Hierarchical Adaptive Control of Modular and Reconfigurable Robot Manipulator Platforms

by

Gokhan Gungor

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2020

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	NAME: Guangjun Liu Title: Professor, Department of Aerospace Engineering, Ryerson University
Supervisor:	NAME: William Melek Title: Professor, Department of Mechanical and Mechatronics Engineering, University of Waterloo
Co-Supervisor:	NAME: Baris Fidan Title: Professor, Department of Mechanical and Mechatronics Engineering, University of Waterloo
Internal Member:	NAME: Amir Khajepour Title: Professor, Department of Mechanical and Mechatronics Engineering, University of Waterloo
Internal Member:	NAME: Behrad Khamesee Title: Professor, Department of Mechanical and Mechatronics Engineering, University of Waterloo
Internal-External Member:	NAME: John Zelek Title:Associate Professor, Department of Systems Design, Engineering, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Within the rapidly growing interest in today's robotics industry, modular and reconfigurable robots (MRRs) are among the most auspicious systems to expand the adaptability of robotic applications. They are adaptable to multiple industrial field applications but they also have additional advantages such as versatile hardware, easier maintenance, and transportability. However, such features render the controller design that manages a variety of robot configurations with reliable performance more complex since their system dynamics involve not only nonlinearities and uncertainties but also changing dynamics parameters after the reconfiguration. In this thesis, the motion control problem of MRR manipulators is addressed and hierarchical adaptive control architecture is developed for MRRs. This hierarchical structure allows the adjustment of the nominal parameters of an MRR system for system parameter identification and control design purposes after the robot is reconfigured. This architecture simplifies the design of adaptive control for MRRs which is effective in the presence of dynamic parameter uncertainty, unmodeled dynamics, and disturbance. The proposed architecture provides flexibility in choosing adaptive algorithms applicable to MRRs. The developed architecture consists of high-level and low-level modules. The high-level module handles the dynamic parameters changes and reconstructs the parametric model used for on-line parameter identification after the modules are reassembled. The low-level structure consists of an adaptive algorithm updated by an on-line parameter estimation to handle the dynamic parameter uncertainties. Furthermore, a robust adaptive term is added into this low-level controller to compensate for the unmodeled dynamics and disturbances. The proposed adaptive control algorithms guarantee uniformly ultimate boundedness (UUB) of the MRR trajectories in terms of robust stability despite the dynamic parameter uncertainty, unmodeled dynamics, changes in the system dynamics, and disturbance.

Acknowledgements

I would like to thank Dr. William Melek and Dr. Baris Fidan for their continuous support and dedication they have provided throughout the whole period of this research thesis. I would also like to thank Dr. William Melek for the great working environment he created within the RoboHub and Laboratory Computational Intelligence and Automation (LCIA). I would like to thank my committee members, Dr. Guangjun Liu, Dr. Amir Khajepour, and Dr. Behrad Khamesee for their helpful inputs to improve the quality of this thesis. I also appreciate Dr. Dana Kulic's suggestions during my comprehensive exam. Last, but not least, I would like to thank my family for the love and support shared with me, especially when I had hard times. Most particularly, my greatest thank goes to Melike Esen who is always believing in me. I would not have been able to enjoy and pass all the challenges without her unconditional love and support.

Gokhan Gungor

Dedication

To my love and family...

Table of Contents

List of Figures x

List of Tables xviii

1 Introduction 1

1.1 Modular and Reconfigurable Robots 1

1.2 Challenges Associated with using MRRs 6

1.3 Problem Domain 8

1.4 Thesis Contributions 9

1.5 Outline of the Thesis 11

2 Background and Literature Review 12

2.1 Background 12

2.1.1 MRR Mechanical Components 12

2.1.2 MRR Kinematics 14

2.1.3 MRR Dynamics 20

2.2 Modeling MRR Dynamics 23
2.3 Literature Review 26
2.4 MRR Control Schemes 27
2.5 Proposed Two-Layer Control Architecture 28

3 Proposed Two-Layer Control Architecture and The High Level Module 29

3.1 Proposed Two-Layer Control Architecture 29
3.2 The High Level Module 31
3.3 MRR System Parameter Identification 34
3.3.1 MRR System Parametric Model 37
3.3.2 Model Reduction 37
3.3.3 MRR Parameter Estimation 38

4 Low Level Decentralized Adaptive Sliding Mode Control Design 41

4.1 The Approach 41
4.2 Adaptive Sliding-Mode Control Design 42
4.3 Summary 48

5 Low Level Model Reference Adaptive Control 49

5.1 The Reference Model and Control Design 50
5.2 Stability Analysis 52
5.3 Summary 54

6 Simulations **55**

6.1 Description of Franka Emika Panda Robot 55

6.2 Simulation Results 55

6.3 Summary 128

7 Conclusion and Future Directions **129**

7.1 Conclusion 129

7.2 Future Directions 130

References **131**

List of Figures

1.1 Some examples of MRRs designed by researchers so far: (a) RMMS [1], [2], (b) Waterloo Modular and Reconfigurable Robot (WMRR) [3], (c) RMM developed by TRACLabs for mobile manipulation [4], (d) autonomously reconfigurable serial MARS manipulator [5]. 3

1.2 Some of MRRs produced by robot companies. (a) JACO arm [6]; (b) MOVO robot consisting of two JACO arms with mobile platform and 2 DoF Kinect [7]; (c) Gen3 ULR [8]; (d) MARA robot [9]; (e) SCHUNK LWA [10]; (f) e.Do [11]; (g) DLR III (LWR III) [12]; (h) The humanoid walking robot TORO (TOrque-controlled humanoid RObot) consisting of four different DLRs [13]. 5

1.3 (a) Some MRR joint modules, (b) some MRR link modules, (c) 4-DoF robot configuration, and (d) 2-DoF robot configuration with payload [14]. 10

2.1 (a) a revolute joint with its input and output coordinate frames (b) some revolute joints. 13

2.2 (a) a rectangular link with its input and output coordinate frames (b) some links. 14

2.3 Denavit-Hartenberg notations related to link i with joint i at what end and joint $i+1$ at the other end [15]. 16

2.4 (a) The general module representation, (b) The input part of the module, and (c) The output part of the module [16], [17], [18]. 18

2.5 Representation of a connection between the output part of the module i and input part of the module $i+1$ including the D-H notations [19], [16], [17], [18]. 19

2.6 Representation of the connection between the output part of i-th link module and the input part of $(i+1)$-th link module, including the dynamic parameters [18]. 21

2.7 Position parameters associated with link i [15]. 22

2.8 Representation of the connection between the output part of i-th link module and the input part of $(i+1)$-th link module, including the kinamatic and dynamic parameter notations [17], [18], [20]. 24

3.1 Illustration of the two layer control architecture composed of a high-level and low level modules. 30

3.2 Indirect adaptive control structure. 36

6.1 The Franka Emika Panda robot configurations. a) Configuration 1, b) Configuration 2, and c) Configuration 3 actuated with (i) no load, (ii) 1.5 Kg load, and (iii) 3 Kg load. 56

6.2 DH parameters for the Franka Emika Panda robot. $d_1 = 0.333$m, $d_3 = 0.316$m, $d_5 = 0.384$m, $d_f = 0.107$m, $a_4 = 0.0825$m, $a_5 = -0.0825$m, $a_7 = 0.088$m [21]. 57

6.3 Pick a payload and place to the desired position in the joint space. 58

6.4 Adaptive sliding mode controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1. 69

6.5 Adaptive model reference controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1. 70

6.6 Robust PD controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1. 71

6.7 Adaptive sliding mode controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 72

6.8 Adaptive model reference controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 73

6.9 Robust PD controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 74

6.10 Adaptive sliding mode controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 3. 75

6.11 Adaptive model reference controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 3. 76

6.12 Robust PD controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 3. 77

6.13 Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1. 78

6.14 Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1. 79

6.15 Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1. 80

6.16 Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 81

6.17 Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 82

6.18 Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 83

6.19 Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 84

6.20 Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 85

6.21 Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 86

6.22 Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 87

6.23 Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 88

6.24 Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 89

6.25 Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 90

6.26 Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 91

6.27 Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 92

6.28 Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 93

6.29 Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 94

6.30 Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2. 96

6.31 Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 1. 97

6.32 Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 1. 98

6.33 Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 1. 99

6.34 Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 1. 100

6.35 Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 2. 101

6.36 Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 2. 102

6.37 Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 2. 103

6.38 Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 3. 104

6.39 Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 3. 105

6.40 Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 3. 106

6.41 Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 1. 107

6.42 Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 1. 108

6.43 Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 1. 109

6.44 Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 1. 110

6.45 Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 2. 111

6.46 Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 2. 112

6.47 Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 2. 113

6.48 Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 3. 114

6.49 Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 3. 115

6.50 Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 3. 116

6.51 Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 1. 117

6.52 Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 1. 118

6.53 Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 1. 119

6.54 Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 1. 120

6.55 Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 2. 121

6.56 Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 2. 122

6.57 Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 2. 123

6.58 Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 2. 124

6.59 Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 3. 125

6.60 Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 3. 126

6.61 Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 3. 127

List of Tables

6.1 The results for the first configuration of Joint 1. 62

6.2 The results for the first configuration of Joint 2 63

6.3 The results for the second configuration of Joint 1. 65

6.4 The results for the second configuration of Joint 2. 66

6.5 The results for the third configuration of Joint 1. 68

Chapter 1

Introduction

1.1 Modular and Reconfigurable Robots

Robot manipulators are well utilized in many industries and used in a vast array of applications [22], [15]. Tasks requiring in the robotics industry, include welding, material handling, palleting, painting, assembly, to name a few [23]. These tasks are currently performed by standard robot manipulators. However, these standard robot manipulators have minimal adaptability and fixed configurations. Robot systems such as Selective Compliance Assembly Robot Arm (SCARA) [24] and Programmable Universal Machine for Assembly (PUMA) [25] only offer software adaptability related primarily to re-programmability. Such systems may not be able to complete tasks effectively when the assembly line, product in-the-line, or manufacturing procedure are changed. The operational envelopes of these systems are limited because of their fixed configurations [26]. For instance, in case of a change in the task space, environment, or product, challenges with mechanical singularities and constraints in joint modules can make it challenging for these robots to be adapted to

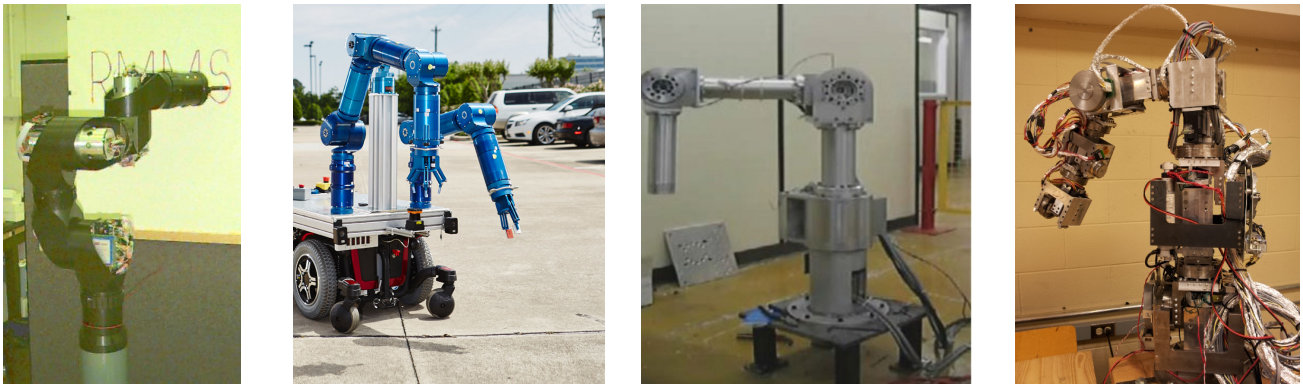
the changes in their environment [27].

For today's robotics industry, modular and reconfigurable robots (MRRs) are among the most promising systems to expand the range of industrial applications. The versatile interchangeable modules of the MRRs can be quickly rearranged in various configurations [20]. Their adaptability allows operators to change the module parts rapidly, and assemble different robot configurations. MRRs are especially valuable in inflexible and versatile automated manufacturing plants where the tasks that robots need to perform change regularity and some may not be known a priori [19].

Furthermore, faced with a competitive universal environment, to reduce the overall costs, today's small and medium-sized automation and manufacturing industries particularly can greatly benefit from MRR platforms to adapt efficiently to changes in production [28], [29], [30]. MRRs offer a tremendous economic advantage stemming from the potential to lower overall tooling costs, as they allow the assembly of complex robotic structures from a few rudimentary mass-produced modules. Besides, MRRs are not only adaptable to different task requirements for industrial applications, but they also have exclusive advantages such as easier maintenance and transportability.

