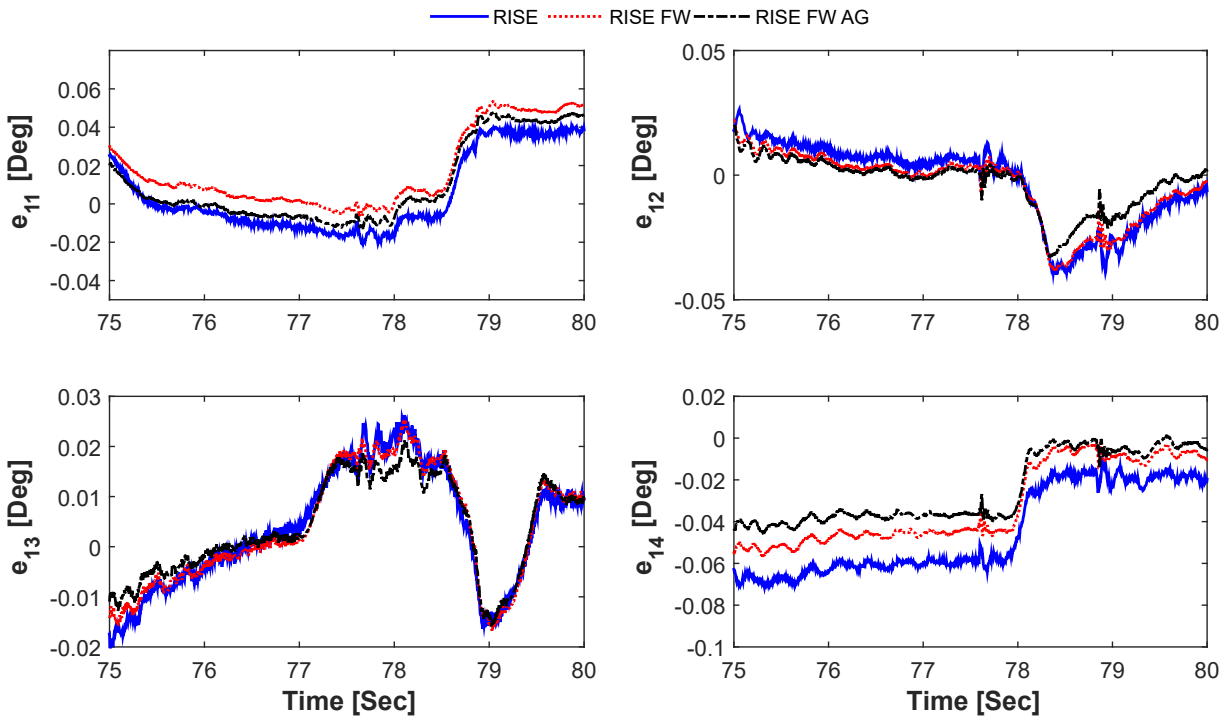
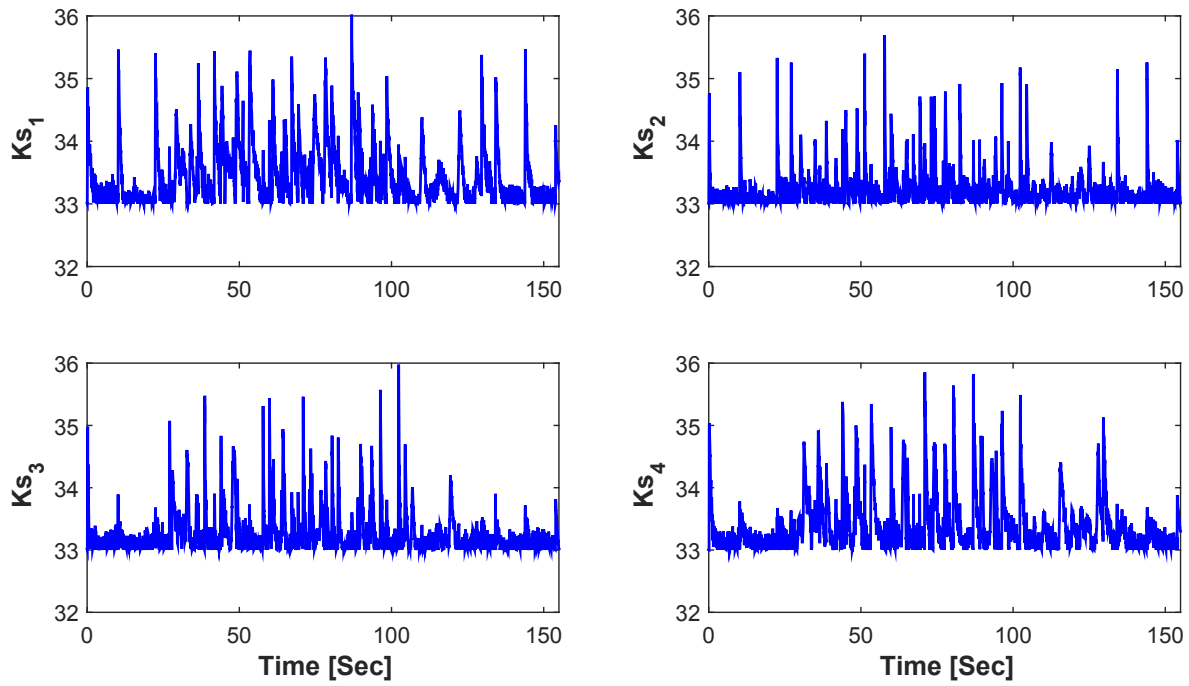
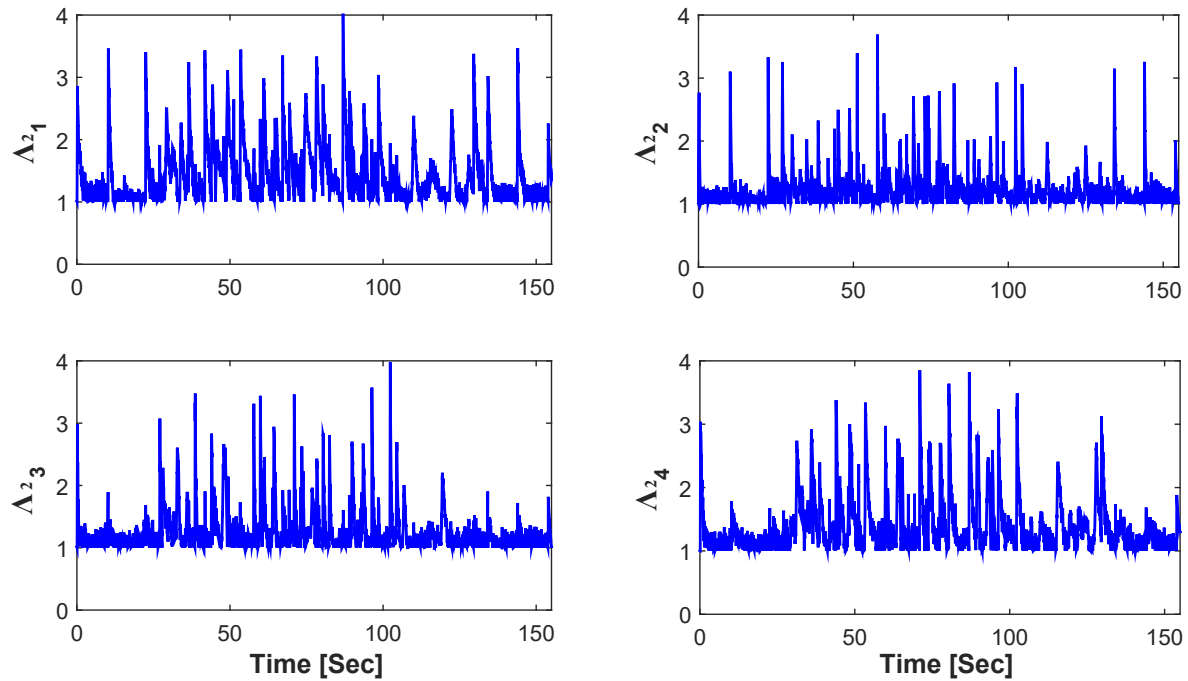


Figure 5.14: Evolution of tracking errors in joint space respect to time for  $F=1200$  mm/min zoomed view

Figure 5.15: Evolution of  $K_s$  gains respect to time for  $F=1200$  mm/minFigure 5.16: Evolution of  $\Lambda$  gains respect to time for  $F=1200$  mm/min

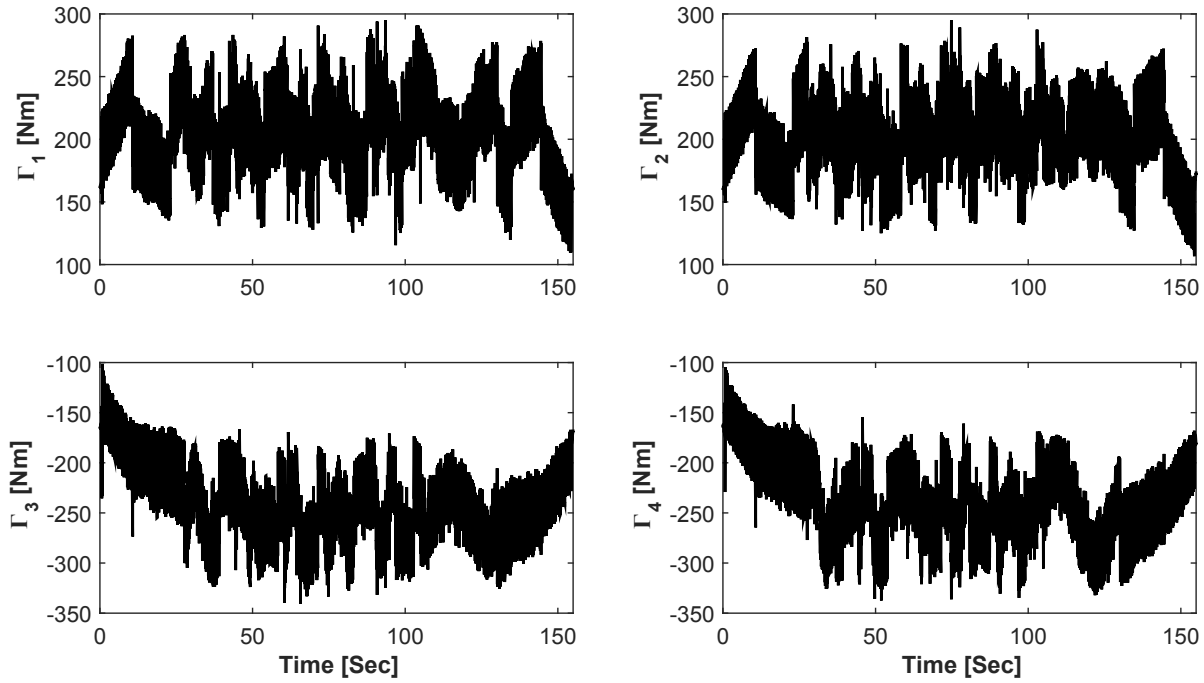


Figure 5.17: Evolution of the control inputs generated by RISE FF AG controller respect to time for  $F=1200$  mm/min

performance of the standard RISE controller was clearly impaired by the increased forward speed. This affirmation can be better stated in Figure 5.19, where a zoom has been applied between 35 and 40 seconds to have a better appreciation of the curves. Regarding the adaptive gains, it can be seen in Figure 5.20 for  $\mathbf{K}_s$  that its maximum peak values increased very little compared to the previous case, it is seen that in some cases it almost reaches values of 37. Moreover, looking at Figure 5.21, one can note that the behavior of  $\mathbf{\Lambda}_2$  also is similar to the previous case whose maximum values it reaches reach up to 5. The control signals are presented in Figure 5.22, where it can be seen that when the forward speed increased, the output torque also increased. We can see that the peak values of motors 1 and 2 are close to 400 Nm, whereas motors 3 and 4 are close to -400 Nm. Making use of the RMSEs formulas, we can confirm the superiority of RISE FW AG in this operative condition concerning the other two controllers. The RMSEs results are presented in the third column of Tables 5.5 and 5.6 for Cartesian and joint space, respectively. The obtained improvements with respect to Standard RISE are 21% in the task space and 25 % in joint space. Comparing to RISE FW, the acquired enhancements are 16% and 18% for Cartesian and joint space. Although the absolute value of the three controllers' tracking errors increase as the forward speed increased, the tracking error of the RISE FW AG controller was hardly affected compared to the other two. Hence the percentage of improvements in control solution 2 has increased considerably compared to the previous case. The machining result of the RISE FW AG controller experimentation at this forward speed is shown in Figure 5.28b. It is appreciated

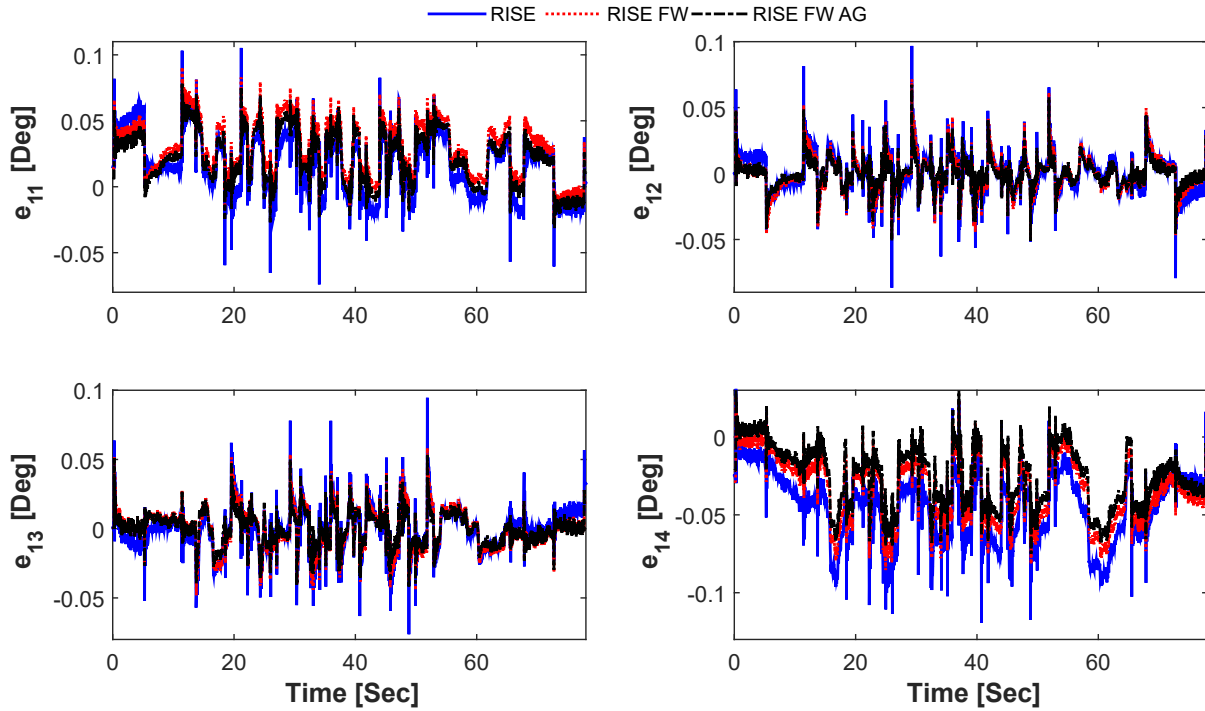


Figure 5.18: Evolution of tracking errors in joint space respect to time for  $F=2400$  mm/min

that the machining quality was low due to the fact that the spindle was positioned for less time in the material, and its turning speed did not increase as the forward speed increased.

### 5.4.3 Trajectory evaluation at high speed

The forward speed used for this test is really not suitable for machining tasks because the quality of the resulting product is quite poor, as we can see in Figure 5.28. Nevertheless, this experiment was proposed to see how the performance of the controllers is affected at such a high forward speed. The evolution of the tracking errors in joint space is presented in Figure 5.23. In comparison to the graphs of the evolution of the tracking errors shown for the previous cases, for this case, it can be seen that the performance of the standard RISE controller is seriously impaired, whereas that the performance of the other two controllers is quite similar. Considering the zoomed view of the described tracking errors in Figure 5.24, it can be distinguished as a slight improvement of the RISE FW AG over the RISE FW controller. So it can be intuited that the improvement of the control solution 2 is not much under this operating condition. Nevertheless, It can be seen in Figures 5.25 and 5.26 that the maximum values of the adaptive gains increased considerably. One can see that  $\mathbf{K}s$  almost reaches values of 45, and some values of  $\mathbf{\Lambda}_2$  are greater than 10. In Figure 5.27, it can be seen that the value of the output torques increased a lot compared to the previous experiments. For motors 1 and 2, these values range from -200 to 600Nm, and for motors 3 and 4, the peak values range from -600 to 200 Nm. As in the previous cases, the RMSEs for this

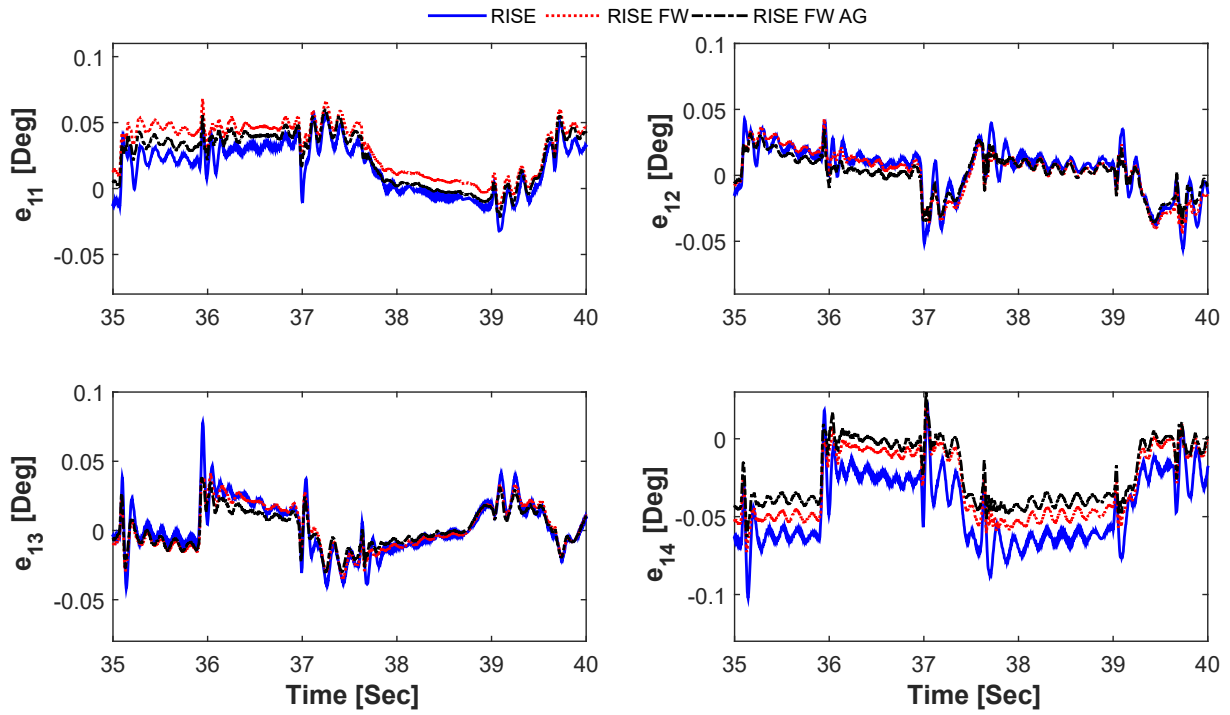


Figure 5.19: Evolution of tracking errors in joint space respect to time for  $F=2400$  mm/min zoomed view

