# Control of Parallel Robots For High-Speed Pick-And-Throw Tasks In Selective Sorting

Ghina Hassan

# THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

**En Systèmes Automatiques et Microélectroniques**

**École doctorale Information, Structures et Systèmes (I2S)**

**Unité de recherche UMR 5506**

**En partenariat international avec Université Libanaise, LIBAN**

# Commande d'un Robot Parallèle pour des Tâches de Pick-and-Throw à Haute Cadence en Tri Sélectif


**Présentée par Ghina HASSAN**


**Le 12 Décembre 2022**


**Sous la direction de Marc GOUTTEFARDE, Clovis FRANCIS et Ahmed CHEMORI, Maher EL RAFEI**


**Devant le jury composé de**

| | |
|---|---|
| Stéphane CARO, Directeur de recherche, CNRS, LS2N | Rapporteur |
| Jacques GANGLOFF, Professeur des universités, Université de Strasbourg, ICube | Rapporteur |
| Estelle COURTIAL, Maître de conférence, Université d'Orléans, PRISME | Examinatrice |
| Naseem DAHER, Maître de conférence, Université Américaine de Beyrouth | Examinateur |
| Damien SALLE, Coordinator at TECNALIA Research & Innovation, Espagne | Invité |
| Marc GOUTTEFARDE, Directeur de recherche, CNRS, LIRMM | Directeur de thèse |
| Clovis FRANCIS, Professeur des universités, Université Libanaise, CRSI | Directeur de thèse |
| Ahmed CHEMORI, Chargé de recherche, CNRS, LIRMM | Encadrant de thèse |
| Maher EL RAFEI, Maître de conférence, Université Libanaise, CRSI | Encadrant de thèse |

UNIVERSITÉ DE MONTPELLIER

# Contents

**Contents** i

**List of Figures** 1

**List of Tables** 7

**Acknowledgements** 9

**General introduction** 11

**1 Context, problem formulation and state of the art** 15
- 1.1 Introduction . . . 15
- 1.2 Classification of robotic manipulators . . . 16
  - 1.2.1 Serial manipulators . . . 17
  - 1.2.2 Parallel manipulators . . . 18
  - 1.2.3 Parallel versus serial manipulators . . . 18
  - 1.2.4 Cable-driven parallel robots . . . 20
  - 1.2.5 Hybrid manipulators . . . 22
- 1.3 A historical overview of parallel manipulators . . . 23
- 1.4 Main applications of parallel manipulators . . . 31
  - 1.4.1 High speed Pick-and-Place tasks . . . 31
  - 1.4.2 Machining applications . . . 32
  - 1.4.3 Motion simulators . . . 32
  - 1.4.4 Medical applications . . . 35
  - 1.4.5 Agricultural applications . . . 35

1.4.6 3D printers . . . 35
1.4.7 Haptic devices . . . 37
1.5 Overview on Waste sorting . . . 38
1.6 Trajectory generation and control Problem formulation . . . 40
1.6.1 Main challenges in trajectory generation for PKMs . . . 40
1.6.1.1 Path constraints . . . 41
1.6.1.2 Continuity constraints . . . 41
1.6.1.3 Dynamic constraints . . . 41
1.6.1.4 Optimal trajectories . . . 41
1.6.2 Main challenges in control of PKMs . . . 42
1.6.2.1 Nonlinear complex dynamics . . . 42
1.6.2.2 Structured and unstructured uncertainties . . . 42
1.6.2.3 Actuation redundancy . . . 43
1.7 Dynamic modeling of parallel manipulators . . . 43
1.7.1 Dynamic modeling representation . . . 45
1.7.2 Properties of the dynamic model . . . 47
1.7.2.1 Property of mass and inertia matrix . . . 47
1.7.2.2 Property of Coriolis and centrifugal matrix . . . 47
1.7.2.3 Property of gravity vector . . . 48
1.7.2.4 Property of the disturbance term . . . 48
1.7.2.5 Linear formulation property of the dynamics . . . 49
1.8 Overview on motion control strategies for parallel robots . . . 49
1.8.1 Kinematic control strategies . . . 51
1.8.1.1 PD/PID control . . . 51
1.8.1.2 Nonlinear PD/PID control . . . 52
1.8.1.3 $\mathscr{L}_1$ adaptive control . . . 52
1.8.1.4 Strategies based on artificial neural networks . . . 53
1.8.2 Dynamic control strategies . . . 54
1.8.2.1 Computed torque control . . . 54
1.8.2.2 Augmented PD control . . . 55
1.8.2.3 PD control with computed feedforward . . . 56
1.8.2.4 Strategies with time-varying feedback gains . . . 56
1.8.2.5 Dynamic adaptive control . . . 57
1.8.2.6 Sliding mode control . . . 60
1.9 Overview on Pick-and-Throw motion generation . . . 61
1.10 Objectives of the thesis . . . 62
1.11 Main contributions of the thesis . . . 63
1.12 Conclusion . . . 65

**2 Proposed time-optimal Pick-and-Throw trajectory 67**
2.1 Introduction . . . 67

2.2 Minimum-time S-curve trajectory . . . 68
2.2.1 Polynomial S-curve motion profile . . . 69
2.2.2 Minimization problem formulation . . . 70
2.2.3 Algorithm to compute the minimum-time S-curve trajectory . . . 71
2.2.4 Time T and velocity V as functions of displacement P . . . 72
2.3 Minimum-time throw motion . . . 74
2.3.1 Problem formulation . . . 74
2.3.2 Recall of mathematical modeling of ballistic motion . . . 76
2.3.3 Throw motion geometric path and S-curve motion profile . . . 77
2.3.4 Determination of the optimal release configuration . . . 78
2.3.4.1 Bound constraints . . . 79
2.3.4.2 Nonlinear inequality constraints . . . 79
2.3.4.3 Nonlinear equality constraint . . . 80
2.3.5 Optimization problem simplification . . . 80
2.4 T3KR robot: A 5-DOFs PKM . . . 83
2.4.1 Description of T3KR PKM . . . 84
2.4.2 Kinematics of T3KR PKM . . . 84
2.4.2.1 Inverse Kinematic Model of T3KR PKM . . . 88
2.4.2.2 Forward Kinematic Model of T3KR PKM . . . 89
2.4.3 Differential Kinematics of T3KR PKM . . . 90
2.4.4 Dynamics of T3KR PKM . . . 91
2.5 Real-time experimental results . . . 94
2.5.1 Description of the experimental testbed . . . 94
2.5.2 Description of P&P and P&T reference trajectories . . . 95
2.5.2.1 Pick-and-Place reference trajectory . . . 95
2.5.2.2 Existing Pick-and-Throw reference trajectory . . . 96
2.5.2.3 Proposed Pick-and-Throw reference trajectory . . . 96
2.5.3 Obtained Experimental Results . . . 97
2.5.3.1 Scenario 1 . . . 97
2.5.3.2 Scenario 2 . . . 98
2.5.3.3 Scenario 3 . . . 98
2.5.4 Results discussion . . . 99
2.6 Conclusion . . . 102

**3 Proposed control solutions for Pick-&-Throw tasks with PKMs 109**
3.1 Introduction . . . 109
3.2 A new augmented RISE feedback controller . . . 110
3.2.1 Background on RISE feedback control . . . 110
3.2.2 RISE control of parallel manipulators . . . 111
3.2.3 Applications of RISE feedback control law . . . 113
3.2.4 Proposed augmented RISE feedback control . . . 114

3.2.4.1 Motivation . . . 114
3.2.4.2 Control design . . . 115
3.2.4.3 Closed-loop error dynamics . . . 117
3.2.4.4 Stability analysis . . . 118
3.3 A novel robust DCAL with adaptive feedback gains . . . 121
3.3.1 General overview of DCAL control strategy . . . 121
3.3.2 Proposed robust DCAL with adaptive feedback gains . . . 122
3.3.2.1 Motivation . . . 122
3.3.2.2 Control design . . . 123
3.3.2.3 Stability analysis . . . 124
3.4 A new intelligent robust control law . . . 128
3.4.1 Background on Model-Free Control . . . 128
3.4.2 Applications of MFC scheme . . . 129
3.4.3 Proposed intelligent robust control . . . 130
3.4.3.1 Control design . . . 130
3.4.3.2 Closed-loop error dynamics . . . 131
3.4.3.3 Stability analysis . . . 132
3.5 Conclusion . . . 134

**4 Proposed control solution numerical simulation results 137**
4.1 Introduction . . . 137
4.2 Generation of pick-and-throw reference trajectories . . . 138
4.3 Performance evaluation criteria . . . 139
4.3.1 Tuning of the feedback control gains . . . 139
4.4 Proposed numerical simulation scenarios . . . 140
4.4.1 Scenario 1 - Robustness towards payload variations . . . 140
4.4.2 Scenario 2 - Robustness towards speed variations . . . 140
4.5 Numerical simulation results of extended RISE control plus compensation . 141
4.5.1 Tuning procedure of the control gains . . . 141
4.5.2 Scenario 1 - Robustness towards payload changes . . . 141
4.5.3 Scenario 2 - Robustness towards speed variations . . . 142
4.6 Numerical simulation results of robust DCAL with adaptive feedback gains . 146
4.6.1 Tuning procedure of the control gains . . . 146
4.6.2 Scenario 1 - Robustness towards payload changes . . . 147
4.6.3 Scenario 2 - Robustness towards speed variations . . . 148
4.7 Numerical simulation results of intelligent robust control . . . 151
4.7.1 State-of-the-art control methods for comparison purposes . . . 152
4.7.2 Tuning procedure of the control gains . . . 153
4.7.3 Scenario 1 - Robustness towards payload changes . . . 153
4.7.4 Scenario 2 - Robustness towards speed variations . . . 154
4.8 Comparison between the proposed controllers . . . 156

4.8.1 Scenario 1 - Robustness towards payload changes . . . 156
4.8.2 Scenario 2 - Robustness towards speed variations . . . 158
4.9 Conclusion . . . 161

**General conclusion** **163**

**Bibliography** **167**

**Appendices** **185**

**A Proof of Algorithm 1** **185**
A.1 Preliminaries: Expressions of the gradients . . . 186
A.2 Case 1: No active constraint . . . 187
A.3 Case 2: One active constraint $C_1(x) = 0$ . . . 187
A.4 Case 3: One active constraint $C_2(x) = 0$ . . . 188
A.5 Case 4: One active constraint $C_3(x) = 0$ . . . 189
A.6 Case 5: One active constraint $C_4(x) = 0$ . . . 190
A.7 Case 6: Two active constraints $C_1(x) = C_2(x) = 0$ . . . 190
A.8 Case 7: Two active constraints $C_1(x) = C_3(x) = 0$ . . . 191
A.9 Case 8: Two active constraints $C_1(x) = C_4(x) = 0$ . . . 191
A.10 Case 9: Two active constraints $C_2(x) = C_3(x) = 0$ . . . 192
A.11 Case 10: Two active constraints $C_2(x) = C_4(x) = 0$ . . . 193
A.12 Case 11: Two active constraints $C_3(x) = C_4(x) = 0$ . . . 195
A.13 Synthesis of the 11 cases and Algorithm 2 . . . 196
A.14 Particular cases . . . 198
A.15 Second-Order Sufficient Conditions . . . 205
A.16 LICQ . . . 209

**B Total time $T$ and velocity $V$ as a functions of displacement $P$** **211**
B.1 Total time $T$ as a function of displacement $P$ . . . 211
B.1.1 Case 1: $\sqrt{JV_{max}} < A_{max}$ . . . 212
B.1.2 Case 2: $\sqrt{JV_{max}} \geq A_{max}$ . . . 213
B.2 Velocity $V$ as a function of displacement $P$ . . . 217
B.2.1 Case 1: $V(P)$ for $\sqrt{JV_{max}} < A_{max}$ . . . 217
B.2.2 Case 2: $V(P)$ for $\sqrt{JV_{max}} \geq A_{max}$ . . . 217

# List of Figures


1.1 Illustration of a serial manipulator with its kinematic configuration. . . . . 17
1.2 Exemplification of a parallel manipulator with its kinematic configuration. . . . 18
1.3 Illustration of the CoGiRo (Control of Giant Robots) prototype built in the framework of a collaborative research project between TECNALIA and LIRMM. . 21
1.4 Sprint Z3 hybrid manipulator with a serial carrier and a parallel wrist. . . . . 22
1.5 Tricept hybrid manipulator with a parallel carrier and a serial wrist. . . . . 23
1.6 Dumbo hybrid manipulator. . . . . 24
1.7 The spherical parallel mechanism proposed by James E. Gwinnett [Gwinnett, 1931]. . . . . 24
1.8 Illustration of the spray painting 5-DOF PKM proposed by Willard L.V. Pollard [Pollard, 1942]. . . . . 25
1.9 View of the original Gough platform in 1954 [Bonev, 2003]. . . . . 25
1.10 Gough tyre-testing prototype, called universal rig, exhibited in the British National Museum of Science and Industry in 2000 [Bonev, 2003]. . . . . 26
1.11 Illustration of the Stewart platform proposed as a flight simulator [Stewart, 1965]. 27
1.12 View of the first flight simulator of Klaus Cappel [Cappel, 1967]. . . . . 27
1.13 Technical drawing of the original Delta robot of Reymond Clavel [Clavel, 1990]. . 28
1.14 Timeline of some PKMs based on the Delta robot. . . . . 29
1.15 ABB's IRB 360 FlexPicker . . . . . 31
1.16 The high-speed Omron Adept Quattro robot . . . . . 31
1.17 Illustration of three examples of PKMs for machining operations. . . . . 33
1.18 View of two examples of flight simulators. . . . . 34
1.19 View of two MISTRAL hexapods swell simulators developed by SYMETRIE. . . . 34
1.20 Illustration of some examples of PKMs for medical applications. . . . . 36
1.21 View of ecoRobotix smart weeding robot. . . . . 36

1.22 View of two examples of 3D printers based on PKMs. . . . 37
1.23 Illustration of two examples of PKM-based haptic devices. . . . 38
1.24 SamurAI is a self-aware sorting robot, presented by "Machinex", that employs superior artificial intelligence technology to identify materials for a positive product recovery. . . . 39
1.25 A dual-arm robot uses pick-and-throw in selective waste sorting. This robot was built by "Bulk Handling Systems" to make recycling more efficient. . . . 39
1.26 Illustration of actuation redundancy in PKM through additional kinematic chain. 44
1.27 Illustration of the loop constraints for a planar 2-DOF redundantly actuated PKM [Mueller, 2011] . . . 45
1.28 Classification of the main proposed control solutions for parallel manipulators in the literature. . . . 50
1.29 Block diagram of the $\mathscr{L}_1$ adaptive control for PKM. . . . 53
1.30 Block diagram of the standard computed torque controller in joint space. . . . 55
1.31 Block diagram of the general adaptive feedforward control law in joint space. . . . 58
1.32 Block diagram of adaptive computed torque control in joint space. . . . 59

2.1 Third order polynomial S-Curve model. . . . 70
2.2 A typical evolution of the increasing function $T(P)$ when $\sqrt{JV_{max}} < A_{max}$ (left) and when $\sqrt{JV_{max}} \geq A_{max}$ (right). . . . 74
2.3 A typical evolution of the nondecreasing function $V(P)$ when $\sqrt{JV_{max}} < A_{max}$ (left) and when $\sqrt{JV_{max}} > A_{max}$ (right). . . . 74
2.4 Illustration of the throwing trajectory of the end-effector of the robot (Solid line) and the free-flight motion of the object (Dashed line). . . . 75
2.5 Illustration of the pick and throw motion consisting of T3KR robot motion acceleration and deceleration phases together with the ballistic motion of the thrown object. . . . 75
2.6 The ballistic motion constraint and the minimum-time S-curve induced constraint as functions $V_r(P_r)$. For a given $\theta_r$, there exists a unique couple $(P_r^*(\theta_r), V_r^*(\theta_r))$ satisfying both constraints. . . . 81
2.7 In the case $\sqrt{JV_{max}} < A_{max}$, this figure shows the minimum $P_r^m = P_r(\theta_r^m)$ of the function in (2.26) (case where $P_r^m < \sqrt{V_{max}^3/J}$) and the optimal release configuration $P_r^*$, $V_r^*$ and $\theta_r^*$. . . . 82
2.8 A schematic view of T3KR parallel robot including ① : Fixed base, ②: Main actuator, ③ : Rear-arm, ④: Forearms, ⑤ : Revolute active joint, ⑥ : Passive spherical joints, ⑦ : Mobile platform, ⑧ : Integrated actuator on the platform . 85
2.9 T3KR workspace limits in the planes (Y,Z) and (X,Y). . . . 85
2.10 Illustration of the kinematics of the T3KR PKM. . . . 86
2.11 Illustration of dynamic parameters of T3KR robot arm. . . . 92
2.12 The experimental testbed: T3KR robot, conveyors, recyclable objects and the sorting box. . . . 95

2.13 3D-view of P&P reference trajectory in Cartesian space. . . . . 97
2.14 3D-view of existing P&T reference trajectory [Raptopoulos et al., 2020]. . . . . 97
2.15 3D-view of the proposed P&T reference trajectory. . . . . 98
2.16 Scenario 1: Evolution of the desired robot's end-effector Cartesian positions versus time for the P&P task [Nabat et al., 2005] and the proposed P&T task for different values of the maximum acceleration. . . . . 100
2.17 Scenario 1: Evolution of the desired robot's end-effector Cartesian velocities versus time for the P&P task [Nabat et al., 2005] and the proposed P&T task for different values of the maximum acceleration. . . . . 101
2.18 Scenario 1: Evolution of the desired robot's end-effector Cartesian acceleration versus time for the P&P task [Nabat et al., 2005] and the proposed P&T task for different values of the maximum acceleration. . . . . 101
2.19 Scenario 2: Evolution of the desired robot's end-effector Cartesian positions versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration. . . . . 102
2.20 Scenario 2: Evolution of the desired robot's end-effector Cartesian velocities versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration. . . . . 102
2.21 Scenario 2: Evolution of the desired robot's end-effector Cartesian accelerations versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration. . . . . 103
2.22 Scenario 3: Evolution of the desired robot's end-effector Cartesian positions versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration. . . . . 103
2.23 Scenario 3: Evolution of the desired robot's end-effector Cartesian velocities versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration. . . . . 104
2.24 Scenario 3: Evolution of the desired robot's end-effector Cartesian accelerations versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration. . . . . 104
2.25 Evolution versus time of the desired robot's end-effector Cartesian positions for the proposed P&T method outside the robot's workspace at 2.1 m/s of maximum velocity and 4.2 G of maximum acceleration. . . . . 105
2.26 Evolution versus time of the desired robot's end-effector Cartesian velocities for the proposed P&T method outside the robot's workspace at 2.1 m/s of maximum velocity and 4.2 G of maximum acceleration. . . . . 105
2.27 Evolution versus time of the desired robot's end-effector Cartesian accelerations for the proposed P&T method outside the robot's workspace at 2.1 m/s of maximum velocity and 4.2 G of maximum acceleration. . . . . 106

2.28 Evolution versus time of the resulting desired Cartesian velocity and acceleration of the robot's end-effector for the proposed P&T method outside the robot's workspace at 2.1 m/s of maximum velocity and 4.2 G of maximum acceleration. 106
2.29 Evolution versus time of the resulting desired Cartesian velocity and acceleration of the robot's end-effector for the proposed P&T method outside the robot's workspace at 2.1m/s of maximum velocity and 4.2 G of maximum acceleration. 107

3.1 Block diagram of the proposed extended RISE control plus compensation. . . . 116
3.2 Block diagram of the proposed RDCAL AG control scheme. . . . . 125
3.3 Block diagram of the proposed IRC scheme. . . . . 132

4.1 3D-view of the P&T reference trajectories of the robot (the red and green lines) with the ballistic motions of the thrown objects in Cartesian space. . . . . 138
4.2 Scenario 1: Evolution of the Cartesian tracking errors of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot. . 143
4.3 Scenario 1: Evolution of the Cartesian tracking errors of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot within the interval [5.1, 5.2] sec. . . . . 143
4.4 Scenario 1: Evolution of the control input torques of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot. . . . . 144
4.5 Scenario 2: Evolution of the Cartesian tracking errors of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot. . 145
4.6 Scenario 2: Evolution of the Cartesian tracking errors of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot within the interval [2.14, 2.24] sec. . . . . 145
4.7 Scenario 2: Evolution of the control input torques of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot. . . . . 146
4.8 Scenario 1: Evolution of the Cartesian tracking errors of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot. . . . . 148
4.9 Scenario 1: Evolution of the estimated mass of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot. . . . . 149
4.10 Scenario 1: Evolution of the adaptive gains, $\Lambda_p(t)$ and $\Lambda_v(t)$, of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot. . . . . 149
4.11 Scenario 1: Evolution of the control input torques of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot. . . . . 150
4.12 Scenario 2: Evolution of the Cartesian tracking errors of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot. . . . . 151
4.13 Scenario 2: Evolution of the estimated mass of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot. . . . . 151
4.14 Scenario 2: Evolution of the adaptive gains, $\Lambda_p(t)$ and $\Lambda_v(t)$, of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot. . . . . 152

4.15 Scenario 2: Evolution of the control input torques of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot. . . . . 152
4.16 Scenario 1: Evolution of the Cartesian tracking errors of the feedforward RISE control, the feedforward ST-SMC, and the proposed IRC scheme on T3KR robot. 155
4.17 Scenario 1: Evolution of the control input torques of the feedforward RISE control, the feedforward ST-SMC, and the proposed IRC scheme on T3KR robot. . . 155
4.18 Scenario 2: Evolution of the Cartesian tracking errors of the feedforward RISE control, feedforward ST-SMC, and the proposed IRC scheme on T3KR robot. . . 156
4.19 Scenario 2: Evolution of the control input torques of the feedforward RISE control, feedforward ST-SMC, and the proposed IRC scheme on T3KR robot. . . . . 157
4.20 Scenario 1: Evolution of the Cartesian tracking errors of the three proposed controllers: (i) the extended RISE control plus compensation, (ii) the robust DCAL with adaptive gains, and (iii) the intelligent robust control on T3KR robot. . . . . 159
4.21 Scenario 1: Evolution of the control input torques of the three proposed controllers: (i) the extended RISE control plus compensation, (ii) the robust DCAL with adaptive gains, and (iii) the intelligent robust control on T3KR robot. . . . . 159
4.22 Scenario 2: Evolution of the Cartesian tracking errors of the three proposed controllers: (i) the extended RISE control plus compensation, (ii) the robust DCAL with adaptive gains, and (iii) the intelligent robust control on T3KR robot. . . . . 160
4.23 Scenario 2: Evolution of the control input torques of the three proposed controllers: (i) the extended RISE control plus compensation, (ii) the robust DCAL with adaptive gains, and (iii) the intelligent robust control on T3KR robot. . . . . 160

B.1 The increasing function $T(P)$ in Case 1 where $\sqrt{JV_{max}} < A_{max}$. . . . . 212
B.2 The increasing function $T(P)$ in Case 2 where $\sqrt{JV_{max}} \geq A_{max}$. . . . . 216
B.3 The nondecreasing function $V(P)$ in Case 1 ($\sqrt{JV_{max}} < A_{max}$). . . . . 217
B.4 The nondecreasing function $V(P)$ in Case 2 when $\sqrt{JV_{max}} = A_{max}$. . . . . 218
B.5 The nondecreasing function $V(P)$ in Case 2 when $\sqrt{JV_{max}} > A_{max}$. . . . . 218

# List of Tables


1.1 The main properties of serial and parallel manipulators. . . . . 20

2.1 Summary of the main geometric and dynamic parameters of T3KR PKM. . . . . 94
2.2 Number of picks per minutes versus operating acceleration for the three case studies. . . . . 100
2.3 Time of the three phase movements . . . . . 101

4.1 The control gains of the original RISE, PD plus compensation, and the proposed extended RISE plus compensation controllers. . . . . 141
4.2 Control performance evaluation of the original RISE, the PD plus compensation, and the proposed extended RISE plus compensation controllers. . . . . 142
4.3 The control gains of the standard DCAL and the proposed robust DCAL with adaptive gains. . . . . 147
4.4 Control performance evaluation of the standard DCAL and the proposed robust DCAL with adaptive feedback gains. . . . . 147
4.5 The control gains of the feedforward RISE control, the feedforward ST-SMC, and the proposed intelligent robust control. . . . . 153
4.6 Control performance evaluation of the feedforward RISE control, the feedforward ST-SMC, and the proposed intelligent robust control. . . . . 154
4.7 Control performance evaluation of the three proposed controllers: (i) the extended RISE control plus compensation, (ii) the robust DCAL with adaptive gains, and (iii) the intelligent robust control on T3KR robot. . . . . 161

# Acknowledgements

First of all, I would like to express my sincere gratitude to my thesis supervisors, Ahmed CHEMORI and Maher EL RAFEI, for the continuous support and guidance they have given me during the last three years, for their patience, their motivation, and for sharing their knowledge and expertise. I would also like to thank Marc GOUTTEFARDE for having accepted to be my thesis director and for allowing me to join the LIRMM and accomplish this thesis. As well as Clovis FRANCIS, my thesis co-director, for giving me the opportunity to establish my thesis as a joint Ph.D. with the Lebanese University. I am grateful to them for their dedication in advising me and giving me ideas to enrich this work considerably.

I would also like to thank my committee members, Stéphane CARO, Jacques GANGLOFF, Estelle COURTIAL, and Naseem DAHER for accepting to review my thesis work, for their presence, and for their insightful comments and encouragement.

Furthermore, I would like to thank TECNALIA for funding my thesis and giving me all the necessary support. I would like to thank Damien SALLE, the robotics and automation coordinator at TECNALIA, for welcoming me to the company. In addition, I owe many thanks to my colleagues at TECNALIA: Micaël MICHELIN, Lotfi CHIKH, Jean-Baptiste IZARD, and Pierre-Elie HERVE, for their enriching discussions and their encouragement. Thank you Pierre-Elie for the precious help in the experiments and for all the experience you shared with me, I learned a lot from you!

I would also like to thank all my colleagues at LIRMM and friends. Thank you for being so supportive, for spending enjoyable moments inside and outside of work, and for giving me help and advice.

To my closest and sweetest friends from Lebanese university and high school, thank you for your unlimited support and encouragement over the past three years. I have been

lucky to have a wonderful group of friends like you. I would also like to thank my teacher Nour YASSINE, she was not only a teacher but also a sister to me. It is not easy to express my gratitude.

A special thanks to my family. Words cannot express how grateful I am to my parents, my brothers, and my sister. I am forever indebted to my parents for giving me the opportunities and experiences that have made me who I am. To Mom and Dad, thank you for being by my side throughout this challenging journey, I dedicate this milestone to you.

My deep and sincere gratitude to my beloved husband Ahmad who spent sleepless nights with me and was always my support in the moments when there was no one to answer my queries. I feel immensely lucky to have met you and to be able to count on you. I hope we face many more complicated and fun ventures together!

Finally, I would like to thank God for giving me the strength, knowledge, ability, and opportunity to undertake this research study and to persevere and complete it satisfactorily. Without his blessings, this achievement would not have been possible.

# General introduction

Automation have gained a wide interest in the last centuries in various fields such as industries, medicine, household life, agriculture, space, etc. As its name indicates, automation aims to automatically carry out processes to increase productivity, improve product quality, and reduce time and costs. An automatic system uses mechanical, electrical, electronic, hydraulic, and pneumatic elements to carry out a specific task. Robotic manipulators have been one of the major automated machines capable of being used in multiple tasks such as welding, packaging, assembly, and quality control. Nowadays, there are two main types of robot manipulator kinematic architectures: serial manipulators and parallel manipulators.

Serial manipulators consist of an open kinematic chain powered by actuators located on or near the joints. Although they have been mostly used in the last centuries thanks to their large workspace and high dexterity, their architecture is not very suitable for moving heavy payloads at high speeds due to the high vibrations and link bending. In the last 40 years, the interest in parallel robots also known as Parallel Kinematic Manipulators (PKMs) has increased thanks to their special features. They consist of multiple kinematic chains attached to a moving platform also called traveling plate. In contrast to serial manipulators, they offer more stiffness, improved precision, high-speed capabilities, and a higher payload-to-weight ratio. Thanks to these advantages, PKMs are used in applications where precision and high-speed are vital. However, they still suffer from some drawbacks such as limited workspace and complex singularities behavior. In fact, parallel manipulators are not replacing serial ones, but they offer various advantages for certain applications that need high accelerations and high accuracy. They have been used in Pick-and-Place (P&P) tasks, remote surgery, precise positioning, even high-speed machining, and recently in waste sorting industry.

Parallel manipulators are known for their complex nonlinear dynamics, abundance of parametric and nonparametric uncertainties, and in some cases, redundant actuation. All of the aforementioned aspects can be considered as sources of errors (if not taken into account) that may deteriorate the performance of parallel manipulators. Therefore, it is necessary to design advanced control schemes that guarantee optimal trajectory tracking under desired speed requirements and with the best possible accuracy despite the presence of some or all of the above mentioned issues. In addition to control design, trajectory planning is considered a crucial aspect from which the dynamic performance of parallel manipulators can be improved. In the motion planning algorithm, various challenges should be considered, such as the path constraints of avoiding obstacles and singularities. In addition, generating a smooth and discontinuity-free trajectory respecting the dynamic constraints of the robotic system is a vital requirement for motion planning. The optimization of certain parameters such as execution time and energy is also a challenge for different applications.

## Objectives of the thesis

In this thesis, we aim to speed up the recycling process and make it as efficient as possible by using parallel manipulators. For this purpose, the main objective is to perform throwing tasks in a robust and fast way, by demonstrating the efficiency of a Pick-and-Throw (P&T) technique compared to a usual P&P process in the context of a waste sorting application. Two aspects will be addressed within this thesis: (i) the generation of P&T trajectories for the robot and the ballistic trajectories of the objects to be thrown, and (ii) the synthesis of advanced control schemes to improve the dynamic performance of the parallel manipulators in terms of accuracy and robustness to changing operating conditions.

## Main contributions of the thesis

The main contributions of this thesis aim at improving the dynamic performance of parallel manipulators from trajectory planning and control design point of view. In this framework, the following contributions are introduced:

1. A time-optimal pick-and-throw S-curve trajectory generation.
2. A new augmented RISE feedback controller that incorporates (i) a dynamic compensation term, (ii) RISE feedback term, and (iii) an nonlinear auxiliary term.
3. A novel robust DCAL with adaptive feedback gains. It consists mainly of (i) a PD adaptive feedback term, (ii) an adaptive feedforward term, and (iii) a nonlinear robust term.
4. A new intelligent robust control law developing based on RISE feedback law and model-free control.

The proposed contributions have been studied and validated in real-time experiments and numerical simulations on a parallel manipulator prototype.

## Organization of the thesis

The thesis is organized as follows.

**Chapter 1** provides the context, problem formulation, and the state of the art of this thesis. The main differences between serial and parallel manipulators are discussed. A historical overview of parallel robots as well as some of their potential applications with their recent incorporation in the waste sorting industry are included. A survey and classification of existing motion control strategies implemented for parallel manipulators is included, as well as the state of the art on existing throwing trajectories and the various methods developed to generate a P&T trajectory. The chapter concludes with the main objectives of the thesis as well as the main contributions.

**Chapter 2** provides a detailed explanation of the proposed time-optimal throwing trajectory and its validation through real-time experiments on parallel robot. In addition, the description and modeling of the parallel robot prototype used as testbed to validate the P&T trajectory and the proposed controllers are addressed within this chapter.

**Chapter 3** describes in detail the three proposed control solutions. The contribution for each adopted control strategy is addressed and explained. Moreover, the stability analysis for the three proposed control solutions are formulated.

**Chapter 4** is devoted to the presentation and discussion of the numerical simulation of the proposed control solutions. The results for each case study are presented and discussed in terms of the dynamic performance of the parallel manipulator. The chapter ends up with a conclusion regarding the proposed control solutions and the obtained results.

Lastly, the thesis finishes up with a general conclusion containing a summary of the main contributions as well as some perspectives on future work.

## Publications of the author

### Papers published in international journals

[**IJ1**] Hassan, G., Gouttefarde, M., Chemori, A., Hervé, P.E., El Rafei, M., Francis, C. and Sallé, D., (2022). Time-Optimal Pick-and-Throw S-Curve Trajectories for Fast Parallel Robots. *IEEE/ASME Transactions on Mechatronics*.

## Papers published in international conferences

[**IC1**] Hassan, G., Chemori, A., Chikh, L., Hervé, P.E., El Rafei, M., Francis, C. and Pierrot, F., (2020). RISE feedback control of cable-driven parallel robots: design and real-time experiments. *IFAC-PapersOnLine*, 53(2), pp.8519-8524.

[**IC2**] Hassan, G., Chemori, A., Gouttefarde, M., El Rafei, M., Francis, C., Hervé, P.E. and Sallé, D., (2022). A Novel Extended Desired Compensation Adaptive Law for High-Speed Pick-and-Throw with PKMs. *IFAC-PapersOnLine*, 55(12), pp.627-633.

[**IC3**] Hassan, G., Chemori, A., El Rafei, M., Gouttefarde, M., Francis, C., Hervé, P.E. and Sallé, D., (2022). Ultra-Local Model-Based Intelligent Robust Control of PKMs: Theory and Simulations. *IFAC-PapersOnLine*, 55(12), pp.634-640.

[**IC4**] Hassan, G., Chemori, A., Gouttefarde, M., El Rafei, M., Francis, C., Hervé, P.E. and Sallé, D., (2022). A New Augmented RISE Feedback Controller for Pick-and-Throw Applications with PKMs. *Accepted in IFAC-SYROCO 2022*.

CHAPTER

# 1 Context, problem formulation and state of the art

**Contents**


1.1 Introduction . . . 15
1.2 Classification of robotic manipulators . . . 16
1.3 A historical overview of parallel manipulators . . . 23
1.4 Main applications of parallel manipulators . . . 31
1.5 Overview on Waste sorting . . . 38
1.6 Trajectory generation and control Problem formulation . . . 40
1.7 Dynamic modeling of parallel manipulators . . . 43
1.8 Overview on motion control strategies for parallel robots . . . 49
1.9 Overview on Pick-and-Throw motion generation . . . 61
1.10 Objectives of the thesis . . . 62
1.11 Main contributions of the thesis . . . 63
1.12 Conclusion . . . 65


## 1.1 Introduction

In this chapter, the concept of robotic manipulator is introduced by distinguishing two main kinematic structures of manipulators: serial and parallel. The advantages and disadvantages of each architecture are highlighted. A historical overview of PKMs is introduced. The first investigations and prototypes of parallel robots started in the 20th century, in particular PKMs with the Gough- Stewart platform architecture and those based on the Delta PKM are presented. Over time, several modifications have been carried out to the original Delta PKM to improve its capabilities in terms of stiffness, workspace, and ability

to achieve higher accelerations. Some relevant applications in different fields realized by PKMs are presented with emphasis on the waste sorting application where PKMs are used as sorting robots.

Many factors converge to enable PKMs to successfully perform general-purpose tasks such as high-speed pick-and-place motion cycles, precise positioning, and accurate surgical treatments. These factors range from mechanical design and modeling to trajectory planning and control design. In the literature, several works have been reported related to improving these aspects to increase the capabilities of parallel robots. As the intended application of this thesis is the use of PKMs in a P&T task under different operating conditions (speed, acceleration and parameters variation), the thesis problem formulation addresses both, trajectory planning and control design. In the context of trajectory planning, the generated trajectory must fulfill the requirements of smoothness, continuity and satisfaction of the kinematic and dynamic constraints of the robot. Besides, the motion planning problem consists in proposing techniques to avoid singularities in the robot workspace or to generate optimal trajectories in terms of energy consumption or travel time. From a control point of view, PKMs are known to be highly nonlinear systems with an inherent presence of uncertainties, parameter variations and, in some cases, actuation redundancy. Therefore, advanced control strategies should be developed to deal with these challenges guaranteeing good dynamic and tracking performance of PKMs in task executions.

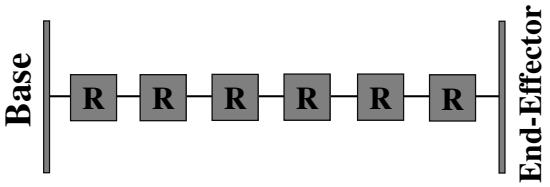
Furthermore, this chapter presented a general overview of existing control solutions for PKMs. A brief discussion on each controller is carried out enlightening the positive and negative points of each strategy. One can distinguish between two types of control strategies of parallel manipulators: kinematic and dynamic control. Because considering the dynamic model within the control design can greatly enhance the dynamic performance of parallel manipulators, the majority of the existing control schemes are full or partial dynamic control strategies. Some of the controllers compensate for a part of the dynamics while the others compensate for all the modeled dynamics. Besides, offline and online dynamic parameter identification techniques exist. In addition, an overview of P&T trajectories and their applications in several domains is provided. Other research works dealing with throwing trajectory generation techniques are also presented in this chapter.

The main contributions of the thesis consist in generating a time-optimal P&T trajectory and in improving the dynamic performance of parallel robots via control design by proposing robust control strategies.

## 1.2 Classification of robotic manipulators

The high demand for introducing new product styles, improving the product quality, and reducing the manufacturing costs has resulted in greater adoption of robotic equip-


(a) Robotic arm IRB 2600 (ABB)



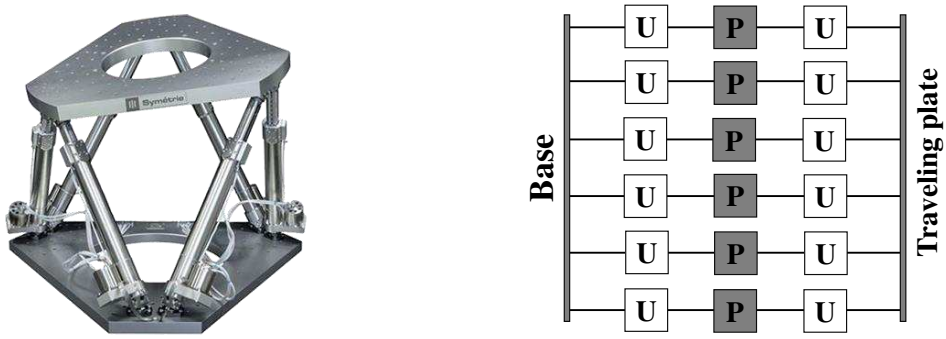

(b) Robotic arm kinematic diagram.

Figure 1.1 – Illustration of a serial manipulator with its kinematic configuration.

ment in various industries. Robots are electromechanical systems composed of mechanical, electrical, and electronic elements controlled by commands generated by a control system (computer, PLC, microcontroller, etc.). The need of robots has been extended beyond industries, nowadays, they are present in space, oceans, agriculture, hospitals, household life, and schools doing productive or leisure activities. According to the International Federation of Robotics under standard ISO 8373, *a robot is an automatically controlled, reprogrammable, multipurpose, manipulator, programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications.* Robotic manipulators are a branch of robotics aiming to perform manipulation and positioning of objects. In the literature, two main kinematic structures of robotic manipulators can be distinguished: Serial and parallel. Two other categories of manipulators worth to be mentioned: The first is a subclass of parallel manipulators. It is a cable-driven parallel manipulator in which the links are constructed from such as cables. The other is a hybrid manipulator that results from the combination of a serial and a parallel structure. Each kinematic architecture has its own interesting features that makes it attractive for carrying out particular tasks. The next paragraphs elaborate on each of them.

### 1.2.1 Serial manipulators

Serial robots are the most commonly used robots in industrial applications. They consist of an open-loop kinematic chain formed by a succession of rigid links connected by prismatic or revolute joints. Each joint is controlled by an actuator located at the joint location or on one of the preceding segments. An example of a serial manipulator is the robotic arm that mimics the parts of the human arm (shoulder, elbow, and wrist). A six-Degree-Of-Freedom (6-DOF) serial robotic arm with its kinematic chain diagram is shown in Figure 1.1, where the gray boxes with the letter R symbolize the active revolute joints.




(a) Hexapod ZONDA from SYMETRIE France (b) Hexapod kinematic diagram.

Figure 1.2 – Exemplification of a parallel manipulator with its kinematic configuration.

### 1.2.2 Parallel manipulators

A generalized definition of PKMs has been given by Merlet in his book *"Parallel Robots"* as follows [Merlet, 2005]: *"a generalized parallel manipulator is a closed-loop kinematic chain mechanism whose end-effector is connected to the base by at least two independent kinematic chains"*. Parallel robots are known to be precise and are able to handle heavy loads while maintaining excellent stiffness. According to the mechanisms, the actuators of these architectures can be either fixed on the base frame or close to it. An exemplification of a parallel manipulator, consisting of a 6-DOF Hexapod with its kinematic configuration, is shown in Figure 1.2. The gray boxes with the letter P denote active prismatic joints, while the white boxes with the letter U represent passive universal joints.

### 1.2.3 Parallel versus serial manipulators

As mentioned above, the main difference between a serial and a parallel manipulator appears obviously in their kinematic construction. A serial manipulator has an open kinematic structure, whereas a parallel manipulator is a closed kinematic chain. Although it is often pointed out that parallel manipulators offer several advantages over their serial counterparts, they are not without their drawbacks. Indeed, each one of them has its own interesting features that makes it suitable for a particular class of applications. Therefore, parallel manipulators do not replace serial ones, but they offer various advantages over serial manipulators for certain applications [Patel et al., 2012]. The various advantages and disadvantages of serial and parallel manipulators are listed below [Krut, 2003; Merlet, 2005; Shayya, 2015; Taghirad, 2013].

**Advantages of serial manipulators**

The main appreciated merits of serial robots are:

- Large workspace volume.

- Good dexterity properties.

**Disadvantages of serial manipulators**
In contrast to the above features, serial manipulators suffer from the following drawbacks [Krut, 2003; Shayya, 2015; Taghirad, 2013]:

- Low stiffness, inherently caused by their open kinematic structure.
- Large moving masses, since they have to carry and move the large weight of most of the actuators.
- Low effective payload capacity.
- Low payload-to-mass ratio.
- Accumulative positioning errors due to the serial configuration.
- Fatigue and wear of the power links supplying the actuators (cables, hoses) or of the active joints.

**Advantages of parallel manipulators**
The most notable advantages of PKMs can be summarized as follows [Krut, 2003; Shayya, 2015; Germain, 2013]:

- High mechanical stiffness thanks to the closed-chain structure.
- Significantly higher payload to robot mass ratio.
- High dynamic capabilities owing to low moving mass and inertia.
- Improved tracking accuracy and better precision thanks to the parallel structure, where the end-effector pose errors are non-cumulative.
- Possibility of placing the actuators directly on the fixed base or very close to it; this feature allows for the following additional benefits:
  - — Higher flexibility regarding the choice of motors and/or gearboxes, as their mass does not influence the mass and inertia of the moving parts.
  - — Significant simplification of problems arising from cable connections between the motors, sensors, and controller (more straightforward and more reliable wiring).
  - — Ease of cooling of the actuators resulting in reduced precision problems due to thermal expansions and high potential power.
  - — Ability to attain high accelerations because all the motors are fixed to the base frame so that the moving masses have a low total weight.

**Disadvantages of parallel manipulators**
Compared to serial robots, the main drawbacks of parallel robots can be summarized as follows:

Table 1.1 – The main properties of serial and parallel manipulators.

| Property | Serial manipulators | Parallel manipulators |
|---|---|---|
| Singularities | Inverse kinematics | Inverse, forward, and combined |
| Workspace | Large | Small |
| Payload/weight ratio | Low | High |
| Actuators' location | On the joints | Near or on the fixed base |
| Stiffness | Low | High |
| Dynamic performance | Poor | Possibly very high |
| Accuracy | Low | High |

- Limited workspace compared to the total volume of the mechanism.
- Complex forward kinematics (sometimes there is no unique solution).
- Presence of singularities that are critical and may lead to uncontrollable motion of the mobile platform or the deterioration of the mechanical system. Avoiding such singularities is a key challenge in the design of a parallel robot.
- The use of many passive links may induce backlash and clearances, making the robot behavior difficult to model. However, it is necessary to master them to define dangerous zones near the singularities and to improve the robot accuracy.
- Internal force generation that may produce mechanical damages in the case of Redundantly Actuated (RA) PKMs.

In accordance with [Pandilov and Dukovski, 2014; Bennehar, 2015; Saied, 2019], and the points mentioned above, one can summarize the proprieties of serial and parallel manipulators as reported in Table 1.1.

### 1.2.4 Cable-driven parallel robots

Cable-Driven Parallel Robots (CDPRs) can be defined as a subclass of PKMs in which rigid links are replaced by cables [Bruckmann and Pott, 2012]. They mainly consist of a mobile platform driven by cables, which transmit the forces generated by winches. Consequently, the movement of the platform is managed by controlling the winch motors. CDPRs have several advantages compared to classical rigid-link parallel robots. Since a large length of cable may be wound on the drum of each winch, the mobile platform may be displaced over a large workspace. Moreover, cables are able to transmit large forces, which make possible the handling of heavy payloads (i.e. high payload to weight ratio). For instance, CoGiro prototype, shown in Figure 1.3, is capable of transporting a payload of 500 Kg ([Lamaury and Gouttefarde, 2013]). As a subclass of parallel robots, CDPRs typically present their actuators fixed to the base frame. In addition, the total mass and inertia of the moving parts may be drastically reduced using cables instead of rigid links allowing high

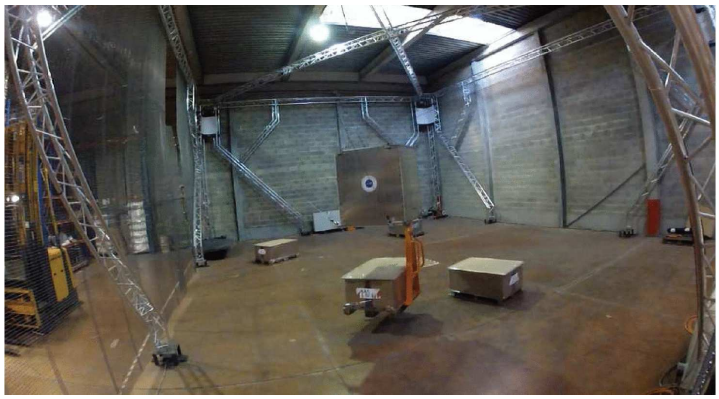
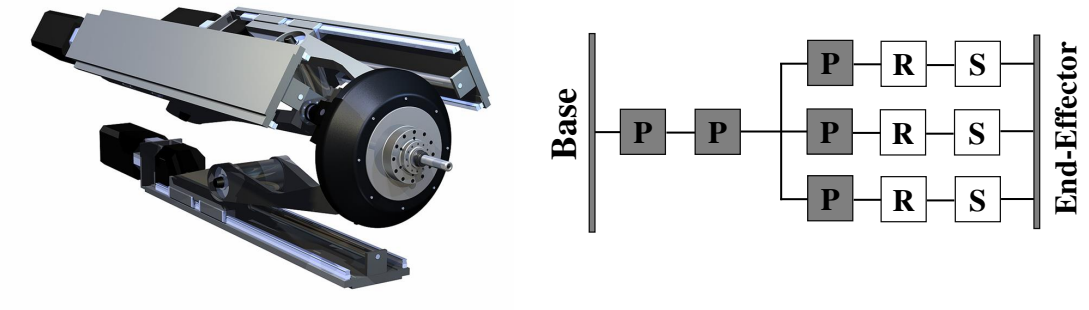


Figure 1.3 – Illustration of the CoGiRo (Control of Giant Robots) prototype built in the framework of a collaborative research project between TECNALIA and LIRMM.

dynamical performances to be achievable. For example, the FALCON robot can generate 13 m/s of maximal velocity and 43 G of maximal acceleration [Kawamura et al., 2000]. Besides, most often the wires can be considered massless and inextensible, which simplifies their modeling. In contrast to these advantages, CDPRs present some important drawbacks. In particular, when cables become lengthy, their mass cannot be any more ignored. Also, when high tension forces exist, the hypothesis of inextensibility is no more justifiable. All these complicate the modeling of CDPRs, as sagging and deformations are to be necessarily considered. Furthermore, as cables can only pull and not push, the feasibility of a pose depends not only on cable length limits and mechanical interference avoidance, but also on the possibility of having static equilibrium. Actually, the problem of tension distribution in CDPRs is a non-trivial issue and one of the key research topics in this field. This is not to mention the challenges that CDPRs present on the control level. Moreover, the cable elasticity may introduce undesired vibrations [Weber et al., 2015; Begey et al., 2018; Baklouti et al., 2019]. Similarly, substantial degradation of the positioning precision may be obtained when neglecting cable elasticity [Schmidt and Pott, 2017; Paty et al., 2021]. For more information regarding CDPRs, their state-of-art, applications, and advancements, the reader may refer to [Lamaury and Gouttefarde, 2013; Baklouti, 2018; Santos, 2020], and to the articles cited here and therein.

In general, three groups of CDPRs can be distinguished, namely the fully-constrained, the suspended CDPRs, and the under-constrained ones [Pott, 2018].

In the case of fully-constrained CDPRs (e.g.[Kawamura et al., 2000]), there is at least one pose in the workspace of the platform such that any wrench can be generated by the cables by pulling on the platform. This configuration necessarily presents a number of cables greater than the number of DoF of the mobile platform. This condition is necessary but not sufficient. A particular geometry and arrangement of cables must also be considered. The exit points of the cables are located above and below the workspace of the




(a) A CAD view of Sprint Z3 (DS Technology) (b) Sprint Z3 kinematic diagram

Figure 1.4 – Sprint Z3 hybrid manipulator with a serial carrier and a parallel wrist.

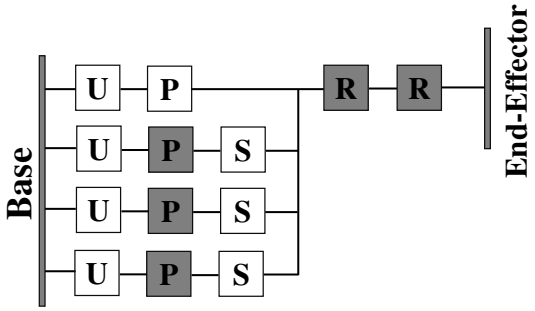
platform. In suspended configuration (e.g. [Yao et al., 2009]), every cable force applied on the platform is directed upwards, this type of CDPR cannot generate a force directed downwards. Therefore, suspended CDPRs rely on the gravitational forces in order to constraint the platform. This configuration requires at least as many cables as the platform DOF. Finally, under-constrained CDPRs are characterized by having fewer active cables than the number of DOF of the end effector, such as the Winch-Bot presented in [Cunningham and Asada, 2009]. In this case, the platform pose cannot be determined geometrically. In addition to the kinematic model, a static or dynamic model should be considered in order to determine the platform pose [Abbasnejad and Carricato, 2015].

### 1.2.5 Hybrid manipulators

A hybrid mechanism is one that results from the combination of a serial and a parallel structure. Three categories of hybrid manipulators can be distinguished:

1. *Mechanisms with a serial carrier and a parallel wrist:* In this configuration, the carrier mechanism has the serial structure coupled with a parallel wrist responsible for orienting the end-effector. The Sprint Z3 machine from DS Technologies is an example of this category that was developed for aeronautical industrial applications [Chen et al., 2014]. It consists of a series-structured carrier providing the $xy$ movement by means of two prismatic actuators. On this carrier, a parallel module is fixed. The latter provides the wrist motions (two rotations) and the translation motion along the $z$-axis. Its design is quite simple, and it has met a real commercial success. This manipulator is depicted in Figure 1.4, where the white boxes with a letter S represent passive spherical joints.
2. *Mechanisms with a parallel carrier and a serial wrist:* In this category of hybrid manipulators, the parallel mechanism is responsible for positioning, while the serial

(a) A general view of Tricept 845 (Neos Robotics)




(b) Tricept kinematic diagram

Figure 1.5 – Tricept hybrid manipulator with a parallel carrier and a serial wrist.

mechanism is used for the end-effector orientation. It is therefore possible to obtain large rotational motion since the wrist is of a totally classical design. The Tricept 845 from Neos, shown in Figure 1.5, is a typical example of such configuration. This machine has three translational movements performed by three universal-prismatic-spherical kinematic chains and two rotational movements performed by a serial wrist mechanism. A particularity of this robot is the presence of a passive kinematic chain that constrains the platform of the parallel structure [Neumann, 2006].

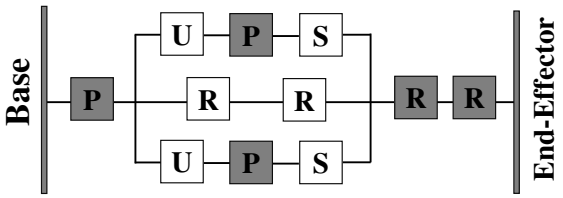
3. *Other hybrid mechanisms:* Dumbo is a 3T-2R hybrid machine (one dof carrier + parallel module + serial wrist) developed at the IFW at the University of Hannover (Germany). It is shown in Figure 1.6. The carrier is a hybrid mechanism on its own since a $z$-axis column supports a parallel mechanism consisting of two hydraulically actuated extensible links and a passive leg. The serial architecture wrist gives the machine large angular deflections.

## 1.3 A historical overview of parallel manipulators

According to [Bonev, 2003], the first patent of a moving platform based on a spherical parallel mechanism was filed in 1928 by James E. Gwinnett. It is consisted of an amusement device intended to be used in the entertainment industry (a dynamic cinema) [Gwinnett, 1931] (see Figure 1.7). Unfortunately, the designed mechanism was never built because it was too complex for the industry at that time.

A decade later, Willard L.V. Pollard invented a 5-DOF parallel manipulator for automated spray painting operations. His invention, shown in Figure 1.8, was later patented by Willard's junior in 1942. In the parallel kinematics community, Pollard's parallel robot


(a) A view of Dumbo (IFW)




(b) Dumbo kinematic diagram

Figure 1.6 – Dumbo hybrid manipulator.

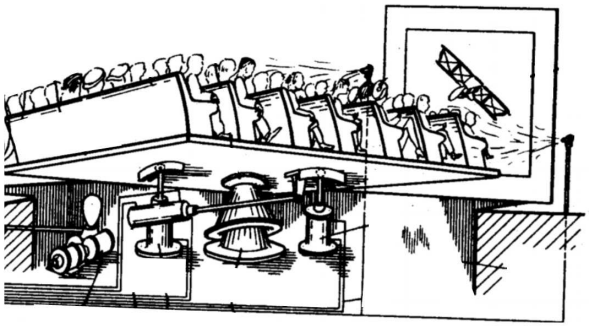


Figure 1.7 – The spherical parallel mechanism proposed by James E. Gwinnett [Gwinnett, 1931].

is well known as the first industrial parallel robot. The mechanism design is composed of three sets of links, where each set is formed by a proximal and a distal arm connected by universal joints [Pollard, 1942]. Three actuators mounted on the base are responsible for the positioning of the tool head, while its orientation is controlled by the other two actuators fixed to the base, and transmitting the movement to the tool through flexible rotary cables. While other spray painting robots have been commercialized, the one of Pollard was never built.

In 1947, Dr. Eric Gough proposed the first octahedral hexapod type parallel mechanism with variable length sides that allows the positioning and orientation of the moving platform [Bonev, 2003] (see Figure 1.9). This device was invented to resolve the problems of air-landing loads and is capable of testing the tyre wear. It was used to check the tires of Dunlop house under loads applied along different axes. The Gough platform has been one of the most popular mechanisms in parallel robotics. It has been widely studied in the literature and often referred to as the first parallel manipulator ever built. The universal

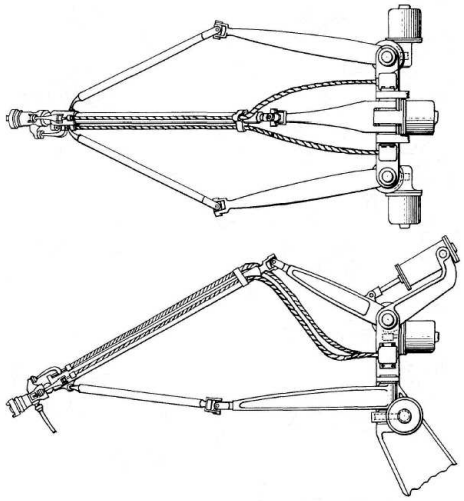


Figure 1.8 – Illustration of the spray painting 5-DOF PKM proposed by Willard L.V. Pollard [Pollard, 1942].

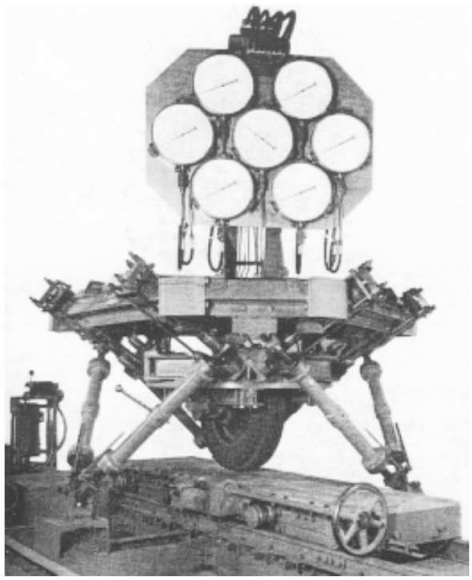


Figure 1.9 – View of the original Gough platform in 1954 [Bonev, 2003].

tire-testing machine, or the universal rig, as Dr. Gough called it, was fully operational in 1954 and still used until 2000 (see Figure 1.10).

Many years later, in 1965, Stewart published a paper in which he proposed a 6-DOF parallel manipulator for use as a flight simulator [Stewart, 1965]. His design was different from the octahedral hexapod developed by Gough that is, ironically, often referred to as the

Figure 1.10 – Gough tyre-testing prototype, called universal rig, exhibited in the British National Museum of Science and Industry in 2000 [Bonev, 2003].

Stewart platform. The Stewart's paper appeared in the proceedings of the British IMechE, and Dr. Gough was one of the reviewers who reminded of the existence of his tire-testing machine. Figure 1.11 depicts the proposed Stewart's platform. However, Stewart is not considered the creator of the first flight simulator based-on PKM even though his paper had a great impact on the subsequent development in the field of parallel robots especially those of hexapod architectures.

In the mid of 1960s, the US engineer, Klaus Cappel, an employer at the Franklin Institute Research Laboratories in Philadelphia, designed and built the first functional flight simulator at the request of Sikorsky Aircraft Division of United Technologies for design and construction of a 6-DOF helicopter flight simulator [Bonev, 2003]. His design was based on the same octahedral hexapod arrangement proposed years ago by Dr. Gough. The Cappel patent was filed in 1964, at which time he was unaware of Gough's invention and Stewart's paper which was not yet published. This mechanism, illustrated in Figure 1.12, is referred to the first-ever flight simulator built based on the octahedral hexapod [Cappel, 1967].

For almost two decades, parallel manipulators have not attracted as much attention as they did in the early 80s, when Reymond Clavel, a professor at the École Polytechnique Fédérale de Lausanne (EPFL) in Switzerland, introduced the design of the Delta PKM with three translational and one rotational degree of freedom [Clavel, 1990]. The key design feature is the use of parallelograms in the kinematic chains, which restrains completely the orientation of the traveling-plate, resulting in only three translational movements. A fourth leg is used to transmit rotary motion from the base to an end-effector mounted on the mobile platform. The original design of Delta robot is shown in Figure 1.13 [Taghirad, 2013]. The use of base-mounted actuators and lightweight arms allows the mobile platform to achieve very high accelerations (up to 50 G). This ingenious idea makes the Delta

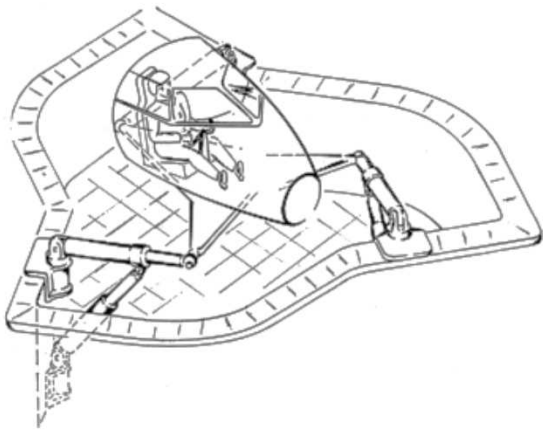


Figure 1.11 – Illustration of the Stewart platform proposed as a flight simulator [Stewart, 1965].

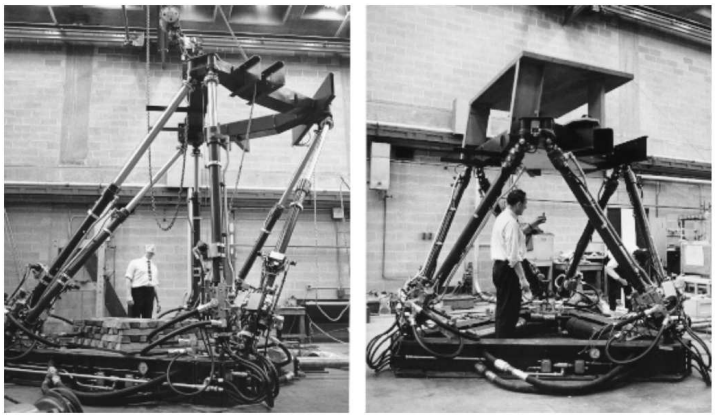


Figure 1.12 – View of the first flight simulator of Klaus Cappel [Cappel, 1967].

robot a perfect candidate for high-speed pick-and-place operations of lightweight objects. In 1999, Dr. Clavel was awarded the "Golden Robot Award", sponsored by ABB Flexible Automation, for his innovative work on the Delta robot [Bonev, 2001]. The Delta robot is one of the most successful PKMs ever built.

**Other variants of Delta PKM over time**

The idea of using parallelograms has inspired many new designs where several modifications have been made to the original Delta robot of Prof. Clavel. Although they often share many similarities, each prototype usually has its own characteristics, advantages, disadvantages and intended applications. For instance, the Star robot of Hervé [Hervé, 1991] was invented to have the same characteristics as the Delta robot without depending on its patent. This architecture also has three motors fixed to the base, as well as three identical kinematic chains composed of spatial parallelograms. The motors drive helicoidal joints

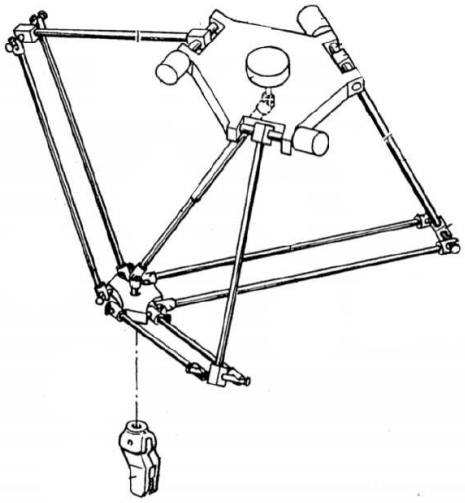


Figure 1.13 – Technical drawing of the original Delta robot of Reymond Clavel [Clavel, 1990].

in translation. These joints are connected to the mobile platform by means of a $\pi$ link and a revolute joint (see the Star robot in Figure 1.14). Even if this robot can theoretically claim to be able to match the dynamic performance of the Delta, its small working volume is very penalizing. In [Pierrot et al., 1991], F. Pierrot presents a new design for a 6-DOF parallel structure, called "Hexa" PKM, as an extension of the 3-DOF Delta robot. In contrast to the Delta robot, the Hexa robot can rotate its traveling plate thanks to six kinematic chains, each of them being actuated by an individual motor. In [Chablat et al., 2000], the Orthoglide machine tool was developed by the IRCCyN. It has prismatic actuators placed in such a way that this mechanism has an isotropic configuration at the center of its working volume. The three spatial parallelograms linking these actuators to the mobile platform impose on the end-effector translational movements. The prototype developed by the IRCCyN was designed for light milling applications. It is capable of accelerations of the order of 20 $m/s^2$ and a speed of 1.2 $m/s$.

In order to avoid the central telescopic leg present in the Delta robot, some mechanical solutions using the concept of articulated traveling plates were proposed. The first prototype, called H4, has 4-Dof, 3 in translations and 1 in rotation about a given axis [Pierrot and Company, 1999]. It uses four actuated kinematic chains instead of three. These chains are connected to an articulated traveling plate made up of three parts linked by means of two passive revolute joints. Later, in [Krut, 2003], the traveling plate of H4 was equipped with an additional gear-based amplification system leading to a large and adjustable range of motion in orientation. Notwithstanding, this PKM has some drawbacks such as the abundance of singularity configurations and the possible internal collisions because of the architecture of its traveling plate. Thus, the relative positions of the four

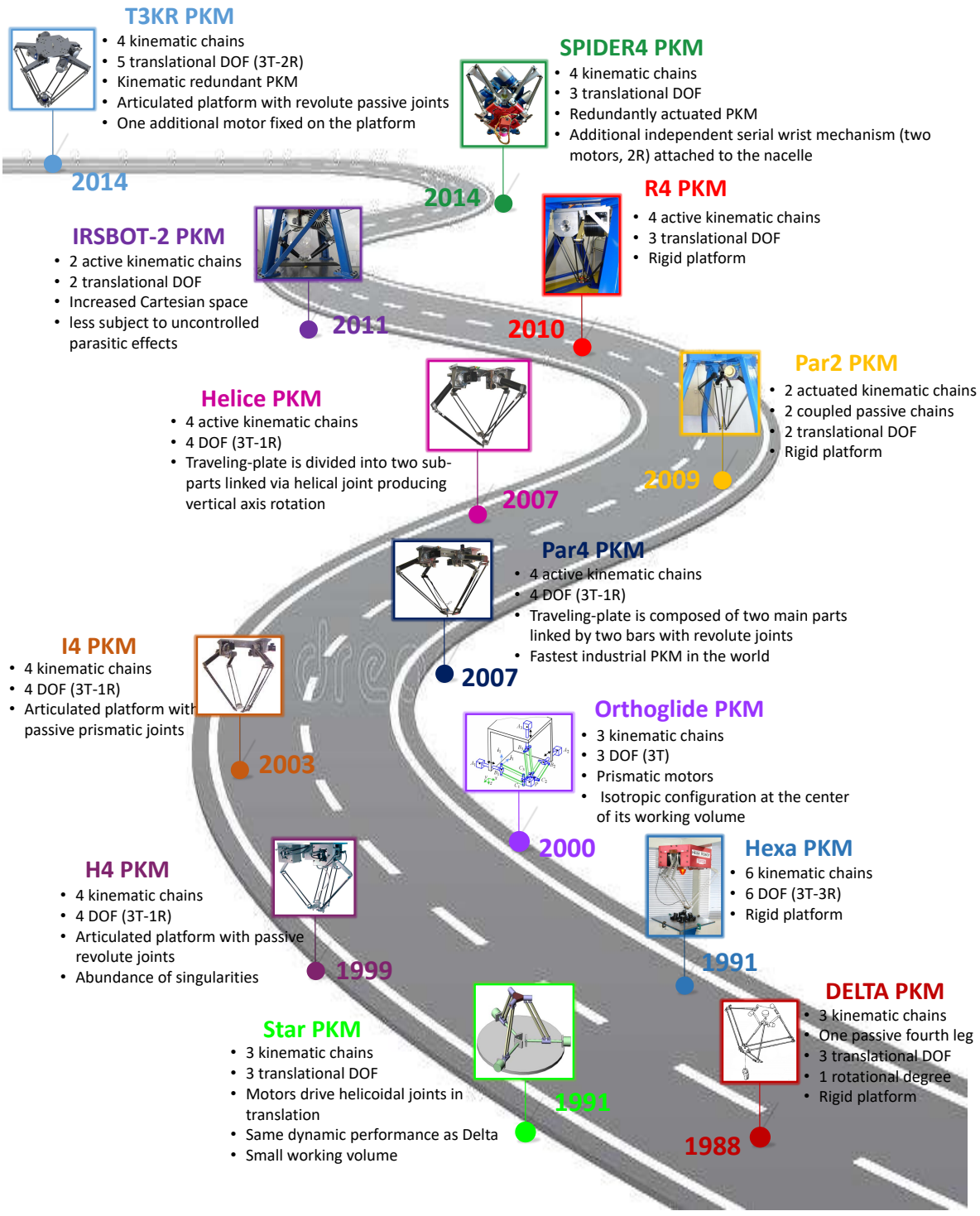



Figure 1.14 – Timeline of some PKMs based on the Delta robot.

spatial parallelograms must be properly selected to avoid singular cases. To compensate for the limitations of H4 PKM, the I4 family were proposed by [Krut et al., 2003]. I4 family has two variants I4L and I4R, which are actuated by linear and rotational actuators, respectively. The basic idea behind I4, as compared with H4, is the replacement of revolute joints by prismatic ones and gears by rack-and-pinion mechanisms. With this new design of the traveling plate, the risk of internal collisions (one part of the traveling plate colliding the other) is drastically reduced and the singularity problem vanishes.

However, the use of prismatic joints in the articulated traveling plate constitutes a main weak point of I4. As a matter of fact, used at high speed, commercial prismatic joints have a short service life, due to high acceleration and pressure exerted on balls. Thus, I4 is well suited for high force/ moderate acceleration application such as machining. To overcome the drawbacks of I4, a new design of a 4-DOF parallel manipulator, called Par4, was developed in [Pierrot et al., 2009b]. Its traveling plate, composed of two main parts linked by two bars with revolute joints, allows the robot to be free of internal singularities and to achieve high performance. Two amplification systems were proposed in order to obtain a complete turn: The first utilizes a gear assembly and the other exploits pulleys and belts. Nowadays, Par4 is commercialized under Omron Quattro's name, considered the fastest industrial PKM in the world [Bennehar, 2015]. It is worth noting that this machine is still over-constraint, so other solutions could be analyzed. A modified version of Par4, called Heli4, was presented in [Nabat, 2007], with an even simpler design and a more compact traveling plate. This system is not over-constraint implying that it is away from singular positions. The innovative feature of Heli4 lies mainly in its traveling-plate which is divided into two sub-parts linked via helical joint producing vertical axis rotation [Company et al., 2013]. Each part is connected to two opposite side kinematic chains. This robot has been later industrialized by Penta Robotics under the name of Veloce. [Shayya, 2015].