Easier maintenance is a great advantage of MRRs, which minimizes downtime needed for system maintainance and repair. Without the need to decommission the entire robot, module parts with damaged components can be quickly replaced by new ones. Besides, the ability to change module parts can be helpful where humans cannot effectively and safely perform large scale repairs and maintenance. MRRs also hold an apparent benefit over the standard robots in terms of their transportability. They can reduce manufacturers' need to invest heavily in tools and structures because of their easy of installation and transportability [31].

Furthermore, MRRs have recently been introduced in tasks involving human-machine interaction (HMI). For example, the FourByThree team [32], [33] designs a modular industrial robotic platform that is implemented to collaborate with humans in order to perform industrial manufacturing tasks in a flexible work environment. Thus, with the corresponding advantages of MRRs, it is expected that the demand for MRRs will continue to grow, aligned with the need for many industries, particularly flexible manufacturing [28].

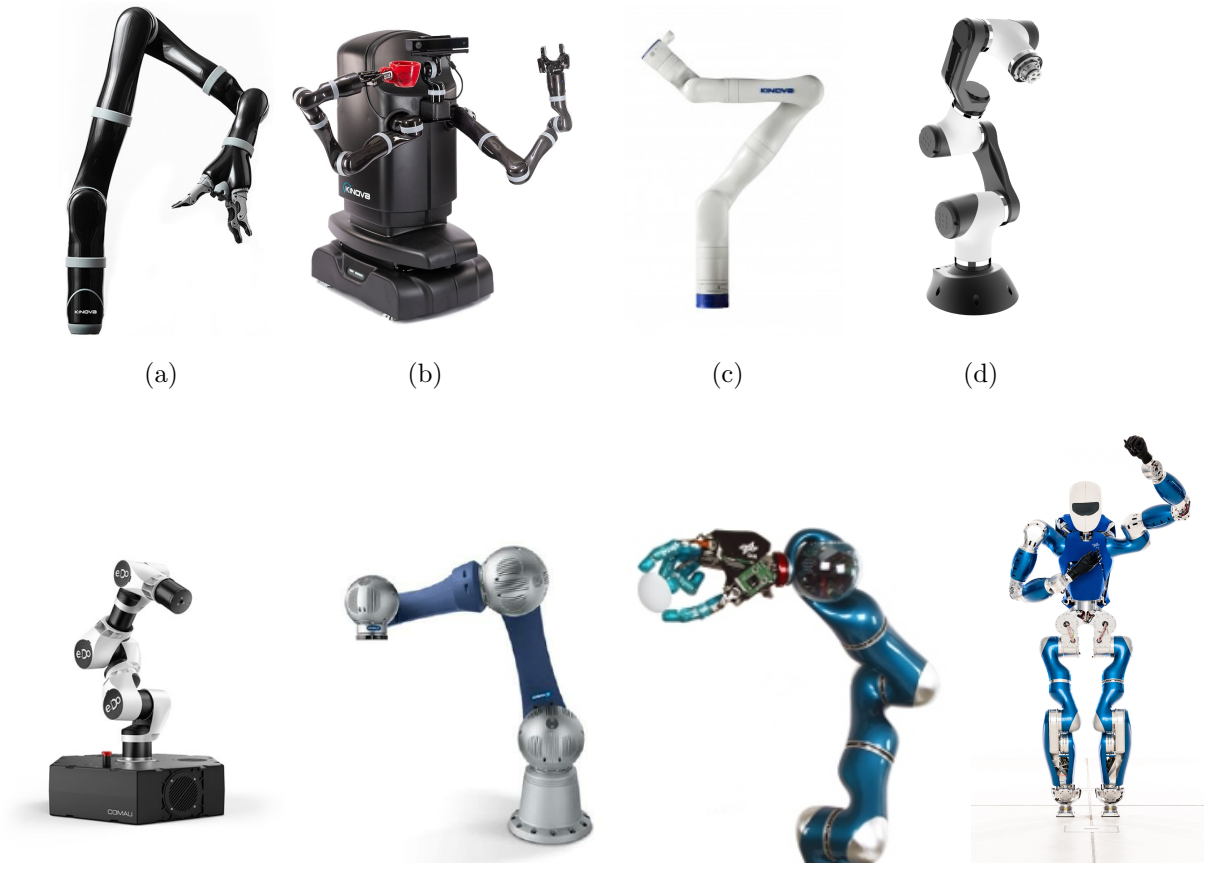


(a) (b) (c) (d)

Figure 1.1: Some examples of MRRs designed by researchers so far: (a) RMMS [1], [2], (b) Waterloo Modular and Reconfigurable Robot (WMRR) [3], (c) RMM developed by TRACLabs for mobile manipulation [4], (d) autonomously reconfigurable serial MARS manipulator [5].

In recent decades, many MRR concepts have been introduced by researchers all around the world. The initial headways of the reconfigurable robotic architecture appeared in the eighties. The author [34] introduces a robotic concept that enables assembling many different robot configurations with rotational joints actuated by alternating current (AC) motors and links of a square cross-section. An expansion of [34] is proposed in [1], [2] (Figure 1.1a), in which the authors develop a Reconfigurable Modular Manipulator System (RMMS) consisting of self-contained module parts and configuration independent control software. This robot architecture is one of the first robotic concepts bringing of a modular, reconfigurable, and portable robotic platforms together. Many reconfigurable systems are designed based on the idea introduced in [1], [35]. For instance, to realize collaborative robot operations such as hand guiding, a reconfigurable robot platform made up of two modular arms is introduced in [36].

A spring-assisted modular robot designed to increase the manipulation and its payload capabilities is introduced in [37] by Liu et al. For investigating MRRs with mobile manipulation, TRACLabs [4] develops a seven degree-of-freedom (DoF) reconfigurable modular manipulator (RMM) (Figure 1.1b) for the National Aeronautics and Space Administration (NASA) to use on planetary analog rovers. To improve the robustness and flexibility of the MRRs and reduce the reconfiguration effort, Tabandeh et al. [3] design an MRR system consisting of joint modules that produce different types of motion (Figure 1.1c). Another concept for MRRs is introduced by the authors [5] who design a Modular Autonomously Reconfigurable Serial manipulator platform for advanced manufacturing, named as the MARS manipulator (see Figure 1.1d). The main advantage of the MARS over the previous designs is its capability to reconfigure autonomously. Besides, the arm shown in Figure 1.2g is designed by the German Aerospace Center (DLR) team [12], which enables usage of the modules construct a humanoid platform [38].



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1.2: Some of MRRs produced by robot companies. (a) JACO arm [6]; (b) MOVO robot consisting of two JACO arms with mobile platform and 2 DoF Kinect [7]; (c) Gen3 ULR [8]; (d) MARA robot [9]; (e) SCHUNK LWA [10]; (f) e.Do [11]; (g) DLR III (LWR III) [12]; (h) The humanoid walking robot TORO (TOrque-controlled humanoid RObot) consisting of four different DLRs [13].

There are also many commercially available modular robots used in many applications. Kinova's JACO, MOVO, and Gen3 Ultra Lightweight Robots (ULR) that is extremely ver-

satile plug-and-play modular arm over JACO arm [6], Modular Articulated Arm (MARA) is a collaborative robotic arm from robot manufacturer Acutronic [39] (see Figures 1.2a -1.2d, respectively), the e.Do robot from Comau SpA, and the Light Weight Arms (LWAs) from robot manufacturer SCHUNK (see Figures 1.2e - 1.2f). Such systems are also conveniently used for efficient task executions in collaboration with humans in many applications for robots used in homes, hospitals, space exploration and manufacturing. More recently, an advanced modular actuator based on variable stiffness actuation was designed by QB robotic for collaborative operations where human and robots are employed together to accomplish tasks [40].

1.2 Challenges Associated with using MRRs

Despite the range of advantages MRRs provide, they have many challenges in terms of configuration selection and motion-control design. The first problem is finding a suitable configuration for a set of tasks which can be assembled by a set of MRR modules that can vary in size and weight and may include several tens to several thousand parts. Determining the most suitable configuration for a given task from the set of modules is a complex and highly nonlinear optimization problem [41], which is contingent on the given task, the optimization criteria, the group of available module parts in the inventory [31].

The solution for the aforementioned problem has been proposed by many researchers [27], [41], [42], [43], [44]. A heuristic search algorithm (HSA) paired with simulated annealing (SA) method is discussed in [42], in which the authors examine the configuration selection and build a penalty function subjected to less likely module combinations according to the desired task. Another method is proposed by the authors in [43], whose design is based on genetic algorithms (GAs). The authors [43] use GA for space configuration opti-

mization to determine the more suitable robot configuration. To improve the convergence speed of GAs, a time-efficient task-based configuration optimization (TBCO) method is conducted by Tabandeh et al. [41], whose work is based on a hybrid GA and Memetic Algorithm (MA) for fast and efficient search in the robot task space to find the most suitable robot morphology for any given task. Another time-efficient strategy based on subsequent elimination of compositions of modules is introduced in [44]. This method uses criteria such as kinematic reachability and self-collision avoidance to guide the optimization process without counting dynamics and cost-optimal solutions. Finding an optimizal robot configuration allows for more versatility in using MRR, however the motion control problem may prevent the robot from achieving the desired Position error because of dynamic uncertainty associated with changing the MRR configuration.

MRR has longstanding control challenges that trace back to the first finding for this class of robotic platform [1], [2], [34]. The first control challenge is that the MRR dynamic parameters need to be updated for each new MRR configuration. When taking into consideration a broad set of available modules as well as additional newly design modules used in a priori unknown environment for specific tasks, the dynamics of each desired MRR configuration may not be known beforehand. Moreover, when the number of modules used to reconfigure the robot increases, identifying all the possible robot morphologies and generating their system models by hand become impractical and cumbersome task [41]. Furthermore, due to changes in dynamics parameters and bounds on the norm of these terms, suitable control gains need to be redesigned to avoid the instability when the modules are reassembled [14]. However, the process of finding the values of the controller gains manually for each MRR configuration to achieve the desired Position error is challenging. Hence, increasing the downtime needed to tune the controllers before the new configuration can be effectively utilized. Therefore, a method should be employed to automatically tune

controller gains after the robot is reconfigured.

Besides, in practice, uncertain knowledge about the MRR configuration arises from dynamic parameter uncertainty, such as coordinates of the centers of mass and inertia tensors. It is well known that the last module dynamic parameters cannot be identified beforehand in the case the robot arm grasps unknown payload [19]. Particularly with a payload in hand, the uncertainty problem becomes more complicated, and the dynamic parameters may vary significantly from the nominal conditions. Moreover, unmodeled dynamics, such as dynamic couplings between the joints; friction; and backlash in the drives contribute to system model uncertainty. On top of that, disturbances, such as varying payloads and other dynamic uncertainties, add to the abovementioned challenges [45].

This thesis is motivated by the need to meet demands of industry for increased hardware flexibility in manufacturing and other sectors that rely on process automation. In specific terms, this thesis introduces a novel decentralized adaptive control architecture for serial MRR platforms. Such control architecture will provide precise and stable motion control Position error to enable reliable use of MRRs in industrial applications while being able to take advantage of their modularity to achieve automation flexibility.

1.3 Problem Domain

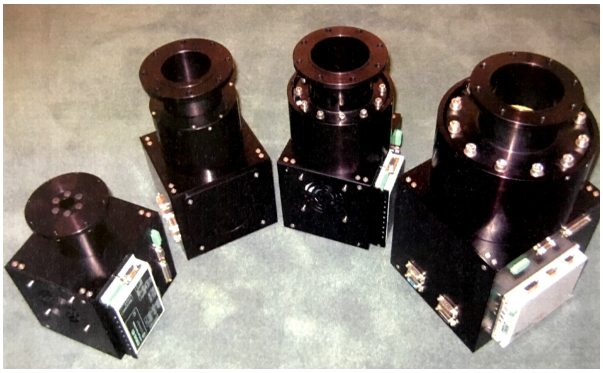
In this thesis, the proposed problem domain is consisting of two aspects of a single problem: Accurate and robust trajectory tracking control of an MRR. The first control aspect of the problem is associated with changes in the dynamics terms and the control design parameters after each reconfiguration. Many MRR configurations are assembled by adding joint and link modules or changing them to new ones with different shapes (see Figure1.3).