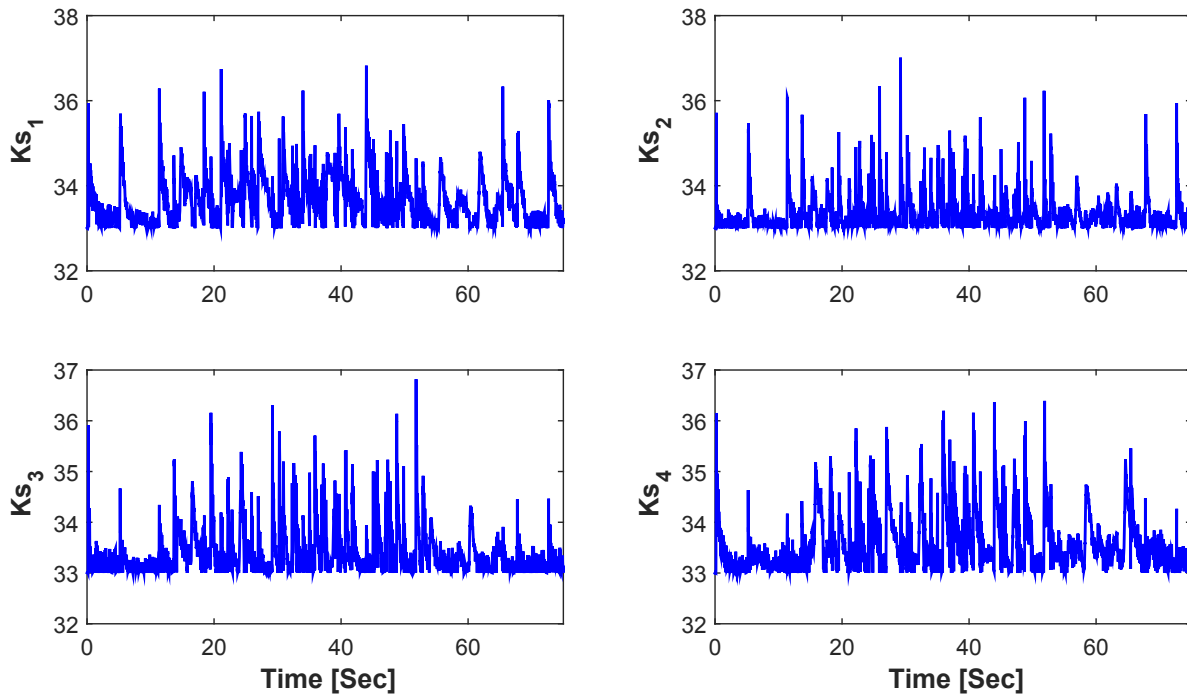
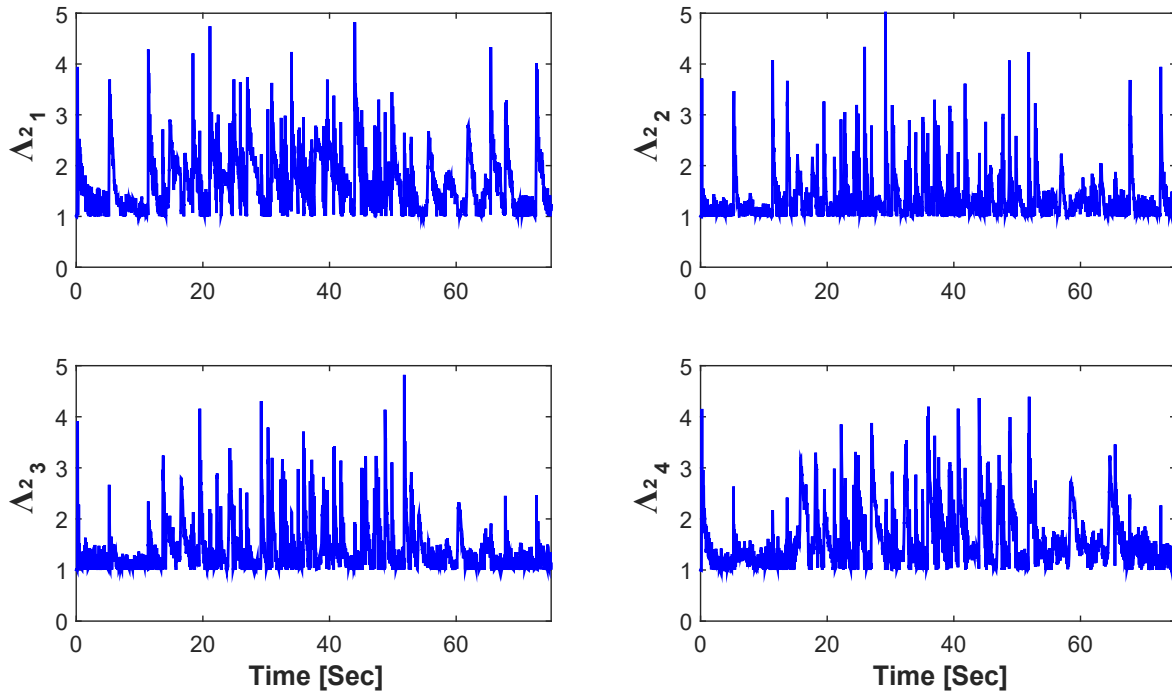
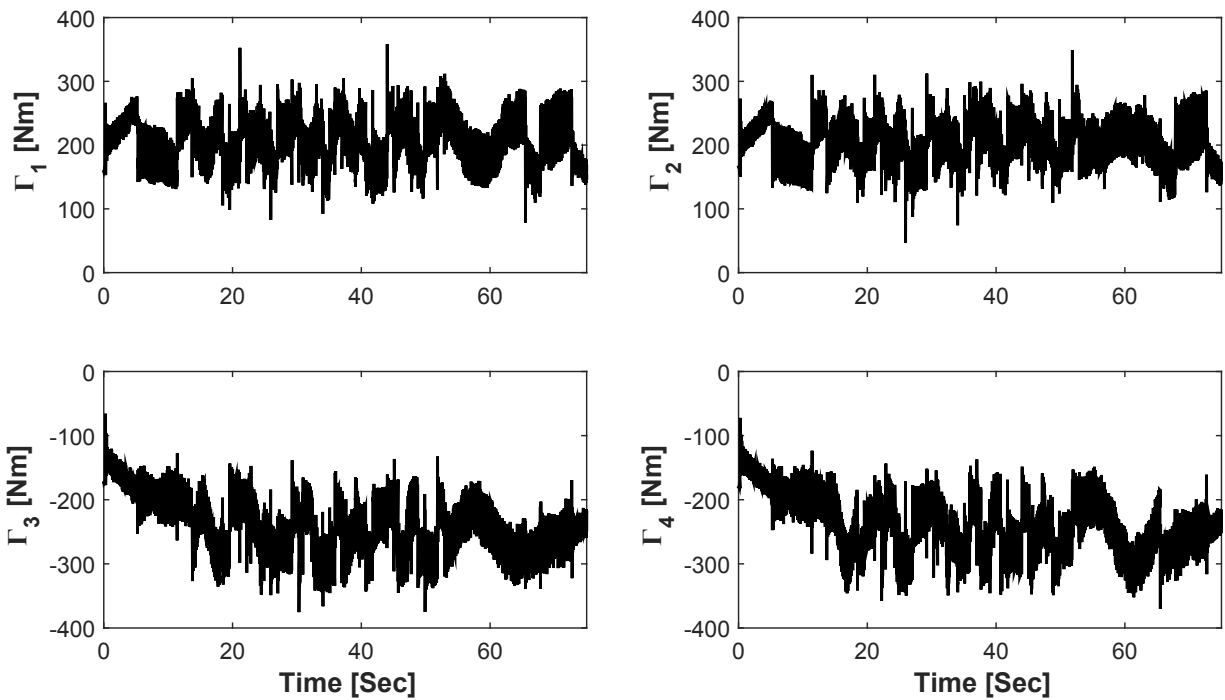


Figure 5.20: Evolution of  $K_s$  gains respect to time for  $F=2400$  mm/min

Figure 5.21: Evolution of  $\Lambda$  gains respect to time for  $F=2400$  mm/minFigure 5.22: Evolution of the control inputs generated by RISE FF AG controller respect to time for  $F=2400$  mm/min

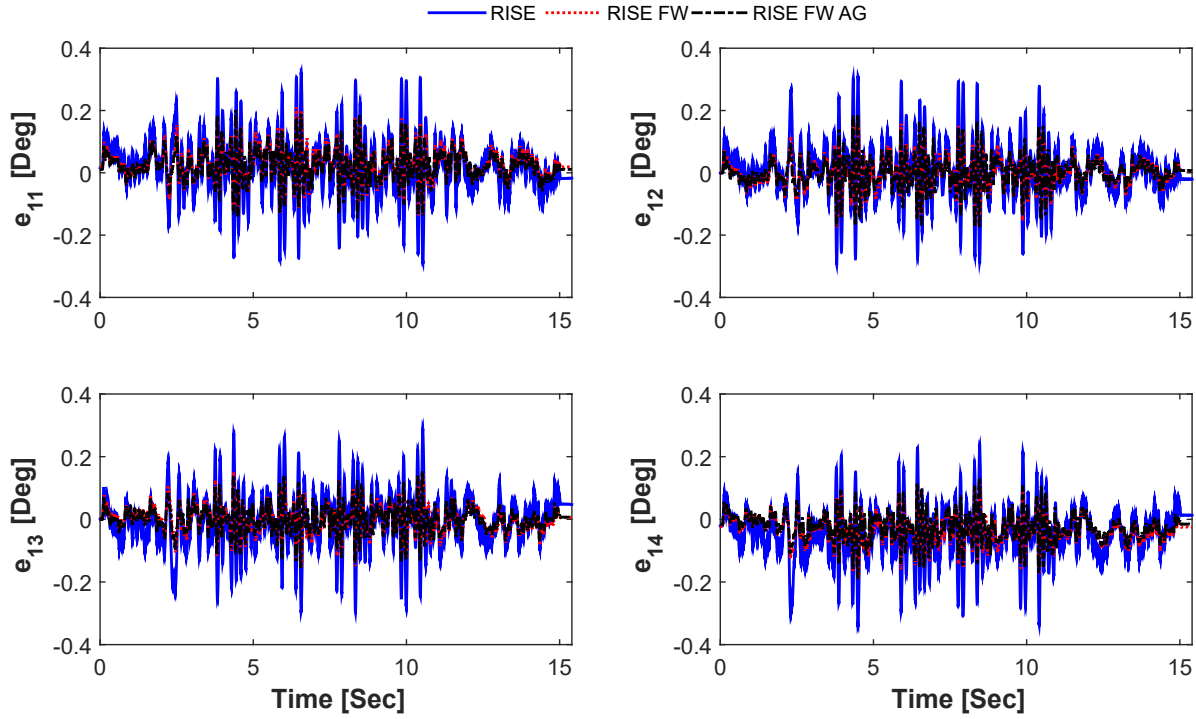


Figure 5.23: Evolution of tracking errors in joint space respect to time for  $F=24000$  mm/min

Table 5.5: Controllers performance evaluation using RMSE for joint space tracking errors (Deg)

Controller	Forward speed (F)		
	1200 mm/min	2400 mm/min	24000 mm/min
RISE	0.0592	0.0618	0.1796
RISE FF	0.0546	0.0566	0.1051
RISE FF AG	0.0449	0.0461	0.1007

experimentation are registered the Tables 5.5, and 5.6 and the corresponding improvements are in Table 5.7. With  $F=24000$  mm/min, the improvement with respect to the standard RISE is 45 %. However, with respect to the RISE FW controller, the improvement was minimal, only 1.47%, which implies that at high speeds, the performance of the RISE FW AG controller is practically the same as the RISE FW.

## 5.5 Conclusion

In this chapter, the experimental results related to SPIDER4 RA-PKM were presented. Firstly, a general overview of the experimental setup was detailed, including a brief description of CNC commands for the trajectory generation on SPIDER4. Two control solutions were proved



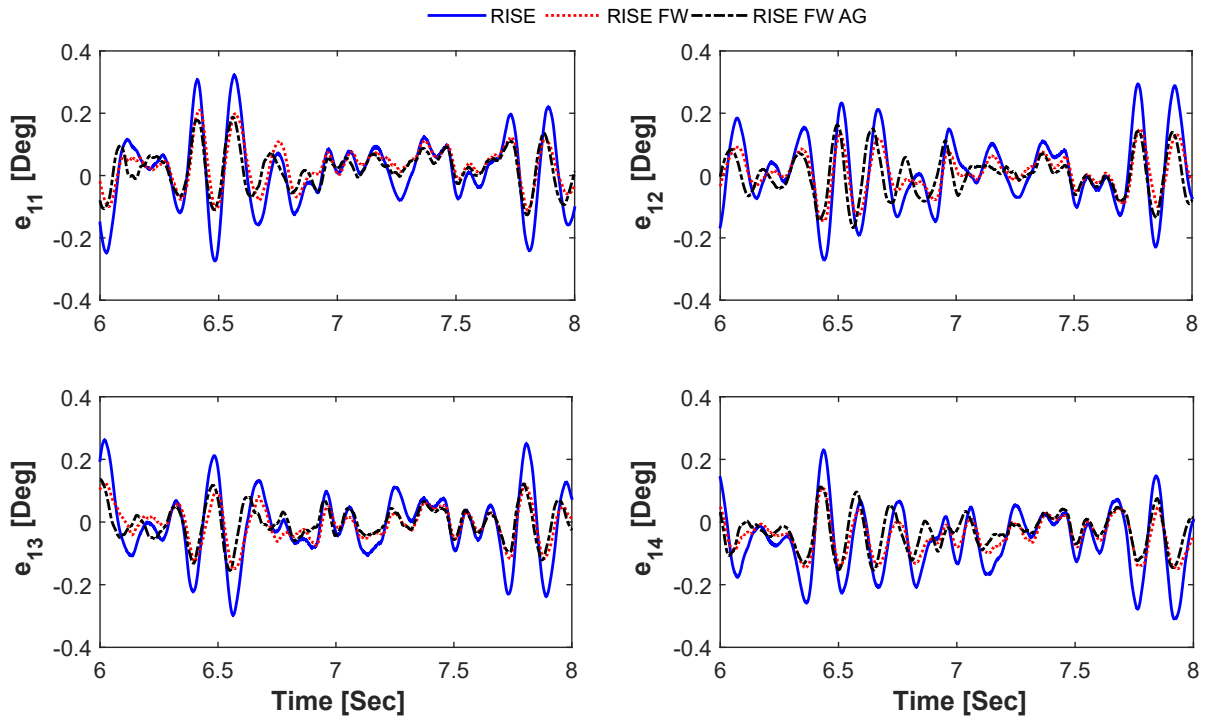


Figure 5.24: Evolution of tracking errors in joint space respect to time for  $F=24000$  mm/min zoomed view

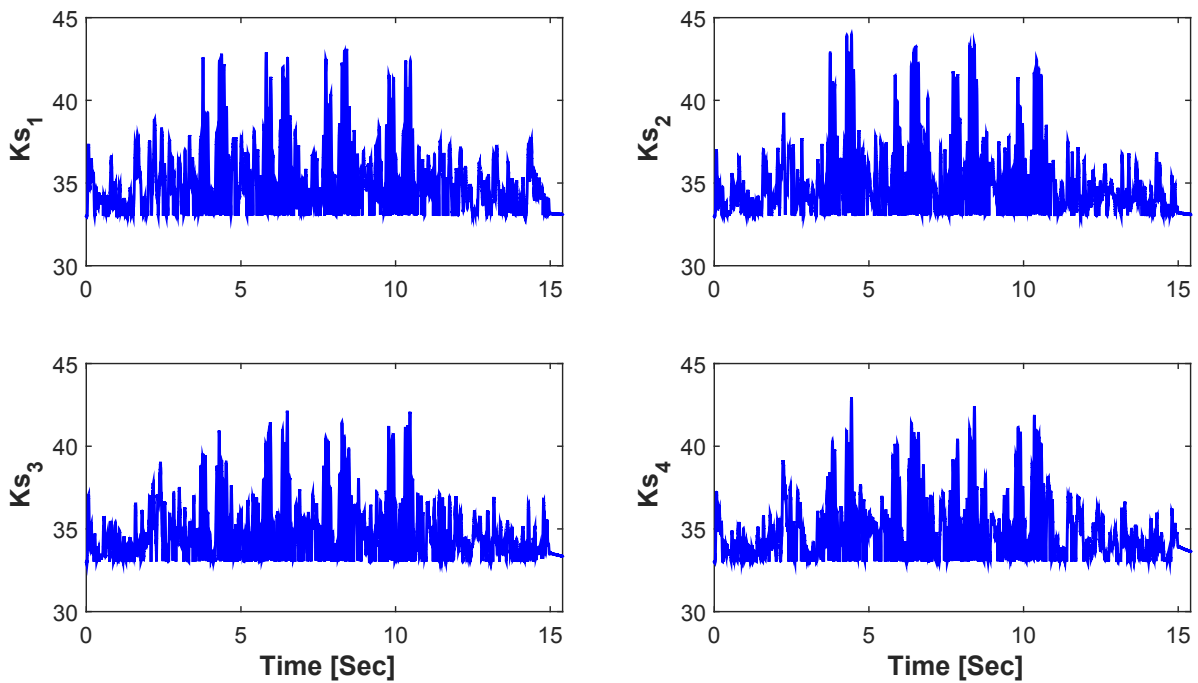


Figure 5.25: Evolution of  $K_s$  gains respect to time for  $F=24000$  mm/min

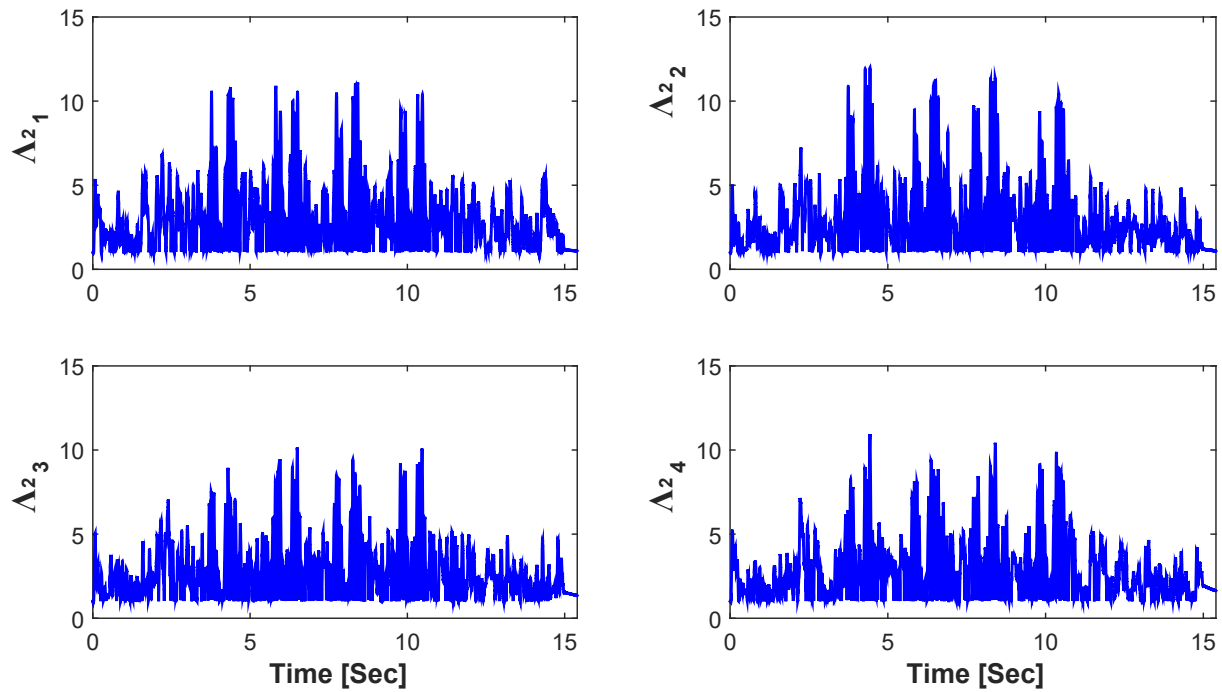


Figure 5.26: Evolution of  $\Lambda$  gains respect to time for  $F=24000$  mm/min

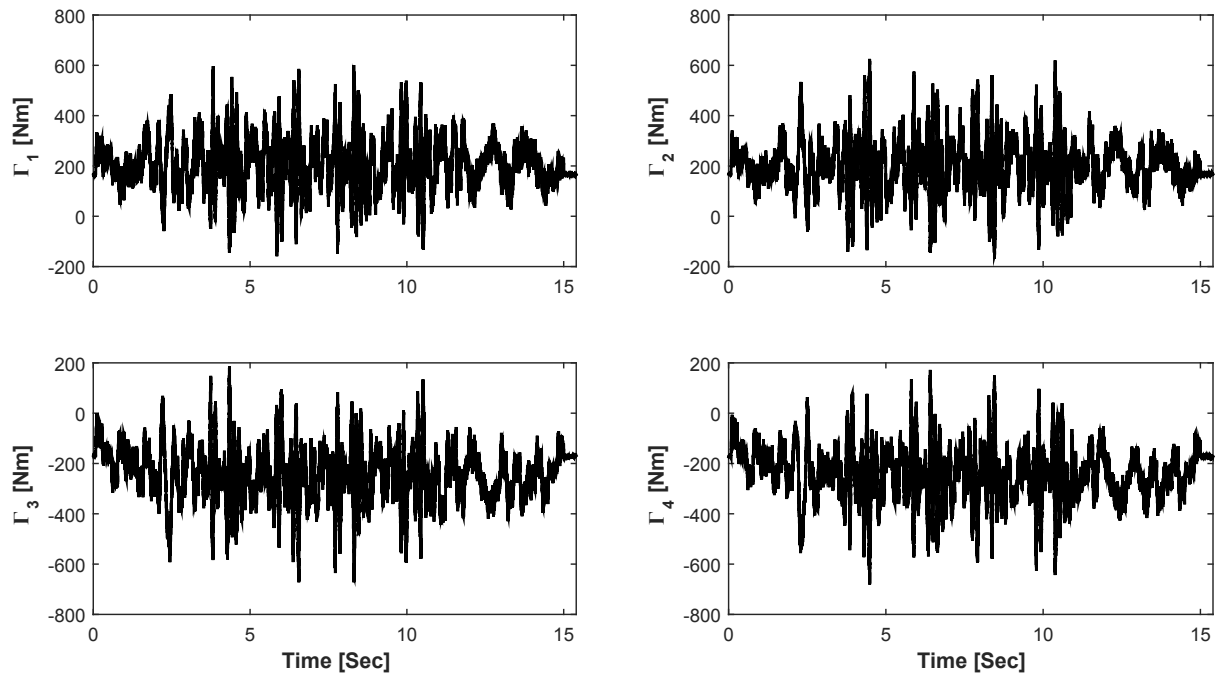


Figure 5.27: Evolution of the control inputs generated by RISE FF AG controller respect to time for  $F=24000$  mm/min