In [Baradat et al., 2009], a 2-DOF parallel manipulator producing two translations in the vertical plane was introduced. This parallel architecture, named Par2, is composed of a rigid platform, two actuated kinematic chains, and two passive chains built in the transversal plane. The key feature of this robot comes from the passive chains which are coupled to restrict the platform movement in only one plane. It has been shown in [Pierrot et al., 2009a] that accelerations higher than 40 G can be achieved with this robot while keeping a low tracking error. In [Germain et al., 2011], a new 2-DOF translational parallel robot, named IRSBot-2, was proposed. It has a special architecture composed of only two active chains in order to reduce the robot complexity and to increase the size of its Cartesian workspace. Besides, it is less subject to uncontrolled parasitic effects that may be produced by the passive chains of Par2. In the literature, there are other prominent PKMs that use a Redundantly Actuated (RA) configuration. For instance, R4 is a RA-PKM, which has 3-DOF (translations the along $x$, $y$, and $z$ axes) and 4 actuators. It has been reported that this PKM can reach extreme accelerations up to 100 G in a P&P application [Corbel et al., 2010b; Natal et al., 2014].


Figure 1.15 – ABB's IRB 360 FlexPicker


Figure 1.16 – The high-speed Omron Adept Quattro robot

SPIDER4 robot is a Delta-like RA-PKM which has 4 actuators fixed on the based allowing three translational movements, and an additional independent serial wrist mechanism (two motors) attached to the nacelle offering two more rotational movements for the machining spindle [Saied, 2019]. It is the first Delta-Like PKM destined for machining applications [Escorcia-Hernández et al., 2020b]. T3KR is a PKM dedicated to P&P applications. It shares several similarities with Par4; However, the rotational movement of the end-effector is performed by an independent actuator located at the traveling plate. More detailts about this PKM will be presented in Chapter 2. Figure 1.14 presents a timeline with all the examples presented in this section.

## 1.4 Main applications of parallel manipulators

Thanks to their potential properties such as high accuracy, improved stiffness, high dynamics, and high payload-to-weight ratio, parallel manipulators have gained a great interest in several applications where such features play a crucial role [Merlet, 2005; Patel et al., 2012]. Nowadays, PKMs have become essential for certain applications such as industrial packaging and motion simulators. Moreover, parallel robots are becoming more popular in several fields such as food packaging, motion simulators, machining applications, medical applications, agricultural applications, 3D printing, and haptic technology.

### 1.4.1 High speed Pick-and-Place tasks

Pick-and-place is the process of taking objects from one location and placing them in another. This task is essential in assembly lines or packaging production lines where

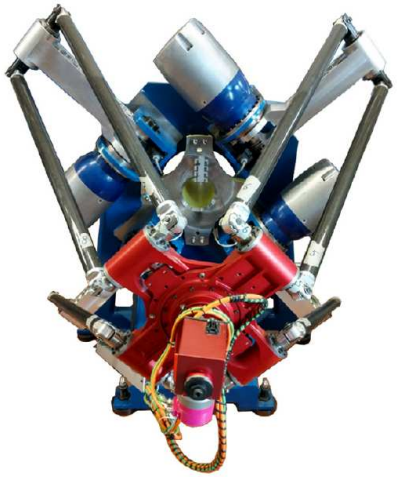
precision is required, mainly at the stop points (start and end points). The most popular robots for P&P operations are the Delta robot and Delta-like robots, owing to their high dynamic capacity. Delta-like robots were developed based on the original Delta robot with some modifications to meet the requirements of PKM, such as high load capacity, high accuracy and high dynamic performance. Figure 1.15 shows ABB's FlexPicker Delta robot, specially designed for industries with a high need for flexible automation, such as P&P operations and assembly. It is extremely powerful with an acceleration of up to 10 G, and a handling capacity of up to 8 kg. Omron Adept Quattro, shown in Figure 1.16, is a 4-DOF Delta-like PKM with a higher maximum acceleration (15 G). This robot was initially designed at LIRMM under the name Par4 [Pierrot et al., 2009b]. The general structure of Par4 was inspired by the Delta robot with an additional kinematic chain and an articulated raveling-plate allowing greater stiffness near the limits of the workspace.

### 1.4.2 Machining applications

Machining, also known as subtractive manufacturing, is a process in which a material is cut to the desired final shape and size by a controlled process of material removal. In machining, several operations take place in a planned sequence to achieve the best results. The most common machining operations are namely turning, drilling and milling. Various types of materials can be machined using these three methods. Metals, plastics, composites, and wood are all possible materials for workpieces. These operations are critical manufacturing processes that require high precision in the positioning of the cutting tool and the desired cutting path. Parallel robots featuring high precision, high acceleration capabilities due to the light weight of the moving parts and high stiffness due to the closed-chain structure are progressively being adopted to develop this type of machines instead of the conventional serial manipulators. A machine tool based on a parallel structure has been proposed in [Toyama et al., 1998] performing improved machining with high stiffness. In [Shayya, 2015], a machining device, ARROW PKM, has been developed at LIRMM and is capable of executing 5-DOFs in a large workspace. In addition, SPIDER4 is a 5-DOFs redundantly actuated PKM designed and manufactured within a collaboration between LIRMM, and TECNALIA for CNC machining tasks of resin materials [Escorcia-Hernández et al., 2020b]. The mentioned examples are illustrated in Figure 1.17.

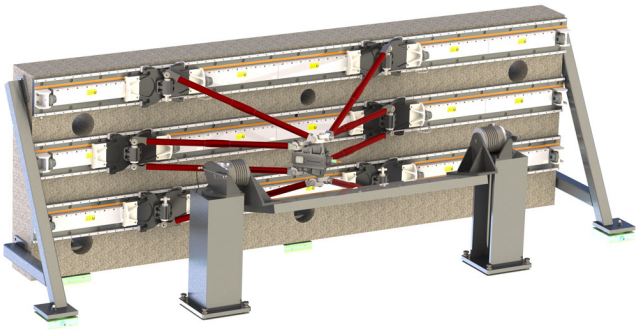
### 1.4.3 Motion simulators

Motion simulators, such as flight, car, ship, and space simulators, are typical examples of the most relevant applications developed by PKMs. Gough-Stewart platform is the most widely used prototype for this type of applications because it offers 6-DOF motions, sufficient capacity to handle heavy loads, high stiffness, and fast and accurate motion. In the aerospace industry, motion simulators plays a crucial role in aircraft design as it allows the analysis of aircraft dynamics within and beyond the design flight envelope, thereby pre-



(a) View of SPIDER4 PKM [Escorcia-Hernández et al., 2020b]


(b) Toyoda HexaM PKM-based machine tool [Toyama et al., 1998]



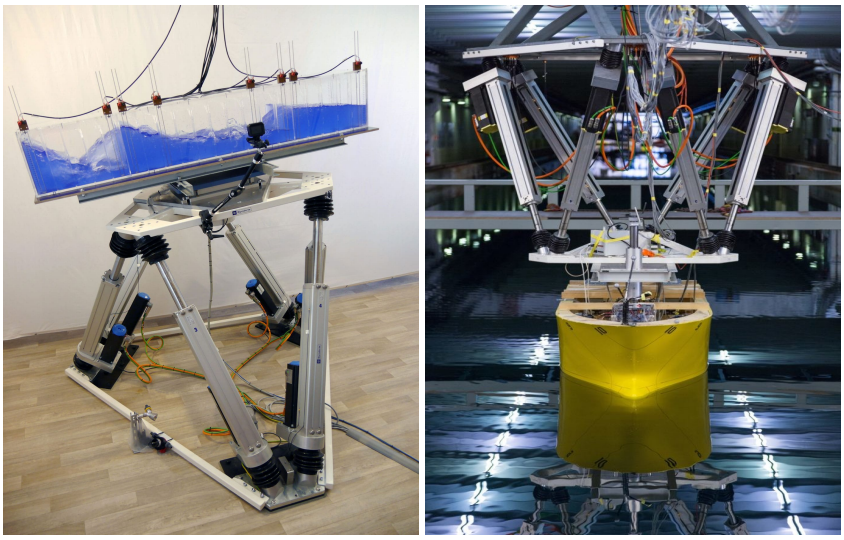
(c) CAD view of ARROW PKM

Figure 1.17 – Illustration of three examples of PKMs for machining operations.

venting fatal accidents [Asif, 2012]. Figure 1.18 presents two examples of flight simulators. In addition to being used as flight simulators, hexapods have also been used as swell simulators in the naval industry for material resistance testing or in the oil and gas industry to test the marinization of processes on floating production units. A hexapod developed by SYMETRIE and used as swell simulators is shown on the left of Figure 1.19. Numerous research studies have investigated the use of SYMETRIE's hexapods to emulate the motion of floating ships [Assima et al., 2015a,b]. In addition, SYMETRIE has developed hexapods to be used in the study of hydrodynamic effects of swell, sloshing or cavitation phenomena for example [Symetrie, 2017]. This hexapod is illustrated on the right of Figure 1.19. As it can be seen, it is attached downwards to a carriage that moves along the basin. The moving platform of the robot moves the ship model to apply on it forces representative of those generated by the swell for testing purposes.


(a) Lufthansa flight simulator (b) FRASCA filght simulator

Figure 1.18 – View of two examples of flight simulators.



(a) MISTRAL hexapod swell simulator at HOPPE Marine (b) MISTRAL hexapod with a ship model at the wave basin of Ifermer Boulogne sur Mer, France

Figure 1.19 – View of two MISTRAL hexapods swell simulators developed by SYMETRIE.

### 1.4.4 Medical applications

PKMs hold many advantages, such as high stiffness, load capacity and accuracy, that make them perfect candidates for some specific medical purposes. For instance, PI M-850 Hexapod is a 6-DOF parallel-kinematics micro-positioning system used to achieve high safety of micro-surgical procedures and feasibility of micro-therapy [Dalvand and Shirinzadeh, 2012]. A Gough-Stewart platform (6-DOF hexapod robot system) has been designed to be used for precision microsurgery in a number of different medical disciplines, such as neurosurgery, ophthalmology, spine surgery and orthopaedics [Wapler et al., 2003]. The surgiScope is a ceiling mounted robotized tool-holder device developed by the Intelligent Surgical Instruments & Systems company (ISIS) and especially dedicated to microscope, applications in neurosurgery. Its mechanical system, which carries the microscope is based on a Delta-like structure [Briot et al., 2007]. In [Li and Xu, 2007], a medical Delta-like parallel robot applicable to chest compression in the process of cardiopulmonary resuscitation (CPR) on patients undergoing cardiac arrest has been designed and developed. Figure 1.20 illustrates the mentioned examples for medical applications.

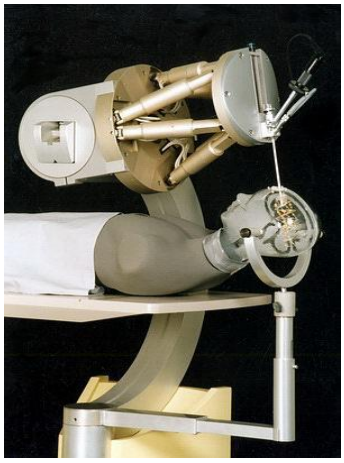
### 1.4.5 Agricultural applications

The agricultural industry is one of the most popular and relevant industries today where robotics is growing due to the use of heavy labor-intensive operations in row crop cultivation and care. These operations include inter-row processing, which relies on mechanical or chemical destruction of weeds [Ovchinnikov et al., 2020]. "The ecoRobotix" company has recently developed an autonomous robotic system for weeding. It is a four-wheeled vehicle equipped with two Delta robots that can perform all weeding duties on the farm strictly on its own (see Figure 1.21). It is equipped with a vision system that allows it to identify crops and detect the presence of weeds among the crops. Once it recognizes the presence of weed, the Delta PKMs apply a micro-dose of herbicide to their exact location, systematically targeting the detected weed without wasting any chemicals. The system can be remotely controlled via a smartphone applications, in addition to a full solar power system [ecoRobotix, 2011]. This invention significantly reduces the use of herbicides and fertilizers, which can be harmful when used in excess.

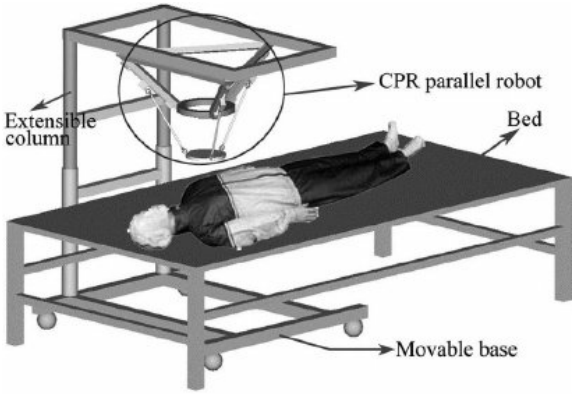
### 1.4.6 3D printers

Three-dimensional (3D) printing, or additive manufacturing, involves building a three-dimensional object from a computer-aided design (CAD) model by adding the material layer by layer. In recent years, 3D printers have gained popularity because they are able to manufacture parts at low cost and reduce waste. Currently, this technology is used to manufacture automotive components, aerospace components, custom orthodontics, among others [Campbell et al., 2011]. A common parallel architecture for 3D printing is given by

(a) SurgiScope PKM designed by the ISIS company



(b) Hexapod medical robot for computer-aided surgery




(c) Conceptual design of a CPR PKM

Figure 1.20 – Illustration of some examples of PKMs for medical applications.

Figure 1.21 – View of ecoRobotix smart weeding robot.


(a) Delta Go Smart Desktop 3D Printer

(b) Monoprice Mini Delta 3D Printer

Figure 1.22 – View of two examples of 3D printers based on PKMs.

the Linear Delta Robot (LDR) that allows three translations of the end-effector in the operational space. The linear delta architecture makes this robot suitable for 3D printing tasks, where cyclic operations are required to deposit the material layer by layer [Carabin et al., 2021]. Figure 1.22 presents some commercial 3D printers based on the linear Delta PKM.

### 1.4.7 Haptic devices

The word haptics refers to sensing and manipulation through the sense of touch. Haptic devices provide force feedback to the users, allowing them to feel and interact indirectly with an external environment. They are useful for certain tasks where visual information is not sufficient and can lead to unacceptable manipulation errors. These mechanisms can be found, for example, in game controllers, teleoperation, remote-controlled surgery, space exploration and neurorehabilitation. For high performance haptic devices, some mechanical features such as high stiffness, low friction, high dynamic range and low inertia must be exhibited. These properties can be found naturally in parallel mechanisms. PKMs are therefore especially appropriate for sophisticated haptic interfaces: [Birglen et al., 2002]. Among the PKM-based haptic devices are the Omega.6 from the Swiss company Force Dimension [Dimension, 2004], which is one of the most advanced force feedback interfaces providing 3D force feedback with decoupled translations and rotations [Aggravi et al., 2021], and the sigma.7 model, which is an haptic interface ever designed by Force Dimension. Featuring a redesigned delta base, this device is the first commercial haptic interface to offer seven active degrees of freedom, including high precision force feedback gripping capabilities [Dimension, 2010]. Figure 1.23 shows the two mentioned examples.

(a) Omega.6 from Force Dimension (b) Sigma.7 from Force Dimension

Figure 1.23 – Illustration of two examples of PKM-based haptic devices.

## 1.5 Overview on Waste sorting

One area that can be significantly enhanced by modern robotic technology is waste sorting. In particular, the need for separating recyclables into bins dedicated to different material types so that they can be used as a second order resource, can be significantly enhanced by robotic technology. This is the only way to effectively manage the ever increasing volume of waste created by people following the modern life style.

To date, industrial waste sorting is mostly based on the traditional pneumatic sorting machines. However, the use of these machines comes along with high-cost and large-volume industrial installations. Moreover, they have great detection capabilities but not a perfect selectivity. Robots are used along side these machines to remove the undesired material from the output flows to increase their purity.

Incorporating robots into recycling industry is challenging due to the need of using tough enough equipment to withstand a beating in harsh recycling environments. Further to that, fast and effective gripping of materials without potential slips during transfer, is highly desirable in waste sorting. Vacuum provides particularly powerful gripping properties [Pham and Yeo, 1991] and has been one of the key enabling technologies that facilitated the use of robots in recyclable sorting industrial applications. Currently, there is already a number of commercial AI-driven robots that have been applied in industrial environments to pick out recyclable materials. Some of the most well known systems are: Max-AI Autonomous Quality Control (Max-AI AQC) robotic sorter [BHS, 2017], SamurAI - Machinex sorting robot [Machinex, 2017] (see Figure 1.24), and AMP Cortex Dual-Robot System (DRS) [AMP, 2019]. Most of the existing systems rely on the particularly fast movement of the Delta robots to achieve a fast transfer of the recyclables from the conveyor belt to the sorting bin. The benefit of the Delta robot kinematic design is that it reduces the weight within the arms and therefore provides very high acceleration capability.

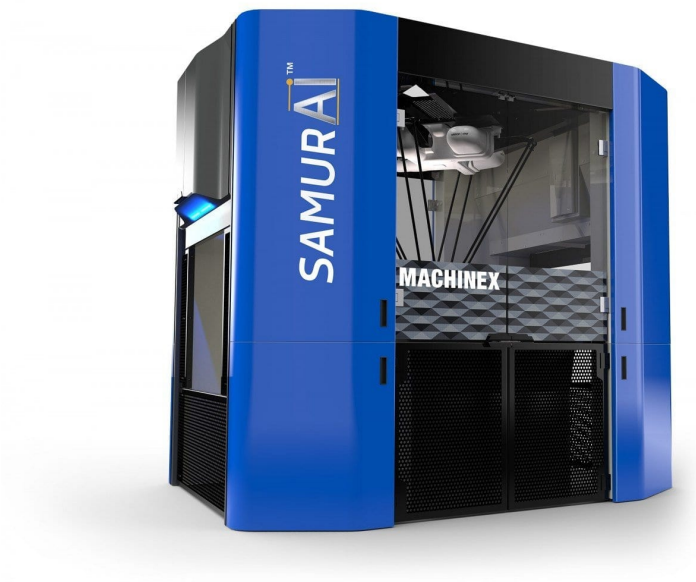



Figure 1.24 – SamurAI is a self-aware sorting robot, presented by "Machinex", that employs superior artificial intelligence technology to identify materials for a positive product recovery.

Figure 1.25 – A dual-arm robot uses pick-and-throw in selective waste sorting. This robot was built by "Bulk Handling Systems" to make recycling more efficient.

Recently, Max-AI in collaboration with Bulk Handling Systems (BHS) released a new sorting solution that employs two collaborating robotic arms to sort recyclables following a material tossing approach [BHS, 2019] (see Figure 1.25). This system has been presented as the main competitor of the commonly used delta-robot approach.

## 1.6 Trajectory generation and control Problem formulation

The growing interest in parallel robots in several domains and applications always motivates the scientific community to study and improve these machines to meet the requirements of each corresponding application. There are multiple aspects from which parallel manipulators can be analyzed, synthesized and improved. These aspects include the mechanical design and architecture optimization problem, the kinematic and dynamic modeling, the motion planning problem and, last but not least, the control design.

In this thesis, we aim at successfully developing a P&T task using a parallel manipulator. Therefore, both aspects, trajectory generation and control design, will be addressed in this framework. On the one hand, the trajectory planning consists in generating a time-optimal P&T trajectory satisfying the kinematic and dynamic constraints of the robotic system. On the other hand, the control design consists in proposing robust control solutions aiming at improving the dynamic performance of parallel manipulators under different operating conditions.

### 1.6.1 Main challenges in trajectory generation for PKMs

Trajectory planning is a major research area in robotics. Research in this area started in the 1970s. It deals with the problem of finding a collision free path from an initial state to a final state given a complete description of robot physical limitations, geometry and environment [Al Homsi, 2016]. To avoid confusion between terms often used as synonyms, the difference between a path and a trajectory is to be explained. A path denotes the locus of points in the joint space, or in the operational space, which the manipulator has to follow in the execution of the assigned motion; a path is then a pure geometric description of motion. On the other hand, a trajectory is a path on which a timing law is specified, for instance in terms of velocities and/or accelerations at each point [Siciliano et al., 2009]. The goal of trajectory planning is therefore to generate the reference inputs to the motion control system which ensures that the manipulator executes the planned trajectories

The techniques for trajectory generation reported in the literature, for both one-dimensional and multidimensional trajectories, are classified according to whether the desired motion is defined assuming only initial and final points (point-to-point trajecto-

ries) or considering a set of intermediate via points (multipoint trajectories, also known as motion through a sequence of points) [Biagiotti and Melchiorri, 2008].

For the execution of a specific robot task, it is worth considering the main challenges of motion planning algorithms as follows.

#### 1.6.1.1 Path constraints

Trajectory planning in the operational space naturally allows the presence of path constraints to be accounted; these are due to regions of workspace which are forbidden to the manipulator, e.g., due to the presence of obstacles. Besides, a lot of papers in the literature mention the motion planning problem proposing some techniques to avoid the singularities in the workspace of the parallel robots [Dash et al., 2005; Reveles et al., 2016].

#### 1.6.1.2 Continuity constraints

Continuity of trajectories is a crucial constraint since discontinuities in the trajectories can generate discontinuous control torques and thus, lead to undesired behavior of the mechanical structure such as vibrations or instabilities [Bennehar et al., 2014]. Consequently, trajectory generation should take into consideration continuity constraints. In particular, it would be desirable to obtain trajectories with continuous joint accelerations, so that the absolute value of the jerk (i.e. the derivative of the acceleration) keeps bounded. Limiting the jerk is very important, because high jerk values can wear out the robot structure, and heavily excite its resonance frequencies [Gasparetto and Zanotto, 2008]. Moreover, low-jerk trajectories can be executed more rapidly and accurately. Various splines including the cubic spline, trigonometric spline, polynomial spline, quintic spline, B-spline are proposed for jerk-limited motion profiles [Perumaal and Jawahar, 2012]. S-curve motion is another approach for jerk-limited motion.

#### 1.6.1.3 Dynamic constraints

The minimal requirement for a manipulator is the capability to move from an initial posture to a final assigned posture. The transition should be characterized by motion laws requiring the actuators to exert joint generalized forces which do not violate the saturation limits, do not excite the typically modelled resonant modes of the structure, and thereby do not exceed the maximum allowable allowable robot velocity and acceleration [Siciliano et al., 2009; Gasparetto and Zanotto, 2008]. It is then necessary to devise planning algorithms that generate suitably smooth trajectories.

#### 1.6.1.4 Optimal trajectories

The algorithm should generate a trajectory which, in respect to the above general requirements, is also capable of optimizing some performance index along the trajectory.

The most significant optimality criteria are: (i) minimum execution time, (ii) minimum energy (or actuator effort) [Zhang and Ming, 2019], and (iii) minimum jerk. Besides the aforementioned approaches, some hybrid optimality criteria have also been proposed (e.g. time–energy optimal trajectory planning, time-jerk optimal planning) [Gasparetto and Zanotto, 2008].

### 1.6.2 Main challenges in control of PKMs

Control of PKMs is often considered in the literature as a challenging task. Indeed, PKMs are complex nonlinear systems known for their abundant uncertainties, parameters variation and actuation redundancy. From a control point of view, the following challenges should be taken into account in the control design for PKMs in order not to deteriorate their dynamic performance [Chemori, 2017].

#### 1.6.2.1 Nonlinear complex dynamics

Parallel manipulators are known by their highly nonlinear dynamics mainly emerging from their closed-loop kinematic structure with the presence of several passive joints. According to [Natal et al., 2014], operating at high accelerations can increase considerably the effect of nonlinearities in PKMs leading to mechanical vibration issues. These vibrations may produce mechanical damages and cause significant loss of precision. Furthermore, their kinematic configuration gives rise to coupled dynamics that require careful control synchronization between the actuators. Any failure of one of the actuators may affect the whole structure of the robot. Thus, the conventional linear control approaches may fail to guarantee the safety and stability of PKMs in such critical operational conditions. Accordingly, the need for advanced nonlinear control strategies arises to minimize the nonlinearities effects while fulfilling the requirements of high accuracy under high acceleration conditions.

#### 1.6.2.2 Structured and unstructured uncertainties

Uncertainties are simply the differences between the formulated dynamic model and the real system [Saied, 2019]. They can be classified into two categories: structured and unstructured. The first type can appear in the form of inaccurate knowledge of the dynamic parameters (e.g, masses and inertia) or parameter variations due to the operating environment (handled payload in P&P tasks, contact forces with a workpiece in machining operations, etc.). In contrast, the second type includes uncertainties resulting from wear of the system components, geometric manufacturing errors, non-modeled phenomena, modeling simplifications (i.e., neglecting the actuators' dynamics or friction), and sensor noise [Siciliano and Khatib, 2016; Chemori, 2017]. Taking these uncertainties into account during the control design phase can ensure high dynamic performance, high accuracy in the

developed task, and robustness towards changes in dynamic operating conditions such as high-speed motions and payload variations.

#### 1.6.2.3 Actuation redundancy

The PKM is called redundantly actuated (RA-PKM) when the number of its actuators is greater than the number of DOFs of its end-effector [Merlet, 2005]. In other words, PKMs with actuation redundancy (AR) are characterized by the following property [Corbel et al., 2010a]: "*For a given wrench (force and moment) acting on the moving platform and a given pose, there is an infinite number of corresponding joint forces/torques.*"

The difference between the number of actuators of the PKM and its number of DOFs is called the degree of actuation redundancy. An illustrative example of actuation redundancy for a 3-DOFs PKM through adding one more kinematic chain with its respective actuator is shown in Figure 1.26. In this example, the degree of actuation redundancy is one (four actuators and three DOFs, (two translational movements along the $x$ and $y$ axes, and one rotational movement, $\phi$, around the $z$ axis)).

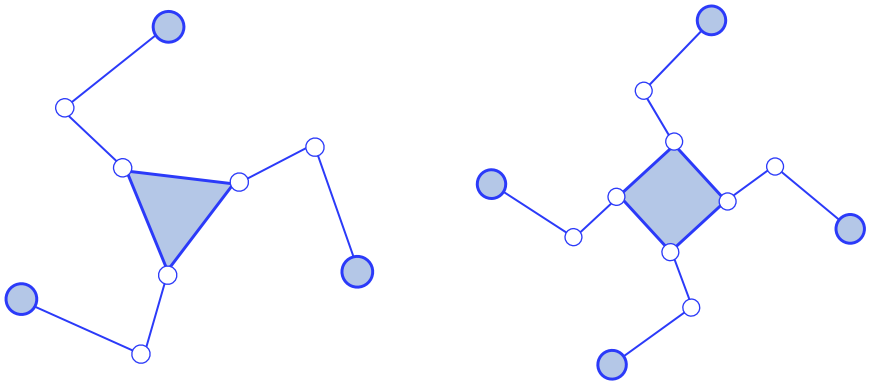
Actuation redundancy, which is a peculiarity of PKMs, offers several advantages for these manipulators described as follows [Müller, 2009]:

- Elimination of singularities and thus enlarging the usable workspace.
- Increasing the dynamic capabilities of the PKM.
- Increasing the homogeneity of the force transmission and the manipulator stiffness.
- Better load distribution reducing the power consumption of the individual actuators.

However, actuation redundancy allows for internal prestresses, due to the antagonistic forces of the redundant actuation, without generating end-effector forces (i.e. without affecting the mobility of a parallel manipulator). These internal prestresses can be purposefully exploited for secondary tasks, such as backlash avoidance [Muller, 2005] and manipulator stiffness control within the workspace [Muller, 2006]. According to [Hufnagel and Muller, 2012], the antagonistic forces can be produced by (i) non-synchronized independent control of the actuators, (ii) geometric uncertainties, and (iii) measurement errors. If not accordingly taken into account in the control design, these generated antagonistic forces may cause damages to the mechanical structure of the robot. For this reason, the use of mere position control is no longer possible in presence of actuation redundancy [Wang and Gosselin, 2004].

## 1.7 Dynamic modeling of parallel manipulators

Dynamic modeling plays a vital role in the study of parallel manipulators. On the one hand, it is necessary for simulation purposes to analyse the behavior of the robot under



(a) Non-redundantly actuated 3-DOF PKM (b) Redundantly actuated 3-DOF PKM

Figure 1.26 – Illustration of actuation redundancy in PKM through additional kinematic chain.

control; this ensures the stability of the closed-loop and the fulfillment of the physical limitations of the system. The obtained simulation results can help researchers to propose improvements to PKM in terms of mechanical design or control synthesis. On the other hand, most modern control strategies require full or partial knowledge of the dynamics of the PKM to be controlled in order to improve its tracking performance and achieve the specified control objective.

The dynamic analysis of PKMs has been extensively investigated in the literature with significant complexity due to the closed-loop nature of these manipulators, involving several kinematic chains. Most of the proposed approaches for the dynamic analysis of PKMs use the following procedure [Merlet, 2005; Taghirad, 2013; Briot et al., 2015]:

1. First, disconnecting the kinematic chains (limbs) from the moving platform; this results in a tree structure consisting of multiple open-loop chains (see Figure 1.27). This process corresponds to removing the loop constraints in the system.
2. Second, deriving the dynamics of each limb separately using their local generalized coordinates (position vector of the limb joints, the Cartesian positions of the center of mass of the moving platform and its orientation angles).
3. Finally, combining the local models to obtain the final dynamic formulation for the whole manipulator in terms of its generalized global coordinates.

The main existing methodologies to derive the dynamic model of PKMs are based on (i) the Newton-Euler formulation [Gosselin, 1996; Dasgupta and Choudhury, 1999; Borchert et al., 2015], (ii) Euler-Lagrange formulation [Cheng et al., 2003; Ahmadi et al., 2008; Abdellatif and Heimann, 2009], and (iii) the third approach is virtual work principle/ D'Alembert principle [Codourey, 1998; Tsai, 2000; Zhao and Gao, 2009].

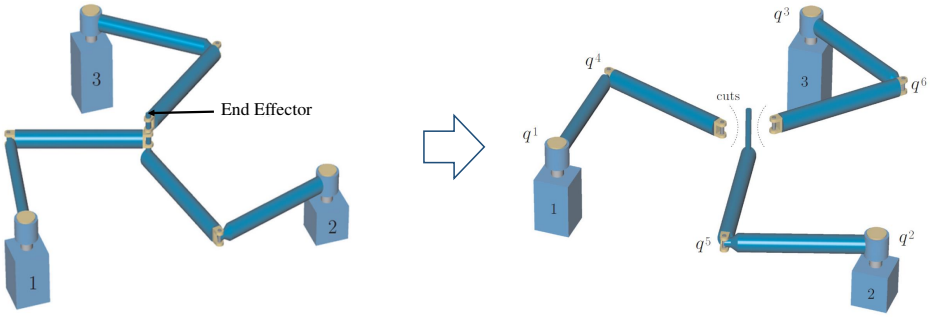



Figure 1.27 – Illustration of the loop constraints for a planar 2-DOF redundantly actuated PKM [Mueller, 2011]

### 1.7.1 Dynamic modeling representation

Dynamic model establishes the relationship between the generalized forces and torques of a robotic system and its corresponding positions,and/or orientations, velocities, and accelerations in the Cartesian and joint spaces. There are two types of dynamic models [Staicu, 2019]:

- *Forward Dynamics:* being given the set generalized forces and torques of the robot, we can compute the resulting motion of the end-effector and joints as a function of time. The representation of the forward dynamic model in terms of Cartesian space variables can be written as follows:

$$\ddot{X} = f(F, X, \dot{X}) \tag{1.1}$$

  where $X$, $\dot{X}$, and $\ddot{X} \in \mathbb{R}^m$ are the Cartesian position, velocity, and acceleration vectors of the end-effector, respectively, while $F \in \mathbb{R}^m$ is the external forces/torques vector applied to the end-effector. $m$ is the number of the robot end-effector DOFs.
  The forward dynamics expressed in terms of joint variables is presented as below:

$$\ddot{q} = f(\Gamma, q, \dot{q}) \tag{1.2}$$

  where $q$, $\dot{q}$, and $\ddot{q} \in \mathbb{R}^n$ are the position, velocity, and acceleration vectors of the actuated joints, respectively, with $n$ being the actuators number. $\Gamma \in \mathbb{R}^m$ is the actuated joint force/torque vector.

- *Inverse Dynamics:* given the position, velocity and acceleration vectors of the end-effector and actuated joints, we can determine the set of forces/torques that produce the corresponding motion. This description can be expressed in terms of Cartesian and joint space coordinates, respectively, by the following formula:

$$F = f(X, \dot{X}, \ddot{X}) \tag{1.3}$$

$$\Gamma = f(q, \dot{q}, \ddot{q}) \tag{1.4}$$

The forces/torques applied on the moving platform and those produced by the actuated joints are related to each other by means of the inverse Jacobian matrix of the manipulator $J_m \in \mathbb{R}^{n \times m}$ as follows:

$$F = J_m^T \Gamma \tag{1.5}$$

Hereafter, the inverse dynamic model will be detailed as it plays an essential role in the synthesis of model-based controllers.

**Representation of the inverse dynamic model in joint space**

The inverse dynamic formulation of any $m$-DOF PKM having $n$ actuators (such that $n \geq m$) can be represented in joint space as follows [Siciliano and Khatib, 2016]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \Gamma_d(t) = \Gamma \tag{1.6}$$

where

- $M(q) \in \mathbb{R}^{n \times n}$ is the total mass and inertia matrix.
- $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the Coriolis/Centrifugal forces matrix,
- $G(q) \in \mathbb{R}^n$ is the gravity vector,
- $\Gamma_d(t) \in \mathbb{R}^n$ gives the vector of external disturbances and unmodeled dynamics,
- $\Gamma(t) \in \mathbb{R}^n$ is the input torque vector.

This formulation is the standard formulation widely used for control design purposes, since most PKMs do not have sensors to measure the position of the mobile platform directly in the fixed coordinate system.

**Representation of the inverse dynamic model in Cartesian space**

The inverse dynamics can also be represented in Cartesian space (operational or task space) employing the following Jacobian transformations:

$$\begin{aligned} \dot{q} &= J_m \dot{X} \\ \ddot{q} &= J_m \ddot{X} + \dot{J}_m \dot{X} \end{aligned} \tag{1.7}$$

Substituting (1.5) and (1.7) in (1.6), the inverse dynamics can be expressed in Cartesian space as follows [Taghirad, 2013]:

$$M_x \ddot{X} + C_x \dot{X} + G_x + \Gamma_d(t) = F \tag{1.8}$$

where

- $M_x = J_m^{-T} M(q) J_m$ is the mass and inertia matrix expressed in Cartesian space,

- $C_x = J_m^{-T} M(q) \dot{J}_m + J_m^{-T} C(q, \dot{q}) J_m$ is the Coriolis/Centripetal matrix expressed in Cartesian space,
- $G_x = J_m^{-T} G(q)$ is the Cartesian gravitational force vector,
- $F \in R^m$ is the input force vector on the end-effector.

In case of redundant parallel manipulators, the direct Jacobian matrix $J$ can be computed from the inverse one using the Moore-Penrose pseudoinverse matrix which can be adopted when a system of equations is under-constrained (infinitely many solutions).

## 1.7.2 Properties of the dynamic model

As common for robotic manipulators, the inverse dynamic model of PKM (1.6) and the terms that constitute it, have some properties which are interesting for control design. These properties are the following [Lewis et al., 2003; Kelly et al., 2006; Taghirad, 2013]:

### 1.7.2.1 Property of mass and inertia matrix

**Property 1.7.1.** *The mass and inertia matrix $M(q)$ and its inverse $M(q)^{-1}$ are symmetric, positive-definite and bounded above and below as follows:*

$$\mu_1 I \leq M(q) \leq \mu_2 I \tag{1.9}$$

$$\frac{1}{\mu_2} I \leq M(q)^{-1} \leq \frac{1}{\mu_1} I \tag{1.10}$$

*where $\mu_1$ and $\mu_2$ are two positive scalars. $\mu_1$ and $\mu_2$ can be function of joint vector $q$ for some cases (for example if using prismatic joints) [Kelly et al., 2006]. Similarly, the boundedness property of $M(q)$ can be expressed as follows:*

$$m_1 \leq \|M(q)\| \leq m_2 \tag{1.11}$$

*where $\|.\|$ is the second norm of matrix. $m_1$ and $m_2$ are two positive scalars.*

*Furthermore, there exists a positive constant $K_M$ such that:*

$$\|M(x)z - M(y)z\| \leq K_M \|x - y\| \|z\| \quad \forall \quad x, y, z \in \mathbb{R}^n \tag{1.12}$$

*Note that this property is only valid for robots with revolute joints [Kelly et al., 2006].*

### 1.7.2.2 Property of Coriolis and centrifugal matrix

**Property 1.7.2.** *Coriolis and centrifugal matrix is bounded as follows:*

$$\|C(q, \dot{q})\dot{q}\| \leq K_{C1} \|\dot{q}\|^2 \tag{1.13}$$

*where* $K_{C1}$ *is a positive number independant of* $q$ *and can be function of* $q$ *in some cases. While* $||.||$ *is the second norm of a vector or a matrix.*

*In addition, the matrix* $S(q,\dot{q}) = \dot{M}(q) - 2C(q,\dot{q})$ *or* $S(q,\dot{q}) = \frac{1}{2}\dot{M}(q) - C(q,\dot{q})$ *is a skew-symmetric matrix fulfilling:*

$$x^T S(q,\dot{q})x = 0 \qquad \forall \quad x \in \mathbb{R}^n \tag{1.14}$$

$$\dot{M}(q) = C(q,\dot{q}) + C^T(q,\dot{q}) \tag{1.15}$$

*Furthermore, there exist positive numbers* $K_{C1}$ *and* $K_{C_2}$ *such that*

$$||C(x,z)w - C(y,v)w|| \leq K_{C1}\,||z-v||\,||w|| + K_{C2}\,||x-y||\,||w||\,||z|| \tag{1.16}$$

*for all vector* $x,y,z,w,v \in \mathbb{R}^n$

#### 1.7.2.3 Property of gravity vector

**Property 1.7.3.** *A bound on the gravity vector is given as follows:*

$$||G(q)|| \leq g_0 \tag{1.17}$$

*where* $g_0$ *is a positive constant that can be function of* $q$ *in some cases (for example if using prismatic joints).*

*Moreover, there exists a positive constant* $K_G$ *such that*

$$||G(x) - G(y)|| \leq K_G\,||x-y|| \quad \forall \quad x,y \in \mathbb{R}^n \tag{1.18}$$

*This property is only valid for a robot using revolute joints.*

#### 1.7.2.4 Property of the disturbance term

**Property 1.7.4.** *The dynamic equation of PKM* (1.6) *has a disturbance term* $\Gamma_d(t)$ *which represents a general class of external disturbances and inaccurately modeled dynamics. We shall assume that this term is bounded as follows:*

$$||\Gamma_d(t)|| \leq d \tag{1.19}$$

*where* $d$ *is a scalar constant that be determined.*

#### 1.7.2.5 Linear formulation property of the dynamics

**Property 1.7.5.** *The manipulators' dynamics is characterized by a fundamental property that is essential for model-based adaptive controllers. Namely, it is the linearity of the dynamics with respect to parameters such as inertia and masses [Craig et al., 1987; Siciliano and Khatib, 2016]. Consider the standard form of the inverse dynamics of a m-DOF PKM given by (1.6), all constant parameters in the dynamic model are considered as coefficients of known functions (linear and nonlinear) of the generalized coordinates. The external disturbances $\Gamma_d(t)$ are excluded from the linear reformulation of the dynamics since they are not modeled and cannot be written in a linear form of the parameters. Therefore, (1.6) can be expressed in a linear form as follows:*

$$W(q,\dot{q},\ddot{q})\Phi + \Gamma_d(t) = \Gamma(t) \tag{1.20}$$

*where $W(q,\dot{q},\ddot{q}) \in \mathbb{R}^{n\times p}$ is the matrix of the known functions called regressor, and $\Phi \in \mathbb{R}^p$ is the vector the parameters to be estimated.*

*It should be noted that not all parameters have to be estimated because of the good knowledge that one may have about some parameters. Accordingly, two sets of parameters can be distinguished: the set of known parameters and the other of unknown, uncertain or time-varying parameters [Ortega and Spong, 1989]. Therefore, the reformulation in (3.54) can be rewritten in general form as follows:*

$$W_n(q,\dot{q},\ddot{q})\Phi_n + W_u(q,\dot{q},\ddot{q})\Phi_u + \Gamma_d(t) = \Gamma(t) \tag{1.21}$$

*where $W_n(q,\dot{q},\ddot{q}) \in \mathbb{R}^{n\times p_n}$ and $W_u(q,\dot{q},\ddot{q}) \in \mathbb{R}^{n\times p_u}$ are partial regression matrices. $\Phi_n \in \mathbb{R}^{p_n}$ is the set of known parameters, while $\Phi_u \in \mathbb{R}^{p_u}$ is the set of parameters considered for real-time estimation.*

## 1.8 Overview on motion control strategies for parallel robots

In the literature, several control approaches have been proposed for motion control of parallel robots [Paccot et al., 2009]. Many of them have been taken from control schemes of serial robots since they share similarities in the dynamics modeling. Control solutions for robots can be classified into two main sets: kinematic control strategies and dynamic control strategies [Liu et al., 2001; Shang and Cong, 2010].

In kinematic control strategies, to simplify the control problem, parallel manipulators are supposed to be decoupled into a group of single-axis systems, so they can be controlled by a group of individual controllers. This type of control does not require any knowledge about the controlled system dynamics, so the complex computation of dynamics can be

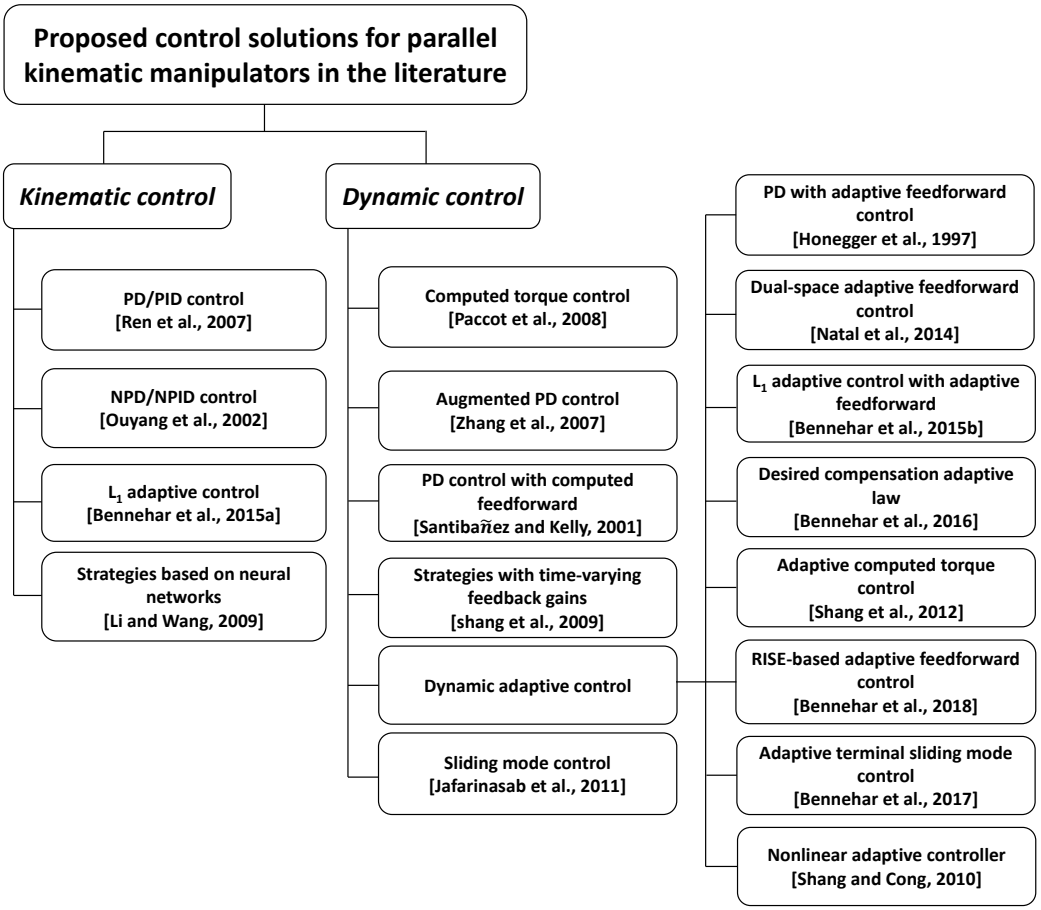



Figure 1.28 – Classification of the main proposed control solutions for parallel manipulators in the literature.

avoided and the controller design can be simplified considerably. However, these controllers do not always produce high performance, and there is no guarantee of stability at high-speed motions.

Dynamic control strategies make use of the whole or some parts of the inverse dynamic model in their closed-loop design, which compensates for the potentially non-negligible effects of the nonlinear dynamics of the manipulator. Thus, in most cases, the obtained tracking performance using dynamic controllers exceeds that of kinematic controllers [Shang and Cong, 2010; Taghirad, 2013]. Figure 1.28 presents a classification of the most relevant control schemes proposed in the literature for PKMs. All the listed controllers will be briefly described in this section except for those based on the Robust Integral of the Sign of the Error (RISE) and Desired Compensation Adaptive Law (DCAL) control, which will be described in more detail in Chapter 3.

## 1.8.1 Kinematic control strategies

### 1.8.1.1 PD/PID control

The Proportional-Integral-Derivative (PID) controller [Ziegler et al., 1942] is the most commonly used control scheme in the industry. This is mainly because of its simplicity and easy implementation. This model independent control strategy provides a relatively adequate performance for most industrial applications. However, the performance of this controller may decrease considerably if the system operates at high accelerations. The most common way to implement a PID controller on parallel robots is in joint space, since most of them are equipped only with encoders measuring the joint positions of the actuators. The PID control equation for parallel manipulator with $n$ actuators [Ren et al., 2007] is expressed in joint space form as follows:

$$\Gamma(t) = K_p e(t) + K_d \dot{e}(t) + K_i \int_{t_0}^{t} e(\tau) d\tau \tag{1.22}$$

Where $K_p$, $K_d$, and $K_i \in \mathbb{R}^{n \times n}$ are positive definite diagonal matrices for the proportional, derivative, and integral actions, respectively. The tracking error in joint space $e(t) = q_d(t)$ - $q(t)$ is defined as the difference between the desired trajectory in joint space $q_d(t) \in \mathbb{R}^n$, and the measured one $q(t) \in \mathbb{R}^n$. $\dot{e} \in \mathbb{R}^n$ denotes the joint velocity errors. Nevertheless, for the particular case where the pose of the end-effector could be directly measured, a Cartesian-space controller is often recommended. The expression of a Cartesian-space PID controller [Natal et al., 2014] can be written as follows:

$$F(t) = K_p e_x(t) + K_d \dot{e}_x(t) + K_i \int_{t_0}^{t} e_x(\tau) d\tau \tag{1.23}$$

Where $e_x(t) = X_d(t)$ - $X(t)$ represents the trajectory tracking error in Cartesian space, being $X_d(t) \in \mathbb{R}^m$, the desired trajectory, and $X(t) \in \mathbb{R}^m$ the measured one, $K_p$, $K_d$, and $K_i$ $\in \mathbb{R}^{m \times m}$ are positive definite feedback gains. In this case of Cartesian-space PID, the computed control effort is the force $F(t)$ instead of $\Gamma(t)$. Thus, the actual control law to be applied to the actuators should be computed. As shown above in the relationship (**??**), the end-effector force vector $F(t)$ and the actuator torques $\Gamma(t)$ are related to each other by means of the inverse Jacobian matrix $J_m$. If the PKM is non-redundantly actuated, $\Gamma(t)$ can be obtained by using the inverse of $J_m$. Otherwise, if it is redundantly actuated, the pseudoinverse of $J_m$, denoted by $J_m^+$ can be used:

$$\Gamma(t) = (J_m^+)^T F(t) \tag{1.24}$$

where $J_m^+ = (J_m^T J_m)^{-1} J_m^T$ is the pseudoinverse of $J_m$.

#### 1.8.1.2 Nonlinear PD/PID control

Nonlinear PD/PID (NPD/NPID) controller arise from the need to improve the response and tracking performance of conventional PID controller. It shares the same structure of the standard PID with the use of nonlinear time-varying feedback gains instead of fixed gains [Seraji, 1998]. The control equation of a NPID controller applied to a robotic manipulator can be written in joint space as:

$$\Gamma(t) = K_p(.)e(t) + K_d(.)\dot{e}(t) + K_i(.)\int_{t_0}^{t} e(\tau)d\tau \tag{1.25}$$

Where $K_p(.)$, $K_d(.)$, and $K_i(.) \in \mathbb{R}^{n\times n}$ are the nonlinear time-varying feedback gains for the proportional, derivative, and integral actions, respectively. Those nonlinear gains are automatically adjusted according to a set of rules; they can be a function of the system errors, control input, and other parameters. The generated values of the gains can handle the instantaneous operating conditions to reduce the error as much as possible. When the error between the desired and actual values of the controlled variable is large, the values of the gains increase to generate a large corrective action to quickly drive the system toward its goal. As the error decreases, the gains are automatically reduced to avoid excessive oscillations and large overshoots in the response. Several methodologies have been proposed in the literature to adjust the nonlinear time-varying feedback gains [Seraji, 1998; Jiang and Gao, 2001; Ouyang et al., 2002], among others. In [Su et al., 2004], the NPID control was implemented on a 6-DOF parallel manipulator offering several advantages over the fixed-gain PID controller. These improvements include guaranteed stability, better tracking performance and improved ability to reject external disturbances. Furthermore, in [Ouyang et al., 2002], a NPD control design obtained accurate tracking of the desired trajectory for a 2-DOF parallel manipulator compared to a linear PD controller. However, like the classical PID controller, NPD/NPID is a decentralized control strategy in which neither the nonlinear coupled dynamics nor the closed-chains constraints are considered. Consequently, its performance deteriorates especially in high-speed tasks due to non-trivial nonlinearities.

#### 1.8.1.3 $\mathscr{L}_1$ adaptive control

$\mathscr{L}_1$ adaptive control was introduced for the first time in [Cao and Hovakimyan, 2006a,b]. This control scheme is inspired from the Model Reference Adaptive Control (MRAC) but decoupling the estimation and control loops which enables fast adaptation while guaranteeing the robustness of the closed-loop system. The first experimental implementation of $\mathscr{L}_1$ adaptive control strategy on a PKM was reported in [Bennehar et al., 2015a]. The control law consists of the combination of two separate terms, a fixed state-feedback term and an adaptive term that compensates partially for the nonlinearities of the system, that is

$$\Gamma(t) = A_m r(t) + \Gamma_{ad}(t) \tag{1.26}$$

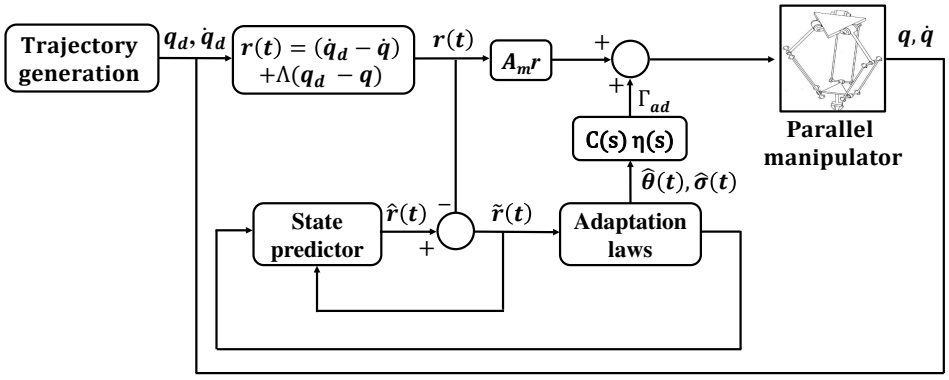



Figure 1.29 – Block diagram of the $\mathscr{L}_1$ adaptive control for PKM.

where $A_m \in \mathbb{R}^{n\times n}$ is a Hurwitz matrix characterizing the transient response of the system, $r(t) \in \mathbb{R}^n$ is the joint-space combined error, $r(t) = \dot{e} + \Lambda e$, with $\Lambda \in \mathbb{R}^{n\times n}$ being a symmetric positive-definite matrix. $\Gamma_{ad}(t) \in \mathbb{R}^n$ is the adaptive control signal designed such that

$$\Gamma_{ad}(s) = C(s)\hat{\eta}(s) \tag{1.27}$$

where $\hat{\eta}(s)$ is the Laplace transform of $\hat{\eta}(t) = \hat{\theta}(t)\|r(t)\|_{\mathscr{L}_\infty} + \hat{\sigma}(t)$ and $C(s)$ is a low-pass filter. $\hat{\theta}(t)$ and $\hat{\sigma}(t)$ estimates the nonlinear functions $\theta(t)$ and $\sigma(t)$ that represent all the nonlinearities and disturbances of the system, respectively. The procedure of such estimation is detailed in [Hovakimyan and Cao, 2010; Bennehar et al., 2015a]. Figure 1.29 shows the general scheme of the $\mathscr{L}_1$ adaptive controller implemented on parallel manipulators. Using projection-based adaptation law, the boundedness of the estimated parameters is ensured as well as the convergence of $r(t)$ to zero [Bennehar et al., 2015a].
The experimental implementation of this control scheme on a 4-DOF PKM shows better tracking performance than a classical PD controller thanks to the compensation of the nonlinearities in the $\mathscr{L}_1$ adaptive controller [Bennehar et al., 2015a].

#### 1.8.1.4 Strategies based on artificial neural networks

Artificial Neural Networks (ANNs) are known by their powerful universal approximation features. They have thus been applied extensively in several robotics fields such as pattern recognition, learning, signal processing, modeling identification, and control [Jiang et al., 2017]. Thanks to their learning ability, artificial neural networks are mostly used to approximate the dynamics of the manipulator. Then, the learned dynamics can be included in the control scheme to compensate for the uncertainties and disturbances.

Several works can be found in the literature regarding the application of neural networks in control of PKMs. One can mention the decentralized Cartesian space PID controller with an artificial neural network term that has been proposed in [Li and Wang, 2009].

The main motivation of this work is to improve the tracking capabilities of a 2-DOF redundantly actuated parallel manipulator. The provided simulation results demonstrated the superiority of the augmented controller with respect to the original PID. The maximum errors were significantly reduced since the additional neural network term accounted for the nonlinear dynamics of the manipulator.

Moreover, in [Escorcia-Hernández et al., 2020a], a RISE (Robust Integral of the Sign Error) controller with an adaptive feedforward compensation term based on BSNNs was proposed to regulate the positioning of a Delta robot. BSNNs make an online approximation of the Delta robot dynamics and integrate it into the control loop. The BSNNs' functions are bounded according to the extreme values of the desired joint space trajectories that are the BSNNs' inputs. In order to evaluate the effectiveness of the proposed control scheme with respect to the standard RISE controller, numerical simulations for different case studies under different scenarios were performed. The obtained results confirm that the use of the BSNNs as a feedforward compensation term is a suitable alternative to improving the trajectory tracking in PKMs even if the system is dealing with parametric uncertainties as sudden changes in the payload. Moreover, the dynamic approximation of the BSNNs is good enough according to the comparison with the nominal Feedforward of a RISE Feedforward controller.

## 1.8.2 Dynamic control strategies

### 1.8.2.1 Computed torque control

Computed Torque (CT) control, also known in some literature as Inverse Dynamic Control (IDC) [Taghirad, 2013], is a widespread control strategy for robotic manipulators because of the fact that it can significantly linearize and decouple dynamic formulation of the closed-loop error dynamics. The joint-space control law applied to any manipulator can be expressed as follows [An et al., 1987]:

$$\Gamma(t) = M(q)(\ddot{q}_d + K_p e + K_d \dot{e}) + C(q,\dot{q})\dot{q} + G(q) \tag{1.28}$$

where $K_p$, $K_d \in \mathbb{R}^{n\times n}$ are positive definite diagonal feedback gain matrices for the proportional, and derivative control actions, respectively. $e(t)$, $\dot{e}(t) \in \mathbb{R}^n$ are the joint position and velocity tracking errors, respectively. Substituting (1.28) into the dynamics of the manipulator in (1.6) (without considering the external disturbances) leads to the following linear closed-loop error system equation:

$$\ddot{e} + K_d \dot{e} + K_p e = 0 \tag{1.29}$$

in which $\ddot{e}$ represents the acceleration error in joint space. As it can be seen in (1.29), the stability of the final obtained system can be proven as for classical linear systems. Moreover, if the feedback gains are chosen appropriately, the tracking error can converge

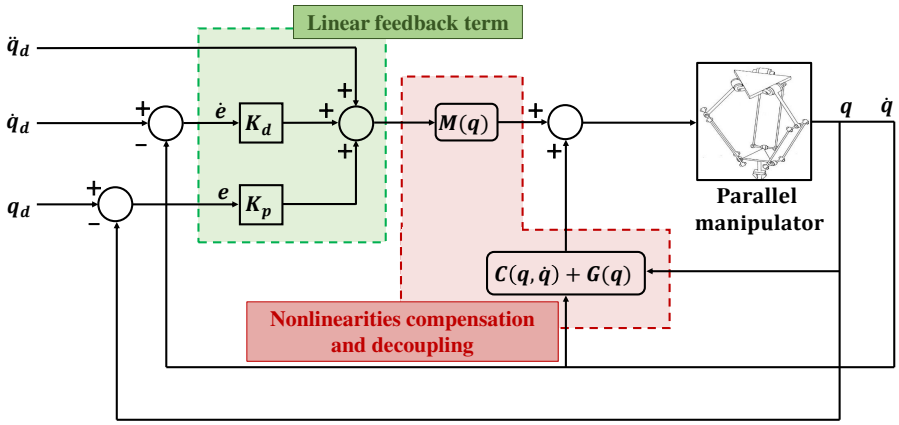



Figure 1.30 – Block diagram of the standard computed torque controller in joint space.

quickly towards zero with a suitable steady-state performance. However, note that for a good tracking performance, an accurate model of the system is required. Furthermore, this technique is computationally intensive in terms of the online computations needed to carry out the closed-loop control structure, and the use of measured variables such as joint positions and velocities may yield noise that can adversely affect the system performance. A block diagram illustrating the joint space computed torque controller (1.28) is shown in Figure 1.30. In [Paccot et al., 2008], a Cartesian space CT control is implemented and validated on the parallel Orthoglide robot using a fast exteroceptive measure of the end-effector pose in the feedback loop. The control scheme was reduced to its simplest expression leading to a better accuracy than the joint space CT control (because of no use of the kinematic model in Cartesian space control).

#### 1.8.2.2 Augmented PD control

The augmented PD (APD), also known as PD+ controller, is one of the simplest control schemes that guarantee the control objective globally. It is mainly composed of a PD feedback term plus the inverse dynamic model of the controlled system. Unlike the CT control scheme, the inertia matrix in APD is outside of the PD feedback loop yielding a faster computation process [Paden and Panja, 1988]. The joint-space expression of the APD controller is established as follows [Zhang et al., 2007]:

$$\Gamma(t) = M(q)\ddot{q}_d + C(q,\dot{q})\dot{q}_d + G(q) + K_p e + K_d \dot{e} \tag{1.30}$$

Where $K_p$, $K_d \in \mathbb{R}^{n\times n}$ are positive definite diagonal feedback gain matrices and $e(t)$, $\dot{e}(t)$ $\in \mathbb{R}^n$ are the tracking errors of the joint positions and velocities, respectively. One can clearly observe that the APD controller compensates for the effect of the full nonlinear dynamics evaluated using the measured and desired trajectories. The feedback PD control

term guarantees the global asymptotic stability. However, an online computation of the nonlinear functions of the dynamic model is required for this controller as well as an accurate knowledge of the manipulator dynamic parameters in order to achieve the desired performance.

#### 1.8.2.3 PD control with computed feedforward

PD control with computed feedforward consists of using the full inverse dynamic model (similarly to the previous controller) to compensate the effect of nonlinearity but within an offline-computation mode. In effect, the inverse dynamic model is evaluated with the desired trajectories instead of the measured and estimated ones since the sensors used to measure the actual joint variables may incorporate noise into the measurements, which may impair the robot performance. Moreover, there is no need to develop velocity and acceleration observers for such control strategy since all the dynamic computations depend on the desired trajectory. This control technique has been widely applied in the control of robots owing to its simplicity, asymptotic stability, and easy implementation, as all nonlinear dynamic terms are calculated before execution of the robot. The joint space control law can be formulated as follows [Santibañez and Kelly, 2001]:

$$\Gamma(t) = M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) + K_p e + K_d \dot{e} \tag{1.31}$$

where $K_p$, $K_d \in \mathbb{R}^{n\times n}$ are positive definite diagonal feedback gain matrices for the proportional, and derivative control actions, respectively. $e(t)$, $\dot{e}(t) \in \mathbb{R}^n$ are the joint position and velocity tracking errors, respectively. A modified version of the standard PD control with computed feedforward named "Dual-Space Control" was developed in [Natal et al., 2012]. In this controller, the dynamic model part is formulated in both coordinate spaces, Cartesian and joint and not only in one as in the original formulation. It was implemented on a RA-PKM called R4, showing a good tracking performance for high-speed pick-and-place tasks compared to the simple Cartesian PID controller. In [Saied et al., 2018], the PD control with computed feedforward was augmented by the actuator and friction dynamics evaluated also based on the reference trajectories. The idea behind this modification is to compensate the effects of the aforementioned dynamics in addition to the high nonlinearities existing in PKMs. It was experimentally implemented on a 4-DOF PKM named VELOCE. The results validated that model-based controllers relying on more comprehensive dynamics outperform simplified controllers in terms of tracking performance when operating in real-time conditions.

#### 1.8.2.4 Strategies with time-varying feedback gains

Control strategies with time-varying feedback gains are mainly based on conventional model-based strategies with the main difference that the linear feedback loop is replaced

by a nonlinear time-varying one. This proposition is motivated by the advantages of nonlinear gains compared to fixed ones. The nonlinear time-varying gains endow the controller with a better ability to reject external disturbances, making it less sensitive to the dynamic changes in operating conditions. In a similar manner to the NPD controller presented above, several dynamic control schemes for parallel manipulators have been improved in the literature.

[Shang et al., 2009] proposed an Augmented Nonlinear PD (ANPD) controller to improve the trajectory tracking accuracy for a redundantly actuated parallel robot. The ANPD controller is designed based on the conventional APD and the use of nonlinear feedback gains instead of the usual constant ones. The stability of the parallel manipulator system with the proposed ANPD controller is proven using the Lyapunov stability guarantee asymptotic convergence to zero of both the tracking error and error rate. The superiority of the ANPD controller over the conventional APD controller is verified through trajectory tracking experiments of a 2-DOF RA-PKM. In [Shang and Cong, 2009], the conventional CT controller was revisited with nonlinear feedback gains. The same procedure in [Shang et al., 2009] was followed to demonstrate the relevance of using nonlinear time-varying gains instead of fixed ones. The proposed Nonlinear CT (NCT) controller inherits merits from the conventional CT controller, such as simple structure and clear physical meaning of each control parameter, it also owns the good performances of the NPD algorithm in elimination of the nonlinear factors such as the modeling error and the nonlinear friction. Stability proof of the proposed controller based on Lyapunov theory was provided. Real-time experiments on a 2-DOF RA-PKM show a net superiority of the proposed NCT controller with respect to the conventional one in terms of tracking performance. However, this NCT controller inherits from the original CT controller its main drawback. This is because the dynamic compensation is calculated based on the dynamic model with the fixed dynamic parameters, but the parameters are variable during the trajectory tracking. Thus, the dynamic compensation in the NCT controller cannot achieve good compensation performance.

#### 1.8.2.5 Dynamic adaptive control

Dynamic adaptive controllers arise because, in reality, the dynamic parameters of a manipulator are often difficult to know precisely, and most of the time, some of these parameters may vary depending on the assigned task. Many of the control schemes belonging to this subcategory are improvements of dynamic controllers with fixed parameters. These adaptive control strategies rely on the linear-in-the-parameters property of the inverse dynamics of the manipulator, where an adaptation loop is added to the control law to estimate in real time the unknown, uncertain or time-varying parameters of the manipulator. In 1989, a tutorial on the existing adaptive control schemes for rigid serial manipulators was reported in [Ortega and Spong, 1989] showing the global convergence of the error

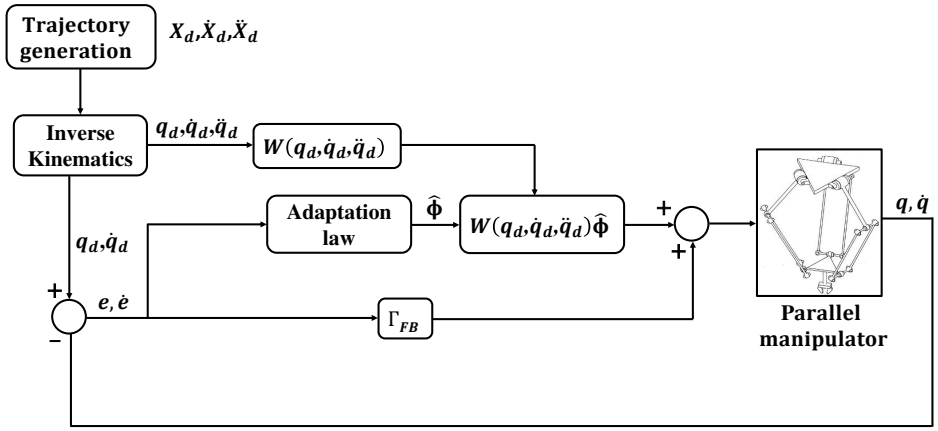



Figure 1.31 – Block diagram of the general adaptive feedforward control law in joint space.

using the adaptation law in the controller, while ensuring boundedness of the estimated parameter. The parameter adaptation law can be driven by the parameter estimation error or by the position tracking error.

- **Adaptive feedforward control:**

For robotic manipulators, the formulation of the Adaptive FeedForward Control (AFFC) equation in joint space can be expressed as follows:

$$\Gamma(t) = Y(q_d, \dot{q}_d, \ddot{q}_d)\hat{\Phi} + \Gamma_{FB} \tag{1.32}$$

where $\hat{\Phi}$ is the vector of the dynamic parameters to be estimated, $Y(q_d, \dot{q}_d, \ddot{q}_d)$ is the regressor matrix, and $\Gamma_{FB}$ is any feedback controller. It should be noted that the AFFC strategies use the desired joint trajectories instead of the measured ones in the regressor matrix and the parameters adaptation rule, which reduces the computational time and the sensitivity to measurement noise. Figure 1.31 illustrates the general block diagram of the AFFC scheme. Most of the time, the parameter adaptation law is derived using the output tracking error. Several AFFC schemes have been proposed in the literature for the control of parallel manipulators. The experimental results ensured the boundedness and convergence of the estimated parameters to the real values by means of an appropriate adaptation law. It has been validated that the adaptive control laws outperform the conventional controllers in terms of accuracy and robustness towards parameters variation. Between the most relevant, one can mention: PD control with adaptive feedforward being successfully implemented in real-time to control the positioning of a 6-DOF PKM called Hexaglide [Honegger et al., 1997] and recently on VELOCE PKM [Saied et al., 2019b]. Dual-space adaptive feedforward control [Natal et al., 2014], augmented $\mathscr{L}_1$ adaptive control with adaptive feedforward [Bennehar et al., 2015b], Desired Compensation Adaptation Law [Bennehar et al., 2016], adaptive terminal sliding mode control [Bennehar et al., 2017], and Robust Integral of the Sign of the Error (RISE)-based adaptive feedforward control [Bennehar et al., 2018].

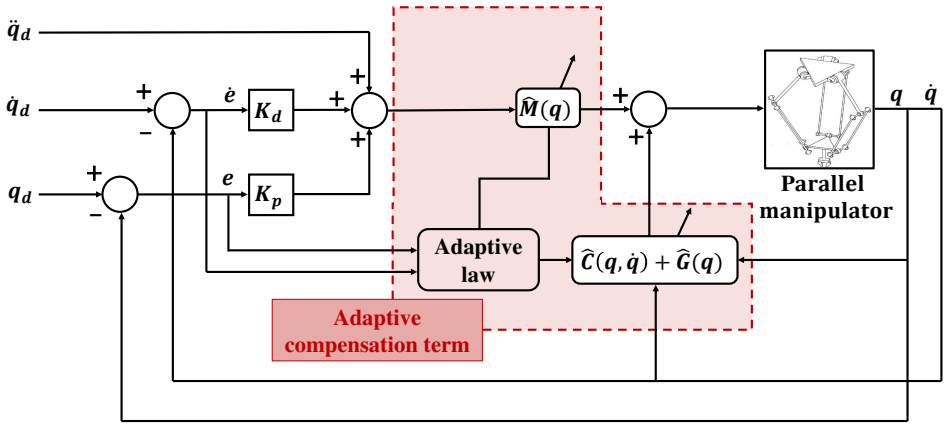



Figure 1.32 – Block diagram of adaptive computed torque control in joint space.

- **Other dynamic adaptive control**

A computed-torque-based adaptive controller was proposed in [Craig et al., 1987] for the control of serial manipulators. The first step of the control design in this work is to consider a computed torque control law with uncertain parameters. Then, the error equation resulting from the application of this control law is obtained. Based on a thorough stability analysis using the Lyapunov theory, an adaptation law for the parameters' estimation is derived. The proposed adaptive controller in conjunction with the adaptation law guarantee that the tracking errors vanish and that the estimated parameters converge to their best steady-state values. The block diagram depicted in Figure 1.32 clarifies the principle of such control strategy. This control scheme was applied in [Shang et al., 2012] to a 2-DOF RA-PKM. The obtained performance was compared to a classical CT control. The adaptive controller shows a net superiority and allows to estimate the system dynamic parameters. However, like its non-adaptive counterpart, adaptive CT control has the drawback of using measured velocities and accelerations, which are computationally intensive.

A nonlinear adaptive controller in the task space was developed in [Shang and Cong, 2010] for the trajectory tracking of a 2-DOF redundantly actuated parallel manipulator. The proposed control law includes an adaptive dynamics compensation term, an adaptive friction compensation term and error elimination items. The parameter update law is derived with the gradient descent algorithm aiming to minimize the performance index composed of the trajectory tracking error and the error rate. The experiment results show that the proposed nonlinear adaptive controller can get better trajectory tracking accuracy of the end-effector, comparing with the APD controller.