Besides, the resulting changes in the dynamics after the reconfiguration requires redesign of control parameters to ensure satisfactory Position error in terms of tracking, for instance. This thesis then needs to address the problem of designing adaptive controllers for MRRs to guarantee tracking of sufficiently smooth motion trajectories after each reconfiguration.

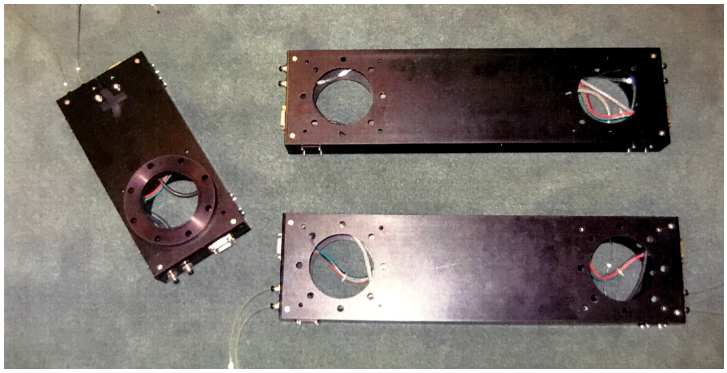
Another aspect of the the proposed control problem arises from existance of system uncertainties and disturbances. In practice, uncertainties arise from the uncertain dynamical parameter terms such as inertia tensor and the center of the mass coordinates. Particularly, the dynamic parameters regarding the last joint-link of the robot cannot be known beforehand and may change significatly if the robot carries payload with varying weight and shapes. Besides, unmodeled dynamics such as dynamic couplings between joints; friction; backlash in the joint drives and external disturbance all contribute to the sources of uncertainty. Furthermore, changing the robot morphology will result in changes to robot dynamics parameters, and that makes it difficult to tune the control system consistently under the uncertainties and disturbances. The focus of this thesis is to introduce adaptive control algorithms that guarantee robust stability and achieve desired Position error after the robot is reconfigured while overcoming the system uncertainties and disturbances.

1.4 Thesis Contributions

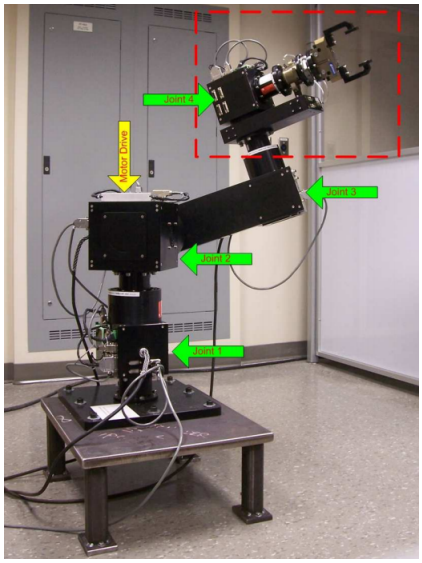
Aiming to address the three problems presented in Section 1.3, the thesis has the following three main contributions: 1) A two-layer adaptive control architecture with a high level reconfiguration supervisor module 2) An adaptive sliding mode control design robust to reconfiguration effects, modeling uncertainties and disturbances, 3) A passivity based model reference adaptive controlller which compensates for the uncertainties and disturbances in unstructured environments after MRR modules are reassembled. The research towards



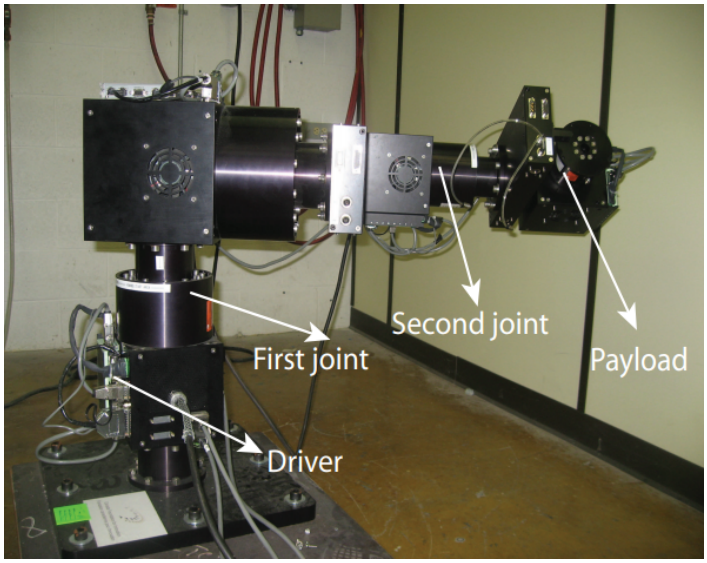
(a)



(b)



(c)



(d)

Figure 1.3: (a) Some MRR joint modules, (b) some MRR link modules, (c) 4-DoF robot configuration, and (d) 2-DoF robot configuration with payload [14].

these three main contributions are presented, respectively, in Chapters 3, 4 and 5.

1.5 Outline of the Thesis

This thesis is organzied as follows: Chapter 2 provides a literature review and background for MRR platforms. The main control design problems and the corresponding works are discussed in the literature review section. The second part of Chapter 2 also presents background regarding MRRs and their kinematics, dynamics, and control Chapter 3 presents the proposed two-layer control architecture with a high level reconfiguration supervisor module. A systematic procedure for MRR system parameter identification is also discussed in this chapter. Based on the two-layer control architecture proposed in Chapter 3, two different adaptive control schemes are designed for trajectory tracking robust to reconfiguration effects, modeling uncertainties, and disturbances: An adaptive sliding mode control design presented in Chapter 4, and a model reference adaptive control design presented in Chapter 5. The test results for these designs are provided in Chapter 6. Chapter 7 concludes the thesis and provides potential future research directions.

Chapter 2

Background and Literature Review

2.1 Background

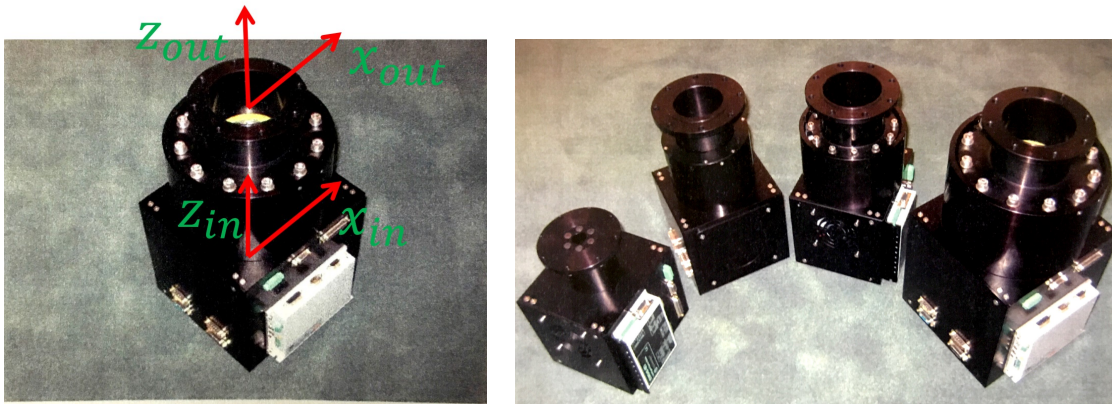
2.1.1 MRR Mechanical Components

Proper assembly of mechanical components of an MRR is important to improve the efficiency of the entire robot operation. Types of mechanical components such as joints and links can be chosen differently to perform various robot movements and workspaces of interest. For example, revolute joints as shown in Fig. 2.1 can be used to produce rotational motion and rectangular links as shown in Fig. 2.2 can be used to allow multiple joint and link connections to enlarge the workspace of the robot.

Joint Modules

Joint modules consist of a motor and connecting interface parts. Revolute joint (1-DoF), prismatic joint (1-DoF), and spherical joint (3-DoF) are the commonly used joint type

modules. These joints are not only employed to generate robot motion but also used to interconnect the link components. These joints can be attached to the links through standardized connecting interfaces. Examples of some revolute joint modules are shown in Fig. 2.1.

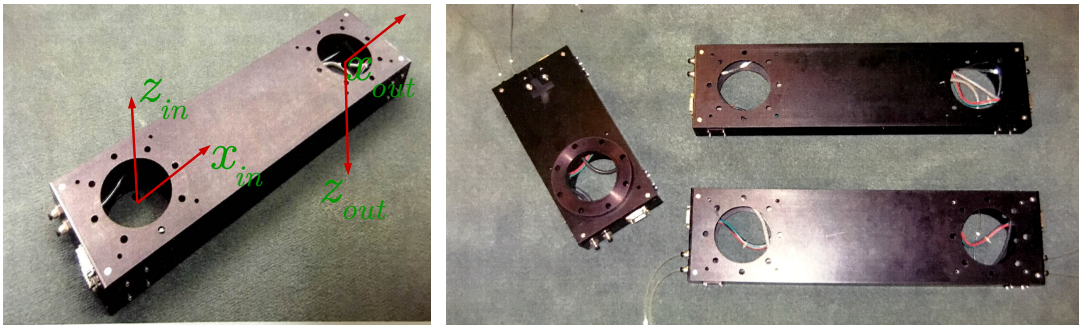


(a) (b)

Figure 2.1: (a) a revolute joint with its input and output coordinate frames (b) some revolute joints.

Link Modules

MRR link modules provide a connection between robot joints. Also, they can be exploited to adjust the range and contour of the robotic system's workspace. Different geometric shaped link modules are also used for enhancing adaptability of MRRs. Each link can also have multiple input and output ports and this enables the link for connecting with various joint and link modules. Some examples of link modules are demonstrated in Fig. 2.2.



(a) (b)

Figure 2.2: (a) a rectangular link with its input and output coordinate frames (b) some links.

2.1.2 MRR Kinematics

MRR platforms allow for having different robot configurations to realize several operations in many industries. However, it is time-consuming and challenging to manually enumerate all possible modular robot configurations as well as derive their corresponding dynamics in the equations of motion. Besides, motion control algorithms are mostly designed and tuned with the knowledge of the dynamics. Therefore, the derivation of the dynamical parameters after the modules are reassembled is essential and an ongoing research in the field [46]. In this subsection, first, the Denavit-Hartengerg (D-H) notations are briefly introduced to have robot manipulators' kinematic parameters. Then, the position notations are presented to derive the corresponding robot dynamics.

The forward kinematics for an open kinematic chain is usually acquired from the multiplication of the homogeneous transformation matrices achieved by compound transformations between frames of the adjacent links. In this thesis, Denavit-Hartengerg (D-H) notation formulated by [15], [16], is considered as seen in Fig. 2.3. Besides, the correspond-

ing D-H parameters are also used to generate the dynamic parameters applied to construct the dynamical terms of the system model for system parameter identification and control design purposes.

The D-H notations introduced in [15], [19], [17], [18] are used to have the kinematic parameters for each link by attaching the coordinates frames to the link components. The frame of each link is indexed by a number according to the link to which it is attached. The origin of fixed frame $\{i\}$ attached to link i with axes x_i, y_i, z_i is positioned at the point of intersection of joint axis i and the common normal between joint axes $i-1$ and i. The axis z_i is aligned with the joint axis i, the axis x_i points along the common normal $v_i^{\perp}$ between joint axes $i-1$ and i in the direction from joint i to joint $i+1$, and the axis y_i is determined by the right-hand rule to complete frame $\{i\}$ as shown in Fig. 2.3.

The origin of the frame $\{i\}$, which is named by O_i, is placed at the point of intersection of $(i+1)$th joint axis with $v_i^{\perp}$. It is also pointed out that x_i is normal to the plane of z_{i-1} and z_i with an arbitrary direction when two intersecting joint axes $i-1$ and i. Also, in the case where joint axes $i-1$ and i are parallel, one of the common normal lines is chosen as x_i arbitrarly, but the one that makes $d_i = 0$ is preferable. Besides, the origin of frame $\{i\}$ is chosen arbitrarly, and x_i is also chosen normal to the joint axes arbitrarily in the case where joint axes $i-1$ and i coincide.