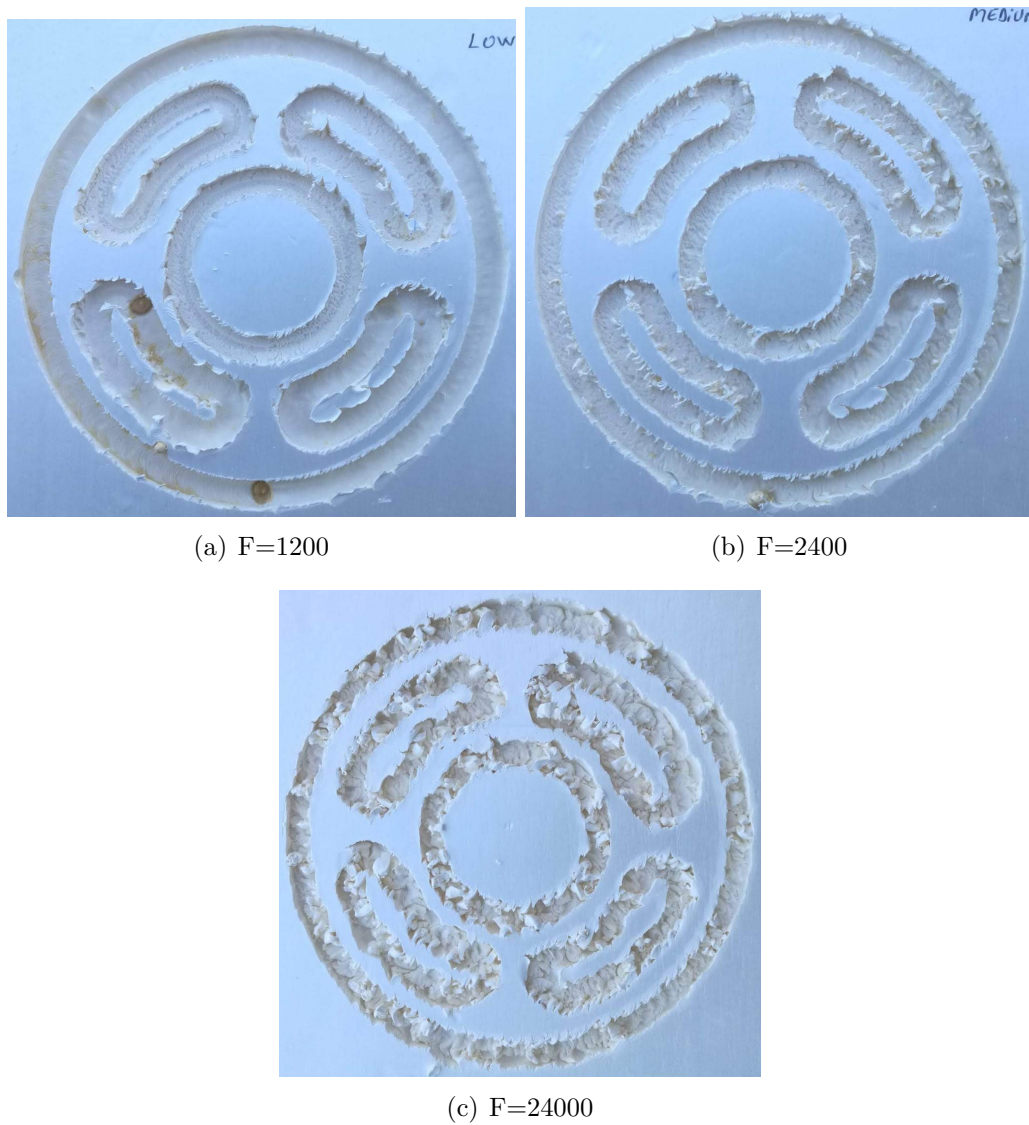


Figure 5.28: Machining results using RISE FF AF with different forward speeds

Table 5.6: Controllers performance evaluation using RMSE for Cartesian space tracking errors (cm)

Controller	Forward speed (F)		
	1200 mm/min	2400 mm/min	24000 mm/min
RISE	0.0383	0.0405	0.1425
RISE FW	0.0367	0.0380	0.0794
RISE FW AG	0.0309	0.0319	0.0782

Table 5.7: Enhancement of the proposed RISE FW AG respect to standard RISE and RISE FW using the RMSE for tracking errors in Cartesian and joint space

<b>Forward speed</b>	<b>Cartesian RISE</b>	<b>Space RISE FW</b>	<b>Joint RISE</b>	<b>Space RISE FW</b>
Low	19.27 %	15.73 %	24.155 %	17.75 %
Medium	21.36 %	16.06 %	25.40 %	18.55 %
High	45.12 %	1.47 %	43.93 %	4.18 %

on SPIDER4 under two scenarios. The first controller tested on SPIDER4 was a RISE control with a nominal feedforward term. This controller aimed mainly to validate the veracity of the computed IDM of SPIDER4 since the IDM was used in this controller as the feedforward term. A free-motion trajectory was used in this experiment, and the obtained performance of the control solution 1 was 23% higher than that of the standard RISE controller. The second controller was a RISE feedforward control with adaptive feedback gains whose purpose is to be an alternative to force control in view that sometimes robotic systems do not have sensors to measure contact forces. Machining is a task where contact forces are involved. Therefore, it is necessary to implement a control scheme to counteract them. The adaptive gains are intended when the error signal increases due to contact with the material; These will increase their value to compensate for this error. By reaching a remarkably small error, the gains will adjust their value optimally. The control solution 2 was validated to control solution 1 and standard RISE in a machining scenario. This scenario was performed with low, medium, and high forward speed. In all speed executions, the proposed control solution 2 outperformed the RISE feedforward and standard RISE. However, at high speed, the performance of the RISE feedforward with adaptive gains deteriorates considerably, having a behavior similar to RISE feedforward. Notwithstanding, the executed maximum speed in these experiments is not suitable for a real machining task so that the intention of using that speed was to know the limitations of the proposed control schemes precisely. According to the results, we can say that the control solution 2 obtained its best performance at medium speed.

## 6

## Simulation and Experimental Results on Delta PKM

*In this chapter, the experimental and simulation result on a Delta PKM are presented. A classical PID controller is validated via real-time experiments on the Delta PKM prototype of the Polytechnic University of Tuancingo. Moreover, the simulation results concerning the control solution 3 detailed in chapter 4 are provided here. The computed kinematic and dynamic parameters of the Delta prototype were used to carry out the numerical simulations.*

## Contents

<b>6.1</b>	<b>Introduction</b>	<b>125</b>
<b>6.2</b>	<b>Description of the Experimental Setup, and Implementation Issues</b>	<b>126</b>
<b>6.3</b>	<b>Experimental Results with a PID Controller</b>	<b>134</b>
<b>6.4</b>	<b>Simulation results of control solution 3</b>	<b>137</b>
<b>6.5</b>	<b>Conclusion</b>	<b>148</b>

## 6.1 Introduction

The Delta robot is a revolutionary PKM whose integration into the industry has been due to its advantages over other types of manipulators in terms of potential accuracy and high operational speed. For such advantages mentioned in previous chapters about the Delta robot architecture, the Polytechnic University of Tulancingo is currently developing a classic prototype with 3-DOF. The prototype aims to be an experimental platform for the validation of advanced nonlinear control techniques. However, there are still details in the electrical system and the mechanical structure, so the prototype is not entirely ready to experiment with complex control schemes. Until now, it has been possible to implement only a PID controller, so the results that have so far been obtained experimentally will be presented in

this chapter. The control solution 3 was specially designed to be implemented in robots that perform Pick and Place tasks such as the Delta robot. Nevertheless, due to the prototype is not already finished, the RISE controller with adaptive feedforward compensation-based-on B-spline neural networks was tested only via numerical simulations using the Delta's computed kinematic and dynamic parameters prototype. With that said, let us start with the description of the experimental setup of the Delta PKM prototype.

## 6.2 Description of the Experimental Setup, and Implementation Issues

### 6.2.1 Description of experimental Setup

The Delta PKM prototype of the Polytechnic University of Tulancingo is still in the development process. The experimental setup of the prototype is depicted in Figure 6.1. In order to generate the motion, the robot uses three Baldor MTE-3363-BLBCN DC servomotors, where each one can provide 8.5 Nm as maximum peak torque each one. The servomotors include an incremental optical encoder giving 1000 lines per revolution. All the functionalities of the prototype, including kinematic algorithms, control law, a GUI, are contained in a project of LabVIEW of National Instruments where each function of the project is programmed in a Virtual Instrument (VI). The project is transferred to the embedded system cRIO-9014, also form National Instruments. This target is in charge of acquiring and sending the data to control the robot. To this end, cRIO is equipped with the following modules:

- NI 9505, which is a controller for brushed motors.
- NI 9264, this is a module of analog outputs with 16 channels. The output voltage range is  $\pm 10V$  with a 16-bit resolution.
- NI 9403, this is a 5V/TTL digital I/O module with 32 channels.
- NI 9219, this is a universal analog input with four channels.

The N9505 is used as a 5V power supply for the three motor encoders', and the NI 9403 module collects the provided signals by the encoders. Figure 6.2 illustrates the connection diagram with the NI modules for data acquisition. Since the servo motor drivers are not available, a power stage was build using a three Lambda GEN 50-30 power supply to supply each servomotor and three H-bridges' composed by optocouplers and IGBTs to control them. The control scheme sends control signals that go out through the NI 9264 module. These signals are amplified by the Lambda source to enter the servomotors and produce the motion. In order to reverse the rotation of the servomotors, the NI module is used, which, based on certain conditions established in the programming code, it sends a signal to activate the servomotor clockwise or another signal to activate it counterclockwise. Figure 6.3 shows the power stage of one of the servomotors. It is worth to mention that the connection diagram of Figure 6.3 is similar for the other two servomotors but using different connection pins.



Figure 6.1: View of Delta PKM prototype of Polytechnic University of Tulancingo

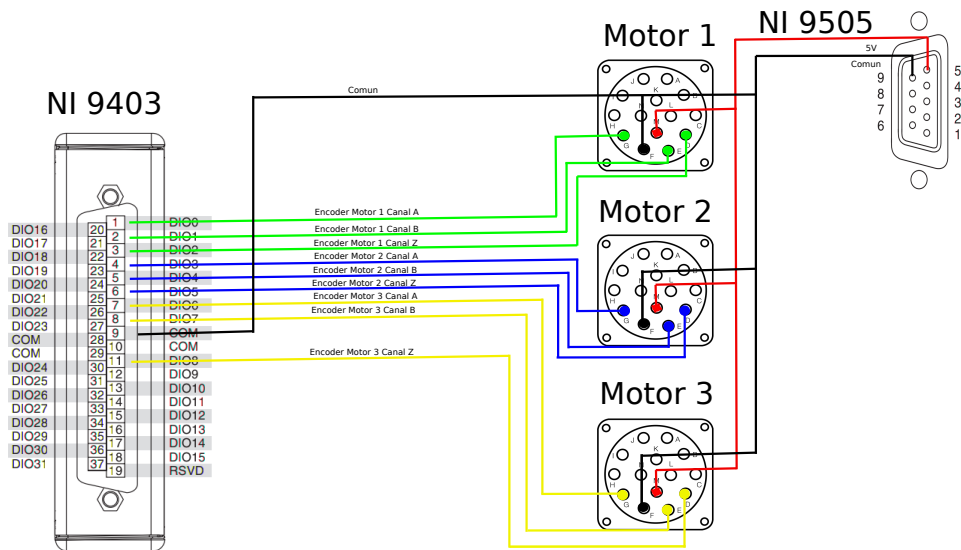


Figure 6.2: Encoders connection diagram

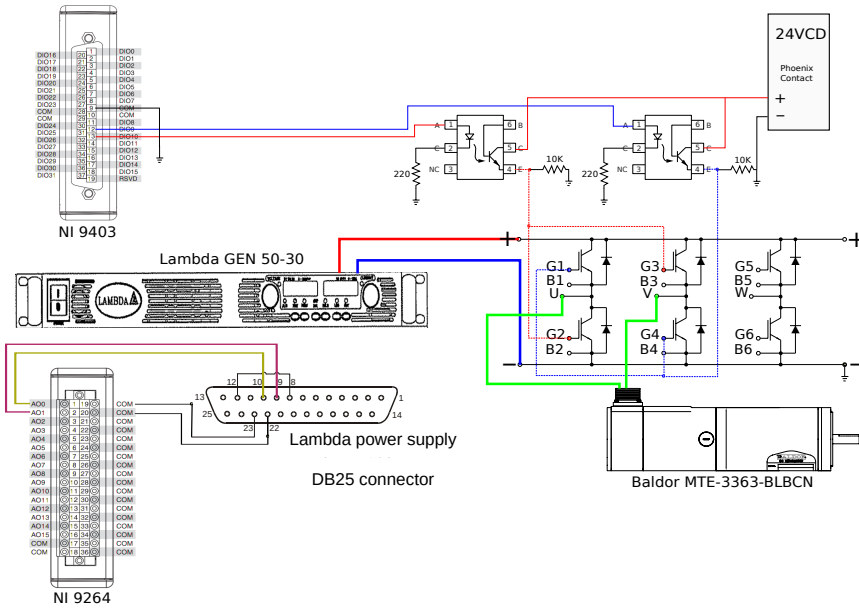


Figure 6.3: Power stage diagram of one servomotor of Delta PKM prototype of Polytechnic University of Tulancingo

## 6.2.2 Trajectory Generation for Real-time Experiments

For Real-time experiments, the algorithm for the Cartesian space trajectories was programmed in a VI of the LabVIEW project. Such trajectories intended to create a path for P&P applications. The desired path was programmed using a fifth-order polynomial algorithm described by the following equations [Natal et al., 2016]:

$$x_f = x_i + r(t)\Delta x, \quad \text{for } 0 \leq t \leq t_f \quad (6.1)$$

$$r(t) = 10 \left( \frac{t}{t_f} \right)^3 - 15 \left( \frac{t}{t_f} \right)^4 + 6 \left( \frac{t}{t_f} \right)^5 \quad (6.2)$$

Where  $x_i$  is the initial position,  $x_f$  is the final position; both are given in Cartesian space,  $r(t)$  is the trajectory function of two points,  $\Delta x = x_f - x_i$ , and  $t_f$  is the duration of each displacement of the trajectories, in this case,  $t_f = 0.5$  sec. Employing (6.1) and (6.2), the following sequence of points expressed in meters are proposed:

1. From (-0.1,-0.1,-0.6) to (-0.1,0,-0.55)
2. From (-0.1,0,-0.55) to (-0.1,0.1,-0.6)
3. From (-0.1,0.1,-0.6) to (0,0.1,-0.55)
4. From (0,0.1,-0.55) to (0.1,0.1,-0.6)
5. From (0.1,0.1,-0.6) to (0.1,0,-0.55)



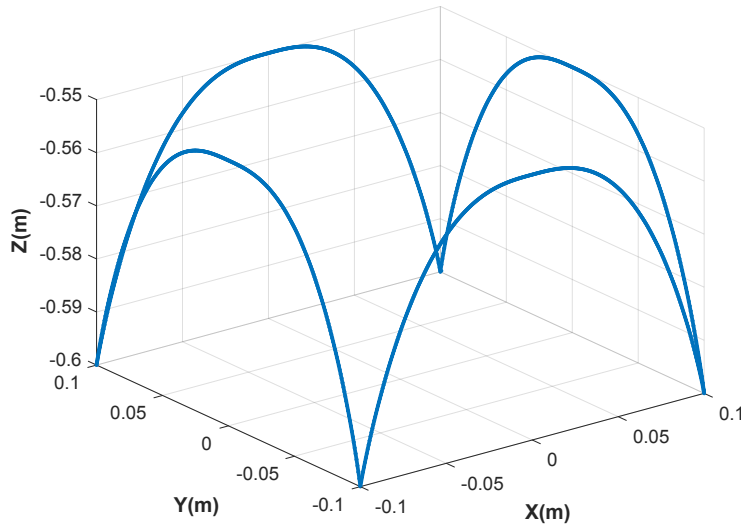


Figure 6.4: Isometric View of trajectory used for Real-time experiments

6. From  $(0.1, 0, -0.55)$  to  $(0.1, -0.1, -0.6)$
7. From  $(0.1, -0.1, -0.6)$  to  $(0, -0.1, -0.55)$
8. From  $(0, -0.1, -0.55)$  to  $(-0.1, -0.1, -0.6)$

The generated trajectory for the previous desired points is presented in Figure 6.4, as long as the evolution of the desired trajectories  $x_d$ ,  $y_d$ , and  $z_d$  with respect to time are depicted in Figure 6.5 where the time units are milliseconds for time and millimeters for Cartesian displacement.

### 6.2.3 Trajectory Generation for Numerical Simulations

The interpolation algorithm for the trajectory generation, in this case, is the same as presented in equations (6.1) and (6.2) to generate another P&P trajectory which is used in the evaluation of the control solution 3. The P&P trajectory is evaluated under two scenarios. Moreover, a third scenario consisting of changes in the execution speed is proposed to see the effectiveness of the control solution 3 under this operative condition. The trajectory for the third scenario is a spiral path whose equations are different, as presented previously.

#### Trajectory for Scenarios 1 and 2

The trajectory proposed for both scenarios corresponds to a P&P path with more positions as presented for real-time experiments. The trajectory is generated using the polynomial interpolation of fifth-order, being the same equations presented in (6.1) and (6.2). The obtained trajectory used for this case study is represented in Cartesian space by Figure 6.6, illustrating the two scenarios. The left illustration represents the tracking trajectory for the

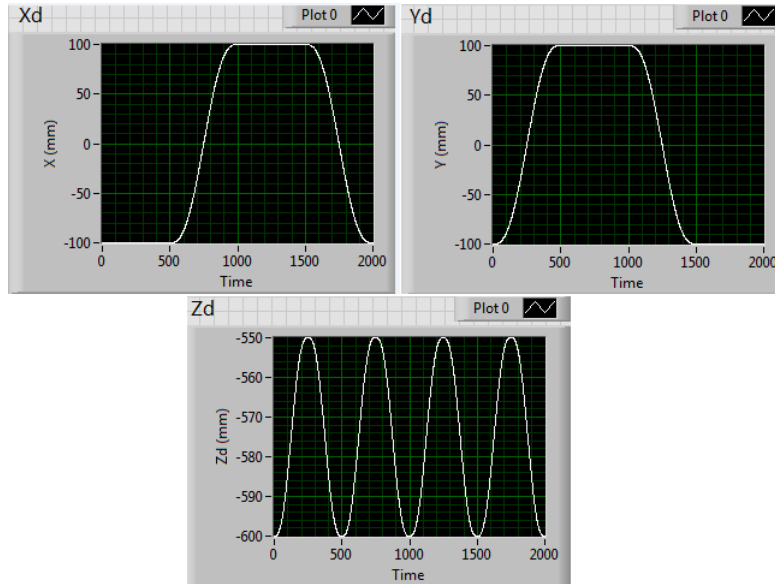
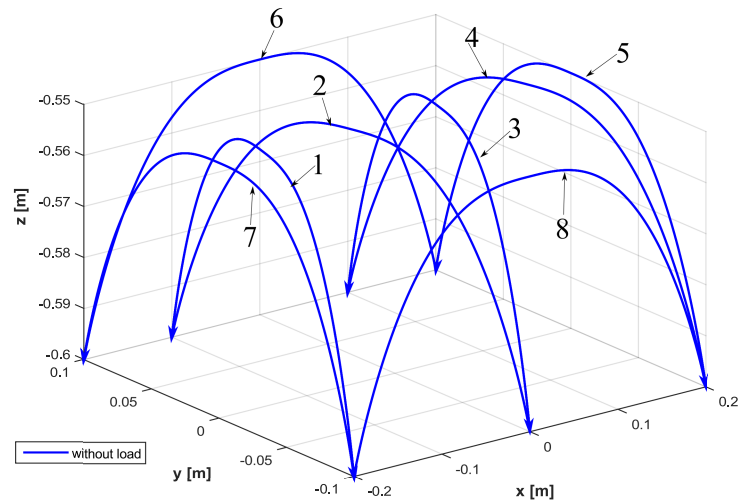


Figure 6.5: Desired trajectories with respect to time generated in LabVIEW

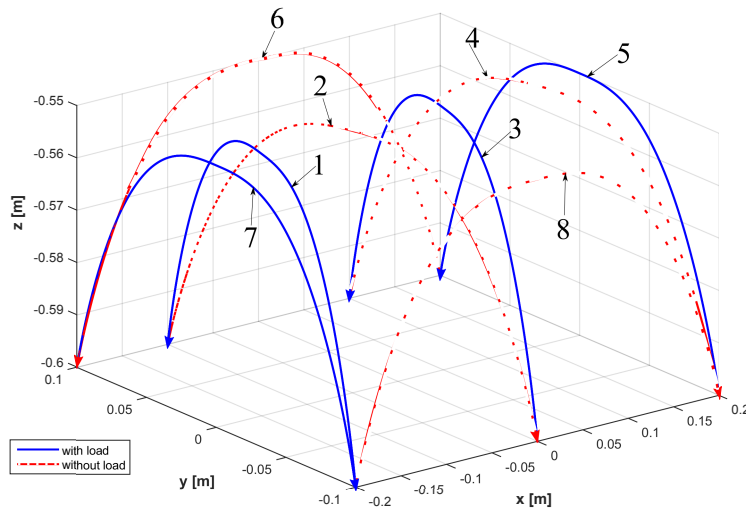
first scenario executed by the Delta PKM without any payload, while in the second scenario, the robot moves masses of 1 Kg along trajectory sections. The sections of the trajectory where the Delta robot's traveling plate moves a mass are depicted with a dotted line in red color, whereas the solid lines in blue are the sections of the trajectory where the Delta PKM is moving without any payload. The evolution of the trajectories with respect to time is given in Figure 6.7. The sequence of movements for this P&P trajectory is presented as follows in the  $(x,y)$  plane.