In [Natal et al., 2014], a Cartesian space control strategy called Dual-Space Adaptive Control is proposed to control RA-PKMs. The proposed controller is based on the dual-space feedforward controller and an adaptive control scheme. The dual-space feedforward

controller consists basically in a PID in the operational space augmented with a feedforward of both the desired Cartesian and articular accelerations to improve its tracking performance. The most important characteristic of the proposed dual-space adaptive Control is its capability of taking into consideration the dynamics of the system and of estimating its parameters automatically in real time. The proposed controller was experimentally validated on the R4 parallel manipulator. Real-time experiments showed that the proposed dual-space adaptive controller significantly improves the tracking performance and and allows the tracking of extremely fast trajectories.

#### 1.8.2.6 Sliding mode control

Sliding mode control (SMC) is a robust nonlinear control strategy that is able to provide the desired performance of the control system despite the presence of disturbances and uncertainties [Shtessel et al., 2014]. The design of SMC schemes mainly consists of two fundamental steps; the first is to ensure the reachability phase and the second is to drive the system states to the equilibrium point. In the reachability phase, the controller drives the states to a stable domain called the sliding surface. Then, during the sliding phase, the controller maintains the states on the surface while sliding to the equilibrium point. The control law that is designed based on the sliding surface features a sign function which compensates for the effects of the bounded disturbances. The sliding surface and the control law of the conventional SMC are expressed as follows [Shtessel et al., 2014]:

$$s = kx_1 + x_2 \tag{1.33}$$

$$u = -sign(s) \tag{1.34}$$

where $s$ is the sliding surface, $x_1$, $x_2$ are the system states, $k$ is a positive gain, and $u$ is the control law. However, this standard SMC produces a discontinuous control signal which is not adequate for real-time implementation. Several techniques have been investigated in the literature to avoid or reduce these undesirable high-frequency oscillations, called chattering, such as Quasi-SMC, integration of sign function, continuous high-order SMC [Shtessel et al., 2014], etc.

Various SMC and SMC-based controllers have been proposed in the literature for parallel manipulators owing to their robustness features. [Jafarinasab et al., 2011] proposed to apply a SMC approach based on the full knowledge of the system dynamic in Cartesian space for the control of a 6-DOF PKM. To demonstrate the capabilities of the SMC, two simulation scenarios were conducted. In the first one, the exact parameters of the manipulator were used in the control law. The second scenario considered large uncertainties in the mass properties of the mobile platform of the parallel manipulator. It was concluded that SMC provides fast transient response and accurate trajectory tracking.

A cascade-control algorithm, consisting of an inner loop and an outer loop based on SMC, was proposed to realize the trajectory tracking control of a hydraulically driven 6-

DOF parallel manipulator [Guo et al., 2008]. This cascade controller is applied to separate the hydraulic dynamics from the mechanical part so that the designed controller considers not only the mechanical dynamics but also the hydraulic dynamics of the manipulator. The experimental results investigate the effectiveness of the proposed approach compared to a P controller with feedforward compensation.

Besides, a robust SMC approach with an active disturbance compensation was proposed in [Singh and Santhakumar, 2015] for the trajectory tracking control of a 3-DOF vertical planar PKM in the presence of parameter uncertainties and external disturbances. The disturbance vector includes disturbances due to the system dynamic variations namely payload and parameter variations, frictional effects and other unmodelled effects. The effectiveness and robustness of the controller is established through extensive numerical simulations and real-time experiments in the presence of the aforementioned disturbances.

A model-based super-twisting (ST) SMC strategy relying on the desired trajectory was proposed for PKMs in [Saied et al., 2021]. The proposed approach includes a feedforward dynamic compensator, the super-twisting feedback control, and a feedback stabilizing term. Real-time experiments were conducted on a Spider4, a 5-DOF RA-PKM, showing better global performances of the proposed controller at low and high dynamic operating conditions, compared to conventional computed-torque ST-SMC and standard PID with feedforward controllers.

In the previous SMC-based controllers, the dynamics of the parallel manipulator were partially or fully included in the closed-loop control, assuming that the system parameters are known and subject to uncertainties. Among these previous controllers, some have compensated for those uncertainties by designing disturbance observers. [Bennehar et al., 2017] proposed to extend terminal sliding mode (TSM) control with an adaptive control loop based on the dynamic model of the manipulator. The adaptive loop estimates in real time the unknown, uncertain, or time-varying parameters of the model-based control loop. Real-time experiments, conducted on a 4-DOF PKM, show the effectiveness of the proposed adaptive TSM controller compared to the standard TSM control in terms of accuracy and robustness to parameter variations (such as the manipulated payload).

## 1.9 Overview on Pick-and-Throw motion generation

Throwing is known by its ability to increase the capability of a robotic manipulator as well as the range of its workspace by throwing objects outside its maximum kinematic range. Thanks to this potential, throwing robots have been used in several applications. For instance, in the military field, a throwing robot has been used to displace goods [Frank, 2010], [Frank, 2009]. Furthermore, a throwing robot can be useful to gather information on

a disaster site for search and rescue operations, such as in [Tsukagoshi et al., 2012] where a casting device is thrown by a robotic system in an unmanned environment. Fagiolini et al. [Fagiolini et al., 2011] dealt with casting manipulation, which consists in throwing the robot end-effector to catch objects located at a relatively large distance from the robot base. This manipulation is done using forces transmitted through a cable connected to the end-effector. Recently, the throwing technique has been applied in waste industry. A dual arm throwing robot has been developed as a collaborative robot (CoBot) working alongside people to sort waste [BHS, 2019]. In [Raptopoulos et al., 2020], a P&T approach with a Delta robot is applied for fast waste sorting. Real-time experimental results prove the improved performance of the throwing procedure, compared to conventional P&P.

Several other research works dealt with throwing. A one-degree-of-freedom direct drive arm has been developed in [Lynch and Mason, 1996], [Lynch and Mason, 1997] to perform a variety of dynamic manipulation tasks including throwing. By getting use of the centrifugal and Coriolis forces, the robot can throw the object to a desired target without grasping it. The authors in [Okada et al., 2015] have made a sensitivity analysis of the landing point with respect to model uncertainties such as joint friction and uncertainty on the initial position. They designed an optimal trajectory that minimizes this sensitivity. An offline motion planning of a throwing robotic arm under kinematic and dynamic constraints has been proposed in [Sintov and Shapiro, 2015]. In [Hu et al., 2018], the authors aimed to throw shuttlecocks through rings placed at some height. They designed the throwing trajectory and studied the influence of the initial throwing speed, the throwing angle as well as the influence of air resistance on the throwing distance. Zeng et al. [Zeng et al., 2020] investigated the challenge of accurately throwing arbitrary objects. They proposed a framework for jointly learning grasping and throwing policies from visual observations that enable TossingBot, a picking robot using a UR5 arm, to pick and throw arbitrary objects outside of its maximum range.

## 1.10 Objectives of the thesis

The present PhD thesis is a joint PhD between the Lebanese University and the University of Montpellier. It falls within the framework of a collaborative research project between the Laboratory of Informatics, Robotics and Microelectronics of Montpellier (LIRMM) and TECNALIA, a research and innovation organization located in Spain.

As mentioned above, recyclable sorting is a task that has to be done as fast and accurately as possible. Incorporating robots in waste recycling is the only way to effectively manage the ever-increasing volume of waste created by people following the modern lifestyle. The present research aims to speed up the recycling process and make it as efficient as possible. To this end, the objective within this thesis is to perform P&T tasks in a robust and fast manner using parallel manipulators (rigid-link and cable-driven parallel

robots), demonstrating the interest and relevance of a P&T technique compared to a usual Pick-and-Place approach in the context of a waste sorting application. The scientific issues addressed will cover the following two aspects:

1. Generation of P&T trajectories for the robot with the ballistic trajectories of objects of different types (i.e. dynamic parameters of the thrown objects may be known, known with uncertainties, or possibly unknown).
2. Synthesis of advanced control schemes to improve the tracking performance of parallel robots in terms of motion speed, accuracy and robustness. The proposed control solutions have to take into account system nonlinearities, uncertainties and parameter variations in order to accurately perform different types of movements (e.g. low and high acceleration).

The proposed P&T trajectory as well as the proposed control solutions will be validated, through numerical simulations and real-time experiments, on different available PKM prototypes. The conducted scenarios will be performed in different operating conditions on a P&T task, to show the effectiveness and robustness of the proposed control solutions in terms of global performances of the parallel robots.

## 1.11 Main contributions of the thesis

To fulfill the aforementioned objectives, the main contributions of this thesis can be listed as follows:

- **Contribution 1: A time-optimal Pick-and-Throw S-curve trajectory generation**
  In suitable robotic applications, throwing an object instead of placing it has the potential of improving the cycle time. In this context, a challenge is to generate time-optimal pick-and-throw trajectories in order to further increase the robot productivity. To this end, a methodology to determine a minimum-time throwing motion is proposed. This methodology consists essentially in determining an optimal release configuration (i.e. position and velocity) allowing an object to be thrown towards a desired target while minimizing the travel time of the throwing motion of the robot. To validate the potential of the proposed P&T approach, a comparison with the standard pick-and-place process and an existing P&T method is made using a Delta-like parallel robot under different operating conditions.

- **Contribution 2: A new augmented RISE feedback controller**
  A new control law based on the Robust Integral of the Sign of the Error (RISE) feedback law is proposed. It is a revision of the standard RISE control law performed by augmenting its control loop with a nonlinear dynamic compensation term. This term is calculated based on a combination of the system dynamic parameters, the tracking errors, and the measured and desired trajectories. In addition, the resulting

controller is extended with a nonlinear feedback function to compensate for the errors resulting from using the desired trajectories instead of the measured ones in the dynamic compensation term. The proposed control contribution can compensate for the high nonlinearities abundant in PKMs, as well as improve the robustness of the standard RISE controller. It is relevant when PKMs handle known objects in the P&T task. It is studied in the Lyapunov stability sense showing that the tracking error converges asymptotically to zero with time.

- **Contribution 3: A novel robust DCAL with adaptive feedback gains**
  The intended application is the use of PKM for P&T task in selective waste sorting. In addition to the highly nonlinear dynamic nature of PKMs as well as the abundant uncertainties, in such an application, the manipulator has to handle different types of objects that may often be unknown or uncertain. Therefore, the need for model-based adaptive schemes, which can online adjust the dynamic parameters, arises. To fulfill this object a new revised Desired Compensation Adaptive Law (DCAL) is proposed. DCAL is a model-based adaptive control strategy with a linear PD feedback term. The proposed contribution consists in amending the original DCAL with adaptive gain functions of the system errors to counteract perturbations and uncertainties. In addition, the controller has been extended by a nonlinear sliding-based term to further improve its robustness against external disturbances. The stability analysis of the proposed control has been investigated in the Lyapunov sense showing a global asymptotic convergence.

- **Contribution 4: A new intelligent robust control law**
  In the context of compensating for unmodeled phenomena and external disturbances that are not considered in model-based adaptive schemes, a novel intelligent robust control (IRC) has been developed. The proposed IRC scheme takes advantage of the RISE control law and model-free control (MFC). The MFC technique is based on an ultra-local model updated continuously in real time based on the input-output behavior of the system. This control approach is characterized by its simple concept and its ability to compensate for the modeled and unmodeled system dynamics without incorporating any apriori knowledge about the physical system. The proposed IRC approach consists in redesigning the original RISE scheme by exploiting an ultra-local model in its control loop. This revision can further improve the overall tracking performance of RISE control law and reinforce its robustness and disturbance rejection. A stability analysis of the proposed controller was included to ensure the asymptotic convergence of the tracking error.

## 1.12 Conclusion

The chapter introduce fundamental definitions and a classification of different kinematic structures of robotic manipulators, making a comparison of the advantages and drawbacks of serial and parallel manipulators. The history of PKMs and their wide range of applications with the recent incorporation of PKMs in the waste sorting industry were also discussed. Due to the growing interest in PKMs in both industry and academia, several PKMs prototypes have been proposed in the literature. Most of these mechanisms are based on the Gough platform and later on the Delta robot. This chapter presented the timeline of the different variants of the Delta PKM, which were developed to improve the overall performance of the Delta robot.

In the literature, there are multiple aspects from which parallel manipulators can be analyzed, synthesized and improved. The trajectory planning and control design aspects have been addressed in this chapter with the goal of improving the dynamic performance of PKMs. For the trajectory planning aspect, the challenge is to develop a continuous and smooth trajectory in the robot workspace and to meet the dynamic constraints of the robot. In the context of control design, control of PKMs is considered challenging due to their highly nonlinear dynamics that increase dramatically at high-speed motions, abundant uncertainties, time-varying parameters, external disturbances, and actuation redundancy in some cases.

This chapter recalled the equations for dynamic modeling of PKMs and how they can be obtained. Then, it presented the state of the art of existing control strategies for PKMs by classifying them into two categories: kinematic control and dynamic control. Kinematic control treats each actuator of the parallel manipulator independently without considering the dynamics in the control loop, while dynamic control relies mainly on a part of the robot dynamic model or on the full dynamic model. Unlike kinematic control, dynamic control compensates for nonlinearities, which improves the overall dynamic performance of the parallel manipulator, especially under highly dynamic operating conditions. One family of dynamic control approaches, dynamic adaptive controllers, provides an online estimate of system parameters and feeds these parameters back into the controller. This survey of control strategies proposed in the literature allowed us to compare the proposed solutions with the existing control strategies and built a solid base from where we start the development of our proposed control schemes. In addition, the state of the art on existing throwing trajectories and the different methods developed to generate a P&T trajectory were provided in this chapter.

The main objectives of this thesis have been presented as the generation of a time-optimal P&T trajectory accomplished by PKMs and the improvement of the dynamic performance of PKMs from a control point of view in terms of motion speed, robustness and

accuracy. The accomplished contributions of this thesis were listed at the end of this chapter.

The following chapter of the thesis presents the proposed time-optimal P&T trajectory. Then, the proposed control solutions are explained in Chapter 3. Finally, the validation of the proposed control solutions through numerical simulations will be demonstrated and discussed in Chapter 4.

CHAPTER

# 2 Proposed time-optimal Pick-and-Throw trajectory

## Contents


2.1 Introduction . . . . . 67
2.2 Minimum-time S-curve trajectory . . . . . 68
2.3 Minimum-time throw motion . . . . . 74
2.4 T3KR robot: A 5-DOFs PKM . . . . . 83
2.5 Real-time experimental results . . . . . 94
2.6 Conclusion . . . . . 102


## 2.1 Introduction

Throwing has the potential to speed up the displacement of objects and maximize productivity of a robotic manipulator. In order to take full advantage of this benefit, a time-optimal throwing motion should be generated. This can be accomplished by first determining the appropriate and feasible geometric path and then optimizing the motion time along this path. This is of great importance for waste sorting since the robot can perform more picks per minute and thereby a large amount of waste can be processed. To the best of our knowledge, none of the existing research works address the time-minimization of a P&T trajectory through the optimization of the throwing parameters. In most of the existing works, one or two of the initial throwing parameters are fixed, while the others are determined according to the target position without any optimization. For instance, in [Raptopoulos et al., 2020], the trajectory is designed as a usual P&P trajectory with the difference that the object is thrown at a given position on the horizontal path between the pick and target positions, resulting in a null initial release angle.

Therefore, the main objective is to determine an appropriate geometric path for a P&T motion and to find along this path the throwing configuration yielding a minimum-time motion. The P&T motion consists mainly of an acceleration and a deceleration phase. For each phase, a third-order polynomial S-curve is adopted as a motion profile to obtain smooth, continuous and fast trajectories. This chapter provides a detailed explanation of the proposed time-optimal P&T method. It mainly consists in determining an optimal release configuration (position and velocity) allowing an object to be thrown at a desired target while minimizing robot motion time. The constraints on the robot workspace, maximum speed, acceleration and jerk are taken into account. The corresponding optimization problem is formulated in two different ways. In the first one, the time between the pick position and the release position is considered as the objective function. The optimization variables are the release position, velocity and acceleration while constraints induced by the minimum-time S-curve and by ballistic motion target are taken into account. In the second method, it is shown that this optimization problem can be simplified as it boils down to minimizing the distance between the pick position and the release position with this distance and the release angle as the only variables. The second method requires less computational time, which is useful for real-time experiments where the optimal release configuration must be calculated online based on the actual pick and target positions. A comparison of the proposed P&T approach with standard P&P and with the P&T method proposed in [Raptopoulos et al., 2020] is conducted through real-time experimental scenarios with the parallel robot T3KR, under different operating conditions, to validate its effectiveness.

## 2.2 Minimum-time S-curve trajectory

A smooth enough trajectory with a limited jerk is necessary to avoid residual vibrations of a robot end-effector and thereby improve trajectory tracking accuracy. S-curve trajectories, developed for the first time by Castain et al. [Castain and Paul, 1984], can meet these requirements by providing high-speed motions with minimum positioning time and minimum residual vibrations. For instance, S-curve profiles with different jerk values have been generated and implemented in real-time experiments on robot manipulators [Piazzi and Visioli, 2000; Macfarlane and Croft, 2003]. An asymmetric S-curve, where the deceleration phase is slower than the acceleration phase, has been proposed to reduce residual vibrations and positioning errors [Rew and Kim, 2009; Li, 2016].

The S-curve profiles proposed in the literature can be based on polynomial, trigonometric and sigmoid equations [Nguyen et al., 2008; Fang et al., 2019]. On the one hand, the trigonometric and sigmoid models are smoother than the polynomial model. However, there is a trade-off between smoothness and motion time that makes the polynomial method more suitable for time minimization. On the other hand, the third order polyno-

mial S-curve (also known as double S trajectories) has a moderate complexity. Accordingly, it is the method adopted in this thesis. Methods to generate time-optimal S-curve trajectories have been previously proposed. For instance, an algorithm to optimize an objective function composed of two terms, the execution time and the jerk, has been developed in [Gasparetto and Zanotto, 2008]. A free-form third order S-curve motion profile, where the jerk values can be different for all the seven segments of the profile, has been proposed in [Bai et al., 2018] and applied to a flexible motion system to minimize the total positioning time and reinforce its robustness. An optimization method for S-curve trajectories, based on a chain of filters, has been proposed in [Biagiotti and Melchiorri, 2020] to eliminate residual vibrations under kinematic constraints.

In this section, an original algorithm formulation to compute the maximum velocity and maximum acceleration corresponding to the minimum-time $3^{rd}$ order S-curve trajectory is introduced. This formulation is equivalent to the algorithm presented in [Biagiotti and Melchiorri, 2008]. While this algorithm is well-known, a mathematical proof of the fact that it yields the minimum-time $3^{rd}$ order polynomial S-curve trajectory can hardly be found in the literature. In Appendix A, the proof is made by analyzing in detail the Karush-Kuhn-Tucker (KKT) conditions of the S-curve time optimization problem. This mathematical analysis is relatively long but allows us to prove time optimality. This alternative formulation can be used to gain insight into the properties of the minimum-time $3^{rd}$ order S-curve trajectory. Specifically, it is proved in Section 2.2.4 that the total time of the minimum-time S-curve trajectory is an increasing continuous function of the displacement. Moreover, the maximum velocity can be proved to be a nondecreasing continuous function of the displacement.

### 2.2.1 Polynomial S-curve motion profile

The $3^{rd}$ order polynomial S-curve is considered in this thesis. As shown in Figure 2.1, the motion profile of the $3^{rd}$ order polynomial S-curve consists of seven segments, among which the first three and the last three constitute the acceleration and deceleration phases, respectively, and the fourth segment constitutes the constant velocity stage. The jerk along the S-curve trajectory is defined by the following function of time:

$$j(t) = \begin{cases} J, & t_0 \le t \le t_1, t_6 \le t \le t_7 \\ 0, & t_1 \le t \le t_2, t_3 \le t \le t_4, t_5 \le t \le t_6 \\ -J, & t_2 \le t \le t_3, t_4 \le t \le t_5 \end{cases} \tag{2.1}$$

where J is the jerk value and the time instants $t_i$ are shown in Figure 2.1. The time evolution of the acceleration, velocity and displacement can be deduced by integration of (2.1) with appropriate initial and final conditions

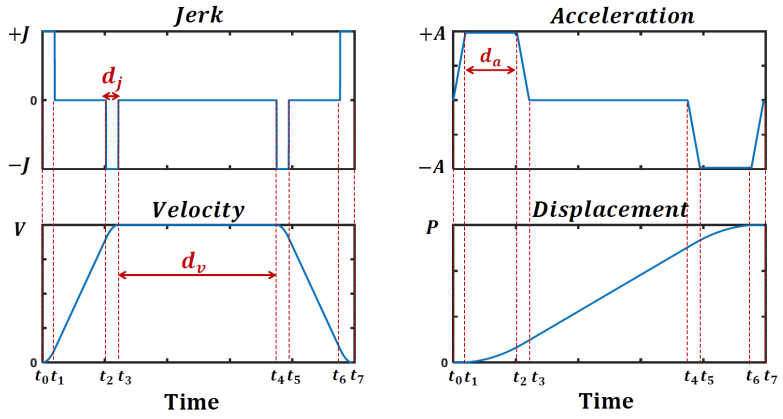



Figure 2.1 – Third order polynomial S-Curve model.

Referring to Figure 2.1, $d_j = t_1 - t_0$ is the time needed to increase the acceleration from zero to the maximum acceleration A or to decrease the acceleration from the maximum value A to zero, i.e. it is the time during which the jerk stays constant at the maximum jerk value J), $d_a = t_2 - t_1$ corresponds to the time during which the acceleration profile remains constant and equal to the maximum acceleration A, and $d_v = t_4 - t_3$ corresponds to the time during which the velocity profile remains constant and equal to V. Note that the symmetrical S-curve has been adopted in the present study (i.e. the acceleration and deceleration phases are symmetrical) and that $d_j$, $d_a$ and $d_v$ are all greater than or equal to zero. The three time intervals can be written as follows:

$$\begin{aligned} d_j &= \frac{A}{J} \\ d_a &= \frac{V}{A} - \frac{A}{J} \\ d_v &= \frac{P}{V} - \frac{V}{A} - \frac{A}{J} \end{aligned} \tag{2.2}$$

A and V are the maximum velocity and acceleration achieved for a given displacement P. Both A and V should be less than or equal to the corresponding maximum actuator capabilities. The total time to travel a distance P is then expressed as follows:

$$T = 4d_j + 2d_a + d_v \tag{2.3}$$

### 2.2.2 Minimization problem formulation

Taking into account constraints on the maximum acceleration and velocity ($A \leq A_{max}$ and $V \leq V_{max}$), the goal is to minimize the total time T needed to travel the distance P. This can be achieved by determining the optimal acceleration A and the optimal velocity

$V$ being given $J$, $P$, $A_{max}$ and $V_{max}$. By substituting the three time intervals (2.2) in the total displacement time (2.3), the function $T$ to be minimized can be expressed as:

$$T = \frac{P}{V} + \frac{V}{A} + \frac{A}{J} \tag{2.4}$$

Let us define $x = (x_1, x_2)$, where $x_1 = V$, $x_2 = A$, as well as $x_{1max} = V_{max}$ and $x_{2max} = A_{max}$. Hence, the objective function is:

$$f(x) = T = \frac{P}{x_1} + \frac{x_1}{x_2} + \frac{x_2}{J} \tag{2.5}$$

The minimization problem of finding the minimum-time S-curve trajectory is then formulated as follows:

$$\min_x f(x) \quad \text{subject to} \begin{cases} C_1(x) = d_a = \dfrac{x_1}{x_2} - \dfrac{x_2}{J} \geq 0 \\ C_2(x) = d_v = \dfrac{P}{x_1} - \dfrac{x_1}{x_2} - \dfrac{x_2}{J} \geq 0 \\ C_3(x) = x_{1max} - x_1 \geq 0 \\ C_4(x) = x_{2max} - x_2 \geq 0 \\ C_5(x) = x_1 > 0 \\ C_6(x) = x_2 > 0 \end{cases} \tag{2.6}$$

### 2.2.3 Algorithm to compute the minimum-time S-curve trajectory

The KKT first-order necessary optimality conditions of the optimization problem (2.6) yield 11 cases: One case of no active constraint, four cases of two active constraints ($C_1$, $C_2$, $C_3$ or $C_4$) and six cases of two active constraints ($C_1 = C_2 = 0$, $C_1 = C_3 = 0$, etc.), being given that the constraints $C_5$ and $C_6$ are always inactive since $V = x_1 > 0$ and $A = x_2 > 0$. A detailed analysis of these 11 cases (cf. Appendix A) shows that the following four cases are impossible at a (local) optimal solution:

- No active constraint
- One active constraint $C_1 = 0$
- One active constraint $C_4 = 0$
- Two active constraints $C_1 = C_4 = 0$

It also shows that, at a local optimal solution:

- One active constraint $C_2 = 0 \iff$ two active constraints $C_1 = C_2 = 0$
- One active constraint $C_3 = 0 \iff$ two active constraints $C_1 = C_3 = 0$
- The case of two active constraints $C_2 = C_3 = 0$ turns out to be a particular case of the case of one active constraint $C_3 = 0$

**Algorithm 1** Minimum-Time S-Curve Trajectory

**Input:** $P$, $J$, $V_{max}$, $A_{max}$
**Output:** $A$ and $V$ yielding the minimum time

**if** $(J^2P \le 2A_{max}^3)$ **then**
  **if** $(JP^2 \le 4V_{max}^3)$ **then**
    $V = \sqrt[3]{\frac{JP^2}{4}}$ and $A = \sqrt[3]{\frac{J^2P}{2}}$
  **else**
    $V = V_{max}$ and $A = \sqrt{JV_{max}}$
  **end if**
**else**
  **if** $(\sqrt{JV_{max}} \le A_{max})$ **then**
    $V = V_{max}$ and $A = \sqrt{JV_{max}}$
  **else**
    $A = A_{max}$
    $V = \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J}$
    **if** $(V > V_{max})$ **then**
      $V = V_{max}$
    **end if**
  **end if**
**end if**

Hence, out of the 11 cases, only four of them needs to be considered to determine the optimal solutions of the optimization problem (2.6):

- One active constraint $C_2 = 0$
- One active constraint $C_3 = 0$
- Two active constraints $C_2 = C_4 = 0$
- Two active constraints $C_3 = C_4 = 0$

As detailed in Appendix A, further analyzing these four cases lead to Algorithm 1 which allows to efficiently determine the values of V and A yielding the minimum-time polynomial S-curve trajectory.

### 2.2.4 Time $T$ and velocity $V$ as functions of displacement $P$

With acceleration $A$ and velocity $V$ computed by Algorithm 1, this section points out that the total displacement time $T$ given in (2.4) is a continuous increasing function of the displacement $P$ and also that the maximum velocity $V$ is a continuous nondecreasing function of $P$. These two properties of the minimum-time S-curve trajectory will be used in Section 2.3 to devise a method to generate minimum-time throw motions.

According to its expression in (2.4), $T$ depends on $P$, $V$ and $A$. Referring to Algorithm 1, either $V$ and $A$ are functions of $P$, e.g. at line 3, or else they are constant, i.e., independent of $P$ (e.g. at line 5). When $V$ and $A$ are independent of $P$, from (2.4), $T$ is directly seen to be an increasing function of $P$. On the contrary, when $V$ or $A$ is a function of $P$, it is not obvious from (2.4) that $T$ is an increasing function of $P$. Indeed, $V$ and $A$ appear both at the numerator and denominator of one of the terms of the sum on the right-hand side of (2.4).

In fact, when $P$ is sufficiently small, $V$ and $A$ are the functions of $P$ given at line 3 of Algorithm 1 and, when $P$ increases, the conditions on $P$ at lines 1, 2 and 13 will not be satisfied anymore for large values of $P$, so that $V$ and $A$ become constant. Let us look at the relationship between the conditions on $P$ at lines 1 and 2 of Algorithm 1, namely

$$P \leq \frac{2A_{max}^3}{J^2} \quad \text{and} \quad P \leq \sqrt{\frac{4V_{max}^3}{J}} \tag{2.7}$$

respectively. In particular, when $P$ increases from zero, line 5 will be executed if and only if

$$\sqrt{\frac{4V_{max}^3}{J}} < \frac{2A_{max}^3}{J^2}. \tag{2.8}$$

which, after some elementary calculations, can be shown to be equivalent to

$$\sqrt{JV_{max}} < A_{max}. \tag{2.9}$$

Hence, to analyze the dependency of $T$ and $V$ on $P$, two cases are distinguished: (2.9) satisfied and (2.9) not satisfied. In both cases, as detailed in Appendix B, the function $T(P)$, obtained with Algorithm 1, can be proved to be a continuous increasing function over $0 \leq P \leq +\infty$ as illustrated in Figure 2.2 for the case where (2.9) is satisfied.

Moreover, as also detailed in Appendix B, the function $V(P)$, obtained with Algorithm 1, is a continuous nondecreasing function of $P$. Since they will be used in Section 2.3.5, the different expressions of $V(P)$ are presented below.

*Case 1* – $\sqrt{JV_{max}} < A_{max}$ The continuous nondecreasing function $V(P)$ is composed of two segments: Namely, for $0 \leq P \leq \sqrt{(4V_{max}^3)/J}$, $V(P) = \sqrt[3]{JP^2/4}$, and for $\sqrt{(4V_{max}^3)/J} < P \leq +\infty$, $V(P) = V_{max}$ (see the left side of Figure 2.3).

*Case 2* – $\sqrt{JV_{max}} \geq A_{max}$ As illustrated on the right side of Figure 2.3, the continuous nondecreasing function $V(P)$ is composed of three segments:

- For $0 \leq P \leq 2A_{max}^3/J^2$, $V(P) = \sqrt[3]{JP^2/4}$.
- For $2A_{max}^3/J^2 \leq P \leq \left(JV_{max}^2 + V_{max}A_{max}^2\right)/(JA_{max})$, $V(P)$ is given by the expression at line 12 of Algorithm 1.
- For $\left(JV_{max}^2 + V_{max}A_{max}^2\right)/(JA_{max}) \leq P \leq +\infty$, $V(P) = V_{max}$.

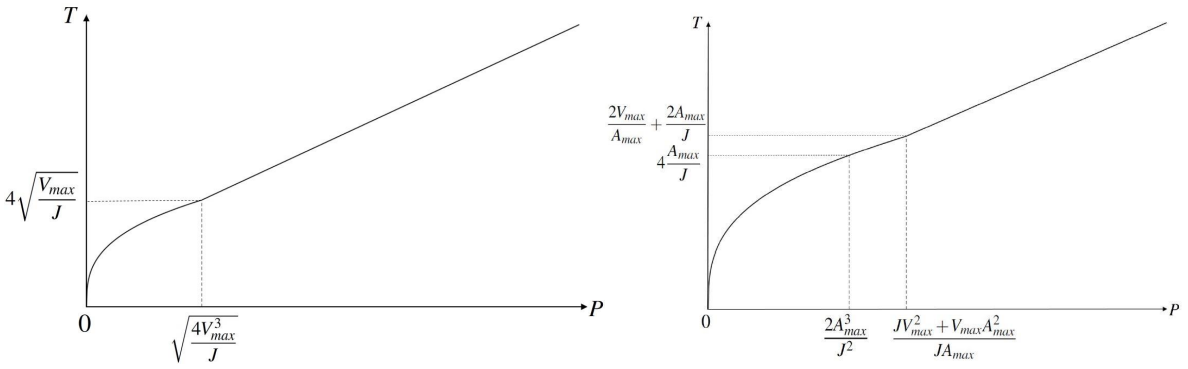



Figure 2.2 – A typical evolution of the increasing function $T(P)$ when $\sqrt{JV_{max}} < A_{max}$ (left) and when $\sqrt{JV_{max}} \geq A_{max}$ (right).

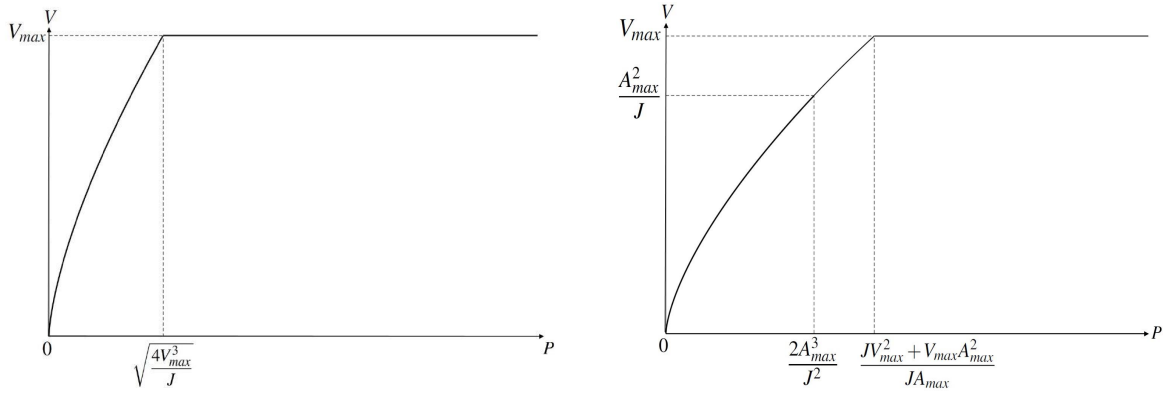



Figure 2.3 – A typical evolution of the nondecreasing function $V(P)$ when $\sqrt{JV_{max}} < A_{max}$ (left) and when $\sqrt{JV_{max}} > A_{max}$ (right).

## 2.3 Minimum-time throw motion

### 2.3.1 Problem formulation

A robot moving in 3D environment has to throw an object towards a desired target point $P_f$, located inside or outside of its workspace. The main objective is then to search for an optimal throwing configuration (i.e. position and velocity) in order to increase as much as possible the number of picks per minute. This configuration should thus allow to throw the object into the desired target while ensuring a minimum-time robot movement. Referring to Figure 2.4, the throw motion of the robot end-effector reference point $P(t)$ and the ballistic motion $P_B(t)$ of the object *B* are defined as follows.

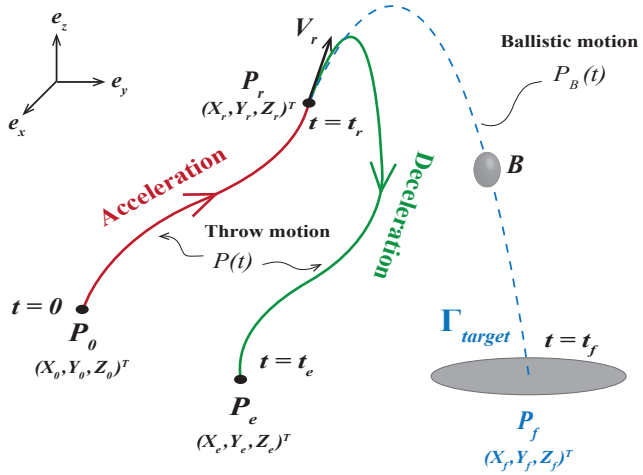



Figure 2.4 – Illustration of the throwing trajectory of the end-effector of the robot (Solid line) and the free-flight motion of the object (Dashed line).

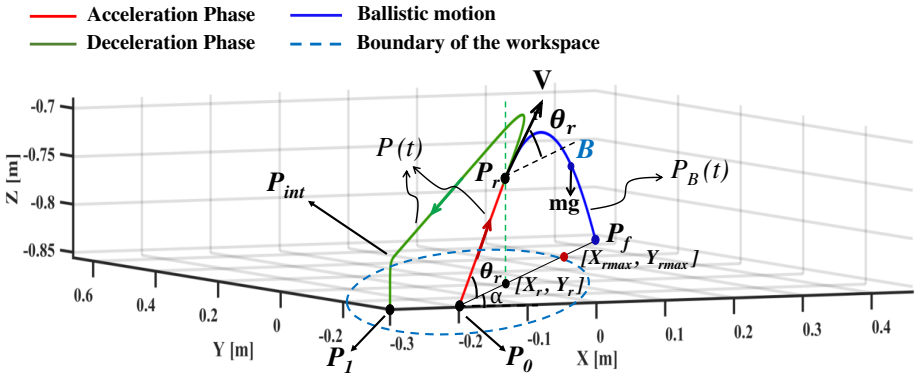



Figure 2.5 – Illustration of the pick and throw motion consisting of T3KR robot motion acceleration and deceleration phases together with the ballistic motion of the thrown object.

*Throw motion:* The robot trajectory consists of successive acceleration and deceleration phases. At $t = 0$, the manipulator is at the pick position with zero velocity, i.e. $P(0) = P_0$ with $\dot{P}(0) = V_0 = 0$. Once the robot picks the object *B*, it accelerates to the release position at time $t_r$. The release configuration is defined by the position $P_r$ and velocity $\dot{P}(t_r) = V_r \neq 0$. After tossing *B* instantaneously, the robot has to decelerate back to the next pick position at time $t_1$ to pick another object at point $P_1$ with velocity $\dot{P}(t_1) = V_1 = 0$.

*Ballistic motion:* Once released, the object *B* follows a free-flight motion from the release point $P_r$ with the velocity $V_r$ to the desired target $P_B(t_f) = P_f$ reached at time $t_f$. During the throw motion, the robot must satisfy two sets of constraints. The first set of

constraints $\Sigma$ includes the limits of the workspace as well as the maximum velocity, acceleration and jerk of the robot's end-effector. The second set of constraints $\Omega$ ensures that the ballistic trajectory interacts with the desired target position set $\Gamma_{target}$. This last one includes the target position point $P_f$ and possibly a tolerance for reaching the target. The target position set can have different forms, if it is a hyper-rectangle, the tolerance can be expressed as follows: $P_{target}^{min} < P_B(t_f) < P_{target}^{max}$.

For given pick positions $\mathbf{P}_0$ and $\mathbf{P}_1$, and target position $\mathbf{P}_f$, the problem addressed in this work is the determination of the release configuration corresponding to a minimum-time throw motion, this configuration being characterized by the release position $\mathbf{P}_r = [X_r, Y_r, Z_r]$ and the corresponding release velocity $\mathbf{V}_r = [V_{xr}, V_{yr}, V_{zr}]$. On the one hand, this configuration shall ensure that the thrown object will reach the desired target and, on the other hand, it should guarantee a minimum-time movement for the robot, while satisfying the sets of constraints $\Sigma$ and $\Omega$. The geometric path from $\mathbf{P}_0$ to $\mathbf{P}_r$ is a straight line lying in the vertical plane containing $\mathbf{P}_0$ and $\mathbf{P}_f$, $\theta_r$ denotes the angle between this straight line and the straight line $P_0 - P_f$. Hence, $\theta_r$ is an unknown to be determined along with the position of $\mathbf{P}_r$ along the straight line and the velocity $\mathbf{V}_r$. The description of the throw motion path will be detailed in Section 2.3.3.

### 2.3.2 Recall of mathematical modeling of ballistic motion

Considering the ballistic motion of an arbitrary object B in 3D space, we assume that there is no obstacles and that air resistance is negligible so that the ballistic trajectory is only affected by gravity. Moreover, the object is considered as a pointwise mass. The ballistic motion of the object B is illustrated in Figure 2.5. Applying Newton's second law to B gives simply $\mathbf{g} = \mathbf{a}$, where $\mathbf{g} \in \mathbb{R}^3$ is the gravity acceleration vector ($\mathbf{g} = [0, 0, -g]^T$, $g = 9.81\ m/s^2$, and $\mathbf{a} \in \mathbb{R}^3$ is the acceleration of B. By integration of the equation $\mathbf{g} = \mathbf{a}$, the trajectory of B along the $x$, $y$ and $z$ axes can be expressed as follows:

$$x_B(t) = V_{xr}t + X_r, \quad y_B(t) = V_{yr}t + Y_r \tag{2.10a}$$

$$z_B(t) = -\frac{1}{2}gt^2 + V_{zr}t + Z_r \tag{2.10b}$$

As shown in Figure 2.5, $\mathbf{P}_f = [X_f, Y_f, Z_f]^T$ is the target position of B, reached at the final time $t_f$ (i.e. $P_B(t_f) = P_f$). With $z_B(t_f) = Z_f$, $t_f$ can be calculated from (2.10b) as follows:

$$t_f = \frac{V_{zr}}{g} + \sqrt{\left(\frac{V_{zr}}{g}\right)^2 + 2\frac{Z_r - Z_f}{g}} \tag{2.11}$$

### 2.3.3 Throw motion geometric path and S-curve motion profile

The geometric path of the throw motion includes four points: The pick position of the object, $\mathbf{P}_0 = [X_0, Y_0, Z_0]$, the release point, $\mathbf{P}_r = [X_r, Y_r, Z_r]$, the pick position of the second object, $\mathbf{P}_1 = [X_1, Y_1, Z_1]$, and the intermediate point, $\mathbf{P}_{int} = [X_1, Y_1, Z_1 + Z_{off}]$. The overall throw motion can then be divided into three phases: (i) An acceleration phase along the straight line from $\mathbf{P}_0$ to $\mathbf{P}_r$, (ii) a deceleration phase from $\mathbf{P}_r$ to $\mathbf{P}_{int}$, and (iii) finally, a vertical movement from $\mathbf{P}_{int}$ to $\mathbf{P}_1$. Once the robot end-effector reaches the release position $\mathbf{P}_r$ with the release velocity $V_r$, it throws the object towards the target position and then moves forwards until the velocity becomes zero. After that, it starts decelerating towards $\mathbf{P}_{int}$ in a continuous motion. The parallel robot T3KR, used in the experiments (cf. Section 2.5), cannot decelerate directly towards $\mathbf{P}_1$ because the end-effector must move down vertically in order to successfully grasp the object. The $z$ coordinate of $\mathbf{P}_{int}$ is then chosen as $Z_1 + Z_{off}$, with $Z_{off} > 0$ being a small vertical offset.

The three phases of the robot throw motion are defined as point-to-point movements. Besides, the S-curve velocity profile described in Section 2.2 is adopted to generate each movement, where a minimum-time point-to-point movement is obtained by using Algorithm 1. As mentioned in Section 2.2.1, the motion profile of a S-curve-based point-to-point movement consists of seven segments (cf. Figure 2.1). For each segment of motion, the evolution of the variation of the motion profile with time, $S(t)$, is governed by the following equation:

$$S(t) = S_i + V_i(t - t_i) + \frac{1}{2}A_i(t - t_i)^2 + \frac{1}{6}J(t - t_i)^3 \tag{2.12}$$

where $J, A_i, V_i, S_i, t_i$ are the initial values of jerk, acceleration, velocity, position and time corresponding to each segment. The desired Cartesian position of the robot end-effector is then generated as follows:

$$X_d(t) = X_i + \frac{S(t)}{P}(X_f - X_i) \tag{2.13}$$

where $X_d(t) = [x_d(t), y_d(t), z_d(t)]$ is the desired Cartesian trajectory, $X_i = [x_i, y_i, z_i]$ and $X_f = [x_f, y_f, z_f]$ are the initial and final positions of a point-to-point movement, respectively. $P = \|X_f - X_i\|$ is the distance to be traveled. By means of equations (2.12) and (2.13) the synchronization between axes is guaranteed.

Regarding the connection of two consecutive phases (i.e. corner blending), we exploit an overlap strategy [Khalil and Dombre, 2004] to smooth the trajectory and eliminate discontinuities. In brief, each trajectory phase is designed as a point-to-point trajectory. The duration of the deceleration phase of the first trajectory is compared to the duration of the acceleration phase of the second trajectory, and a minimum overlap is determined. This ensures that the first trajectory will smoothly disappear during the second trajectory acceleration phase yielding a continuous switching between the two consecutive trajectories.

### 2.3.4 Determination of the optimal release configuration

The optimal release configuration is defined as the one leading to a minimum-time throw motion of the robot. This motion consists of the three phases described in Section 2.3.3. Since the third phase is a small vertical movement required to pick the next object, which is assumed to be located close to $P_0$, the optimal release configuration is the one that minimizes the time of the first two phases. By design of the deceleration phase, the smaller the duration of the acceleration phase, the smaller is the duration of the deceleration phase. Hence, the minimum-time throw motion is obtained by minimizing the time of the acceleration phase. The release configuration, including $\mathbf{P}_r$ and $\mathbf{V}_r$, allowing to minimize this time, while allowing the object to reach the target position $\mathbf{P}_f$, is the optimal one.

As mentioned above, the adopted motion profile along the straight line $P_0 - P_r$ is a $3^{rd}$ order polynomial S-curve. For the object to be thrown at a certain distance (to the target position $\mathbf{P}_f$), the release velocity $V_r = \|\mathbf{V}_r\|$ should be relatively high and, according to Section 2.2, the velocity along the S-curve is maximal (equal to $V$) at the half of the traveling distance, i.e. at $P/2$. Hence, the traveled distance $P$ of the S-curve is defined as being equal to twice $P_0P_r$, $P = 2\|\mathbf{P}_r - \mathbf{P}_0\|$. Accordingly, the time $t_r$ required to travel the distance from $\mathbf{P}_0$ to $\mathbf{P}_r$ is then equal to the half of the time calculated in (2.4). In this way, the release velocity is equal to the S-curve maximum velocity, i.e. $V_r = V$. From (2.4), the time $t_r$ can then be expressed as a function of $V_r$ and of the coordinates of $\mathbf{P}_r$ as follows:

$$t_r = \frac{1}{2}\left(\frac{2\|\mathbf{P}_r - \mathbf{P}_0\|}{V_r} + \frac{V_r}{A} + \frac{A}{J}\right) \tag{2.14}$$

The optimal release configuration $(\mathbf{P}_r, V_r)$ is then the one that minimizes the traveling time $t_r$ in (2.14) while satisfying the two sets of constraints $\Sigma$ and $\Omega$ defined in Section 2.2.

Let us now define $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]$, where $x_1 = V = V_r$, $x_2 = A$, $x_3 = X_r$, $x_4 = Y_r$ and $x_5 = Z_r$ as well as $x_{1max} = V_{max}$, $x_{2max} = A_{max}$, $x_{3max} = X_{r_{max}}$, $x_{4max} = Y_{r_{max}}$ and $x_{5max} = Z_{r_{max}}$. As illustrated in Figure 2.5, $X_{r_{max}}$, $Y_{r_{max}}$ and $Z_{r_{max}}$ are respectively the $x$, $y$ and $z$ coordinates of the intersection point between the straight line $P_0 - P_f$ and the boundary of the robot workspace. With these notations and being given that the goal is to minimize $t_r$ given in (2.14), the objective function $f(\mathbf{x})$ to be minimized over $\mathbf{x}$ can be defined as:

$$f(\mathbf{x}) = \frac{2\sqrt{(x_3 - X_0)^2 + (x_4 - Y_0)^2 + (x_5 - Z_0)^2}}{x_1} + \frac{x_1}{x_2} + \frac{x_2}{J} \tag{2.15}$$

The optimization problem includes also the following constraints.

#### 2.3.4.1 Bound constraints

$$0 < x_1 \le x_{1max} \tag{2.16a}$$

$$0 < x_2 \le x_{2max} \tag{2.16b}$$

$$X_0 \le x_3 \le x_{3max}, \quad Y_0 \le x_4 \le x_{4max} \tag{2.17a}$$

$$Z_0 \le x_5 \le x_{5max} \tag{2.17b}$$

#### 2.3.4.2 Nonlinear inequality constraints

$$d_a = \frac{x_1}{x_2} - \frac{x_2}{J} \ge 0 \tag{2.18}$$

$$d_v = \frac{2\sqrt{(x_3 - X_0)^2 + (x_4 - Y_0)^2 + (x_5 - Z_0)^2}}{x_1} - \frac{x_1}{x_2} - \frac{x_2}{J} \ge 0 \tag{2.19}$$

$$X_f - \delta \le x_1 \cos\theta_r \cos(\alpha_{xy}) t_f + x_3 \le X_f + \delta \tag{2.20}$$

$$Y_f - \delta \le x_1 \cos\theta_r \sin(\alpha_{xy}) t_f + x_4 \le Y_f + \delta \tag{2.21}$$

where $\mathbf{P}_f = [X_f, Y_f, Z_f]$, $\alpha_{xy} = \arctan(x_4 - Y_0, x_3 - X_0)$ is the angle between the x-axis and the vertical plane containing $P_0$ and $P_r$, and $\theta_r = \arctan\left(x_5 - Z_0, \sqrt{(x_3 - X_0)^2 + (x_4 - Y_0)^2}\right)$ is the angle between the horizontal plane and $P_0 - P_r$ as shown in Figure 2.5. The constraints (2.18) and (2.19) are related to the S-curve ensuring that $d_a$ and $d_v$ are nonnegative, while (2.20) and (2.21) ensure that the object B reaches the target horizontal position $[X_f, Y_f]$ with a tolerance of $\delta$ (chosen as $\delta = 0.005$ m in Section 2.5). Note that the final time $t_f$ in (2.11) is a function of $x_1$ and $x_5$ as follows:

$$t_f = \frac{x_1 \sin\theta_r}{g} + \sqrt{\left(\frac{x_1 \sin\theta_r}{g}\right)^2 + 2\frac{x_5 - Z_f}{g}} \tag{2.22}$$

since the release velocity vector is given by $\mathbf{V}_r = x_1[\cos\theta_r \cos\alpha_{xy}, \cos\theta_r \sin\alpha_{xy}, \sin\theta_r]$.

#### 2.3.4.3 Nonlinear equality constraint

$$\alpha_{xy} - \alpha = 0 \tag{2.23}$$

where $\alpha = \arctan(Y_f - Y_0, X_f - X_0)$. This constraint guarantees that the release point lies in the vertical plane containing $P_0$ and $P_f$.

To sum up, the problem of minimizing $f(\mathbf{x})$ over $\mathbf{x}$ subject to the constraints defined above is a nonlinear constrained optimization problem where the S-curve time minimization is coupled with the determination of the ballistic motion release configuration. This problem can be solved with standard nonlinear programming solvers.

### 2.3.5 Optimization problem simplification

The nonlinear constrained optimization problem introduced in Section 2.3.4 can be substantially simplified. First, as shown in Figure 2.5, since both the robot throw motion acceleration phase and the ballistic motion lie in the vertical plane containing $P_0$, $P_r$ and $P_f$, the three-dimensional problem can easily be converted into a planar one (setting the tolerance zone defined in (2.20) and (2.21) aside). Moreover, the pick position $P_0$ is known. Hence, without loss of generality, we can choose the reference frame origin as being $P_0$ and its $x$-axis oriented along the projection of the straight line $P_0 - P_r$ on the $(x, y)$ plane. Then, the vertical plane containing the acceleration phase and the ballistic motion is the $(x, z)$ plane. Accordingly, referring to (2.10a) and 2.10b, $y_B(t) = 0, \forall t$ and after substituting the time $t$ calculated from the equation of $x_B(t)$ into $z_B(t)$, and rearranging the terms, the following expression is obtained:

$$z_B = -\frac{g}{2}\frac{(x_B - P_r\cos\theta_r)^2}{(V_r\cos\theta_r)^2} + x_B\tan\theta_r \tag{2.24}$$

where, with the chosen reference frame, $\alpha_{xy} = 0$ and $\mathbf{P}_0 = \mathbf{0}$ so that $\mathbf{P}_r = [X_r, Y_r, Z_r] = P_r[\cos\theta_r, 0, \sin\theta_r]$ and $\mathbf{V}_r = V_r[\cos\theta_r, 0, \sin\theta_r]$. By substituting the coordinates of the target position $P_f$ into (2.24) ($x_B = X_f$ and $z_B = Z_f$), the release velocity $V_r$ can be expressed as a function of the distance $P_r$ and the release angle $\theta_r$ as follows:

$$V_r = \frac{-P_r + (X_f/\cos\theta_r)}{\sqrt{\frac{2}{g}(X_f\tan\theta_r - Z_f)}} \tag{2.25}$$

with $\arctan(Z_f/X_f) \le \theta_r \le \pi/2$. Equation (2.25) is the ballistic motion target constraint that shall be satisfied for the object to reach the target $P_f$.

Besides, as explained in Section 2.3.4, we have $P = 2P_r$ and $V = V_r$ where $P$ and $V$ are the displacement and maximum velocity of the S-curve motion profile used to generate the

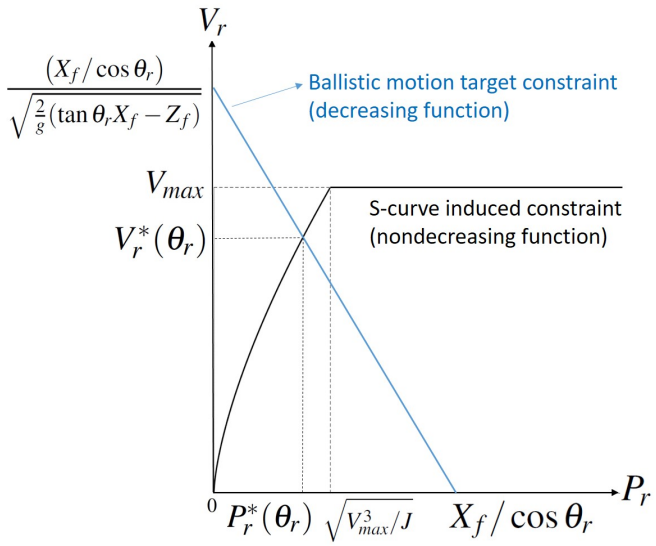



Figure 2.6 – The ballistic motion constraint and the minimum-time S-curve induced constraint as functions $V_r(P_r)$. For a given $\theta_r$, there exists a unique couple $(P_r^*(\theta_r), V_r^*(\theta_r))$ satisfying both constraints.

throw motion acceleration phase. Referring to Section 2.2.4, the S-curve total time $T = 2t_r$ is an increasing function of the distance $P$ and thus of $P_r = P/2$. Consequently, minimizing $t_r$, which is $f(\mathbf{x})/2$ in Section 2.3.4, is equivalent to minimizing $P_r$, i.e., the minimum-time throw motion is obtained by minimizing $P_r$.

The minimization of $P_r$ is subjected to the ballistic motion target constraint in (2.25) and to a constraint due to the minimum-time S-curve motion profile. Indeed, the latter imposes a relationship between $P$ and $V$ and thus also between $P_r$ and $V_r$ since $P = 2P_r$ and $V = V_r$. As pointed out in Section 2.2.4, this relationship takes the form of $V_r$ being a continuous nondecreasing function of $P_r$ composed of either two or three segments. For each of these segments, the expression of $V_r(P_r)$ is known as detailed in Section 2.2.4 for $V(P)$. It is worth noting that this function $V_r(P_r)$ is nondecreasing while, in (2.25), $V_r$ is a decreasing linear function of $P_r$ for a given $\theta_r$. Hence, as illustrated in Figure 2.6, for any value of $\theta_r$, $\arctan(Z_f/X_f) \leq \theta_r \leq \pi/2$, there exists a unique couple $(P_r^*(\theta_r), V_r^*(\theta_r))$ satisfying both the function $V_r(P_r)$ from the constraint induced by the minimum time S-curve and the function $V_r(P_r)$ from the ballistic motion target constraint in (2.25). These values of $P_r$ and $V_r$ depend on $\theta_r$, and this is the rationale of the notations $P_r^*(\theta_r)$ and $V_r^*(\theta_r)$. Hence, minimizing $P_r$ boils down to finding the value of $\theta_r$ that yields the smallest $P_r^*(\theta_r)$.

Based on the above analysis, an efficient method to determine $P_r^*(\theta_r)$ can be devised as follows. First, note that obtaining a closed-form expression of $P_r$ as a function of $\theta_r$, i.e. eliminating $V_r$ in (2.25) by means of the expression $V_r(P_r)$ obtained from the S-curve induced constraint (Section 2.2.4), is difficult except in the case where $V_r(P_r) = V_{max}$. In-

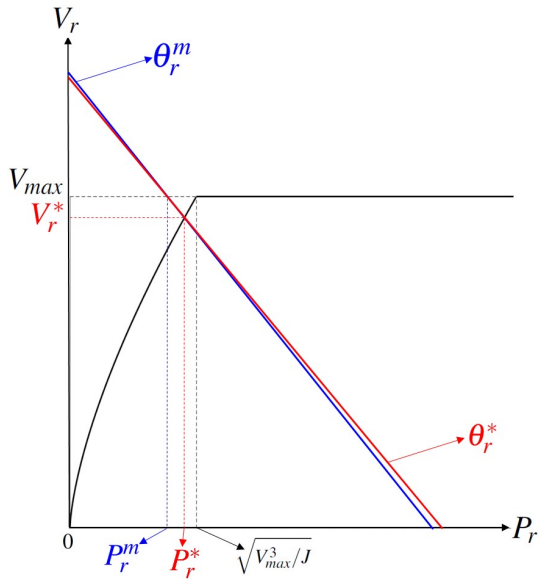



Figure 2.7 – In the case $\sqrt{JV_{max}} < A_{max}$, this figure shows the minimum $P_r^m = P_r(\theta_r^m)$ of the function in (2.26) (case where $P_r^m < \sqrt{V_{max}^3/J}$) and the optimal release configuration $P_r^*$, $V_r^*$ and $\theta_r^*$.

deed, in this case, (2.25) implies

$$P_r = (X_f/\cos\theta_r) - V_{max}\sqrt{2(X_f\tan\theta_r - Z_f)/g}. \tag{2.26}$$

An analysis of this function shows that it possesses a unique minimum $P_r^m = P_r(\theta_r^m)$ for $\arctan(Z_f/X_f) \le \theta_r \le \pi/2$. Moreover, this minimum can be straightforwardly calculated by solving a univariate nonlinear equation in $\theta_r$ obtained from $dP_r/d\theta_r = 0$. Then, the two cases defined above in Section 2.2.4 have to be distinguished.

*Case 1* – $\sqrt{JV_{max}} < A_{max}$. Referring to Figure 2.6, if $P_r^m \ge \sqrt{V_{max}^3/J}$, the problem is solved since the optimal release configuration is $P_r^* = P_r^m$ and $V_r^* = V_{max}$. Otherwise, as illustrated in Figure 2.7, the S-curve induced constraint is not satisfied at $P_r^m$ and the optimal release configuration $(P_r^*, V_r^*)$ lies on the first segment $V_r = \sqrt[3]{JP_r^2}$ of this constraint. The optimal release configuration is obtained by solving the following nonlinear optimization problem where the nonlinear equation $g(P_r, \theta_r) = 0$ is obtained by replacing $V_r$ in (2.25) by $\sqrt[3]{JP_r^2}$.

$$\begin{aligned} \min_{P_r,\theta_r} P_r \quad \text{s.t.} \quad & g(P_r,\theta_r) = 0 \\ & 0 \le P_r \le \sqrt{V_{max}^3/J} \\ & \arctan(Z_f/X_f) \le \theta_r \le \pi/2 \end{aligned} \tag{2.27}$$

The solution of this optimization problem is $P_r^*$ and $\theta_r^*$ where $P_r^*$ is the smallest value of $P_r$ allowing to reach the ballistic motion target with the minimum-time S-curve motion profile for the throw motion acceleration phase. In other words, $P_r^*$, $\theta_r^*$ and $V_r^* = \sqrt[3]{JP_r^{*2}}$ constitute the optimal release configuration. As shown in Figure 2.7, it is worth noting that the ballistic motion target constraints for $\theta_r^m$ and $\theta_r^*$ are close to each other (this comes from the fact that $X_f/\cos\theta_r$ is an increasing function of $\theta_r$). Hence, using $P_r^m$ and $\theta_r^m$ as an initial estimate of the solution of (2.27) leads to a fast solving of this optimization problem.

*Case 2* – $\sqrt{JV_{max}} \geq A_{max}$. Referring to Figure 2.3 and Section 2.2.4, if the point $(P_r^m, V_{max})$ lies on the third segment of the S-curve induced constraint, i.e.

$$P_r \geq \left(JV_{max}^2 + V_{max}A_{max}^2\right) / \left(2JA_{max}\right) \tag{2.28}$$

then the optimal release configuration is $P_r^* = P_r^m$ and $V_r^* = V_{max}$. Otherwise, a closed-form expression of $P_r$ as a function of $\theta_r$ is obtained by replacing $V_r$ in (2.25) by $A_{max}^2/J$. Similarly to the case of (2.26) discussed above (with $A_{max}^2/J$ in place of $V_{max}$), this function possesses a unique minimum $P_r^M = P_r(\theta_r^M)$ which can be easily found by solving a univariate nonlinear equation. If $P_r^M \leq A_{max}^3/J^2$, then the optimal release configuration $(P_r^*, V_r^*)$ lies on the first segment of the S-curve induced constraint and can be efficiently calculated by solving the optimization problem (2.27) with the initial estimate $(P_r^M, \theta_r^M)$. Otherwise, the optimal release configuration $(P_r^*, V_r^*)$ lies on the second segment of the S-curve induced constraint. It can then be obtained by solving an optimization problem similar to (2.27) but with $g(P_r, \theta_r) = 0$ obtained from the expression of $V(P)$ at line 12 of Algorithm 1, and with the initial estimate $(P_r^M, \theta_r^M)$.

In summary, the optimal release configuration $(P_r^*, V_r^*)$ yielding a minimum-time throw motion can be efficiently obtained by solving one or two univariate nonlinear equations and one optimization problem (2.27) having two variables with bound constraints and one equality constraint. Solving this rather simple optimization problem takes a short time especially since an initial estimate close to the optimal solution is known. Hence, compared to the nonlinear constrained optimization problem introduced in Section 2.3.4, the solving method presented in this section is more efficient. Furthermore, it provides insight into the nature of the problem of determining the optimal release configuration. It has notably been pointed out that the corresponding optimization problem possesses a unique minimum, i.e., as defined in this work, the optimal release configuration is unique.

## 2.4 T3KR robot: A 5-DOFs PKM

Modeling is a crucial and challenging topic in robotics research. It plays an important role in the simulation of motion, the analysis of manipulator structures, as well the design of control algorithms. Kinematic analysis refers to the study or of the geometry of the robot itself, without considering the forces or torques that generate the motion [Taghirad,

2013]. Most kinematic models of parallel robots have developed from a geometric analysis of kinematic chains resulting in equations describing the closed-loop kinematic chains and implying joint and Cartesian variables [Merlet, 2005]. This section provides description of the experimental platform, T3KR PKM, that has been used for the implementation of the pick-and-throw task. A general description of the mechanical structure is presented, the inverse and direct kinematic analyses are worked out in detail, and the dynamic model is mathematically derived and explained. The established dynamic model of T3KR PKM will be used in the design of some control schemes in Chapter 3.

### 2.4.1 Description of T3KR PKM

The T3KR robot, is a rigid-link Delta-like parallel robot, developed within the framework of a cooperation between SATT AxLR, TECNALIA and LIRMM in 2015. It was proposed as a new industrial P&P machine with an optimized footprint (i.e. it has an optimized mechanical structure) and designed for simple position control. It consists of a fixed-base support holding four actuators controlling four identical kinematic chains. Each kinematic chain consists of a revolute actuator, a movable rear-arm and a parallelogram composed of two parallel rods, named forearm. The forearm is linked at one extremity to the rear-arm and at its second extremity to the mobile platform by means of passive spherical joints. The mobile platform holds a small end-effector that picks and places objects by means of a vacuum suction cups. Figure 2.8 provides the main components of T3KR PKM.

The robot has five DOFs: The parallel structure with the four motors allows the translation of the mobile platform along $x, y$ and $z$ axes as well as its rotation $\psi$ around the $z$-axis; The rotation of the platform is a parallelogram mechanism movement, where the tool control point (TCP) is on the neutral point of this mechanism. Accordingly, a rotation $\psi$ of the moving platform does not induce a motion of the TCP, this is why $\psi$ is kept equal to zero. A further actuator, integrated on the mobile platform, allows a rotation $\phi$ of the end-effector around the vertical $z$-axis. Due to its asymmetrical mechanical structure, T3KR has an elliptical workspace in top view, as illustrated in Figure 2.9.

In this thesis we are only interested in the control of the four main actuators of the parallel structure of T3KR robot. Advanced control solutions were proposed for the trajectory tracking problem of the mobile platform in the workspace. All the modeling in the sequel is based only on the parallel structure of the T3KR robot.

### 2.4.2 Kinematics of T3KR PKM

Consider the vector $E = [x, y, z, \psi, \phi]^T \in \mathbb{R}^5$ as a representation of the end-effector configuration in the fixed reference frame. The formed angles by the actuated joints located on the parallel structure and the traveling-plate are represented by the vector

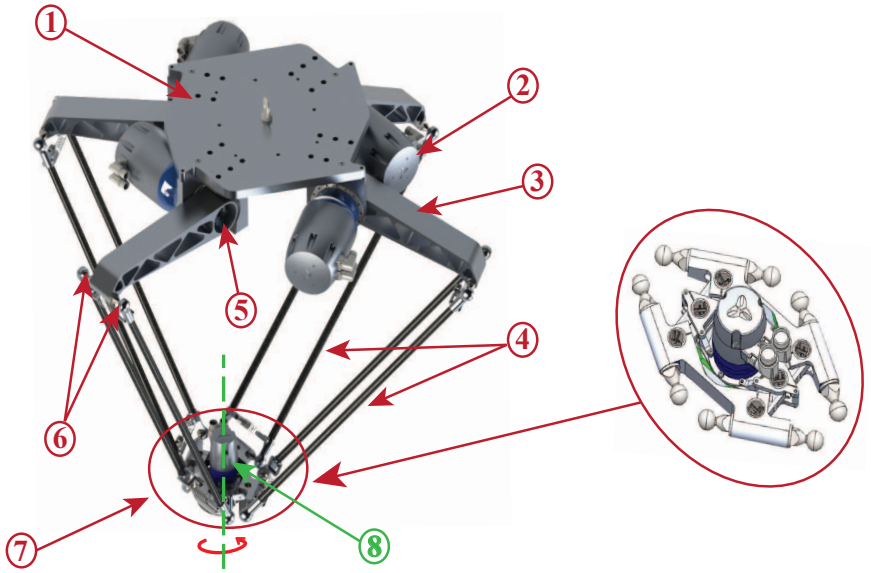



Figure 2.8 – A schematic view of T3KR parallel robot including ① : Fixed base, ②: Main actuator, ③ : Rear-arm, ④: Forearms, ⑤ : Revolute active joint, ⑥ : Passive spherical joints, ⑦ : Mobile platform, ⑧ : Integrated actuator on the platform

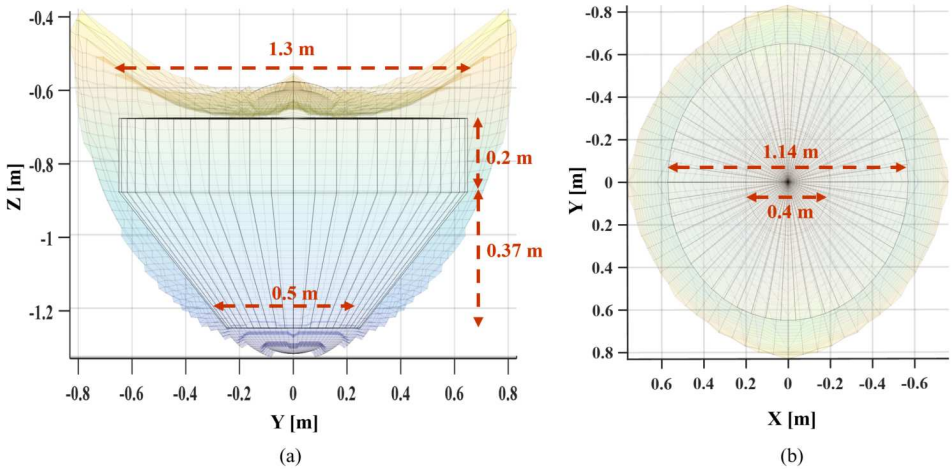



Figure 2.9 – T3KR workspace limits in the planes (Y,Z) and (X,Y).

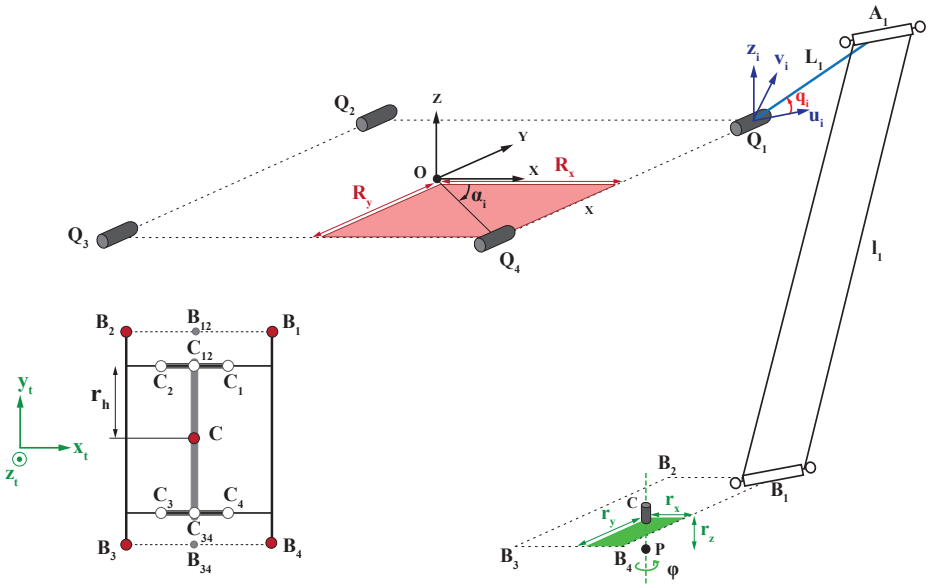



Figure 2.10 – Illustration of the kinematics of the T3KR PKM.

$Q = [q_1, q_2, q_3, q_4, \phi]^T \in \mathbb{R}^5$. It is worth to mention that the variable $\phi$ is the same for operational and joint space. Since the actuator on the mobile platform is decoupled from the others actuators on the fixed base, the kinematics of the parallel structure can be decoupled from the rotational movement provided by the actuator integrated on the traveling-plate. Let us define a vector $q = [q_1, q_2, q_3, q_4]^T \in \mathbb{R}^4$ which involves only the parallel structure joint variables and $X = [x, y, z, \psi]^T \in \mathbb{R}^4$ as the corresponding Cartesian vector pose.

The kinematics of T3KR is shown in Figure 2.10. Let O be the center of a parallelogram whose vertices are the actuated joints represented by $Q_i$ for $i = 1,2,3,4$. The fixed reference frame attached to O is $R = \{O, e_x, e_y, e_z\}$, where $e_x, e_y, e_z$ are the corresponding unit vectors.