MRR configurations can be described kinematically by employing four different quantities for each link. The corresponding kinematic parameters of i-th link are denoted by $a_i, d_i, \alpha_i, \theta_i$. a_i and α_i define the link itself, and other two kinematic parameters describe the link's connection to the adjacent link. These four link parameters are depicted as follows: the link length a_i is the length of the common normal v_i between z_{i-1} and z_i, the link offset d_i denotes the signed distance from x_{i-1} to x_i measured along $+z_{i-1}$, the twist angle α_i is the (right-hand) angle between axis z_{i-1} and z_i measured about $+x_i$, and the

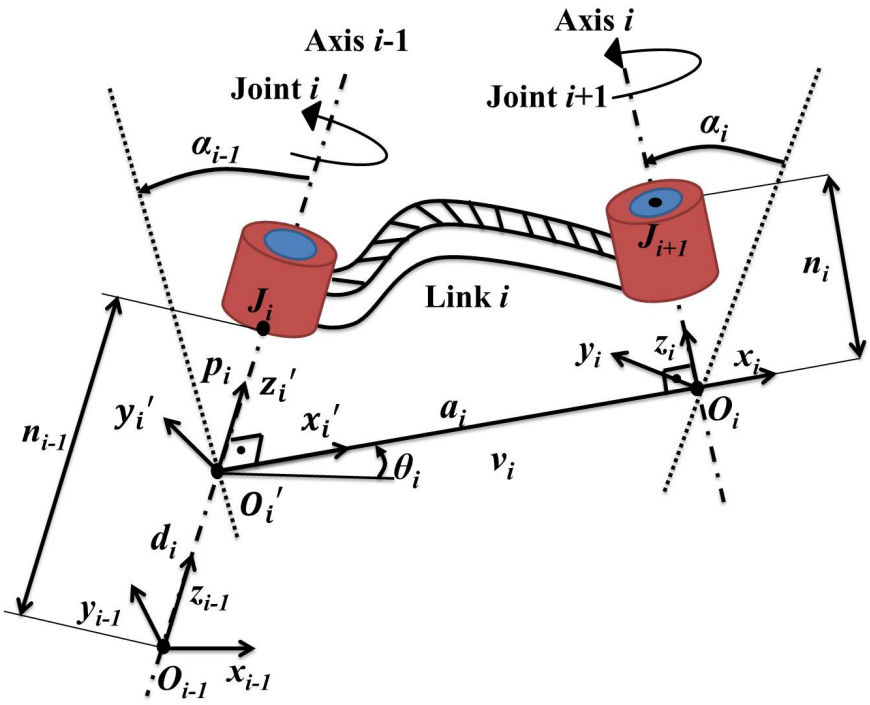


Figure 2.3: Denavit-Hartenberg notations related to link i with joint i at what end and joint $i+1$ at the other end [15].

joint angle θ_i is the (right hand) angle between x_{i-1} to x_i measured about $+z_{i-1}$. It is also pointed out that there always exist $a_i \geq 0$; however, α_i, d_i and θ_i are signed quantities. J_i and J_{i+1} denote joint connection points of link i and link $(i+1)$.

Moreover, p_i and n_i are required for obtaining the derivation of d_i for each link where p_i denotes the signed distance from $O_i^{'}$ to J_i along z_{i-1}; n_i is the signed distance from O_i to J_{i+1} measured along z_i. As shown in Fig. 2.3, based on the corresponding kinematic

parameters, d_i is computed recursively as follows:

$$d_i = n_{i-1} - p_i \quad \text{(revolute joint)},$$
$$d_i = n_{i-1} - p_i + q_i \quad \text{(prismatic joint)}$$

where q_i is the i-th joint displacement. If link i starts with a prismatic joint d_i is called the joint variable and d_i is a constant link parameter if link i starts with a revolute joint.

The D-H notations to characterize each arbitrary MRR module is considered in this thesis [17], [19] , [18]. The following procedure is employed to acquire the extended D-H parameters for each module used in an MRR configuration. The definitions of the kinematic parameters $(a_i, \alpha_i, p_i, n_i)$ are addressed in Chapter 2. The general module representation is depicted in Fig. 2.4 and i-th connection between output part of i-th link and input part of $(i+1)$th link is demonstrated in Fig. 2.5. The MRR module consists of joint and link components and it is separated into an input (in) part and output (out) part, as demonstrated in Fig. 2.4(b) and Fig. 2.4(c), respectively.

Two reference frames O_{in} and O_{out} are used for the input link and the output link of the module, respectively. These two frames are attached to the connectors of the input and output ports of the link module, as shown in Fig.2.4(b) and Fig.2.4(c) respectively. z_{in} is located to be pointed inwards the axis of the input link connector and z_{out} is set to be pointed outwards from the axis of the output link connector. The axes x_{in} and x_{out} are located at the connection surfaces of the input and output ports with a standardized direction. To finalize the input and output frames, y_{in} and y_{out} are determined by the right-hand rule. Four parameters are then defined for the input link part as a_{in}, α_{in}, p_{in}, n_{in}, and for the output link part as a_{out}, α_{out}, p_{out}, n_{out}. Then, two auxiliary frames O'_{in} and O''_{in} with reference to the input port are defined as shown in Fig. 2.4(b).

The origin of the first frame O'_{in} is attached to the intersection of input link connector

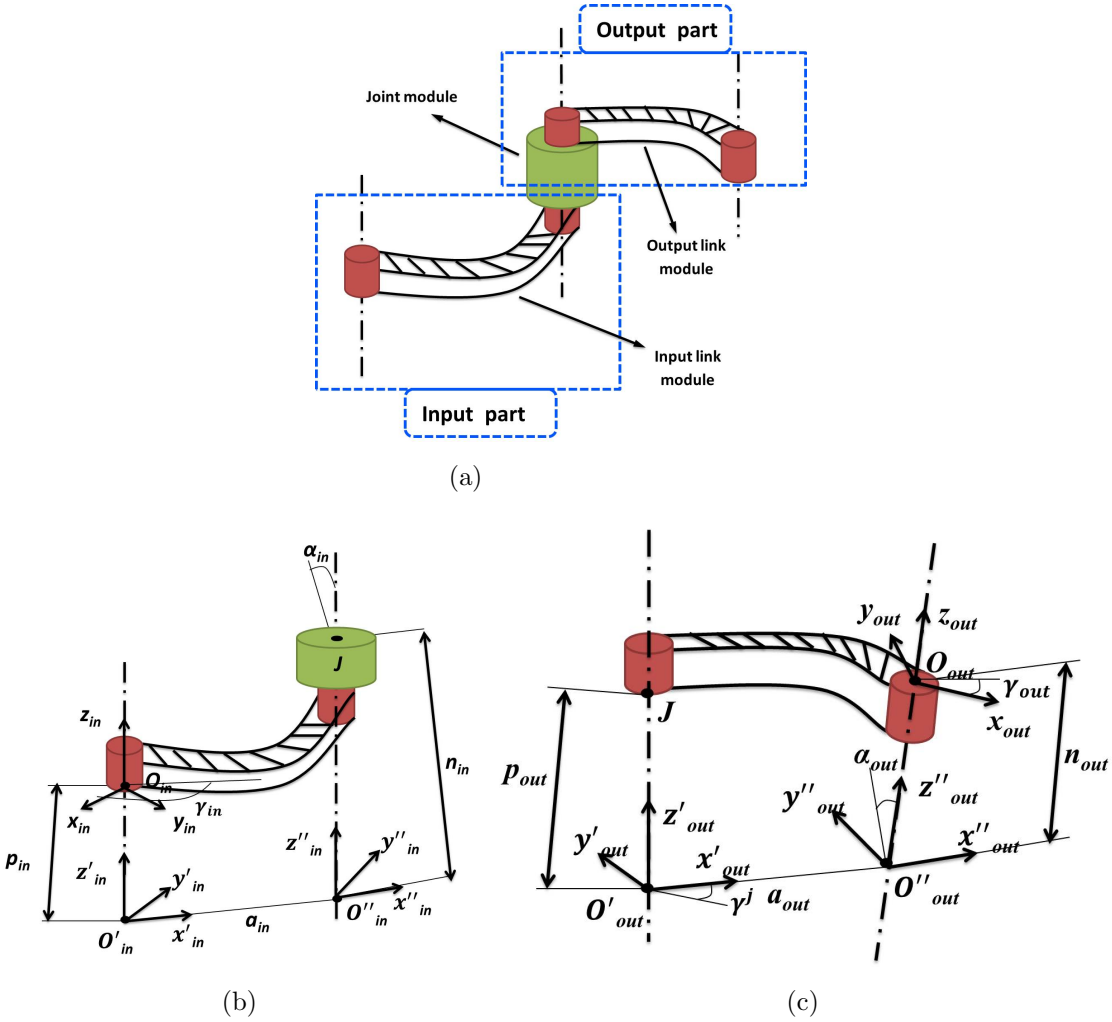


Figure 2.4: (a) The general module representation, (b) The input part of the module, and (c) The output part of the module [16], [17], [18].

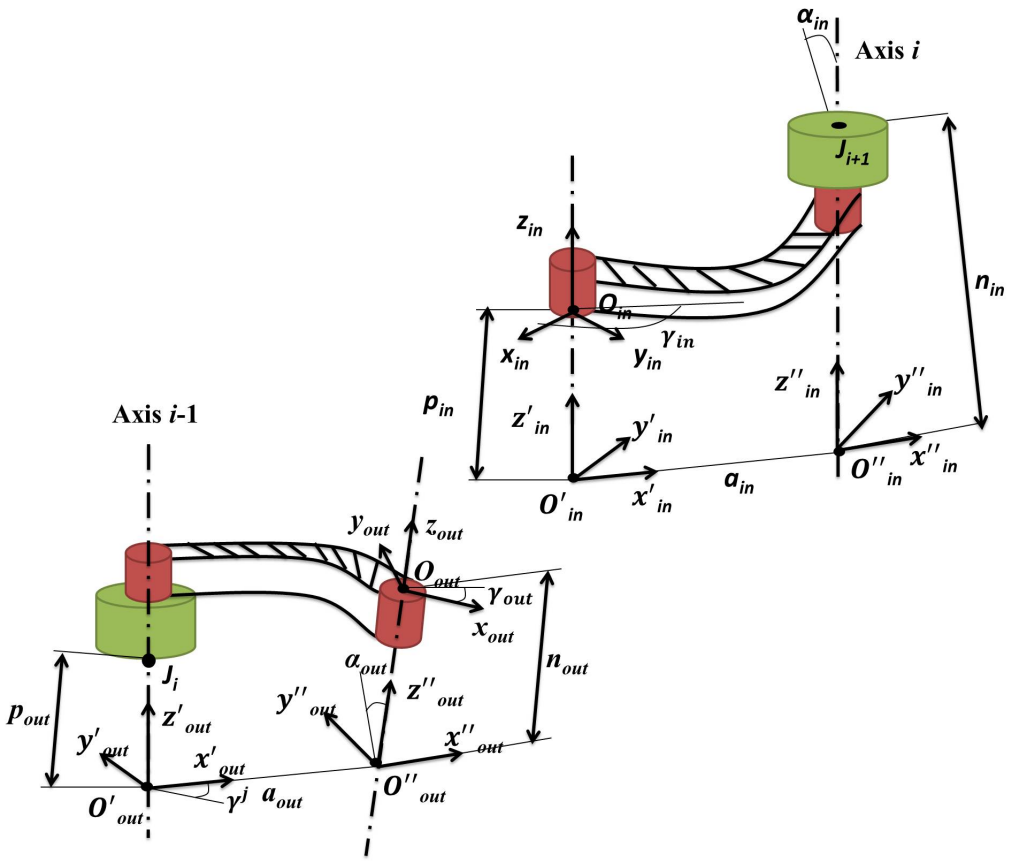


Figure 2.5: Representation of a connection between the output part of the module i and input part of the module $i+1$ including the D-H notations [19], [16], [17], [18].

axis with the common normal between input link connector and joint axes. The axis of $x_{in}^{'}$ is placed along the common normal in direction from $z_{in}^{'}$ to $z_{in}^{''}$, $z_{in}^{'}$ is aligned with the axis of z_{in}, and $y_{in}^{'}$ is fixed by the right-hand rule to complete the frame. The axis $z_{in}^{''}$ is aligned with the rotation axis of the joint, $x_{in}^{''}$ points along the common normal between the axes $z_{in}^{'}$ and $z_{in}^{''}$ in direction from $z_{in}^{'}$ to $z_{in}^{''}$. Then, $y_{in}^{''}$ is determined by the right-hand rule to complete the frame. The parameters of the input link part of the module are defined as follows: a_{in} is length of the common normal between $O_{in}^{'}$ and $O_{in}^{''}$; α_{in} denotes the angle from the axis z_{in} to the joint axis about $x_{in}^{''}$; p_{in} defines distance from $O_{in}^{'}$ to the input connection frame O_{in} along the z coordinate; and n_{in} denotes the distance from $O_{in}^{''}$ to the joint connection point J measured along the z coordinate. Same notations and description are also applied to the output part of the module.