1. Start-Pick: from  $(-0.2,-0.1)$  to  $(-0.1,0.1)$ .
2. Pick-Place: from  $(-0.1,0.1)$  to  $(0,-0.1)$ .
3. Place-Pick: from  $(0,-0.1)$  to  $(0.1,0.1)$ .
4. Pick-Place: from  $(0.1,0.1)$  to  $(0.2,-0.1)$ .
5. Place-Pick: from  $(0.2,-0.1)$  to  $(0.2,0.1)$ .
6. Pick-Place: from  $(0.2,0.1)$  to  $(-0.2,0.1)$ .
7. Place-Pick: from  $(-0.2,0.1)$  to  $(-0.2,-0.1)$ .
8. Pick-Place: from  $(-0.2,-0.1)$  to  $(0.2,-0.1)$ .

The previous movement sequences are performed in 0.3 seconds for both scenarios.



(a) Desired trajectory in Cartesian space for scenario 1 case study 1



(b) Desired trajectory in Cartesian space for scenario 2 case study 1

Figure 6.6: Desired 3D trajectory for a P&P Task. The lines in red color corresponds to trajectory portions where the Delta PKM is moving with a payload and the blue lines are the corresponding portions without payload

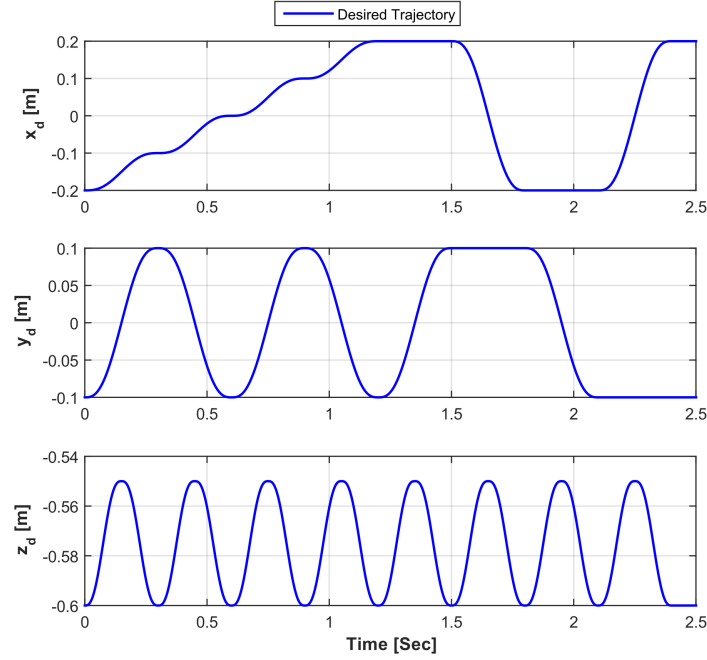


Figure 6.7: Evolution of the desired trajectories in Cartesian space versus time for scenarios 1 and 2

### Trajectory For Scenario 3

The desired trajectory for this case study is a spiral path on the plane  $(x,y)$  (see Figure 6.8). The proposed three scenarios for this case study are subject to changes in the speed execution of the trajectory (low, medium and high). The following equations are used to generate the desired spiral path:

$$\begin{aligned} x_d &= r \cos(2\pi ft) \\ y_d &= r \sin(2\pi ft) \\ z_d &= -0.6 \end{aligned} \tag{6.3}$$

$$r = 0.04ft \tag{6.4}$$

where  $r$  denotes the separation distance between circular turns and  $f$  is the frequency of the circular movements. The speed changes are achieved by modifying the value of  $f$ , we define:

- $f = 0.5Hz$  for low speed
- $f = 1.75Hz$  for medium speed
- $f = 3.5Hz$  for high speed

The initial and final positions of the spiral trajectory given in Cartesian coordinates are  $(0,0,-0.6)$  and  $(0,0.2,-0.6)$ .

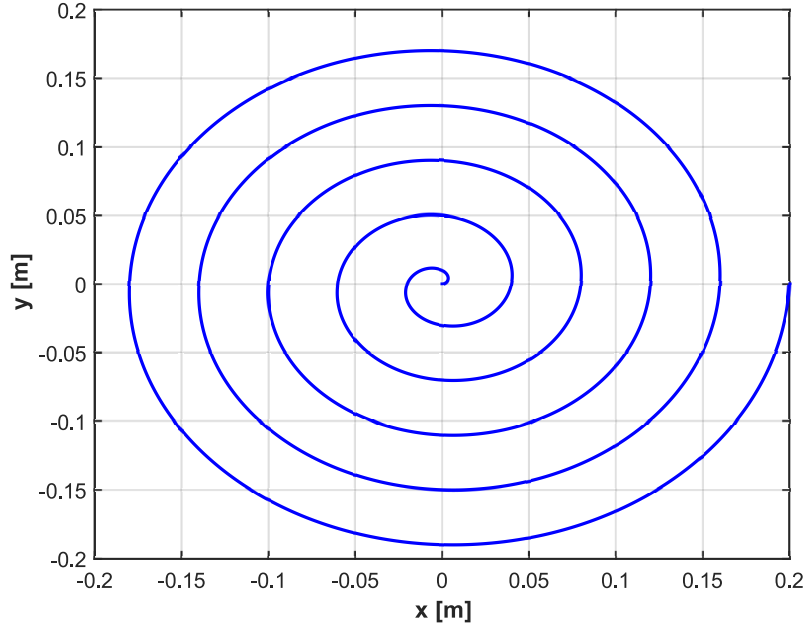


Figure 6.8: Desired spiral trajectory in the plane  $(x,y)$  for the Scenario 3

### 6.2.4 Performance evaluation criteria

Before proceeding to show the graphs of the experimental and simulation results of Delta PKM. Let us to formulate the RMSE formulas to have concrete data about the performance of the controllers. The following equations established the RMSE in Cartesian and joint space form respectively:

$$RMSE_C = \sqrt{\frac{1}{N} \sum_{k=1}^N (e_x^2(k) + e_y^2(k) + e_z^2(k))} \quad (6.5)$$

$$RMSE_J = \sqrt{\frac{1}{N} \sum_{k=1}^N (e_{11}^2(k) + e_{12}^2(k) + e_{13}^2(k))} \quad (6.6)$$

where  $e_x, e_y, e_z$  denote the Cartesian position tracking error of the traveling plate along the  $x, y, z$  axes, while  $e_{11}, e_{12}, e_{13}$  are the different joint space tracking errors. Moreover,  $N$  is the number of samples and  $k$  the current sample.

### 6.2.5 Tuning gains procedure

Given the significant number of tuning parameters for the control solution 3 and the amount of uncertainties present in the prototype, which is not fully completed, the controller's gains for real-time experiments and those of the controllers implemented via numerical simulations are adjusted by trial and error method.

Table 6.1: Controller parameters used in the Delta PKM

PID	RISE/RISE FW	RISE BSNN
$K_p=50$	$\Lambda_1=110$	$\Lambda_1=110$
$K_d=2.2$	$\Lambda_2=8$	$\Lambda_2=8$
$K_i=8.3$	$K_s=60$	$K_s=60$
	$\beta=3$	$\beta=3$
		$\gamma=0.53$

### Tuning gains for PID controller

The gains for the real-time experiments using a PID controller were sated in the following form: First, all gains were initialized in zero, then the proportional gain was increased until the robot began to move and whose movements oscillated around the proposed trajectory. In order to attenuate the generated oscillations, the derivative gain was incremented little by little until the overshoots were considerably reduced and taking care not to generate excessive noise. Finally, the integral gain was augmented until reaching values closer to the reference path. Table 6.1 present the gains' values of the PID controller in the first column.

### Tuning gains for RISE-based controllers

For the numerical simulations, a standard RISE and a RISE feedforward are evaluated together with the control solution 3. The common denominator of the three controllers is the RISE feedback part. Therefore, the feedback gains' values will be the same for these controllers. To adjust their values first, one chooses a sufficiently large value for  $K_s$  and  $\Lambda_1$ , and the gains  $\Lambda_2$  and  $\beta$  are initialized in zero. Then, the program is executed to compute the RMSE in the joint space. With the obtained result, the  $\Lambda_2$  and Beta values are increased until reaching the smallest possible RMSE at each simulation. In the case of  $\beta$  gain, it must be adjusted in such a way that there is not the least possible chattering in the control signals. For RISE with adaptive compensation, based-on BSNN (RISE BSNN), the learning rate can take a value between zero and one. And that value is adjusted based on the desired tracking speed. If high speeds are demanded, it is preferable to use an intermediate value because if a high gamma value is selected, there will be times when the neural network will tend to produce indeterminacies. The parameters of RISE based controllers used in the numerical simulations, are summarized in Table 6.1 columns 2 and 3.

## 6.3 Experimental Results with a PID Controller

Since the Delta PKM of the Polytechnic University of Tulancingo is still in development, there exist some issues are related to the mechanism and the electronic stage. For instance, a planetary gearhead must integrate into each servomotor to increase the output torque and the output shaft rigidity. Because of every time that the Delta PKM moves, there are multiple oscillations of their kinematic chains due to the servomotors' lack of force. For these reasons, the precision that the robot can achieve is very low. This experimentation aims not to make

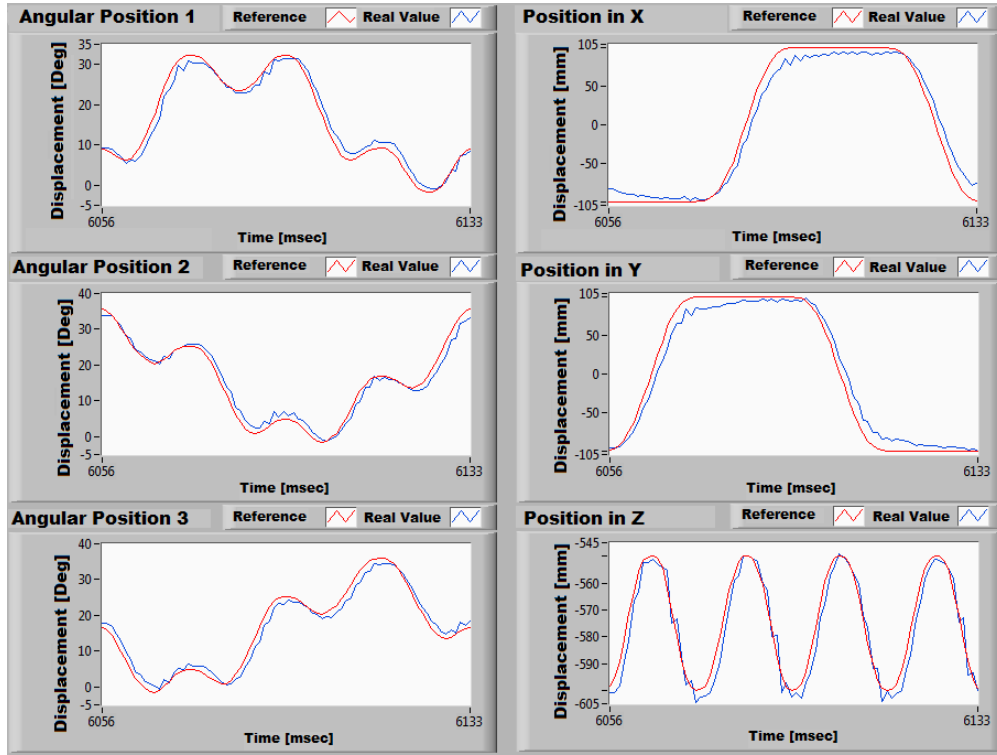


Figure 6.9: Tracking trajectory comparison in Cartesian and joint Space with respect to time

a performance comparison with other controllers as has been done in the previous chapter, but to show the progress that has been achieved in the development of the prototype. With this in mind, let us remember the equation of the PID controller in joint space:

$$\Gamma(t) = \mathbf{K}_p \mathbf{e}_q(t) + \mathbf{K}_d \dot{\mathbf{e}}_q(t) + \mathbf{K}_i \int_0^t \mathbf{e}_q(\tau) d\tau \quad (6.7)$$

Where  $\mathbf{K}_p$ ,  $\mathbf{K}_d$ ,  $\mathbf{K}_i \in \mathbb{R}^{n \times n}$  correspond to the proportional, derivative and integral gain matrices. The tracking error in joint space  $\mathbf{e}_q \in \mathbb{R}^n$  is defined as  $\mathbf{e}_q = \mathbf{q}_d - \mathbf{q}$ . The tracking trajectory graphs for Cartesian and joint space are presented in Figure 6.9. One can see that the robot is apparently approaching the desired references. However, looking at the graphs of the tracking errors in Figure 6.10, one can see that these errors are quite large. For the joint space, the error signals can reach 5 degrees, whereas that for the Cartesian space, there are tracking errors that reach up to 30 mm. So it is not recommended at this time to perform an operational task. Figure 6.11 shows the control voltage generated by the PID controller. Although the control signal can be expressed as a function of torque by means of the following relationship:

$$\Gamma(t) = \frac{K_a}{R_a} \mathbf{V}(t) \quad (6.8)$$

In which,  $K_a$  is the torque constant and  $R_a$  is the armature resistance of each of the servomotors. These constants can be found in the datasheet of the actuators.

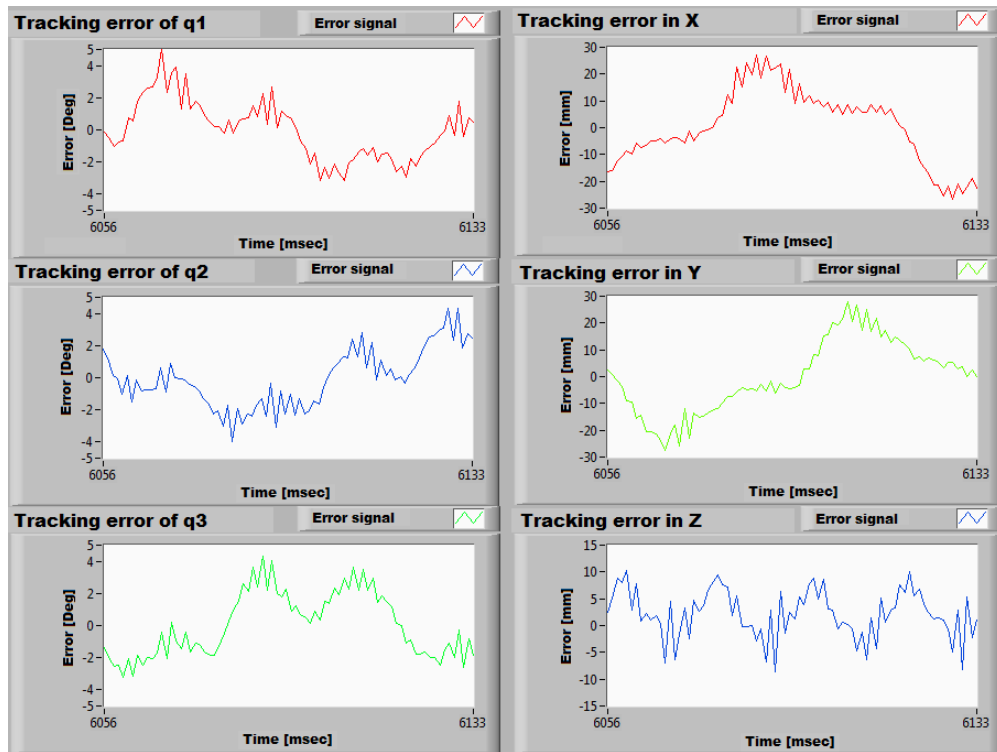


Figure 6.10: Tracking error evolution in Cartesian and joint space with respect to time

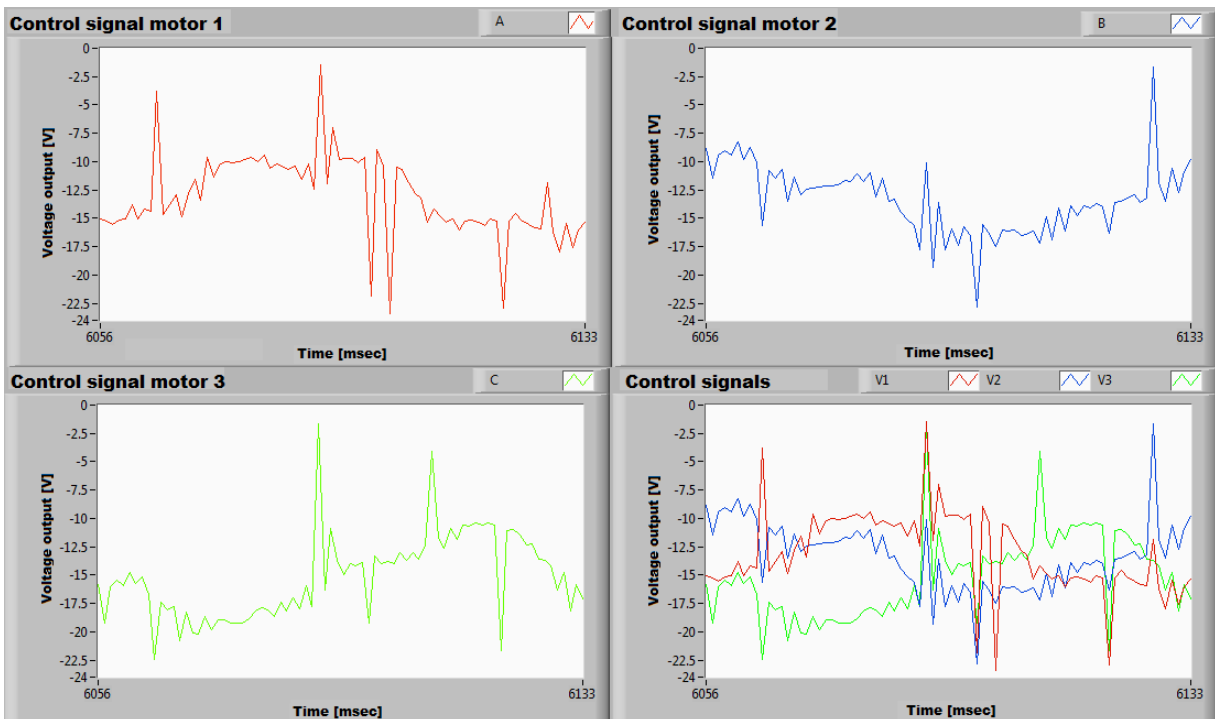


Figure 6.11: Controller signals



## 6.4 Simulation results of control solution 3

In this section, the results obtained by numerical simulations with the RISE BSNN are presented. The obtained performance of the control solution 3 is compared to a Standard RISE control under three different scenarios described previously in subsection 6.2.3. Moreover, a RISE FW controller is used to validate the obtained IDM approximation by the BSNN feedforward term. Both feedforward terms are compared in graphs to see the estimation given by AMNN.

### 6.4.1 Scenario 1: Nominal case

The tracking errors for the P&P trajectory without moving payload are presented in Figure 6.12 for both Cartesian and joint space. As it can be noted, the tracking errors of RISE BSNN are noticeably smaller than those of Standard RISE control due to the BSNN compensation terms reducing the effect of nonlinearities, resulting in better tracking performance. Figure 6.13 depicts the generated torques by the Standard RISE and our proposed RISE BSNN in the first column graphs, whereas the control signals that form our proposed controller (i.e., RISE contribution and BSNN contribution) are in the second column. It is noteworthy that the BSNNs outputs' behavior is very similar to the torques produced for both control schemes due to the accurate approximation of the inverse dynamic model of Delta PKM computed by the BSNNs. Moreover, as can be seen in the same Figure, the BSNN control term produces most of the torque required to reach the desired position this is due to its good approximation of the IDM for the Delta robot and, on the other hand, the term corresponding to the RISE control produces the extra torque needed to achieve the desired position accurately. The RMSEs' for this scenario 1 are registered in Table 6.4.3 with the improvements of RISE BSNN over standard RISE. One can appreciate in the Table that the Cartesian space's improvement is over 80% and 79% in the joint space, which represents an improvement in performance more than acceptable. Finally, in Figure 6.14, the evolution of the BSNNs' adaptive weighs for scenario 1 is presented. Three sets of eight adaptive weighs are used to estimate the dynamic behavior of each kinematic chain of the Delta PKM. In the graphs, it can be appreciated that the initial value of all weights is zero, and as the trajectories are executed, not all the weights evolve together; this is because of the BSNNs update only the associate weights to the current input values of the BSNNs.