Let $A_i$ and $B_i$ be two virtual points located at the midpoints of each two ball-and-socket joints connecting one rear-arm to one forearm and one forearm to the traveling-plate, respectively. An auxiliary frame $R_i = \{Q_i, u_i, v_i, z_i\}$ is attached to each actuated joint. The following transformation matrix T is applied to transform a vector from the fixed frame R to the auxiliary frame $R_i$.

$$T = \begin{pmatrix} \cos(\alpha_i) & -\sin(\alpha_i) & 0 & \text{sign}(\cos(\alpha_i))R_x \\ \sin(\alpha_i) & \cos(\alpha_i) & 0 & \text{sign}(\sin(\alpha_i))R_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.29}$$

where $\alpha_1 = \frac{\pi}{6}$, $\alpha_2 = pi - \frac{\pi}{6}$, $\alpha_3 = pi + \frac{\pi}{6}$ and $\alpha_4 = -\frac{\pi}{6}$. $R_x$ and $R_y$ are half the lengths of the edges of the fixed base parallelogram whose vertices are the actuated joints (see Figure 2.10). $sign(.)$ is the nonlinear signum function. It is clear that the rotation in the transformation matrix T is performed around $z$-axis by angle $\alpha_i$. Therefore $\overrightarrow{z_i} = \overrightarrow{e_z}$. $q_i$ is defined as an angle between $\overrightarrow{u_i}$ and $\overrightarrow{Q_iA_i}$ starting from $\overrightarrow{u_i}$ in the vertical plane $(u_i, z_i)$ as shown in Figure 2.10.

On the left side of Figure 2.10 showing the kinematics of the traveling-plate with points $C_i$ and $C_{ij}$ are passive revolute joints. Given the kinematics constraints set on points $B_i$ by the parallelogram structure of the arms, any rotational movement of the points $B_i$ together with points $C_i$ generates a motion that does not impact the position of point C and the orientation of bar $C_{12}C_{34}$. C is the control point of the traveling-plate. Let us consider the traveling-plate parallelogram of center C and whose vertices are the virtual points $B_i$. $r_x$ and $r_y$ are half the lengths of the edges of this parallelogram. A local frame $R_t = \{C, x_t, y_t, z_t\}$ is attached to C.

Let $P = [x, y, z]^T$ be the Cartesian position of the end-effector with respect to the reference frame R. Then, a translation $r_z$ along $z$-axis, where $r_z$ is the length of the end-effector, is performed to obtain the position vector of $C = [x, y, z + r_z]^T$ (see Figure 2.10). Point $B_i$ is always lying in the horizontal plane $(x_t, y_t)$. Before proceeding to the development of the coordinates of points $B_i$ in reference frame R, let us define the following vectors:

$$CC_{12} = [-r_h sin(\psi), r_h cos(\psi), 0]^T, \quad CC_{34} = [r_h sin(\psi), -r_h cos(\psi), 0]^T \tag{2.30}$$

$$C_{12}B_{12} = [0, (r_y - r_h), 0]^T, \quad C_{34}B_{34} = [0, -(r_y - r_h), 0]^T \tag{2.31}$$

where $\psi$ is the rotational angle of the platform which is kept at zero all times as explained above and $r_h$ is the distance from C to $C_{12}$ or from C to $C_{34}$. While $B_{12}$ and $B_{34}$ are two virtual points located at the midpoints of the distances from $B_1$ to $B_2$ and from $B_3$ to $B_4$, respectively, as shown on the left side of Figure 2.10.

$$B_{12}B_1 = [r_x, 0, 0]^T, \quad B_{12}B_2 = [-r_x, 0, 0]^T \tag{2.32a}$$

$$B_{34}B_3 = [-r_x, 0, 0]^T, \quad B_{34}B_4 = [r_x, 0, 0]^T \tag{2.32b}$$

$$CB_1 = CC_{12} + C_{12}B_{12} + B_{12}B_1, \quad CB_2 = CC_{12} + C_{12}B_{12} + B_{12}B_2 \tag{2.33a}$$

$$CB_3 = CC_{34} + C_{34}B_{34} + B_{34}B_3, \quad CB_4 = CC_{34} + C_{34}B_{34} + B_{34}B_4 \tag{2.33b}$$

Now, the position vector of points $B_i$ in frame R can be obtained as follows:

$$B_i = C + CB_i \quad \forall \quad i = 1, 2, 3, 4 \tag{2.34}$$

Developing the previous equation for $i = 1 \ldots 4$, yielding the coordinates of all points $B_i$ as follows:

$$\begin{aligned} B_1 &= C + [r_x - r_h \sin(\psi), r_y - r_h + r_h \cos(\psi), 0]^T \\ B_2 &= C + [-r_x - r_h \sin(\psi), r_y - r_h + r_h \cos(\psi), 0]^T \\ B_3 &= C + [-r_x + r_h \sin(\psi), -(r_y - r_h) - r_h \cos(\psi), 0]^T \\ B_4 &= C + [r_x + r_h \sin(\psi), -(r_y - r_h) - r_h \cos(\psi), 0]^T \end{aligned} \tag{2.35}$$

As mentioned before, $\psi$ is equal to zero at all times, thus, the coordinates of points $B_i$ can be simply expressed as follows:

$$B_i = C + [sign(\cos(\alpha_i))r_x, sign(\sin\alpha_i)r_y, 0]^T \tag{2.36}$$

The actuated points $Q_i$ are expressed in the Cartesian reference frame R by the following vector:

$$Q_i = [sign(\cos(\alpha_i))R_x, sign(\sin(\alpha_i))R_y, 0]^T \tag{2.37}$$

For each kinematic chain, the coordinates of $A_i$ in the frame $R_i$ can be calculated from the given $q_i$ as follows:

$${}^{R_i}A_i = [L_i cos(q_i), 0, L_i sin(qi)]^T \tag{2.38}$$

As common for all parallel robots, their arms are rigid enough to have always fixed lengths. Therefore, the rigidity of the rear-arms and forearms of T3KR robot gives the following equalities respectively:

$$\|Q_i A_i\| = L_i \tag{2.39}$$

$$\|A_i B_i\| = l_i \tag{2.40}$$

#### 2.4.2.1 Inverse Kinematic Model of T3KR PKM

Starting from a known Cartesian pose vector X, one can compute the joint position vector q using the IKM. Developing (2.39) and (2.40) in the auxiliary frame $R_i$ leads to a system of two equations as follows:

$$u_{A_i}^2 + z_{A_i}^2 = L_i^2 \tag{2.41}$$

$$(u_{B_i} - u_{A_i})^2 + v_{B_i}^2 + (z_{B_i} - z_{A_i})^2 = l_i^2 \tag{2.42}$$

Solving these two equations for $u_{A_i}$ and $z_{A_i}$ in the auxiliary frame $R_i$ gives the intersection point $A_i$ between the circle of center $Q_i$ and radius $L_i$ that describes the motion of each rear-arm (2.41) and the sphere of center $B_i$ and radius $l_i$ that describes the motion of each

forearm (2.42). The coordinates of point $B_i$ with respect to the frame $R_i$ is obtained by the inverse of the transformation matrix T (2.29) as follows:

$$\begin{pmatrix} {}^{R_i}B_i \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\alpha_i) & -\sin(\alpha_i) & 0 & sign(\cos(\alpha_i))R_x \\ \sin(\alpha_i) & \cos(\alpha_i) & 0 & sign(\sin(\alpha_i))R_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} {}^{R}B_i \\ 1 \end{pmatrix} \tag{2.43}$$

Thus, the coordinates of the four actuated joints representing the inverse kinematic solution can be obtained as follows:

$$q_i = atan2(z_{A_i}, u_{A_i}) \tag{2.44}$$

#### 2.4.2.2 Forward Kinematic Model of T3KR PKM

The FKM of T3KR PKM provides the Cartesian pose vector of the platform starting from a known configuration of the four actuated joint angles. In Delta-like PKMs, the forearms are composed of two parallel bars whose purpose is to restrict the orientation of the moving plate, yielding a relatively straightforward solution for the FKM [Taghirad, 2013]. This solution lies in finding the intersection of n virtual spheres whose radius is equal to the length of the forearms. Hereafter, it is explained how the FKM of the T3KR robot can be obtained.

For each kinematic chain, the coordinates of points $A_i$ in the reference frame R can be computed from their coordinates in the auxiliary frame $R_i$ ($^{R_i}A_i$) (2.38) using the transformation matrix T (2.29) as follows:

$$\begin{pmatrix} A_i \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\alpha_i) & -\sin(\alpha_i) & 0 & sign(\cos(\alpha_i))R_x \\ \sin(\alpha_i) & \cos(\alpha_i) & 0 & sign(\sin(\alpha_i))R_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{R_i}A_i \\ 1 \end{pmatrix} \tag{2.45}$$

$$A_i = Q_i + L_i[\cos(\alpha_i)\cos(q_i), \sin(\alpha_i)\cos(q_i), \sin(q_i)]^T$$

Now, developing the equality that expresses the constant length of the forearms in Delta-like parallel robots (2.40) in the reference frame R leads to the following system of equations:

$$(x_{B_i} - x_{A_i})^2 + (y_{B_i} - y_{A_i})^2 + (z_{B_i} - z_{A_i})^2 = l_i^2 \quad \forall \quad i = 1,2,3 \tag{2.46}$$

These equations are known as the closed-loop equations which involve all geometrical variables of Delta-like parallel robots. Inserting the coordinates of $B_i$ in the reference frame

R (2.35) into this equation (2.46) leads to another system of four equations and four unknowns, $x, y, z$ and $\psi$, the Cartesian coordinates of P:

$$\begin{aligned}
&(x+r_x-r_h\sin(\psi)-x_{A_1})^2+(y+r_y-r_h+r_h\cos(\psi)-y_{A_1})^2+(z+r_z-z_{A_1})^2=l_1^2\\
&(x-r_x-r_h\sin(\psi)-x_{A_2})^2+(y+r_y-r_h+r_h\cos(\psi)-y_{A_2})^2+(z+r_z-z_{A_2})^2=l_2^2\\
&(x-r_x+r_h\sin(\psi)-x_{A_3})^2+(y-r_y+r_h-r_h\cos(\psi)-y_{A_3})^2+(z+r_z-z_{A_3})^2=l_3^2\\
&(x+r_x+r_h\sin(\psi)-x_{A_4})^2+(y-r_y+r_h-r_h\cos(\psi)-y_{A_4})^2+(z+r_z-z_{A_4})^2=l_4^2
\end{aligned} \tag{2.47}$$

where $l_1=l_2=l_3=l_4=l$ are the lengths of the forearms. For $\psi$ equal to zero, equation (2.46) can be simply written as follows:

$$(x+sign(cos(\alpha_i))r_x-x_{A_i})^2+(y+sign(sin(\alpha_i))r_y-y_{A_i})^2+(z+r_z-z_{A_i})^2=l_i^2 \tag{2.48}$$

The numerical solution of (2.47) gives the coordinates of P in the reference frame R which is represented by the intersection of four spheres of equations shown in (2.47).

### 2.4.3 Differential Kinematics of T3KR PKM

Differentiating with respect to time the kinematic relationship in (2.46) provides the relation between the operational velocity vector $\dot{X}$ and the joint velocity vector $\dot{q}$ using the Jacobian matrix $J(X,q)$. Furthermore, the Jacobian matrix constructs a transformation that relates the actuator torques and the forces acting on the traveling-plate. The resulting equality from the differentiation of (2.46) can be written as follows:

$$\begin{aligned}
&(x_{B_i}-x_{A_i})\dot{x}_{Bi}+(y_{B_i}-y_{A_i})\dot{y}_{B_i}+(z_{B_i}-z_{A_i})\dot{z}_{B_i}\\
&\qquad=(x_{B_i}-x_{A_i})\dot{x}_{A_i}+(y_{B_i}-y_{A_i})\dot{y}_{A_i}+(z_{B_i}-z_{A_i})\dot{z}_{A_i}
\end{aligned} \tag{2.49}$$

From (2.35), one can compute the time derivative of the position vector $B_i$ as a function of the Cartesian velocity vector of the end-effector $\dot{X}$ as follows:

$$\begin{aligned}
\dot{X}_{B_1}&=[\dot{x}-r_h\dot{\psi}\cos(\psi),\dot{y}-r_h\dot{\psi}\sin(\psi),\dot{z}]^T=[\dot{x},\dot{y},\dot{z}]^T+\dot{\psi}\,[-r_h\cos(\psi),-r_h\sin(\psi),0]^T\\
\dot{X}_{B_2}&=[\dot{x}-r_h\dot{\psi}\cos(\psi),\dot{y}-r_h\dot{\psi}\sin(\psi),\dot{z}]^T=[\dot{x},\dot{y},\dot{z}]^T+\dot{\psi}\,[-r_h\cos(\psi),-r_h\sin(\psi),0]^T\\
\dot{X}_{B_3}&=[\dot{x}+r_h\dot{\psi}\cos(\psi),\dot{y}+r_h\dot{\psi}\sin(\psi),\dot{z}]^T=[\dot{x},\dot{y},\dot{z}]^T+\dot{\psi}\,[r_h\cos(\psi),r_h\sin(\psi),0]^T\\
\dot{X}_{B_4}&=[\dot{x}+r_h\dot{\psi}\cos(\psi),\dot{y}+r_h\dot{\psi}\sin(\psi),\dot{z}]^T=[\dot{x},\dot{y},\dot{z}]^T+\dot{\psi}\,[r_h\cos(\psi),r_h\sin(\psi),0]^T
\end{aligned} \tag{2.50}$$

The Cartesian velocity of $A_i$ can be derived from (2.45) as follows:

$$\dot{X}_{A_i}=t_i\dot{q}_i \tag{2.51}$$

where $t_i$ is the tangent vector at point $A_i$ to the circle of the rear-arm given as follows:

$$t_i = [-L\sin(q_i)\cos(\alpha_i), -L\sin(q_i)\sin(\alpha_i), L\cos(q_i)]^T \tag{2.52}$$

Therefore, (2.49) can be arranged and rewritten in the form below:

$$J_x \dot{X} = J_q \dot{q} \tag{2.53}$$

where $J_q$ and $J_x$ are calculated as follows:

$$J_x = \begin{pmatrix} A_1B_1^T & -r_h\cos(\psi)(x_{B_1} - x_{A_1}) - r_h\sin(\psi)(y_{B_1} - y_{A_1}) \\ A_1B_2^T & -r_h\cos(\psi)(x_{B_2} - x_{A_2}) - r_h\sin(\psi)(y_{B_2} - y_{A_2}) \\ A_1B_3^T & r_h\cos(\psi)(x_{B_3} - x_{A_3}) + r_h\sin(\psi)(y_{B_3} - y_{A_3}) \\ A_1B_4^T & r_h\cos(\psi)(x_{B_4} - x_{A_4}) + r_h\sin(\psi)(y_{B_4} - y_{A_4}) \end{pmatrix} \tag{2.54}$$

$$J_q = diag\{t_1^T A_1B_1,\ t_2^T A_2B_2,\ t_3^T A_3B_3,\ t_4^T A_4B_4\} \tag{2.55}$$

Finally, the Jacobian matrix is computed as follows:

$$J = J_x^{-1} J_q \tag{2.56}$$

It is worth to note that for T3KR robot, the Jacobian matrix is square (4x4), thus, the inverse of $J_x$ always exists as long as the robot follows trajectories away from singularities. The differential kinematic model of the T3KR robot is given by the following equations:

$$\dot{X} = J\dot{q} \tag{2.57}$$

$$\dot{q} = J_m\dot{X} = J^{-1}\dot{X} \tag{2.58}$$

### 2.4.4 Dynamics of T3KR PKM

Before proceeding with the development of the dynamic modeling of the T3KR robot, the following simplifying assumptions, which are common to Delta-like PKMs, must be considered, to obtain a suitable dynamic model appropriate for real-time implementation [Pierrot et al., 1990]:

**Assumption 2.4.1.** *Both dry and viscous frictions in all passive and active joints are neglected.*

**Assumption 2.4.2.** *The rotational inertia of the forearm is neglected. Nevertheless, its mass is split up into two point-masses, one is added to the rear-arm mass, and the other one is joined to the traveling plate mass (see Figure 2.11).*

The second simplification is justified only if the ratio between the mass of the forearms

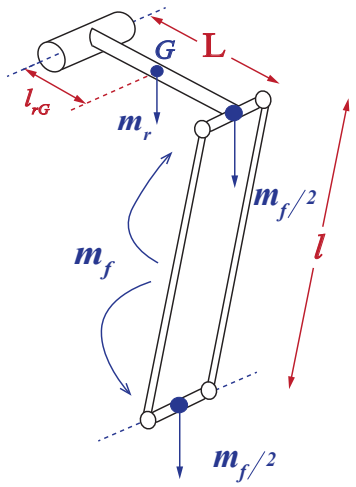



Figure 2.11 – Illustration of dynamic parameters of T3KR robot arm.

and the mass of the rear-arms is sufficiently small (less than 0.3), as it is established on the analysis study developed for PKMs with delta-like structure in [Nabat, 2007]. Regarding the dynamics parameters of T3KR presented in Table. 2.1, the mass of one rear-arm is 3.28 Kg and the mass of one forearm is 0.8 kg considering the two parallel bars yielding a ratio of 0.2439 which is less than 0.3. Therefore, Assumption 2.4.2 can be considered for the T3KR dynamic model.

The dynamic model of T3KR robot is established, as in [Saied, 2019], based on the virtual work principle described in [Codourey, 1998]. It can be reduced to analyzing the dynamics of mainly two bodies, namely the traveling-plate and the main actuators in conjunction with their corresponding rear-arms.

Regarding the moving platform dynamics, one can define two kinds of forces acting on it: the gravitational force $G_{tp} \in \mathbb{R}^4$, and the inertial force $F_{tp} \in \mathbb{R}^4$.

The gravitational force $G_{tp}$ can be denoted by the following equation:

$$G_{tp} = -M_{tp} G \tag{2.59}$$

where $M_{tp} = diag\{m_{tp}, m_{tp}, m_{tp}, 0\}$ being $m_{tp} = m_n + 4\frac{m_f}{2}$ is the total mass of the traveling-plate. It involves the mass of the nacelle including the mass of the actuator onboard the platform and the half-masses of the four forearms as shown in Figure 2.11. $G \in \mathbb{R}^4$ is the gravity vector, $G = [0, 0, g, 0]^T$.

The inertial force $F_{tp}$ arising from the mobile platform acceleration is defined as fol-

lows:

$$F_{tp} = M_{tp}\ddot{X} \tag{2.60}$$

where $\ddot{X} \in \mathbb{R}^4$ denotes the acceleration vector. The contributions of the gravitational and inertial forces to the actuator torques are computed using the Jacobian matrix as follows:

$$\Gamma_{G_{tp}} = J^T G_{tp} \tag{2.61}$$

$$\Gamma_{F_{tp}} = J^T F_{tp} \tag{2.62}$$

Regarding the rear-arms dynamics, we consider three contributing torques. Firstly, the Firstly, the actuators input torque $\Gamma \in \mathbb{R}^4$. Secondly, the torque produced by the gravitational forces acting on the rear-arms $\Gamma_{G_{arm}} \in \mathbb{R}^4$ is given by:

$$\Gamma_{G_{arm}} = -g\,Mr\,Cos(q) \tag{2.63}$$

$$M_r = diag\{m_{req}, m_{req}, m_{req}, m_{req}\} \tag{2.64}$$

where $m_{req} = m_r L_{rG} + L\dfrac{m_f}{2}$, $m_r$ is the mass of each rear-arm, $l_{rG}$ is the distance from the axis of rotation of each rear-arm to its center of gravity, while L is the complete length of each rear-arm as illustrated in Figure 2.11. $m_f$ is the mass of each forearm as shown in Figure 2.11 and $Cos(q) \triangleq [cos(q_1), cos(q_2), cos(q_3), cos(q_4)]$. Thirdly, the inertial contribution torque due to rear-arms acceleration $\Gamma_{arm} \in \mathbb{R}^4$ is defined as follows:

$$\Gamma_{arm} = I_{arm}\ddot{q} \tag{2.65}$$

where $\ddot{q} \in \mathbb{R}^4$ is the acceleration vector in joint space and $I_{arm} \in \mathbb{R}^{4x4}$ is a diagonal matrix whose elements are formed by:

$$I_{arm} = I_{act} + I_{ra} + \frac{L^2 m_f}{2} \tag{2.66}$$

where $I_{act}$ and $I_{ra}$ represent the actuators inertia and the reararms inertia, respectively. The term $\dfrac{L_i^2 m_f}{2}$ corresponds to the inertial contribution of the forearms taking into consideration the statement of Assumption 2.4.2. All these inertia values are computed with respect to the actuator rotation axes.

After applying the virtual work principle, stating that the contribution of all non-inertial forces must be equal to the contribution of all inertial forces, one can formulate the inverse dynamic model of T3KR PKM as follows:

$$\begin{aligned} \Gamma + \Gamma_{G_{arm}} + \Gamma_{G_{tp}} &= \Gamma_{arm} + \Gamma_{F_{tp}} \\ \Gamma &= I_{arm}\ddot{q} + J^T M_{tp}\ddot{X} - \Gamma_{G_{arm}} - \Gamma_{G_{tp}} \end{aligned} \tag{2.67}$$

Table 2.1 – Summary of the main geometric and dynamic parameters of T3KR PKM.

| Parameter | Description | Value | Parameter | Description | Value |
|---|---|---|---|---|---|
| $L$ | Rear-arm length | 400 mm | $m_r$ | Rear-arm mass | 3.78 kg |
| $l$ | Fore-arm length | 900 mm | $m_f$ | Forearm mass | 0.8 kg |
| $r_z$ | End-effector length | 50 mm | $m_n$ | Traveling-plate mass | 5.28 kg |
| $I_{ra}$ | Rear-arm inertia | 1.737 kg.m$^2$ | $I_{act}$ | Actuator inertia | 2.732 kg.m$^2$ |

To express the dynamic model in joint space, we compute the first time derivative of (2.57) to obtain the relation between joint and Cartesian accelerations as follows:

$$\ddot{X} = J\ddot{q} + \dot{J}\dot{q} \tag{2.68}$$

where $\dot{J}$ is the time derivative of $J$.

After substituting (2.68) in (2.67) and rearranging the terms, the inverse dynamic model of T3KR robot can be expressed in terms of the joint coordinates $q$ as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \Gamma(t) \tag{2.69}$$

where $M(q) = I_{arm} + J^T M_{tp} J$ is the total mass and inertia matrix of the robot, $C(q,\dot{q})\dot{q} = J^T M_{tp}\dot{J}$ is the Coriolis and centrifugal forces matrix, $G(q) = -\Gamma_{G_{arm}} - \Gamma_{G_{tp}}$ is the gravitational forces vector, and $\Gamma(t)$ is the control input vector. The main geometric and dynamic parameters of T3KR parallel robot are summarized in Table. 2.1.

## 2.5 Real-time experimental results

In this section, the obtained experimental results of the proposed P&T method are presented and compared with the standard P&P method and with an existing P&T approach from the literature.

### 2.5.1 Description of the experimental testbed

T3KR parallel robot uses five actuators to perform the mobile platform and the end-effector movements. Four WITTESTEIN TPMA025S-022M actuators allow translational motions along $x$, $y$ and $z$ axes and a rotational motion of the platform around the $z$ axis. A gearbox of ratio 1:22 is integrated in each actuator, providing a maximum torque of 530 Nm and a maximum joint rotation speed of 220 rpm. The actuator mounted on the platform is WITTESTEIN TPM+ dynamic 004 with a gear ratio of 1:16. This actuator is responsible for the rotation of the end-effector around the $z$ axis and can provide a peak torque

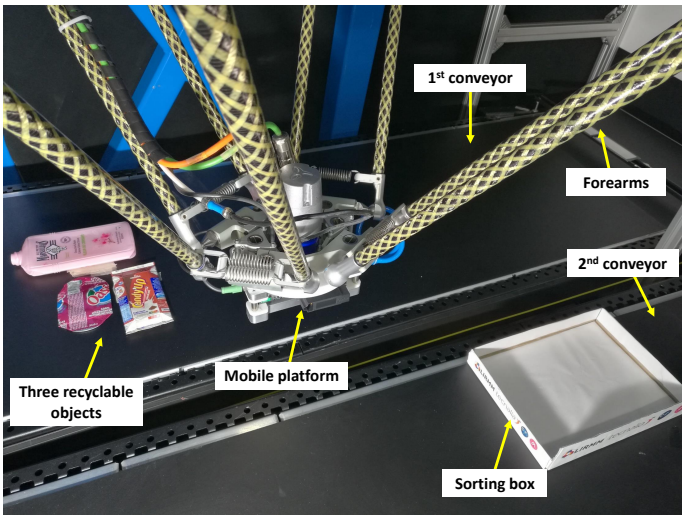



Figure 2.12 – The experimental testbed: T3KR robot, conveyors, recyclable objects and the sorting box.

of about 30 Nm and a maximum speed of about 375 rpm. The joint positions are measured by means of multi-turn absolute encoders. The robot is controlled by an industrial PC (5PC910) equipped with the B&R Automation studio with a sampling frequency of 2500 Hz (i.e. a sampling time of 0.4 ms). The motors motion control is performed by a PID controller integrated in ACOPOSmulti B&R drive. The mobile platform can reach a maximum speed of 6 m/s (i.e. $V_{max} = 6\,\mathrm{m/s}$), a maximum acceleration of 12 G (i.e. $A_{max} = 12\mathrm{G} = 120\,\mathrm{m/s^2}$) and carry a maximum payload of 5 kg. The mobile platform has a total mass of about 5.68 kg. The robot is equipped with a Cognex smart camera. The experimental setup is displayed in Figure 2.12. It includes the T3KR robot, two conveyors, three objects (recyclables) of different masses, sizes and materials, and a sorting box to deposit the objects.

## 2.5.2 Description of P&P and P&T reference trajectories

To validate the efficiency of the proposed P&T technique, i.e., the minimum-time throw motion introduced in Section 2.3, comparisons with a standard P&P method as the one used in [Nabat et al., 2005] and with the P&T approach proposed in [Raptopoulos et al., 2020] are conducted in real-time experiments. The three associated reference trajectories are described hereafter. All the length units are in meters.

### 2.5.2.1 Pick-and-Place reference trajectory

The 3D view of the adopted P&P trajectory in Cartesian space is depicted in Figure 2.13. A standard P&P trajectory consists of a vertical movement, followed by a hori-

zontal movement and a final vertical movement. The robot moves from the initial position $\mathbf{P}_{initial} = [0, -0.77, 0]$ to the first pick position $\mathbf{P}_{pick1} = [-0.2, -0.25, -0.85]$ (the conveyor height is $-0.85$). After picking the object, it follows the first P&P trajectory (shown in the red) to place the object at $\mathbf{P}_{place} = [0.3, 0.4, -0.85]$. Then, the robot follows the second P&P trajectory (shown in green) to pick the second object at $\mathbf{P}_{pick2} = [-0.2, -0.35, -0.85]$ and moves back to $\mathbf{P}_{place}$. The same movement is repeated for the third object located at $\mathbf{P}_{pick3} = [-0.3, -0.35, -0.85]$. After placing the third object, the robot goes back to the initial position $\mathbf{P}_{initial}$.

#### 2.5.2.2 Existing Pick-and-Throw reference trajectory

The P&T trajectory used in [Raptopoulos et al., 2020] is shown in Figure 2.14. This trajectory is similar to a P&P trajectory and can be described as follows: After moving the end effector from its initial position $\mathbf{P}_{initial}$ to the first pick position $\mathbf{P}_{pick1}$, the robot performs a vertical movement followed by a horizontal movement towards the release point. It throws the object to the target position $\mathbf{P}_f = [0.3, 0.4, -0.85]$ along the horizontal movement and then makes a U-turn to decelerate towards the next pick position $\mathbf{P}_{pick2}$. The same movement is repeated for the third object located at $\mathbf{P}_{pick3}$. After throwing the third object, the robot returns back to its initial position. In [Raptopoulos et al., 2020], the path of the P&T trajectory is presented without describing the release point computation. In the present experiments, the release points are computed with the methodology proposed in Section 2.3 where only the horizontal distance to the release position has to be optimized because, in the P&T trajectory of [Raptopoulos et al., 2020], $Z_r$ is fixed, $V_{zr} = 0$ m/s and $\theta_r = 0$ deg.

#### 2.5.2.3 Proposed Pick-and-Throw reference trajectory

The P&T trajectory proposed in this thesis is depicted in Figure 2.15. The robot follows the first picking motion from $\mathbf{P}_{initial}$ to $\mathbf{P}_{pick_1}$. After picking the object, the optimal release configuration, including $\mathbf{P}_{throw_1} = \mathbf{P}_{r1}$ and $\mathbf{V}_{r1}$, is calculated as described in Section 2.3. The robot accelerates while moving along a straight line towards the calculated release point $\mathbf{P}_{throw_1}$ at which it throws the object towards the target $\mathbf{P}_f$. Once released, the object follows a ballistic trajectory to $\mathbf{P}_f$ while the robot decelerates back to pick the second object. The same cyclic movement is repeated for the second and the third objects, located at $\mathbf{P}_{pick_2}$ and $\mathbf{P}_{pick_3}$, respectively. After throwing the third object, the robot moves back to $\mathbf{P}_{initial}$. As a numerical example, in a case study with 30% of the maximum dynamic performances of the T3KR robot (i.e. 30% of $V_{max} = 6\,m/s$, $A_{max} = 12\,m/s^2$ and $J = 3000\,m/s^3$) and for a target position inside its workspace, $P_f = [0.3, 0.4, -0.85]$, the following optimal release configurations are calculated with the method proposed in Section 2.3: $\mathbf{P}_{throw_1} = [0.09, 0.13, -0.75]$ and $V_{r1} = 1.8\,m/s$, $\mathbf{P}_{throw_2} = [0.15, 0.17, -0.75]$ and $V_{r2} = 1.8\,m/s$, $\mathbf{P}_{throw_3} = [0.12, 0.17, -0.75]$ and $V_{r3} = 1.8\,m/s$.

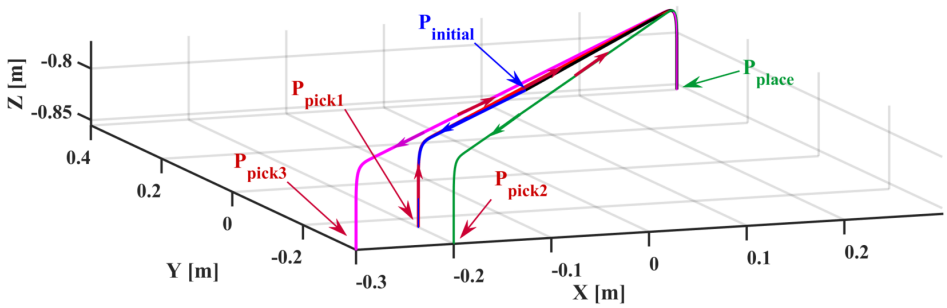



Figure 2.13 – 3D-view of P&P reference trajectory in Cartesian space.

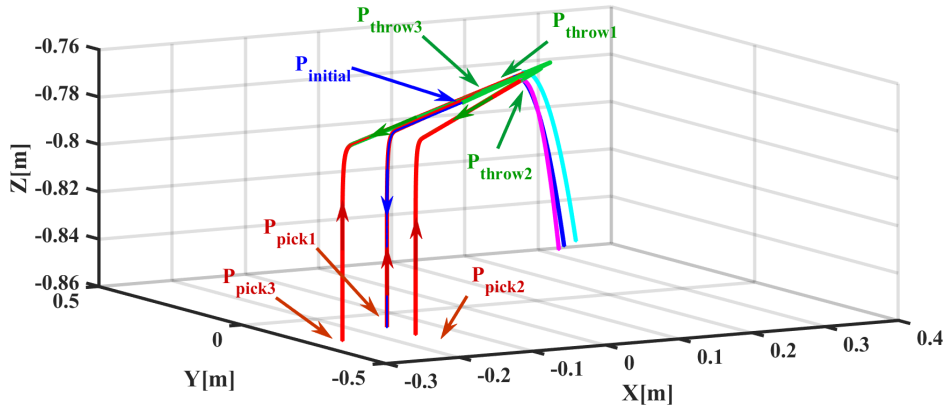



Figure 2.14 – 3D-view of existing P&T reference trajectory [Raptopoulos et al., 2020].

### 2.5.3 Obtained Experimental Results

The performance of the proposed P&T approach is evaluated through three experimental scenarios. These scenarios are performed under different operating conditions (acceleration, speed, and different types of objects of different sizes, materials and masses, etc.). The three types of objects considered are: A paper object of 7 g of mass, a metallic can with a mass of 16 g ($\Delta_{mass} = +128\%$ w.r.t the first object) and a plastic package of 49 g of mass ($\Delta_{mass} = +600\%$ w.r.t the first object). The demonstration video of the three tested scenarios is available at: https://youtu.be/4bRvqKurMJU. These three scenarios are detailed in the sequel.

#### 2.5.3.1 Scenario 1

It consists in comparing the usual P&P (cf. Section 2.5.2.1) and the proposed P&T (cf. Section 2.5.2.3) trajectories inside the workspace, i.e., both the pick and the target positions are located inside the robot workspace. For comparison purposes, we use different percentages of the maximum speed ($V_{max} = 6\,m/s$) and maximum acceleration

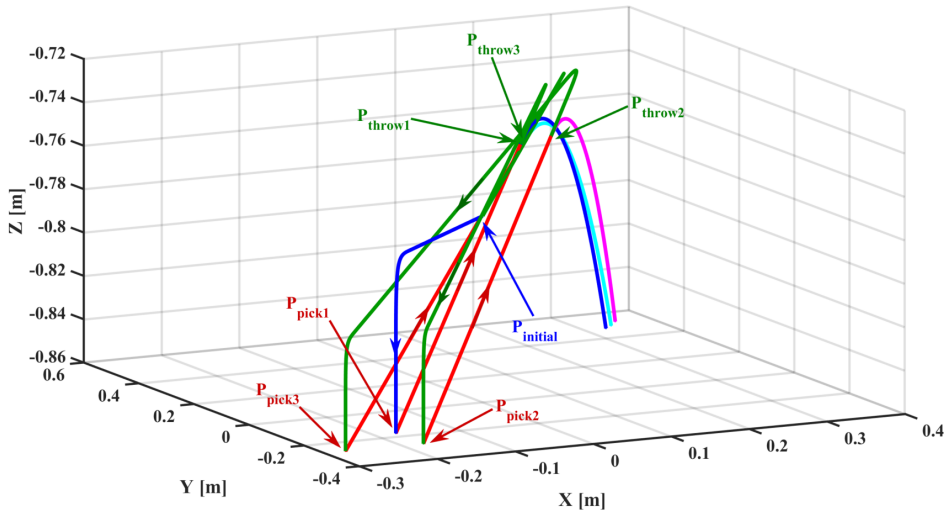



Figure 2.15 – 3D-view of the proposed P&T reference trajectory.

($A_{max} = 12$ G) of T3KR end-effector. These percentages are 15%, 30% and 40%, which correspond to 0.9 m/s, 1.8 m/s and 2.4 m/s of maximum velocity, respectively and to 1.8 G, 3.6 G and 4.8 G of maximum accelerations, respectively. The desired Cartesian positions, velocities and accelerations, along the $x$, $y$, and $z$ axes, generated using the minimum-time S-curve motion profile, are shown on the left side of Figures 2.16, 2.17, and 2.18 for the P&P tasks and on the right side of Figures 2.16, 2.17, and 2.18 for the proposed P&T tasks inside the workspace.

#### 2.5.3.2 Scenario 2

This scenario consists in comparing, inside the robot workspace, the proposed P&T method with the existing P&T strategy of [Raptopoulos et al., 2020] (cf. Section 2.5.2.2). As in the previous scenario, this test is performed under 15%, 30% and 40% of maximum acceleration and maximum velocity. The desired Cartesian positions, velocities, and accelerations of the existing P&T strategy are depicted on the left side of Figures. 2.19.

#### 2.5.3.3 Scenario 3

In this scenario, the proposed P&T method is compared with the existing P&T method of [Raptopoulos et al., 2020], but with a target position outside the robot workspace. The generated P&T trajectories are executed with 35%, 40% and 45% of maximum velocity and maximum acceleration, which corresponds to 2.1 m/s, 2.4 m/s and 2.7 m/s of maximum velocity, respectively and to 4.2 G, 4.8 G and 5.4 G of maximum acceleration, respectively. We start with a percentage of 35% in this scenario because, with a lower percentage, the speed is not sufficient to throw the object towards the desired target located outside the

workspace, $P_f = [0.45, 0.65, -0.85]$. The evolution of the desired positions, velocities, and accelerations is depicted on the left side of Figures 2.22, 2.20, 2.21 for the existing P&T method [Raptopoulos et al., 2020] and on the right side of Figures 2.22, 2.20, 2.21 for the proposed P&T approach. As it can be seen, with the increase in the operating acceleration, the calculated release point is closer to the picking position. Therefore, the duration of the whole trajectory is reduced and the robot can perform more picks per minute.

### 2.5.4 Results discussion

The obtained experimental results demonstrate the superiority of the proposed P&T approach over the standard P&P method and the existing P&T method of [Raptopoulos et al., 2020]. The number of picks per minute obtained by each method in each case study are summarized in the Table 2.2. Inside the workspace, using the P&P method, the robot can perform 24 to 51 picks per minute when the acceleration increases from 1.8 G to 4.8 G. While for the existing P&T method, it can perform 29 to 75 picks/min. However, with the proposed P&T approach, the obtained number of picks per minute goes from 30 to 120. Therefore, the proposed P&T method outperforms the two other strategies. At 4.8 G of maximum acceleration, the existing P&T method increases the number of picks per minute by up to 32% compared to the P&P approach. The proposed P&T strategy improves the performance by 57.5% over the standard P&P approach and by 37% over the existing P&T method, which are significant performance improvements for applications requiring reduced processing time and high productivity. Moreover, the improvement brought by the proposed P&T method over the existing approaches illustrate the relevance of determining an optimal release configuration. Regarding scenario 3, the P&T is the only candidate to perform such a task since the P&P method cannot place an object outside the workspace of the robot. At an acceleration of 5.4 G, the robot can reach 65 picks per minute by adopting the existing P&T method and 81 picks per minute by using the proposed P&T method. Therefore, the proposed P&T method largely outperforms the two other methods in all case studies.

All the presented experimental results confirm clearly the continuity of the proposed P&T trajectory as well as its time optimality, since the velocities and accelerations along the three axes, for the three scenarios, reach their corresponding maximum values. To further clarify this point, we propose to analyze the experimental results of one case study corresponding to the execution of the proposed P&T method for a target position outside the workspace, with 2.1 m/s of maximum velocity and 4.2 G of maximum acceleration. Figures 2.25, 2.26, and 2.27 hereafter depict the desired Cartesian positions, velocities, and accelerations of the robot end-effector along the $x$, $y$ and $z$ axes of this case study. Figure 2.28 displays the time evolution of the resulting velocity and acceleration of the robot end-effector. The proposed trajectory is optimized in a way that the resulting velocity and acceleration of the end-effector do not exceed their maximum values (i.e. 2.1 m/s for velocity and 4.2 G for acceleration in this case study). The time evolution of the resulting

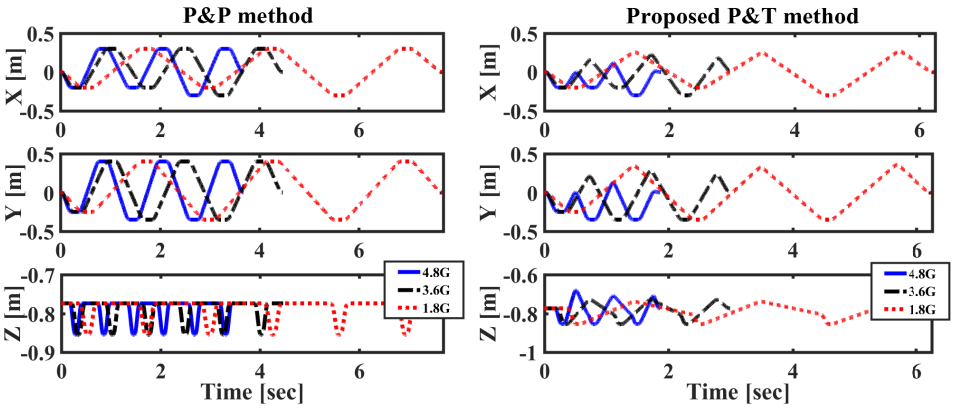



Figure 2.16 – Scenario 1: Evolution of the desired robot's end-effector Cartesian positions versus time for the P&P task [Nabat et al., 2005] and the proposed P&T task for different values of the maximum acceleration.

Table 2.2 – Number of picks per minutes versus operating acceleration for the three case studies.

| | | **1.8 G** | **3.6 G** | **4.2 G** | **4.8 G** | **5.4 G** |
|---|---|---|---|---|---|---|
| Inside Workspace | P&P | 24 | 42 | - | 51 | - |
| | Existing P&T | 29 | 56 | - | 75 | - |
| | Proposed P&T | 30 | 68 | - | 120 | - |
| Outside Workspace | Existing P&T | - | - | 54 | 60 | 65 |
| | Proposed P&T | - | - | 62 | 71 | 81 |

velocity $V_s(t)$ and acceleration $A_s(t)$ of the end-effector can be calculated from the individual velocity and acceleration along each axis based on the following relationships:

$$V_s(t) = \sqrt{V_x(t)^2 + V_y(t)^2 + V_z(t)^2} \tag{2.70}$$

$$A_s(t) = \sqrt{A_x(t)^2 + A_y(t)^2 + A_z(t)^2} \tag{2.71}$$

As shown in Figure 2.28, the velocity and acceleration of the end-effector reach their maximum constraints (i.e. the maximum of $V_s(t)$ is 2.1 m/s, while the maximum of $A_s(t)$ is 42 m/s$^2$, equivalent to 4.2 G). Figure 2.29 shows the time of the three-phase movements (i.e. the acceleration phase, the deceleration phase and the vertical movement) as well as the time of the two overlaps (the first overlap is between the acceleration and deceleration phases, while the second one is between the deceleration phase and the vertical movement). The values of these different times are reported in Table 2.3.

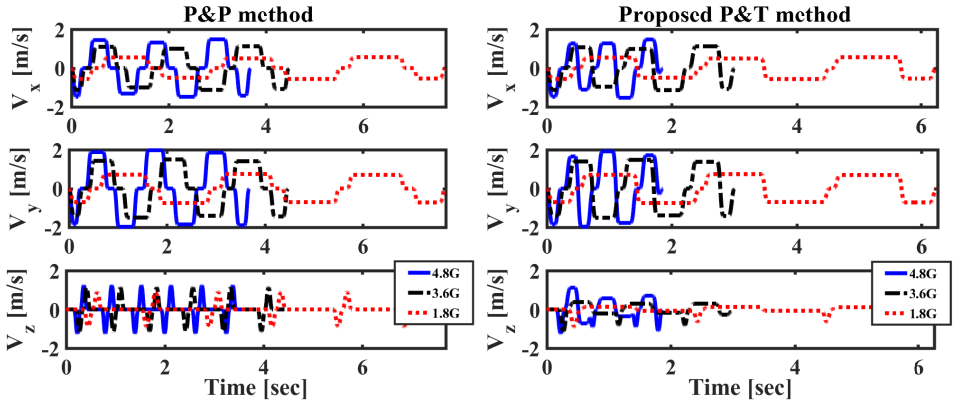



Figure 2.17 – Scenario 1: Evolution of the desired robot's end-effector Cartesian velocities versus time for the P&P task [Nabat et al., 2005] and the proposed P&T task for different values of the maximum acceleration.

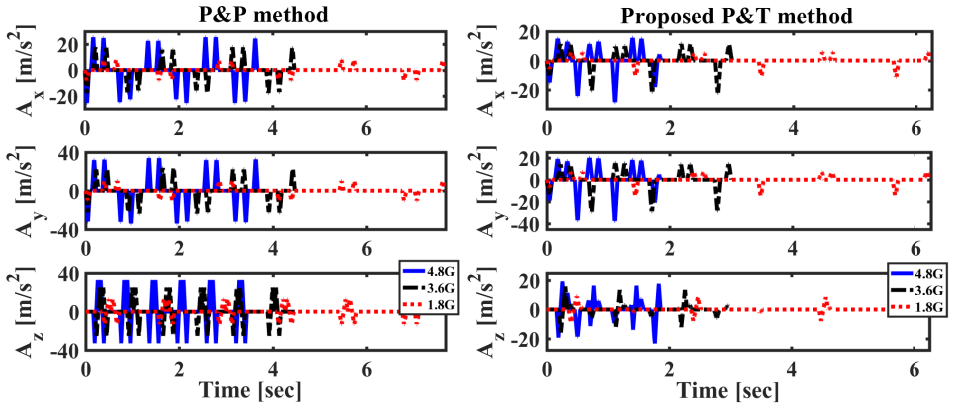



Figure 2.18 – Scenario 1: Evolution of the desired robot's end-effector Cartesian acceleration versus time for the P&P task [Nabat et al., 2005] and the proposed P&T task for different values of the maximum acceleration.

Table 2.3 – Time of the three phase movements

| Acceleration phase | Deceleration phase | Vertical movement | First overlap | Second overlap |
|---|---|---|---|---|
| 0.5354 [sec] | 0.5604 [sec] | 0.1151 [sec] | 0.0900 [sec] | 0.0575 [sec] |

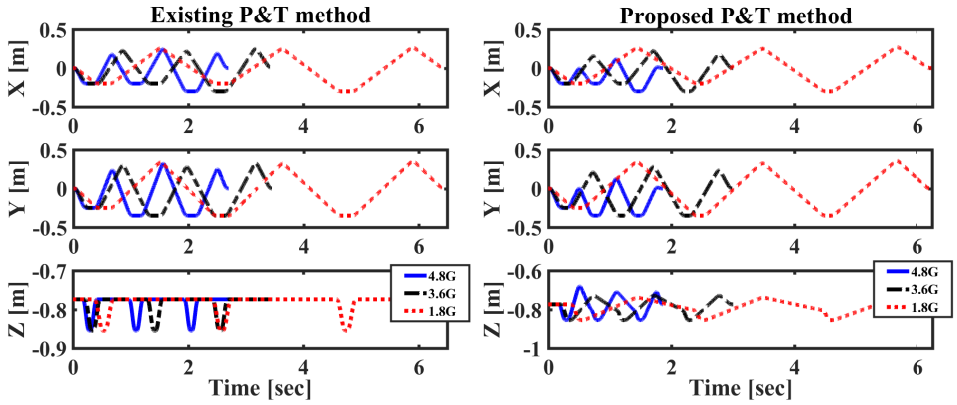



Figure 2.19 – Scenario 2: Evolution of the desired robot's end-effector Cartesian positions versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration.

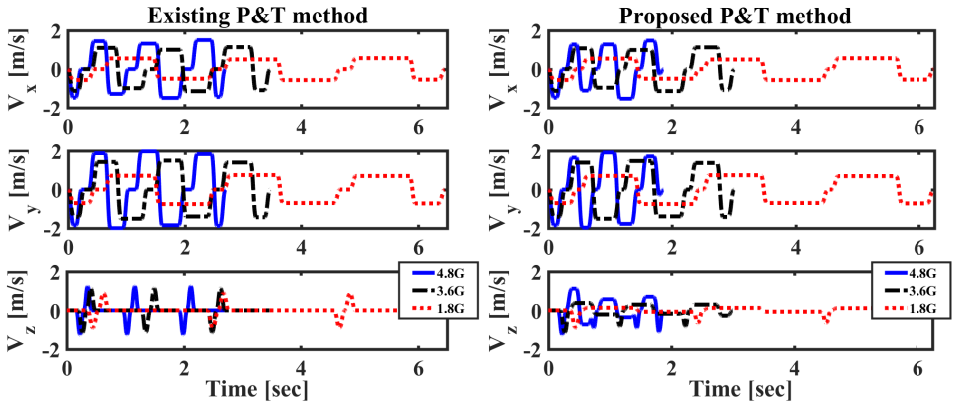



Figure 2.20 – Scenario 2: Evolution of the desired robot's end-effector Cartesian velocities versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration.

## 2.6 Conclusion

In this chapter, a new approach to generate a time-optimal throwing trajectory has been proposed. This approach consists in determining an optimal release configuration resulting in a minimum-time throw motion while ensuring that the released object will reach the desired target. The corresponding optimization problem has been formulated in two different ways where the second one allows the optimal release configuration to be efficiently computed. A comparison between the proposed P&T procedure, an existing P&T method and the conventional P&P has been conducted through real-time experiments on the T3KR parallel robot. The obtained experimental results validate the efficiency of the

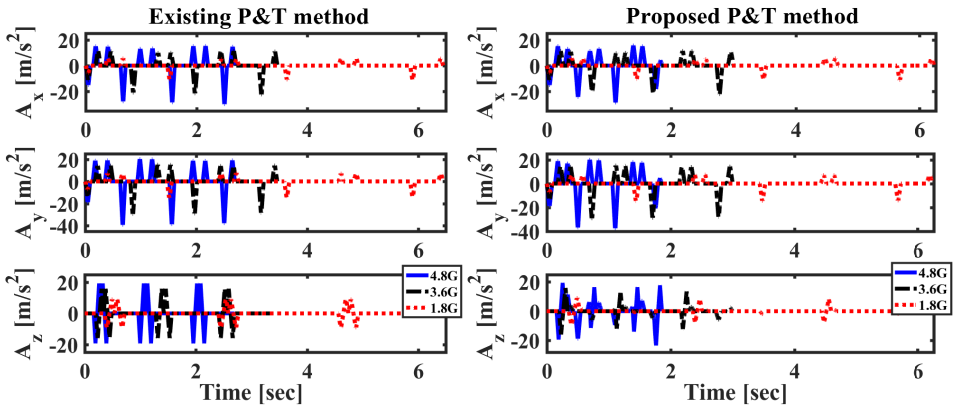



Figure 2.21 – Scenario 2: Evolution of the desired robot's end-effector Cartesian accelerations versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration.

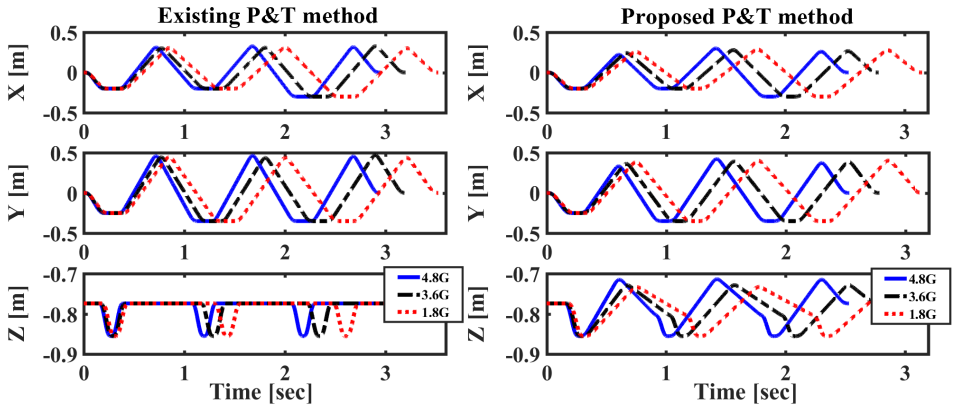



Figure 2.22 – Scenario 3: Evolution of the desired robot's end-effector Cartesian positions versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration.

proposed P&T method, over the two other methods, in terms of processing time minimization and, thereby, of productivity maximization.

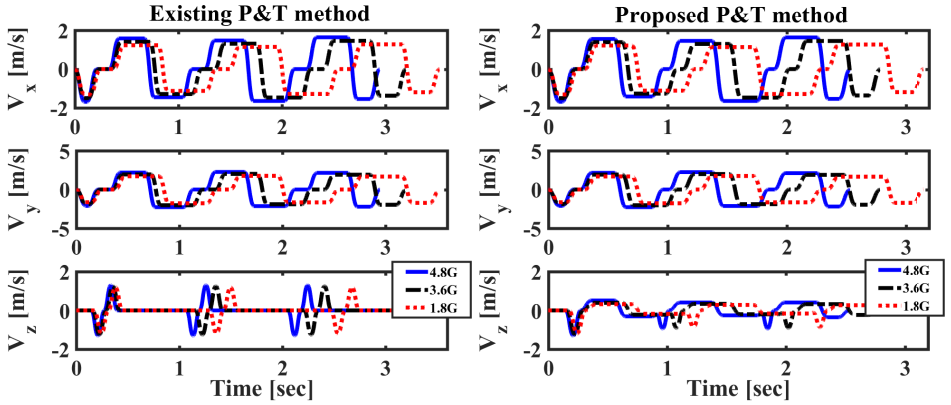



Figure 2.23 – Scenario 3: Evolution of the desired robot's end-effector Cartesian velocities versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration.

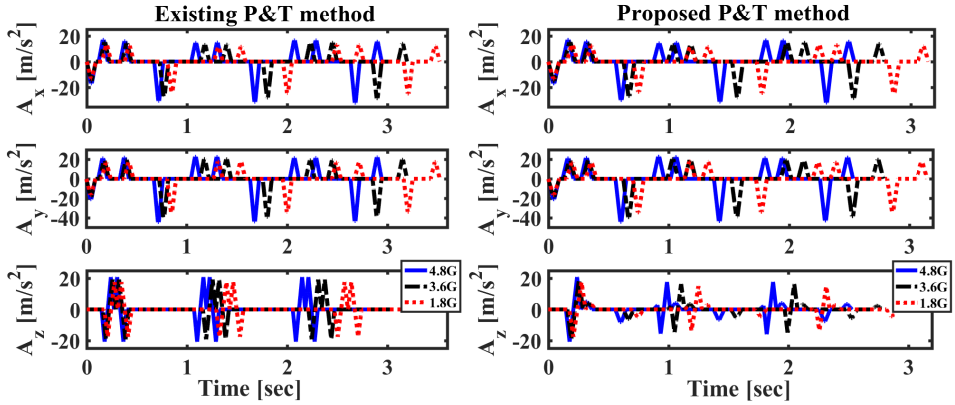



Figure 2.24 – Scenario 3: Evolution of the desired robot's end-effector Cartesian accelerations versus time for the existing P&T method [Raptopoulos et al., 2020] and the proposed P&T one for different values of the maximum acceleration.

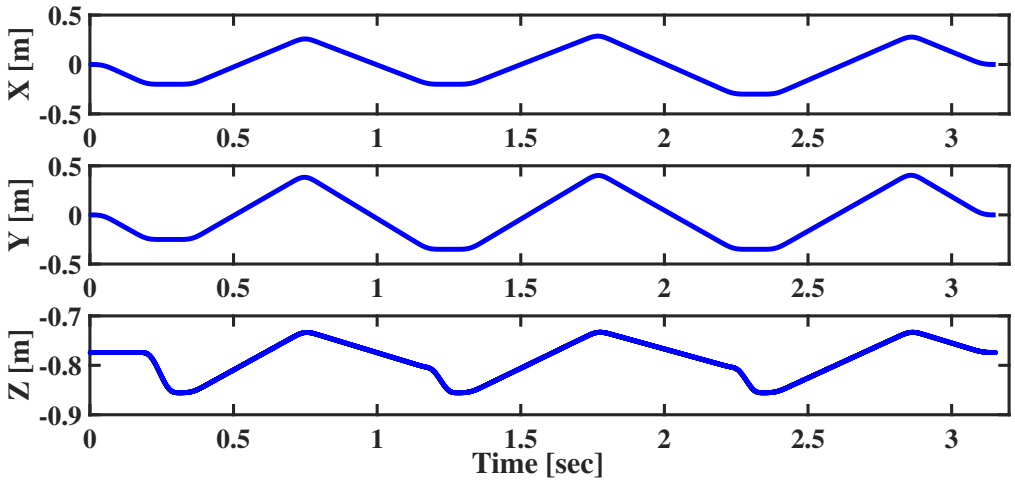



Figure 2.25 – Evolution versus time of the desired robot's end-effector Cartesian positions for the proposed P&T method outside the robot's workspace at 2.1 m/s of maximum velocity and 4.2 G of maximum acceleration.

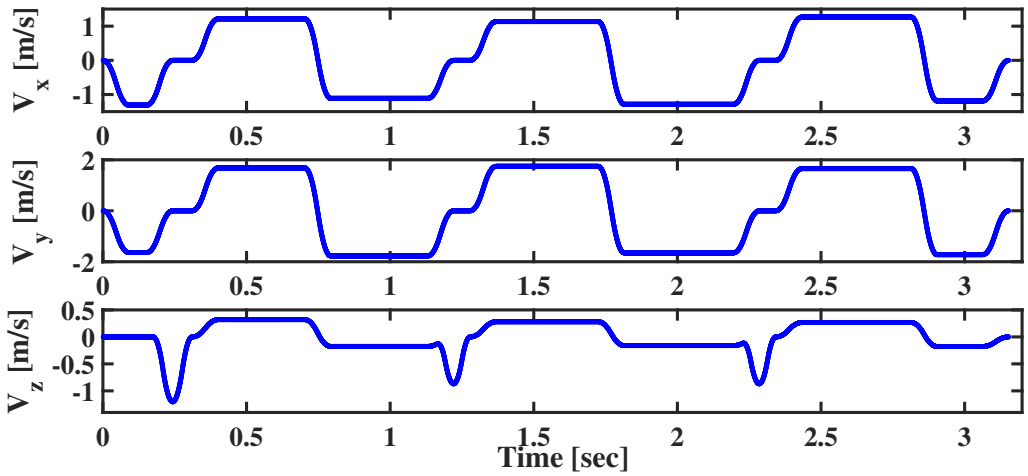



Figure 2.26 – Evolution versus time of the desired robot's end-effector Cartesian velocities for the proposed P&T method outside the robot's workspace at 2.1 m/s of maximum velocity and 4.2 G of maximum acceleration.

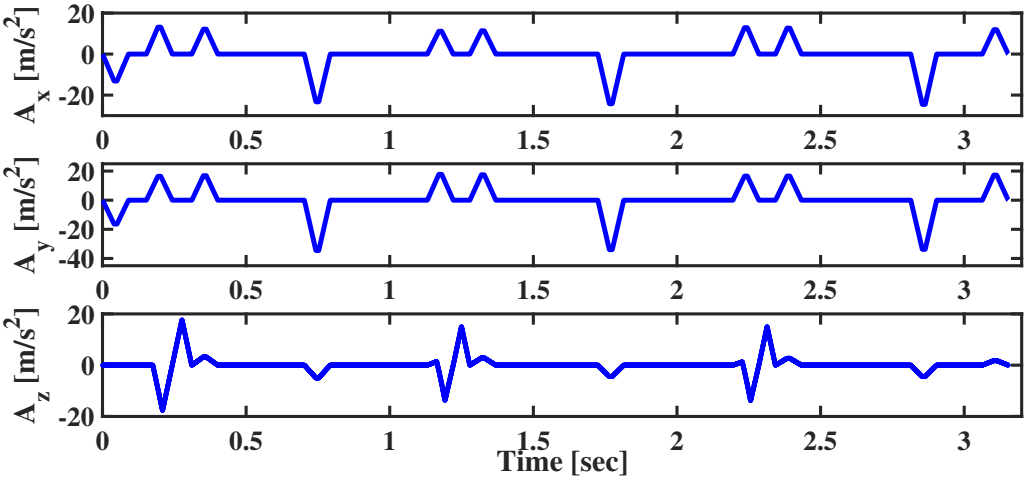



Figure 2.27 – Evolution versus time of the desired robot's end-effector Cartesian accelerations for the proposed P&T method outside the robot's workspace at 2.1 m/s of maximum velocity and 4.2 G of maximum acceleration.

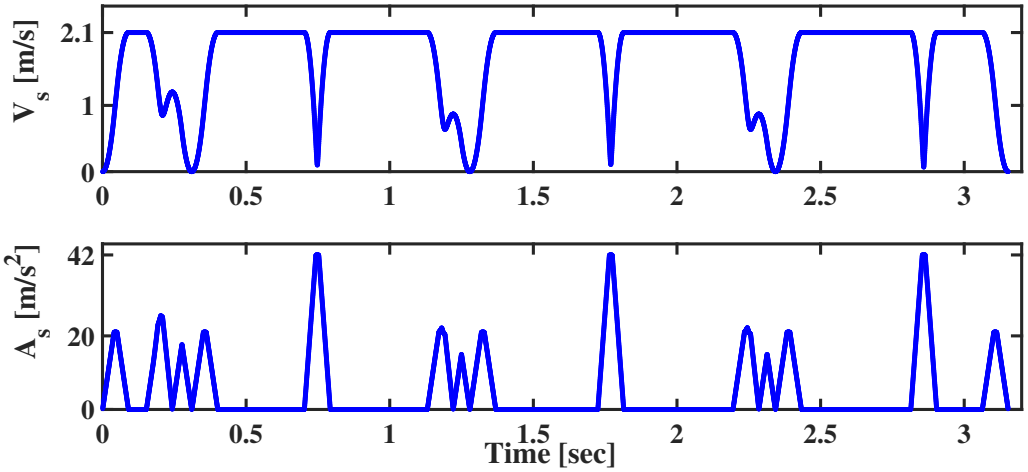



Figure 2.28 – Evolution versus time of the resulting desired Cartesian velocity and acceleration of the robot's end-effector for the proposed P&T method outside the robot's workspace at 2.1 m/s of maximum velocity and 4.2 G of maximum acceleration.

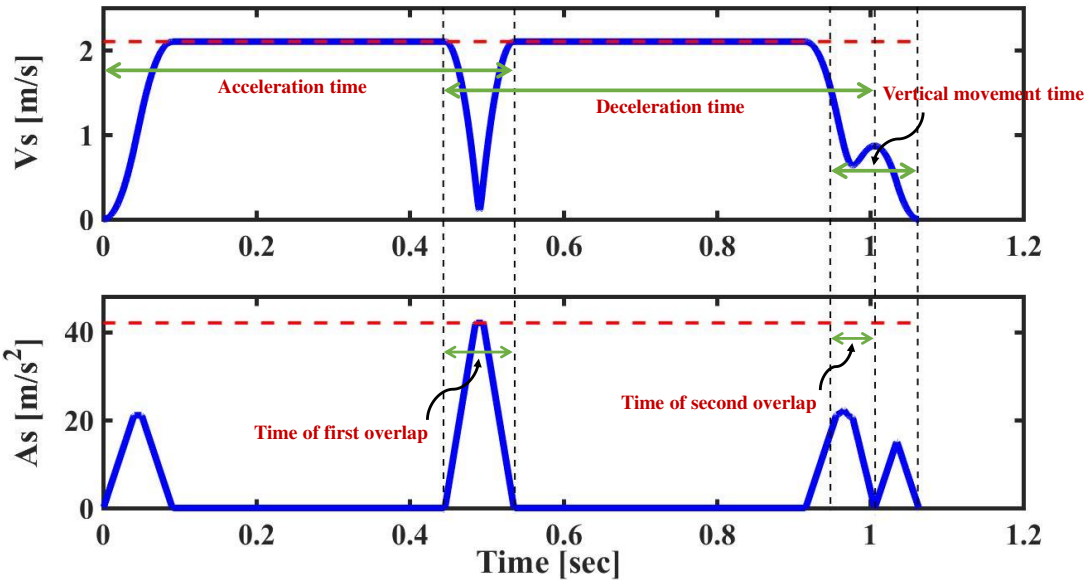



Figure 2.29 – Evolution versus time of the resulting desired Cartesian velocity and acceleration of the robot's end-effector for the proposed P&T method outside the robot's workspace at 2.1m/s of maximum velocity and 4.2 G of maximum acceleration.

CHAPTER

# 3 Proposed control solutions for Pick-&-Throw tasks with PKMs

**Contents**


3.1 Introduction . . . . . 109
3.2 A new augmented RISE feedback controller . . . . . 110
3.3 A novel robust DCAL with adaptive feedback gains . . . . . 121
3.4 A new intelligent robust control law . . . . . 128
3.5 Conclusion . . . . . 134


## 3.1 Introduction

Control design is an important research area of robotics because it determines the robotic system motion. As mentioned in Chapter 1, the control task of parallel manipulators is considered complicated and challenging due to the presence of different types of uncertainties, highly nonlinear dynamics, and, in the case of RA-PKMs, the problem of internal forces generation. To overcome these issues, advanced nonlinear robust control techniques may be a good solution to obtain improved dynamic performance in terms of high-speed motions, precision, and robustness. Aware of the importance of designing advanced control schemes for parallel robots, in this chapter, we will describe in detail three proposed control solutions for PKMs. The contribution for each adopted control strategy is addressed and explained. The proposed controllers aim to keep the joint tracking errors as small as possible under different operating conditions. Although these control schemes were designed for P&T tasks with PKMs in a selective sorting application, it is also possible to implement them in any other types of robot manipulators.

## 3.2 A new augmented RISE feedback controller

The proposed augmented RISE feedback controller is the second contribution of this thesis.

### 3.2.1 Background on RISE feedback control

Consider the dynamic equation of $p^{th}$ order MIMO uncertain nonlinear system as follows:

$$M(x,\dot{x},...,x^{(p-1)})x^{(p)} + F(x,\dot{x},...,x^{(p-1)}) = U(t) \tag{3.1}$$

where $x(t),\dot{x}(t),...,x^{(p-1)}(t) \in \mathbb{R}^n$ are the system states. $M(x,\dot{x},...,x^{(p-1)}) \in \mathbb{R}^{n\times n}$ and $F(x,\dot{x},...,x^{(p-1)}) \in \mathbb{R}^n$ being uncertain nonlinear functions, $U(t) \in \mathbb{R}^n$ is the control input vector, and $n$ is the number of actuators.

Let's define the output tracking error as follows:

$$e_1 = x_d - x \tag{3.2}$$

where $x_d(t) \in \mathbb{R}^n$ is the desired trajectory. In order to achieve an asymptotic tracking of a reference trajectory $x_d(t)$ (i.e. $e_1(t) \to 0$ as $t \to \infty$), the system and the reference trajectory should have the assumed properties below [Xian et al., 2004].

**Property 3.2.1.** *$M(.)$ is a symmetric positive-definite matrix that satisfies the following inequalities for any $\gamma \in \mathbb{R}^n$:*

$$\underline{m}\|\gamma\|^2 \le \gamma^T M(.)\gamma \le \overline{m}(x)\|\gamma\|^2$$

*where $\underline{m} \in \mathbb{R}$ is a known positive constant and $\overline{m}(x) \in \mathbb{R}$ is a positive non-decreasing function. Notice that $\|.\|$ stands for the classical Euclidean norm.*

**Property 3.2.2.** *The nonlinear functions $M(.)$ and $F(.)$ are second-order differentiable and bounded.*

**Property 3.2.3.** *The chosen desired trajectory $x_d(t) \in \mathbb{R}^n$ is continuously differentiable with respect to time until the $(p+2)^{th}$ derivative.*

Lets us now consider the following auxiliary error signals before introducing the control law:

$$\begin{aligned}
e_2(t) &= \dot{e}_1(t) + e_1(t) \\
e_3(t) &= \dot{e}_2(t) + e_2(t) + e_1(t) \\
e_4(t) &= \dot{e}_3(t) + e_3(t) + e_2(t) \\
&\vdots \\
e_p(t) &= \dot{e}_{p-1}(t) + e_{p-1}(t) + e_{p-2}(t)
\end{aligned} \tag{3.3}$$

Based on the stability analysis developed in [Xian et al., 2004], RISE feedback control law that can achieve the control objective is expressed as follows:

$$U(t) = (K_s + I)e_p(t) - (K_s + I)e_p(t_0) + \int_{t_0}^{t} [(K_s + I)\Lambda e_p(\sigma) + \beta sgn(e_p(\sigma))]d\sigma \tag{3.4}$$

where $K_s$, $\Lambda$ and $\beta \in \mathbb{R}^{n\times n}$ are diagonal positive-definite control gain matrices, $I \in \mathbb{R}^{n\times n}$ is the identity matrix, $t_0$ is the initial time and $sgn(.)$ represents the standard signum function. The integral of signum constitutes the robustness term of the RISE control law thanks to which smooth bounded disturbances can be held. It is worth to note that the second term of the R.H.S of (3.4) (i.e., $(K_s + I)e_p(t_0)$) is introduced to guarantee a zero control input at time $t = t_0$ (i.e., $U(t_0) = 0$).

### 3.2.2 RISE control of parallel manipulators

For a PKM with m-DOF and n actuators, equation (3.1) may be expressed as:

$$M(q)\ddot{q} + F(q,\dot{q}) = \Gamma(t) \tag{3.5}$$

where $F(q,\dot{q}) = C(q,\dot{q})\dot{q} + G(q) + f(q,\dot{q}) + \Gamma_d$, being $f(q,\dot{q}) \in \mathbb{R}^n$ the vector containing the friction effects, and $\Gamma_d \in \mathbb{R}^n$ a general nonlinear bounded vector of disturbances (i.e. external disturbances, interaction with the environment, etc). The nonlinear dynamics (3.5) is clearly a $2^{nd}$ order nonlinear system with n-dimensional input vector $\Gamma(t)$. The joint tracking error $e_1(t) \in \mathbb{R}^n$ is then defined as the difference between the desired joint trajectories $q_d(t)$ and the measured ones $q(t)$:

$$e_1(t) = q_d(t) - q(t) \tag{3.6}$$

To develop the closed-loop error system equation, the auxiliary tracking errors $e_2$ and $r \in \mathbb{R}^n$ should be defined as follows:

$$e_2 = \dot{e}_1 + \Lambda_1 e_1 \tag{3.7}$$

$$r = \dot{e}_2 + \Lambda_2 e_2 \tag{3.8}$$

where $\Lambda_1$, $\Lambda_2 \in \mathbb{R}^{n\times n}$ are positive-definite, diagonal gain matrices added to increase the flexibility of tuning.