2.1.3 MRR Dynamics

The equations of motion of MRRs describe the relationship between the motion of links (position, velocities, and accelerations) and applied torque/force at the joints, while considering link masses and mass-moment of inertia. The Lagrangian and recursive Newton-Euler (N-E) formulations are mostly used for deriving the dynamic model of a mechanical system. In this thesis, the recursive N-E method for obtaining the required dynamic parameters is used and it analyzes an MRR system module by module to arrive at a description of the robot as a whole. This recursive approach is not only applied to obtain the closed-form dynamic model, including symbolic computation but also practiced for designing the control algorithm with algebraic calculation [15].

Position notations of each module such as a center of mass position and distance between two consecutive parallel joint axes are introduced briefly to generate the required dynamics.

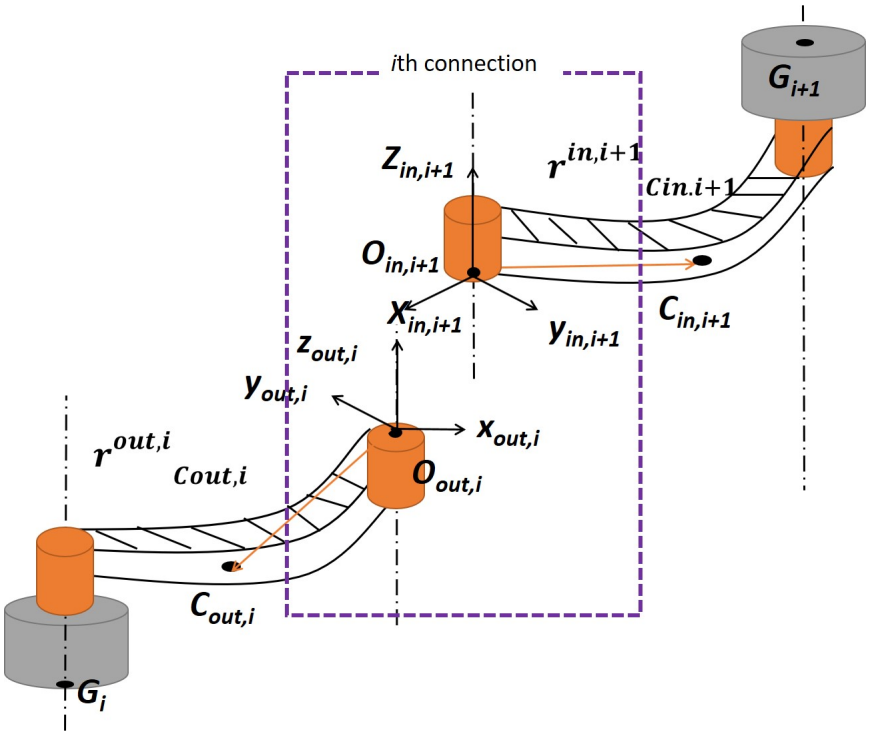


Figure 2.6: Representation of the connection between the output part of i-th link module and the input part of $(i+1)$-th link module, including the dynamic parameters [18].

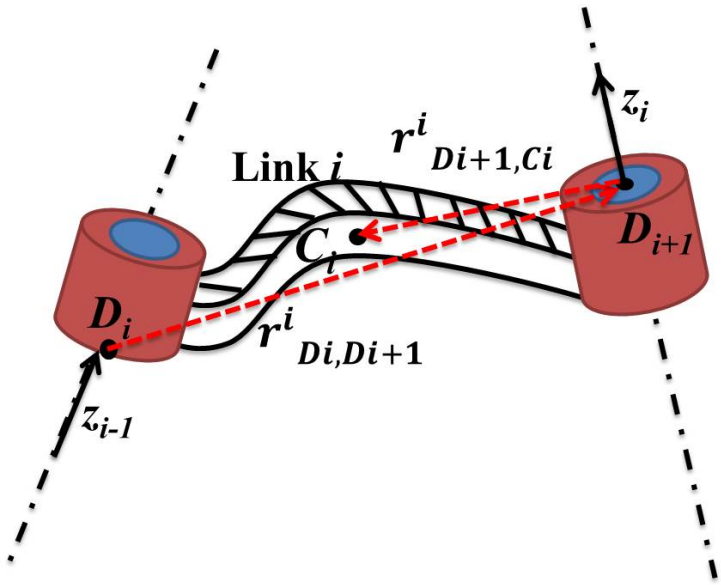


Figure 2.7: Position parameters associated with link i [15].

The connection between two joints axes i and $i+1$ with i-th link is depicted in Fig. 2.7. According to Fig. 2.7, the vector from D_i to D_{i+1} for i-th link is denoted by $r^i_{D_i,D_{i+1}}$ and its center of mass vector from D_{i+1} to C_i is denoted by the vector $r^i_{D_{i+1},C_i}$ [15].

In this thesis a methodology to have the corresponding dynamic parameters after the robot is reconfigured is introduced according to [15], [18]. Similarly to the kinematic approach introduced in the previous subsection, the input and output frames are set to describe the dynamic parameters. Assuming that i-th connection O_i is realized between i-th and $(i+1)$-th modules. Fig. 2.6 shows a graphical representation of the i-th module connection between the modules i and $i+1$, including their dynamic parameters. Then, the general i-th connection of the inertia tensor can be found by $I_i^{out} + I_{i+1}^{in} = I_i^{O_i}$ where I_i^{out} is i-th output link inertia and I_{i+1}^{in} denotes $(i+1)$-th input link inertia as shown in

Fig. 2.8. The center of mass of the i-th module connection is given by

$$r_i^{O_i,c_i} = \frac{m_i^{out} r_{c,i}^{out} + m_{i+1}^{in} r_{c,i+1}^{in}}{m_i}, \tag{2.1}$$

where m_{i+1}^{in} and m_i^{out} denote $(i+1)$-th input link mass and i-th output link mass, respectively, $m_i = m_i^{out} + m_{i+1}^{in}$ is the total mass of the i-th connection point of the modules, $r_{c,i+1}^{in}$ denotes the coordinates of the center of $(i+1)$-th input link mass and $r_{c,i}^{out}$ represents the coordinates of the center of i-th output link mass. Then, the center of mass of the connection c_i relative to joint G_{i+1} is found by

$$\begin{bmatrix} r_i^{G_{i+1},c_i} \\ 1 \end{bmatrix}_{4\times 1} = \left[T_i^{O_i,G_{i+1}}\right]_{4\times 4}^{-1} \begin{bmatrix} r_i^{O_i,c_i} \\ 1 \end{bmatrix}_{4\times 1}. \tag{2.2}$$

where $T_i^{O_i,G_{i+1}}$ denotes the homogeneous transformation matrix used for geometric description the input part of $(i+1)$-th module. The inertia tensor of i-th connection relative to G_{i+1} is given by

$$I_i^i = R_i^{(O_i,G_{i+1})^{\mathsf{T}}} [I_i^{O_i} - m_i \mathbf{S}^{\mathsf{T}}(r_i^{O_i,c_i}) \mathbf{S}(r_i^{O_i,c_i})] R^{O_i,G_{i+1}} \tag{2.3}$$

where $\mathbf{S}(\bullet)$ is an a skew-symmetric matrix.

2.2 Modeling MRR Dynamics

The dynamic model of the i-th joint of an n-module MRR system ($i \in \{1,\ldots,n\}$) can be formulated in decentralized form as [47]

$$\begin{aligned} \mathsf{S}_i : \tau_i(t) &= M_i\Big[q_i(t)\Big]\ddot{q}_i(t) + N_i\Big[q_i(t),\dot{q}_i(t)\Big]\dot{q}_i(t) + G_i[q_i] \\ &+ \tau_{iu}\Big[t,q(t),\dot{q}(t),\ddot{q}(t)\Big] + \tau_{ie}(t), \end{aligned} \tag{2.4}$$

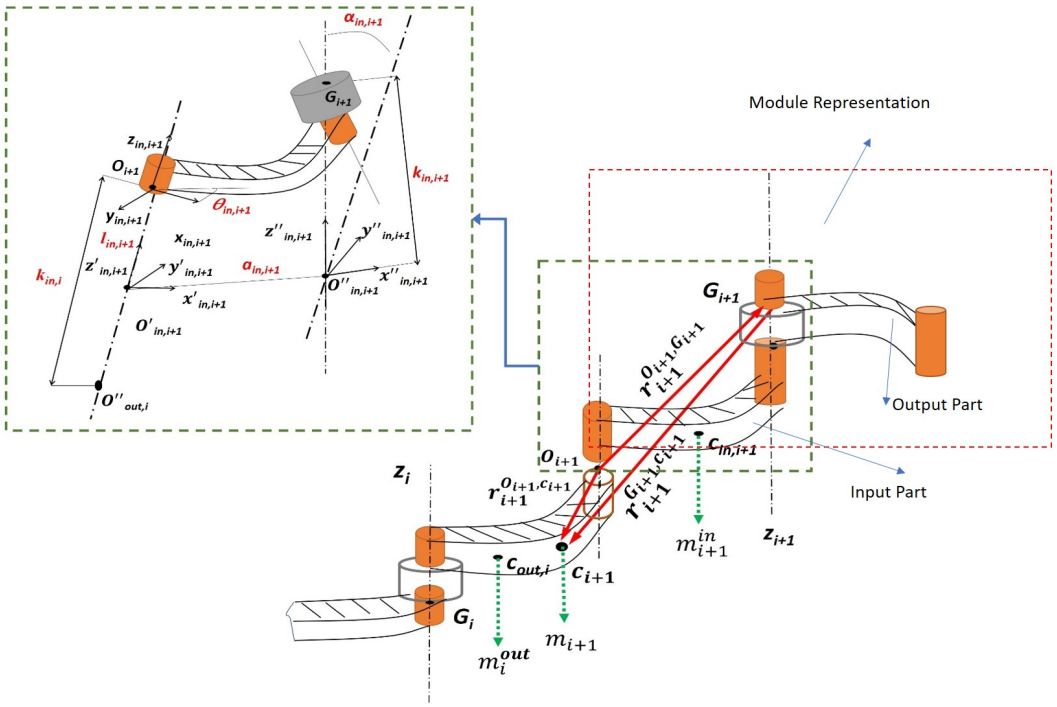


Figure 2.8: Representation of the connection between the output part of i-th link module and the input part of $(i+1)$-th link module, including the kinamatic and dynamic parameter notations [17], [18], [20].

where $t \geq 0$ denotes time; $q = [q_1, \ldots, q_n]^\mathsf{T} \in \mathbb{R}^n$, $\dot{q}(t) = [\dot{q}_1, \ldots, \dot{q}_n]^\mathsf{T} \in \mathbb{R}^n$ and $\ddot{q}(t) = [\ddot{q}_1, \ldots, \ddot{q}_n]^\mathsf{T} \in \mathbb{R}^n$ are the vectors of joint position, velocity and acceleration variables; with each entry $q_i(t), \dot{q}_i(t), \ddot{q}_i(t)$ corresponding to joint i, respectively. $\tau_i(t)$ is the torque applied to joint i; $M_i[q_i(t)]$ represents the i-th diagonal entry of the inertia matrix $M[q(t)] : \mathbb{R}^n \longmapsto \mathbb{R}^{n\times n}$; $N_i[q_i(t), \dot{q}_i(t)]$ is the i-th diagonal entry of the Coriolis and centrifugal force matrix $N[q(t), \dot{q}(t)] : \mathbb{R}^n \times \mathbb{R}^n \longmapsto \mathbb{R}^{n\times n}$; $G_i[q_i]$ is the i-th entry of the gravity vector $G[q(t)] : \mathbb{R}^n \longmapsto \mathbb{R}^n$; $\tau_{iu}(q, \dot{q}, \ddot{q})$ represents the i-th entry of unmodeled dynamics vector $\tau_u[t, q, \dot{q}, \ddot{q}]$ which includes cross coupling effects between modules and friction term; $\tau_{ie}(t)$ denotes the i-th entry of the external disturbance vector $\tau_e(t)$; $\tau_i(t)$ is the i-th entry of the control input vector $\tau(t)$. The dynamic model (2.4) can be written in a vector form as

$$\begin{aligned} \mathsf{S} : &\bar{M}\Big[q(t)\Big]\ddot{q}(t) + \bar{N}\Big[q(t), \dot{q}(t)\Big]\dot{q}(t) + G\Big[q(t)\Big] \\ &= \tau(t) - \underbrace{(\tau_u[t, q, \dot{q}, \ddot{q}] + \tau_e(t))}_{\kappa}, \end{aligned} \tag{2.5}$$

where $\bar{M}[q(t)] = \text{diag}(M_1[q_1], \ldots, M_n[q_n])$ denotes the diagonal inertia matrix, $\bar{N}[q, \dot{q}] = \text{diag}(N_1[q_1, \dot{q}_1], \ldots, N_n[q_n, \dot{q}_n])$ is the diagonal Coriolis and centrifugal force matrix, $G[q(t)] = [G_1[q_1], \ldots, G_n[q_n]]^\mathsf{T}$ denotes the gravity vector. Note that these diagonal matrices are different from the original M and N matrices. The term represents the unmodeled dynamics and disturbance, is captured by

$$\begin{aligned} \kappa = [\kappa_1, \cdots, \kappa_n]^\mathsf{T}, \quad \kappa_i(t, q, \dot{q}, \ddot{q}) = &\sum_{j=1, j\neq i}^{n} M_{ij}(q)\ddot{q}_j \\ + &\sum_{j=1, j\neq i}^{n} N_{ij}(q, \dot{q})\dot{q}_j + F_i(q_i, \dot{q}_i) + \tau_{ie}, \end{aligned} \tag{2.6}$$

where $F_i(q_i, \dot{q}_i)$ denotes the i-th entry of the friction vector $F[q, \dot{q}]$, which is assumed to be bounded.