### 6.4.2 Scenario 2: Changes in the payload

By adding a mass of one kilogram to certain parts of the P&P trajectory, as shown in Figure 6.6b, the following results are obtained. The tracking errors for the second scenario are represented in the graphs of Figure 6.15. It can be appreciated that the amplitude of tracking errors has increased for the two controllers as a consequence of the addition of the moving mass. However, the RISE BSNN control law's performance is still widely better than the Standard RISE controller. The produced control signals of both control schemes are provided

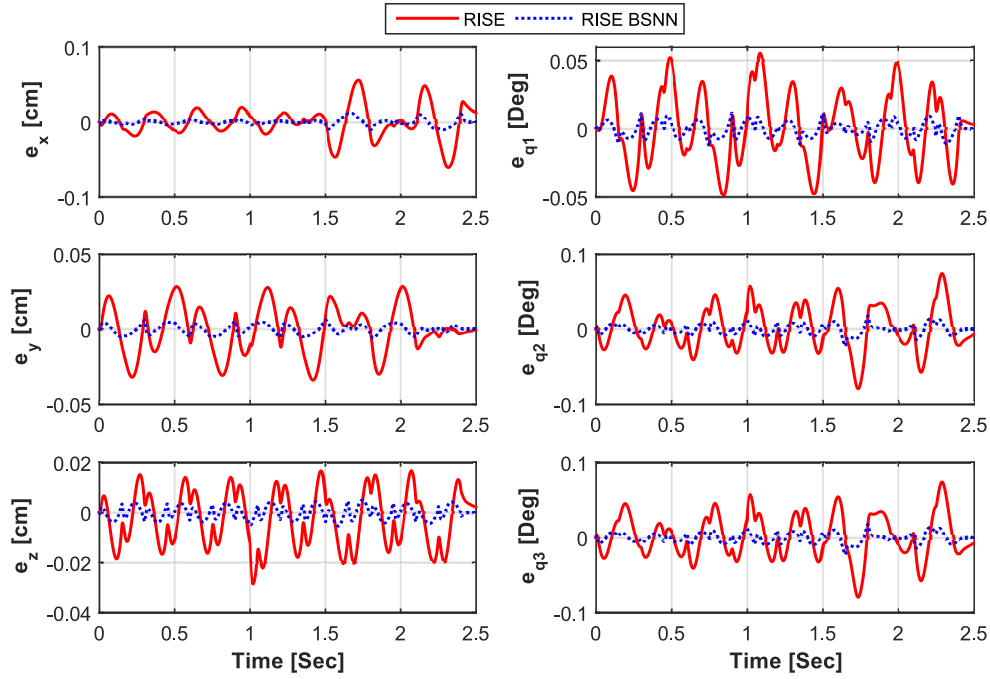


Figure 6.12: Evolution of the tracking errors versus time in Cartesian and joint space for scenario 1

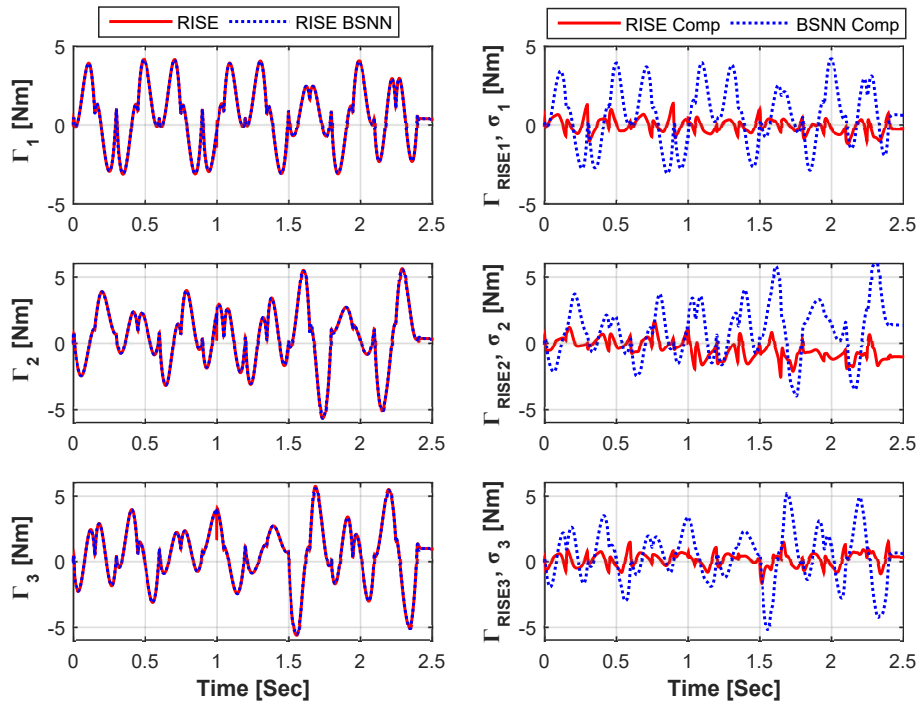


Figure 6.13: Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time for scenario 1

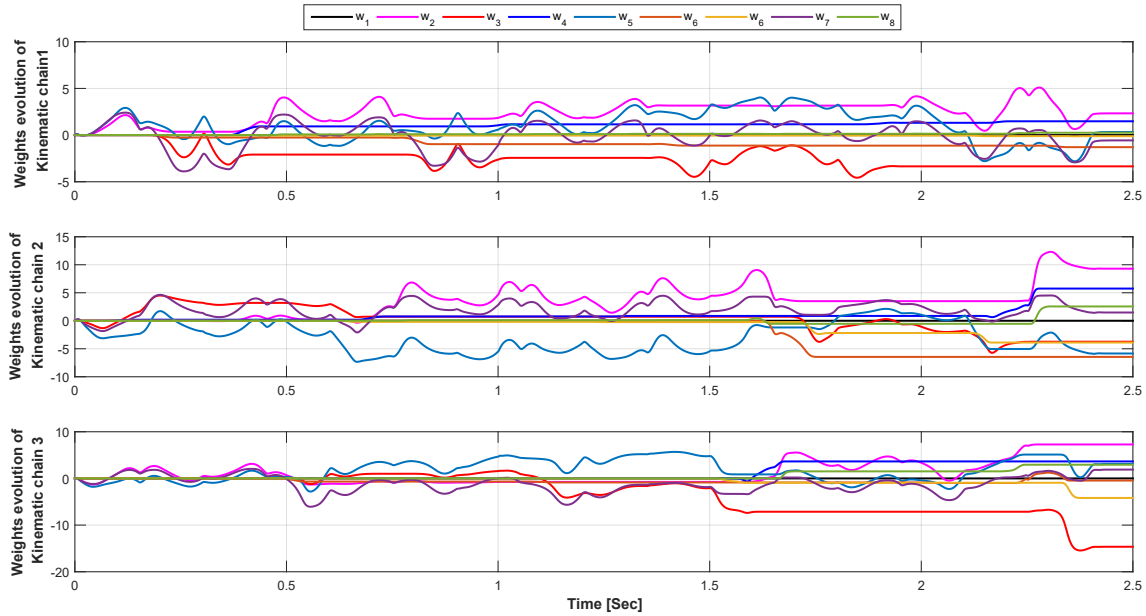


Figure 6.14: Evolution of the BSNNs' weights of scenario 1

in the first column of Figure 6.16, and the torques' contributions of the RISE term and the BSNN term of control solution 3 are in the second column. In all graphs, it can be noted that the curves have doubled compared to control signals for scenario 1, owing to the controllers requiring more energy to move the payload from one point to another. To quantify the obtained controllers' performance for this scenario, we compute the RMSEs using the tracking errors in Cartesian and joint space. The results are presented in Table 6.4.3. One can see that even though the RMSE values of the two controllers increased. The RISE BSNN controller improvement over the standard RISE remains above 80% for Cartesian space and 79% for joint space. Lastly, the evolution of the BSNNs' adaptive weights is presented in Figure 6.17. It can be seen that, as the previous scenario, all the values start at zero, and only some of the weights change their value along the trajectory. The main change in this simulation with respect to the previous one is the amplitude of these weights since they reach values greater than those presented in Figure 6.14.

### 6.4.3 Scenario 3: Changes in the speed

This scenario's objective is to know how much the changes in speed affect the controllers' performance. The spiral trajectory used for this scenario is expected to be completed in 14.8 s, 2.85 s, and 1.42 s according to the speed execution. The tracking errors in Cartesian and Joint space are presented in Figures 6.18, 6.19, and 6.20 for the speeds executions (low, medium, and high). As it can be noticed, as the speed is increasing, the overshoots amplitude on the tracking error signals also increases. Nevertheless, the proposed RISE BSNN controller's tracking errors always remain lower than those produced by the standard RISE controller. The produced torques of both controllers are presented the first column of Figures 6.21, 6.22,

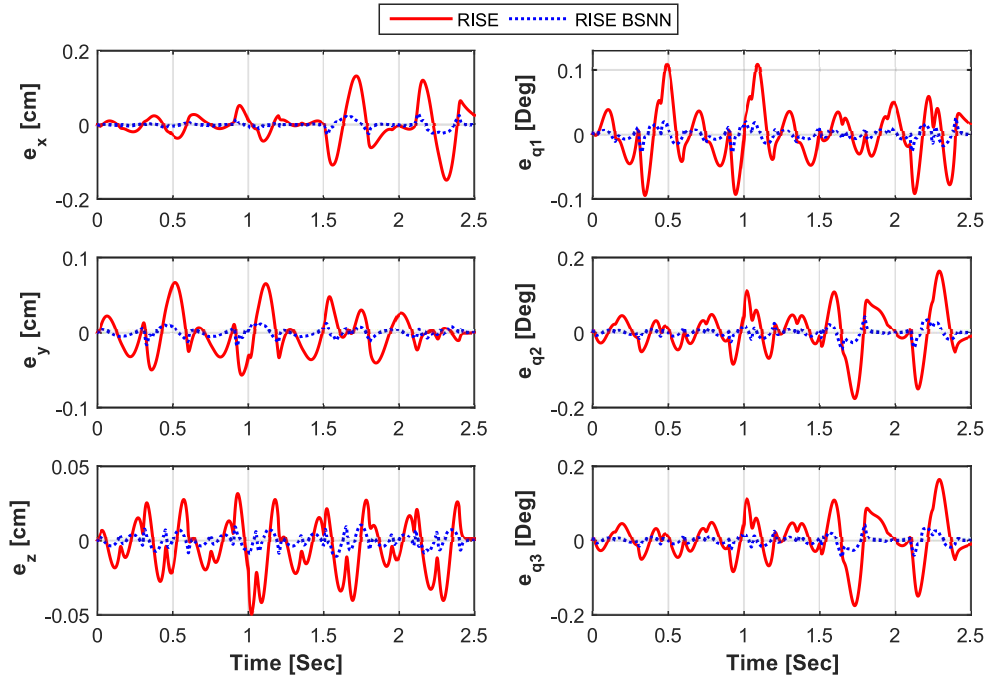


Figure 6.15: Evolution of the tracking errors versus time in Cartesian and joint scenario 2

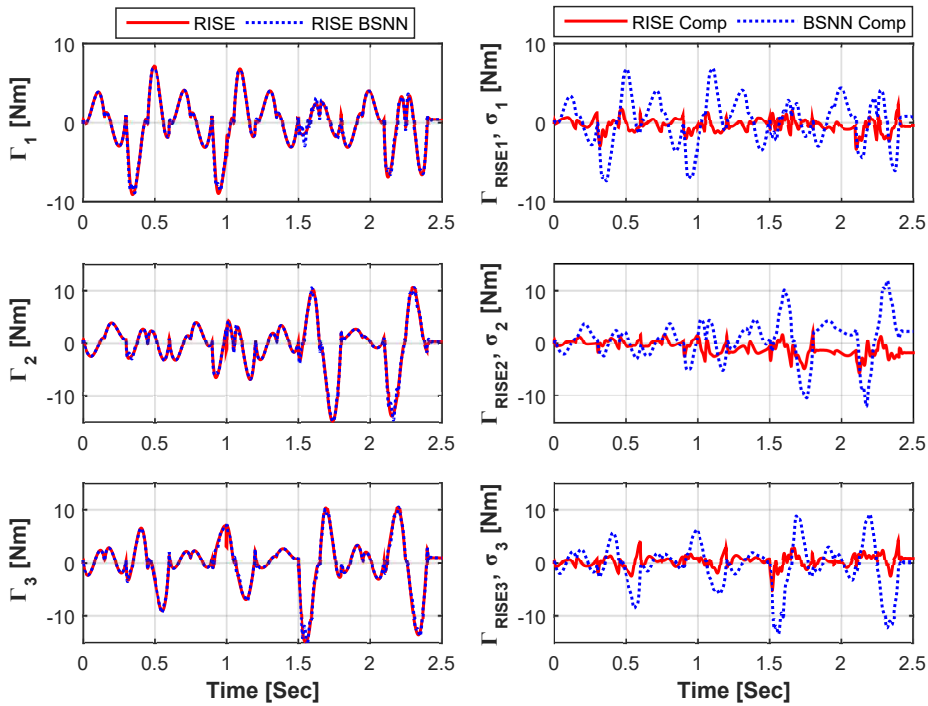


Figure 6.16: Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time scenario 2

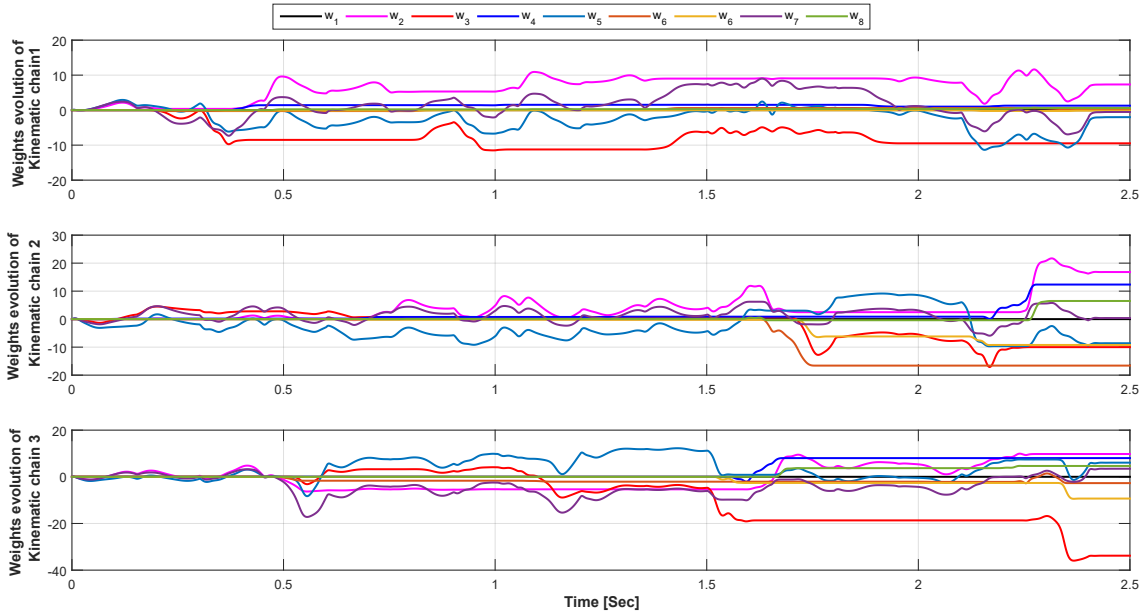


Figure 6.17: Evolution of the BSNNs' weights of scenario 2

Table 6.2: Controllers performance evaluation for scenario 1 and 2

Scenario	Controller	RMSE <sub>C</sub> [cm]	RMSE <sub>J</sub> [Deg]
Scenario 1	RISE	0.0285	0.0491
	RISE BSNN	0.0055	0.0102
	<b>Enhancement</b>	80.6%	79.1%
Scenario 2	RISE	0.0571	0.0929
	RISE BSNN	0.0109	0.0194
	<b>Enhancement</b>	80.9%	79.1%

and 6.23 and the contributions of the RISE part and the BSNN of the control solution 3 are in the second column of the same figures. It is possible to see that when the speed increases, also the energy provide by the computed control signals increase. Moreover, the control actions of the BSNNs contribute in a more significant proportion than the RISE contribution. Figure 6.24 presents the weighs evolution respect to time for the three execution speeds (low, medium, and high). Similar as scenarios 1 and 2, all the weights values are initialized in zero as can be appreciate in the six graphs. In low speed, we can see that only four weights are changed along the trajectories owing to the input values of the desired trajectories stay in the range values of only one basis function; unlike in high speed where all weights are in involved since the desired trajectories reach the maximum limits of the knot-points distribution. Table 6.4.3 presents the comparison of different RMSEs for the three speeds reinforcing the advantages of our proposed control solution. In all scenarios of this case study, the improvement of our controller compared to Standard RISE is between 60% and 80%. To have a better comprehension of how great the deterioration of the control schemes as the speed increases is, the RMSE is plotted in Figure 6.25.

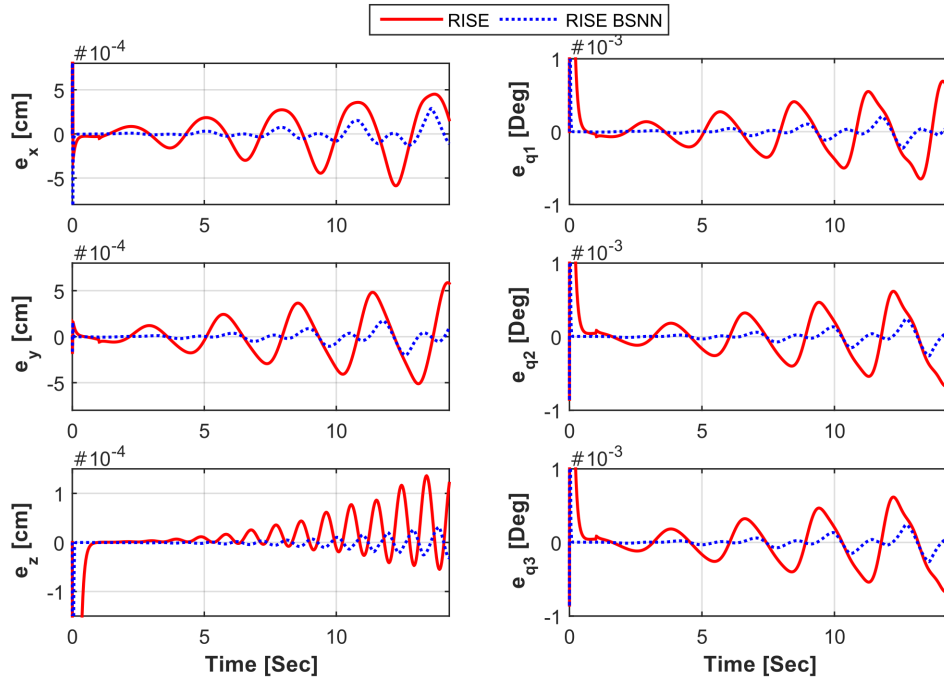


Figure 6.18: Evolution of the tracking errors versus time in Cartesian and joint space corresponding to low speed for scenario 3

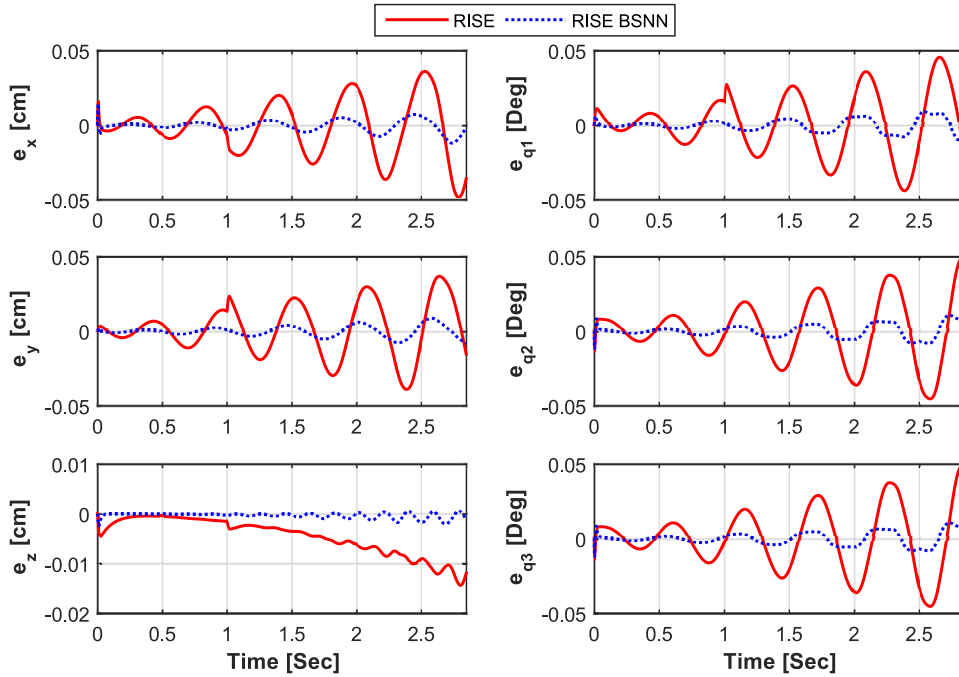


Figure 6.19: Evolution of the tracking errors versus time in Cartesian and joint space corresponding to medium speed for scenario 3