At first, one needs to differentiate (3.8) with respect to time, then multiply the right and left sides of the resulting equation by the nonlinear matrix $M(q)$. After that, using the dynamic equation of the system (3.5), we obtain the following equation:

$$M(q)\dot{r} = M(q)(\ddot{q}_d + \Lambda_1\ddot{e}_1 + \Lambda_2\dot{e}_2) + \dot{M}(q)\ddot{q} + \dot{F}(q,\dot{q}) - \dot{\Gamma}_{RISE} \tag{3.9}$$

in which, $\Gamma_{RISE}(t) = \Gamma(t)$. The RISE control law (3.4) for systems whose dynamics are governed by (3.5) is expressed as follows:

$$\Gamma_{RISE}(t) = (K_s + I)e_2(t) - (K_s + I)e_2(t_0) + \int_{t_0}^{t} [(K_s + I)\Lambda_2 e_2(\sigma) + \beta sgn(e_2(\sigma))]d\sigma. \quad (3.10)$$

By adding and subtracting $\frac{1}{2}\dot{M}(q)r$ and $e_2$ to the R.H.S of the previous equation (3.9), it can be re-expressed as below:

$$M(q)\dot{r} = -\frac{1}{2}\dot{M}(q)r - e_2 - \dot{\Gamma}_{RISE} + W(q, \dot{q}, \ddot{q}, t) \quad (3.11)$$

where $W(q, \dot{q}, \ddot{q}, t)$ is a nonlinear auxiliary function defined as follows:

$$W(.) \equiv W(q, \dot{q}, \ddot{q}, t) = M(q)(\ddot{q}_d + \Lambda_1 \ddot{e}_1 + \Lambda_2 \dot{e}_2) + \dot{M}(q)\ddot{q} + \dot{F}(q, \dot{q}) + \frac{1}{2}\dot{M}(q)r + e_2 \quad (3.12)$$

After substituting the first time derivative of the RISE controller (3.10) into (3.11), one can obtain the following closed-loop error system equation:

$$M(q)\dot{r} = -\frac{1}{2}\dot{M}(q)r - e_2 - (K_s + I)r - \beta sgn(e_2) + W(.) \quad (3.13)$$

Let's now introduce the auxiliary function, $W_d(t) = W_d(q_d, \dot{q}_d, \ddot{q}_d, t)$, defined as follows:

$$W_d(t) = M(q_d)\ddot{q}_d + \dot{M}(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\ddot{q}_d + \dot{C}(q_d, \dot{q}_d)\dot{q}_d + \dot{G}(q_d) + \dot{f}(q_d, \dot{q}_d) \quad (3.14)$$

By adding and subtracting $W_d(q_d, \dot{q}_d, \ddot{q}_d, t)$ to the R.H.S of (3.13), one can obtain:

$$M(q)\dot{r} = -\frac{1}{2}\dot{M}(q)r - e_2 - (K_s + I)r - \beta sgn(e_2) + \tilde{W} + W_d \quad (3.15)$$

where $\tilde{W}(q, \dot{q}, \ddot{q}, t) = W(q, \dot{q}, \ddot{q}, t) - W_d(q_d, \dot{q}_d, \ddot{q}_d, t)$. Thanks to above properties 3.2.1, 3.2.2 and 3.2.3, one can deduce that functions $W_d$ and $\dot{W}_d \in \mathscr{L}_\infty$ (i.e. exist and bounded).

Since $W(.)$ is continuous, one can show that $\tilde{W}(.)$ can be upper bounded as follows:

$$\|\tilde{W}\| \leq \rho(\|z\|)\|z\| \quad (3.16)$$

where $z(t) = [e_1 \ e_2 \ r]^T$ and $\rho(.)$ is a globally invertible nondecreasing function. For the proof of (3.16), the reader can refer to Lemma1 in the appendix of [Patre et al., 2008].

Following the same analysis of [Xian et al., 2004], the stability of the RISE feedback law can be proved, meaning that the tracking error $e_1(t)$ and its derivatives are bounded and converge to zero as time goes to infinity

$$e_1^{(i)}(t) \to 0 \quad as \quad t \to \infty, \quad for \quad i = 0,1,2 \tag{3.17}$$

provided that the control design gains $K_s$ is chosen large enough relative to the initial conditions of the system, $\Lambda_1$, $\Lambda_2 > 1/2$, and $\beta$ satisfies the following condition:

$$\beta > \|W_d(t)\|_{\mathscr{L}_\infty} + \frac{1}{\Lambda_2}\|\dot{W}_d(t)\|_{\mathscr{L}_\infty} \tag{3.18}$$

where $\|.\|_{\mathscr{L}_\infty}$ is the $\mathscr{L}_\infty$ norm.

### 3.2.3 Applications of RISE feedback control law

RISE is a robust nonlinear controller dedicated to the control of high-order MIMO nonlinear systems. It is a full-state feedback non-model-based control scheme. RISE was developed in [Xian et al., 2004] to ensure a semi-global asymptotic tracking of uncertain nonlinear systems under some assumptions on the controlled system and the reference trajectory. It is characterized by a unique feature which is the integral of the sign of the error, assuring its continuity as well as the rejection of external disturbances.

RISE control strategies known by their robustness and disturbance rejection ability were successfully implemented in various real-time applications such as direct energy platform (jitter) [Feemster, 2014], nonlinear teleoperation system [Kawai and Namerikawa, 2013], hard disk drive [Taktak-Meziou et al., 2014], DC motor drive system [Yao et al., 2014]. Moreover, the high efficiency of RISE control schemes has been proved experimentally on different robotic applications such as Autonomous Underwater Vehicle (AUV) [Fischer et al., 2011], exoskeleton devices [Sherwani et al., 2020], rigid parallel manipulator [Escorcia-Hernández et al., 2020b]. Recently, in [Hassan et al., 2020], the RISE feedback strategy has been applied, for the first time, on a cable-driven parallel robot (CDPR), showing a high positioning accuracy, despite the significant uncertainties inherent to such a system. The RISE-based controllers show in the many researches work a high performance compared to standard RISE control law. For instance, in [Bennehar et al., 2018], a RISE-based adaptive control has been proposed as a solution for the control problem of PKMs. It consists in adding model-based adaptive feedforward term to the control loop in order to compensate for parameter variations. Another interesting extension of RISE control was introduced 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. In [Escorcia-Hernández et al., 2020a], a RISE feedback controller has been extended by an adaptive feedforward compensation term based on B-Spline Neural Networks (BSNNs) to improve tracking performance of PKMs.

### 3.2.4 Proposed augmented RISE feedback control

#### 3.2.4.1 Motivation

Parallel manipulators are well known for their unfavorable nonlinearities, abundant modeling uncertainties, external disturbances, and parameter variations (i.e., payload), especially in high-speed industrial applications. In order to achieve better output trajectory tracking despite all of the above challenges, a controller must be robust enough to counteract external disturbances as well as rich in dynamic model knowledge to compensate for system uncertainties and nonlinearities.

RISE and RISE-based control strategies show, in different applications, significant performance and robustness against disturbances and uncertainties. Therefore, the improvement of this controller is of great interest. The RISE feedback control law is a non-model based controller that depends only on system states, which may lead to low performance in the presence of large uncertainties and hard nonlinearities. Enriching the control loop of the original RISE controller by a compensation term based on the dynamic model and the system errors has the potential to improve the performance by accommodating more nonlinearities and variations in the dynamic parameters.

In most industrial applications, non-model based control strategies are applied due to their simplicity and ease of implementation. However, as mentioned above, PKMs are often subject to dynamic nonlinearities, uncertainties, parameter variations, external disturbances, etc. Accordingly, non-model-based controllers may lead to poor performance and even instability when operating at critical conditions (e.g. high speed applications, payload changes). Enriching a controller with knowledge on the manipulator dynamics can compensate for nonlinearities and parameter uncertainties, especially for high-order nonlinear systems [Kelly et al., 2006; Ren et al., 2007]. Several model-based controllers have already been reported in section 1.8.2, highlighting their improved overall tracking performance compared to non-model-based controllers.

Motivated by the advantages of enriching the controller with a dynamic compensation term and the significant performance of the RISE feedback law for different applications, the revision of the original RISE feedback law is carried out by augmenting its control loop with a nonlinear dynamic compensation term. This term is computed based on a combination of the dynamic parameters of the controlled system, which is a PKM in our study, its state errors, and the measured and desired trajectories. In addition, the resulting controller is extended by an auxiliary nonlinear feedback term to account for the errors that occur by replacing the actual trajectories (specifically the actual velocities and accelerations), in the dynamic compensation term, with the desired trajectories. The proposed control contribution can compensate for PKMs high nonlinearities as well as improve the robustness of the standard RISE controller.

#### 3.2.4.2 Control design

In what follows, we recall the general overview of PD control plus compensation. Then, the proposed control solution is introduced.

**General Overview on PD Control plus Compensation**

The PD plus compensation control law, called $PD^+$, is one of the model-based controllers that is popular in academia. It is a non-adaptive version of the first adaptive controller proposed in 1987 by Slotine and Li, and it is referred to by the names of its creators: "Slotine and Li controller" [Slotine and Li, 1987; Slotine and Weiping, 1988]. From a structural point of view, this controller is composed of a PD feedback term plus a compensation term based on the full knowledge of the dynamic model combined with the system state errors. Let us first consider the classical PD control law as follows:

$$\Gamma_{PD} = K_p e(t) + K_d \dot{e}(t) \tag{3.19}$$

where $K_p$ and $K_d \in \mathbb{R}^+$ are constant feedback gains that are tuned to ensure the stability of the system, and $e(t) = q_d - q$ is the tracking error. Enriching this controller explicitly with the dynamic model of the nonlinear controlled system allows to compensate for system nonlinearities, and thereby, enhancing the trajectory tracking accuracy. The following equation describes the control law of PD plus compensation:

$$\Gamma_{PD^+} = M(q)(\ddot{q}_d + \alpha\dot{e}(t)) + C(q,\dot{q})(\dot{q}_d + \alpha e(t)) + G(q) + K_p e(t) + K_d \dot{e}(t) \tag{3.20}$$

where $\alpha \in \mathbb{R}^+$ is defined as follows:

$$\alpha = K_v^{-1} K_p \tag{3.21}$$

**Proposed Extended RISE Control plus Compensation**

RISE is a non-model based controller consisting of two main parts: A linear state feedback term similar to a PI controller, depending on the combined tracking error, and a nonlinear robustness term based on the integral of the sign of the combined error. It does not take advantage of knowledge of the manipulator dynamic model in its control loop. Thus, the dynamic parameter uncertainties and system nonlinearities are not well compensated for by the standard RISE feedback law, especially under critical operating conditions. These issues may lead to high-gain or high-frequency feedback and poor performance in the case of high nonlinearities or in presence of large disturbances. Therefore, to improve the performance of the original RISE controller, we propose to enrich its control structure by adding a full dynamic compensation term. As it can be seen from the equation of PD control plus compensation (3.20), the compensation term consists of the system dynamic parameters computed online using the measured trajectories and then

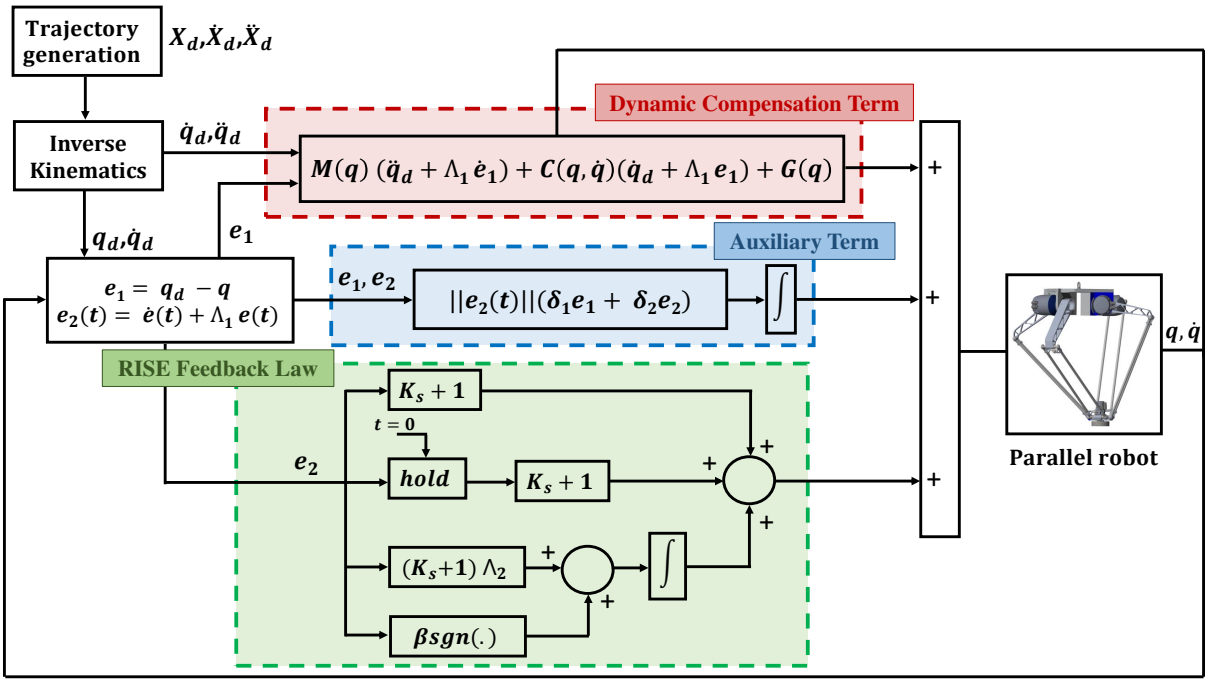



Figure 3.1 – Block diagram of the proposed extended RISE control plus compensation.

multiplied by the sum of the desired trajectories with the state errors. To overcome the error resulting from using the desired trajectory signals, RISE control plus compensation is revised by adding a nonlinear auxiliary term depending on the tracking error $e_1$ and the combined error $e_2$. The idea of adding this auxiliary term was inspired by the Desired Compensation Adaptive law (DCAL) proposed in [Sadegh and Horowitz, 1990]. In the latter, an additional nonlinear function was added to the DCAL control law to compensate for the error emerging from using desired trajectories instead of the measured ones in the adaptive feedforward term. The proposed extended RISE control plus compensation, called ERISE$^+$, is expressed as follows:

$$\begin{aligned}\Gamma_{\mathrm{ERISE}^+} = {} & M(q)(\ddot{q}_d + \Lambda_1 \dot{e}_1(t)) + C(q,\dot{q})(\dot{q}_d + \Lambda_1 e_1(t)) + G(q) + (K_s + I)e_2(t) \\ & - (K_s(t_0) + I)e_2(t_0) + \int_{t_0}^{t} \|e_2(\sigma)\| [\delta_1 e_1(\sigma) + \delta_2 e_2(\sigma)]\, d\sigma \\ & + \int_{t_0}^{t} [(K_{s0} + I)\Lambda_2 e_2(\sigma) + \beta sgn(e_2(\sigma))]\, d\sigma\end{aligned} \tag{3.22}$$

Figure 3.2 illustrates the block diagram of the proposed ERISE$^+$ control scheme for parallel manipulators.

#### 3.2.4.3 Closed-loop error dynamics

Following the same previous procedure: differentiating the filtered error $r(t)$, multiplying both sides by the matrix $M(q)$, substituting the dynamic model of the system (3.5), getting use of the derivative of the proposed controller leads to the following equation:

$$\begin{aligned} M(q)\dot{r} &= M(q)(\ddot{q}_d + \Lambda_1\ddot{e}_1 + \Lambda_2\dot{e}_2) + \dot{M}(q)\ddot{q} + \dot{C}(q,\dot{q})\dot{q} + C(q,\dot{q})\ddot{q} + \dot{G}(q) \\ &+ \dot{f}(q,\dot{q}) + \dot{\Gamma}_d(t) - \dot{M}(q)(\ddot{q}_d + \Lambda_1\dot{e}_1) - M(q)(\ddot{q}_d + \Lambda_1\ddot{e}_1) - \dot{C}(q,\dot{q})(\dot{q}_d + \Lambda_1 e_1) \\ &- C(q,\dot{q})(\ddot{q}_d + \Lambda_1\dot{e}_1) - \dot{G}(q) - (K_s + I)r - \beta\ sgn(e_2) - \|e_2\|(\delta_1 e_1 + \delta_2 e_2) \end{aligned} \tag{3.23}$$

After arranging the elements of (3.23) , we obtain the simplified equation as follows:

$$\begin{aligned} M(q)\dot{r} &= M(q)(\Lambda_2\dot{e}_2) - \dot{M}(q)(\ddot{q}_d - \ddot{q} + \Lambda_1\dot{e}_1) - \dot{C}(q,\dot{q})(\dot{q}_d - \dot{q} + \Lambda_1 e_1) + \dot{f}(q,\dot{q}) + \dot{\Gamma}_d(t) \\ &- C(q,\dot{q})(\ddot{q}_d - \ddot{q} + \Lambda_1\dot{e}_1) - (K_s + I)r - \beta\ sgn(e_2) - \|e_2\|(\delta_1 e_1 + \delta_2 e_2) \end{aligned} \tag{3.24}$$

Using $\dot{e}_1 = \dot{q}_d - \dot{q}$ and $\ddot{e}_1 = \ddot{q}_d - \ddot{q}$, adding and subtracting the two terms $\frac{1}{2}\dot{M}(.)r$ and $e_2$ to the R.H.S of (3.24), the closed-loop error of the proposed controller is arranged as:

$$\begin{aligned} M(q)\dot{r} &= -\frac{1}{2}\dot{M}(q)r - e_2 + M(q)(\Lambda_2\dot{e}_2) - (\ddot{e}_1 + \Lambda_1\dot{e}_1)(\dot{M}(q) + C(q,\dot{q})) + \dot{f}(q,\dot{q}) + \dot{\Gamma}_d(t) \\ &- \dot{C}(q,\dot{q})(\dot{e}_1 + \Lambda_1 e_1) - (K_s + I)r - \beta\ sgn(e_2) - \|e_2\|(\delta_1 e_1 + \delta_2 e_2) + \frac{1}{2}\dot{M}(q)r + e_2 \end{aligned} \tag{3.25}$$

After adding and subtracting the term $\frac{3}{2}\dot{M}(q)\dot{e}_2$ to the R.H.S of (3.25), and getting use the skew-symmetric property of $\frac{1}{2}\dot{M}(q) - C(q,\dot{q})$, (3.25) can be rewritten as follows

$$\begin{aligned} M(q)\dot{r} &= -\frac{1}{2}\dot{M}(q)r - e_2 + M(q)(\Lambda_2\dot{e}_2) - \frac{3}{2}\dot{M}(q)\dot{e}_2 - \dot{C}(q,\dot{q})(e_2) + \dot{f}(q,\dot{q}) + \dot{\Gamma}_d(t) \\ &- (K_s + I)r - \beta\ sgn(e_2) - \|e_2\|(\delta_1 e_1 + \delta_2 e_2) + \frac{1}{2}\dot{M}(q)r + e_2 \end{aligned} \tag{3.26}$$

The new auxiliary function is defined as below:

$$W(.) = \Lambda_2 M(q)\dot{e}_2 + \dot{f}(q,\dot{q}) + \dot{\Gamma}_d(t) + \frac{1}{2}\dot{M}(q)r + e_2 \tag{3.27}$$

Another auxiliary function is considered:

$$W_d(.) = \dot{f}(q_d, \dot{q}_d) \tag{3.28}$$

Let's consider the additional error resulting from using the desired joint accelerations and velocities in the dynamic compensation term as follows:

$$h(.) = -\frac{3}{2}\dot{M}(q)\dot{e}_2 - \dot{C}(q,\dot{q})(e_2) \tag{3.29}$$

Using the definitions of errors, $e_2$ and $r$, in (3.7) and (3.8), respectively, $h(.)$ can be rewritten as follows

$$h(.) = -\frac{3}{2}\dot{M}(q)(r - \Lambda_2 e_2) - \dot{C}(q, \dot{q}_d - e_2 + \Lambda_1 e_1)(e_2) \tag{3.30}$$

Adding and subtracting $W_d(.)$ to the closed-loop error dynamics (3.26), leads to:

$$M(q)\dot{r} = -\frac{1}{2}\dot{M}(q)r - e_2 - (K_s + I)r - \beta\ sgn(e_2) - \|e_2\|(\delta_1 e_1 + \delta_2 e_2) + h(.) + \tilde{W}(.) + W_d(.) \tag{3.31}$$

with $\tilde{W}(.) = W(.) - W_d(.)$

#### 3.2.4.4 Stability analysis

**Theorem 3.2.1.** *The proposed control scheme (3.22) applied to a PKM whose dynamics model is denoted by (3.5) ensures that the joint position tracking error $e_1(t)$ and its derivatives are bounded and converge asymptotically to zero with time going to infinity provided that the control design gains $\Lambda_1 > \frac{1}{2}$, $\Lambda_2$, $\delta_1$, $\delta_2$, and the feedback gain $K_s$ are chosen large enough and $\beta > \|W_d(t)\|_{\mathscr{L}_\infty} + \frac{1}{\Lambda_2}\|\dot{W}_d(t)\|_{\mathscr{L}_\infty}$.*

*Proof.* A function $L(t) \in \mathbb{R}$ defined as follows is considered [Xian et al., 2004]:

$$L(t) = r(W_d(t) - \beta\ sgn(e_2)) \tag{3.32}$$

With the use of Lemma 1 in [Xian et al., 2004], one can conclude that if $\beta$ is chosen satisfying the following condition:

$$\beta > \|W_d(t)\|_{\mathscr{L}_\infty} + \frac{1}{\Lambda_2}\|\dot{W}_d(t)\|_{\mathscr{L}_\infty} \tag{3.33}$$

then the following inequality holds:

$$\int_{t_0}^{t} L(\tau)d\tau \leq \beta|e_2(t_0)| - e_2(t_0)W_d(t_0) \tag{3.34}$$

Then, an additional function $P(t) \in \mathbb{R}$ needs to be defined as follows:

$$P(t) = \beta|e_2(t_0)| - e_2(t_0)W_d(t_0) - \int_{t_0}^{t} L(\tau)d\tau \tag{3.35}$$

knowing that $P(t) \geq 0$, $\forall t \geq 0$ is ensured from (3.33) and (3.34).

Now, let us define the following Lyapunov candidate function $V(y,t) : \mathbb{R}^{3n} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ being a continuously differentiable positive-definite function as [Xian et al., 2004]:

$$V(y,t) = e_1^T e_1 + \frac{1}{2}e_2^T e_2 + \frac{1}{2}r^T M(q)r + P \tag{3.36}$$

where $y = [z^T \sqrt{P}]^T$ and $z(t)$ is defined previously. Thanks to the property of the mass matrix $M(q)$ mentioned above and its bounds, the considered function $V(y,t)$ is bounded as follows:

$$\zeta_1 \|y\|^2 \leq V(y,t) \leq \zeta_2(\|y\|)\|y\|^2 \tag{3.37}$$

with $\zeta_1 = \frac{1}{2}\min\{1, \underline{m}\}$ and $\zeta_2(\|y\|) = \max\{\frac{1}{2}\overline{m}(\|y\|), 1\}$. Differentiating the Lyapunov candidate (3.36) with respect to time and using the equations (3.31), (3.32) and (3.35), the following equation holds:

$$\dot{V}(y,t) = 2e_1^T e_2 - 2\Lambda_1 e_1^T e_1 - \Lambda_2 e_2^T e_2 - (K_s + I)r^T r + r^T \tilde{W} + r^T h(.) - r^T \|e_2\|(\delta_1 e_1 + \delta_2 e_2) \tag{3.38}$$

Following the same reasoning in [Sadegh and Horowitz, 1990], $h(.)$ can be upper-bounded as follows

$$r^T \|h(.)\| \leq \eta_1 \|r\|^2 + \eta_2 \|r\|\|e_2\| + \eta_3 \|r\|\|e_2\|^2 + \eta_4 \|r\|\|e_1\|\|e_2\| \tag{3.39}$$

Considering the fact that for any two vectors, a and b we have $a^T b \leq (\|a\|^2 + \|b\|^2)/2$, $\dot{V}(y,t)$ is upper bounded as follows:

$$\begin{aligned}\dot{V}(y,t) \leq \|e_1\|^2 + \|e_2\|^2 - 2\Lambda_1\|e_1\|^2 - \Lambda_2\|e_2\|^2 - (K_s+1)\|r\|^2 + \|r\|\rho(\|z\|)\|z\| + \eta_1\|r\|^2 \\ + \frac{\eta_2}{2}(\|r\|^2 + \|e_2\|^2) + \eta_3\|r\|\|e_2\|^2 + \eta_4\|r\|\|e_1\|\|e_2\| - \|r^T\|\|e_2\|(\delta_1\|e_1\| + \delta_2\|e_2\|)\end{aligned} \tag{3.40}$$

By rearranging the terms, the above inequality is rewritten as follows:

$$\begin{aligned}\dot{V}(y,t) \leq -\xi_1\|e_1\|^2 - \xi_2\|e_2\|^2 - \xi_3\|r\|^2 - \mu\|r\|^2 + \|r\|\rho(\|z\|)\|z\| \\ - \alpha_1\|r\|\|e_1\|\|e_2\| - \alpha_2\|r\|\|e_2\|^2\end{aligned} \tag{3.41}$$

where $\xi_1$, $\xi_2$, $\xi_3$ , $\mu$, $\alpha_1$ and $\alpha_2$ are positive constants defined as follows:

$$\xi_1 = \Lambda_1 - \frac{1}{2},\ \xi_2 = \Lambda_2 - 1 - \frac{\eta_2}{2},\ \xi_3 = 1 - \eta_1 - \frac{\eta_2}{2},\ \mu = K_s,\ \alpha_1 = \delta_1 - \eta_4,\ \alpha_2 = \delta_2 - \eta_3 \tag{3.42}$$

Regarding the above equations $\Lambda_1$, $\Lambda_2$, $\delta_1$, and $\delta_2$ should be chosen such that

$$\Lambda_1 > \frac{1}{2} \tag{3.43}$$

$$\Lambda_2 > 1 + \frac{\eta_2}{2} \tag{3.44}$$

$$\delta_1 > \eta_4 \tag{3.45}$$

$$\delta_2 > \eta_3 \tag{3.46}$$

Let us rearrange the equation (3.41) as follows:

$$\dot{V} \leq -\zeta_3 \|z\|^2 - (\mu \|r\|^2 - \|r\| \rho(\|z\|) \|z\|) \tag{3.47}$$

with $\zeta_3 = \min\{\xi_1, \xi_2, \xi_3\}$. Using the mathematical remarkable square identities, the upper-bound of $\dot{V}$ is obtained as follows:

$$\dot{V} \leq -(\zeta_3 - \frac{\rho^2(\|z\|)}{4\mu}) \|z\|^2 \triangleq -c\|z\|^2 \tag{3.48}$$

with c is a positive constant $\in \mathbb{R}^+$, the following inequality shall hold:

$$\zeta_3 > \frac{1}{4\mu} \rho^2(\|z\|) \tag{3.49}$$

Based on the inequality (3.49), we can introduce the domain $\mathscr{D}$ as follows:

$$\mathscr{D} = \{y \in \mathbb{R}^{3n} \times \mathbb{R}_{\geq 0} \mid \|y\| < \rho^{-1}(2\sqrt{\zeta_3 \mu})\} \tag{3.50}$$

Let's define the subset $\mathscr{A}$ of $\mathscr{D}$ such that the initial condition $y(0) \in \mathscr{A}$ as follows:

$$\mathscr{A} = \{y(t) \in \mathscr{D} \mid \ c\|z\|^2 < \zeta_1 (\rho^{-1}(2\sqrt{\zeta_3 \mu}))^2\} \tag{3.51}$$

According to [Escorcia-Hernandez et al., 2022], $c\|z\|^2$ is consistently continuous in $\mathscr{D}$. Thus, based on theorem 8.4 of [Khalil and Grizzle, 2002], we can conclude that

$$c\|z\|^2 \to 0 \quad as \quad t \to \infty, \quad \forall y(0) \in \mathscr{A} \tag{3.52}$$

Therefore, based on the definition of z(t), we can deduce that

$$\|e_1\| \to 0 \quad as \quad t \to \infty, \quad \forall y(0) \in \mathscr{A} \tag{3.53}$$

Hence, the stability of the proposed controller is proven.

## 3.3 A novel robust DCAL with adaptive feedback gains

The proposed robust DCAL with adaptive feedback gains is the third contribution of the thesis. Before presenting the proposed control solution, let us first recall a crucial property of the dynamics of robot manipulators. As mentioned in Chapter 1, the dynamic model of the manipulator denoted by (3.5) is characterized by its linearity with respect to dynamic parameters such as inertia and masses. Except for external disturbances, all constant parameters of the dynamic model are considered as coefficients of known functions (linear and nonlinear) of $q, \dot{q}, \ddot{q}$. The linear reformulation of the dynamics can be rewritten as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + f(q,\dot{q}) + \Gamma_d(t) = W(q,\dot{q},\ddot{q})\Phi(t) + f(q,\dot{q}) + \Gamma_d(t) \tag{3.54}$$

where $W(q,\dot{q},\ddot{q}) \in \mathbb{R}^{n\times p}$ is called the regression matrix and is formed by known nonlinear functions of $q, \dot{q}, \ddot{q}$. The vector $\Phi \in \mathbb{R}^p$ gathers the geometrical and dynamic parameters of the robot. In the sequel, a background on the standard DCAL will be provided. Afterwards, the proposed control approach will be detailed.

### 3.3.1 General overview of DCAL control strategy

DCAL is a model-based adaptive control scheme developed for the first time in [Sadegh and Horowitz, 1990] for the control of serial manipulators. DCAL control technology has been inspired from the exact compensation adaptation law (ECAL) which consists of a model-based adaptive term and a PD linear feedback loop. In the latter, the nonlinear dynamic-based compensation term and the online adaptation law are computed using the actual joint positions and velocities. Indeed, measuring the system states at each sample time is a complex and heavy computation task for the real-time applications especially for the velocities. Most of the time, the velocities cannot be measured, hence, a numerical estimation from the measured positions is required. As is well known, the measurement noise is amplified by such an estimation, yielding a deterioration of the performance of the controller. To handle this issue, DCAL has been formulated such that, both the control and adaptation laws, are evaluated using the desired quantities while still inherit the advantages of the original controller. This particular choice leads to off-line compute the compensation term. Consequently, the amount of online computations is reduced and the robustness of the controller is enhanced since the measurement noises are eliminated by using the desired states. The control law of DCAL can be split up into three main parts: (i) a model-based adaptive feedforward part, (ii) a linear feedback part, and (iii) an additional nonlinear feedback function to accommodate for the errors resulting from using the desired states instead of the measured ones. The joint-space control law of DCAL is expressed as follows [Sadegh and Horowitz, 1990]:

$$\Gamma_{DCAL} = W(q_d,\dot{q}_d,\ddot{q}_d)\hat{\Phi}(t) + \Lambda_p\, e(t) + \Lambda_v\, e_v(t) + \sigma\|e(t)\|^2\, e_v(t) \tag{3.55}$$

where $e(t) = q_d(t) - q(t)$ is the joint position tracking error, with $q_d(t) \in \mathbb{R}^n$ is the vector of desired joint positions and $q(t) \in \mathbb{R}^n$ is the vector of measured ones. $e_v(t) = \dot{e}(t) + \lambda e(t)$ is the combined position-velocity tracking error, $\lambda \in \mathbb{R}^+$ is a positive design gain. $\Lambda_p$, $\Lambda_v$ $\in \mathbb{R}^{n\times n}$ are positive-definite gain matrices, usually chosen diagonal. $W(q_d, \dot{q}_d, \ddot{q}_d) \in \mathbb{R}^{n\times p}$ is the regressor matrix function depending on desired joint positions, velocities and accelerations. $\hat{\Phi}(t) \in \mathbb{R}^{n\times p}$ is an online estimation of the unknown parameters vector $\Phi$, and $\sigma$ $\in \mathbb{R}^+$ is a positive design control parameter.
The time-evolution of the estimated parameters $\hat{\Phi}(t)$ in (3.55) is expressed by the following adaptation law:

$$\dot{\hat{\Phi}}(t) = K W^T(q_d, \dot{q}_d, \ddot{q}_d) e_v(t) \tag{3.56}$$

where $K \in \mathbb{R}^{p\times p}$ is a diagonal positive-definite adaptation gain matrix. As it can be seen, the regressor $W$ in the adaptation law (3.56) is also evaluated based on the desired trajectories instead of the measurements.

### 3.3.2 Proposed robust DCAL with adaptive feedback gains

#### 3.3.2.1 Motivation

As mentioned in Chapter 1, the objective is to use PKMs in a selective waste sorting task. Such an application is considered a difficult task for PKMs, since the manipulator has to handle different types of objects with different physical parameters, that may often be unknown or uncertain. Therefore, model-based adaptive schemes, characterized by dynamic parameter identification in an online algorithm, are the most appropriate control solutions for such applications. For instance, the aforementioned DCAL may be a good candidate, thanks to its simple structure easy to implement, its real-time estimation of the model parameters, and its robustness against measurement noise.

The stability analysis of DCAL demonstrated that it ensures the asymptotic tracking of reference trajectories as well as the boundedness of the estimated parameters [Sadegh and Horowitz, 1990]. However, it does not say much about the transient performance of the closed-loop system. In fact, during transient phase, the estimated parameters may be away from their best steady-state values or oscillating around them. This means that the inherent nonlinearities of the manipulator are not properly compensated. Consequently, the overall control scheme relies mainly on the feedback loop to achieve the control objective. The PD feedback loop of DCAL will be responsible of both tracking the reference trajectories and rejecting the time-varying nonlinearities. It is known in control theory that static feedback control algorithms can provide good performance only in nominal steady state and when no changes in operating conditions occur. Consequently, it would be interesting to revisit DCAL with a carefully designed feedback loop. It has been shown through various research works that control methods with adaptive dynamic feedback gains can

counteract external disturbances and accommodate the variations in dynamic parameters [Gholami et al., 2009; Tijjani et al., 2020; Escorcia-Hernandez et al., 2022].

With the above in mind, we propose to exploit the advantages of the real-time estimation of the model parameters provided by DCAL and the corrective action produced by an adequate adaptation law for the feedback gains. In addition, to better counteract the external disturbances, we propose to extend the resulting controller by a nonlinear sliding-based term computed from the signum of the system state errors. The addition of this robustness related term will accommodate for the lack of robustness and may improve the overall tracking performance. In a real-time implementation, the discontinuous signum function may be replaced by a continuous sigmoid function to avoid chattering. The adaptive feedback gains of the proposed control solution produce corrective actions when the tracking errors are considerably increased due to changes in the operating conditions. These gains are adjusted according to a criterion based on the values of the joint tracking errors. The proposed enhancement to DCAL should preserve all its advantages while improving the overall closed-loop behavior, with more focus on the tracking performance.

#### 3.3.2.2 Control design

Despite the efficiency of the standard DCAL, it exhibits a lack of performance due to the static linear feedback gains and the potential presence of external disturbances not compensated by the control law. To significantly improve the overall performance of such a controller, we first propose to revisit the linear feedback term by adopting an adaptive one where the gains are adjusted online according to the system state errors. Second, to further improve its robustness against disturbances, a sliding-based term depending on the combined error $e_v$ is added. The resulting expression of the proposed control law can be written as follows:

$$\begin{aligned}\Gamma_{RDCAL-AG} = W(q_d, \dot{q}_d, \ddot{q}_d)\hat{\Phi}(t) + \Lambda_p(t)\, e(t) + \Lambda_v(t)\, e_v(t) \\ + \sigma\|e(t)\|^2\, e_v(t) + \beta sgn(e_v(t))\end{aligned} \tag{3.57}$$

where $\beta \in \mathbb{R}^{n\times n}$ is a positive-definite diagonal gain matrix used to increase the controller robustness. $\Lambda_p(t)$ and $\Lambda_v(t) \in \mathbb{R}^{n\times n}$ are the adaptive gain matrices whose adaptation algorithms are inspired by the concept of the adaptive gains presented in [Plestan et al., 2010]. The adaptive rules for the gain matrices are as follows:

$$\Lambda_p(t) = \bar{\Lambda}_p|\eta_p| + \Lambda_{pm} \tag{3.58}$$

$$\Lambda_v(t) = \bar{\Lambda}_v|\eta_v| + \Lambda_{vm} \tag{3.59}$$

Where $\bar{\Lambda}_p$ and $\bar{\Lambda}_v \in \mathbb{R}^{n\times n}$ denote positive-definite constant diagonal matrices used in the adaptation process of the control feedback gains. $\Lambda_{pm}$ and $\Lambda_{vm} \in \mathbb{R}^{n\times n}$ are other positive-definite constant diagonal matrices that establish the minimum possible value for each

adaptive feedback gain. |.| is the modulus vector function used to obtain only positive gain values, $\eta_p$ and $\eta_v \in \mathbb{R}^n$ are nonlinear functions depending on the tracking error $e$ and the combined tracking error $e_v$, respectively.

$$\dot{\eta}_p = \tanh(e) - \eta_p \tag{3.60}$$

$$\dot{\eta}_v = \tanh(e_v) - \eta_v \tag{3.61}$$

These two equations (3.60) and (3.61) represent the dynamics of the adaptive gains $\Lambda_p(t)$ and $\Lambda_v(t)$, respectively. A hyperbolic tangent function is used to produce an effect similar to the signum function, but without generating prominent discontinuities for a better smoothness.

It is worth to note that when the tracking error increases, the adaptive gains of the proposed control law produce a corrective action to reduce this large tracking error. Once it decreases, the adopted strategy begins to reduce the control action and adjusts the gains to avoid oscillations and sufficiently counteract the current uncertainties and disturbances. When it comes to the estimation of unknown dynamic parameters, the same adaptation law (3.56) is adopted for the proposed controller. Therefore, this new control technique inherits the advantages of the original DCAL in terms of noise measurement reduction and low computational time. Figure 3.2 illustrates the block diagram of the proposed controller for parallel manipulators.

It is worth mentioning that, the main difference between this study and the contribution of [Bennehar et al., 2016] lies in nature of the control schemes. In [Bennehar et al., 2016] the control solution uses nonlinear time-varying gains, whose variation is based on predefined fixed nonlinear functions. However, the proposed RDCAL-AG scheme is based on adaptive feedback gains, whose variation is governed by adaptation laws. Furthermore, beyond the different kinematics as well as the target application, a sliding-based robustness term is used in the proposed RDCAL-AG to reinforce the controller robustness and thereby improve the tracking performance.

#### 3.3.2.3 Stability analysis

**Theorem 3.3.1.** *The proposed control law defined by* (3.57) *and applied to the second-order nonlinear MIMO dynamic system governed by* (3.5) *ensures that the joint position and velocity tracking errors converge asymptotically to zero as time goes to infinity, as long as the lower bounds of the feedback gains* $\Lambda_{p_m}$ *and* $\Lambda_{v_m}$ *in (3.58) and (3.59), respectively, are chosen sufficiently large.*

*Proof.* In order to study the stability analysis of the system error dynamic in closed loop with the proposed control solution, let us first rewrite the IDM of PKM (3.5) in terms of

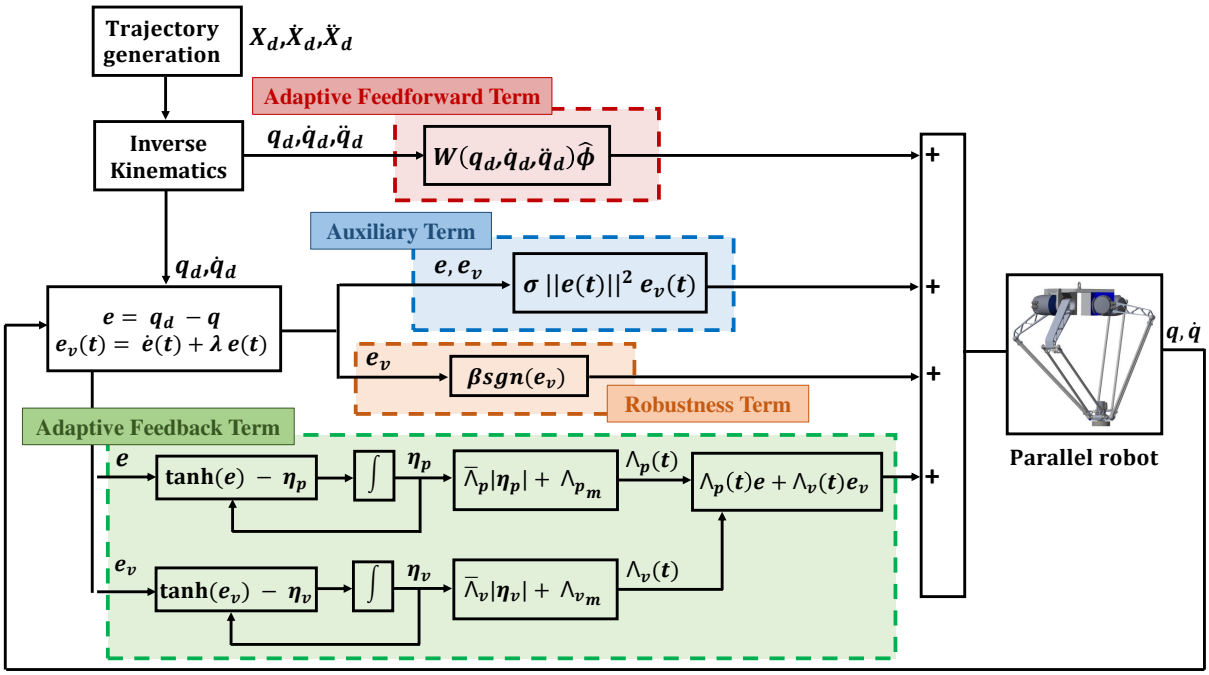



Figure 3.2 – Block diagram of the proposed RDCAL AG control scheme.

$e(t)$, $\dot{e}(t)$ and $e_v(t)$ as follows

$$M(q)\dot{e}_v = M(q)(\ddot{q}_d + \lambda\dot{e}) + C(q,\dot{q})(\dot{q}_d + \lambda e) - C(q,\dot{q})e_v + G(q) + f(q,\dot{q}) + \Gamma_d(t) - \Gamma \quad (3.62)$$

Let $h \equiv h(q,\dot{q},t) = f(q,\dot{q}) + \Gamma_d(t)$ be the general nonlinear perturbation function including friction effects. After adding and subtracting $W(q_d,\dot{q}_d,\ddot{q}_d)\Phi$ to (3.62), the following closed loop error dynamic is obtained

$$M(q)\dot{e}_v = -C(q,\dot{q})e_v + W(q_d,\dot{q}_d,\ddot{q}_d)\Phi + \tilde{W}(.)\Phi + h(q,\dot{q},t) - \Gamma \quad (3.63)$$

where $\tilde{W}(.)$ is the dynamic error resulting from using the desired joint quantities instead of the actual ones in both the compensation of the nonlinearities in the dynamic model and the adaptation law, denoted by

$$\tilde{W}(.)\Phi = M(q)(\ddot{q}_d + \lambda\dot{e}) + C(q,\dot{q})(\dot{q}_d + \lambda e) + G(q) - W(q_d,\dot{q}_d,\ddot{q}_d)\Phi \quad (3.64)$$

By substituting the proposed controller (3.57) into the system error dynamic (3.63), the resulting equation is

$$\begin{aligned} M(q)\dot{e}_v = &- C(q,\dot{q})e_v + W(q_d,\dot{q}_d,\ddot{q}_d)\tilde{\Phi} + \tilde{W}(.)\Phi + h(q,\dot{q},t) - \Lambda_p(t)e - \Lambda_v(t)e_v \\ &- \sigma\|e\|^2 e_v - \beta sgn(e_v) \end{aligned} \quad (3.65)$$

where $\tilde{\Phi} = \Phi - \hat{\Phi}$ represents the vector of the estimation error. Now, to analyse the stability of the resulting closed loop error dynamic (3.65), the following Lyapunov candidate function is considered as below [Sadegh and Horowitz, 1990]

$$V = \frac{1}{2} e_v^T M(q) e_v + \frac{1}{2} \Lambda_{p_m} e^T e + \frac{1}{2} \tilde{\Phi}^T K^{-1} \tilde{\Phi} \tag{3.66}$$

Differentiating (3.66) with respect to time leads to

$$\dot{V} = \frac{1}{2} e_v^T \dot{M}(q) e_v + e_v^T M(q) \dot{e}_v + \Lambda_{p_m} e^T \dot{e} + \tilde{\Phi}^T K^{-1} \dot{\tilde{\Phi}} \tag{3.67}$$

By considering the above adaptation law (3.56) and the property 1.7.2, expression (3.67) can be rewritten as follows

$$\dot{V} = \Lambda_{p_m} e^T \dot{e} - e_v^T \Lambda_p(t) e - e_v^T \Lambda_v(t) e_v - \sigma \|e\|^2 \|e_v\|^2 - e_v^T \beta sgn(e_v) + e_v^T h + e_v^T \tilde{W}(.)\Phi \tag{3.68}$$

According to [Sadegh and Horowitz, 1990], the upper limit of the term $\tilde{W}(.)\Phi$ is given as follows

$$\|\tilde{W}(.)\Phi\| \leq \xi_1 \|e\| + \xi_2 \|e\|^2 + \xi_r \|e_v\| + \xi_4 \|e_v\| \|e\| \tag{3.69}$$

with $\xi_1, \xi_2, \xi_3$ and $\xi_4$ are positive bounding constants $\in \mathbb{R}^+$. Moreover, assuming that the general disturbance function $h(q, \dot{q}, t)$ is globally bounded by

$$\|h\| \leq \mu \|e_v\| \tag{3.70}$$

with $\mu$ is a positive bounding constant $\in \mathbb{R}^+$. Let us now consider the lower bounds $\Lambda_{p_m}$ and $\Lambda_{v_m}$ of $\Lambda_p(t)$ and $\Lambda_v(t)$, respectively, together with the bounded expressions (3.69) and (3.70), the upper bound of the derivative of the Lyapunov function can be expressed as below

$$\begin{aligned} \dot{V} \leq -\lambda \Lambda_{p_m} \|e\|^2 - \Lambda_{v_m} \|e_v\|^2 - \sigma \|e\|^2 \|e_v\|^2 - \beta \|e_v\|^2 + \mu \|e_v\|^2 + \xi_1 \|e\| \|e_v\| + \xi_2 \|e\|^2 \|e_v\| \\ + \xi_3 \|e_v\|^2 + \xi_4 \|e_v\|^2 \|e\| \end{aligned} \tag{3.71}$$

After rearranging the terms of above equation (3.71) together with using the mathematical remarkable square identities, we obtain

$$\begin{aligned} \dot{V} \leq -\lambda \Lambda_{p_m} \|e\|^2 - \Lambda_{v_m} \|e_v\|^2 - \sigma \|e\|^2 \|e_v\|^2 - (\beta - \mu) \|e_v\|^2 + \xi_1 \|e\| \|e_v\| - \xi_2 \|e\|^2 \left[ \frac{1}{2} - \|e_v\| \right]^2 \\ - \xi_4 \|e_v\|^2 \left[ \frac{1}{2} - \|e\| \right]^2 + (\xi_2 + \xi_4) \|e\|^2 \|e_v\|^2 + \frac{\xi_2}{4} \|e\|^2 + \left( \xi_3 + \frac{\xi_4}{4} \right) \|e_v\|^2 \end{aligned} \tag{3.72}$$

To further simplify more the expression of (3.72), let us gather the common terms as follows

$$\dot{V} \leq -\left(\lambda\Lambda_{p_m} - \frac{\xi_2}{4}\right)\|e\|^2 - \left(\Lambda_{v_m} - \xi_3 - \frac{\xi_4}{4}\right)\|e_v\|^2 + \xi_1\|e\|\|e_v\| - \xi_2\|e\|^2\left[\frac{1}{2} - \|e_v\|\right]^2 - \xi_4\|e_v\|^2\left[\frac{1}{2} - \|e\|\right]^2 - (\sigma - \xi_2 - \xi_4)\|e\|^2\|e_v\|^2 - (\beta - \mu)\|e_v\|^2 \tag{3.73}$$

Regarding the expression of (3.73), $\beta$ and $\sigma$ should be chosen such that $(\beta - \mu) > 0$ and $(\sigma - \xi_2 - \xi_4) > 0$, which leads to the new upper bound of $\dot{V}$

$$\dot{V} \leq -\left(\lambda\Lambda_{p_m} - \frac{\xi_2}{4}\right)\|e\|^2 - \left(\Lambda_{v_m} - \xi_3 - \frac{\xi_4}{4}\right)\|e_v\|^2 + \xi_1\|e\|\|e_v\| \tag{3.74}$$

Then, getting use the conventional inequality for any two vectors a and b, $a^T b \leq (\|a\|^2 + \|b\|^2)/2$, the new upper bound of $\dot{V}$ can rewritten as follows

$$\dot{V} \leq -\left(\lambda\Lambda_{p_m} - \frac{\xi_2}{4} - \frac{\xi_1}{2}\right)\|e\|^2 - \left(\Lambda_{v_m} - \xi_3 - \frac{\xi_4}{4} - \frac{\xi_1}{2}\right)\|e_v\|^2 \tag{3.75}$$

Consequently, $\dot{V}$ is negative semi-definitive and $V$ is upper bounded, if the $\Lambda_{p_m}$ and $\Lambda_{v_m}$ lower limits are tuned in such a way that

$$\Lambda_{p_m} > \frac{\xi_2}{4\lambda} + \frac{\xi_1}{2\lambda} > 0 \tag{3.76}$$

$$\Lambda_{v_m} > \xi_3 + \frac{\xi_4}{4} + \frac{\xi_1}{2} > 0 \tag{3.77}$$

The expression of $V$ (3.66) show that the combined tracking error and the estimation error, $e_v$ and $\tilde{\Phi}$, respectively, are bounded. Given the fact that the inertia matrix $M(q)$ is lower bounded, it can be noticed that $V$ in (3.66) is lower bounded as well. From the resulting equation of the system error dynamic (3.65), we conclude that $\dot{e}_v$, $\ddot{q}$ and, consequently, $\ddot{V}$ are all bounded. Based on the Lyapunov-like Lemma [Khalil and Grizzle, 2002], the fact that $\dot{V}$ is uniformally continuous in time is guaranteed (i.e. $\dot{V}$ converges to zero as long as time goes to infinity), since the following three arguments are satisfied:

1. $V$ is lower bounded
2. $\dot{V}$ is negative semi-definite
3. $\ddot{V}$ is bounded

Consequently, all the closed-loop system state errors (i.e. $e_v$, $\dot{e}$ and $e$) converge very close to zero with time.

$$\begin{aligned} e_v &\to 0 \ \text{ as } \ t \to \infty \\ \dot{e} &\to 0 \ \text{ as } \ t \to \infty \\ e &\to 0 \ \text{ as } \ t \to \infty \end{aligned} \tag{3.78}$$

The estimated parameter bounds and the proof of stability are thereby established.

## 3.4 A new intelligent robust control law

The proposed intelligent robust control shceme is the fourth contribution of this thesis.

### 3.4.1 Background on Model-Free Control

For several decades, model-based controllers have been considered as good candidates for the control of uncertain nonlinear systems. They have shown their ability to compensate for the system nonlinearities by introducing an apriori knowledge on the dynamic model in the control design. However, it is well known that the development of an accurate dynamic model for a complex nonlinear system is almost impossible. On one hand, unmodeled phenomena cannot be considered into such a model, and on the other hand, the system parameters are often subject to variations and uncertainties. Therefore, classical model-based controllers may lead to poor control performance of uncertain nonlinear systems, especially when changes in operating conditions occur or when the system parameters used in the control design do not match the actual parameters. To address this problem of parameter variations and uncertainties, model-based adaptive controllers have been proposed in the literature. As mentioned in Chapter 1, these control schemes include an adaptive feedforward term responsible for the online estimation of the unknown, uncertain or time-varying dynamic parameters. Despite the simplicity of the principle of online parameters estimation, its real-time implementation requires a considerable number of calculations, which leads to a significant computing time. To deal with these issues, Fliess and Join proposed a Model-Free Control (MFC) strategy to control uncertain high-order nonlinear MIMO systems in an intelligent and simple manner [Fliess et al., 2006; Fliess, 2009]. Typically, model-based control approaches require an accurate knowledge of the system dynamic model, however, MFC can compensate for structured and unstructured phenomena without incorporating any knowledge about the system dynamics, in the control design. The MFC technique, also known as "intelligent control," incorporates, into the control loop, a numerical model that is computed at each sampling time based on few online parameters. This numerical model is only valid for a small time interval. In 2008, MFC was implemented for the first time in a real-time numerical experiment [Fliess and Join, 2008].

The basic principle of MFC is that the input-output behavior of a high-order nonlinear system, can be represented by an ultra-local model, continuously restructured, as follows:

$$y^{(\nu)} = \delta + \alpha U \tag{3.79}$$

where $\alpha \in \mathbb{R}^{n \times n}$ is positive-definite diagonal gain matrix, chosen by the designer to ensure some control performance. $y \in \mathbb{R}^n$ is the output vector, $U \in \mathbb{R}^n$ is the control input vector and $\delta \in \mathbb{R}^n$ is a vector gathering the modeled and unmodeled system dynamics. $\nu \in \mathbb{N}$ denotes the order of the anticipated model.

$\delta$ can be estimated at each sampling time from the measured output $y$, and the known input $U$, as follows [Fliess and Join, 2013]:

$$[\delta(k)]_e = [y^{(\nu)}(k)]_e - \alpha U(k-1) \tag{3.80}$$

where $[\delta(k)]_e$ is the estimation at the sample $k$ of the function $\delta$ (i.e. $t = kT_s$ is the sampling time with $k = 0, 1, ...$ and $T_s$ is the sampling period), $[y^{(\nu)}(k)]_e$ denotes the estimation of the $\nu$-derivative of $y$ at time $k$, and $U(k-1)$ is the control input at the previous sample time. It is worth to note that this estimation is only valid for a short interval of time, and should be updated frequently and continuously to maintain the system stability with a good tracking performance.

In general, the model-free control input $U$ is computed at each iteration, based on the following expression:

$$U(k) = -\frac{[\delta(k)]_e}{\alpha} + \frac{y_d^{(\nu)}(k) + U_c(k)}{\alpha} \tag{3.81}$$

where $y_d^{(\nu)}$ is the $\nu^{th}$ derivative of the desired trajectory, and $U_c$ is the feedback control law that should be selected in a way to guarantee asymptotic convergence of the output signal to the desired trajectory. If the feedback control term is a conventional PID controller, then the model-free control is called intelligent PID (iPID).

The designer must choose the value of the order $\nu$, carefully, taking into account the order of the controlled system and the structure of the feedback control input. Otherwise, the stability of the system may be deteriorated. In general, $\nu$ can be chosen to be 1 or 2. For instance, intelligent PID is designed with $\nu = 2$, while intelligent PI (iPI) is used with $\nu = 1$.

In practice, the measured output is often subject to noise, which may be amplified by the numerical derivative. Regarding this issue, some methods have been proposed in the literature to compute the $\nu^{th}$ derivative of noisy signals using an algebraic approach. The differential algebraic operations are composed of iterated integrals of the noisy signal taking the form of classical finite impulse response (FIR) digital filters [Fliess and Join, 2013; Mboup et al., 2009; Gédouin et al., 2011]; while other methods have approximated the value of $\delta$ by exploiting an extended state observer [Wang et al., 2018; Zhang et al., 2020b].

This control method is characterized by its simplicity and ease of implementation. Moreover, its design parameters can be adjusted in a straightforward manner.

### 3.4.2 Applications of MFC scheme

Owing to their simple structure and robustness, model-free controllers have been successfully implemented in various application fields showing a good tracking performance. They have been applied to hydroelectric power plants [ROBERT and FLIESS, 2010], dc/dc converters [Michel et al., 2010], active shape memory alloy spring [Gédouin et al., 2009],

[Gédouin et al., 2011], underactuated mechanical systems such as the inverted pendulum flywheel [Andary et al., 2012], active magnetic bearing [De Miras et al., 2013] and quadrotor vehicle [Al Younes et al., 2014]. Furthermore, a model-free fractional order sliding mode control (MFFOSTM) based on an extended state observer (ESO) has been proposed and applied on the active suspension systems of a quarter car [Wang et al., 2018]. Recently, predictive current control (PCC) has been combined with an ultra-local model to control a dual-fed induction generator (DFIG) [Zhang et al., 2020a]. In [Zhang et al., 2020b], the model-free predictive current control (MFPCC) scheme was implemented on a PMSM drive system based on an extended state observer (ESO).

### 3.4.3 Proposed intelligent robust control

#### 3.4.3.1 Control design

As previously presented, RISE is a non-model-based controller. This means that the system nonlinearities and uncertainties are not well compensated by such a controller, resulting in a degraded tracking performance in critical situations (e.g. high-speed movements, and in the presence of large disturbances). Moreover, the extension of this controller with a classical model-based feedforward term is not an effective idea since it is almost impossible to obtain an accurate dynamic model of a complex nonlinear system. This issue becomes considerable in industrial P&P or P&T [Raptopoulos et al., 2020] applications, where the robot should handle different types of objects with unknown/uncertain dynamic parameters. The aforementioned MFC scheme has the ability to compensate for parameter variations, modeled and unmodeled dynamics without considering any knowledge of the system dynamics in the control design. Accordingly, we propose to revise the standard RISE feedback control law by redesigning its control equation based on an ultra-local model inspired from MFC scheme. This revision can significantly improve the tracking performance of RISE as well as its robustness towards external disturbances, while preserving the simplicity of its scheme. This is highly important because to perform complex tasks, it is necessary to ensure precise trajectory tracking despite the problems caused by parameter variations and abrupt speed changes. The resulting controller, called Intelligent Robust Control (IRC), is continuous, robust and easy to implement. The proposed IRC strategy for second-order MIMO systems such as PKMs can be expressed as follows:

$$\Gamma_{IRC} = -\frac{[\delta]_e}{\alpha} + \frac{\ddot{q}_d + \Gamma_{RISE}}{\alpha} \tag{3.82}$$

where $\Gamma_{RISE}$ is the RISE feedback law presented in (3.10). As reported in [Gédouin et al., 2011], if the order of the controlled system is well known, then the $\nu$ order of the numerical model as given in equation (3.81) will be equal to that of the system. Therefore, in our case, for a system of second order as PKMs, $\nu$ is equal to 2. The block diagram of the proposed IRC scheme for parallel manipulators is illustrated in Figure 3.3.

#### 3.4.3.2 Closed-loop error dynamics

In the sequel, the closed-loop error equation of the system represented by (3.5) is derived in order to analyse the stability of the proposed IRC scheme. Following the same procedure detailed above: Firstly, differentiating the auxiliary error $r(t)$, then multiplying both sides by the matrix $M(q)$, substituting the system dynamics (3.5) and getting use of the time derivative of the proposed IRC leads to:

$$\begin{aligned} M(q)\dot{r} = -\frac{1}{2}\dot{M}(q)r - e_2 + \frac{[\dot{\delta}]_e}{\alpha} - \frac{\ddot{q}_d + \dot{\Gamma}_{RISE}}{\alpha} + M(q)(\ddot{q}_d + \Lambda_1\ddot{e}_1 + \Lambda_2\dot{e}_2) \\ + \dot{M}(q)(\dot{q} + \frac{1}{2}r) + \dot{F}(q,\dot{q}) + e_2 \end{aligned} \tag{3.83}$$

By adding and subtracting the term $\frac{\dot{\delta}}{\alpha}$ to the R.H.S of the above-obtained equation (3.83), it can be rewritten as follows:

$$\begin{aligned} M(q)\dot{r} = -\frac{1}{2}\dot{M}(q)r - e_2 - \frac{\dot{\tilde{\delta}}}{\alpha} + \frac{\dot{\delta}}{\alpha} - \frac{\ddot{q}_d + \dot{\Gamma}_{RISE}}{\alpha} + M(q)(\ddot{q}_d + \Lambda_1\ddot{e}_1 + \Lambda_2\dot{e}_2) \\ + \dot{M}(q)(\dot{q} + \frac{1}{2}r) + \dot{F}(q,\dot{q}) + e_2 \end{aligned} \tag{3.84}$$

with $\tilde{\delta} = \delta - [\delta]_e$ is the estimation error of the nonlinear function $\delta(t)$. This nonlinear term $\delta$, its estimated function $[\delta]_e$ and their first two time derivatives are bounded (i.e., $\delta$, $\dot{\delta}$, $\ddot{\delta}$ $\in \mathscr{L}_\infty$ as well as $[\delta]_e$, $[\dot{\delta}]_e$, $[\ddot{\delta}]_e \in \mathscr{L}_\infty$). Accordingly, the estimation error $\tilde{\delta}$ and its first two time derivatives, $\dot{\tilde{\delta}}$ and $\ddot{\tilde{\delta}}$, are bounded.

The new auxiliary function is introduced as follows:

$$W(q,\dot{q},\ddot{q},t) = \frac{\dot{\delta}}{\alpha} - \frac{\ddot{q}_d}{\alpha} + M(q)(\ddot{q}_d + \Lambda_1\ddot{e}_1 + \Lambda_2\dot{e}_2) + \dot{M}(q)(\dot{q} + \frac{1}{2}r) + \dot{F}(q,\dot{q}) + e_2 \tag{3.85}$$

Another auxiliary function, $W_d(.) = W_d(q_d,\dot{q}_d,\ddot{q}_d,t)$ is considered as below:

$$W_d(.) = -\frac{\ddot{q}_d}{\alpha} + M(q_d)\ddot{q}_d + \dot{M}(q_d)\dot{q}_d + C(q_d,\dot{q}_d)\dot{q}_d + \dot{C}(q_d,\dot{q}_d)\dot{q}_d + \dot{G}(q_d) + \dot{f}(q_d,\dot{q}_d) \tag{3.86}$$

Adding and subtracting $W_d(.)$ to the closed-loop error equation (3.84), leads to:

$$M(q)\dot{r} = -\frac{1}{2}\dot{M}(q)r - e_2 - \frac{\dot{\tilde{\delta}}}{\alpha} - \frac{\dot{\Gamma}_{RISE}}{\alpha} + \tilde{W}(.) + W_d(.) \tag{3.87}$$

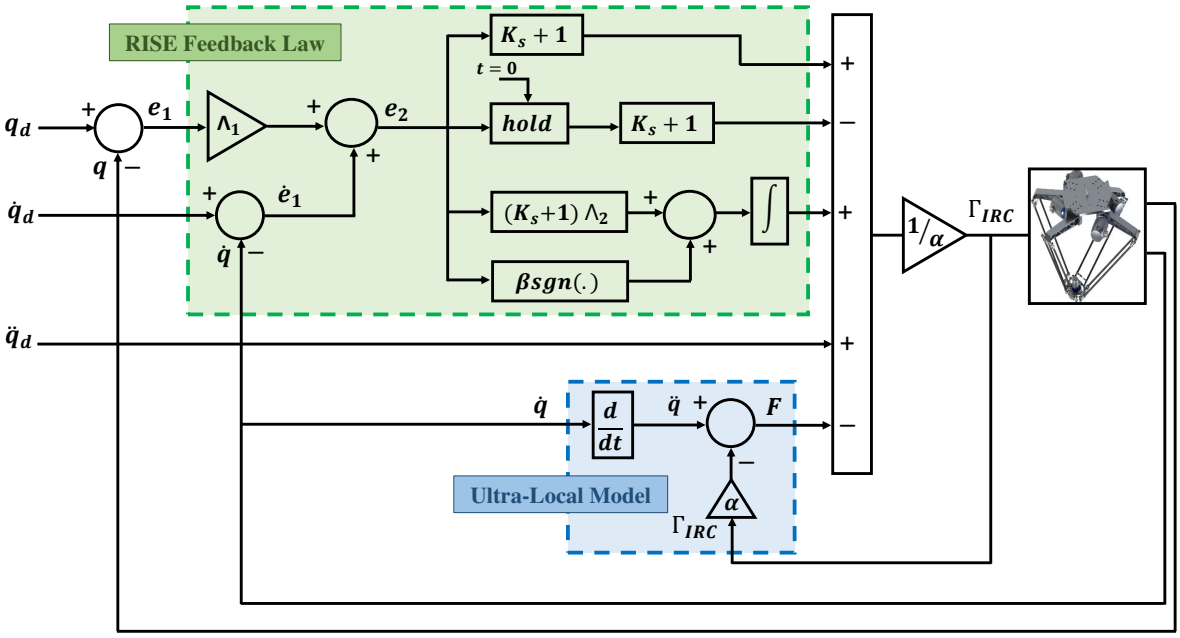



Figure 3.3 – Block diagram of the proposed IRC scheme.

#### 3.4.3.3 Stability analysis

**Theorem 3.4.1.** *The proposed control strategy (3.82) applied to the* $2^{nd}$ *order nonlinear uncertain MIMO system whose dynamics model is denoted by (3.5) guarantees that the tracking error* $e_1(t)$ *and its derivatives are bounded and go asymptotically to zero as time goes to infinity provided that the control parameters are chosen such that* $K_s$ *is large enough,* $\Lambda_1 > \frac{1}{2}$ *and* $\Lambda_2 > 1$*, while the robustness gain* $\beta$ *should verify the inequality given by:*

$$\beta > \alpha \|W_d(t)\|_{\mathscr{L}_\infty} + \frac{\alpha}{\Lambda_2} \|\dot{W}_d(t)\|_{\mathscr{L}_\infty} \tag{3.88}$$

*Proof.* A function $L(t) \in \mathbb{R}$ given by the following equation is considered:

$$L(t) = r(W_d(t) - \frac{\beta}{\alpha} sgn(e_2)) \tag{3.89}$$

Knowing that the gain $\beta$ should be chosen to satisfy the above condition (3.88).

Consider the following inequality:

$$\int_{t_0}^{t} L(\tau)d\tau \leq \frac{\beta}{\alpha}|e_2(t_0)| - e_2(t_0)W_d(t_0) \tag{3.90}$$

An additional function $P(t)$ is introduce as follows:

$$P(t) = \frac{\beta}{\alpha}|e_2(t_0)| - e_2(t_0)W_d(t_0) - \int_{t_0}^{t} L(\tau)d\tau \tag{3.91}$$

According to (3.89) and the condition of the chosen $\beta$ value, we can conclude that $P(t) \geq 0$, $\forall t \geq 0$. A continuously differentiable positive-definite function is considered as follows [Xian et al., 2004]:

$$V(y,t) = e_1^T e_1 + \frac{1}{2}e_2^T e_2 + \frac{1}{2}r^T M(q)r + P \tag{3.92}$$

where $y = [z^T \sqrt{P}]^T$ and $z(t)$ is defined previously. Thanks to the property of the mass matrix $M(q)$ mentioned above and its bounds, the considered function $V(y,t) : \mathbb{R}^{3n} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is bounded as below:

$$\eta_1 \|y\|^2 \leq V(y,t) \leq \eta_2(\|y\|)\|y\|^2 \tag{3.93}$$

with $\eta_1 = \frac{1}{2}\min\{1, \underline{m}\}$ and $\eta_2(\|y\|) = \max\{\frac{1}{2}\overline{m}(\|y\|), 1\}$. Differentiating the Lyapunov candidate (3.92) with respect to time and using the equations (3.87), (3.89) and (3.91), the following equation holds:

$$\dot{V}(y,t) = 2e_1^T e_2 - 2\Lambda_1 e_1^T e_1 - \Lambda_2 e_2^T e_2 - \frac{1}{\alpha}(K_s + I)r^T r + r^T \tilde{W} - \frac{1}{\alpha}r^T \dot{\tilde{\delta}} \tag{3.94}$$

Considering the fact that for any tow vectors a and b, $a^T b \leq (\|a\|^2 + \|b\|^2)/2$, the $\dot{V}(y,t)$ is upper bounded as follows:

$$\begin{aligned} \dot{V}(y,t) \leq \|e_1\|^2 + \|e_2\|^2 - 2\Lambda_1\|e_1\|^2 - \Lambda_2\|e_2\|^2 - \frac{1}{\alpha}(K_s + 1)\|r\|^2 \\ + \|r\|\rho(\|z\|)\|z\| - \frac{1}{2\alpha}(\|r\|^2 + \|\dot{\tilde{\delta}}\|^2) \end{aligned} \tag{3.95}$$

By rearranging the terms, the above inequality is rewritten as follows:

$$\dot{V}(y,t) \leq -\zeta_1\|e_1\|^2 - \zeta_2\|e_2\|^2 - \zeta_3\|r\|^2 - \mu\|r\|^2 + \|r\|\rho(\|z\|)\|z\| - \frac{1}{2\alpha}\|\dot{\tilde{\delta}}\|^2 \tag{3.96}$$

where $\zeta_1$, $\zeta_2$, $\zeta_3$ and $\mu$ are positive constants given by:

$$\zeta_1 = \Lambda_1 - \frac{1}{2}, \;\; \zeta_2 = \Lambda_2 - 1, \;\; \zeta 3 = \frac{3}{2\alpha}, \;\; \mu = \frac{K_s}{\alpha} \tag{3.97}$$

Regarding the above equations $\Lambda_1$ and $\Lambda_2$ should be chosen such that $\Lambda_1 > 1/2$ and $\Lambda_2 > 1$. Rearranging the equation (3.96) as follows:

$$\dot{V}(y,t) \leq -\eta_3 \|z\|^2 - (\mu \|r\|^2 - \|r\| \rho(\|z\|) \|z\|) \tag{3.98}$$

being $\eta_3 = \min\{\zeta_1, \zeta_2, \zeta_3\}$. Using the mathematical remarkable square identities, the upper-bound of $\dot{V}$ is obtained as follows:

$$\dot{V}(y,t) \leq -(\eta_3 - \frac{\rho^2(\|z\|)}{4\mu}) \|z\|^2 \triangleq -\xi \|z\|^2 \tag{3.99}$$

with $\xi$ is a positive constant $\in \mathbb{R}^+$, the following inequality stands:

$$\eta_3 > \frac{1}{4\mu} \rho^2(\|z\|) \tag{3.100}$$

Based on the inequality (3.100), we can introduce the domain $\mathscr{D}$ as follows:

$$\mathscr{D} = \{y \in \mathbb{R}^{3n} \times \mathbb{R}_{\geq 0} \mid \|y\| \leq \rho^{-1}(2\sqrt{\eta_3 \mu})\} \tag{3.101}$$

From the equations (3.93) and (3.99), $W_1(y) \leq V(y,t) \leq W_2(y)$ and $\dot{V}(y,t) \leq -W(y)$, respectively, it is clearly observed that $V(y,t) \in \mathscr{L}_\infty$. This can prove that $e_1$, $e_2$, $r$ are bounded.

$W_1(y), W_2(y)$ are continuous positive-definite functions $\forall t \geq 0$ and $\forall y \in \mathscr{D}$, and $W(y)$ is uniformly continuous positive-semidefinite function.

Let's define the subset $\mathscr{A}$ of $\mathscr{D}$ such that the initial condition $y(0) \in \mathscr{A}$ as follows:

$$\mathscr{A} = \{y(t) \in \mathscr{D} \mid \ W_2(y) < \eta_1 (\rho^{-1}(2\sqrt{\eta_3 \mu}))^2\} \tag{3.102}$$

Regarding the Lemma 2 in [Xian et al., 2004], we can conclude that $\|z\|^2 \to 0$ as $t \to 0$, $\forall\ y(0) \in \mathscr{A}$. As a consequence, the tracking error and its derivatives asymptotically go to zero as time converges to infinity.

$$e_1^{(i)}(t) \to 0 \ \text{ as } \ t \to \infty \ \ \forall \ \ y(0) \in \mathscr{A} \tag{3.103}$$

Hence, the proof of the stability of the proposed controller is made.