2.3 Literature Review

In recent decades, the control design problems of modular and reconfigurable robots (MRRs) have attracted the attention of many researchers and industry. Decentralized and centralized schemes are commonly used for the control design problems of MRRs as they are also applied to the standard robot manipulators. A comprehensive summary of such classical control methods for robot manipulators can be found in [22].

Most published research papers regarding MRR control problems have focused on decentralized schemes. These works are mostly developed by model-free Proportional, Integral, and Derivative (PID) control designs. Such works are opted for avoiding the dynamic parameters uncertainties after the reconfiguration. The decentralized control schemes can achieve satisfactory control Position error in the case of employing light-weight modules where the resultant coupling forces can be assumed as minimal. However, these model-free algorithms do not guarantee the complete stability of MRRs when their modules are reassembled into a new configuration. Besides, to achieve global stability, the methods that improve the simple PID algorithms with nonlinear terms have also been studied. However, most of these methods do not guarantee stability in presence of uncertain dynamics parameters.

Obtaining the dynamical terms and the bounds on the norm of the corresponding dynamics are challenging for MRRs whose morphology and work environment are often changing. Dynamics model parameters and the knowledge of the bounds are obtained using the algorithm proposed in Chapter 3 of this thesis. This will enable to design decentralized adaptive controllers which can overcome parametric uncertainties, guarantee stability and achieve desired tracking Position error. The literature works regarding MRR control schemes is summarized in the next subsection.

2.4 MRR Control Schemes

Several control schemes for MRRs have been developed in recent decades. A decentralized joint torque-sensing based decomposition methodology is introduced to handle the unmodeled dynamics and the dynamic parameter uncertainties [48]. A neuro-fuzzy control (NFC) design is presented for MRR systems by [45], whose neural network control design parameters are estimated on-line, but the fuzzy parameters off-line. The authors [45] introduced a distributed PID control design architecture based on a fuzzy gain tuning principle. This control architecture needs no knowledge of the MRR dynamic parameters to overcome the MRR regulation control problem.

Another control scheme was designed by [49], where a modular distributed control is utilized based on joint torque sensors. [50] considers the full-body of the MRR system as a collection of interconnected subsystems. The control scheme [50] is constructed based on a fuzzy control law to eliminate the interconnection/coupling terms. [51] introduces a type of decentralized control architecture, whose design is based on the robust backstepping procedure. Since the control design in [51] is not depending on the robot configuration, this control concept enables quick reconfigurability of the manipulator. Besides, the authors in [14] design an interval type-2 Takagi-Sugeno-Kang fuzzy logic controller (IT2 TSK FLC) for compensating the uncertainties and varying dynamics. Moreover, the authors in [19] introduce a centralized robust control scheme based on interval arithmetic that allows the MRR motions by using global asymptotic stability with the knowledge of the MRR dynamics.

Some of the proposed decentralized motion control designs, such as [14], [45], [48], [51] are tuned and developed without the dynamics knowledge of the final configuration that can lead to instabilities [19]. Furthermore, the corresponding schemes usually depend on

the minimum/maximum bound of each configuration dynamics parameters such as mass, inertia, and gravity term. To overcome aforementioned control challenges, a decentralized adaptive sliding mode control scheme that guarantees robustness in presence of uncertainties and disturbances for each new robot configuration is developed in this thesis. Besides, the proposed control algorithm can generate the required minimum/maximum bound of dynamic parameters which are used for control design after each reconfiguration to guarantee globally uniformly ultimately bounded (GUUB) of the MRR system trajectories in terms of robust stability despite changes in the system dynamics.

2.5 Proposed Two-Layer Control Architecture

A two layer control architecture consisting of high-level and low-level modules is developed. The high-level module accounts for the dynamic parameters changes and reconstructs the parametric model used in the on-line parameter identification process with the projection after the robot configuration is changed. The low-level control module consists of an adaptive algorithm provided by a recursive least squares (RLS) based on-line parameter estimator for the parametric uncertainties. Furthermore, a robust adaptive term is added into this low-level controller to compensate for the unmodeled dynamics and disturbance. This control architecture also covers the robust stability of the uncertain MRR system, and it proves globally uniformly ultimately boundedness of the MRR trajectories in terms of robust stability despite the parametric uncertainty, unmodeled dynamics, changes in the system dynamics, and disturbance.

Chapter 3

Proposed Two-Layer Control Architecture and The High Level Module

3.1 Proposed Two-Layer Control Architecture

In this thesis, a two layer control architecture consisting of high-level and low-level modules is developed as shown in Fig. 3.1. The high-level module accounts for the dynamic parameters changes and reconstructs the parametric model used in the on-line parameter identification process with the projection after the robot configuration is changed. The low-level control module consists of an adaptive algorithm provided by a recursive least squares (RLS) based on-line parameter estimator for the parametric uncertainties. Furthermore, a robust adaptive term is added into this low-level controller to compensate for the unmodeled dynamics and disturbance. This control architecture also covers the

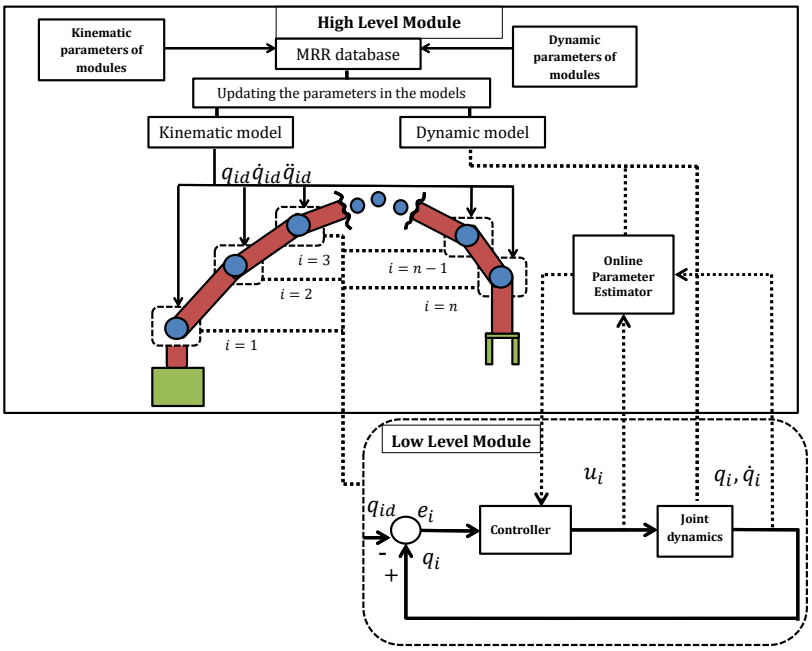


Figure 3.1: Illustration of the two layer control architecture composed of a high-level and low level modules.

robust stability of the uncertain MRR system, and it proves globally uniformly ultimately boundedness of the MRR trajectories in terms of robust stability despite the parametric uncertainty, unmodeled dynamics, changes in the system dynamics, and disturbance.

3.2 The High Level Module

The high-level module of the two-layer control architecture in Figure 3.1 is developed to update dynamic parameters in the equations of motion from the physical parameter knowledge of the MRR modules after the robot is reconfigured. The homogeneous transformations and recursive Newton-Euler (N-E) methods are used for the proposed remodeling procedure. The high-level module remodels MRR dynamic parameters from a database that includes the kinematic parameters such as the link length and twist angle, and the dynamic parameters such as module mass and center of module mass [16], [17], [18]. First, the reaction force/torque of i-th MRR module is derived to find the general dynamics modeling of the robot configuration for the system identification and control purposes. After generating the required dynamics parameters such as $r_i^{G_{i+1},c_i}$ and I_i^i, Algorithm 1 is introduced to obtain the initial sets that are required for the dynamic model generation based on two sets of recursions: the forward and backward. The algorithm consisting of the initial parameters is proposed as follows:

Algorithm 1 Initial part for Recursive N-E algorithm [52]

Input: q, $\dot{q}, \ddot{q}$, T^{G_{i-1},G_i}, g

Output: $R^{G_i,G_{i-1}}$, k_i, $r_i^{G_{i-1},G_i}$

1: Set the D-H parameters of the available modules

2: Set ω_0^0, $\dot{\omega}_0^0$, $a_0^0 \leftarrow 0$ and $k_0 \leftarrow [0,0,1]^\mathsf{T}$

3: Set gravity effect $a_0^0 \leftarrow a_0^0 - g$

4: **for** $i = 1$ *to* n **do**

5: $\quad R^{G_{i-1},G_i} \leftarrow T^{G_{i-1},G_i}$ and q_i

6: $\quad R^{G_i,G_{i-1}} \leftarrow$ transpose (R^{G_{i-1},G_i})

7: $\quad k_i \quad \leftarrow R^{G_{i-1},G_i} k_0$

8: $\quad r_i^{G_{i-1},G_i} \leftarrow T^{G_{i-1},G_i}$ and q_i

9: **end for**

where T^{G_{i-1},G_i} denotes the homogeneous transformation matrix that maps from $(i-1)$-th joint module to i-th joint module, ω_0^0, $\dot{\omega}_0^0$, and a_0^0 denote the initial angular velocity, angular acceleration, and linear acceleration, respectively. The forward and backward recursion algorithms are then introduced to model the overall MRR dynamics as follows:

Algorithm 2 Forward recursion of N-E algorithm [52]

Input: q, $\dot{q}, \ddot{q}$, ω, $R^{G_i,G_{i-1}}$, g, k_i, $r_i^{G_{i-1},G_i}$

Output: $a_i^{c,i}$

for $i = 1$ *to* n **do**
  if i-th joint is revolute **then**
    $\omega_i^i \leftarrow R^{G_i,G_{i-1}}(\omega_{i-1}^{i-1} + \dot{q}_i k_0)$
    $\dot{\omega}_i^i \leftarrow R^{G_i,G_{i-1}}(\dot{\omega}_{i-1}^{i-1} + \ddot{q}_i k_0 + \omega_{i-1}^{i-1} \times \dot{q}_i k_0)$
    $a_i^i \leftarrow R^{G_i,G_{i-1}} a_{i-1}^{i-1} + \dot{\omega}_i^i \times r_i^{G_{i-1},G_i} + \omega_i^i \times (w_i^i \times r_i^{G_{i-1},G_i})$
  end if
  $a_i^{c,i} \leftarrow a_i^i + (\dot{\omega}_i^i \times r_i^{G_i,c_i}) + \left(\omega_i^i \times (\omega_i^i \times r_i^{G_i,c_i})\right)$
end for

Algorithm 3 Backward recursion of N-E algorithm [52]

Input: q, $\dot{q}, \ddot{q}$, ω_0^0, $R^{G_i,G_{i-1}}$, g, k_i, $r_i^{G_{i-1},G_i}$, $a_i^{c,i}$, m