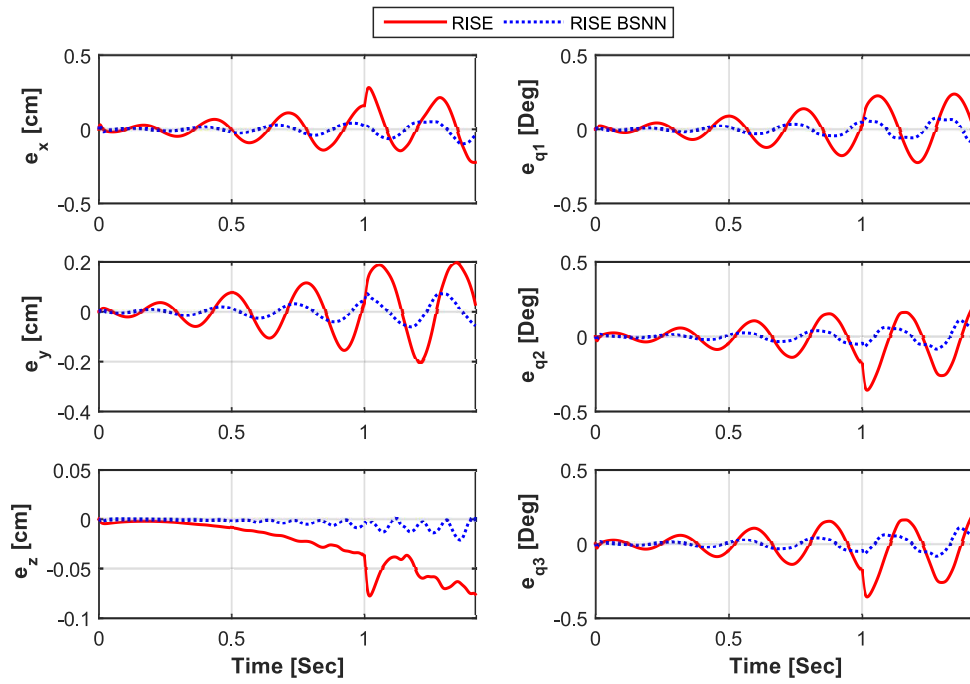


Figure 6.20: Evolution of the tracking errors versus time in Cartesian and joint space corresponding to high speed for scenario 3

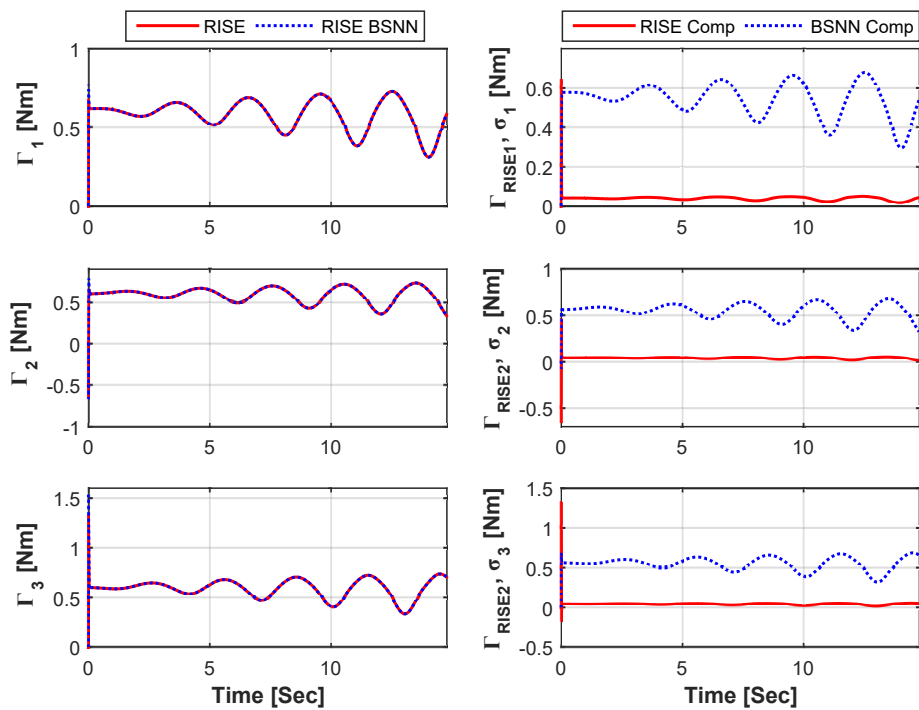


Figure 6.21: Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time that corresponds to low speed scenario 3

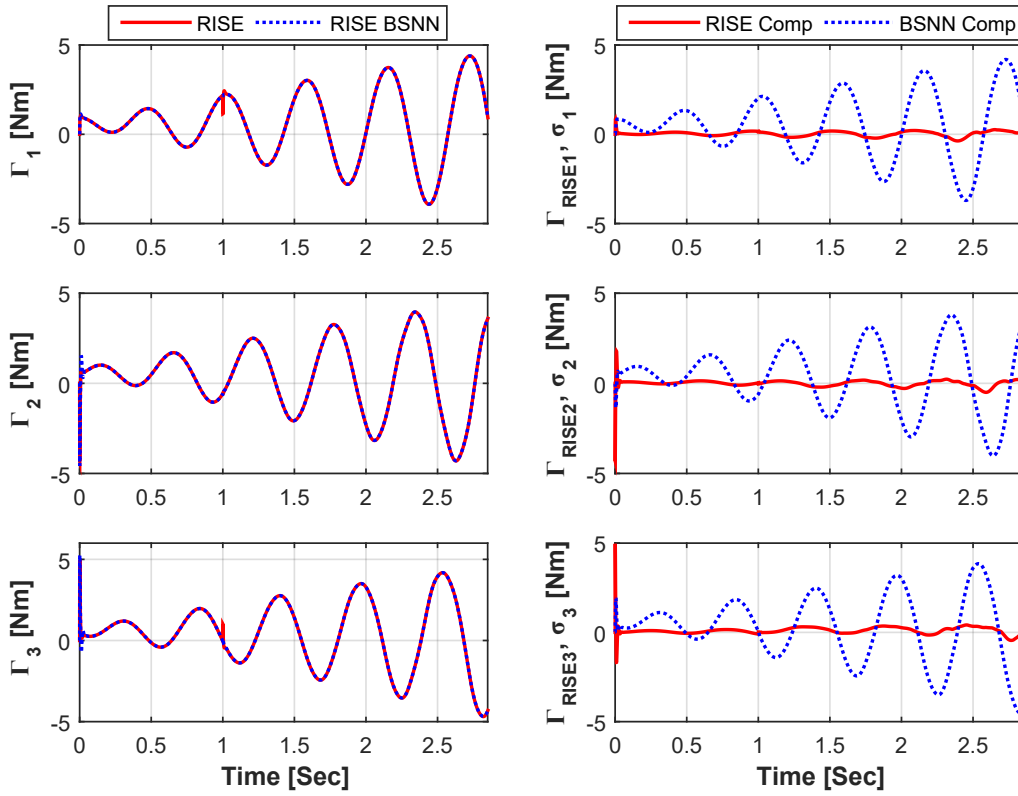


Figure 6.22: Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time that corresponds to medium speed scenario 3

Table 6.3: Controllers performance evaluation for scenario 3

Speed	Controller	RMSE <sub>C</sub> [cm]	RMSE <sub>J</sub> [Deg]
Low	RISE	$4.622 \times 10^{-4}$	0.0010
	RISE BSNN	$1.314 \times 10^{-4}$	$2.670 \times 10^{-4}$
	<b>Enhancement</b>	71.5%	73.6%
Medium	RISE	0.0262	0.0362
	RISE BSNN	0.0052	0.0073
	<b>Enhancement</b>	80.0%	79.8%
High	RISE	0.1413	0.1947
	RISE BSNN	0.0434	0.0606
	<b>Enhancement</b>	69.8%	68.5%



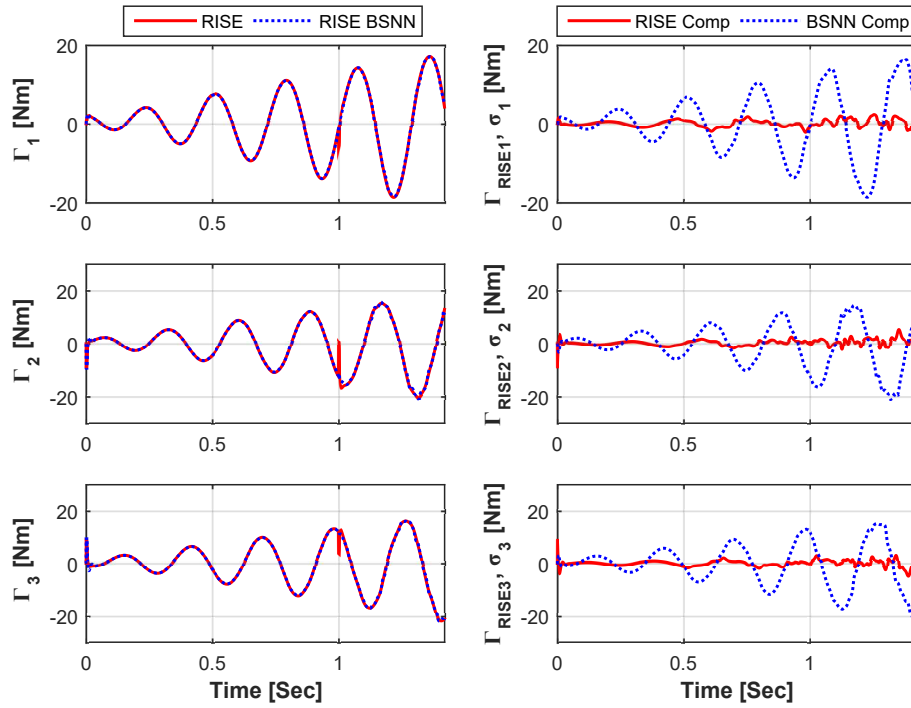


Figure 6.23: Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time that corresponds to medium high scenario 3

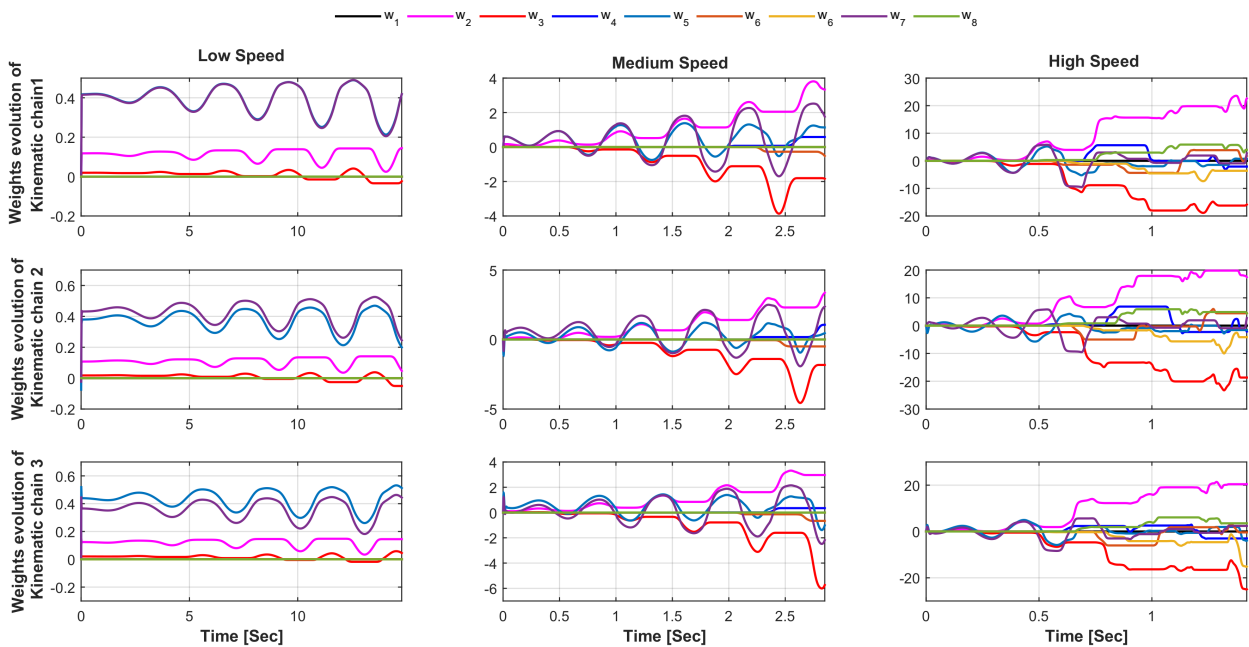


Figure 6.24: Evolution of the BSNNs' weights for scenario 3 when the Delta PKM is subjected to changes in the speed

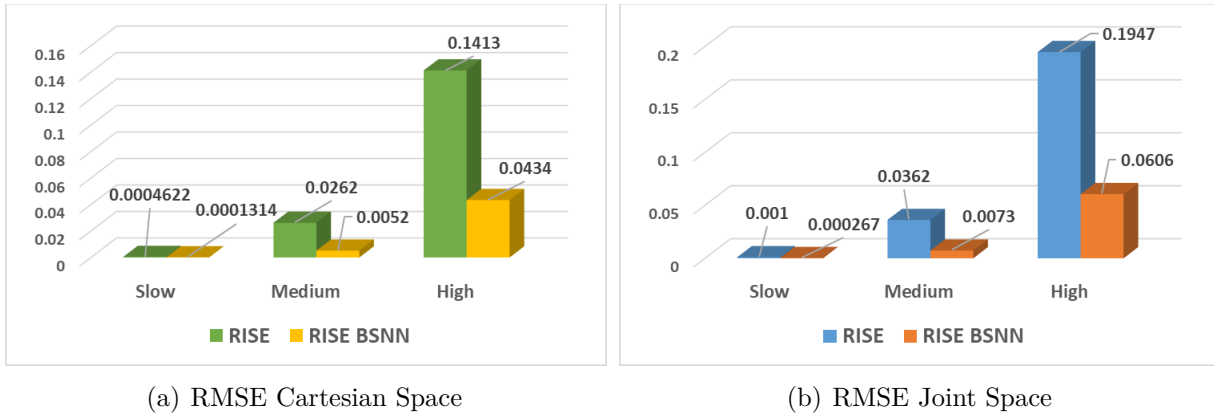


Figure 6.25: Degradation graphs of RMSE at different speeds for Cartesian and joint space in scenario 3

#### 6.4.4 Comparison of BSNN compensation against nominal feed-forward

In the previous case studies, the proposed RISE with BSNN compensation was evaluated to standard RISE control, and the results obtained were notably superior. However, as it was mentioned before, the BSNN compensation term aims to emulate the Nominal feedforward term. Therefore, in this subsection, our proposed control solution is compared to a RISE feedforward controller to validate the approximation of the dynamics given by the BSNN. The case study 1, including the two scenarios, is considered for this validation. Figure 6.26 depicts the tracking error in the joint space of the RISE feedforward and RISE BSNN and the components compensation of both controllers when no payload is moving. It can be appreciated that the tracking error of the RISE feedforward is prominently better than the RISE BSNN due to the evaluated dynamic parameters in the feedforward part are entirely known, unlike the RISE BSNN, where the dynamic behavior of the Delta PKM is on-line estimated. However, note that the produced compensation terms of the BSNN are similar to those produced by the nominal feedforward even without any information on the system dynamics. The obtained  $RMSE_q$  is 0.0045 for the RISE feedforward and 0.0102 for the RISE BSNN, the first controller outcomes the second one in 56.69% for this scenario. Nevertheless, for the second scenario where a mass of 1 kg is moved in some portions of the trajectory, the performance of the RISE BSNN is better than the RISE feedforward, due to RISE BSNN can compensate for the parametric uncertainty produced by the changes in the payload along the trajectory, unlike the RISE feedforward, where the dynamic parameters are not updated as can be demonstrated on Figure 6.27. The resulting  $RMSE_q$  for the second scenario is 0.0494 for RISE feedforward and 0.0194 for RISE BSNN, yielding an improvement of 60% of RISE BSNN over the RISE feedforward. With the curves provided in the second column of Figures 6.26 and 6.27, one can conclude that the dynamic approximation provided by the BSNNs is quite good.

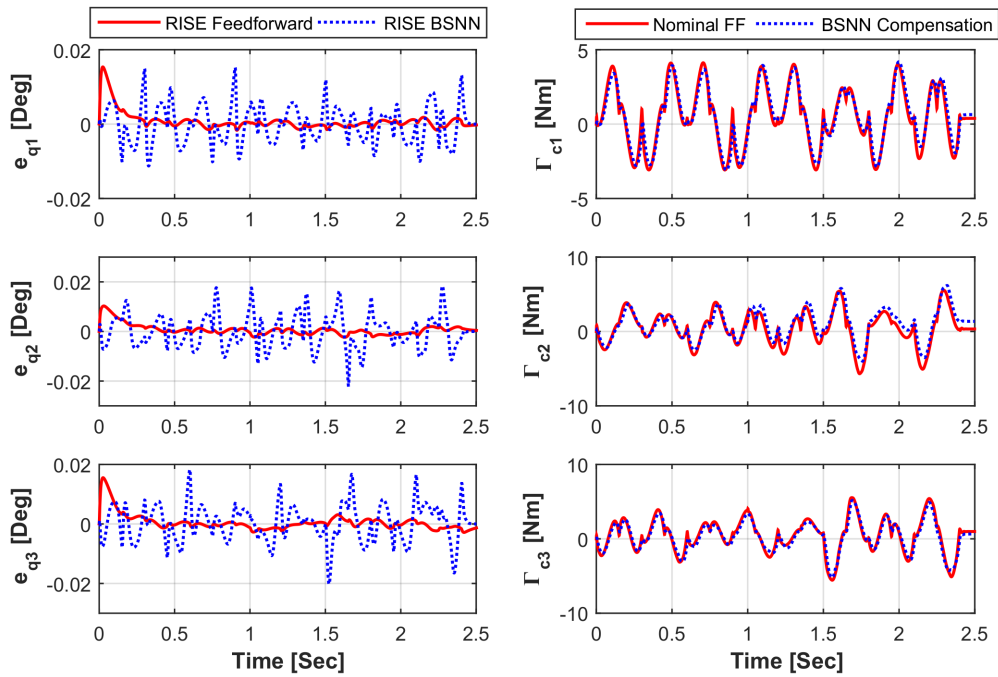


Figure 6.26: Performance comparison between RISE feedforward and RISE BSNN scenario 1

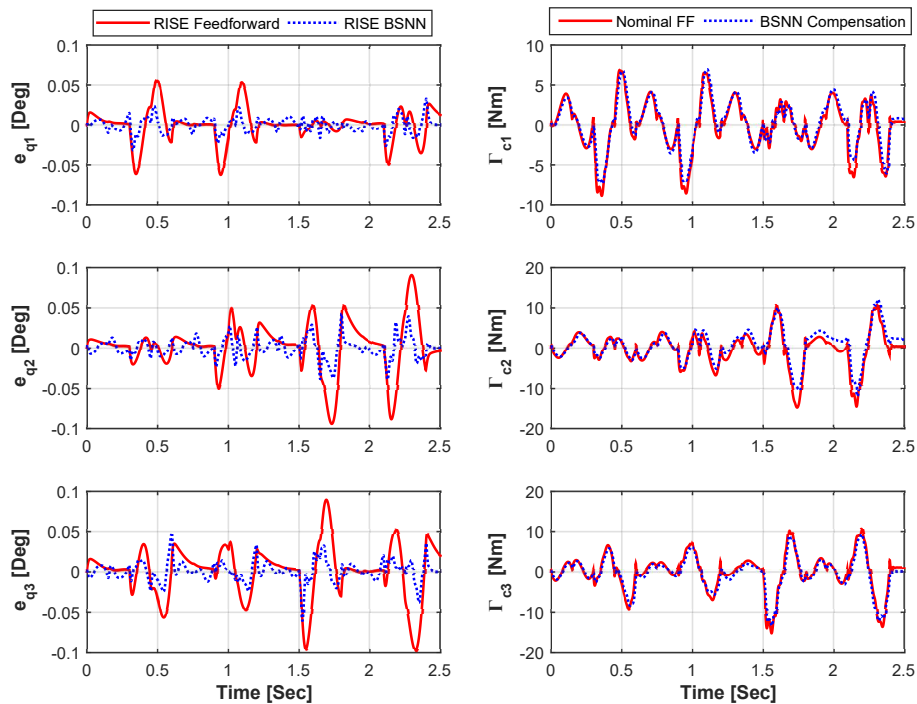


Figure 6.27: Performance comparison between RISE feedforward and RISE BSNN scenario 2

## 6.5 Conclusion

In this chapter, the experimental and simulation results carried out on the Delta PKM were presented. Since the Delta PKM prototype of the Polytechnic University of Tulancingo is still in development, there are many issues related to the mechanism and the electronic stage. Therefore, it is not suitable for the prototype to experiment with nonlinear complex control techniques. However, to demonstrate the progress in the prototype development, a PID controller was tested on a kind of P&P trajectory. The chapter also presents the numerical simulation results related to the RISE controller with adaptive feedback compensation based on B-spline neural networks. This controller was tested under three scenarios: nominal case, changes in the payload, and changes in the speed. The performance of this controller was validated against the standard RISE control showing a remarkable superiority in all aspects. Moreover, to confirm the dynamic model approximation obtained by the BSNN, a RISE feedforward control was tested together with the control solution 3 under scenarios 1 and 2. The obtained curves of the BSNN term and the nominal feedforward were compared in graphs where it was appreciated that the BSNNs make a fairly good approximation of the robot's dynamics.