## 3.5 Conclusion

In this chapter, the thesis contributions related to the control of PKMs have been presented. The main objective was to design robust and efficient control strategies able to

compensate for errors coming from the controller itself, the system nonlinearities, the external disturbances, the measurement noise, etc.

As already mentioned, the intended application of this thesis is the use of PKMs in a selective sorting P&T task under different operating conditions. In such an application, the robot has to handle different types of objects (known, known with uncertainties, and possibly unknown). In the case of known objects where no significant variation of the parameters is noticed, a robust model-based controller that relies on the system dynamic model can be a good candidate for trajectory tracking. In this context, a new class of RISE control strategy applied to PKMs has been proposed. It involves adding a nonlinear model-based compensation term, inspired by PD control with compensation, to the standard RISE feedback control law. The compensation term is composed of dynamic parameters calculated online from the measured trajectories and then multiplied by the sum of the desired trajectories with the system state errors. To overcome the error resulting from the use of the desired joint accelerations and velocities, the resulting controller is extended by a nonlinear auxiliary function based on the tracking error.

However, when dealing with uncertain/unknown objects, PKMs are subject to uncertainties and time-varying dynamic parameters that are very complex to be modeled. Thus, by adopting a model-based control scheme such as the proposed extended RISE control plus compensation, the system nonlinearities and abundant uncertainties cannot be well compensated especially at critical operating conditions, which may deteriorate the tracking performance. To solve this problem, a model-based adaptive controller based on DCAL was proposed as a second control solution. We proposed to amend the original DCAL with adaptive feedback gains according to the system errors to counteract disturbances and uncertainties. In addition, the controller was extended by a nonlinear sliding-based term to further improve its robustness against external disturbances. The proposed $\mathrm{RDCAL-AG}$ controller can be a good solution when an online estimation of uncertain and unknown dynamic parameters is needed, resulting in improved tracking performance.

Even though adaptive controllers such as the proposed robust DCAL with adaptive gains can overcome the drawbacks of model-based controllers by estimating dynamic parameters online, their performance is limited to the compensation of modeled dynamics, while robustness against unmodeled dynamics cannot always be ensured. In addition, the real-time implementation of parameter identification requires a high computational burden. To address these issues, an intelligent robust controller dedicated to high order nonlinear MIMO systems was proposed as a third control solution. It is a new design of the RISE control scheme consisting in exploiting a model-free term in its control equation. The extension by an ultra-local model can further empower the original RISE control with more robustness against disturbances and uncertainties as well as improve its overall tracking performance. The proposed IRC technique can compensate for modeled and unmodeled dynamics without requiring any apriori knowledge about the controlled system.

As a result, if there is no need to estimate the dynamic parameters online, the proposed IRC scheme may be the appropriate solution when we are looking for a robust and efficient control scheme that is simple to implement in real time.

Moreover, the stability analysis for the three proposed control solutions was formulated. In the next thesis chapter, the numerical simulation results using the proposed control solutions on T3KR PKM will be presented and discussed.

CHAPTER

# 4 Proposed control solution numerical simulation results

**Contents**


4.1 Introduction . . . . . 137
4.2 Generation of pick-and-throw reference trajectories . . . . . 138
4.3 Performance evaluation criteria . . . . . 139
4.4 Proposed numerical simulation scenarios . . . . . 140
4.5 Numerical simulation results of extended RISE control plus compensation 141
4.6 Numerical simulation results of robust DCAL with adaptive feedback gains 146
4.7 Numerical simulation results of intelligent robust control . . . . . 151
4.8 Comparison between the proposed controllers . . . . . 156
4.9 Conclusion . . . . . 161


## 4.1 Introduction

In this chapter, the numerical simulation results obtained with the control solutions proposed in Chapter 3 are demonstrated and discussed. Due to the limitations in the use of the prototype T3KR, the proposed controllers were tested only via numerical simulations using the T3KR computed kinematic and dynamic parameters. Indeed, high frequency torque control is required for the T3KR robot, and standard configurations are not sufficient. A specific drive configuration is required, which only the manufacturer can develop and deploy. Two scenarios are conducted for each simulation test: Robustness towards payload changes and robustness towards speed changes. The purpose behind these scenarios is to test our proposed controllers at different dynamic operating conditions. The

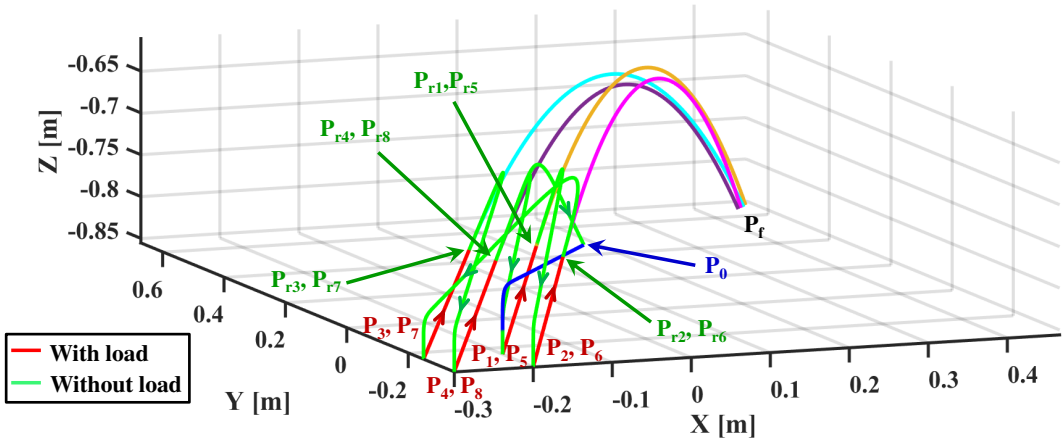



Figure 4.1 – 3D-view of the P&T reference trajectories of the robot (the red and green lines) with the ballistic motions of the thrown objects in Cartesian space.

results for each experimental test are plotted, clarified, and discussed in terms of the dynamic performance of the parallel manipulator. But first, let us start with the description of the reference trajectory and the adopted performance evaluation criteria.

## 4.2 Generation of pick-and-throw reference trajectories

The reference trajectories, illustrated in Figure 4.1, are generated using the method presented in Chapter 2. These generated trajectories correspond to the scenario where the robot has to successively throw eight objects of different masses to a target position, $\mathbf{P}_f$, located outside the robot workspace. First, the robot has to move from the central position $\mathbf{P}_0$ to the first pick position, $\mathbf{P}_1$, to grasp the detected object. Then, and according to the pick and target positions of the corresponding object, a release position $\mathbf{P}_{r_1}$ is calculated. After finding the appropriate release point, the robot accelerates to this point (i.e., $\mathbf{P}_{r_1}$), and then throws the object to the desired target $\mathbf{P}_f$. After throwing the object, the robot decelerates to pick the next object, while the released object follows its free flight ballistic trajectory from $\mathbf{P}_{r_1}$ to $\mathbf{P}_f$. The same cyclic movement is repeated for the second, the third, and the fourth objects, located at $\mathbf{P}_2$, $\mathbf{P}_3$, and $\mathbf{P}_4$, respectively. After throwing the fourth object, the robot moves to $\mathbf{P}_5$, to pick the fifth object. The same throw motion is performed for the last fourth objects located at $\mathbf{P}_5$, $\mathbf{P}_6$, $\mathbf{P}_7$, and $\mathbf{P}_8$, respectively. After throwing the last object, the robot moves back to $\mathbf{P}_0$. Referring to Figure 4.1, the red lines represent the portions of the trajectory where the robot carries the object, while the green lines correspond to the portions after the release point where the robot is moving without payload.

## 4.3 Performance evaluation criteria

To quantify the performance of the proposed control contributions, the Root Mean Square Error (RMSE) criterion is adopted. It is the most commonly used performance index for the evaluation of the tracking performance of control algorithms. It allows to quantify how close the actual trajectory is to the desired one. The RMSEs expressions for T3KR robot in Cartesian ($RMSE_x$) and joint space ($RMSE_J$) are given as follows:

$$RMSE_x = \sqrt{(\frac{1}{N}\sum_{i=1}^{N}(e_x^2(i)+e_y^2(i)+e_z^2(i))} \tag{4.1}$$

$$RMSE_J = \sqrt{(\frac{1}{N}\sum_{i=1}^{N}(e_{q_1}^2(i)+e_{q_2}^2(i)+e_{q_3}^2(i)+e_{q_4}^2(i))} \tag{4.2}$$

with $e_x$, $e_y$ and $e_z$ denote the Cartesian position tracking errors along the three translational axes, x, y and z. $e_{q_1}$, $e_{q_2}$, $e_{q_3}$ and $e_{q_4}$ are the joint position tracking errors. N is the total number of samples.

In fact, the control effort is reduced with the compensation of dynamic uncertainties and system nonlinearities. Therefore, the input-torque-based criterion is adopted to quantify the energy consumption. This evaluation criterion is denoted as follows:

$$E_\Gamma = \sum_{i=1}^{n}\sum_{j=1}^{N}|\Gamma_i(j)| \tag{4.3}$$

where $E_\Gamma$ is the total summation of the absolute values of the input torques delivered by the n actuators.

### 4.3.1 Tuning of the feedback control gains

Using the Trial-and-Error method, the feedback gains of all implemented controllers are adjusted. It is characterized by trying manually and continuously different sets of control gains in a real-time framework to get the best tracking performance while avoiding the saturation of the actuators. Increasing further the gains may amplify the noise effect and degrade the tracking performance. This method is mostly used when the formulated dynamic model does not exactly match the physical system, and thus automatic numerical closed-loop tuning methods may yield inadequate control gains for real-time experiments. It should be noted that when tuning the control gains, the desired control performance of each control law is achieved when the real system trajectory becomes close to desired one. It means that the minimum possible tracking error is obtained while respecting the actuator limits and avoiding chattering effects and noise amplifications.

## 4.4 Proposed numerical simulation scenarios

Numerical simulations were conducted on the T3KR robot in a P&T task to demonstrate the performance of the proposed controllers. These simulations were established in Matlab/Simulink environment with a fixed step solver equal to 0.4 ms, using the P&T reference trajectories illustrated in Figure 4.1. Two main scenarios have been implemented on this validation: 1) *scenario 1*: Robustness towards payload changes, 2) *scenario 2*: Robustness towards speed variations. For more realistic simulations, white noise of $7 * 10^{-}7$ of variance was added to the output joint positions, coulomb and viscous friction, $F_c = 0.8 and F_v = 0.5$, respectively, were added to the dynamic model of the robot as well as a 25% of uncertainty on the inertia values including the inertia of the actuators, the rear arms, and the forearms.

### 4.4.1 Scenario 1 - Robustness towards payload variations

This scenario has been performed with 4.2 G as maximum acceleration (with a speed of 2100 mm/s). It is the minimum sufficient value required by T3KR robot to throw an object outside of its workspace. The eight objects, used for this demonstration, have different masses, which allows for evaluating the robustness of the proposed controller towards variations in the payload. The $1^{st}$ and $5^{th}$ objects located at $\mathbf{P_1}$ and $\mathbf{P_5}$, respectively, are of 50 g of mass, the $2^{nd}$ and $6^{th}$ objects located at $\mathbf{P_2}$ and $\mathbf{P_6}$, respectively, have a mass of 100 g (i.e. $\Delta_{mass} = +100\,\%$ w.r.t the $1^{st}$ object), the $3^{rd}$ and $7^{th}$ objects at $\mathbf{P_3}$ and $\mathbf{P_7}$, respectively, have a mass of 150 g (i.e. $\Delta_{mass} = +200\,\%$ w.r.t the $1^{st}$ object), while the $4^{th}$ and $8^{th}$ ones located at $\mathbf{P_4}$ and $\mathbf{P_8}$, respectively, have a mass of 200 g (i.e. $\Delta_{mass} = +300\,\%$ w.r.t the $1^{st}$ object).

### 4.4.2 Scenario 2 - Robustness towards speed variations

The T3KR robot is intended to be used for high-speed P&T sorting applications. Accordingly, it is useful to evaluate the tracking performance of the proposed control strategies at high-acceleration conditions, where the nonlinear effects of the parallel manipulator increase substantially. The operating acceleration is increased up to 9 G (with a speed of 4500 mm/s). In this scenario, the robot performs the same P&T trajectory with the same manipulated objects as in the previous scenario.

Table 4.1 – The control gains of the original RISE, PD plus compensation, and the proposed extended RISE plus compensation controllers.

| **RISE** | | **PD$^+$** | **Proposed ERISE$^+$** | |
|---|---|---|---|---|
| $\Lambda_1 = 674$ | $\beta = 2.5$ | $K_p = 3400$ | $\Lambda_1 = 180$ | $\beta = 2.5$ |
| $\Lambda_2 = 7.2$ | | $K_d = 23$ | $\Lambda_2 = 1.3$ | $\delta_1 = 50$ |
| $K_s = 22$ | | $\alpha = 1.4$ | $K_s = 22$ | $\delta_2 = 100$ |

## 4.5 Numerical simulation results of extended RISE control plus compensation

To demonstrate the performance of the proposed ERISE$^+$ controller, a comparative study has been performed with the original RISE and the PD control plus compensation, presented in Chapter 3, through numerical simulations on T3KR robot in a P&T task.

### 4.5.1 Tuning procedure of the control gains

The control gains of the proposed ERISE$^+$ controller are tuned in the same way of those of the original RISE controller [Saied, 2019] using the following procedure:

1. Set $\Lambda_2 = 0$, $\beta = 0$, $\delta_1 = 0$, and $\delta_2 = 0$.
2. Set $\Lambda_1$ and $K_s$ as if it was a PD control plus compensation, where $\Lambda_1$ $(K_s + 1)$ is the proportional gain and $(K_s + 1)$ is the derivative gain until reaching values closer to the reference path.
3. Start increasing $\Lambda_2$ by changing again $\Lambda_1$ and $K_s$ either increasing or decreasing until obtaining the smallest possible RMSE at each simulation, taking care not to exceed the actuator capabilities.
4. Increase $\beta$ gradually to avoid chattering effects, to improve the robustness of the controller towards disturbances.
5. Increase $\delta_1$ and $\delta_2$ to improve the overall performance while keeping control input torques below saturation.

The obtained control gains for the proposed ERISE$^+$, the standard RISE, and PD control plus compensation are summarized in Table 4.1.

### 4.5.2 Scenario 1 - Robustness towards payload changes

The Cartesian tracking errors of the three tested controllers are plotted in Figure 4.2. The obtained results clearly show the superiority of the proposed controller over the other two ones along all axes. The RMSE performance indices are evaluated for all controllers, in

both Cartesian and joint spaces, and the obtained results are reported in Table 4.2. According to these indices, the proposed ERISE$^+$ outperforms the standard RISE by 80.9% and 69.6% for the Cartesian and joint spaces, respectively. Compared to the PD control plus compensation, the proposed ERISE$^+$ improves the tracking performance by up to 45.3% and 40.0% in the Cartesian and joint spaces, respectively.

For clarification purpose, the plots are zoomed to the interval [5.1, 5.2] seconds as shown in Figure 4.3. One can notice that the best dynamic performance is obtained by the proposed ERISE$^+$ controller showing the high impact of extending the robust RISE controller with a dynamic compensation on the global performance of parallel robots.

The evolution of the generated control input torques for all controllers is shown in Figure 4.4. The control signals show, for all controllers, a good and smooth behavior within the admissible limits of the actuators of the robot (the maximum torque of T3KR actuators is 28.9 N.m). In addition, a slight reduction in energy consumption is notified for the proposed ERISE$^+$ controller compared to the standard RISE controller. However, compared to the the PD$^+$, no significant improvement is observed in terms of energy consumption.

This scenario confirms the effectiveness of the proposed controller compared to the standard RISE and PD$^+$ controllers. The proposed ERISE$^+$ control scheme is more robust towards variations in payload, thus, it is more suitable for arbitrary P&T applications such as waste sorting.

Table 4.2 – Control performance evaluation of the original RISE, the PD plus compensation, and the proposed extended RISE plus compensation controllers.

| **Scenario** | **Control** | $RMSE_x$ **[mm]** | $RMSE_J$ **[Deg]** |
|---|---|---|---|
| Scenario 1 | RISE | 0.0634 | 0.0069 |
| | PD$^+$ | 0.0221 | 0.0035 |
| | ERISE$^+$ | 0.0121 | 0.0021 |
| | **Improvement w.r.t RISE** | **80.9 %** | **69.6 %** |
| | **Improvement w.r.t PD$^+$** | **45.3%** | **40.0 %** |
| Scenario 2 | RISE | 0.1332 | 0.0173 |
| | PD$^+$ | 0.0342 | 0.0074 |
| | ERISE$^+$ | 0.0266 | 0.0060 |
| | **Improvement w.r.t RISE** | **80.0 %** | **65.3%** |
| | **Improvement w.r.t PD$^+$** | **22.2 %** | **18.9 %** |

### 4.5.3 Scenario 2 - Robustness towards speed variations

In Figure 4.5, one can obviously see the significant improvements obtained by the proposed ERISE$^+$ control scheme along all translational axes. These improvements are quan-

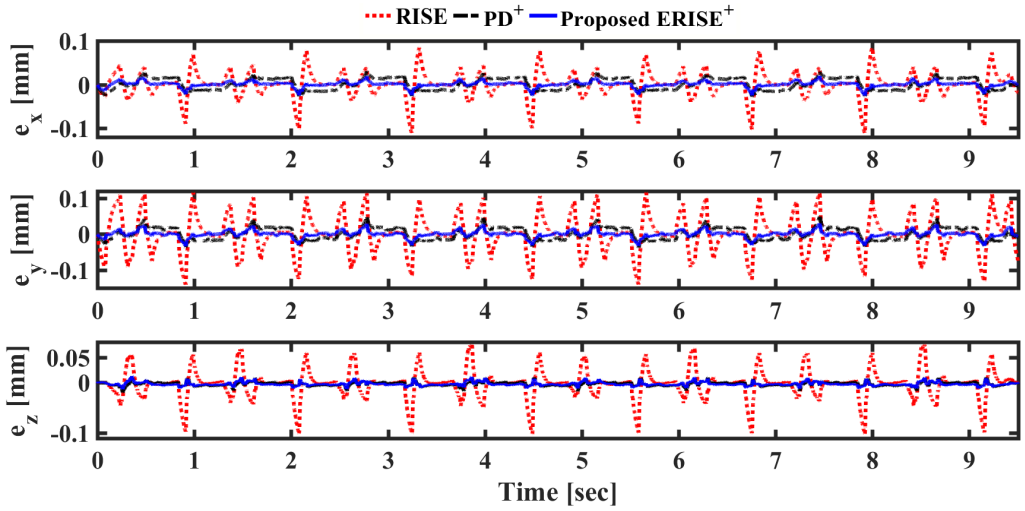



Figure 4.2 – Scenario 1: Evolution of the Cartesian tracking errors of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot.

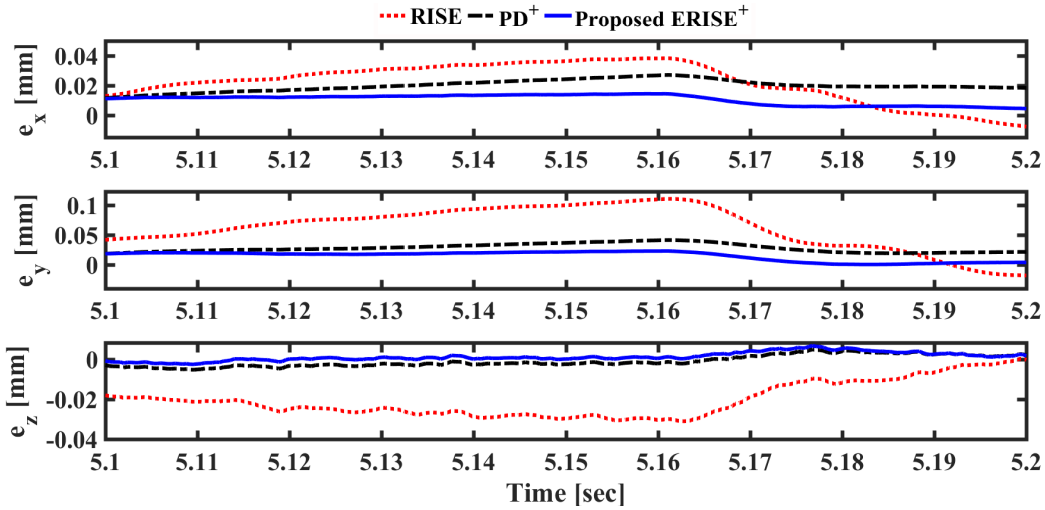



Figure 4.3 – Scenario 1: Evolution of the Cartesian tracking errors of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot within the interval [5.1, 5.2] sec.

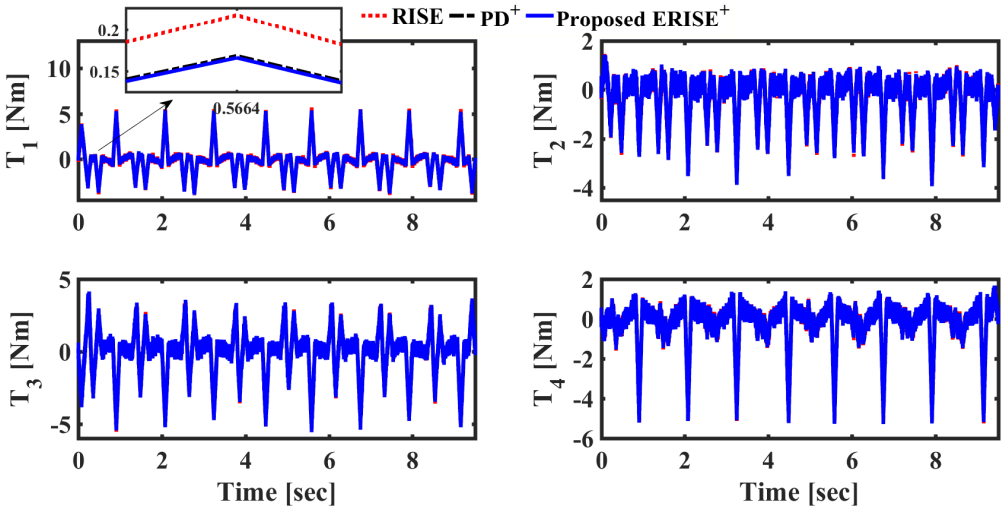



Figure 4.4 – Scenario 1: Evolution of the control input torques of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot.

tified by exploiting the RMSE evaluation criteria in Cartesian and joint spaces. The obtained results, summarized in Table 4.2, show improvements of 80.0% in the Cartesian space and 65.3% in the joint space compared to the standard RISE controller. In comparison to the PD control plus compensation, the tracking performance is improved by up to 22.2% and 18.9% in the Cartesian and joint spaces, respectively.

For better illustration, the plots are zoomed to the interval [2.14, 2.24] seconds as shown in Figure 4.6. One can notice the superiority of the ERISE$^+$ controller compared to the standard RISE and the PD$^+$ controllers in terms of the precision.

The evolution of the control input torques, generated by the three controllers, are depicted in Figure 4.7. It can be seen that all the control signals are continuous and evolve within the admissible range of the actuator capabilities. Moreover, as shown in Figure 4.7, the proposed controller slightly reduces the power consumption as it generates less input torques, compared to the standard RISE controller, while there is no noticeable reduction compared to the PD$^+$ controller.

The overall performance improvement, obtained by the proposed ERISE$^+$ scheme, can be explained by the good compensation of the system nonlinearities provided by the contribution of the designed compensation dynamic term.

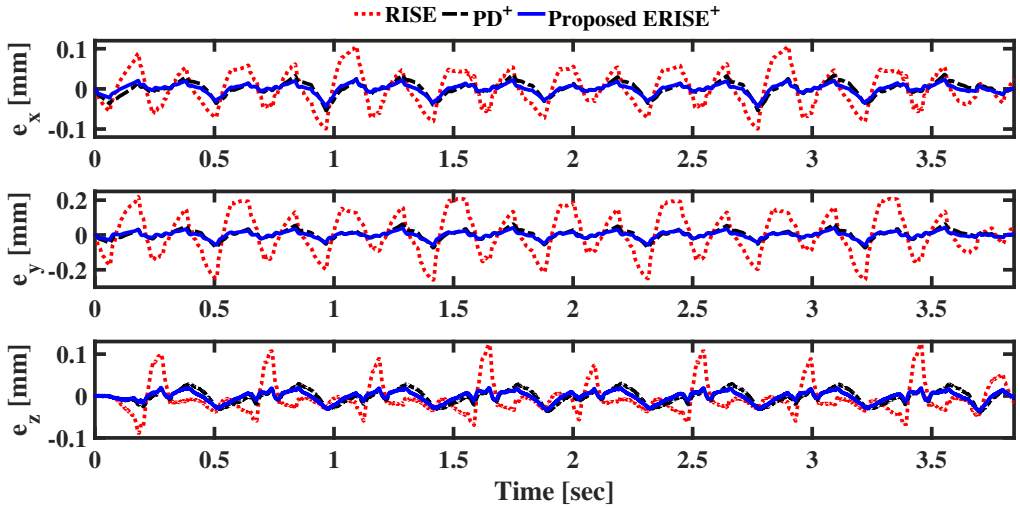



Figure 4.5 – Scenario 2: Evolution of the Cartesian tracking errors of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot.

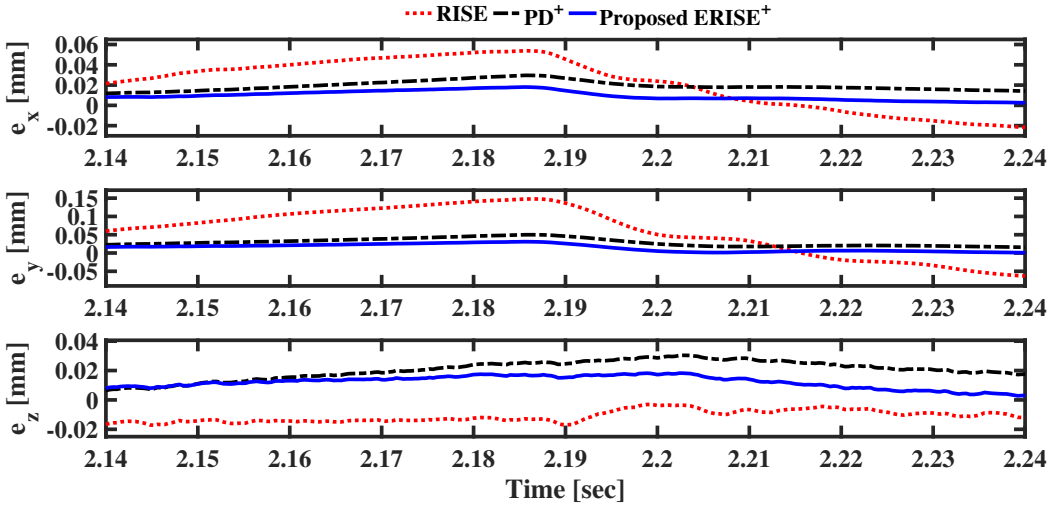



Figure 4.6 – Scenario 2: Evolution of the Cartesian tracking errors of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot within the interval [2.14, 2.24] sec.

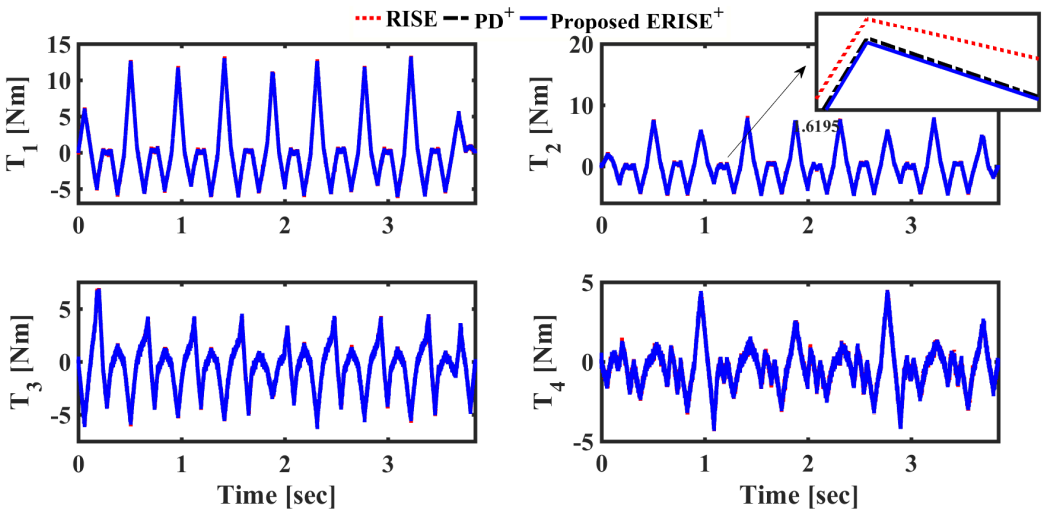



Figure 4.7 – Scenario 2: Evolution of the control input torques of the original RISE, the PD plus compensation, and the proposed ERISE$^+$ controllers on T3KR robot.

## 4.6 Numerical simulation results of robust DCAL with adaptive feedback gains

A comparison between the standard DCAL and the proposed robust DCAL with adaptive gain controller has been established using the P&T reference trajectories illustrated in Figure 4.1.

### 4.6.1 Tuning procedure of the control gains

For the proposed RDCAL-AG controller, we adopt the following procedure:

1. Set a high value for $\lambda$ and minimum possible values for $\Lambda_{pm}$ and $\Lambda_{vm}$.
2. Adjust the gains $\bar{\Lambda}_p$ and $\bar{\Lambda}_v$, either increasing or decreasing, until a good performance index is achieved, while being careful to avoid actuator saturation.
3. Increase the K gain, responsible for the parameter estimation, gradually till obtaining a good convergence of the mass of the platform.
4. Increase the $\sigma$ gain to improve the overall performance while keeping the control input values away from saturation.
5. The gain $\beta$, responsible for the robustness of the controller, is increased progressively in order to obtain better performance while maintaining low chattering input signals.

The resulting gains values of the proposed controllers are summarized in Table 4.3.

Table 4.3 – The control gains of the standard DCAL and the proposed robust DCAL with adaptive gains.

| **Standard DCAL** | | **Proposed RDCAL-AG** | |
|---|---|---|---|
| $\Lambda_p = 3600$ | $\sigma = 6.8\times10^6$ | $\Lambda_{pm} = 6364$ | $\sigma = 6.8\times10^6$ |
| $\Lambda_v = 23$ | $K = 200$ | $\overline{\Lambda}_p = 11\times10^7$ | $\lambda = 200$ |
| $\lambda = 200$ | | $\Lambda_{vm} = 23$ | $K = 300$ |
| | | $\overline{\Lambda}_v = 77$ | $\beta = 2.5$ |

Table 4.4 – Control performance evaluation of the standard DCAL and the proposed robust DCAL with adaptive feedback gains.

| **Scenario** | **Control** | $RMSE_x$**[mm]** | $RMSE_J$**[deg]** |
|---|---|---|---|
| Scenario 1 | DCAL | 0.0428 | 0.0062 |
| | RDCAL-AG | 0.0241 | 0.0035 |
| | **Improvement w.r.t DCAL** | **43.7 %** | **43.5 %** |
| Scenario 2 | DCAL | 0.0489 | 0.0065 |
| | RDCAL-AG | 0.0293 | 0.0039 |
| | **Improvement w.r.t DCAL** | **40.1 %** | **40.0%** |

### 4.6.2 Scenario 1 - Robustness towards payload changes

The Cartesian tracking errors of the standard DCAL and the proposed robust DCAL with adaptive feedback gains are plotted in Figure 4.8. It is clearly shown that the proposed controller outperforms the original DCAL especially for z-axis. This can validate the robustness of the proposed RDCAL-AG controller, towards the effect of gravity, compared to the original DCAL. The RMSE performance indices, in both Cartesian and joint spaces, are evaluated for both controllers and the obtained results are summarized in Table 4.4. These indices show a significant improvement of 43.7% in Cartesian space and 43.5% in joint space w.r.t to the standard DCAL.

The evolution of the estimated parameter $\hat{m}_{tp}$, initialized to zero, is displayed in Figure 4.9. It is worth to note that this mass includes both the mass of the carried payload and the mobile platform. This explains why the adaptive mass increases or returns to its nominal value depending on whether the robot is carrying a payload.

Figure 4.10 illustrates the evolution of the adaptive gains, $\Lambda_p(t)$ and $\Lambda_v(t)$, versus time. The produced variations of both feedback gains versus time give always strictly positive bounded values. One can observe, on the left side of Figure 4.10, that the minimum value of $\Lambda_p(t)$ is 6364 as defined in the value of $\Lambda_{pm}$; similarly, on the right side of Figure 4.10, the minimum value taken by $\Lambda_v(t)$ is the one as established in $\Lambda_{vm}$. Besides, it can be

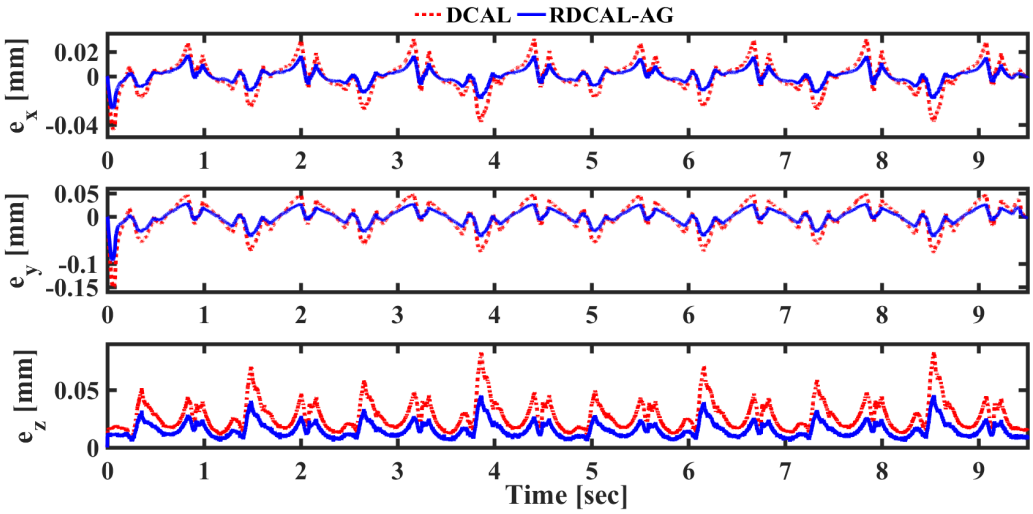



Figure 4.8 – Scenario 1: Evolution of the Cartesian tracking errors of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot.

seen from this figure that owing to this execution speed, $\Lambda_p(t)$ can reach values of up to 11500, whereas $\Lambda_v(t)$ can reach 23.5. This special behavior of the adaptive gains produces corrective actions when an increase in the tracking errors is due to changes in the handled payload during the P&T task for instance.

The evolution of the control input torques is depicted in Figure 4.11. The control signals show, for both controllers, a good and smooth behavior within the admissible limits of the actuators of the robot.

As a result, this scenario demonstrates the superiority of the proposed controller over the standard DCAL. The RDCAL-AG control scheme is more robust towards variations in payload thanks to the dynamic behavior of the proposed adaptive feedback gains and the addition of the sliding-based term. Therefore, the proposed RDCAL-AG strategy is more suitable for P&T applications.

### 4.6.3 Scenario 2 - Robustness towards speed variations

The Cartesian tracking errors for both controllers are recorded and depicted in Figure 4.12. The tracking errors of the both control schemes increased notably on all translational axes. Nevertheless, the proposed RDCAL-AG controller shows noticeably better performance, compared to the original DCAL. Table 4.4 summarizes the evaluation of the RMSE of both controllers in this scenario. It shows an improvement of 40.1% and 40.0% in Cartesian and joint spaces, respectively, compared to the standard DCAL.

Figure 4.13 shows the evolution of the estimated parameter $\hat{m}_{tp}$ with respect to time.

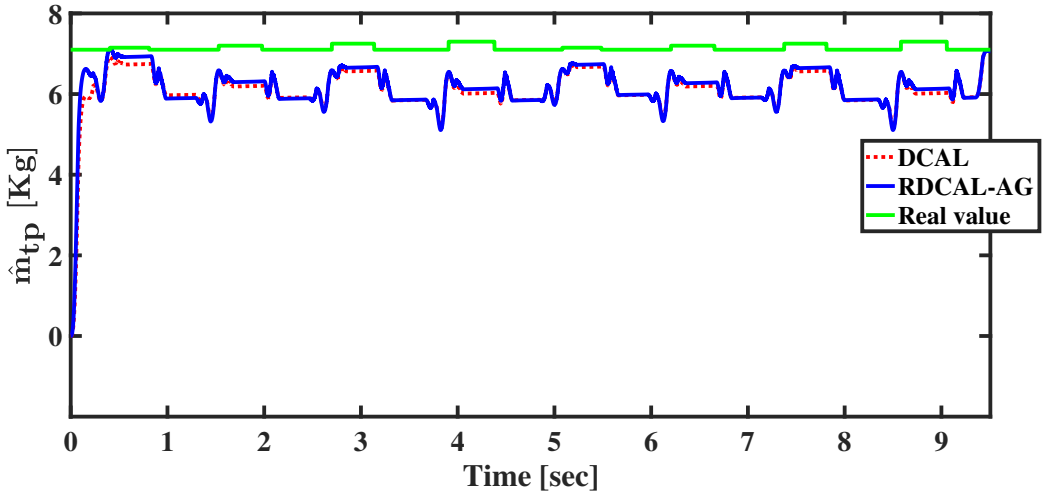



Figure 4.9 – Scenario 1: Evolution of the estimated mass of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot.

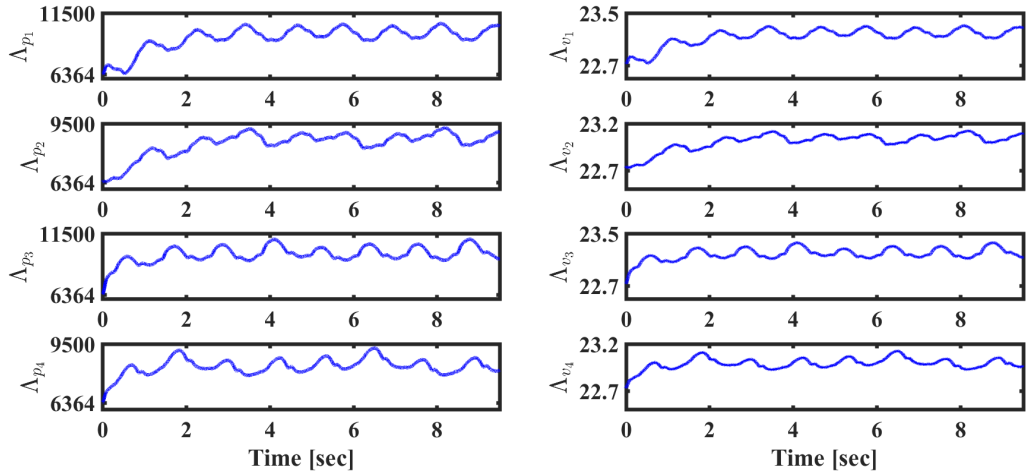



Figure 4.10 – Scenario 1: Evolution of the adaptive gains, $\Lambda_p(t)$ and $\Lambda_v(t)$, of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot.

The adaptation law adjusts the mass, from an initial zero value, similarly for both controllers. One can clearly observe the oscillations induced by the changes of the payload at each pick-and-throw cycle as well as by the increasing of the operating acceleration.

The evolution of the adaptive gains, $\Lambda_p(t)$ and $\Lambda_v(t)$, versus time is depicted in Figure 4.14. As it can be seen, the behavior of the adaptive gains, $\Lambda_p(t)$ and $\Lambda_v(t)$, is slightly modified by the increase in the operating acceleration, while remaining positive

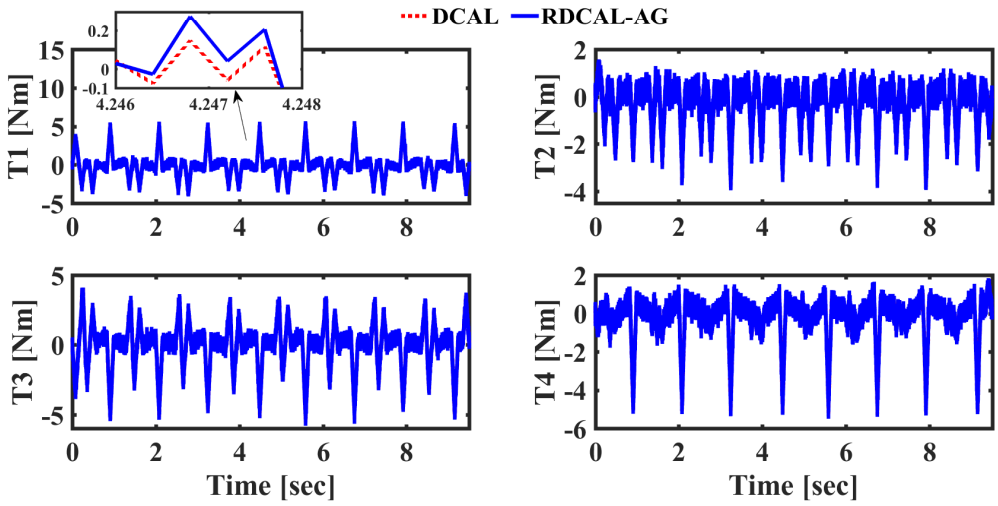



Figure 4.11 – Scenario 1: Evolution of the control input torques of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot.

and bounded along the desired trajectory. It is obvious from Figure 4.14 that the minimum values of these gains are those previously established for $\Lambda_{pm}$ and $\Lambda_{vm}$. However, the maximum values of these gains are modified; $\Lambda_p(t)$ manages to reach values close to 13000, while $\Lambda_v(t)$ reaches values up to 23.7. This explains the better performance of the proposed robust DCAL controller with adaptive gains compared to the original DCAL controller by the dynamic behavior of the feedback gains with the change of the operating conditions.

The control input signals for the four motors of the robot, for both controllers, are displayed in Figure 4.15. As one can see, the control signals of the two control schemes are within the allowable capacities of the motors. They are continuous and chattering-free since the sign function in the proposed controller is replaced by a continuous sigmoid function. Furthermore, it is clear that the values of the generated torques increase with the operating acceleration.

Besides, it can be noticed that the high nonlinearities and disturbances induced by the increase in the accelerations are significantly compensated by the proposed RDCAL-AG controller compared to the original DCAL. This is due to the corrective action produced by the adaptive gains and the robustness provided by the sliding-based term when the tracking errors are considerably increased by the changes in the dynamic operating conditions.

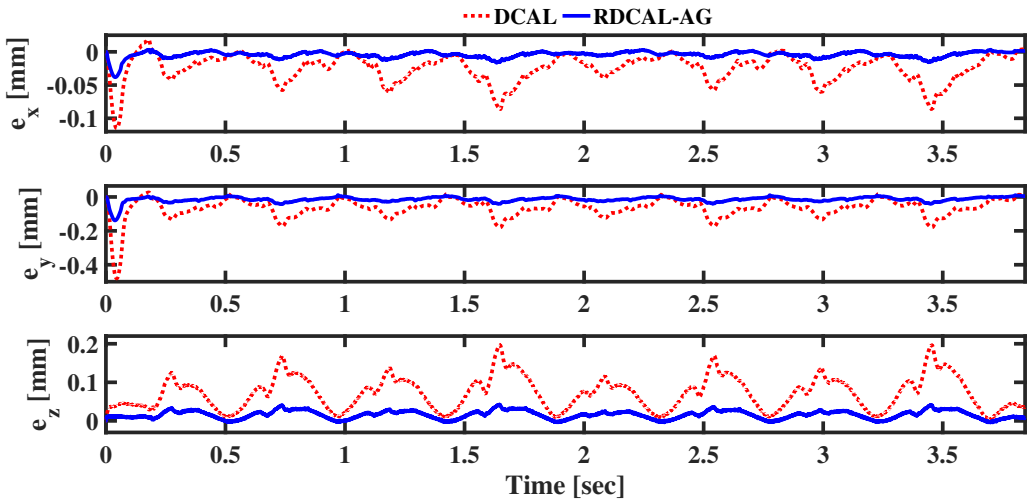



Figure 4.12 – Scenario 2: Evolution of the Cartesian tracking errors of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot.

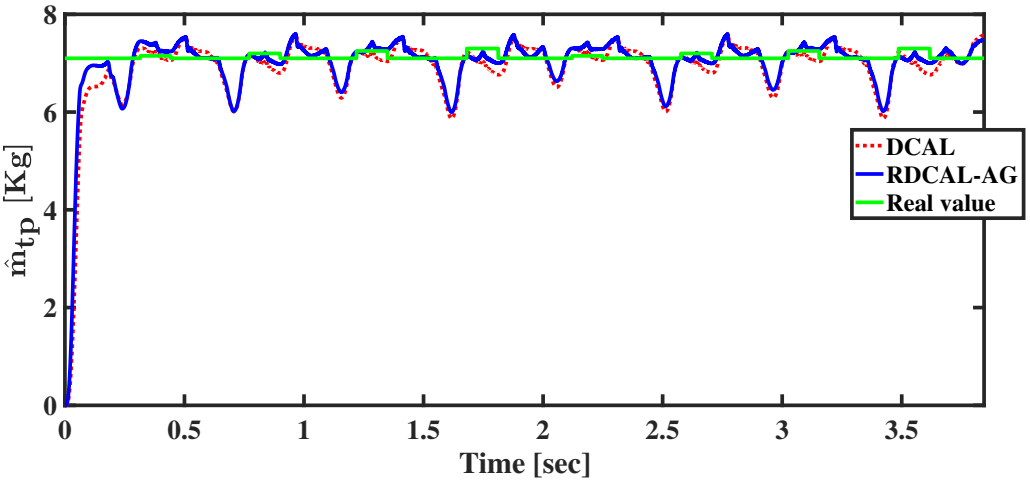



Figure 4.13 – Scenario 2: Evolution of the estimated mass of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot.

## 4.7 Numerical simulation results of intelligent robust control

In order to evaluate the performance of the proposed intelligent robust controller, a comparative study has been performed through numerical simulations with two model-based control schemes developed in the literature.

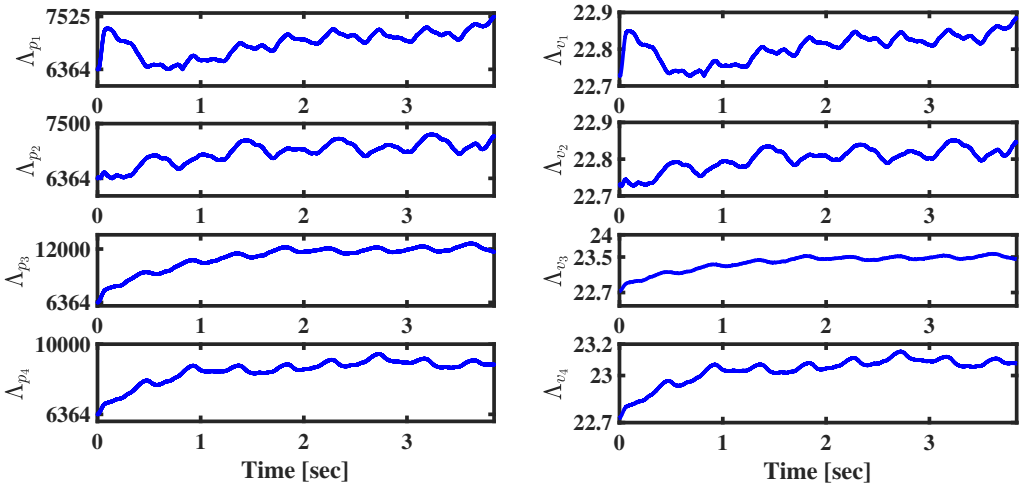



Figure 4.14 – Scenario 2: Evolution of the adaptive gains, $\Lambda_p(t)$ and $\Lambda_v(t)$, of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot.

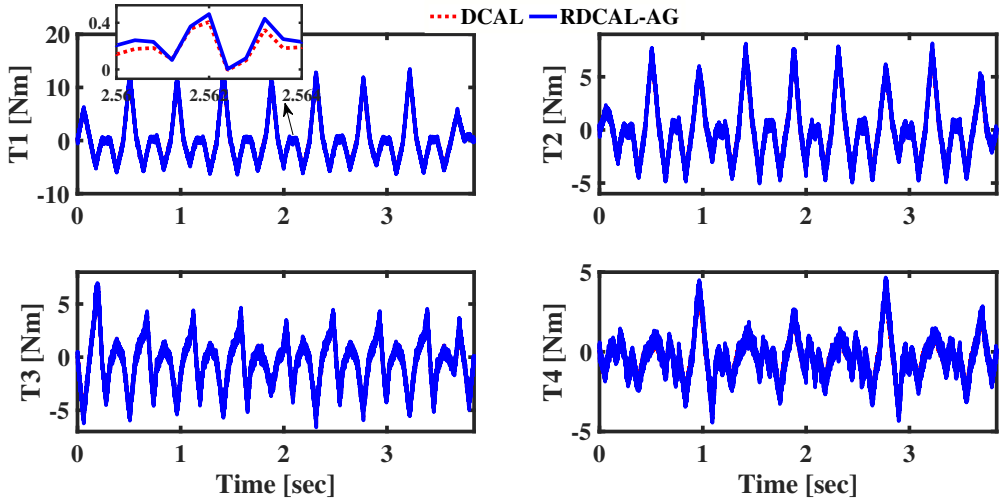



Figure 4.15 – Scenario 2: Evolution of the control input torques of the standard DCAL and the proposed robust DCAL with adaptive gains on T3KR robot.

### 4.7.1 State-of-the-art control methods for comparison purposes

The model-based feedforward RISE control (FF-RISE) developed in [Escorcia-Hernández et al., 2020b], and the model-based feedforward super-twisting sliding mode control (FF-ST-SMC) proposed in [Saied et al., 2021] for the control of PKMs, are also implemented for comparison purposes, since they are both model-based robust controllers.

Table 4.5 – The control gains of the feedforward RISE control, the feedforward ST-SMC, and the proposed intelligent robust control.

| **FF-RISE** | | **FF-ST-SMC** | | **Proposed IRC** | |
|---|---|---|---|---|---|
| $\Lambda_1 = 172$ | $\beta = 2.5$ | $K_1 = 18.2$ | $\lambda = 150$ | $\Lambda_1 = 150$ | $\beta = 2.5$ |
| $\Lambda_2 = 1.4$ | | $K_2 = 0.23$ | | $\Lambda_2 = 1.6$ | $\alpha = 2$ |
| $K_s = 22$ | | $K_3 = 2.5$ | | $K_s = 22$ | |

Their control laws are given as follows:

$$\Gamma_{FF-RISE} = M(q_d)\ddot{q}_d + C(q_d,\dot{q}_d)\dot{q}_d + G(q_d) + (K_s+I)e_2(t) - (K_s+I)e_2(t_0) + \int_{t_0}^{t} [(K_s+I)\Lambda e_2(\sigma) + \beta sgn(e_2(\sigma))]d\sigma. \tag{4.4}$$

$$\Gamma_{FF-ST-SMC} = M(q_d)\ddot{q}_d + C(q_d,\dot{q}_d)\dot{q}_d + G(q_d) + K_1 s + K_2|s|^{1/2}sgn(s) + \int K_3\, sgn(s). \tag{4.5}$$

where $s$ is the standard sliding surface defined as $s = \dot{e}_1 + \lambda e_1$. $K_1$, $K_2$, and $K_3$ are three positive definite diagonal gain matrices, and $\lambda$ is a positive constant gain.

### 4.7.2 Tuning procedure of the control gains

The gains of the proposed IRC strategy can be adjusted using the following procedure:

1. First, set the values of $\alpha$ and $\Lambda_1$ large enough to have a small steady-state error.
2. Decrease the value of $\alpha$ until getting a quick oscillatory response of the output.
3. Modify $\Lambda_1$ and $K_s$ to stabilize the system with a good tracking performance.
4. Increase $\Lambda_2$, and at the same time, adjust $\Lambda_1$ and $K_s$ up or down until tracking performance is improved as much as possible while keeping actuator torques in the safe range.
5. The robustness of the controller may be improved by increasing the feedback gain $\beta$ gradually to avoid chattering in the control signals.

The obtained control gains for all controllers are summarized in Table 4.5 .

### 4.7.3 Scenario 1 - Robustness towards payload changes

The Cartesian tracking errors of the three controllers are depicted in Figure 4.16. The obtained results show that the contribution of the ultra-local model improves the performance of the conventional FF-RISE controller by 51.6% and 63.3% for the Cartesian and

joint spaces, respectively (see Table 4.6). Compared to the FF-ST-SMC strategy, the proposed IRC improves the tracking performance by up to 24.8% and 47.6% in the Cartesian and joint spaces, respectively.

Figure 4.17 represents the evolution of the control input torques generated by all controllers, and clearly shows that the control signals are below the saturation limits of the actuators (i.e. below 28.9 N.m). Moreover, we can notice from Figure 4.17 that the proposed IRC strategy reduces the power consumption since it generates less input torques, compared to FF-RISE and FF-ST-SMC schemes.

This scenario confirms the improvement achieved by considering an ultra-local model, updated in real time, instead of a conventional feedforward dynamic model. According to the evaluation criteria, reported in Table 4.6, the proposed IRC algorithm shows higher accuracy and better robustness to payload variations than the two other model-based controllers; this is highly relevant for P&T sorting applications where the robot has to deal with different types of objects.

### 4.7.4 Scenario 2 - Robustness towards speed variations

In Figure 4.18, we can see the relevant improvements obtained by the proposed control scheme along all translation axes. These improvements are quantified by exploiting the RMSE evaluation criteria in Cartesian and joint spaces. The obtained results, summarized in Table 4.6, show improvements of 31.2% in the Cartesian space and 50.0% in the joint space relative to the FF-RISE controller. In comparison to the FF-ST-SMC scheme, the

Table 4.6 – Control performance evaluation of the feedforward RISE control, the feedforward ST-SMC, and the proposed intelligent robust control.

| **Scenario** | **Control** | $RMSE_x$ **[mm]** | $RMSE_J$ **[deg]** |
|---|---|---|---|
| Scenario 1 | FF-RISE | 0.0188 | 0.0030 |
| | FF-ST-SMC | 0.0121 | 0.0021 |
| | Proposed IRC | 0.0091 | 0.0011 |
| | **Improvement w.r.t FF-RISE** | **51.6 %** | **63.3 %** |
| | **Improvement w.r.t FF-ST-SMC** | **24.8 %** | **47.6 %** |
| Scenario 2 | FF-RISE | 0.0292 | 0.0064 |
| | FF-ST-SMC | 0.0270 | 0.0061 |
| | Proposed IRC | 0.0201 | 0.0032 |
| | **Improvement w.r.t FF-RISE** | **31.2 %** | **50.0 %** |
| | **Improvement w.r.t FF-ST-SMC** | **25.6 %** | **47.5 %** |

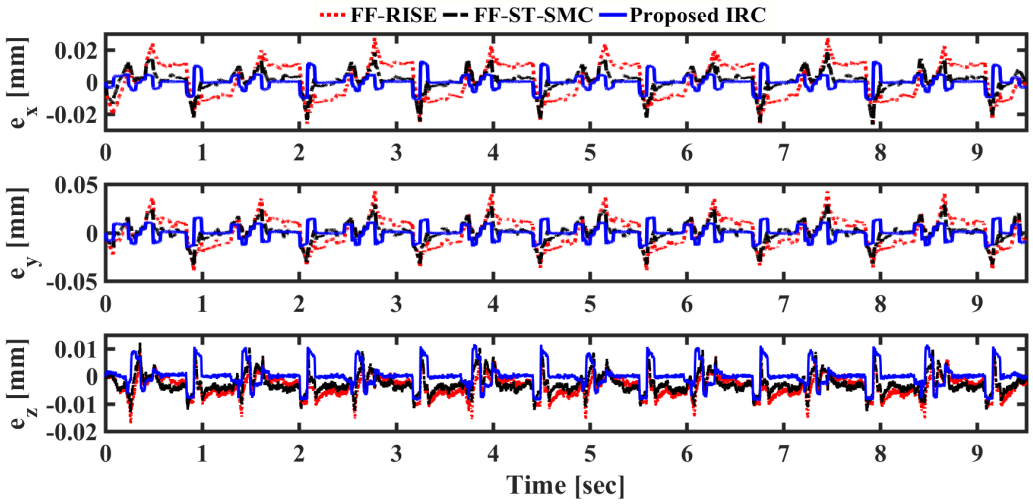



Figure 4.16 – Scenario 1: Evolution of the Cartesian tracking errors of the feedforward RISE control, the feedforward ST-SMC, and the proposed IRC scheme on T3KR robot.

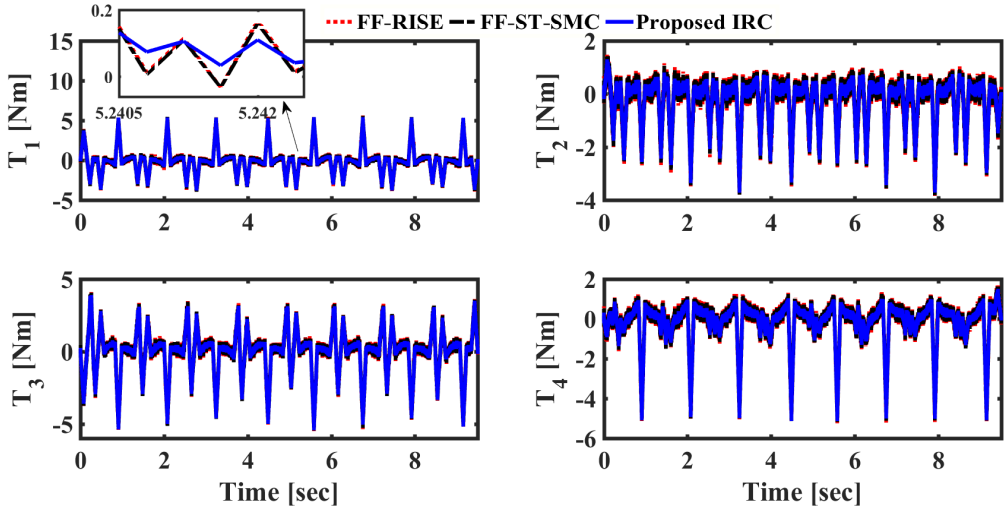



Figure 4.17 – Scenario 1: Evolution of the control input torques of the feedforward RISE control, the feedforward ST-SMC, and the proposed IRC scheme on T3KR robot.

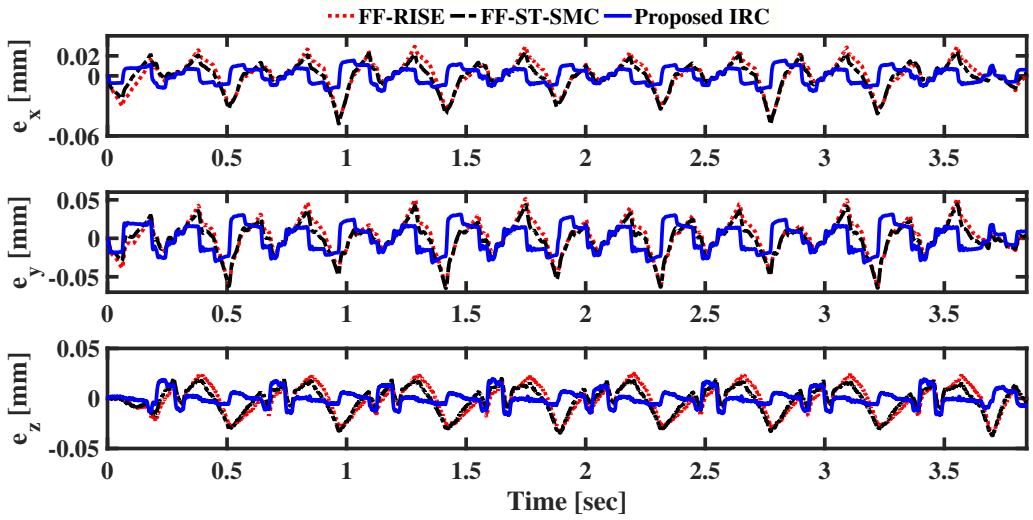



Figure 4.18 – Scenario 2: Evolution of the Cartesian tracking errors of the feedforward RISE control, feedforward ST-SMC, and the proposed IRC scheme on T3KR robot.

tracking performance is improved by up to 25.6% and 47.5% in the Cartesian and joint spaces, respectively.

The evolution of the control input signals, generated by the three implemented controllers, are displayed in Figure 4.19. Indeed, One can clearly see that all the control signals evolve within the admissible range of the actuator capabilities. As it can be seen from Figure 4.19, a slight reduction in energy consumption is notified for the proposed controller compared to the two other controllers.

The overall performance improvement, obtained by the proposed IRC scheme, can be explained by the good compensation of the system nonlinearities provided by the contribution of the ultra-local model with the robust RISE feedback law.

## 4.8 Comparison between the proposed controllers

In this section, the performance of the three proposed controllers is compared using the same desired trajectory shown in Figure 4.1. Scenarios 1 and 2, described above, are evaluated in this comparison to conclude which control performance is better in terms of robustness and accuracy among the three proposed controllers.

### 4.8.1 Scenario 1 - Robustness towards payload changes

The resulted tracking errors in Cartesian space are displayed in Figure 4.20 for the three proposed controllers. On the one hand, it is noticeable that the extended RISE control plus

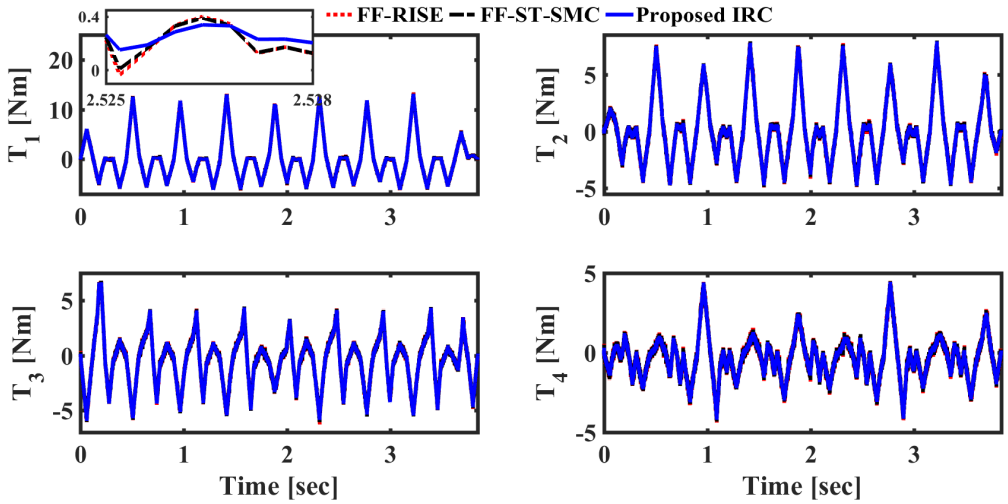



Figure 4.19 – Scenario 2: Evolution of the control input torques of the feedforward RISE control, feedforward ST-SMC, and the proposed IRC scheme on T3KR robot.

compensation provides better tracking performance compared to the robust DCAL with adaptive feedback gains. According to Table 4.7, when evaluating the RMSE performance index, the ERISE$^+$ controller outperforms the RDCAL-AG strategy by up to 49.8% and 40% for Cartesian and join spaces, respectively. On the other hand, a remarkable downsizing in the magnitudes of the tracking errors is obtained with the intelligent robust control scheme compared to the robust DCAL with adaptive gains and the extended RISE control plus compensation. A significant improvement of 62.2% in terms of Cartesian space accuracy and 68.6% for the joint space accuracy is noticed w.r.t the RDCAL-AG (see Table 4.7). Compared to the ERISE$^+$ controller, the IRC strategy improves the tracking precision by up to 24.8% and 47.6% for both Cartesian and joint spaces, respectively.

Figure 4.21 displays the evolution of the control input torques along the reference trajectory (shown in Figure 4.1) for the three proposed controllers. The three proposed control schemes, RDCAL-AG, ERISE$^+$, and IRC, guarantee an input signal within the safety margins of T3KR robot actuators. Nevertheless, it is worth to highlight the reduced input torques generated by the ERISE$^+$ controller compared to the RDCAL-AG as well as those generated by the IRC compared to ERISE$^+$ and RDCAL-AG controllers. The energy consumption of the three controllers, calculated using the input-torques based criterion proposed in (4.3), is reported in Table 4.7. An improvement of 2.9% in terms of energy consumption is noticed for the ERISE$^+$ controller compared to the RDCAL-AG. On the other hand, the IRC strategy reduces energy consumption by 5.2% compared to the ERISE$^+$ controller and by 2.3% compared to the RDCAL-AG.

According to the obtained results, it can be concluded that the proposed IRC scheme performs better than the other proposed controllers. The IRC strategy has the smallest

reference tracking error with reduced power consumption, and thus better performance overall reference trajectory. This is due to the special behavior provided by the ultra-local model in addition to the robust RISE feedback law in terms of robustness towards payload changes.

### 4.8.2 Scenario 2 - Robustness towards speed variations

Following the same reference trajectory shown in Figure 4.1, the Cartesian tracking errors for the three proposed controllers are recorded and plotted in Figure 4.22. A significant attenuation in the tracking errors, overall the reference trajectory, in all axes, can be observed for ERISE$^+$ compared to RDCAL-AG and for IRC compared to ERISE$^+$ and RDCAL-AG controllers. The RMSE performance index is evaluated in both Cartesian and joint space and is reported in Table 4.7. It shows a 9.2% improvement in terms of tracking precision in Cartesian space for ERISE$^+$ compared to RDCAL-AG. However, in terms of accuracy in joint space, the RDCAL-AG outperforms the ERISE$^+$ by 35.0%. With regard to the IRC scheme, its performance exceeds that of RDCAL-AG by 31.4% in Cartesian space and by 17.9% in joint space. Compared to ERISE$^+$, IRC improves tracking accuracy by 24.4% and 46.7% in Cartesian and joint space respectively.

The performance in terms of energy consumption is slightly improved from $7.55\times10^4$ Nm for the RDCAL-AG to $7.51\times10^4$ Nm for the ERISE$^+$ and with a reduction of 0.5% (see Table 4.7). Similarly, a slight reduction in energy consumption of 0.8% and 0.3% is noticed for the IRC compared to RDCAL-AG and ERISE$^+$, respectively. Note that the control signals for the three proposed controllers evolve within the safe range of the actuators capabilities.

Thus, the proposed IRC scheme produces smaller tracking errors at high dynamic operating conditions (high-speed motions) with manipulated payloads compared to the ERISE$^+$ and RDCAL-AG controllers. This ensures the robustness of the IRC strategy towards operating condition changes. Thanks to the ultra-local model, the increased effect of the nonlinear dynamics of parallel manipulators during high-speed motions can be better compensated by the proposed IRC scheme compared to ERISE$^+$ and RDCAL-AG controllers, even though no knowledge of the dynamic model is taken into account in the control loop.

To sum up, on the one hand, the better tracking performance of ERISE$^+$ compared to RDCAL-AG is explained by three reasons. First, the initial value of the estimated mass in the RDCAL-AG scheme is zero, therefore during the transition phase of the mass estimation, the initial tracking error of RDCAL-AG is higher than the errors of ERISE$^+$ and IRC. Second, the robustness term that characterizes the RISE control law can compensate for a large class of general uncertainties and disturbances. Moreover, the dynamic compensation term used in ERISE$^+$ is close to the one used in the simulation, even though some

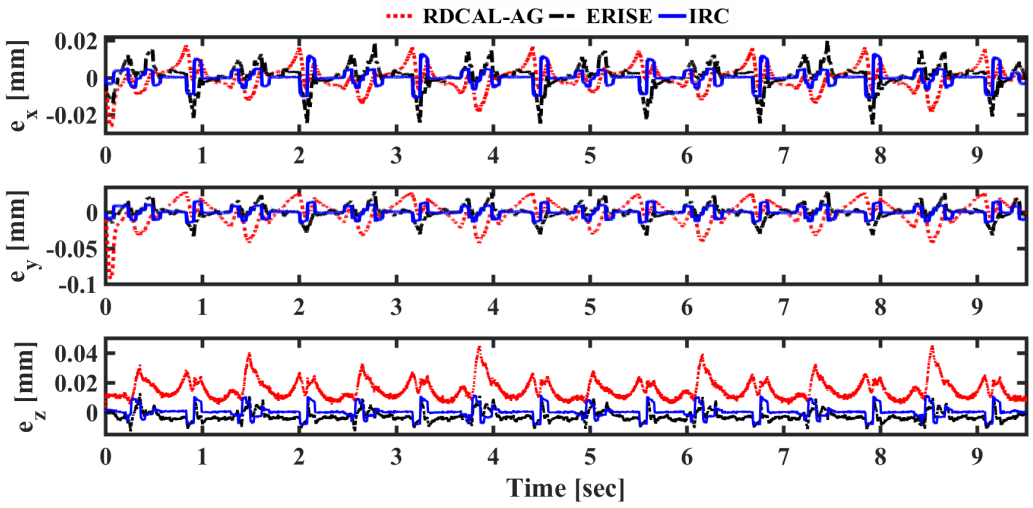



Figure 4.20 – Scenario 1: Evolution of the Cartesian tracking errors of the three proposed controllers: (i) the extended RISE control plus compensation, (ii) the robust DCAL with adaptive gains, and (iii) the intelligent robust control on T3KR robot.