Output: f_i^i, n_i^i

Set f_{n+1}^{n+1}, n_{n+1}^{n+1}, and $R^{G_i,G_{i+1}}$
for $i = n$ *to* 1 **do**
  $L_i^i \leftarrow m_i a_i^{c,i}$
  $f_i^i \leftarrow (R^{G_i,G_{i+1}} f_{i+1}^{i+1}) + L_i^i$
  $n_i^i \leftarrow (R^{G_i,G_{i+1}} n_{i+1}^{i+1}) + (r_i^{G_i,c_i} \times L_i^i) + (r_i^{G_{i-1},G_i} \times f_i^i) + (I_i^i \dot{\omega}_i^i) + \left(\omega_i^i \times (I_i^i \omega_i^i)\right)$
  $\tau_i = n_i^{i\mathsf{T}} k_i$
end for

where n_i^i is i-th acting reaction force, τ_i denotes the generalized forces on i-th module. The forward recursion transforms the kinematic variables from the first frame to the last

frame, and the backward recursion algorithm transforms the forces and moments from the end-effector to the base, and culminate with the calculation of the joint torques/forces. Now, the generalized force for MRR configurations is generated. It is pointed out that $r_i^{G_{i+1},c_i}$ and I_i^i are obtained for each MRR configuration by updating module kinematic and dynamic parameters. Therefore, the dynamical terms in the equations of motion can be updated for system identification and control design purposes. With the help of the high level module, a system identification scheme is designed for an MRR platform to estimate its unknown dynamic parameters after the robot modules are reassembled. This scheme is consisting of an algorithm that constructs a parametric model for the MRR configuration and a modified recursive least squares (RLS) algorithm that estimates the unknown inertial parameters based on the changes in the system dynamics. Moreover, the corresponding dynamic parameters are employed to find the required bounds on the norms of the dynamical terms. First, the required nominal dynamic parameters for the system model are computed to adjust the parametric model after the reconfiguration. Then, the RLS algorithm is applied in order to estimate the dynamic parameters that is also used in the adaptive control algorithm to compensate for changes in the system dynamics because of the robot reconfiguration and deployment of a payload. The methodology for on-line identification of the MRR dynamics is introduced in the following subsection.

3.3 MRR System Parameter Identification

The performance of a low-level controller depends on the precision of the dynamical terms in the equations of motion. The uncertain dynamic model arises from the inertial uncertainty, such as coordinates of the centers of mass and inertia tensors. In practice, the inertial parameters are produced by the manufacturer and computed using computer-aided design

(CAD) software. However, those values might be inexact due to some reasons. For example, the gear ratio and motor inertia can offset the ground values of the inertia parameters. The CAD data might include neither mass parameters of wiring and possibly light segment electronics nor items that are results of hardware evolution. Also, embedded PC, cameras, and grippers can also change the mass distribution of the robot's rigid body.

Besides, the mass distribution may be affected during operation in case of the end-effector that grabs a heavy object. It is well known that the last module dynamic parameters cannot be known a priori in case that the robot grasps unknown payload. Particularly with a payload in hand, the uncertainty problem becomes more complicated, and the dynamic parameters may vary significantly from the nominal conditions. Therefore, MRR system should handle the changing loads, model uncertainties, and unexpected changes in their sensor or actuator behaviors.

In this chapter, an on-line parameter identification scheme that provides a framework for system identification and control design of MRRs is introduced. This scheme estimates the unknown dynamic parameters based on the measurable input and output data when the robot performs the desired motion task in hand after each reconfiguration. This high-level module is also synthesized with the low-level adaptive sliding mode control design. Besides, the dynamic parameters have physical constraints that need to be considered to have more accurate estimations. The identification scheme based on regression procedures can produce nonphysical estimates. The use of the dynamic model with a nonpositive inertial matrix leads to an unstable system. For having meaningful estimation results, minimizing a cost function with the constraints is introduced. The proposed scheme for this identification problem consists of the following steps:

- construct a parametric model for system identification according to the modules used

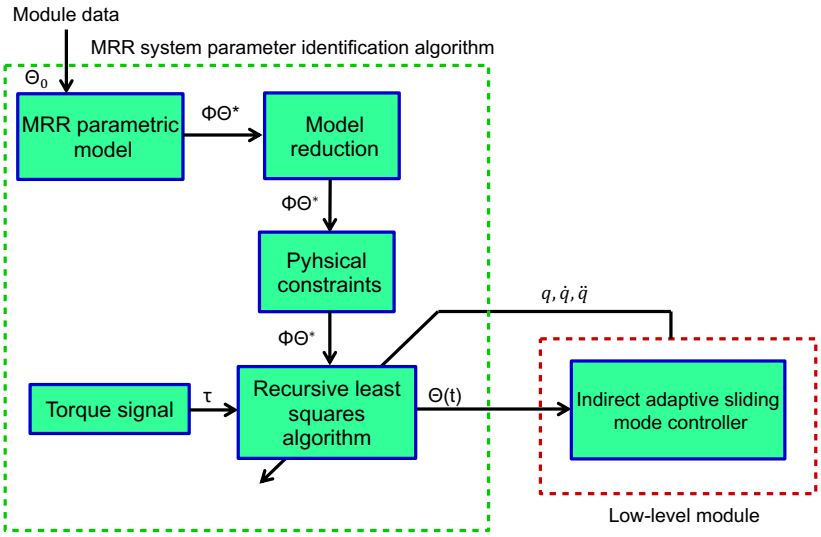


Figure 3.2: Indirect adaptive control structure.

in an MRR configuration,

- reduce the inertial parameters vector Θ_d set to base parameters Θ,
- consider the physical restrictions to identify the dynamic parameters in a more realistic way,
- estimate the inertial parameter vector Θ on-line by a modified recursive least-squares (RLS) algorithm used in the low-level adaptive sliding mode control algorithm.

The indirect adaptive controllers, like those of [53], [54], [22] use prediction errors on the joint torques to generate parameter estimates to be used in the control law. The joint position $q(t)$ can be conveniently measured by encoders at the joints, and the joint torque $\tau(t)$ is available from torque sensors utilized for parameter estimation. Given the available

measurements $\tau(t)$ and $\Phi(q, \dot{q}, \ddot{q})$ at each time t generate estimation of the unknown vector Θ^*, $\Theta(t)$. The algorithm updates $\Theta(t)$ and $\Theta(t)$ approaches to Θ^*. Since online parameter identification model is utilized for adaptive control design purpose, the adaptive algorithm updates the estimate $\Theta(t)$ if Θ^* changes.

3.3.1 MRR System Parametric Model

The first step is contructing a parametric model. The unknown dynamic parameters are lumped in a vector Θ^* and separated from measurable torque and joint position signals. For all available $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$, if $\kappa = 0$ then Eq. (2.5) can be written in linear parametric form

$$\tau = M(q)\ddot{q} + N(q, \dot{q})\dot{q} + G(q) = \Phi_d(q, \dot{q}, \ddot{q})\Theta_d^*, \tag{3.1}$$

where $\Phi_d(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n \times 10n}$ is a known regression matrix and $\Theta_d^* = [\Theta_{1d}^*, \cdots, \Theta_{nd}^*]^\mathsf{T} \in \mathbb{R}^{10n}$ is a certain parameter vector. Above $\Theta_{id}^* \in \mathbb{R}^{10}$ is considered to be in the form

$$\begin{aligned}\Theta_{id}^* =& [m_i, m_i r_{ix}^{O_i,c_i}, m_i r_{iy}^{O_i,c_i}, m_i r_{iz}^{O_i,c_i} \\ & I_{ixx}^{Oi}, I_{ixy}^{Oi}, I_{ixz}^{Oi}, I_{iyy}^{Oi}, I_{iyz}^{Oi}, I_{izz}^{Oi}]^T,\end{aligned} \tag{3.2}$$

where $(I_{ixx}^{Oi}, \cdots, I_{izz}^{Oi})$ are the elements of the inertia matrix $I_i^{Oi} \in \mathbb{R}^{3\times3}$, $(m_i r_{ix}^{O_i,c_i}, m_i r_{iy}^{O_i,c_i}, m_i r_{iz}^{O_i,c_i})$ are the elements of $m_i r_i^{O_i,c_i} \in \mathbb{R}^{3\times1}$, and m_i is a scalar value of i-th module mass.

3.3.2 Model Reduction

Not all the parameters in Θ_d^* of (3.1) will directly contribute to each MRR joint torque due to motion constraints imposed upon each robot link through the joints and other

constraints. The dynamic parameter vector, Θ_d^*, is reduced to a vector of minimal number of dynamic model parameters, $\Theta^* \in \mathbb{R}^b$. After the model reduction is applied, Eq. (3.1) is rewritten in reduced-order model form as

$$\tau = M(q)\ddot{q} + N(q,\dot{q})\dot{q} + G(q) = \Phi(q,\dot{q},\ddot{q})\Theta^*, \tag{3.3}$$

where $\Phi \in \mathbb{R}^{n\times b}$ is the reduced regressor matrix after obtained by model reduction algorithm and $\Theta^* \in \mathbb{R}^b$ is the corresponding unknown parameter vector. Note that the actual equivalent of (2.5) is obtained including uncertainties and disturbances in (3.3) as

$$\tau = M(q)\ddot{q} + N(q,\dot{q})\dot{q} + G(q) - \kappa = \Phi(q,\dot{q},\ddot{q})\Theta^* - \kappa. \tag{3.4}$$

3.3.3 MRR Parameter Estimation

Using the torque estimate to be used in adaptive control laws is generated as

$$\hat{\tau} = \Phi(q,\dot{q},\ddot{q})\Theta, \tag{3.5}$$

where Θ is the time varying estimate of Θ^* to be generated by an adaptive parameter estimation law whose design is presented in this section. Since Θ^* is unknown, the parameter estimation error $\tilde{\Theta}(t) = \Theta(t) - \Theta^*$ is not available for measurement. Instead, the measurable output error signal

$$\varepsilon = \frac{\tau - \hat{\tau}}{m_s^2} = \frac{\tau - \Phi\Theta}{m_s^2} \tag{3.6}$$

is utilized, where $m_s^2 \geq 1$ is a normalization signal designed to guarantee that $\dfrac{\Phi}{m_s}$ is bounded. A straightforward choice for m_s is $m_s^2 = 1 + \alpha\|\Phi^2\|$, $\alpha > 0$.

Adaptive Law

An RLS based adaptive law with parameter projection is designed to estimate the unknown parameters, solving the following minimization problem with constraints:

$$\begin{cases} \text{minimize} \quad J(\Theta) = \dfrac{1}{2}\int_0^t e^{-\beta(t-\chi)}\dfrac{[z(\chi)-\Phi(t)\Theta(\chi)]^2}{m_s^2(\chi)}d\chi \\ +\dfrac{1}{2}e^{-\beta t}(\Theta-\Theta_0)^{\mathsf{T}}Q_0(\Theta-\Theta_0) \\ \text{subject to:} \quad \Theta \in \mathcal{S}, \end{cases} \tag{3.7}$$

where $\mathcal{S}$ is a convex subset of $\mathbb{R}^b$ with smooth boundary almost everywhere, defining the feasible set for the entries of the reduce parameter vector Θ^*. The corresponding RLS based adaptive law with parameter projection is given by [55]

$$\dot{\Theta} = \Pr(P\Phi^{\mathsf{T}}\varepsilon) = \begin{cases} P\Phi^{\mathsf{T}}\varepsilon & \text{if } \Theta \in \mathcal{S}^0 \\ \quad \text{or if } \Theta \in \delta(\mathcal{S}) \text{ and } (P\Phi^{\mathsf{T}}\varepsilon)^{\mathsf{T}}\nabla g \le 0, \\ P\Phi^{\mathsf{T}}\varepsilon - P\dfrac{\nabla g \nabla g^{\mathsf{T}}}{\nabla g^{\mathsf{T}} P \nabla g}P\Phi^{\mathsf{T}}\varepsilon & \text{otherwise}, \end{cases} \tag{3.8}$$

$$\dot{P} = \begin{cases} \beta P - \dfrac{P\Phi^{\mathsf{T}}\Phi}{m_s^2}P & \text{if } \Theta \in \mathcal{S}^0 \\ \qquad \text{or if } \Theta \in \delta(\mathcal{S}) \text{ and } (P\varepsilon\Phi)^{\mathsf{T}}\nabla g \le 0 \\ 0 \qquad \text{otherwise}, \end{cases} \tag{3.9}$$

where $\delta(\mathcal{S}) = \{\Theta \in \mathbb{R}^b \mid g(\Theta) = 0\}$ and $\mathcal{S}^0 = \{\Theta \in \mathbb{R}^b \mid g(\Theta) < 0\}$ denote the boundary

and the interior, respectively, of $\mathcal{S}$, $\Theta(0) \in \mathcal{S}$, $P(0) = P_0 = P_0^{\mathsf{T}} \in \mathbb{R}^{b\times b}$. The adaptive law should also guarantee that the parameter estimate $\Theta(t)$ and the speed of adaptation $\dot{\Theta}$ are bounded. The estimation error ε should also get smaller with time. Under a suitable adaptive control design, the output of the system can track the desired signal, despite the fact that the parameter error vector $\tilde{\Theta}(t)$ may not converge to zero in case of the non-persistent excitation signals [55].