---

## General Conclusion

The objectives of this thesis have been mainly two which are intertwined with each other. The first was to design a dynamic model for the new machining platform called SPIDER4, and the second was to propose and validate different control techniques that help improve the performance of parallel robots in different operating environments such as handling and machining tasks. The real-time experiments and simulation results corroborated the new control proposals' effectiveness against speed changes, sudden changes in payload, and contact forces.

### Resume of the work

It was previously mentioned that the final performance of a parallel robot is a function of various factors among themselves, such as mechanical design, mathematical modeling, and the control scheme. All these factors must be well designed in order to get the most out of the closed-loop architecture of PKMs. The thesis focused on the context of parallel robot control. As described in the first chapter, its control involves many challenges, such as highly nonlinear dynamics, the presence of parametric and non-parametric uncertainties, and in some cases, redundantly actuation. Therefore, the controllers that must be implemented must guarantee good precision in the task that the manipulator is developing despite the problems mentioned above. The contributions of the thesis can be summarized as follows:

1. The proposal of the kinematic and dynamic models for the SPIDER4 platform, which can be said to be a hybrid architecture robot formed by the parallel architecture positioning device which represents most of the robot and the wrist, which is a series type mechanism in charge of orienting the robot's cutting spindle. The IKM of the parallel device is an adaptation of the 3-DOF Delta PKM, while the IKM of the wrist was obtained by employing transformation matrices as in serial robots. The FKM of the parallel mechanism was obtained through a virtual sphere intercession algorithm, while that of the wrist was calculated using the same transformation matrices used for its IKM. The IDM of the SPIDER4 parallel positioning device was calculated based on the method developed in AA for robots with delta-type architecture. The main modification lies in

the formulation of the acceleration of gravity in the kinematic chains of SPIDER4 since these are oriented horizontally instead of vertically, which produces a direct effect on the output torque of the motors. The dynamic model of the wrist was calculated using the Euler-Lagrange formulation. In the end, both models were combined to express the total SPIDER4 IDM in the joint space. However, for the experimentation of control techniques, only the IDM of the parallel positioning device was considered, since due to software problems, it was not possible to modify the wrist control law. Finally, the kinematic and dynamic parameters of the robot were obtained through datasheets and Solidworks calculations.

2. A RISE controller with nominal feedforward was proposed to validate the IDM of the SPIDER4 parallel positioning device via real-time experiments. It is well known that the total or partial integration of the IDM in a control scheme markedly improves the system's performance to be controlled. However, a bad approximation of the model, together with an imperfect estimation of the dynamic parameters, can degrade the system's performance instead of improving it. The proposed RISE feedforward controller was tested together with the standard RISE control in order to see the improvement produced by incorporating the dynamic model proposed as the lead compensation term. The experimental results showed improvements of over 23%, demonstrating the efficacy of the proposed IDM.
3. SPIDER4 RA-PKM is designed to perform machining tasks such as drilling or milling in resin materials. Due to the existence of contact forces, some CNC machines incorporate force controllers to counteract these forces. In order to implement a force controller, the robot must have sensors that measure the contact force or the current consumed by the robot's actuators. However, SPIDER4 does not have this type of sensor to implement a force control loop. So it was proposed to modify the RISE feedforward controller by replacing fixed feedback gains with adaptive gains. The purpose of these gains is that when the tracking errors grow noticeably due to contact forces, the gains automatically increase their values to counteract such tracking errors. By reducing the tracking error, the gains will automatically adjust to the most optimal value. The validation of this control scheme was carried out through real-time experiments. The experiments consisted of the machining of a piece of Styrofoam. The machining trajectory was executed with different advanced speeds to see its impact on the SPIDER4's performance. The performance of the RISE feedforward controller with adaptive gains was compared to that of the RISE feedforward and the standard RISE. The results showed a very noticeable improvement respecting the other controllers when the machining task was executed with  $F=1200$  mm/min and  $F=2400$  mm/min. However, when the speed was exceeded at  $F=24000$  mm/min, the performance obtained was slightly higher than that of RISE feedforward. Therefore, it can be concluded that this control proposal is suitable for machining tasks in soft materials as long as the forward speed is not exceeded.
4. A RISE controller with feedforward compensation term based-on BSNN was proposed to regulate the positioning of a 3-DOF Delta PKM performing P&P tasks. The objective

of the BSNN is the estimation of the dynamics in real-time of the kinematic chains of the robot and the mobile platform. This objective arises due to the fact that in reality, some dynamic parameters and portions of the dynamic model are not precisely known; in addition, the mass variations in the payload in PP operations represent an additional uncertainty. Therefore, the use of ANN represents an attractive solution to estimate the dynamic model even in the presence of parametric uncertainties, such as the change in mass values. This control proposal was validated under three different scenarios with numerical simulations because the prototype developed at the Polytechnic University of Tulancingo has not yet been completed. The performance of this control proposal was compared with that obtained through the standard RISE control. The results obtained showed a remarkable improvement even with variations in the payload and the speed execution. Regarding the under development Delta PKM prototype, real-time experimental results of implementing a PID controller were presented. This implementation intended only to show the achieved progress in the development of the prototype.

## Future works perspective

The work presented in research focused on two experimental platforms: the SPIDER4 robot from LIRMM and the Delta PKM prototype from the Polytechnic University of Tulancingo. Based on the results obtained in this work, the following future works can be formulated for both platforms.

- **Implementation of a force control loop for SPIDER4.** To carry out this action, it will be necessary to reconfigure the inputs of the X20 system to obtain data from the current measurements of each actuator, and, in this way, the system will be ready for the second control loop to be applied. The force control will provide greater robustness to the position control, and it opens the possibility of working with slightly harder materials.
- **Experimentation of control schemes in Cartesian space for SPIDER4.** It is speculated that the implementation of control schemes in the workspace may have better results than those implemented in the joint space because there are slight inaccuracies when performing inverse kinematics calculations. To avoid that, SPIDER4 includes a measurement system that allows obtaining the nacelle's real position without performing inverse kinematics calculations. Therefore, the robot's precision could be further improved by implementing a control scheme of this type.
- **Real-time implementation of the RISE BSNN controller on the Delta PKM prototype.** To carry out this action, the prototype must be completed, including the improvement of the power stage and gearboxes' addition to the servo motors. Once the experimental platform has been completed, the obtained numerical simulation results of the RISE BSNN controller can be verified via real-time experiments.





## APPENDIX



### Proof of lemma 1

This proof is implied in the stability analysis for the RISE control with nominal feedforward, and RISE feedforward with adaptive feedback gains.

Establishing the integral with respect to time of (4.31), the following is obtained:

$$\int_0^t \mathbf{L}(\tau) d\tau = \int_0^t \mathbf{r}(\tau) (\mathbf{N}_d(\tau) - \mathbf{N}_{FW}(\tau) - \beta \text{sgn}(\mathbf{e}_2(\tau))) d\tau \quad (\text{A.1})$$

Substituting (4.32) into (A.1) we get:

$$\begin{aligned} \int_0^t \mathbf{L}(\tau) d\tau &= \int_0^t \Lambda_2 \mathbf{e}_2(\tau)^T (\mathbf{N}_d(\tau) - \mathbf{N}_{FW}(\tau) - \beta \text{sgn}(\mathbf{e}_2(\tau))) d\tau + \int_0^t \frac{d\mathbf{e}_2(\tau)^T}{dt} \mathbf{N}_d(\tau) d\tau \\ &\quad - \int_0^t \frac{d\mathbf{e}_2(\tau)^T}{d\tau} \mathbf{N}_{FW}(\tau) d\tau - \int_0^t \frac{d\mathbf{e}_2(\tau)^T}{d\tau} \beta \text{sgn}(\mathbf{e}_2(\tau)) d\tau \end{aligned} \quad (\text{A.2})$$

Using the integration by parts for the right-hand of equation (A.2), the following is obtained

$$\begin{aligned} \int_0^t \mathbf{L}(\tau) d\tau &= \int_0^t \Lambda_2 \mathbf{e}_2(\tau)^T (\mathbf{N}_d(\tau) - \mathbf{N}_{FW}(\tau) - \int_0^t \frac{d\mathbf{e}_2(\tau)^T}{d\tau} \beta \text{sgn}(\mathbf{e}_2(\tau)) d\tau \beta \text{sgn}(\mathbf{e}_2(\tau))) d\tau \\ &\quad + \mathbf{e}_2(\tau)^T \mathbf{N}_d|_0^t - \int_0^t \mathbf{e}_2(\tau) \frac{d\mathbf{N}_d(\tau)}{d\tau} - \mathbf{e}_2(\tau)^T \mathbf{N}_{FW}|_0^t + \int_0^t \mathbf{e}_2(\tau) \frac{d\mathbf{N}_{FW}(\tau)}{d\tau} \end{aligned} \quad (\text{A.3})$$

Rearranging the terms of (A.3), we obtain:

$$\begin{aligned} \int_0^t \mathbf{L}(\tau) d\tau &= \int_0^t \Lambda_2 \mathbf{e}_2(\tau)^T \left( \mathbf{N}_d(\tau) - \mathbf{N}_{FW}(\tau) + \frac{1}{\Lambda_2} \left( \frac{d\mathbf{N}_{FW}(\tau)}{d\tau} - \frac{d\mathbf{N}_d(\tau)}{d\tau} \right) - \beta \text{sgn}(\mathbf{e}_2(\tau)) \right) d\tau \\ &\quad + \mathbf{e}_2^T(t) (\mathbf{N}_d(t) - \mathbf{N}_{FW}(t)) - \mathbf{e}_2^T(0) (\mathbf{N}_d(0) - \mathbf{N}_{FW}(0)) - \beta \|\mathbf{e}_2(t)\| + \beta \|\mathbf{e}_2(0)\| \end{aligned} \quad (\text{A.4})$$

Upper bounding the right-hand side of (A.4), the following is obtained:

$$\begin{aligned} \int_0^t \mathbf{L}(\tau) d\tau &\leq \int_0^t \Lambda_2 \|\mathbf{e}_2(\tau)\| \left( \|\mathbf{N}_d(\tau)\| - \|\mathbf{N}_{FW}(\tau)\| + \frac{1}{\Lambda_2} \left( \left\| \frac{d\mathbf{N}_{FW}(\tau)}{d\tau} \right\| - \left\| \frac{d\mathbf{N}_d(\tau)}{d\tau} \right\| \right) - \boldsymbol{\beta} \right) d\tau \\ &\quad + \|\mathbf{e}_2(t)\| (\|\mathbf{N}_d(t)\| - \|\mathbf{N}_{FW}(t)\| - \boldsymbol{\beta}) + \boldsymbol{\beta} \|\mathbf{e}_2(0)\| + \mathbf{e}_2^T(0)(\mathbf{N}_{FW}(0) - \mathbf{N}_d(0)) \end{aligned} \quad (\text{A.5})$$

Therefore, one can see from (A.5) that if  $\boldsymbol{\beta}$  is selected according to (4.30), then (4.31) is satisfied.



## Trajectory points for SPIDER4 Real-time experiments

This appendix presents the values of the commands used to generate trajectories in the SPIDER4 workspace for the nominal scenario (scenario 1) and the machining scenario (scenario 2). The units for  $x$ ,  $y$ , and  $z$  are given in  $mm$ , whereas that for  $\phi$ ,  $\psi$ ,  $I$ ,  $K$ ,  $H$ , and  $R$  are in degrees.

### B.1 Trajectory points for Scenario 1

The following sequence of points produces the graph of the trajectory for the first scenario; which can be viewed in the third dimension in Figure 5.4.

Table B.1: List of commands used for trajectory generation for scenario 1

Line	G command	$x$	$y$	$z$	$\phi$	$\psi$	I	J	H	R
N01	G1	0	0	-1500	0	0	0	0	0	0
N02	G1	0	0	-1650	0	0	0	0	0	0
N03	G1	100	100	-1650	0	0	0	0	0	0
N04	G1	-100	100	-1650	0	0	0	0	0	0
N05	G1	-100	-100	-1650	0	0	0	0	0	0
N06	G1	100	-100	-1650	0	0	0	0	0	0
N07	G1	0	0	-1650	0	0	0	0	0	0
N08	G1	0	0	-1650	20	0	0	0	0	0
N09	G1	0	0	-1650	-20	0	0	0	0	0
N10	G1	0	0	-1650	20	0	0	0	0	0
N11	G1	0	0	-1650	-20	0	0	0	0	0
N12	G1	0	0	-1650	0	0	0	0	0	0
N13	G1	0	0	-1650	-15	20	0	0	0	0
N14	G1	0	0	-1650	15	-20	0	0	0	0

N15	G1	0	0	-1650	-15	20	0	0	0	0
N16	G1	0	0	-1650	15	-20	0	0	0	0
N17	G1	0	0	-1650	0	0	0	0	0	0
N18	G1	0	0	-1600	0	0	0	0	0	0
N19	G1	0	100	-1600	0	0	0	0	0	0
N20	G3	0	100	-1600	0	0	0	-100	720	0
N21	G1	0	100	-1600	0	0	0	0	0	0
N22	G1	0	100	-1500	0	0	0	0	0	0

## B.2 Trajectory points for scenario 2

The following sequence of points produces the graph of the trajectory for the second scenario; which can be viewed in the third dimension in Figure 5.7.

Table B.2: List of commands used for trajectory generation for scenario 2

Line	G command	$x$	$y$	$z$	$\phi$	$\psi$	I	J	H	R
N01	G1	0	0	-1500	0	0	0	0	0	0
N02	G1	0	0	-1700	0	0	0	0	0	0
N03	G1	0	-245	-1700	0	0	0	0	0	0
N04	G1	0	-245	-1792	0	0	0	0	0	0
N05	G3	0	-245	-1792	0	0	0	-30	0	0
N06	G1	0	-245	-1750	0	0	0	0	0	0
N07	G1	44.166	-291.074	-1750	0	0	0	0	0	0
N08	G1	44.166	-291.074	-1792	0	0	0	0	0	0
N09	G2	59.194	-296.548	-1792	0	0	0	0	0	8
N10	G2	21.548	-334.194	-1792	0	0	0	0	0	63
N11	G2	16.074	-319.166	-1792	0	0	0	0	0	8
N12	G3	44.166	-291.074	-1792	0	0	0	0	0	47
N13	G1	44.166	-291.074	-1750	0	0	0	0	0	0
N14	G1	-44.166	-291.074	-1750	0	0	0	0	0	0
N15	G1	-44.166	-291.074	-1792	0	0	0	0	0	0
N16	G3	-59.194	-296.548	-1792	0	0	0	0	0	8
N17	G3	-21.548	-334.194	-1792	0	0	0	0	0	63
N18	G3	-16.074	-319.166	-1792	0	0	0	0	0	8
N19	G2	-44.166	-291.074	-1792	0	0	0	0	0	47
N20	G1	-44.166	-291.074	-1750	0	0	0	0	0	0
N21	G1	-44.166	-258.926	-1750	0	0	0	0	0	0
N22	G1	-44.166	-258.926	-1792	0	0	0	0	0	0
N23	G2	-59.194	-253.452	-1792	0	0	0	0	0	8
N24	G2	-21.548	-215.806	-1792	0	0	0	0	0	63
N25	G2	-16.074	-230.834	-1792	0	0	0	0	0	8

---

N26	G3	-44.166	-258.926	-1792	0	0	0	0	0	47
N27	G1	-44.166	-258.926	-1750	0	0	0	0	0	0
N28	G1	44.166	-258.926	-1750	0	0	0	0	0	0
N29	G1	44.166	-258.926	-1792	0	0	0	0	0	0
N30	G3	59.194	-253.452	-1792	0	0	0	0	0	8
N31	G3	21.548	-215.806	-1792	0	0	0	0	0	63
N32	G3	16.074	-230.834	-1792	0	0	0	0	0	8
N33	G2	44.166	-258.926	-1792	0	0	0	0	0	47
N34	G1	44.166	-258.926	-1750	0	0	0	0	0	0
N35	G1	0	-195	-1750	0	0	0	0	0	0
N36	G1	0	-195	-1792	0	0	0	0	0	0
N37	G3	0	-195	-1792	0	0	0	-80	0	0
N38	G1	0	-195	-1700	0	0	0	0	0	0
N39	G1	0	0	-1700	0	0	0	0	0	0
N40	G1	0	0	-1500	0	0	0	0	0	0

---





---

## References

- [Aboukarima et al., 2015] Aboukarima, A., Elsoury, H., and Menyawi, M. (2015). Artificial neural network model for the prediction of the cotton crop leaf area. *Int. J. Plant Soil Sci*, 8(4):1–13.
- [Ahmadi et al., 2008] Ahmadi, M., Dehghani, M., Eghtesad, M., and Khayatian, A. R. (2008). Inverse dynamics of hexa parallel robot using lagrangian dynamics formulation. In *2008 International Conference on Intelligent Engineering Systems*, pages 145–149. IEEE.
- [An et al., 1987] An, C., Atkeson, C., Griffiths, J., and Hollerbach, J. (1987). Experimental evaluation of feedforward and computed torque control. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 165–168. IEEE.
- [Angeles et al., 2005] Angeles, J., Morozov, A., and Bai, S. (2005). A novel parallel-kinematics machine tool. In *Proc. Sixth International Conference on Advanced Manufacturing Technologies*.
- [Armstrong and Wade, 2000] Armstrong, B. and Wade, B. A. (2000). Nonlinear pid control with partial state knowledge: Damping without derivatives. *The International Journal of Robotics Research*, 19(8):715–731.
- [Asif, 2012] Asif, U. (2012). Design of a parallel robot with a large workspace for the functional evaluation of aircraft dynamics beyond the nominal flight envelope. *International Journal of Advanced Robotic Systems*, 9(2):51.
- [Asl and Janabi-Sharifi, 2017] Asl, H. J. and Janabi-Sharifi, F. (2017). Adaptive neural network control of cable-driven parallel robots with input saturation. *Engineering applications of artificial intelligence*, 65:252–260.
- [Asl et al., 2018] Asl, H. J., Narikiyo, T., and Kawanishi, M. (2018). Neural network-based bounded control of robotic exoskeletons without velocity measurements. *Control Engineering Practice*, 80:94–104.

- [Baradat et al., 2009] Baradat, C., Nabat, V., Krut, S., Pierrot, F., et al. (2009). Par2: A spatial mechanism for fast planar, 2-dof, pick-and-place applications. In *Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, page 10.
- [Bennehar, 2015] Bennehar, M. (2015). *Some contributions to nonlinear adaptive control of PKMs: from design to real-time experiments*. PhD thesis, Université de Montpellier II.
- [Bennehar et al., 2018] Bennehar, M., Chemori, A., Bouri, M., Jenni, L. F., and Pierrot, F. (2018). A new rise-based adaptive control of pkms: design, stability analysis and experiments. *International Journal of Control*, 91(3):593–607.
- [Bennehar et al., 2016a] Bennehar, M., Chemori, A., Krut, S., and Pierrot, F. (2016a). Adaptive control of parallel manipulators: Design and real-time experiments.
- [Bennehar et al., 2014a] Bennehar, M., Chemori, A., and Pierrot, F. (2014a). A new extension of desired compensation adaptive control and its real-time application to redundantly actuated pkms. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1670–1675. IEEE.
- [Bennehar et al., 2014b] Bennehar, M., Chemori, A., and Pierrot, F. (2014b). A novel rise-based adaptive feedforward controller for redundantly actuated parallel manipulators. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2389–2394. IEEE.
- [Bennehar et al., 2015a] Bennehar, M., Chemori, A., and Pierrot, F. (2015a). L1 adaptive control of parallel kinematic manipulators: Design and real-time experiments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1587–1592. IEEE.
- [Bennehar et al., 2016b] Bennehar, M., Chemori, A., and Pierrot, F. (2016b). A new revised desired compensation adaptive control for enhanced tracking: application to ra-pkms. *Advanced Robotics*, 30(17-18):1199–1214.
- [Bennehar et al., 2015b] Bennehar, M., Chemori, A., Pierrot, F., and Creuze, V. (2015b). Extended model-based feedforward compensation in l1 adaptive control for mechanical manipulators: Design and experiments. *Frontiers in Robotics and AI*, 2:32.
- [Bennehar et al., 2017] Bennehar, M., El-Ghazaly, G., Chemori, A., and Pierrot, F. (2017). A novel adaptive terminal sliding mode control for parallel manipulators: Design and real-time experiments. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6086–6092. IEEE.
- [Birglen, 2003] Birglen, L. (2003). Haptic devices based on parallel mechanisms. state of the art. <http://www.parallemic.org/Reviews/Review003.html>.
- [Bonev, 2003] Bonev, I. (2003). The true origins of parallel robots. <http://www.parallemic.org/Reviews/Review007.html>.