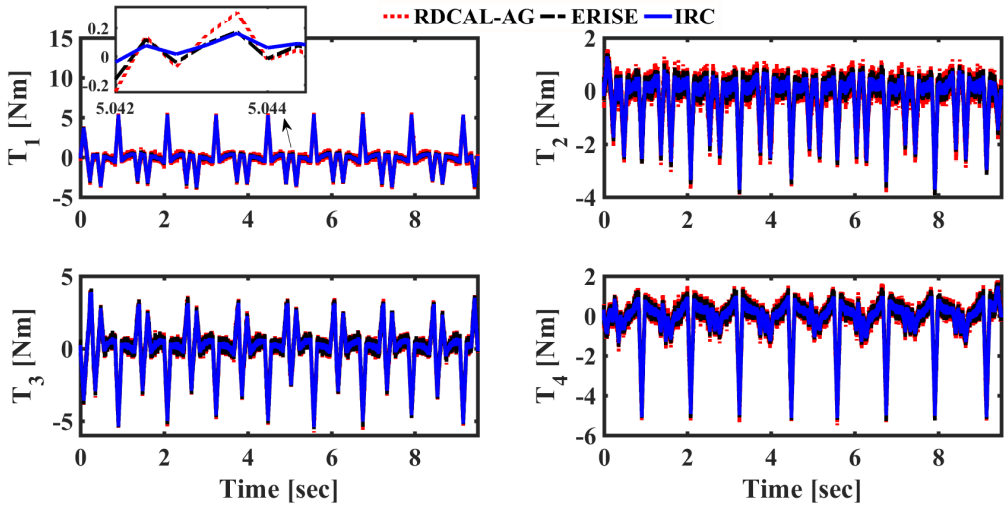



Figure 4.21 – Scenario 1: Evolution of the control input torques of the three proposed controllers: (i) the extended RISE control plus compensation, (ii) the robust DCAL with adaptive gains, and (iii) the intelligent robust control on T3KR robot.

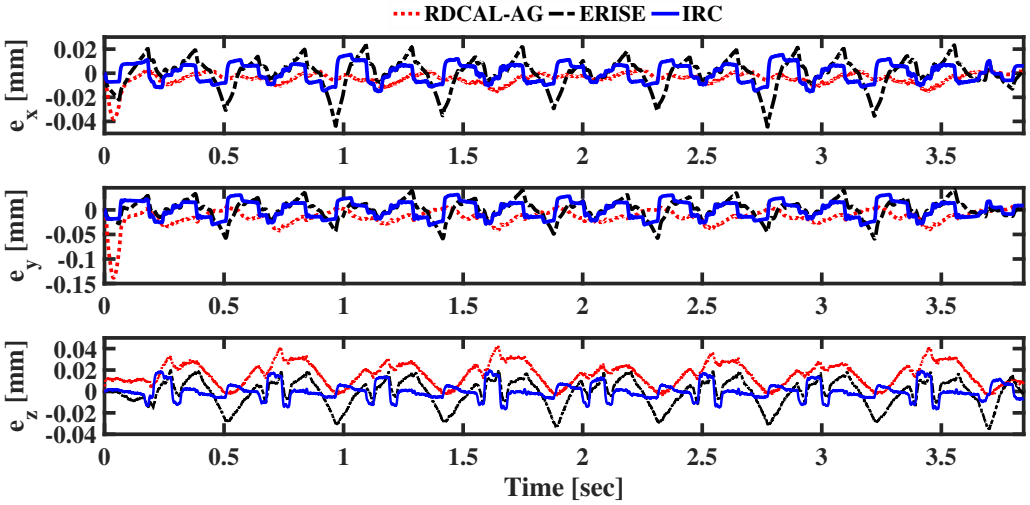



Figure 4.22 – Scenario 2: Evolution of the Cartesian tracking errors of the three proposed controllers: (i) the extended RISE control plus compensation, (ii) the robust DCAL with adaptive gains, and (iii) the intelligent robust control on T3KR robot.

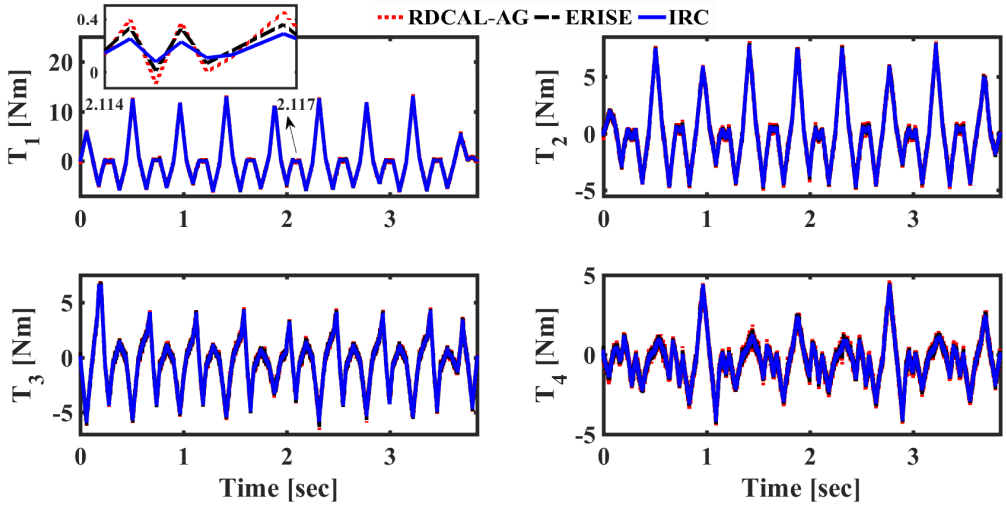



Figure 4.23 – Scenario 2: Evolution of the control input torques of the three proposed controllers: (i) the extended RISE control plus compensation, (ii) the robust DCAL with adaptive gains, and (iii) the intelligent robust control on T3KR robot.

uncertainties and disturbances were added to the dynamic model. On the other hand, according to the obtained results, it can be concluded that the IRC strategy provides the best tracking performance among the three proposed controllers. It shows higher robustness towards payload and speed changes compared to the other two controllers. Moreover, the tracking accuracy obtained by IRC is better than that obtained by ERISE$^+$ and RDCAL-AG. This can be explained by the fact that with the IRC control structure, uncertainties, nonlinearities, unmodeled phenomena, and parameter variations (i.e. payload changes) can be compensated for without considering dynamic model knowledge in the control. The latter confirm the effectiveness of an ultra-local model compared to an augmented dynamic compensation term and even compared to an adaptive feedforward term. This is of great importance for applications requiring high accuracy with an efficient computational burden. However, an online estimation of the model dynamic parameters (e.g., mass and inertia) is not offered by the IRC approach. This is because, as previously mentioned, the IRC scheme is not based on a dynamic model of the system. Indeed, it includes only a nonlinear function that is estimated at each sampling time and this function gathers the modeled and unmodeled system dynamics.

## 4.9 Conclusion

This chapter provided the numerical simulation validation and demonstration of the three control solutions proposed in this thesis. It began with a brief description of the trajectory generation, the tuning procedure of the control gains and the adopted performance evaluation criteria. All the proposed controllers were tested under two scenarios: Changes in the payload, and speed changes. Numerical simulations of the proposed extended RISE control plus compensation verified its superiority over the original RISE control and the PD plus compensation in terms of precision and robustness towards payload and speed variations. It was verified that incorporating more dynamic terms within

Table 4.7 – Control performance evaluation of the three proposed controllers: (i) the extended RISE control plus compensation, (ii) the robust DCAL with adaptive gains, and (iii) the intelligent robust control on T3KR robot.

| **Scenario** | **Control** | $RMSE_x$**[mm]** | $RMSE_J$**[deg]** | $E_\Gamma$**[Nm]** |
|---|---|---|---|---|
| Scenario 1 | RDCAL-AG | 0.0241 | 0.0035 | $7.49\times10^4$ |
| | ERISE$^+$ | 0.0121 | 0.0021 | $7.27\times10^4$ |
| | IRC | 0.0091 | 0.0011 | $7.10\times10^4$ |
| Scenario 2 | RDCAL-AG | 0.0293 | 0.0039 | $7.55\times10^4$ |
| | ERISE$^+$ | 0.0266 | 0.0060 | $7.51\times10^4$ |
| | IRC | 0.0201 | 0.0032 | $7.49\times10^4$ |

a model-based control strategy can improve the dynamic performance of parallel manipulators in terms of precision, nonlinearities compensation, and robustness towards payload and speed variations. Moreover, simulation results of the proposed robust DCAL with adaptive feedback gains showed a high dynamic performance in terms of high-speed motions, precision, and robustness towards payload and speed changes over the standard DCAL. This is due to the particular behavior of the proposed adaptive feedback gains with the robustness term. Furthermore, the comparison of the novel intelligent robust control with the model-based feedforward RISE and feedforward super-twisting SMC exploiting different performance indices confirms the superiority of the proposed IRC approach. The proposed IRC technique can compensate for modeled and unmodeled dynamics without requiring any apriori knowledge about the controlled system. A comparison between the three proposed controllers was carried out in this chapter. Although there is an improvement of the ERISE$^+$ controller over the RDCAL-AG controller as well as a superiority of the IRC scheme over ERISE$^+$ and RDCAL-AG, this cannot prevent the use of one of the proposed controllers. Indeed, each proposed controller is better suited to a specific type of applications. For instance, the proposed ERISE$^+$ controller is considered a suitable control solution in the case where the dynamic parameters are known or known with small uncertainties. However, the proposed RDCAL-AG strategy is useful when online estimation of the dynamic parameters is required. Lastly, the proposed IRC scheme is recommended when a high accuracy is required, with the emphasis on the simplicity of the control structure and its implementation in a real-time framework.

# General conclusion

The objectives of this thesis were mainly two. The first one was to generate a time-optimal P&T trajectory for parallel manipulators, and the second one was to propose and validate different control techniques to improve the dynamic performance of parallel manipulators. Real-time experiments of the proposed throwing trajectory show its relevance in terms of cycle time reduction and productivity maximization compared to the conventional P&P technique and an existing P&T method from the literature. Numerical simulation results of the control solutions corroborated the effectiveness of the new control proposals in dealing with uncertainties, speed changes and sudden payload changes.

## Summary of the work

It was mentioned earlier that the performance of a parallel robot is a function of various factors, such as mechanical design, mathematical modeling, motion planning, and control scheme. All of these factors must be well designed in order to get the most out of the closed-loop architecture of PKMs. This thesis focused on the trajectory generation and control of PKMs. As described in the first chapter, trajectory generation presents several challenges such as kinematic and dynamic constraints, continuity constraints and parameter optimization. In addition, the control of PKMs involves many challenges, such as highly nonlinear dynamics, the presence of parametric and non-parametric uncertainties, and in some cases, redundant actuation. Therefore, the control schemes should take into account all the above mentioned problems and challenges to ensure good tracking performance. In this thesis, the proposed, analyzed and validated contributions can be listed as follows:

1. A time-optimal Pick-and-Throw S-curve trajectory generation has been developed

and implemented on T3KR parallel robot with the goal of increasing robot productivity and reducing cycle time. The proposed technique consists in determining an optimal release configuration (i.e. position and velocity) allowing an object to be thrown towards a desired target while minimizing the travel time of the throwing motion of the robot. The corresponding optimization problem was formulated in two different ways where the second one is more efficient in terms of reducing computational time and simplicity in real-time implementation. A comparison between the proposed P&T technique, an existing P&T method and the conventional P&P has been carried out through real-time experiments on the T3KR parallel robot. The obtained experimental results show the efficiency of the proposed P&T method, over the two other methods, in terms of processing time minimization and, thereby, of productivity maximization.

2. A new augmented RISE feedback controller has been proposed and developed. It consists in revisiting the original RISE control law by incorporating a nonlinear dynamic compensation term in its control loop. This term, inspired by the PD control with compensation, consists of model parameters calculated online from the measured trajectories and multiplied by the sum of the desired trajectories with state errors. In addition, we propose to extend the resulting controller with a nonlinear feedback function to compensate for the errors resulting from using the desired trajectories (accelerations and velocities) instead of the measured ones. The proposed control contribution can compensate for PKM high nonlinearities as well as improve the robustness of the standard RISE controller. Numerical simulations were conducted on a T3KR parallel robot in a P&T task under different operating conditions to confirm the relevance of the proposed control scheme compared to the original RISE and the PD plus compensation controllers.
3. An extended version of the standard DCAL strategy was proposed and developed in which the PD linear feedback term was revisited by adopting adaptive gain functions of system errors and the resulting controller was extended by a sliding-based robustness term. The idea was motivated by the proved effectiveness of adaptive feedback gains used with different control schemes such as RISE control law and the robustness provided by a sliding-based term. DCAL is a model-based adaptive control strategy characterized by online estimation of dynamic parameters and its robustness towards measurement noise. Thus, the proposed robust DCAL with adaptive feedback gains takes the advantages of the adaptive feedback gains, the online estimation of the DCAL scheme, and the robustness against disturbances provided by the sliding-based term. The stability of the proposed control solution has been studied in the sense of Lyapunov stability showing an asymptotic convergence of the tracking error. The T3KR robot was used to validate the proposed controller in numerical simulations. The obtained results verified the superiority of the proposed

robust DCAL with adaptive feedback gains over the original DCAL algorithm in terms of precision and robustness towards payload and velocity variations.

4. A new robust intelligent control has been proposed and designed with the aim of taking into account parametric and non-parametric uncertainties and disturbances that are not compensated for in conventional model-based feedforward schemes or even in adaptive model-based schemes. The original RISE control law is a non-model-based controller that does not incorporate any knowledge of the system dynamics into its control loop. This can negatively affect the dynamic performance of the manipulator and reduce the robustness to changing operating conditions, especially in critical situations. The proposed intelligent robust control consists of revising the original RISE feedback law by incorporating an ultra-local model inspired by model-free control into its control equation. This extension can further improve the overall tracking performance of the original RISE and enhance its robustness against disturbances. The stability analysis of the proposed control solution has been addressed in the Lyapunov sense ensuring the asymptotic convergence of the tracking error and its derivatives. The T3KR parallel robot prototype was used to validate the proposed control strategy. Numerical simulations showed the superiority of the proposed control approach over the feedforward RISE control and the feedforward super-twisting sliding mode control under different operating conditions (payload and operating speed changes).

A comparison between the three proposed controllers was performed from a theoretical point of view in Chapter 3, and by numerical simulations at the end of Chapter 4. This comparison allows us to analyse the performance of each proposed controller compared to the others and to conclude the best controller meeting the main objective of this thesis. As previously stated, the objective is to use a PKM in a P&T sorting application accurately and efficiently despite challenges such as uncertainties, significant parameter variations, and high nonlinearities due to high operating acceleration. Therefore, it is recommended to use a robust controller that provides high precision all along the reference trajectory and is easy to implement in a real-time framework. Based on the comparison and discussion presented in the previous chapters, the proposed IRC scheme can meet these requirements. The IRC strategy can provide high accuracy as well as robustness towards payload and speed changes owing to the special behavior of its control structure. The ultra-local model with the RISE feedback term can compensate for the aforementioned challenges without considering any knowledge of the system dynamics in the control loop, thus achieving good tracking performance with a simple structure. This does not contradict the fact that the other two proposed controllers have effective performance and robustness, but the IRC scheme is the most suitable for the purpose of the thesis.

## Future works

In this thesis, a time-optimized trajectory has been proposed with three control strategies to improve the dynamic performance of parallel manipulators aiming to efficiently perform the waste sorting task. Indeed, different possibilities exist to extend the throwing trajectory and the control solutions proposed in this work and obtain better performances. The following extension possibilities can be mentioned:

- The optimized Pick-and-Throw trajectory, combined with an AI technique to identify recyclables, can be applied in real waste recycling industry. Furthermore, this work may be extended to consider not only the position of the object into the desired target but also its orientation in the target landing position.
- Even if uncertainties were included in simulations, it is well known that real experiments in robots tend to differ from simulations. In particular, model-based methods can perform really differently in reality than simulations. Also, parameters defined in simulations might not be achievable in reality without suffering problems (such as chattering). It would be interesting to compare on a real robot the proposed control solutions. Therefore, future directions may focus on validating the proposed control solutions through real-time experiments on a parallel manipulator prototype.
- In general, the dynamic parameters of the manipulator may vary over time or be unknown, thus, a real-time estimation of these modeled parameters may be considered to further improve the tracking performance of the extended RISE plus compensation contribution.
- For the robust DCAL with adaptive feedback gains contribution, the gain of the sliding-based term was chosen manually. Therefore, theoretical approaches can be considered to justify how the gain of the sliding-based term can be chosen to maximize robustness.
- Nonlinear or adaptive gains depending on the system state errors can be adopted in the feedback control term of the proposed intelligent robust control scheme instead of static gains. The time-varying gains can produce corrective control actions ensuring the establishment of the desired performance.

# Bibliography


Abbasnejad, G. and Carricato, M. (2015). Direct geometrico-static problem of under-constrained cable-driven parallel robots with n cables. *IEEE Transactions on Robotics*, 31(2):468–478. Cited page 22.

Abdellatif, H. and Heimann, B. (2009). Computational efficient inverse dynamics of 6-DOF fully parallel manipulators by using the lagrangian formalism. *Mechanism and Machine Theory*, 44(1):192–207. Cited page 44.

Aggravi, M., Estima, D. A., Krupa, A., Misra, S., and Pacchierotti, C. (2021). Haptic teleoperation of flexible needles combining 3D ultrasound guidance and needle tip force feedback. *IEEE Robotics and Automation Letters*, 6(3):4859–4866. Cited page 37.

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. Cited page 44.

Al Homsi, S. (2016). *Online generation of time-optimal trajectories for industrial robots in dynamic environments*. PhD thesis, Université Grenoble Alpes. Cited page 40.

Al Younes, Y., Drak, A., Noura, H., Rabhi, A., and El Hajjaji, A. (2014). Model-free control of a quadrotor vehicle. In *2014 International conference on unmanned aircraft systems (ICUAS)*, pages 1126–1131. IEEE. Cited page 130.

AMP (2019). AMP Cortex Dual-Robot System (DRS). https://www.amprobotics.com/. Cited page 38.

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. Cited page 54.

Andary, S., Chemori, A., Benoit, M., and Sallantin, J. (2012). A dual model-free control of underactuated mechanical systems, application to the inertia wheel inverted pendulum. In *2012 American Control Conference (ACC)*, pages 1029–1034. IEEE. Cited page 130.

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. Cited page 33.

Assima, G. P., Larachi, F., Schleicher, E., and Schubert, M. (2015a). Capacitance wire mesh imaging of bubbly flows for offshore treatment applications. *Flow Measurement and Instrumentation*, 45:298–307. Cited page 33.

Assima, G. P., Motamed-Dashliborun, A., and Larachi, F. (2015b). Emulation of gas-liquid flow in packed beds for offshore floating applications using a swell simulation hexapod. *AIChE Journal*, 61(7):2354–2367. Cited page 33.

Bai, Y., Chen, X., Sun, H., and Yang, Z. (2018). Time-optimal freeform s-curve profile under positioning error and robustness constraints. *IEEE/ASME Transactions on Mechatronics*, 23(4):1993–2003. Cited page 69.

Baklouti, S. (2018). *Vibration Analysis and Reduction of Cable-Driven Parallel Robots*. PhD thesis, Rennes, INSA. Cited page 21.

Baklouti, S., Courteille, E., Lemoine, P., and Caro, S. (2019). Vibration reduction of cable-driven parallel robots through elasto-dynamic model-based control. *Mechanism and Machine Theory*, 139:329–345. Cited page 21.

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. Cited page 30.

Begey, J., Cuvillon, L., Lesellier, M., Gouttefarde, M., and Gangloff, J. (2018). Dynamic control of parallel robots driven by flexible cables and actuated by position-controlled winches. *IEEE Transactions on Robotics*, 35(1):286–293. Cited page 21.

Bennehar, M. (2015). *Some contributions to nonlinear adaptive control of PKMs: from design to real-time experiments*. PhD thesis, Université Montpellier. Cited pages 20 and 30.

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. Cited pages 58 and 113.

Bennehar, M., Chemori, A., Krut, S., and Pierrot, F. (2014). Continuous closed form trajectories generation and control of redundantly actuated parallel kinematic manipulators. In *2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14)*, pages 1–6. IEEE. Cited page 41.

Bennehar, M., Chemori, A., and Pierrot, F. (2015a). $\mathscr{L}_1$ 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. Cited pages 52 and 53.

Bennehar, M., Chemori, A., and Pierrot, F. (2016). A new revised desired compensation adaptive control for enhanced tracking: application to RA-PKMs. *Advanced Robotics*, 30(17-18):1199–1214. Cited pages 58 and 124.

Bennehar, M., Chemori, A., Pierrot, F., and Creuze, V. (2015b). Extended Model-Based Feedforward Compensation in $\mathscr{L}_1$ Adaptive Control for Mechanical Manipulators: Design and Experiments. *Frontiers in Robotics and AI*, 2:32. Cited page 58.

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. Cited pages 58 and 61.

BHS (2017). Max-AI Autonomous Quality Control (Max-AI AQC) robotic sorter. https://www.youtube.com/watch?v=2gjUpDnJrZA. Cited page 38.

BHS (2019). Max-AI AQC-C Recycling collaborative robot CoBot. https://www.youtube.com/watch?v=tEAr1w1Jxww. Cited pages 40 and 62.

Biagiotti, L. and Melchiorri, C. (2008). *Trajectory Planning for Automatic Machines and Robots*. Springer-Verlag. Cited pages 41 and 69.

Biagiotti, L. and Melchiorri, C. (2020). Optimization of generalized s-curve trajectories for residual vibration suppression and compliance with kinematic bounds. *IEEE/ASME Transactions on Mechatronics*. Cited page 69.

Birglen, L., Gosselin, C., Pouliot, N., Monsarrat, B., and Laliberté, T. (2002). SHaDe, a new 3-DOF haptic device. *IEEE Transactions on Robotics and Automation*, 18(2):166–175. Cited page 37.

Bonev, I. (2001). Delta parallel robot - the story of success. http://www.parallemic.org/Reviews/Review002.html. Cited page 27.

Bonev, I. (2003). The True Origins of Parallel Robots. http://www.parallemic.org/Reviews/Review007.html. Cited pages 1, 23, 24, 25, and 26.

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. Cited page 44.

Briot, S., Khalil, W., et al. (2015). Dynamics of parallel robots. *From rigid bodies to flexible elements. Springer*. Cited page 44.

Briot, S. b., Baradat, C. d., Guégan, S., and Arakelian, V. (2007). Contribution to the mechanical behavior improvement of the robotic navigation device Surgiscope®. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 48094, pages 653–661. Cited page 35.

Bruckmann, T. and Pott, A. (2012). *Cable-driven parallel robots*, volume 12. Springer. Cited page 20.

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. Cited page 35.

Cao, C. and Hovakimyan, N. (2006a). Design and analysis of a novel $\mathscr{L}_1$ adaptive controller, part i: control signal and asymptotic stability. In *2006 American Control Conference*, pages 3397–3402. IEEE. Cited page 52.

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. Cited page 52.

Cappel, K. (1967). Motion simulator. https://patents.google.com/patent/US3295224A/en. Cited pages 1, 26, and 27.

Carabin, G., Scalera, L., Wongratanaphisan, T., and Vidoni, R. (2021). An energy-efficient approach for 3D printing with a Linear Delta Robot equipped with optimal springs. *Robotics and Computer-Integrated Manufacturing*, 67:102045. Cited page 37.

Castain, R. H. and Paul, R. P. (1984). An on-line dynamic trajectory generator. *The International Journal of Robotics Research*, 3(1):68–72. Cited page 68.

Chablat, D., Wenger, P., and Angeles, J. (2000). Conception isotropique d'une morphologie paralléle: Application à l'usinage. In *3rd International Conference On Integrated Design and Manufacturing in Mechanical Engineering*. Cited page 28.

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. Cited page 42.

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. Cited page 22.

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. Cited page 44.

Clavel, R. (1990). Device for the movement and positioning of an element in space. https://patents.google.com/patent/US4976582A. Cited pages 1, 26, and 28.

Codourey, A. (1998). Dynamic modeling of parallel robots for computed-torque control implementation. *The International Journal of Robotics Research*, 17(12):1325–1336. Cited pages 44 and 92.

Company, O., Pierrot, F., and Krut, S. (2013). Fast pick-and-place parallel robot with compact travelling plate. https://patents.google.com/patent/NL2010124C2/en. Cited page 30.

Corbel, D., Gouttefarde, M., Company, O., and Pierrot, F. (2010a). Actuation redundancy as a way to improve the acceleration capabilities of 3t and 3t1r pick-and-place parallel manipulators. *Journal of Mechanisms and Robotics*. Cited page 43.

Corbel, D., Gouttefarde, M., Company, O., and Pierrot, F. (2010b). 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. Cited page 30.

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. Cited pages 49 and 59.

Cunningham, D. and Asada, H. H. (2009). The winch-bot: A cable-suspended, underactuated robot utilizing parametric self-excitation. In *2009 IEEE International Conference on Robotics and Automation*, pages 1844–1850. IEEE. Cited page 22.

Dalvand, M. M. and Shirinzadeh, B. (2012). Remote centre-of-motion control algorithms of 6-RRCRR parallel robot assisted surgery system (PRAMiSS). In *2012 IEEE International Conference on Robotics and Automation*, pages 3401–3406. IEEE. Cited page 35.

Dasgupta, B. and Choudhury, P. (1999). A general strategy based on the Newton–Euler approach for the dynamic formulation of parallel manipulators. *Mechanism and machine theory*, 34(6):801–824. Cited page 44.

Dash, A. K., Chen, I.-M., Yeo, S. H., and Yang, G. (2005). Workspace generation and planning singularity-free path for parallel manipulators. *Mechanism and Machine Theory*, 40(7):776–805. Cited page 41.

De Miras, J., Join, C., Fliess, M., Riachy, S., and Bonnet, S. (2013). Active magnetic bearing: A new step for model-free control. In *52nd IEEE Conference on Decision and Control*, pages 7449–7454. IEEE. Cited page 130.

Dimension, F. (2004). Omega.6 haptic device from force dimension. https://www.forcedimension.com/products/omega. Cited page 37.

Dimension, F. (2010). Sigma.7 haptic device from force dimension. https://www.forcedimension.com/products/sigma. Cited page 37.

ecoRobotix (2011). The autonomous robot weeder from ecorobotix. https://ecorobotix.com/en/autonomous-scouting-robot/. Cited page 35.

Escorcia-Hernández, J. M., Aguilar-Sierra, H., Aguilar-Mejia, O., Chemori, A., and Arroyo-Núñez, J. H. (2020a). A new adaptive rise feedforward approach based on associative memory neural networks for the control of pkms. *Journal of Intelligent & Robotic Systems*, 100(3):827–847. Cited pages 54 and 113.

Escorcia-Hernandez, J. M., Chemori, A., and Aguilar-Sierra, H. (2022). Adaptive rise feedback control for robotized machining with pkms: Design and real-time experiments. *IEEE Transactions on Control Systems Technology*. Cited pages 120 and 123.

Escorcia-Hernández, J. M., Chemori, A., Aguilar-Sierra, H., and Monroy-Anieva, J. A. (2020b). A new solution for machining with ra-pkms: Modelling, control and experiments. *Mechanism and Machine Theory*, 150:103864. Cited pages 31, 32, 33, 113, and 152.

Fagiolini, A., Arisumi, H., and Bicchi, A. (2011). Casting robotic end-effectors to reach faraway moving objects. *Computing Research Repository (CoRR)*, abs/1101.2268. Cited page 62.

Fang, Y., Hu, J., Liu, W., Shao, Q., Qi, J., and Peng, Y. (2019). Smooth and time-optimal s-curve trajectory planning for automated robots and machines. *Mechanism and Machine Theory*, 137:127–153. Cited page 68.

Feemster, M. (2014). Jitter reduction in a directed energy application using rise. In *2014 American Control Conference*, pages 2451–2455. IEEE. Cited page 113.

Fischer, N., Bhasin, S., and Dixon, W. (2011). Nonlinear control of an autonomous underwater vehicle: A rise-based approach. In *Proceedings of the 2011 American Control Conference*, pages 3972–3977. IEEE. Cited page 113.

Fliess, M. (2009). Model-free control and intelligent pid controllers: towards a possible trivialization of nonlinear control? *IFAC Proceedings Volumes*, 42(10):1531–1550. Cited page 128.

Fliess, M. and Join, C. (2008). Intelligent pid controllers. In *2008 16th Mediterranean Conference on Control and Automation*, pages 326–331. IEEE. Cited page 128.

Fliess, M. and Join, C. (2013). Model-free control. *International Journal of Control*, 86(12):2228–2252. Cited page 129.

Fliess, M., Join, C., Mboup, M., and Sira-Ramirez, H. (2006). Vers une commande multivariable sans modèle. *arXiv preprint math/0603155*. Cited page 128.

Frank, H. (2009). Throwing of objects: A new technology for factory automation. In *2009 First International Conference on Computational Intelligence, Communication Systems and Networks*, pages 4–5. IEEE. Cited page 61.

Frank, H. (2010). Throwing of axial-symmetric objects in production systems. In *2010 12th International Conference on Computer Modelling and Simulation*, pages 1–1. IEEE. Cited page 61.

Gasparetto, A. and Zanotto, V. (2008). A technique for time-jerk optimal planning of robot trajectories. *Robotics and Computer-Integrated Manufacturing*, 24(3):415–426. Cited pages 41, 42, and 69.

Gédouin, P.-A., Delaleau, E., Bourgeot, J.-M., Join, C., Chirani, S. A., and Calloch, S. (2011). Experimental comparison of classical pid and model-free control: position control of a shape memory alloy active spring. *Control Engineering Practice*, 19(5):433–441. Cited pages 129 and 130.

Gédouin, P.-A., Join, C., Delaleau, E., Bourgeot, J.-M., Arbab-Chirani, S., and Calloch, S. (2009). A new control strategy for shape memory alloys actuators. In *8th European Symposium on Martensitic Transformations*, page CDROM. Cited page 129.

Germain, C. (2013). *Conception d'un robot parallèle à deux degrés de liberté pour des opérations de prise et de dépose*. PhD thesis, Ecole Centrale de Nantes (ECN). Cited page 19.

Germain, C., Briot, S. b., Glazunov, V., Caro, S. p., and Wenger, P. (2011). Irsbot-2: A novel two-dof parallel robot for high-speed operations. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 54839, pages 899–909. Cited page 30.

Gholami, P., Aref, M., and Taghirad, H. D. (2009). Adaptive cascade control of the kntu cdrpm: a cable driven redundant parallel manipulator. In *IEEE International Conference on Robotics and Automation*. Cited page 123.

Gosselin, C. (1996). Parallel computational algorithms for the kinematics and dynamics of planar and spatial parallel manipulators. Cited page 44.

Guo, H., Liu, Y., Liu, G., and Li, H. (2008). Cascade control of a hydraulically driven 6-DOF parallel robot manipulator based on a sliding mode. *Control Engineering Practice*, 16(9):1055–1068. Cited page 61.

Gwinnett, J. E. (1931). Amusement device. https://patents.google.com/patent/US1789680A. Cited pages 1, 23, and 24.

Hassan, G., Chemori, A., Chikh, L., Hervé, P.-E., El Rafei, M., Francis, C., and Pierrot, F. (2020). Rise feedback control of cable-driven parallel robots: design and real-time experiments. *IFAC-PapersOnLine*, 53(2):8519–8524. Cited page 113.

Hervé, J. (1991). Dispositif pour le déplacement en translation spatiale d'un élément dans l'espace en particulier pour robot mécanique. hhttps://patents.google.com/patent/EP0494565A1/fr. Cited page 27.

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. Cited page 58.

Hovakimyan, N. and Cao, C. (2010). *L1 adaptive control theory: Guaranteed robustness with fast adaptation*. SIAM. Cited page 53.

Hu, F., Liu, Q., Cheng, J., and He, Y. (2018). Flight trajectory simulation of robotic throwing shuttlecock. In *2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, pages 77–82. IEEE. Cited page 62.

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. Cited page 43.

Jafarinasab, M., Keshmiri, M., Azizan, H., and Danesh, M. (2011). Sliding mode control of a novel 6-DOF parallel manipulator with rotary actuators. In *2011 16th International Conference on Methods & Models in Automation & Robotics*, pages 218–223. IEEE. Cited page 60.

Jiang, F. and Gao, Z. (2001). An application of nonlinear pid control to a class of truck abs problems. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, volume 1, pages 516–521. IEEE. Cited page 52.

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. Cited page 53.

Kawai, Y. and Namerikawa, T. (2013). Passivity and rise based robust control for bilateral teleoperation system with communication delay. In *2013 IEEE International Conference on Control Applications (CCA)*, pages 270–275. IEEE. Cited page 113.

Kawamura, S., Kino, H., and Won, C. (2000). High-speed manipulation by using parallel wire-driven robots. *Robotica*, 18(1):13–21. Cited page 21.

Kelly, R., Davila, V. S., and Perez, J. A. L. (2006). *Control of robot manipulators in joint space*. Springer Science & Business Media. Cited pages 47 and 114.

Khalil, H. K. and Grizzle, J. W. (2002). *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ. Cited pages 120 and 127.

Khalil, W. and Dombre, E. (2004). *Modeling Identification and Control of Robots*. Butterworth-Heinemann. Cited page 77.

Krut, S. (2003). *Contribution à l'étude des robots parallèles légers, 3T-1R et 3T-2R, à forts débattements angulaires*. PhD thesis, Université Montpellier. Cited pages 18, 19, and 28.

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. Cited page 30.

Lamaury, J. and Gouttefarde, M. (2013). Control of a large redundantly actuated cable-suspended parallel robot. In *2013 IEEE International Conference on Robotics and Automation*, pages 4659–4664, Karlsruhe, Germany. IEEE. Cited pages 20 and 21.

Lewis, F. L., Dawson, D. M., and Abdallah, C. T. (2003). *Robot manipulator control: theory and practice*. CRC Press. Cited page 47.

Li, H. (2016). A jerk-constrained asymmetric motion profile for high-speed motion stages to reduce residual vibration. *International Journal of Computer Applications in Technology*, 53(2):149–156. Cited page 68.

Li, Y. and Wang, Y. (2009). Trajectory tracking control of a redundantly actuated parallel robot using diagonal recurrent neural network. In *2009 Fifth International Conference on Natural Computation*, volume 2, pages 292–296. IEEE. Cited page 53.

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. Cited page 35.

Liu, G., Wu, Y., Wu, X., Kuen, Y., and Li, Z. (2001). Analysis and control of redundant parallel manipulators. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 4, pages 3748–3754. IEEE. Cited page 49.

Lynch, K. M. and Mason, M. T. (1996). Dynamic underactuated nonprehensile manipulation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96*, volume 2, pages 889–896. IEEE. Cited page 62.

Lynch, K. M. and Mason, M. T. (1997). Dynamic manipulation with a one joint robot. In *Proceedings of International Conference on Robotics and Automation*, volume 1, pages 359–366. IEEE. Cited page 62.

Macfarlane, S. and Croft, E. A. (2003). Jerk-bounded manipulator trajectory planning: design for real-time applications. *IEEE Transactions on robotics and automation*, 19(1):42–52. Cited page 68.

Machinex (2017). SamurAI - Machinex Sorting Robot. https://www.machinexrecycling.com/products/samurai-sorting-robot/. Cited page 38.

Mboup, M., Join, C., and Fliess, M. (2009). Numerical differentiation with annihilators in noisy environment. *Numerical algorithms*, 50(4):439–467. Cited page 129.

Merlet, J.-P. (2005). *Parallel robots*, volume 128. Springer Science & Business Media. Cited pages 18, 31, 43, 44, and 84.

Michel, L., Join, C., Fliess, M., Sicard, P., and Chériti, A. (2010). Model-free control of dc/dc converters. In *2010 IEEE 12th workshop on control and modeling for power electronics (COMPEL)*, pages 1–8. IEEE. Cited page 129.

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. Cited pages 2 and 45.

Muller, A. (2005). Internal preload control of redundantly actuated parallel manipulators—its application to backlash avoiding control. *IEEE Transactions on Robotics*, 21(4):668–677. Cited page 43.

Muller, A. (2006). Stiffness control of redundantly actuated parallel manipulators. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1153–1158. IEEE. Cited page 43.

Müller, A. (2009). Consequences of geometric imperfections for the control of redundantly actuated parallel manipulators. *IEEE Transactions on Robotics*, 26(1):21–31. Cited page 43.

Nabat, V. (2007). *Robots parallèles à nacelle articulée, du concept à la solution industrielle pour le pick-andplace*. PhD thesis, Université Montpellier. Cited pages 30 and 92.

Nabat, V., O Rodriquez, 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. Cited pages 3, 95, 100, and 101.

Natal, G. S., Chemori, A., Michelin, M., and Pierrot, F. (2012). A dual-space feedforward pid control of redundantly actuated parallel manipulators with real-time experiments. In *PID'12: IFAC Conference on Advances in PID Control*, page 6. Cited page 56.

Natal, G. S., Chemori, A., and Pierrot, F. (2014). Dual-space control of extremely fast parallel manipulators: Payload changes and the 100g experiment. *IEEE Transactions on Control Systems Technology*, 23(4):1520–1535. Cited pages 30, 42, 51, 58, and 59.

Neumann, K. (2006). Exechon concept. *Parallel Kinematic Machines in Research and Practice*, 33:787–802. Cited page 23.

Nguyen, K. D., Ng, T.-C., and Chen, I.-M. (2008). On algorithms for planning s-curve motion profiles. *International Journal of Advanced Robotic Systems*, 5(1):11. Cited page 68.

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media. Cited pages 185, 205, and 209.

Okada, M., Pekarovskiy, A., and Buss, M. (2015). Robust trajectory design for object throwing based on sensitivity for model uncertainties. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3089–3094. IEEE. Cited page 62.

Ortega, R. and Spong, M. W. (1989). Adaptive motion control of rigid robots: A tutorial. *Automatica*, 25(6):877–888. Cited pages 49 and 57.

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. IEEE. Cited page 52.

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, page 012103. IOP Publishing. Cited page 35.

Paccot, F., Andreff, N., and Martinet, P. (2009). A review on the dynamic control of parallel kinematic machines: Theory and experiments. *The International Journal of Robotics Research*, 28(3):395–416. Cited page 49.

Paccot, F., Lemoine, P., Andreff, N., Chablat, D., and Martinet, P. (2008). A vision-based computed torque control for parallel kinematic machines. In *2008 IEEE International Conference on Robotics and Automation*, pages 1556–1561. IEEE. Cited page 55.

Paden, B. and Panja, R. (1988). Globally asymptotically stable 'PD+' controller for robot manipulators. *International Journal of Control*, 47(6):1697–1712. Cited page 55.

Pandilov, Z. and Dukovski, V. (2014). Comparison of the characteristics between serial and parallel robots. *Acta Technica Corviniensis-Bulletin of Engineering*, 7(1). Cited page 20.

Patel, Y., George, P., et al. (2012). Parallel manipulators applications—a survey. *Modern Mechanical Engineering*, 2(03):57. Cited pages 18 and 31.

Patre, P. M., MacKunis, W., Kaiser, K., and Dixon, W. E. (2008). Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and rise feedback control structure. *IEEE Transactions on Automatic Control*, 53(9):2180–2185. Cited page 112.

Paty, T., Binaud, N., Caro, S., and Segonds, S. (2021). Cable-driven parallel robot modelling considering pulley kinematics and cable elasticity. *Mechanism and Machine Theory*, 159:104263. Cited page 21.

Perumaal, S. and Jawahar, N. (2012). Synchronized trigonometric s-curve trajectory for jerk-bounded time-optimal pick and place operation. *International Journal of Robotics and Automation*, 27(4):385. Cited page 41.

Pham, D. and Yeo, S. (1991). Strategies for gripper design and selection in robotic assembly. *The International Journal of Production Research*, 29(2):303–316. Cited page 38.

Piazzi, A. and Visioli, A. (2000). Global minimum-jerk trajectory planning of robot manipulators. *IEEE transactions on industrial electronics*, 47(1):140–149. Cited page 68.

Pierrot, F., Baradat, C., Nabat, V., Company, O., Krut, S., and Gouttefarde, M. (2009a). 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. Cited page 30.

Pierrot, F. and Company, O. (1999). H4: a new family of 4-dof parallel robots. In *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No. 99TH8399)*, pages 508–513. IEEE. Cited page 28.

Pierrot, F., Dauchez, P., and Fournier, A. (1991). Hexa: a fast six-dof fully-parallel robot. In *Fifth International Conference on Advanced Robotics' Robots in Unstructured Environments*, pages 1158–1163. IEEE. Cited page 28.

Pierrot, F., Nabat, V., Company, O., Krut, S., and Poignet, P. (2009b). Optimal design of a 4-DOF parallel manipulator: From academia to industry. *IEEE Transactions on Robotics*, 25(2):213–224. Cited pages 30 and 32.

Pierrot, F., Reynaud, C., and Fournier, A. (1990). Delta: a simple and efficient parallel robot. *Robotica*, 8(2):105–109. Cited page 91.

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. Cited page 123.

Pollard, W. L. (1942). Position controlling apparatus. https://patents.google.com/patent/US2286571A. Cited pages 1, 24, and 25.

Pott, A. (2018). *Cable-Driven Parallel Robots: Theory and Application*. Springer International Publishing. Cited page 21.

Raptopoulos, F., Koskinopoulou, M., and Maniadakis, M. (2020). Robotic pick-and-toss facilitates urban waste sorting. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1149–1154. IEEE. Cited pages 3, 62, 67, 68, 95, 96, 97, 98, 99, 102, 103, 104, and 130.

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. Cited pages 51 and 114.

Reveles, D., Wenger, P., et al. (2016). Trajectory planning of kinematically redundant parallel manipulators by using multiple working modes. *Mechanism and Machine Theory*, 98:216–230. Cited page 41.

Rew, K.-H. and Kim, K.-S. (2009). A closed-form solution to asymmetric motion profile allowing acceleration manipulation. *IEEE Transactions on Industrial Electronics*, 57(7):2499–2506. Cited page 68.

ROBERT, G. and FLIESS, M. (2010). Model-free based water level control for hydroelectric power plants. *IFAC Proceedings Volumes*, 43(1):134–139. Cited page 129.

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. Cited pages 116, 119, 121, 122, and 126.

Saied, H. (2019). *On Control of Parallel Robots for High Dynamic Performances: From Design to Experiments*. PhD thesis, LIRMM, University of Montpellier. Cited pages 20, 31, 42, 92, and 141.

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. Cited page 113.

Saied, H., Chemori, A., El Rafei, M., and Francis, C. (2021). A Novel Model-Based Robust Super-Twisting Sliding Mode Control of PKMs: Design and Real-Time Experiments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8029–8035. IEEE. Cited pages 61 and 152.

Saied, H., Chemori, A., El Rafei, M., Francis, C., and Pierret, F. (2018). Actuator and friction dynamics formulation in control of pkms: From design to real-time experiments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5634–5639. IEEE. Cited page 56.

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. Cited page 58.

Santibañez, V. and Kelly, R. (2001). PD control with feedforward compensation for robot manipulators: analysis and experimentation. *Robotica*, 19(1):11–19. Cited page 56.

Santos, J. C. (2020). *Model Predictive Tracking Control of Cable-Driven Parallel Robots: From Concept to Real-Time Validation*. PhD thesis, Université Montpellier. Cited page 21.

Schmidt, V. and Pott, A. (2017). Increase of position accuracy for cable-driven parallel robots using a model for elongation of plastic fiber ropes. In *New trends in mechanism and machine science*, pages 335–343. Springer. Cited page 21.

Seraji, H. (1998). A new class of nonlinear PID controllers with robotic applications. *Journal of Robotic Systems*, 15(3):161–181. Cited page 52.

Shang, W. and Cong, S. (2009). Nonlinear computed torque control for a high-speed planar parallel manipulator. *Mechatronics*, 19(6):987–992. Cited page 57.

Shang, W. and Cong, S. (2010). Nonlinear adaptive task space control for a 2-dof redundantly actuated parallel manipulator. *Nonlinear Dynamics*, 59(1):61–72. Cited pages 49, 50, and 59.

Shang, W.-W., Cong, S., and Ge, Y. (2012). Adaptive computed torque control for a parallel manipulator with redundant actuation. *Robotica*, 30(3):457–466. Cited page 59.

Shang, W. W., Cong, S., Li, Z. X., and Jiang, S. L. (2009). Augmented nonlinear PD controller for a redundantly actuated parallel manipulator. *Advanced Robotics*, 23(12-13):1725–1742. Cited page 57.

Shayya, S. A. (2015). *Towards rapid and precise parallel kinematic machines*. PhD thesis, Université Montpellier. Cited pages 18, 19, 30, and 32.

Sherwani, K. I., Kumar, N., Chemori, A., Khan, M., and Mohammed, S. (2020). Rise-based adaptive control for eicosi exoskeleton to assist knee joint mobility. *Robotics and Autonomous Systems*, 124:103354. Cited page 113.

Shtessel, Y., Edwards, C., Fridman, L., Levant, A., et al. (2014). *Sliding mode control and observation*, volume 10. Springer. Cited page 60.

Siciliano, B. and Khatib, O. (2016). *Springer handbook of robotics*. Springer. Cited pages 42, 46, and 49.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Force control*. Springer. Cited pages 40 and 41.

Singh, Y. and Santhakumar, M. (2015). Inverse dynamics and robust sliding mode control of a planar parallel (2-PRP and 1-PPR) robot augmented with a nonlinear disturbance observer. *Mechanism and Machine Theory*, 92:29–50. Cited page 61.

Sintov, A. and Shapiro, A. (2015). A stochastic dynamic motion planning algorithm for object-throwing. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2475–2480. IEEE. Cited page 62.

Slotine, J.-J. and Weiping, L. (1988). Adaptive manipulator control: A case study. *IEEE transactions on automatic control*, 33(11):995–1003. Cited page 115.

Slotine, J.-J. E. and Li, W. (1987). On the adaptive control of robot manipulators. *The international journal of robotics research*, 6(3):49–59. Cited page 115.

Staicu, S. (2019). *Dynamics of Parallel Robots*. Springer. Cited page 45.

Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the institution of mechanical engineers*, 180(1):371–386. Cited pages 1, 25, and 27.

Su, Y., Duan, B., and Zheng, C. (2004). Nonlinear PID control of a six-DOF parallel manipulator. *IEE Proceedings-Control Theory and Applications*, 151(1):95–102. Cited page 52.

Symetrie (2017). Wave basin. https://symetrie.fr/en/applications/naval/wave-basin/. Cited page 33.

Taghirad, H. D. (2013). *Parallel robots: mechanics and control*. CRC press. Cited pages 18, 19, 26, 44, 46, 47, 50, 54, 83, and 89.

Taktak-Meziou, M., Chemori, A., Ghommam, J., and Derbel, N. (2014). A prediction-based optimal gain selection in rise feedback control for hard disk drive. In *2014 IEEE Conference on Control Applications (CCA)*, pages 2114–2119. IEEE. Cited page 113.

Tijjani, A. S., Chemori, A., and Creuze, V. (2020). Robust adaptive tracking control of underwater vehicles: Design, stability analysis and experiments. *IEEE/ASME Transactions on Mechatronics*. Cited page 123.

Toyama, T., Yamakawa, Y., and Suzuki, H. (1998). Machine tool having parallel structure. https://patents.google.com/patent/US5715729A/en. Cited pages 32 and 33.

Tsai, L.-W. (2000). Solving the inverse dynamics of a Stewart-Gough manipulator by the principle of virtual work. *J. Mech. Des.*, 122(1):3–9. Cited page 44.

Tsukagoshi, H., Watari, E., Fuchigami, K., and Kitagawa, A. (2012). Casting device for search and rescue aiming higher and faster access in disaster site. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4348–4353. IEEE. Cited page 62.

Wang, H., Mustafa, G. I., and Tian, Y. (2018). Model-free fractional-order sliding mode control for an active vehicle suspension system. *Advances in Engineering Software*, 115:452–461. Cited pages 129 and 130.

Wang, J. and Gosselin, C. m. M. (2004). Kinematic analysis and design of kinematically redundant parallel mechanisms. *J. Mech. Des.*, 126(1):109–118. Cited page 43.

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. Cited page 35.

Weber, X., Cuvillon, L., and Gangloff, J. (2015). Active vibration canceling of a cable-driven parallel robot in modal space. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1599–1604. IEEE. Cited page 21.

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. Cited pages 110, 111, 113, 118, 119, 133, and 134.

Yao, J., Jiao, Z., and Ma, D. (2014). Rise-based precision motion control of dc motors with continuous friction compensation. *IEEE Transactions on Industrial Electronics*, 61(12):7067–7075. Cited page 113.

Yao, R., Tang, X., Wang, J., and Huang, P. (2009). Dimensional optimization design of the four-cable-driven parallel manipulator in fast. *IEEE/ASME Transactions On Mechatronics*, 15(6):932–941. Cited page 22.

Zeng, A., Song, S., Lee, J., Rodriguez, A., and Funkhouser, T. (2020). Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4):1307–1319. Cited page 62.

Zhang, X. and Ming, Z. (2019). Trajectory planning and optimization for a par4 parallel robot based on energy consumption. *Applied Sciences*, 9(13):2770. Cited page 42.

Zhang, Y., Jiang, T., and Jiao, J. (2020a). Model-free predictive current control of a dfig using an ultra-local model for grid synchronization and power regulation. *IEEE Transactions on Energy Conversion*, 35(4):2269–2280. Cited page 130.

Zhang, Y., Jin, J., and Huang, L. (2020b). Model-free predictive current control of pmsm drives based on extended state observer using ultralocal model. *IEEE Transactions on Industrial Electronics*, 68(2):993–1003. Cited pages 129 and 130.

Zhang, Y.-X., Cong, S., Shang, W.-W., Li, Z.-X., and Jiang, S.-L. (2007). Modeling, identification and control of a redundant planar 2-DOF parallel manipulator. *International Journal of Control, Automation, and Systems*, 5(5):559–569. Cited page 55.

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. Cited page 44.

Ziegler, J. G., Nichols, N. B., et al. (1942). Optimum settings for automatic controllers. *trans. ASME*, 64(11). Cited page 51.

APPENDIX

# A Proof of Algorithm 1

The details of this appendix were written by Dr. Marc Gouttefarde. The KKT first-order necessary conditions [Nocedal and Wright, 2006] for $x^* = (x_1^*, x_2^*)$ to be a local solution to the optimization problem (2.6) can be stated as follows.

*First-order necessary conditions:* There is a Lagrange multiplier vector $\lambda^*$, with components $\lambda_i^*$, $1 \leq i \leq 6$, such that the following conditions are satisfied:

$$\nabla_x \mathscr{L}(x^*, \lambda^*) = 0 \tag{A.1}$$
$$C_i(x^*) \geq 0, \quad 1 \leq i \leq 6 \tag{A.2}$$
$$\lambda_i^* \geq 0, \quad 1 \leq i \leq 6 \tag{A.3}$$
$$\lambda_i^* C_i(x^*) = 0, \quad 1 \leq i \leq 6 \tag{A.4}$$

where the constraints $C_i(x)$ are defined in (2.6) and $\nabla_x \mathscr{L}(x, \lambda)$ is the gradient with respect to $x$ of the Lagrangian function $\mathscr{L}(x, \lambda)$ defined as:

$$\mathscr{L}(x, \lambda) = f(x) - \sum_{i=1}^{6} \lambda_i C_i(x) \tag{A.5}$$

The first-order necessary conditions stated above in (A.1) to (A.4) are valid if the functions $f(x)$ and $C_i(x)$ are continuously differentiable and the so-called Linear Independence Constraint Qualification (LICQ) holds at $x^*$ [Nocedal and Wright, 2006]. According to the definitions of $f(x)$ and $C_i(x)$, the continuous differentiability condition is true for problem (2.6). The fact that the LICQ holds is proved in Section A.16.
The well-known method to use the first-order necessary conditions to find local optimal solutions $x^*$ consists in distinguishing all the possible cases of sets of active constraints

which allows the determination of $x^*$ and of the Lagrange multiplier vector $\lambda^*$.
A constraint $C_i(x)$ is defined as being active if $C_i(x) = 0$. According to (A.4), for a given $i$, either $C_i(x^*)$ is active, i.e. $C_i(x) = 0$, or $\lambda_i^* = 0$. Since $C_5(x) = x_1 = V > 0$ and $C_6(x) = x_2 = A > 0$, $C_5(x)$ and $C_6(x)$ are never active, and according to (A.4), $\lambda_5 = 0$ and $\lambda_6 = 0$ at a local optimal solution. Then, only four constraints can be active, $C_1(x)$ to $C_4(x)$, and the problem (2.6) having two variables, $x_1$ and $x_2$, the following 11 cases of possible active constraint sets must be studied:

- No active constraint (one case)
- Four cases of one active constraint
- Six cases of two active constraints

These relatively small number of possible active constraint sets and the rather simple expressions of the functions $f(x)$ and $C_i(x)$ make the analysis of each of these eleven cases possible, as detailed in the following subsections.

## A.1 Preliminaries: Expressions of the gradients

In (A.1), the gradient with respect to $x$ of the Lagrangian function $\mathscr{L}(x,\lambda)$ in (A.5) is given by:

$$\nabla_x \mathscr{L}(x,\lambda) = \nabla_x f(x) - \sum_{i=1}^{4} \lambda_i \nabla_x C_i(x) \tag{A.6}$$

where the summation is taken for $i = 1$ to 4 since, as pointed above, $\lambda_5 = 0$ and $\lambda_6 = 0$ at a local optimal solution. According to (2.5) and (2.6), the gradients in (A.6) are as follows:

$$\nabla_x f(x) = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{x_2} - \dfrac{P}{x_1^2} \\ \dfrac{1}{J} - \dfrac{x_1}{x_2^2} \end{bmatrix} = \begin{bmatrix} \dfrac{x_1^2 - Px_2}{x_1^2 x_2} \\ \dfrac{x_2^2 - Jx_1}{Jx_2^2} \end{bmatrix} \tag{A.7}$$

$$\nabla_x C_1(x) = \begin{bmatrix} \dfrac{\partial C_1}{\partial x_1} \\ \dfrac{\partial C_1}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{x_2} \\ -\dfrac{1}{J} - \dfrac{x_1}{x_2^2} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{x_2} \\ \dfrac{-x_2^2 - Jx_1}{Jx_2^2} \end{bmatrix} \tag{A.8}$$

$$\nabla_x C_2(x) = \begin{bmatrix} \frac{-P}{x_1^2} - \frac{1}{x_2} \\ \frac{x_1}{x_2^2} - \frac{1}{J} \end{bmatrix} = \begin{bmatrix} \frac{-Px_2 - x_1^2}{x_1^2 x_2} \\ \frac{Jx_1 - x_2^2}{Jx_2^2} \end{bmatrix} \tag{A.9}$$

$$\nabla_x C_3(x) = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \tag{A.10}$$

$$\nabla_x C_4(x) = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tag{A.11}$$

## A.2 Case 1: No active constraint

According to (A.4), $\lambda_i^* = 0$ for $i = 1 \dots 4$ if there is no active constraint at a local optimal solution $x^*$. Then, (A.1) yields:

$$\nabla_x \mathscr{L}(x^*, \lambda^*) = \nabla_x f(x^*) = 0 \Longleftrightarrow \begin{cases} x_1^{*2} - Px_2^* = 0 \\ x_2^{*2} - Jx_1^* = 0 \end{cases} \Longleftrightarrow \begin{cases} x_1^{*3} = JP^2 \\ x_2^{*3} = J^2 P \end{cases} \tag{A.12}$$

However, the constraint $C_2(x)$ is:

$$C_2(x) = \frac{P}{x_1} - \frac{x_1}{x_2} - \frac{x_2}{J} = \frac{J\left(Px_2 - x_1^2\right) - x_1 x_2^2}{Jx_1 x_2} \tag{A.13}$$

so that, with (A.12), the fact that $C_2(x)$ is not active leads to:

$$C_2(x^*) > 0 \Longleftrightarrow J\left(Px_2^* - x_1^{*2}\right) - x_1^* x_2^{*2} > 0 \Longleftrightarrow -x_1^* x_2^{*2} > 0 \tag{A.14}$$

where the second equivalence comes from $Px_2^* - x_1^{*2} = 0$ which is a consequence of (A.12). The inequality (A.14) is impossible since $x_1^* > 0$ and $x_2^* > 0$ according to constraints $C_5$ and $C_6$. Consequently, this first case of no active constraint is not feasible and is discarded.

## A.3 Case 2: One active constraint $C_1(x) = 0$

According to (A.4), we have $\lambda_i^* = 0$ for $i = 2, \dots, 4$. Hence, $x_1^*$, $x_2^*$ and $\lambda_1^*$ can be determined from the following equation system obtained from (A.1) and $C_1(x^*) = 0$:

$$\begin{cases} \nabla_x \mathscr{L}(x^*, \lambda^*) = \nabla_x f(x^*) - \lambda_1^* \nabla_x C_1(x^*) = 0 \\ C_1(x^*) = \frac{x_1^*}{x_2^*} - \frac{x_2^*}{J} = 0 \end{cases} \tag{A.15}$$

From (A.7) and (A.8), the system (A.15) is equivalent to:

$$\begin{cases} x_1^{*2} - Px_2^* - \lambda_1^* x_1^{*2} = 0 \\ x_2^{*2} - Jx_1^* + \lambda_1^* \left(x_2^{*2} + Jx_1^*\right) = 0 \\ Jx_1^* - x_2^{*2} = 0 \end{cases} \tag{A.16}$$

This second equation system yields $\lambda_1^* = 0$, $x_1^{*3} = JP^2$ and $x_2^{*3} = J^2P$. From these values of $x_1^*$ and $x_2^*$, it follows that, similarly to Case 1 detailed in the previous subsection, the constraint $C_2$ is not feasible which discards Case 2.

## A.4 Case 3: One active constraint $C_2(x) = 0$

When the only active constraint is $C_2$, (A.4) gives $\lambda_i^* = 0$ for all $i \neq 2$, and $x_1^*$, $x_2^*$ and $\lambda_2^*$ can then be determined from:

$$\begin{cases} \nabla_x \mathscr{L}(x^*, \lambda^*) = \nabla_x f(x^*) - \lambda_2^* \nabla_x C_2(x^*) = 0 \\ C_2(x^*) = \dfrac{J\left(Px_2^* - x_1^{*2}\right) - x_1^* x_2^{*2}}{Jx_1^* x_2^*} = 0 \end{cases} \tag{A.17}$$

which, from (A.7) and (A.9), is equivalent to:

$$\begin{cases} x_1^{*2} - Px_2^* + \lambda_2^* \left(Px_2^* + x_1^{*2}\right) = 0 \\ x_2^{*2} - Jx_1^* - \lambda_2^* \left(Jx_1^* - x_2^{*2}\right) = 0 \\ J\left(Px_2^* - x_1^{*2}\right) - x_1^* x_2^{*2} = 0 \end{cases} \tag{A.18}$$

The second equation of this system can be written $\left(x_2^{*2} - Jx_1^*\right)(1 + \lambda_2^*) = 0$ which implies $x_2^{*2} = Jx_1^*$ since $\lambda_2^* \geq 0$ according to (A.3). Then, the third equation of (A.18) gives $x_2^{*3} = \left(J^2P\right)/2$, which in turn leads to $x_1^{*3} = \left(JP^2\right)/4$ and $\lambda_2^* = 1/2$.
Since $\lambda_2^* = 1/2 > 0$ and $\lambda_i^* = 0$ for all $i \neq 2$, (A.3) is true. For the first order necessary conditions to be all true, it remains to verify that (A.2) is satisfied for $i = 1, 2$ and 3, i.e., $C_1(x^*) \geq 0$, $C_3(x^*) \geq 0$ and $C_4(x^*) \geq 0$.
$C_1(x^*) \geq 0$ is equivalent to $Jx_1^* - x_2^{*2} \geq 0$ which is true since $x_2^{*2} = Jx_1^*$. Finally, since $x_1^{*3} = \left(JP^2\right)/4$ and $x_2^{*3} = \left(J^2P\right)/2$, $C_3(x^*) \geq 0$ and $C_4(x^*) \geq 0$ yield the two following conditions on J, P, $x_{1max} = V_{max}$ and $x_{2max} = A_{max}$:

$$x_1^* \leq x_{1max} \Longleftrightarrow JP^2 \leq 4V_{max}^3 \tag{A.19}$$

$$x_2^* \leq x_{2max} \Longleftrightarrow J^2P \leq 2A_{max}^3 \tag{A.20}$$

To conclude this third case, if the two conditions (A.19) and (A.20) are verified, the following vector is a local minimum candidate:

$$x^* = \begin{bmatrix} \sqrt[3]{\frac{JP^2}{4}} \\ \sqrt[3]{\frac{J^2P}{2}} \end{bmatrix} \tag{A.21}$$

## A.5 Case 4: One active constraint $C_3(x)=0$

When the only active constraint is $C_3$, (A.4) gives $\lambda_i^* = 0$ for all $i \neq 3$ and $C_3(x^*) = x_{1max} - x_1^* = 0$ gives $x_1^* = x_{1max} = V_{max}$. Moreover, (A.1) is $\nabla_x f(x^*) - \lambda_3^* \nabla_x C_3(x^*) = 0$, i.e.:

$$\begin{cases} x_1^{*2} - Px_2^* + \lambda_3^* x_1^{*2} x_2 = 0 \\ x_2^{*2} = Jx_1^* \end{cases} \tag{A.22}$$

With $x_1^* = V_{max}$ and the second equation of (A.22), after some calculations, the first equation of (A.22) leads to:

$$\lambda_3^* = \frac{P\sqrt{J} - \sqrt{V_{max}}^3}{V_{max}^2\sqrt{J}} \tag{A.23}$$

Then, since $\lambda_3^* \geq 0$ according to (A.3), the following condition must hold:

$$\sqrt{V_{max}}^3 \leq P\sqrt{J} \iff V_{max}^3 \leq JP^2 \tag{A.24}$$

For the first order necessary conditions to be true, it remains to verify that (A.2) is satisfied for $i = 1$, 2 and 4, i.e., $C_1(x^*) \geq 0$, $C_2(x^*) \geq 0$ and $C_4(x^*) \geq 0$. First, since $x_2^{*2} = Jx_1^*$, we have $C_1(x^*) = 0$ so that $C_1(x^*) \geq 0$, and $C_2(x^*) \geq 0$ is equivalent to $x_1^{*3} \leq JP^2/4$, i.e., from $x_1^* = V_{max}$, the following condition is obtained:

$$V_{max}^3 \leq \frac{JP^2}{4} \tag{A.25}$$

Note that condition (A.25) implies condition (A.24) so that only (A.25) is to be retained. Finally, $C_4(x^*) \geq 0$, $x_2^{*2} = Jx_1^*$ and $x_1^* = V_{max}$ yields the following condition:

$$\sqrt{JV_{max}} \leq A_{max} \tag{A.26}$$

To summarize this fourth case, if the two conditions (A.25) and (A.26) are verified, the following vector is a local minimum candidate:

$$x^* = \begin{bmatrix} V_{max} \\ \sqrt{JV_{max}} \end{bmatrix} \tag{A.27}$$

## A.6 Case 5: One active constraint $C_4(x) = 0$

When the only active constraint is $C_4$, (A.4) gives $\lambda_i^* = 0$ for all $i \neq 4$ and $C_4(x^*) = x_{2max} - x_2^* = 0$ gives $x_2^* = x_{2max} = A_{max}$. Moreover, (A.1) is $\nabla_x f(x^*) - \lambda_4^* \nabla_x C_4(x^*) = 0$ which is a system of two equations. The first equation of this system yields $x_1^{*2} = P x_2^*$. Then, according to (A.13), $C_2(x^*) \geq 0$ is equivalent to $-x_1^* x_2^{*2} \geq 0$ which is impossible since $x_1^* > 0$ and $x_2^* > 0$. In conclusion, this fifth case is not feasible and is thus discarded.

## A.7 Case 6: Two active constraints $C_1(x) = C_2(x) = 0$

When only $C_1$ and $C_2$ are active, $C_3(x^*) > 0$ and $C_4(x^*) > 0$ imply with (A.4) that $\lambda_3^* = 0$ and $\lambda_4^* = 0$. Then, $x_1^*$, $x_2^*$, $\lambda_1^*$ and $\lambda_2^*$ have to be determined. The determination of $\lambda_1^*$ and $\lambda_2^*$ shall be done to verify that (A.3) is satisfied.
First, the $x_1^*$ and $x_2^*$ are calculated from the following system of two equations obtained from $C_1(x^*) = C_2(x^*) = 0$:

$$\begin{cases} J x_1^* - x_2^{*2} = 0 \\ J\left(P x_2^* - x_1^{*2}\right) - x_1^* x_2^{*2} = 0 \end{cases} \tag{A.28}$$

whose solution is:

$$x^* = \begin{bmatrix} \sqrt[3]{\frac{JP^2}{4}} \\ \sqrt[3]{\frac{J^2P}{2}} \end{bmatrix} \tag{A.29}$$

Then, Eq. (A.1) with $\lambda_3^* = 0$ and $\lambda_4^* = 0$ is used to determine $\lambda_1^*$ and $\lambda_2^*$:

$$\begin{aligned} & \nabla_x f(x^*) - \lambda_1^* \nabla_x C_1(x^*) - \lambda_2^* \nabla_x C_2(x^*) = 0 \\ \iff & \begin{cases} x_1^{*2} - P x_2^* - \lambda_1^* x_1^{*2} + \lambda_2^* \left(P x_2^* + x_1^{*2}\right) = 0 \\ x_2^{*2} - J x_1^* + \lambda_1^* \left(x_2^{*2} + J x_1^*\right) - \lambda_2^* \left(J x_1^* - x_2^{*2}\right) = 0 \end{cases} \\ \iff & \begin{bmatrix} -x_1^{*2} & P x_2^* + x_1^{*2} \\ x_2^{*2} + J x_1^* & x_2^{*2} - J x_1^* \end{bmatrix} \begin{bmatrix} \lambda_1^* \\ \lambda_2^* \end{bmatrix} = \begin{bmatrix} P x_2^* - x_1^{*2} \\ J x_1^* - x_2^{*2} \end{bmatrix} \end{aligned}$$

With the expressions of $x^*$ in (A.29), the determinant of the matrix in the last equation is equal to $\left(-3J^2P^2\right)/2$. and the equation system possesses a unique solution. Once solved, e.g. with Cramer's rule, it yields $\lambda_1^* = 0$ and $\lambda_2^* = 1/3$ which verify (A.3).
It remains to verify (A.2), i.e., $C_3(x^*) \geq 0$ and $C_4(x^*) \geq 0$ which leads to the two following conditions:

$$JP^2 \leq 4V_{max}^3 \quad \text{and} \quad J^2P \leq 2A_{max}^3 \tag{A.30}$$

To conclude on this sixth case, if conditions (A.30) are true, $x^*$ in (A.29) is a local minimum candidate. Moreover, it appears that Case 6 is equivalent to Case 3 since $x^*$ in (A.29) is the same as $x^*$ in (A.21) and conditions (A.30) are the same as those in (A.19) and (A.20) (which is a consequence of $x^*$ being the same).

## A.8 Case 7: Two active constraints $C_1(x) = C_3(x) = 0$

$C_1(x^*) = 0$ and $C_3(x^*) = 0$ give:

$$x^* = \begin{bmatrix} V_{max} \\ \sqrt{JV_{max}} \end{bmatrix} \tag{A.31}$$

Moreover, $C_2(x^*) > 0$ and $C_4(x^*) > 0$ imply with (A.4) that $\lambda_2^* = 0$ and $\lambda_4^* = 0$, and (A.1) yields the following equation system in $\lambda_1^*$ and $\lambda_3^*$

$$\nabla_x f(x^*) - \lambda_1^* \nabla_x C_1(x^*) - \lambda_3^* \nabla_x C_3(x^*) = 0 \Longleftrightarrow \begin{cases} x_1^{*2} - Px_2^* - \lambda_1^* x_1^{*2} + \lambda_3^* \left(x_1^{*2} x_2^*\right) = 0 \\ x_2^{*2} - Jx_1^* + \lambda_1^* \left(x_2^{*2} + Jx_1^*\right) = 0 \end{cases}$$

With (A.31), the second equation yields $\lambda_1^* = 0$ and then the first equation gives $\lambda_3^* = \left(Px_2^* - x_1^{*2}\right) / \left(x_1^{*2} x_2^*\right)$. With (A.31), for (A.3) to be true, i.e. $\lambda_3^* \geq 0$, the following condition must hold:

$$V_{max}^3 \leq JP^2 \tag{A.32}$$

It remains to verify that (A.2) is satisfied for $i = 2$ and $i = 4$, i.e., $C_2(x^*) \geq 0$ and $C_4(x^*) \geq 0$, which leads to the two following conditions:

$$V_{max}^3 \leq \frac{JP^2}{4} \quad \text{and} \quad \sqrt{JV_{max}} \leq A_{max} \tag{A.33}$$

where the first condition in (A.33) is stronger than (A.32), i.e., (A.33) implies (A.32).
To conclude on case 7, if conditions (A.33) are true, $x^*$ in (A.31) is a local minimum candidate. Moreover, it turns out that Case 7 is equivalent to Case 4 since $x^*$ in (A.31) is the same as $x^*$ in (A.27) and conditions (A.33) are the same as those in (A.25) and (A.26) (which is a consequence of $x^*$ being the same).

## A.9 Case 8: Two active constraints $C_1(x) = C_4(x) = 0$

$C_1(x^*) = 0$ and $C_4(x^*) = 0$ give:

$$x^* = \begin{bmatrix} \frac{A_{max}^2}{J} \\ A_{max} \end{bmatrix} \tag{A.34}$$

Moreover, $C_2(x^*) > 0$ and $C_3(x^*) > 0$ imply with (A.4) that $\lambda_2^* = 0$ and $\lambda_3^* = 0$, and (A.1) yields the following equation system in $\lambda_1^*$ and $\lambda_4^*$:

$$\nabla_x f(x^*) - \lambda_1^* \nabla_x C_1(x^*) - \lambda_4^* \nabla_x C_4(x^*) = 0 \Longleftrightarrow \begin{cases} x_1^{*2} - P x_2^* - \lambda_1^* x_1^{*2} = 0 \\ x_2^{*2} - J x_1^* + \lambda_1^* \left( x_2^{*2} + J x_1^* \right) + \lambda_4^* J x_2^{*2} = 0 \end{cases}$$

From the first equation and (A.34), we have:

$$\lambda_1^* = \frac{x_1^{*2} - P x_2^*}{x_1^{*2}} = 1 - \frac{J^2 P}{A_{max}^3} \tag{A.35}$$

and, from the second equation:

$$\lambda_4^* = \frac{J x_1^* - x_2^{*2} - \lambda_1^* \left( x_2^{*2} + J x_1^* \right)}{J x_2^{*2}} = -\frac{2}{J} \lambda_1^* \tag{A.36}$$

since, according to (A.34), $J x_1^* - x_2^{*2} = 0$ and $x_2^{*2} + J x_1^* = 2 x_2^{*2}$. Eq. (A.36) and (A.3) imply that $\lambda_1^* = \lambda_4^* = 0$ so that (A.35) gives:

$$J^2 P = A_{max}^3 \tag{A.37}$$

However, $C_2(x^*) \geq 0$, which must hold true according to (A.2), leads to:

$$J \left( P x_2^* - x_1^{*2} \right) - x_1^* x_2^{*2} \geq 0 \Longrightarrow J P A_{max} - 2 \frac{A_{max}^4}{J} \geq 0 \Longrightarrow \frac{J^2 P}{2} \geq A_{max}^3$$

which is impossible in view of (A.37). In conclusion, Case 8 is impossible and thus discarded.

## A.10 Case 9: Two active constraints $C_2(x) = C_3(x) = 0$

$C_3(x^*) = 0$ means that $x_1^* = V_{max}$ while $C_2(x^*) = 0$ yields a second-order polynomial equation in $x_2^*$. The latter equation can be solved to obtain two solutions for $x_2^*$, but it is better to first consider (A.1). Since, with (A.4), $C_1(x^*) > 0$ and $C_4(x^*) > 0$ imply that $\lambda_1^* = 0$ and $\lambda_4^* = 0$, (A.1) yields:

$$\Longleftrightarrow \quad \begin{array}{c} \nabla_x f(x^*) - \lambda_2^* \nabla_x C_2(x^*) - \lambda_3^* \nabla_x C_3(x^*) = 0 \\ \begin{cases} x_1^{*2} - P x_2^* + \lambda_2^* \left( x_1^{*2} + P x_2^* \right) + \lambda_3^* \left( x_1^{*2} x_2^* \right) = 0 \\ x_2^{*2} - J x_1^* + \lambda_2^* \left( x_2^{*2} - J x_1^* \right) = 0 \end{cases} \end{array} \tag{A.38}$$

The second equation implies that either $x_2^{*2} = Jx_1^*$ or $\lambda_2^* = -1$. Since $\lambda_2^* \geq 0$ according to (A.3), $\lambda_2^* = -1$ is not possible so that $x_2^{*2} = Jx_1^*$, i.e., $x_2^* = \sqrt{JV_{max}}$. Then, with $x_1^* = V_{max}$ and $x_2^* = \sqrt{JV_{max}}$, $C_2(x^*) = 0$ is equivalent to:

$$V_{max}^3 = \frac{JP^2}{4} \tag{A.39}$$

Going back to the equation system (A.38), note that in the second equation, since $x_2^{*2} = Jx_1^*$, $\lambda_2^*$ is undetermined so that any $\lambda_2^* \geq 0$ satisfies this equation and (A.3) as well. Now, let us verify if there exist $\lambda_2^* \geq 0$ and $\lambda_3^* \geq 0$ such that the first equation of (A.38) is verified. With $x_1^* = V_{max}$ and $x_2^* = \sqrt{JV_{max}}$, this first equation is equivalent to:

$$\lambda_3^* = \frac{P\sqrt{JV_{max}} - V_{max}^2 - \lambda_2^* \left(P\sqrt{JV_{max}} + V_{max}^2\right)}{V_{max}^2\sqrt{JV_{max}}} \tag{A.40}$$

so that $\lambda_3^* \geq 0$ if and only if:

$$\lambda_2^* \leq \frac{P\sqrt{JV_{max}} - V_{max}^2}{P\sqrt{JV_{max}} + V_{max}^2} \tag{A.41}$$

Since $\lambda_2^*$ must be non-negative, (A.41) is possible if and only if:

$$P\sqrt{JV_{max}} - V_{max}^2 \geq 0 \Longleftrightarrow JP^2 \geq V_{max}^3 \tag{A.42}$$

which, according to (A.39), is true. Hence, taking $\lambda_2^*$ equal to the right-hand side of the inequality (A.41), we have $\lambda_2^* \geq 0$ and $\lambda_3^*$ in (A.40) is also non-negative which proves that there exist $\lambda_2^* \geq 0$ and $\lambda_3^* \geq 0$ such that (A.38) is verified.
Finally, the inequalities $C_1(x^*) \geq 0$ and $C_4(x^*) \geq 0$ should be verified for (A.2) to be true. First, we have $C_1(x^*) = 0$ since $x_2^{*2} = Jx_1^*$. Second, $C_4(x^*) \geq 0$ is $x_2^* \leq A_{max}$ which yields:

$$\sqrt{JV_{max}} \leq A_{max} \tag{A.43}$$

To conclude Case 9, if conditions (A.39) and (A.43) are verified, the following vector is a local minimum candidate:

$$x^* = \begin{bmatrix} V_{max} \\ \sqrt{JV_{max}} \end{bmatrix} \tag{A.44}$$

Comparing (A.39) with (A.25), (A.43) with (A.26) and (A.44) with (A.27), Case 9 appears to be a particular case of Case 4 where (A.25) is verified as an equality.

## A.11 Case 10: Two active constraints $C_2(x) = C_4(x) = 0$

$C_4(x^*) = x_{2max} - x_2^* = 0$ gives $x_2^* = x_{2max} = A_{max}$ while $C_2(x^*) = 0$ yields the following quadratic equation in $x_1^*$:

$$Jx_1^{*2} + x_2^{*2}x_1^* - JPx_2^* = 0 \tag{A.45}$$

The discriminant of this equation is:

$$\Delta = x_2^* \left( x_2^{*3} + 4J^2P \right) > 0 \tag{A.46}$$

and (A.45) possesses the following two solutions:

$$x_1^* = \frac{-x_2^{*2} \pm \sqrt{\Delta}}{2J} \tag{A.47}$$

Since $x_1^* > 0$ according to (2.6), the only possible solution is:

$$x_1^* = \frac{-x_2^{*2} + \sqrt{\Delta}}{2J} = \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2 P A_{max}}}{2J} \tag{A.48}$$

since the other one is negative. Note that $x_1^*$ in (A.48) is positive because $-x_2^{*2} + \sqrt{\Delta} > 0$, i.e., $x_2^{*4} < \Delta$ which can be deduced from (A.46) and $4J^2P > 0$.
Now, let us determine the Lagrange multipliers $\lambda_2^*$ and $\lambda_4^*$ and establish the conditions for $\lambda_2^*$ and $\lambda_4^*$ to be non-negative and thus to satisfy (A.3). Since, with (A.4), $C_1(x^*) > 0$ and $C_3(x^*) > 0$ imply that $\lambda_1^* = 0$ and $\lambda_3^* = 0$, (A.1) yields:

$$\begin{aligned} &\nabla_x f(x^*) - \lambda_2^* \nabla_x C_2(x^*) - \lambda_4^* \nabla_x C_4(x^*) = 0 \\ &\qquad \Longleftrightarrow \quad \begin{cases} x_1^{*2} - Px_2^* + \lambda_2^* \left( x_1^{*2} + Px_2^* \right) = 0 \\ Jx_1^* - x_2^{*2} + \lambda_2^* \left( Jx_1^* - x_2^{*2} \right) = \lambda_4^* \end{cases} \end{aligned} \tag{A.49}$$

Since $x_1^{*2} + Px_2^* > 0$, we have from the first equation:

$$\lambda_2^* = \frac{Px_2^* - x_1^{*2}}{Px_2^* + x_1^{*2}} \tag{A.50}$$

With the expression of $x_1^*$ in (A.48), one can verify that $Px_2^* - x_1^{*2} \geq 0$ is equivalent to $\Delta \geq x_2^{*4}$ which is true according to (A.46) and $4J^2P > 0$. Consequently, $\lambda_2^*$ in (A.50) is non-negative. Besides, from the second equation of (A.49), we have:

$$\lambda_4^* = \left( Jx_1^* - x_2^{*2} \right) \left( 1 + \lambda_2^* \right) \tag{A.51}$$

Since $\lambda_2^* \geq 0$, $\lambda_4^* \geq 0$ is equivalent to $Jx_1^* - x_2^{*2} \geq 0$. With (A.48) and (A.46), the latter inequality is equivalent to $x_2^{*3} \leq \frac{J^2P}{2}$. Hence, $\lambda_4^* \geq 0$ leads to the following condition:

$$A_{max}^3 \leq \frac{J^2P}{2} \tag{A.52}$$

Finally, the inequalities $C_1(x^*) \geq 0$ and $C_3(x^*) \geq 0$ should be verified for (A.2) to be true. $C_1(x^*) \geq 0$ is equivalent to $Jx_1^* - x_2^{*2} \geq 0$ i.e. to $\lambda_4^* \geq 0$ which is verified if (A.52) is true. $C_3(x^*) \geq 0$ is:

$$x_1^* = \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2 P A_{max}}}{2J} \leq V_{max} \tag{A.53}$$

To conclude Case 10, if conditions (A.52) and (A.53) are verified, the following vector is a local minimum candidate:

$$x^* = \begin{bmatrix} \dfrac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2 P A_{max}}}{2J} \\ A_{max} \end{bmatrix} \tag{A.54}$$

## A.12 Case 11: Two active constraints $C_3(x) = C_4(x) = 0$

In this last case, the two active constraints $C_3(x) = C_4(x) = 0$ yield directly the following local minimum candidate:

$$x^* = \begin{bmatrix} V_{max} \\ A_{max} \end{bmatrix} \tag{A.55}$$

Since, with (A.4), $C_1(x^*) > 0$ and $C_2(x^*) > 0$ imply that $\lambda_1^* = 0$ and $\lambda_2^* = 0$, (A.1) gives:

$$\nabla_x f(x^*) - \lambda_3^* \nabla_x C_3(x^*) - \lambda_4^* \nabla_x C_4(x^*) = 0 \Longleftrightarrow \begin{cases} \lambda_3^* = \dfrac{P x_2^* - x_1^{*2}}{x_1^{*2} x_2^*} \\ \lambda_4^* = \dfrac{J x_1^* - x_2^{*2}}{J x_2^{*2}} \end{cases}$$

so that, with (A.55), the following equality should hold for (A.3) to be true:

$$\lambda_3^* \geq 0 \Longleftrightarrow P x_2^* - x_1^{*2} \geq 0 \Longleftrightarrow P A_{max} \geq V_{max}^2 \tag{A.56}$$

and also:

$$\lambda_4^* \geq 0 \Longleftrightarrow J x_1^* - x_2^{*2} \geq 0 \Longleftrightarrow J V_{max} \geq A_{max}^2 \tag{A.57}$$

Finally, the inequalities $C_1(x^*) \geq 0$ and $C_2(x^*) \geq 0$ should be verified for (A.2) to hold. $C_1(x^*) \geq 0$ is equivalent to $Jx_1^* - x_2^{*2} \geq 0$, i.e., to (A.57). $C_2(x^*) \geq 0$ is equivalent to:

$$J\left(P x_2^* - x_1^{*2}\right) - x_1^* x_2^{*2} \geq 0 \Longleftrightarrow J P A_{max} - J V_{max}^2 \geq V_{max} A_{max}^2 \tag{A.58}$$

In summary, if the inequalities (A.56), (A.57) and (A.58) are satisfied, $x^*$ in (A.55) is a local minimum candidate.

## A.13 Synthesis of the 11 cases and Algorithm 2

Let us now summarize all the 11 cases:

- ases 1, 2, 5 and 8 are impossible in the sense that they do not verify the KKT first-order necessary conditions.
- Case 6 is equivalent to Case 3 and Case 7 is equivalent to Case 4.
- Case 9 is a particular case of Case 4.

Hence, only four cases, namely Cases 3, 4, 10 and 11, need to be considered to find (local) solutions $x^* = (x_1^*, x_2^*)$ to the optimization problem (2.6).
In order to ease the analysis of the relationships between these cases, the conditions to be fulfilled for each one of these four cases are summarized below.

*Conditions for Case 3*:

$$JP^2 \leq 4V_{max}^3 \tag{A.59}$$

$$J^2P \leq 2A_{max}^3 \tag{A.60}$$

*Conditions for Case 4*:

$$4V_{max}^3 \leq JP^2 \tag{A.61}$$

$$JV_{max} \leq A_{max}^2 \tag{A.62}$$

*Conditions for Case 10*:

$$2A_{max}^3 \leq J^2P \tag{A.63}$$

$$JPA_{max} - JV_{max}^2 \leq V_{max}A_{max}^2 \tag{A.64}$$

where (A.64) is equivalent (A.53) as can be shown with some elementary calculations. Moreover, again after some calculations, it turns out that (A.63) and (A.64) imply that:

$$A_{max}^2 \leq JV_{max} \tag{A.65}$$

*Conditions for Case 11*:

$$V_{max}^2 \leq PA_{max} \tag{A.66}$$

$$A_{max}^2 \leq JV_{max} \tag{A.67}$$

$$V_{max}A_{max}^2 \leq JPA_{max} - JV_{max}^2 \tag{A.68}$$

Note that, from (A.67) and (A.68), we have:

$$2A_{max}^3 \leq J^2P \tag{A.69}$$

Indeed, the sum of (A.69) and (A.67) multiplied by $V_{max}$ gives $2V_{max}A_{max} \leq JP$. Using the latter inequality and multiplying (A.67) by $2A_{max}$ lead to (A.69).
Carefully analyzing all these conditions leads to the following relationships between the four remaining cases (Cases 3, 4, 10 and 11) and in turn to Algorithm 1.
Let us first assume that condition (A.60) of Case 3 is satisfied as a strict inequality. Then, Cases 10 and 11 are not possible because of (A.63) and (A.69), respectively, and there are two possibles cases:

- If (A.59) of Case 3 is verified (as a strict inequality), Case 4 is not possible because of (A.61). The only possible case is Case 3 which means that the sole local minimum candidate is $x^*$ in (A.21) which corresponds to line 3 of Algorithm 1.
- If (A.59) of Case 3 is not verified, Case 3 is not possible and condition (A.61) of Case 4 is verified. Case 4 is then the only possible case provided that condition (A.62) is true. It turns out that (A.60) and (A.61) imply that condition (A.62) is verified. Indeed, (A.60) is equivalent to:

  $$\sqrt[3]{\frac{J^4P^2}{4}} \leq A_{max}^2 \tag{A.70}$$

  (A.61) is equivalent to:

  $$\sqrt[3]{\frac{4}{JP^2}} \leq \frac{1}{V_{max}} \tag{A.71}$$

  and (A.70) and (A.71) imply that:

  $$\sqrt[3]{J^3} \leq \frac{A_{max}^2}{V_{max}} \Longleftrightarrow JV_{max} \leq A_{max}^2 \tag{A.72}$$

  which is (A.62). Hence, Case 4 is the only possible case meaning that the sole local minimum candidate is $x^*$ in (A.27) which corresponds to line 5 of Algorithm 1.

Let us now assume that condition (A.60) of Case 3 is not satisfied. Case 3 is then not possible and:

- If (A.62) is satisfied (as a strict inequality), Cases 10 and 11 are not possible because of (A.65) and (A.67), respectively. Moreover, since (A.60) is not satisfied, we have:

  $$2A_{max}^3 < J^2P \Longleftrightarrow A_{max} < \sqrt[3]{\frac{J^2P}{2}} \tag{A.73}$$

  and using the latter in (A.62) (satisfied as a strict inequality) yields:

  $$J < \frac{A_{max}^2}{V_{max}} < \frac{1}{V_{max}}\sqrt[3]{\frac{J^4P^2}{4}} \tag{A.74}$$

which implies that:

$$V_{max} < \sqrt[3]{\frac{JP^2}{4}} \Longleftrightarrow V_{max}^3 < \frac{JP^2}{4} \tag{A.75}$$

i.e. (A.61) is satisfied. The two conditions for Case 4 are then satisfied and Case 4 is the only possible case meaning that the unique local minimum candidate is $x^*$ in (A.27) which corresponds to line 9 of Algorithm 1.

- If (A.62) is not satisfied, Case 4 is not possible and the only possible cases are Cases 10 and 11. Let us then consider the two following complementary situations:
  - — If (A.64) is satisfied (as a strict inequality), Case 11 is not possible because of (A.68). Case 10 is possible since (A.63) is true because (A.60) is not satisfied. Case 10 is then the only possible case and the unique local minimum candidate is $x^*$ in (A.54) which corresponds to lines 11 and 12 of Algorithm 1.
  - — If (A.64) is not satisfied, Case 10 is not possible but Case 11 is then feasible. Indeed (A.68) is true since (A.64) is not satisfied, (A.67) is true since (A.62) is not satisfied, and (A.66) is true since it turns out to be implied by (A.68). The latter result comes from the fact that (A.68) is equivalent to:

    $$V_{max}A_{max}^2 + JV_{max}^2 \leq JPA_{max} \tag{A.76}$$

    which implies that:

    $$JV_{max}^2 \leq JPA_{max} \Longleftrightarrow V_{max}^2 \leq PA_{max} \tag{A.77}$$

    the latter inequality being (A.66). Hence, Case 11 is the only possible case and the unique local minimum candidate is $x^*$ in (A.55) which corresponds to lines 11 and 14 of Algorithm 1.

Algorithm 2 summarizes the above analysis. It is exactly the same as Algorithm 1 which proves that the latter is correct. Note that line 12 of Algorithm 2 corresponds to lines 11 and 12 of Algorithm 1 as can be seen from the expression of $x^*$ in (A.54). Moreover, line 14 of Algorithm 2 corresponds to lines 11 and 14 of Algorithm 1 since, with $V = \frac{-A_{max}^2+\sqrt{A_{max}^4+4J^2PA_{max}}}{2J}$ as calculated at line 12 of Algorithm 1, the condition $V > V_{max}$ at line 13 of Algorithm 1 means that (A.53) is not satisfied. Hence, since (A.53) and (A.64) are equivalent (as can be shown with some elementary calculations), (A.64) is also not satisfied which corresponds to the condition for line 14 of Algorithm 2.
It is important to note that, according to the above analysis, Algorithm 1 and Algorithm 2 computes in fact the global minimum of the minimization problem (2.6).

## A.14 Particular cases

For the proof to be complete, it remains to be shown that Algorithm 2, and hence Algorithm 1, is correct in a number of particular cases. The latter were overlooked at var-

**Algorithm 2** Minimum-Time S-Curve Trajectory Rewritten

```
if (A.60) then
  if (A.59) then
    x* in (A.21) {Case 3}
  else
    x* in (A.27) {Case 4}
  end if
else
  if (A.62) then
    x* in (A.27) {Case 4}
  else
    if (A.64) then
      x* in (A.54) {Case 10}
    else
      x* in (A.55) {Case 11}
    end if
  end if
end if
```

ious places in the proof presented in Section A.13 where conditions (A.59), (A.60), (A.62) and (A.64) were supposed to be verified as strict inequalities. In fact, the particular cases to be considered to complete the proof are the following ones:

- (A.60) and (A.59) satisfied as equalities;
- (A.60) satisfied as an equality and (A.59) satisfied as a strict inequality;
- (A.60) satisfied as an equality and (A.59) not satisfied;
- (A.60) not satisfied and (A.62) satisfied as an equality;
- (A.60) and (A.62) not satisfied and (A.64) satisfied as an equality.

*First particular case:* (A.60) *and* (A.59) *satisfied as equalities*

$$J^2P = 2A_{max}^3 \tag{A.78}$$

$$JP^2 = 4V_{max}^3 \tag{A.79}$$

These two equalities imply that (A.21) (line 3 of Algorithm 2) is:

$$x^* = \begin{bmatrix} \sqrt[3]{\frac{JP^2}{4}} \\ \sqrt[3]{\frac{J^2P}{2}} \end{bmatrix} = \begin{bmatrix} V_{max} \\ A_{max} \end{bmatrix} \tag{A.80}$$

Now, let us consider $x^*$ in (A.27) (line 5 of Algorithm 2):

$$x^* = \begin{bmatrix} V_{max} \\ \sqrt{JV_{max}} \end{bmatrix} = \begin{bmatrix} V_{max} \\ A_{max} \end{bmatrix} \tag{A.81}$$

To show that the second equality in the previous equation holds, consider $P = \frac{2A_{max}^3}{J^2}$ from (A.78) and $P^2 = \frac{4V_{max}^3}{J}$ from (A.79). Then, we have:

$$\frac{4A_{max}^6}{J^4} = \frac{4V_{max}^3}{J} \iff A_{max} = \sqrt{JV_{max}} \tag{A.82}$$

Next, (A.54) (line 12 of Algorithm 2) is:

$$x^* = \begin{bmatrix} \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J} \\ A_{max} \end{bmatrix} = \begin{bmatrix} V_{max} \\ A_{max} \end{bmatrix} \tag{A.83}$$

since, with $J^2P = 2A_{max}^3$ in (A.78):

$$\begin{aligned} \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J} &= \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 8A_{max}^4}}{2J} \\ &= \frac{-A_{max}^2 + \sqrt{9A_{max}^4}}{2J} = \frac{A_{max}^2}{J} = V_{max} \end{aligned} \tag{A.84}$$

where the last equality is obtained from (A.82).
Hence, in this first particular case, it turns out that $x^*$ in (A.21), (A.27) and (A.54) are all equal to $x^*$ in (A.55), i.e, all possible local minima are equal. Then, Algorithm 2 necessarily computes the optimal value $x^* = [V_{max}, A_{max}]^T$ at line 3.

*Second particular case:* (A.60) *satisfied as an equality and* (A.59) *satisfied as a strict inequality*

$$J^2P = 2A_{max}^3 \tag{A.85}$$

$$JP^2 < 4V_{max}^3 \tag{A.86}$$

First, note that Case 3 is possible since both (A.60) and (A.59) are satisfied and Case 4 is not possible since (A.61) is not satisfied. Moreover, Algorithm 2 computes $x^*$ at line 3 according to (A.21):

$$x^* = \begin{bmatrix} \sqrt[3]{\frac{JP^2}{4}} \\ \sqrt[3]{\frac{J^2P}{2}} \end{bmatrix} = \begin{bmatrix} \sqrt[3]{\frac{JP^2}{4}} \\ A_{max} \end{bmatrix} \tag{A.87}$$

where $\sqrt[3]{\frac{JP^2}{4}} < V_{max}$ since $JP^2 < 4V_{max}^3$.

Now, let us check whether or not Cases 10 and 11 are possible and, if they are, verify that $x^*$ in (A.87) also corresponds to their minimum.

The first condition for Case 10 to be possible is (A.63). This condition is verified since $J^2P = 2A_{max}^3$. The second condition for Case 10 is (A.64) which can equivalently be written as:

$$V_{max}A_{max}^2 + JV_{max}^2 \geq JPA_{max} \tag{A.88}$$

Since $V_{max} > \sqrt[3]{\frac{JP^2}{4}}$ and $J^2P = 2A_{max}^3$, one can write:

$$\begin{aligned} V_{max}A_{max}^2 + JV_{max}^2 > A_{max}^2\sqrt[3]{\frac{JP^2}{4}} + J\sqrt[3]{\frac{J^2P^4}{16}} &= \sqrt[3]{\frac{J^4P^2}{4}}\sqrt[3]{\frac{JP^2}{4}} + J\sqrt[3]{\frac{J^2P^4}{16}} \\ &= \sqrt[3]{\frac{J^5P^4}{16}} + \sqrt[3]{\frac{J^5P^4}{16}}2\sqrt[3]{\frac{J^5P^4}{16}} = \sqrt[3]{\frac{J^5P^4}{2}} = JP\sqrt[3]{\frac{J^2P}{2}} \end{aligned}$$

Hence, $V_{max}A_{max}^2 + JV_{max}^2 > JP\sqrt[3]{\frac{J^2P}{2}} = JPA_{max}$, i.e., (A.88) is verified as a strict inequality so that (A.64) is also verified as a strict inequality and Case 10 is possible. Furthermore, the minimum $x^*$ in Case 10 is given in (A.54):

$$x^* = \begin{bmatrix} \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J} \\ A_{max} \end{bmatrix} = \begin{bmatrix} \frac{A_{max}^2}{J} \\ A_{max} \end{bmatrix} \tag{A.89}$$

where the second equality is from (A.84). Since $J^2P = 2A_{max}^3$, we have:

$$\sqrt[3]{\frac{JP^2}{4}} = \sqrt[3]{\frac{J}{4}\frac{4A_{max}^6}{J^4}} = \frac{A_{max}^2}{J} \tag{A.90}$$

so that the minimum $x^*$ of Case 10 is equal to the minimum $x^*$ of Case 3 given in (A.87).
Finally, Case 11 is not possible because (A.64) is verified as a strict inequality and, hence, (A.68) is not verified.
Summarizing, only Cases 3 and 10 are possible, the minimum $x^*$ of these two cases are equal and Algorithm 2 indeed computes this $x^*$ at line 3.

*Third particular case:* (A.60) *satisfied as an equality and* (A.59) *not satisfied*

$$J^2P = 2A_{max}^3 \tag{A.91}$$

$$JP^2 > 4V_{max}^3 \tag{A.92}$$

In this particular case, Case 3 is not possible since (A.59) is not satisfied and Algorithm 2 computes $x^*$ at line 5 according to (A.27):

$$x^* = \begin{bmatrix} V_{max} \\ \sqrt{JV_{max}} \end{bmatrix} \tag{A.93}$$

First, note that Case 4 is possible since (A.61) is true since $J^2P = 2A_{max}^3$. Moreover, (A.62) is also verified since:

$$J^2P = 2A_{max}^3 \iff A_{max}^2 = \sqrt[3]{\frac{J^4P^2}{4}} = J\sqrt[3]{\frac{JP^2}{4}} \tag{A.94}$$

and:

$$JP^2 > 4V_{max}^3 \iff V_{max} < \sqrt[3]{\frac{JP^2}{4}} \tag{A.95}$$

imply that:

$$JV_{max} < J\sqrt[3]{\frac{JP^2}{4}} = A_{max}^2 \tag{A.96}$$

Furthermore, let us examine Cases 10 and 11. Case 10 is not feasible since (A.64) is not verified. Indeed, in the previous paricular case, it was shown that $J^2P = 2A_{max}^3$ and $JP^2 < 4V_{max}^3$ leads to (A.64) being verified as a strict inequality, i.e., $JPA_{max} - JV_{max}^2 < V_{max}A_{max}^2$. Then, the same reasoning allows to conclude that $J^2P = 2A_{max}^3$ and $JP^2 > 4V_{max}^3$ imply $JPA_{max} - JV_{max}^2 > V_{max}A_{max}^2$ i.e. (A.64) is not verified in the present particular case. Besides, Case 11 is also not feasible since $A_{max}^2 > JV_{max}$ from (A.96) so that (A.67) is not true.
Hence, only Case 4 is possible and the corresponding minimum $x^*$ is indeed computed at line 5 of Algorithm 2.

*Fourth particular case:* (A.60) *not satisfied and* (A.62) *satisfied as an equality*

$$J^2P > 2A_{max}^3 \tag{A.97}$$

$$JV_{max} = A_{max}^2 \tag{A.98}$$

First, note that Case 3 is not possible since (A.60) is not satisfied and that Algorithm 2 computes $x^*$ at line 9 according to (A.27):

$$x^* = \begin{bmatrix} V_{max} \\ \sqrt{JV_{max}} \end{bmatrix} = \begin{bmatrix} V_{max} \\ A_{max} \end{bmatrix} \tag{A.99}$$

which is also the minimum $x^*$ of Case 11 as given in (A.55) and where the second equality in (A.99) comes from (A.98). Then, let us check whether or not Cases 4, 10 and 11 are possible.

Case 4 is possible since (A.62) is truce according to (A.98). Moreover, (A.97) and (A.98) imply that:

$$4V_{max}^3 = 4\frac{A_{max}^6}{J^3} < \frac{4}{J^3}\frac{J^4P^2}{4} = JP^2 \tag{A.100}$$

so that (A.61) is satisfied.

Case 10 is not possible since (A.64) is not satisfied. Indeed, with (A.98), we have:

$$V_{max}A_{max}^2 + JV_{max}^2 = \frac{2A_{max}^4}{J} \tag{A.101}$$

and (A.97) then implies that:

$$\frac{2A_{max}^4}{J} < J^2P\frac{A_{max}}{J} = JPA_{max} \tag{A.102}$$

so that:

$$V_{max}A_{max}^2 + JV_{max}^2 < JPA_{max} \iff JPA_{max} - JV_{max}^2 > V_{max}A_{max}^2 \tag{A.103}$$

Case 11 turns out to be possible since (A.103) shows that (A.68) is true and (A.67) is satisfied according to (A.98). Moreover, (A.97) and (A.98) imply that:

$$V_{max}^2 = \frac{A_{max}^4}{J^2} < \frac{A_{max}}{J^2}\frac{J^2P}{2} = \frac{A_{max}P}{2} < PA_{max} \tag{A.104}$$

i.e. (A.66) is satisfied.

In conclusion of this fourth particular case, only Cases 4 and 11 are possible, their minimum $x^*$ are equal and indeed computed by Algorithm 2 at line 9.

*Fifth particular case:* (A.60) *and* (A.62) *not satisfied and* (A.64) *satisfied as an equality*

$$J^2P > 2A_{max}^3 \tag{A.105}$$

$$JV_{max} > A_{max}^2 \tag{A.106}$$

$$JPA_{max} - JV_{max}^2 = V_{max}A_{max}^2 \tag{A.107}$$

Cases 3 and 4 are then not possible since (A.60) and (A.62) are not satisfied, respectively. Case 10 is possible since (A.105) implies (A.63) and (A.64) is true according to (A.107). Then, about the feasibility of Case 11, it is not straightforward to verify whether or not (A.66) is true from (A.105), (A.106) and (A.107). Fortunately, this verification is in fact not required. Indeed, in the present particular case, Algorithm 2 computes $x^*$ of Case 10 at line 12 and, as proved below, this $x^*$ is equal to the one of Case 11, $x^* = [V_{max}, A_{max}]^T$. Hence, if Case 11 is not possible, Algorithm 2 computes $x^*$ of Case 10 which is the only possible minimum and, if Case 11 is possible, $x^*$ of Cases 10 and 11 are equal and Algorithm 2 indeed computes this minimum at line 12.
It remains to prove that the $x^*$ of Case 10, which according to (A.54) is:

$$x^* = \begin{bmatrix} \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J} \\ A_{max} \end{bmatrix} \tag{A.108}$$

is equal to the one of Case 11 which is $x^* = [V_{max}, A_{max}]^T$, i.e., to prove that

$$\frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J} = V_{max} \tag{A.109}$$

To this end, let us consider (A.107) as a quadratic equation in $V_{max}$:

$$JV_{max}^2 + A_{max}^2 V_{max} - JPA_{max} = 0 \tag{A.110}$$

whose two solutions are:

$$V_{max} = \frac{-A_{max}^2 \pm \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J} \tag{A.111}$$

The only possible solution for $V_{max}$ among those two is the following one since the other one is negative:

$$V_{max} = \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J} \tag{A.112}$$

which shows that $x^*$ of Case 10 in (A.108) is equal to $x^*$ of Case 11.

## A.15 Second-Order Sufficient Conditions

Algorithm 2 (and thus Algorithm 1) is based on the KKT first-order necessary conditions. Hence, the vectors $x^*$ computed in the various cases in Algorithm 2 are local minima candidates and it remains to be verified that these vectors $x^*$ are indeed local minima. To this end, the following second-order sufficient conditions can be used [Nocedal and Wright, 2006].

*Second-Order Sufficient Conditions:* Suppose that at some feasible vector $x^*$, there exists a Lagrange multiplier vector $\lambda^*$ satisfying the KKT conditions and that the Lagrangian Hessian $\nabla^2_{xx}\mathscr{L}(x^*,\lambda^*)$ is positive definite. Then, $x^*$ is a strict local minimum to the optimization problem (2.6).
From (A.6), the Lagrangian Hessian is:

$$\nabla^2_{xx}\mathscr{L}(x,\lambda) = \nabla^2_{xx}f(x) - \sum_{i=1}^{4}\lambda_i\nabla^2_{xx}C_i(x) \tag{A.113}$$

The individual Hessian matrices appearing in the above expression of $\nabla^2_{xx}\mathscr{L}$ are obtained from (A.7) to (A.11) as follows.

$$\nabla^2_{xx}f(x) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} \dfrac{2P}{x_1^3} & -\dfrac{1}{x_2^2} \\ -\dfrac{1}{x_2^2} & \dfrac{2x_1}{x_2^3} \end{bmatrix} \tag{A.114}$$

$$\nabla^2_{xx}C_1(x) = \begin{bmatrix} 0 & -\dfrac{1}{x_2^2} \\ -\dfrac{1}{x_2^2} & \dfrac{2x_1}{x_2^3} \end{bmatrix} \tag{A.115}$$

$$\nabla^2_{xx}C_2(x) = \begin{bmatrix} \dfrac{2P}{x_1^3} & \dfrac{1}{x_2^2} \\ \dfrac{1}{x_2^2} & -\dfrac{2x_1}{x_2^3} \end{bmatrix} \tag{A.116}$$

$$\nabla^2_{xx}C_3(x) = 0 \tag{A.117}$$

$$\nabla^2_{xx} C_4(x) = 0 \tag{A.118}$$

Therefore, the Lagrangian Hessian is:

$$\nabla^2_{xx}\mathscr{L}(x,\lambda) = \begin{bmatrix} (1-\lambda_2)\dfrac{2P}{x_1^3} & (\lambda_1-\lambda_2-1)\dfrac{1}{x_2^2} \\ (\lambda_1-\lambda_2-1)\dfrac{1}{x_2^2} & (1-\lambda_1+\lambda_2)\dfrac{2x_1}{x_2^3} \end{bmatrix} \tag{A.119}$$

Let us now verify the second-order sufficient conditions at the local minimum candidate $x^*$ of Case 3 in (A.21), Case 4 in (A.27), Case 10 in (A.54) and Case 11 in (A.55).
In Case 3, we have $\lambda_1^* = 0$, $\lambda_2^* = 1/2$, $x_2^{*2} = Jx_1^*$ and:

$$x^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} \sqrt[3]{\dfrac{JP^2}{4}} \\ \sqrt[3]{\dfrac{J^2P}{2}} \end{bmatrix} \tag{A.120}$$

so that:

$$\nabla^2_{xx}\mathscr{L}(x^*,\lambda^*) = \begin{bmatrix} \dfrac{P}{x_1^{*3}} & -\dfrac{3}{2x_2^{*2}} \\ -\dfrac{3}{2x_2^{*2}} & \dfrac{3x_1^*}{x_2^{*3}} \end{bmatrix} \tag{A.121}$$

According to Sylvester's criterion, since $\dfrac{P}{x_1^{*3}} > 0$, the Hessian $\nabla^2_{xx}\mathscr{L}(x^*,\lambda^*)$ is positive definite if and only if its determinant is strictly positive. Being given that $x_2^{*2} = Jx_1^*$ and with (A.120), the determinant is:

$$\begin{aligned} \det\left(\nabla^2_{xx}\mathscr{L}(x^*,\lambda^*)\right) &= \frac{3P}{x_1^{*2}x_2^{*3}} - \frac{9}{4x_2^{*4}} && \text{(A.122)} \\ &= \frac{3}{x_2^{*4}}\left(\frac{J^2P}{x_2^{*3}} - \frac{3}{4}\right) && \text{(A.123)} \\ &= \frac{3}{x_2^{*4}}\left(\frac{2J^2P}{J^2P} - \frac{3}{4}\right) && \text{(A.124)} \\ &= \frac{3}{x_2^{*4}}\frac{5}{4} > 0 && \text{(A.125)} \end{aligned}$$

which is positive since $x_2^* = \sqrt[3]{\frac{J^2P}{2}} > 0$. Hence, the second-order sufficient conditions are satisfied and $x^*$ of Case 3, given in (A.21) and computed at line 3 of Algorithm 2, is indeed a strict local minimum.
In Case 4, we have $\lambda_1^* = \lambda_2^* = 0$, $x_2^{*2} = Jx_1^*$ and:

$$x^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} V_{max} \\ \sqrt{JV_{max}} \end{bmatrix} \tag{A.126}$$

so that:

$$\nabla^2_{xx}\mathscr{L}(x^*,\lambda^*) = \begin{bmatrix} \frac{2P}{x_1^{*3}} & -\frac{1}{x_2^{*2}} \\ -\frac{1}{x_2^{*2}} & \frac{2x_1^*}{x_2^{*3}} \end{bmatrix} \tag{A.127}$$

whose determinant is:

$$\begin{aligned} \det\left(\nabla^2_{xx}\mathscr{L}(x^*,\lambda^*)\right) &= \frac{4P}{x_1^{*2}x_2^{*3}} - \frac{1}{x_2^{*4}} && \text{(A.128)} \\ &= \frac{4J^2P}{x_2^{*4}x_2^{*3}} - \frac{1}{x_2^{*4}} && \text{(A.129)} \\ &= \frac{1}{x_2^{*4}}\left(\frac{4J^2P}{x_2^{*3}} - 1\right) && \text{(A.130)} \\ &= \frac{1}{x_2^{*4}}\left(\frac{4J^2P}{\sqrt{J^3V_{max}^3}} - 1\right) && \text{(A.131)} \end{aligned}$$

Since $x_2^* > 0$, this determinant is positive if $4J^2P > \sqrt{J^3V_{max}^3}$ and hence if $16JP^2 > V_{max}^3$. The latter inequality is true since it is a consequence of (A.61):

$$JP^2 \geq 4V_{max}^3 \iff 16JP^2 \geq 64V_{max}^3 \tag{A.132}$$

Hence, $\det\left(\nabla^2_{xx}\mathscr{L}(x^*,\lambda^*)\right) > 0$ and since $\frac{2P}{x_1^{*3}} > 0$, Sylvester's criterion implies that $\nabla^2_{xx}\mathscr{L}(x^*,\lambda^*)$ is positive definite. The second-order sufficient conditions are thus satisfied and $x^*$ of Case 4, given in (A.27) and computed at lines 5 and 9 of Algorithm 2, is a strict local minimum.
In Case 10, we have $\lambda_1^* = 0$ and:

$$\lambda_2^* = \frac{Px_2^* - x_1^{*2}}{Px_2^* + x_1^{*2}} \tag{A.133}$$

$$x^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2 P A_{max}}}{2J} \\ A_{max} \end{bmatrix} \tag{A.134}$$

With $\lambda_1^* = 0$, the Hessian in (A.119) becomes:

$$\nabla_{xx}^2 \mathscr{L}(x^*, \lambda^*) = \begin{bmatrix} (1-\lambda_2^*)\frac{2P}{x_1^{*3}} & -(1+\lambda_2^*)\frac{1}{x_2^{*2}} \\ -(1+\lambda_2^*)\frac{1}{x_2^{*2}} & (1+\lambda_2^*)\frac{2x_1^*}{x_2^{*3}} \end{bmatrix} \tag{A.135}$$

and with the expression of $\lambda_2^*$ in (A.133):

$$\nabla_{xx}^2 \mathscr{L}(x^*, \lambda^*) = \frac{1}{Px_2^* + x_1^{*2}} \begin{bmatrix} \frac{4P}{x_1^*} & -\frac{2P}{x_2^*} \\ -\frac{2P}{x_2^*} & \frac{4Px_1^*}{x_2^{*2}} \end{bmatrix} \tag{A.136}$$

Referring to Section A.11, $x_1^* > 0$ and $x_2^* > 0$ so that $Px_2^* + x_1^{*2} > 0$ and $\frac{4P}{x_1^*} > 0$ hold. Then, according to Sylvester's criterion, $\nabla_{xx}^2 \mathscr{L}(x^*, \lambda^*)$ is positive definite if and only if the determinant of the matrix in (A.136) is positive. This determinant is positive since it is calculated as follows:

$$\begin{vmatrix} \frac{4P}{x_1^*} & -\frac{2P}{x_2^*} \\ -\frac{2P}{x_2^*} & \frac{4Px_1^*}{x_2^{*2}} \end{vmatrix} = \frac{16P^2}{x_2^{*2}} - \frac{4P^2}{x_2^{*2}} \tag{A.137}$$

$$= \frac{12P^2}{x_2^{*2}} \tag{A.138}$$

Hence, the second-order sufficient conditions are satisfied and $x^*$ of Case 10, given in (A.54) and computed at line 12 of Algorithm 2, is a strict local minimum.
Finally, in Case 11, we have $\lambda_1^* = 0$ and $\lambda_2^* = 0$ and:

$$x^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} V_{max} \\ A_{max} \end{bmatrix} \tag{A.139}$$

so that:

$$\nabla^2_{xx}\mathscr{L}(x^*,\lambda^*) = \begin{bmatrix} \frac{2P}{x_1^{*3}} & -\frac{1}{x_2^{*2}} \\ -\frac{1}{x_2^{*2}} & \frac{2x_1^*}{x_2^{*3}} \end{bmatrix} \tag{A.140}$$

whose determinant is:

$$\begin{aligned} \det\left(\nabla^2_{xx}\mathscr{L}(x^*,\lambda^*)\right) &= \frac{4P}{x_1^{*2}x_2^{*3}} - \frac{1}{x_2^{*4}} && \text{(A.141)} \\ &= \frac{1}{x_2^{*3}}\left(\frac{4P}{x_1^{*2}} - \frac{1}{x_2^*}\right) && \text{(A.142)} \\ &= \frac{1}{A_{max}^3}\left(\frac{4P}{V_{max}^2} - \frac{1}{A_{max}}\right) && \text{(A.143)} \\ &= \frac{1}{A_{max}^3}\frac{4PA_{max} - V_{max}^2}{V_{max}^2 A_{max}} && \text{(A.144)} \end{aligned}$$

This determinant is positive since $PA_{max} \geq V_{max}^2$ according to (A.66). Hence, the second-order sufficient conditions are satisfied and $x^*$ of Case 11, given in (A.55) and computed at line 14 of Algorithm 2, is a strict local minimum.

## A.16 LICQ

In the KKT first-order necessary conditions stated at the beginning of Appendix A, the LICQ should hold. As defined in Definition 12.4 of [Nocedal and Wright, 2006], the LICQ holds at a given $x^*$ if the set of active constraint gradients are linearly independent at $x^*$.
For completeness of the proof of Algorithm 2 (and thus of Algorithm 1), let us verify that the LICQ holds at the local minimum $x^*$ of Case 3 in (A.21), Case 4 in (A.27), Case 10 in (A.54) and Case 11 in (A.55).
For Cases 3 and 4, only one constraint is active, $C_2(x) = 0$ and $C_3(x) = 0$, respectively. Hence, the LICQ holds at $x^*$ of Case 3 if $\nabla_x C_2(x^*)$ in (A.9) is nonzero which is necessarily the case since $-Px_2 - x_1^2$ is strictly negative for any $x$. The LICQ holds at $x^*$ of Case 4 if $\nabla_x C_3(x^*)$ in (A.10) is nonzero which is always the case.
In Case 10, there are two active constraints, $C_2(x) = C_4(x) = 0$. From (A.9) and (A.11), $\nabla_x C_2(x)$ and $\nabla_x C_4(x)$ are easily seen to be linearly independent whatever $x$ so that the LICQ holds.
Finally, in Case 11, there are two active constraints, $C_3(x) = C_4(x) = 0$, and, according to (A.10) and (A.11), $\nabla_x C_3(x)$ and $\nabla_x C_4(x)$ are trivially linearly independent whatever $x$ so that the LICQ holds.

APPENDIX

# B Total time $T$ and velocity $V$ as a functions of displacement $P$

The details of this appendix were written by Dr. Marc Gouttefarde.

## B.1 Total time $T$ as a function of displacement $P$

In the case of the minimum-time S-curve trajectory, with acceleration $A$ and velocity $V$ computed by Algorithm 1, this section proves that the total time $T$ given in (2.4) is an increasing function of the displacement $P$.

According to the expression of $T$ in (2.4), $T$ depends on $P$, $V$ and $A$. Referring to the different possible cases in Algorithm 1, either $V$ and $A$ are functions of $P$ or else $V$ and $A$ are independent of $P$ i.e. they are constant. In the latter case, from (2.4), $T$ is directly seen to be an increasing function of $P$. On the contrary, when $V$ or $A$ is a function of $P$, it is not obvious from (2.4) that $T$ is an increasing function of $P$.

In fact, when $P$ is sufficiently small, $V$ and $A$ are functions of $P$ as given at line 3 and, when $P$ increases, the conditions on $P$ at lines 1, 2 and 13 will not be satisfied anymore for large values of $P$ so that $V$ and $A$ become constant. Let us first look at the relationship between the conditions on $P$ at lines 1 and 2 of Algorithm 1, i.e.

$$P \leq \frac{2A_{max}^3}{J^2} \tag{B.1}$$

and

$$P \leq \sqrt{\frac{4V_{max}^3}{J}} \tag{B.2}$$

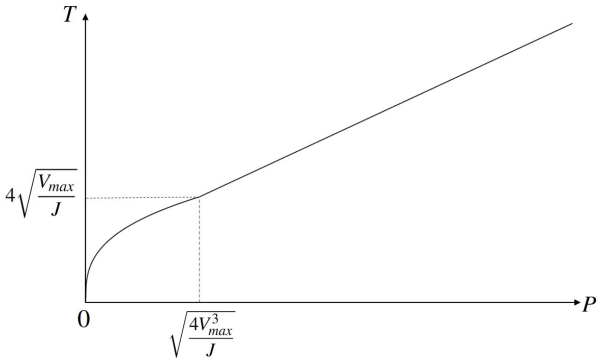



Figure B.1 – The increasing function $T(P)$ in Case 1 where $\sqrt{JV_{max}} < A_{max}$.

respectively. In particular, when P increases from zero, line 5 will be executed if and only if

$$\sqrt{\frac{4V_{max}^3}{J}} < \frac{2A_{max}^3}{J^2}. \tag{B.3}$$

which, after some elementary calculations, can be shown to be equivalent to

$$\sqrt{JV_{max}} < A_{max}. \tag{B.4}$$

To analyze the dependence of the total time T on P, we will thus distinguish two cases: (B.4) satisfied (Subsection B.1.1) and (B.4) not satisfied (Subsection B.1.2). This analysis will show that the function $T(P)$ defined by Algorithm 1 is always an increasing and continuous function of P.

### B.1.1 Case 1: $\sqrt{JV_{max}} < A_{max}$

For $0 \le P \le \sqrt{\frac{4V_{max}^3}{J}}$, according to line 3 of Algorithm 1, $V = \sqrt[3]{\frac{JP^2}{4}}$ and $A = \sqrt[3]{\frac{J^2P}{2}}$ so that

$$\begin{aligned} T = \frac{P}{V} + \frac{V}{A} + \frac{A}{J} &= \frac{P}{\sqrt[3]{\frac{JP^2}{4}}} + \frac{\sqrt[3]{\frac{JP^2}{4}}}{\sqrt[3]{\frac{J^2P}{2}}} + \frac{\sqrt[3]{\frac{J^2P}{2}}}{J} = \sqrt[3]{\frac{4P}{J}} + \sqrt[3]{\frac{P}{2J}} + \sqrt[3]{\frac{P}{2J}} \\ &= \sqrt[3]{\frac{4P}{J}} + 2\sqrt[3]{\frac{P}{2J}} = \sqrt[3]{\frac{4P}{J}} + \sqrt[3]{\frac{4P}{J}} = 2\sqrt[3]{\frac{4P}{J}}. \end{aligned}$$

Hence, for $0 \le P \le \sqrt{\frac{4V_{max}^3}{J}}$, T is a continuous increasing function of P since $T = 2\sqrt[3]{\frac{4P}{J}}$.

For $\sqrt{\frac{4V_{max}^3}{J}} < P \leq \frac{2A_{max}^3}{J^2}$, according to line 5 of Algorithm 1, $V = V_{max}$ and $A = \sqrt{JV_{max}}$ which implies from (2.4) that

$$T = \frac{P}{V} + \frac{V}{A} + \frac{A}{J} = \frac{P}{V_{max}} + \frac{V_{max}}{\sqrt{JV_{max}}} + \frac{\sqrt{JV_{max}}}{J} = \frac{P}{V_{max}} + 2\sqrt{\frac{V_{max}}{J}}$$

so that T is an increasing (affine) function of P.

Note that for $P = \sqrt{\frac{4V_{max}^3}{J}}$, the values of T obtained from (B.5) and (B.5) are both equal to $4\sqrt{\frac{V_{max}}{J}}$ so that the function $T(P)$ is continuous on $0 \leq P \leq \frac{2A_{max}^3}{J^2}$. Indeed, with (B.5)

$$T = 2\sqrt[3]{\frac{4}{J}\sqrt{\frac{4V_{max}^3}{J}}} = 4\sqrt[3]{\sqrt{\frac{V_{max}^3}{J^3}}} = 4\sqrt{\sqrt[3]{\frac{V_{max}^3}{J^3}}} = 4\sqrt{\frac{V_{max}}{J}} \tag{B.5}$$

and with (B.5)

$$T = \frac{1}{V_{max}}\sqrt{\frac{4V_{max}^3}{J}} + 2\sqrt{\frac{V_{max}}{J}} = \sqrt{\frac{4V_{max}}{J}} + 2\sqrt{\frac{V_{max}}{J}} = 4\sqrt{\frac{V_{max}}{J}} \tag{B.6}$$

Finally, for $\frac{2A_{max}^3}{J^2} < P \leq +\infty$, since $\sqrt{JV_{max}} < A_{max}$, according to line 9 of Algorithm 1, $V = V_{max}$ and $A = \sqrt{JV_{max}}$ which is similar to the previous case i.e. T is given by (B.5) and it is an increasing and continuous function of P.

In summary, when $\sqrt{JV_{max}} < A_{max}$, the function $T(P)$ defined by Algorithm 1 is an increasing and continuous function over $0 \leq P \leq +\infty$ such as shown in Figure B.1.

### B.1.2 Case 2: $\sqrt{JV_{max}} \geq A_{max}$

In this second case, since (B.3) is equivalent to (B.4), we have

$$\sqrt{\frac{4V_{max}^3}{J}} \geq \frac{2A_{max}^3}{J^2} \tag{B.7}$$

and line 5 of Algorithm 1 is thus never executed. Moreover, line 9 is executed only in the particular case where $A_{max} = \sqrt{JV_{max}}$.

For $0 \leq P \leq \frac{2A_{max}^3}{J^2}$, according to line 3 of Algorithm 1, $V = \sqrt[3]{\frac{JP^2}{4}}$ and $A = \sqrt[3]{\frac{J^2P}{2}}$ so that T is given by (B.5), i.e. $T = 2\sqrt[3]{\frac{4P}{J}}$, and $T(P)$ is thus a continuous increasing continuous of P.

Then, in the particular case where $\sqrt{JV_{max}} = A_{max}$, for $\frac{2A_{max}^3}{J^2} < P \leq +\infty$, line 9 yields $V = V_{max}$ and $A = \sqrt{JV_{max}} = A_{max}$. Eq. (2.4) then implies that

$$T = \frac{P}{V} + \frac{V}{A} + \frac{A}{J} = \frac{P}{V_{max}} + \frac{V_{max}}{A_{max}} + \frac{A_{max}}{J} = \frac{PJ}{A_{max}^2} + 2\frac{A_{max}}{J}. \tag{B.8}$$

$T(P)$ is thus a continuous increasing function of P on $\frac{2A_{max}^3}{J^2} < P \leq +\infty$. Moreover, at $P = \frac{2A_{max}^3}{J^2}$, we have

$$2\sqrt[3]{\frac{4P}{J}} = 4\frac{A_{max}}{J} \tag{B.9}$$

and

$$\frac{PJ}{A_{max}^2} + 2\frac{A_{max}}{J} = 4\frac{A_{max}}{J} \tag{B.10}$$

so that $T(P)$ is continuous at $P = \frac{2A_{max}^3}{J^2}$. In summary, in the particular case $A_{max} = \sqrt{JV_{max}}$, $T(P)$ defined by Algorithm 1 is an increasing and continuous function over $0 \leq P \leq +\infty$.

Now, it remains to analyze the case where $\sqrt{JV_{max}} > A_{max}$ and $P > \frac{2A_{max}^3}{J^2}$. Let us first consider the condition in the if statement of line 13 of Algorithm 1 which is

$$\begin{aligned} & V = \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J} > V_{max} \\ \iff \quad & \sqrt{A_{max}^4 + 4J^2PA_{max}} > 2JV_{max} + A_{max}^2 \\ \iff \quad & JPA_{max} > JV_{max}^2 + V_{max}A_{max}^2 \\ \iff \quad & P > \frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}}. \end{aligned} \tag{B.11}$$

Since $\sqrt{JV_{max}} > A_{max} \Leftrightarrow V_{max} > \frac{A_{max}^2}{J}$, note that

$$\frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}} > \frac{2A_{max}^4}{J^2A_{max}} = \frac{2A_{max}^3}{J^2} \tag{B.12}$$

which implies that for $\frac{2A_{max}^3}{J^2} < P \leq \frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}}$, A and V are given by lines 11 and 12 of Algorithm 1, respectively, while for $\frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}} < P \leq +\infty$, A and V are given by lines 11 and 14, respectively.

For $\frac{2A_{max}^3}{J^2} < P \leq \frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}}$, $A = A_{max}$ and $V = \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J}$ so that T is calculated from (2.4) as follows:

$$T = \frac{P}{V} + \frac{V}{A} + \frac{A}{J} = \frac{2JP}{-A_{max}^2 + \sqrt{g(P)}} + \frac{\sqrt{g(P)}}{2JA_{max}} + \frac{A_{max}}{2J} \tag{B.13}$$

with $g(P) = A_{max}^4 + 4J^2PA_{max}$. The derivative of T with respect to P is

$$\begin{aligned}
\frac{dT}{dP} &= \frac{2J\left(-A_{max}^2 + \sqrt{g(P)}\right) - 2JP\left(\frac{2J^2A_{max}}{\sqrt{g(P)}}\right)}{\left(-A_{max}^2 + \sqrt{g(P)}\right)^2} + \frac{J}{\sqrt{g(P)}} \\
&= \frac{2J\left(-A_{max}^2\sqrt{g(P)} + g(P) - 2J^2A_{max}P\right)}{\left(-A_{max}^2 + \sqrt{g(P)}\right)^2\sqrt{g(P)}} + \frac{J}{\sqrt{g(P)}} \\
&= \frac{2JA_{max}\left(-A_{max}\sqrt{g(P)} + A_{max}^3 + 2J^2P\right)}{\left(-A_{max}^2 + \sqrt{g(P)}\right)^2\sqrt{g(P)}} + \frac{J}{\sqrt{g(P)}}
\end{aligned} \tag{B.14}$$

Hence, $\frac{dT}{dP}$ is non-negative, i.e. $T(P)$ is an increasing function, if $-A_{max}\sqrt{g(P)} + A_{max}^3 + 2J^2P \geq 0$. We have

$$\begin{aligned}
&-A_{max}\sqrt{g(P)} + A_{max}^3 + 2J^2P \geq 0 \Longleftrightarrow \left(A_{max}^3 + 2J^2P\right)^2 \geq A_{max}^2 g(P) \\
&\Longleftrightarrow \left(A_{max}^3 + 2J^2P\right)^2 \geq A_{max}^2\left(A_{max}^4 + 4J^2PA_{max}\right) \Longleftrightarrow 4J^4P^2 \geq 0.
\end{aligned} \tag{B.15}$$

Thus, $T(P)$ is an increasing function of P. Moreover, at $P = \frac{2A_{max}^3}{J^2}$, $g(P) = 3A_{max}^2$ and from (B.13), T is found after some elementary calculations to be equal to $4\frac{A_{max}}{J}$. The latter is equal to the value given by (B.5) for $P = 2A_{max}^3/J^2$ (see (B.9)) which implies that $T(P)$ is continuous at $P = 2A_{max}^3/J^2$.

Finally, for $\frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}} < P \leq +\infty$, A and V are given by lines 11 and 14, respectively, i.e. $A = A_{max}$ and $V = V_{max}$, $T(P)$ is given by

$$T = \frac{P}{V} + \frac{V}{A} + \frac{A}{J} = \frac{P}{V_{max}} + \frac{V_{max}}{A_{max}} + \frac{A_{max}}{J} \tag{B.16}$$

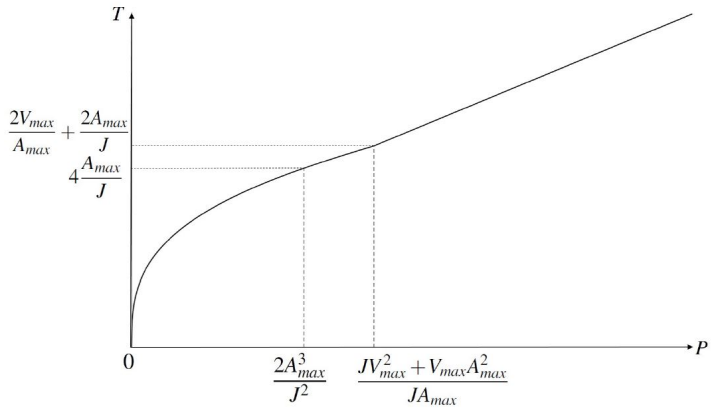



Figure B.2 – The increasing function $T(P)$ in Case 2 where $\sqrt{JV_{max}} \geq A_{max}$.

and it is thus a continuous increasing function of P. It only remains to check if $T(P)$ is continuous at $P = \frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}}$. From (B.16), we have

$$T = \frac{P}{V_{max}} + \frac{V_{max}}{A_{max}} + \frac{A_{max}}{J} = \frac{V_{max}}{A_{max}} + \frac{A_{max}}{J} + \frac{V_{max}}{A_{max}} + \frac{A_{max}}{J} = \frac{2V_{max}}{A_{max}} + \frac{2A_{max}}{J}.$$

Besides, we have $g(P) = \left(A_{max}^2 + 2JV_{max}\right)^2$ at $P = \frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}}$ so that from (B.13)

$$\begin{aligned} T &= \frac{2JP}{-A_{max}^2 + \sqrt{g(P)}} + \frac{\sqrt{g(P)}}{2JA_{max}} + \frac{A_{max}}{2J} \\ &= \frac{JV_{max}^2 + V_{max}A_{max}^2}{JV_{max}A_{max}} + \frac{A_{max}^2 + 2JV_{max}}{2JA_{max}} + \frac{A_{max}}{2J} \\ &= \frac{V_{max}}{A_{max}} + \frac{A_{max}}{J} + \frac{A_{max}}{2J} + \frac{V_{max}}{A_{max}} + \frac{A_{max}}{2J} \\ &= \frac{2V_{max}}{A_{max}} + \frac{2A_{max}}{J}. \end{aligned} \tag{B.17}$$

which is equal to the value obtained for T in (B.17), i.e., $T(P)$ is continuous at $P = \frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}}$.

In summary, when $\sqrt{JV_{max}} \geq A_{max}$, the function $T(P)$ defined by Algorithm 1 is an increasing and continuous function over $0 \leq P \leq +\infty$ such as shown in Figure B.2.

Since this was also the case for $\sqrt{JV_{max}} < A_{max}$, the function $T(P)$ defined by Algorithm 1 is always an increasing and continuous function of P.

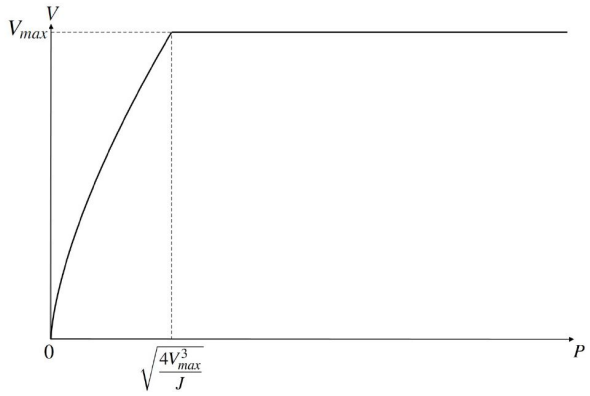



Figure B.3 – The nondecreasing function $V(P)$ in Case 1 ($\sqrt{JV_{max}} < A_{max}$).

## B.2 Velocity $V$ as a function of displacement $P$

The analysis made in Section B.1 shows that the function $T(P)$ defined by Algorithm 1 is an increasing and continuous function over $0 \leq P \leq +\infty$. From this analysis, one can directly deduce that the maximum velocity value $V$ of the $3^{rd}$ order polynomial S-curve trajectory is a continuous nondecreasing function of the displacement $P$, as detailed below.

### B.2.1 Case 1: $V(P)$ for $\sqrt{JV_{max}} < A_{max}$

For $0 \leq P \leq \sqrt{\frac{4V_{max}^3}{J}}$, according to line 3 of Algorithm 1, $V = \sqrt[3]{\frac{JP^2}{4}}$ and, for $\sqrt{\frac{4V_{max}^3}{J}} < P \leq \frac{2A_{max}^3}{J^2}$, according to line 5, $V = V_{max}$. Furthermore, for $\frac{2A_{max}^3}{J^2} < P \leq +\infty$, since $\sqrt{JV_{max}} < A_{max}$, according to line 9 of Algorithm 1, $V = V_{max}$. Besides, at $P = \sqrt{\frac{4V_{max}^3}{J}}$, $V = \sqrt[3]{\frac{JP^2}{4}} = V_{max}$ so that $V(P)$ is a nondecreasing continuous function of $P$ such as shown in Figure B.3.

### B.2.2 Case 2: $V(P)$ for $\sqrt{JV_{max}} \geq A_{max}$

For $0 \leq P \leq \frac{2A_{max}^3}{J^2}$, according to line 3 of Algorithm 1, $V = \sqrt[3]{\frac{JP^2}{4}}$.

Then, in the particular case where $\sqrt{JV_{max}} = A_{max}$, for $\frac{2A_{max}^3}{J^2} < P \leq +\infty$, line 9 implies that $V = V_{max}$. At $P = \frac{2A_{max}^3}{J^2}$, since $\sqrt{JV_{max}} = A_{max}$, we have $V = \sqrt[3]{\frac{JP^2}{4}} = V_{max}$.

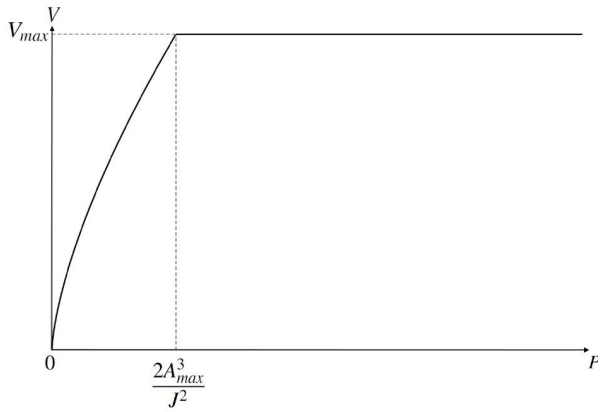



Figure B.4 – The nondecreasing function $V(P)$ in Case 2 when $\sqrt{JV_{max}} = A_{max}$.

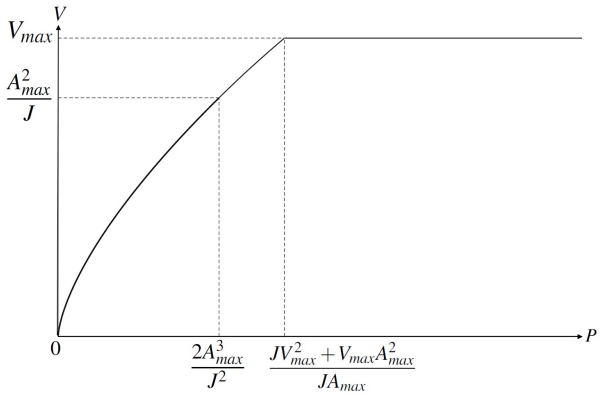



Figure B.5 – The nondecreasing function $V(P)$ in Case 2 when $\sqrt{JV_{max}} > A_{max}$.

Hence, when $\sqrt{JV_{max}} = A_{max}$, $V(P)$ is a nondecreasing continuous function of P as shown in Figure B.4.

Now, let us consider the case $\sqrt{JV_{max}} > A_{max}$. For $\frac{2A_{max}^3}{J^2} < P \leq \frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}}$, we have

$$V = \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J} \tag{B.18}$$

while for $\frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}} < P \leq +\infty$, we have $V = V_{max}$.

Moreover, $V(P)$ is continuous at $P = \frac{2A_{max}^3}{J^2}$ since straightforward calculations yield

$$V = \sqrt[3]{\frac{JP^2}{4}} = \frac{A_{max}^2}{J} \tag{B.19}$$

and

$$V = \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2 P A_{max}}}{2J} = \frac{A_{max}^2}{J} \tag{B.20}$$

Similarly, $V(P)$ is continuous at $P = \frac{JV_{max}^2 + V_{max}A_{max}^2}{JA_{max}}$ since

$$\begin{aligned} A_{max}^4 + 4J^2 P A_{max} &= A_{max}^4 + 4J^2 V_{max}^2 + 4JV_{max}A_{max}^2 \\ &= \left(A_{max}^2 + 2JV_{max}\right)^2 \end{aligned} \tag{B.21}$$

so that

$$V = \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2 P A_{max}}}{2J} = V_{max}. \tag{B.22}$$

In summary, $V(P)$ is a nondecreasing continuous function of P such as shown in Figure B.5.

**Abstract**

The popularity of parallel robots has been steadily increasing in recent decades. This popularity has been strongly stimulated by the many advantages these robots offer over their serial counterparts, in some industrial applications requiring high acceleration and very good accuracy. However, in order to fully exploit their potential and make the most of their capabilities, there is still much to be done. In addition to mechanical design, calibration and structure optimization, the development of efficient control approaches plays a key role in improving the overall performance of these robots. Besides, selective sorting consists in sorting and recovering waste according to its nature: metals, paper, glass, organic, etc, to facilitate its recycling. They are sorted either by those who produce them, or by specialized organizations in sorting centers. The objective of this thesis is to study the use of parallel robots for pick-and-throw (P&T) applications in the selective and fast sorting of waste. The goal is to perform P&T tasks in a robust and fast way using a parallel manipulator (made available by TECNALIA under a collaborative research contract), demonstrating the interest and relevance of a P& T approach compared to a traditional P&P approach in the context of a selective waste sorting application. In this context, trajectory generation and control design are addressed in this thesis. On the one hand, motion planning for PKMs is not trivial. Different constraints such as kinematics and dynamics constraints, continuity, etc., should be taken into account to generate a feasible and appropriate trajectory that meets the requirements of a specific application. On the other hand, the control of PKMs is often considered in the literature as a challenging task due to their highly nonlinear dynamics, abundant uncertainties, parameter variation, and actuation redundancy. In this thesis, we aim to generate a fast and accurate P&T task using a parallel manipulator. Thus, we first propose a time-optimal P&T trajectory that significantly reduces the cycle time compared to the usual P&P technique. Real-time experiments have been conducted for the validation of the proposed P&T method, showing the relevance of this method with respect to the P&P process and to an existing P&T technique in the literature. Second, advanced robust control strategies have been proposed, which are extensions of (i) the standard RISE (Robust Integral of the Sign of the Error) feedback control, (ii) the DCAL (Desired Compensation Adaptive Law), and (iii) the model-free control (MFC). Lyapunov-based stability analysis is established for all the proposed controllers verifying the asymptotic convergence of the tracking errors. In order to validate the proposed controllers, numerical simulations are conducted on a parallel robot prototype, called T3KR. Several simulations are tested including robustness towards payload changes, and robustness towards speed variations. The relevance of the proposed control schemes is proved through the improvement of the tracking errors at different dynamic operating conditions.

**Keywords:** *Parallel manipulators, Pick-and-Throw, time-optimal trajectory, RISE control, DCAL strategy, Model-Free Control, stability analysis, real-time experiments, numerical simulations*

**Résumé**

La popularité des robots parallèles n'a cessé de croître au cours des dernières décennies. Cette popularité a été fortement stimulée par les nombreux avantages que ces robots offrent par rapport à leurs homologues sériels, dans certaines applications industrielles nécessitant de fortes accelerations et une très bonne précision. Toutefois, afin d'exploiter pleinement leur potentiel et de tirer le meilleur parti de leurs capacités, il reste encore beaucoup à faire. Outre la conception mécanique, la calibration et l'optimisation de la structure, le développement d'approches de commande efficaces joue un rôle clé dans l'amélioration des performances globales de ces robots. Par ailleurs, le tri sélectif consiste à trier et à récupérer les déchets en fonction de leur nature : métaux, papier, verre, organique, etc, pour faciliter leur recyclage. Ils sont triés soit par ceux qui les produisent, soit par des organismes spécialisés dans des centres de tri. L'objectif de cette thèse est d'étudier l'utilisation de robots parallèles pour des applications de "pick-and-throw" (P&T) dans le tri sélectif et rapide des déchets. L'objectif est de réaliser des tâches de P&T de manière robuste et rapide à l'aide d'un manipulateur parallèle (mis à disposition par Tecnalia dans le cadre d'un contrat de recherche collaborative), démontrant l'intérêt et la pertinence d'une approche P&T par rapport à une approche P&P traditionnelle dans le cadre d'une application de tri sélectif de déchets. Dans ce contexte, la génération de trajectoires et la conception de commandes sont abordées dans cette thèse. D'une part, la planification du mouvement pour les PKMs n'est pas triviale. Différentes contraintes telles que les contraintes cinématiques et dynamiques, la continuité, etc., doivent être prises en compte pour générer une trajectoire réalisable et appropriée qui répond aux exigences d'une application spécifique. D'autre part, la commande des PKMs est souvent considérée dans la littérature comme une tâche difficile en raison de leur dynamique hautement non linéaire, des incertitudes abondantes, de la variation des paramètres, et de la redondance de l'actionnement. Dans cette thèse, nous visons à générer une tâche de P&T rapide et précise en utilisant un manipulateur parallèle. Ainsi, nous proposons d'abord une trajectoire P&T optimale en temps qui réduit considérablement le temps de cycle par rapport à la technique P&P habituelle. Des expérimentations en temps réel ont été menées pour la validation de la méthode P&T proposée, montrant la pertinence de cette méthode par rapport au processus P&P et à une technique P&T existante dans la littérature. Deuxièmement, des stratégies de commande robustes avancées ont été proposées, qui sont des extensions (i) de la commande RISE (Robust Integral of the Sign of the Error), (ii) de la commande DCAL (Desired Compensation Adaptive Law), et (iii) du commande sans modèle (MFC). L'analyse de stabilité basée de type Lyapunov est établie pour tous les contrôleurs proposés, vérifiant la convergence asymptotique des erreurs de suivi. Afin de valider les contrôleurs proposés, des simulations numériques sont réalisées sur un prototype de robot parallèle, appelé T3KR. Plusieurs simulations sont testées, notamment la robustesse aux changements de charge et la robustesse aux variations de vitesse. La pertinence des schémas de contrôle

proposés est prouvée par l'amélioration des erreurs de suivi dans différentes conditions de fonctionnement dynamiques.

**Mots clefs :** *Robot manipulateurs parallèles, Pick-and-Throw, trajectoire en temps optimal, commande RISE, stratégie DCAL, commande sans modèle, analyse de stabilité, expérimentation en temps réel, simulations numériques.*

LIRMM — 161, rue Ada — 34095 Montpellier cedex 5 — France