- [Borchert et al., 2015] Borchert, G., Battistelli, M., Runge, G., and Raatz, A. (2015). Analysis of the mass distribution of a functionally extended delta robot. *Robotics and Computer-Integrated Manufacturing*, 31:111–120.
- [Briot et al., 2007] Briot, S., Baradat, C., Arakelian, V., et al. (2007). Contribution to the mechanical behavior improvement of the robotic navigation device surgiscope®. In *ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 653–661. American Society of Mechanical Engineers Digital Collection.
- [Briot and Khalil, 2015] Briot, S. and Khalil, W. (2015). *Dynamics of Parallel Robots: From Rigid Bodies to Flexible Elements*, volume 35. Springer.
- [Brown and Harris, 1994] Brown, M. and Harris, C. J. (1994). Neurofuzzy adaptive modelling and control.
- [Campbell et al., 2011] Campbell, T., Williams, C., Ivanova, O., and Garrett, B. (2011). Could 3d printing change the world. *Technologies, Potential, and Implications of Additive Manufacturing*, Atlantic Council, Washington, DC, 3.
- [Cao and Hovakimyan, 2006a] Cao, C. and Hovakimyan, N. (2006a). Design and analysis of a novel l1 adaptive controller, part i: control signal and asymptotic stability. In *2006 American Control Conference*, pages 3397–3402. IEEE.
- [Cao and Hovakimyan, 2006b] Cao, C. and Hovakimyan, N. (2006b). Design and analysis of a novel l1 adaptive controller, part ii: Guaranteed transient performance. In *2006 American Control Conference*, pages 3403–3408. IEEE.
- [Cappel, 1971] Cappel, K. (1971). Motion simulator. US Patent RE27,051.
- [Castañeda et al., 2014] Castañeda, L. A., Luviano-Juárez, A., and Chairez, I. (2014). Robust trajectory tracking of a delta robot through adaptive active disturbance rejection control. *IEEE Transactions on control systems technology*, 23(4):1387–1398.
- [Chablat and Wenger, 2007] Chablat, D. and Wenger, P. (2007). A new three-dof parallel mechanism: milling machine applications. *arXiv*.
- [Chemori, 2017] Chemori, A. (2017). Control of complex robotic systems: Challenges, design and experiments. In *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 622–631. IEEE.
- [Chen et al., 2014] Chen, X., Liu, X.-J., Xie, F., and Sun, T. (2014). A comparison study on motion/force transmissibility of two typical 3-dof parallel manipulators: the sprint z3 and a3 tool heads. *International Journal of Advanced Robotic Systems*, 11(1):5.
- [Cheng et al., 2003] Cheng, H., Yiu, Y.-K., and Li, Z. (2003). Dynamics and control of redundantly actuated parallel manipulators. *IEEE/ASME Transactions on mechatronics*, 8(4):483–491.

- [Clavel, 1990] Clavel, R. (1990). Device for the movement and positioning of an element in space. US Patent 4,976,582.
- [Clavel, 1991] Clavel, R. (1991). Conception d'un robot parallèle rapide à 4 degrés de liberté. Technical report, EPFL.
- [Codourey, 1996] Codourey, A. (1996). Dynamic modelling and mass matrix evaluation of the delta parallel robot for axes decoupling control. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96*, volume 3, pages 1211–1218. IEEE.
- [Codourey, 1998] Codourey, A. (1998). Dynamic modeling of parallel robots for computed-torque control implementation. *The International Journal of Robotics Research*, 17(12):1325–1336.
- [Company and Pierrot, 1999] Company, O. and Pierrot, F. (1999). A new 3t-1r parallel robot. In *Proc. of IEEE ICAR'99: 9th International Conference on Advanced Robotics*, pages 557–562, Tokyo, Japan.
- [Corbel et al., 2010] Corbel, D., Gouttefarde, M., Company, O., and Pierrot, F. (2010). Towards 100g with pkm. is actuation redundancy a good solution for pick-and-place? In *2010 IEEE International Conference on Robotics and Automation*, pages 4675–4682. IEEE.
- [Craig et al., 1987] Craig, J. J., Hsu, P., and Sastry, S. S. (1987). Adaptive control of mechanical manipulators. *The International Journal of Robotics Research*, 6(2):16–28.
- [Deng et al., 2010] Deng, H., Srinivasan, D., and Oruganti, R. (2010). A b-spline network based neural controller for power electronic applications. *Neurocomputing*, 73(4):593–601.
- [dos Santos Coelho and Pessoa, 2009] dos Santos Coelho, L. and Pessoa, M. W. (2009). Non-linear identification using a b-spline neural network and chaotic immune approaches. *Mechanical Systems and Signal Processing*, 23(8):2418–2434.
- [EcoRobotix, 2011] EcoRobotix (2011). Ara switch to smart scouting. <https://www.ecorobotix.com/en/autonomous-scouting-robot/>.
- [Feng et al., 2002] Feng, Y., Yu, X., and Man, Z. (2002). Non-singular terminal sliding mode control of rigid manipulators. *Automatica*, 38(12):2159–2167.
- [ForceDimension, 2010] ForceDimension (2010). sigma.7 haptic device. <https://www.forcedimension.com/products/sigma-7/overview>.
- [Gao, 2006] Gao, Z. (2006). Active disturbance rejection control: a paradigm shift in feedback control system design. In *2006 American control conference*, pages 7–pp. IEEE.
- [Geng et al., 1992] Geng, Z., Haynes, L. S., Lee, J. D., and Carroll, R. L. (1992). On the dynamic model and kinematic analysis of a class of stewart platforms. *Robotics and autonomous systems*, 9(4):237–254.

- [Germain, 2013] Germain, C. (2013). *Conception d'un robot parallèle à deux degrés de liberté pour des opérations de prise et de dépose*. PhD thesis, L'Université Nantes Angers Le Mans.
- [Germain et al., 2011] Germain, C., Briot, S., Glazunov, V., Caro, S., Wenger, P., et al. (2011). Irsbot-2: A novel two-dof parallel robot for high-speed operations. In *ASME 2011 international design engineering technical conferences and computers and information in engineering conference*, pages 899–909. American Society of Mechanical Engineers Digital Collection.
- [Gough and Whitehall, 1962] Gough, V. and Whitehall, S. (1962). Universal tyre test machine. In *Proc. FISITA 9th Int. Technical Congress*, pages 117–137.
- [Graupe, 2013] Graupe, D. (2013). *Principles of artificial neural networks*, volume 7. World Scientific.
- [Gwinnett, 1931] Gwinnett, J. E. (1931). Amusement device. US Patent 1,789,680.
- [Harib and Srinivasan, 2003] Harib, K. and Srinivasan, K. (2003). Kinematic and dynamic analysis of stewart platform-based machine tool structures. *Robotica*, 21(5):541.
- [Honegger et al., 1997] Honegger, M., Codourey, A., and Burdet, E. (1997). Adaptive control of the hexaglide, a 6 dof parallel manipulator. In *Proceedings of International Conference on Robotics and Automation*, volume 1, pages 543–548. IEEE.
- [Hovakimyan and Cao, 2010] Hovakimyan, N. and Cao, C. (2010). *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. SIAM.
- [Hufnagel and Muller, 2012] Hufnagel, T. and Muller, A. (2012). A projection method for the elimination of contradicting decentralized control forces in redundantly actuated pkm. *IEEE transactions on robotics*, 28(3):723–728.
- [Jain et al., 1996] Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3):31–44.
- [Jiang et al., 2017] Jiang, Y., Yang, C., Na, J., Li, G., Li, Y., and Zhong, J. (2017). A brief review of neural networks based learning and control and their applications for robots. *Complexity*, 2017.
- [Kelly et al., 2006] Kelly, R., Davila, V. S., and Perez, J. A. L. (2006). *Control of robot manipulators in joint space*. Springer Science & Business Media.
- [Krut, 2003] Krut, S. (2003). *Contribution à l'étude des robots parallèles légers, 3T-1R et 3T-2R, à forts débattements angulaires*. PhD thesis, Montpellier 2.
- [Krut et al., 2003] Krut, S., Benoit, M., Ota, H., and Pierrot, F. (2003). I4: A new parallel mechanism for scara motions. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pages 1875–1880. IEEE.

- [Kumar and Bandyopadhyay, 2012] Kumar, P. R. and Bandyopadhyay, B. (2012). Stabilization of stewart platform using higher order sliding mode control. In *2012 7th International Conference on Electrical and Computer Engineering*, pages 945–948. IEEE.
- [Li and Xu, 2005] Li, Y. and Xu, Q. (2005). Kinematics and inverse dynamics analysis for a general 3-prs spatial parallel mechanism. *Robotica*, 23(2):219–229.
- [Li and Xu, 2007] Li, Y. and Xu, Q. (2007). Design and development of a medical parallel robot for cardiopulmonary resuscitation. *IEEE/ASME transactions on Mechatronics*, 12(3):265–273.
- [Lopes et al., 2000] Lopes, J., Ruano, A., and Fleming, P. (2000). Identification of aircraft gas-turbine dynamics using b-splines neural networks. *IFAC Proceedings Volumes*, 33(6):123–128.
- [Lou et al., 2011] Lou, Y., Li, Z., Zhong, Y., Li, J., and Li, Z. (2011). Dynamics and contouring control of a 3-dof parallel kinematics machine. *Mechatronics*, 21(1):215–226.
- [Merlet, 2006] Merlet, J.-P. (2006). *Parallel robots*, volume 128. Springer Science & Business Media.
- [Mirea, 2014] Mirea, L. (2014). Dynamic multivariate b-spline neural network design using orthogonal least squares algorithm for non-linear system identification. In *System Theory, Control and Computing (ICSTCC), 2014 18th International Conference*, pages 720–725. IEEE.
- [Mueller, 2011] Mueller, A. (2011). Robust modeling and control issues of parallel manipulators with actuation redundancy. *Recent Advances in Robust Control: Theory and Applications in Robotics and Electromechanics*, page 207.
- [Müller and Hufnagel, 2012] Müller, A. and Hufnagel, T. (2012). Model-based control of redundantly actuated parallel manipulators in redundant coordinates. *Robotics and Autonomous systems*, 60(4):563–571.
- [Nabat, 2007] Nabat, V. (2007). *Robots parallèles à nacelle articulée, du concept à la solution industrielle pour le pick-and-place*. PhD thesis, Université de Montpellier II.
- [Nabat et al., 2005] Nabat, V., de la O RODRIGUEZ, M., Company, O., Krut, S., and Pierrot, F. (2005). Par4: very high speed parallel robot for pick-and-place. In *2005 IEEE/RSJ International conference on intelligent robots and systems*, pages 553–558. IEEE.
- [Natal et al., 2015] Natal, G. S., Chemori, A., and Pierrot, F. (2015). Dual-space control of extremely fast parallel manipulators: Payload changes and the 100g experiment. *IEEE Transactions on Control Systems Technology*, 4(23):1520–1535.
- [Natal et al., 2016] Natal, G. S., Chemori, A., and Pierrot, F. (2016). Nonlinear control of parallel manipulators for very high accelerations without velocity measurement: stability analysis and experiments on par2 parallel manipulator. *Robotica*, 34(1):43–70.

- [Natal et al., 2009] Natal, G. S., Chemori, A., Pierrot, F., and Company, O. (2009). Nonlinear dual mode adaptive control of par2: a 2-dof planar parallel manipulator, with real-time experiments. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2114–2119. IEEE.
- [Neumann, 2006] Neumann, K. (2006). Exechon concept. *Parallel Kinematic Machines in Research and Practice*, 33:787–802.
- [Ouyang et al., 2002] Ouyang, P. R., Zhang, W. J., and Wu, F. X. (2002). Nonlinear pd control for trajectory tracking with consideration of the design for control methodology. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 4, pages 4126–4131 vol.4.
- [Ovchinnikov et al., 2020] Ovchinnikov, A., Bocharnikova, O., Vorobyeva, N., Dyashkin, A., Bocharnikov, V., and Fomin, S. (2020). Kinematic study of a robot-weeder with a sprayer function and fertigation. In *IOP Conference Series: Earth and Environmental Science*, volume 422. IOP Publishing.
- [Paden and Panja, 1988] Paden, B. and Panja, R. (1988). Globally asymptotically stable ‘pd+’controller for robot manipulators. *International Journal of Control*, 47(6):1697–1712.
- [Park and Kim, 1999] Park, F. and Kim, J. W. (1999). Singularity analysis of closed kinematic chains. *Journal of mechanical design*, 121(1):32–38.
- [Park et al., 2013] Park, S. B., Kim, H. S., Song, C., and Kim, K. (2013). Dynamics modeling of a delta-type parallel robot (isr 2013). In *IEEE ISR 2013*, pages 1–5. IEEE.
- [Patre et al., 2006] Patre, P. M., MacKunis, W., Makkar, C., and Dixon, W. E. (2006). Asymptotic tracking for systems with structured and unstructured uncertainties. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 441–446. IEEE.
- [Pierrot, 1991] Pierrot, F. (1991). *Robots pleinement parallèles légers: conception, modélisation et commande*. PhD thesis, Université Montpellier 2.
- [Pierrot et al., 2009] Pierrot, F., Baradat, C., Nabat, V., Company, O., Krut, S., and Gouttefarde, M. (2009). Above 40g acceleration for pick-and-place with a new 2-dof pkm. In *2009 IEEE International Conference on Robotics and Automation*, pages 1794–1800. IEEE.
- [Pierrot et al., 1991] Pierrot, F., Dauchez, P., and Fournier, A. (1991). Fast parallel robots. *Journal of Robotic Systems*, 8(6):829–840.
- [Pierrot and Shibukawa, 1999] Pierrot, F. and Shibukawa, T. (1999). From hexa to hexam. In *Parallel Kinematic Machines*, pages 357–364. Springer.
- [Plestan et al., 2010] Plestan, F., Shtessel, Y., Bregeault, V., and Poznyak, A. (2010). New methodologies for adaptive sliding mode control. *International journal of control*, 83(9):1907–1919.

- [Pollard, 1940] Pollard, W. (1940). Spray painting machine: Usa. *US2213108*.
- [Razmi and Macnab, 2020] Razmi, M. and Macnab, C. (2020). Near-optimal neural-network robot control with adaptive gravity compensation. *Neurocomputing*.
- [Ren et al., 2007] Ren, L., Mills, J. K., and Sun, D. (2007). Experimental comparison of control approaches on trajectory tracking control of a 3-dof parallel robot. *IEEE Transactions on Control Systems Technology*, 15(5):982–988.
- [Rosales-Díaz et al., 2019] Rosales-Díaz, I., López-Gutiérrez, J. R., Suárez, A. E. Z., Salazar, S. R., Osorio-Cordero, A., Aguilar-Sierra, H., and Lozano, R. (2019). Comparison of control techniques in a weight lifting exoskeleton. *Journal of Bionic Engineering*, 16(4):663–673.
- [Sabater et al., 2007] Sabater, J. M., Aracil, R., Saltaren, R. J., and Payá, L. (2007). A novel parallel haptic interface for telerobotic systems. In *Advances in Telerobotics*, pages 45–59. Springer.
- [Sadegh and Horowitz, 1990] Sadegh, N. and Horowitz, R. (1990). Stability and robustness analysis of a class of adaptive controllers for robotic manipulators. *The International Journal of Robotics Research*, 9(3):74–92.
- [Saied, 2019] Saied, H. (2019). *On Control of Parallel Robots for High Dynamic Performances: From Design to Experiments*. PhD thesis, Université de Montpellier.
- [Saied et al., 2019a] Saied, H., Chemori, A., Bouri, M., El Rafei, M., Francis, C., and Pierrot, F. (2019a). A new time-varying feedback rise control for second-order nonlinear mimo systems: theory and experiments. *International Journal of Control*, pages 1–14.
- [Saied et al., 2019b] Saied, H., Chemori, A., El Rafei, M., Francis, C., and Pierrot, F. (2019b). From non-model-based to model-based control of pkms: a comparative study. In *Mechanism, Machine, Robotics and Mechatronics Sciences*, pages 153–169. Springer.
- [Santibañez and Kelly, 2001] Santibañez, V. and Kelly, R. (2001). Pd control with feedforward compensation for robot manipulators: analysis and experimentation. *Robotica*, 19(1):11.
- [Sartori-Natal et al., 2012] Sartori-Natal, G., Chemori, A., Michelin, M., and Pierrot, F. (2012). A dual-space feedforward pid control of redundantly actuated parallel manipulators with real-time experiments.
- [Schindele and Aschemann, 2010] Schindele, D. and Aschemann, H. (2010). Trajectory tracking of a pneumatically driven parallel robot using higher-order smc. In *2010 15th International Conference on Methods and Models in Automation and Robotics*, pages 387–392. IEEE.
- [Seraji, 1998] Seraji, H. (1998). A new class of nonlinear pid controllers with robotic applications. *Journal of Robotic Systems*, 15(3):161–181.
- [Shang et al., 2012] Shang, W.-W., Cong, S., and Ge, Y. (2012). Adaptive computed torque control for a parallel manipulator with redundant actuation. *Robotica*, 30(3):457.

- [Sharma et al., 2012] Sharma, V., Rai, S., and Dev, A. (2012). A comprehensive study of artificial neural networks. *International Journal of Advanced research in computer science and software engineering*, 2(10).
- [Shayya, 2015] Shayya, S. (2015). *Towards Rapid and Precise Parallel Kinematic Machines*. PhD thesis, Ph. D. thesis, University of Montpellier, Montpellier.
- [Siciliano et al., 2010] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2010). *Robotics: modelling, planning and control*. Springer Science & Business Media.
- [Simaan, 1999] Simaan, N. (1999). *Analysis and synthesis of parallel robots for medical applications*. PhD thesis, Technion-Israel Institute of Technology, Faculty of Mechanical Engineering.
- [Staicu, 2019] Staicu, S. (2019). *Dynamics of Parallel Robots*. Springer.
- [Stewart, 1965] Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the institution of mechanical engineers*, 180(1):371–386.
- [Su et al., 2004] Su, Y. X., Duan, B. Y., and Zheng, C. H. (2004). Nonlinear pid control of a six-dof parallel manipulator. *IEE Proceedings - Control Theory and Applications*, 151(1):95–102.
- [Taghirad, 2013] Taghirad, H. D. (2013). *Parallel robots: mechanics and control*. CRC press.
- [Tsai, 1999] Tsai, L.-W. (1999). *Robot analysis: the mechanics of serial and parallel manipulators*. John Wiley & Sons.
- [Utkin and Lee, 2006] Utkin, V. and Lee, H. (2006). Chattering problem in sliding mode control systems. In *International Workshop on Variable Structure Systems, 2006. VSS'06.*, pages 346–350. IEEE.
- [Venkataraman and Gulati, 1992] Venkataraman, S. T. and Gulati, S. (1992). Control of nonlinear systems using terminal sliding modes. In *1992 American Control Conference*, pages 891–893.
- [Wapler et al., 2003] Wapler, M., Urban, V., Weisener, T., Stallkamp, J., Dürr, M., and Hiller, A. (2003). A stewart platform for precision surgery. *Transactions of the Institute of Measurement and Control*, 25(4):329–334.
- [Williams II, 2016] Williams II, R. L. (2016). The delta parallel robot: Kinematics solutions.
- [Xian et al., 2004] Xian, B., Dawson, D. M., de Queiroz, M. S., and Chen, J. (2004). A continuous asymptotic tracking control strategy for uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 49(7):1206–1211.
- [Zhang, 2009] Zhang, D. (2009). *Parallel robotic machine tools*. Springer Science & Business Media.

- [Zhao and Gao, 2009] Zhao, Y. and Gao, F. (2009). Inverse dynamics of the 6-dof out-parallel manipulator by means of the principle of virtual work. *Robotica*, 27(2):259–268.
- [Zhao et al., 2005] Zhao, Y., Yang, Z., and Huang, T. (2005). Inverse dynamics of delta robot based on the principle of virtual work. *Transactions of Tianjin University*, 11(4):268–273.
- [Zhuang and Wang, 1997] Zhuang, H. and Wang, Y. (1997). A coordinate measuring machine with parallel mechanisms. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 3256–3261. IEEE.
- [Ziegler et al., 1942] Ziegler, J. G., Nichols, N. B., et al. (1942). Optimum settings for automatic controllers. *trans. ASME*, 64(11).
- [Zou et al., 2008] Zou, J., Han, Y., and So, S.-S. (2008). Overview of artificial neural networks. In *Artificial Neural Networks*, pages 14–22. Springer.