Theorem 3.1 The adaptive law (3.8) guarantees that

(i) $\varepsilon, \Theta \in \mathcal{L}_\infty$ *and* $\varepsilon, \varepsilon n_s, \dot{\Theta} \in \mathcal{L}_2$

(ii) *If* $n_s, \Phi, \dot{\Phi} \in \mathcal{L}_\infty$ *and* Φ is persistently exciting (PE), i.e. there exists positive numbers $\bar{\alpha}_1, \bar{\alpha}_2, \bar{T}$ such that $\bar{\alpha}_1 I \leq \int_t^{t+\bar{T}} \Phi(\tau)\Phi^T(\tau)d\tau \leq \bar{\alpha}_2 I$ for all $t \geq 0$, then $\Theta(t) \to \Theta^*$ exponentially fast.

Proof. The proof is a direct corollary of Theorem 3.8.1 and Theorem 3.10.1 of [55]. ■

Remark Under a suitable adaptive control law, the output of the system can track the desired signal, despite the fact that the parameter error vector may not converge to zero [55].

Chapter 4

Low Level Decentralized Adaptive Sliding Mode Control Design

The main objective of this chapter is designing an effective adaptive sliding mode low-level motion control scheme to be utilized in the low-level module of the proposed control architecture in Figure 3.1. A low-level decentralized adaptive sliding mode controller is developed and also applied to a commercially available robot manipulator.

4.1 The Approach

The low-level part of the proposed control architecture consists of an adaptive sliding mode control structure to handle the dynamic parameter uncertainties, unmodeled dynamics, and disturbance. This control structure is composed of two parts. The first part of this structure consists of an adaptive controller with an RLS based parameter estimator that generates parameter estimates on-line with the adaptive law. This part is introduced to

handle the MRR system whose nominal parameters are entirely unknown or could change when the robot morphology is reconfigured. A robust adaptive controller is also introduced, as the second part of this control structure, to compensate for unmodeled dynamics and external disturbance.

Furthermore, to achieve desired accurate positioning after the robot is reconfigured, the re-tuning of the controller gains are realized automatically. The idea of the method to have the fine-tuning controller gains after each robot reconfiguration begins with the characterization of available MRR modules to have their kinematic and dynamic information introduced in Chapter 3. The control architecture in this chapter collects the kinematic and dynamic information of each robot module from the high-level structure and processes them for the generation of the dynamical terms of the robot configuration and finding knowledge of bounds on the norms of the corresponding dynamic terms. This lets the MRRs operate with global asymptotic stability despite the uncertainties, unmodeled dynamics, and disturbance. The details of the design and analysis of the adaptive sliding mode control scheme are presented in the following sections.

4.2 Adaptive Sliding-Mode Control Design

The objective is to design a sufficiently smooth joint-space tracking control scheme for the MRR system given by Eq. (2.5). To meet this control objective, the n-dimensional joint position tracking error

$$e = q - q_d = [e_1, e_2, ..., e_n]^{\mathsf{T}} \tag{4.1}$$

is defined, where $q_d \in \mathbb{R}^n$ denotes the desired joint position vector. The main problem of interest then is defined as follows:

Problem 4.1 Given the uncertain MRR system in Eq. (2.5) and a desired trajectory $q_d \in \mathbb{R}^n$ such that $q_d, \dot{q}_d$ and $\ddot{q}_d$ are bounded, design a control law $u = \tau(t)$ for any initial condition q_0 and $\dot{q}_0$, such that all the system variables of (2.5) and u are bounded, and

$$\lim_{t\to\infty} e(t) = \lim_{t\to\infty} \dot{e}(t) = 0. \tag{4.2}$$

For the proposed sliding mode adaptive control design, the reference velocity signal is defined as follows:

$$\dot{q}_r = \dot{q}_d - \Lambda e, \tag{4.3}$$

where $\Lambda \in \mathbb{R}^{n\times n}$ is a positive definite gain matrix. By Eq. (4.3), the reference acceleration is

$$\ddot{q}_r = \ddot{q}_d - \Lambda \dot{e}. \tag{4.4}$$

The sliding mode variable vector is defined as

$$s = \dot{q} - \dot{q}_r = \dot{e} + \Lambda e \tag{4.5}$$

to be used in regulation of the tracking error $e(t)$. By Eq. (4.5), $\dot{q}$ and $\ddot{q}$ can be written as

$$\dot{q} = s + \dot{q}_r, \quad \ddot{q} = \dot{s} + \ddot{q}_r. \tag{4.6}$$

Substituting Eq. (4.6) into Eq. (2.5), it yields

$$M(q)\dot{s} + N(q,\dot{q})s = -M(q)\ddot{q}_r - N(q,\dot{q})\dot{q}_r - G(q) + \kappa + \tau. \tag{4.7}$$

The reference torque vector τ_r corresponding to the reference joint vector q_r is given by

$$\tau_r \triangleq M(q)\ddot{q}_r + N(q,\dot{q})\dot{q}_r + G(q) = \bar{\Phi}_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)\Theta_d^*, \tag{4.8}$$

which applying (4.8), can be rewritten in reduced order model form

$$\tau_r = M(q)\ddot{q}_r + N(q,\dot{q})\dot{q}_r + G(q) = \bar{\Phi}(q,\dot{q},\dot{q}_r,\ddot{q}_r)\Theta^*, \tag{4.9}$$

where $\bar{\Phi}_d$ and $\bar{\Phi}$ are, respectively equivalent to Φ_d and Φ, with the terms due to $M(\cdot)$, $N(\cdot,\cdot)$, $G(\cdot)$ evaluated at the current joint vector q and those due to the multipliers $\dot{q}_r$ and $\ddot{q}_r$ are evaluated at the reference joint vector so that these two equations exactly hold. Next, the general dynamics in Eq. (4.7) is rewritten by substituting Eq. (4.8) as follows:

$$M\dot{s} + Ns = -\underbrace{\bar{\Phi}(q,\dot{q},\dot{q}_r,\ddot{q}_r)\Theta^*}_{\tau_r} + \tau + \kappa. \tag{4.10}$$

The following properties of the dynamic model are used in the controller design [15]:

Property 4.1 The inertia matrix $M[q]$ is positive definite, diagonal, bounded, and satisfies $\mu_m\|s\|^2 \leq s^\mathsf{T} M s \leq \mu_M\|s\|^2$ for some known $\mu_m > 0$, $\mu_M > 0$, and all $s \in \mathbb{R}^n$.

Property 4.2 The centrifugal and Coriolis matrix $N[q,\dot{q}]$ satisfies $\|N\| \leq \mu_N\|\dot{q}\|$ for some bounded positive constant $\mu_N > 0$, where $q, \dot{q} \in \mathbb{R}^n$.

Property 4.3 The vector $G(q)$ satisfies $\|G\| \leq \mu_G$ for some constant $\mu_G > 0$.

Property 4.4 The centrifugal and Coriolis matrix N satisfies that $(\dot{M} - 2N)$ is skew-symmetric, noting that such selection of N is guaranteed to exist.

The following assumptions are also asserted to have the problem tractable.

Assumption 4.1 The sliding gain matrix $\Lambda \in \mathbb{R}^{n\times n}$ is positive definite, constant, and diagonal: $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_n) > 0$ where $\lambda_i > 0, i = 1, \cdots, n$.

Assumption 4.2 The reference joint position vector $q_r(t) \in \mathbb{R}^n$ is a bounded twice differentiable signal vector, i.e., $\dot{q}_r(t)$ and $\ddot{q}_r(t)$ are also bounded.

The following robust adaptive control law is proposed to handle the system uncertainties, unmodeled dynamics, and disturbances.

$$\begin{aligned}\tau(t) = &- k_1 s(t) - \underbrace{k_2(t)\text{sgn}(s(t)) - k_3 s(t)}_{\tau_{rs}} + \bar{\Phi}(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\Theta(t) \\ &- k_n \|\bar{\Phi}(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\|^2 s(t)\end{aligned} \tag{4.11}$$

where k_1 is a positive constant; k_2 is adjusted by the following adaptive law:

$$k_2(t) = (\alpha_0 + \alpha_1\|s\| + \alpha_2\|s\|^2 + \alpha_3\|\dot{s}\|) \tag{4.12}$$

where $\text{sgn}(\cdot)$ denotes the signum function. $\Theta(t) \in \mathbb{R}^b$ is the estimate of the unknown parameter vector $\Theta^* \in \mathbb{R}^b$ that is generated by (3.8), (3.9), $k_n \in \mathbb{R}$ is a positive constant, $k_n\|\bar{\Phi}\|^2 s$ term is a nonlinear damping term introduced in [56], [57], [58].

Lemma 4.1 *There are positive constants α_0, α_1, α_2, and α_3 such that along the trajectory of the robot the following is satisfied:*

$$\|\kappa\| \leq \alpha_0 + \alpha_1\|s\| + \alpha_2\|s\|^2 + \alpha_3\|\dot{s}\| \tag{4.13}$$

where $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ are positive constants that only depend on the matrix Λ, desired trajectory, and initial condition of the robot and parameters in (2.5).

Proof of Lemma 4.1 Let $\mu_{mij}, \mu_{G_i}, \mu_{F_i}$ be the positive maximum bounds for the magnitudes of $M_{ij}(q), N_{ij}(q)$, $G_i(q)$, $F_i + \tau_{ie}$ respectively. Consider the unmodeled dynamics and disturbance terms introduced in (2.6), the upper bound of κ_i is

$$\begin{aligned}\|\kappa_i\| \leq (&\sum_{j=1, j\neq i}^{n} \mu_{mij}[\|\dot{s}_j\| + \|\dot{q}_{jr}\|] \\ &+ \sum_{j=1, j\neq i}^{n} \mu_{nij}[\|s_j\| + \|q_{jr}\|][\dot{q}_j] + \mu_{Gi} + \mu_{Fi}\end{aligned}$$

Hence, applying Assumption 4.2, we obtain the inequality (4.13). The proof is thus completed. ■

Theorem 4.1 *Consider the system (2.5). The control law (4.11), (4.12) with the adaptive law (3.8), (3.9) guarantees that (i) the closed-loop trajectories in (2.5) are uniformly ultimately bounded (UUB) with an ultimate bound w on $\|s(t)\|$ given by*

$$w = \frac{1}{2\sqrt{\mu_m \varphi k_n}} \tag{4.14}$$

where $\varphi = \dfrac{k_1}{\mu_M}$; (ii) the tracking error $e(t)$ converges to a small neighborhood around the origin as $t \to \infty$; (iii) if the PE condition in Theorem 3.1 is satisfied then $e(t)$ converges to zero as $t \to \infty$.

Proof of Theorem 4.1

Define the Lyapunov function candidate

$$V_1 = \frac{1}{2} s^{\mathsf{T}} M s, \tag{4.15}$$

whose time derivative is

$$\dot{V}_1 = s^{\mathsf{T}} M \dot{s} + \frac{1}{2} s^{\mathsf{T}} \dot{M} s. \tag{4.16}$$

Substituting Eq. (4.10) into Eq. (4.16) and use of Property 4.4 yields

$$\begin{aligned}\dot{V}_1 &= s^{\mathsf{T}}[-k_1 s + \bar{\Phi}\Theta - \bar{\Phi}\Theta^* - k_n\|\Phi\|^2 s] + s^{\mathsf{T}}[-\tau_{rs} + \kappa] \\ &= s^{\mathsf{T}}[-k_1 s + \bar{\Phi}\tilde{\Theta} - k_n\|\bar{\Phi}\|^2 s] + s^{\mathsf{T}}[-\tau_{rs} + \kappa]. \\ &= s^{\mathsf{T}}[-k_1 s + \bar{\Phi}\tilde{\Theta} - k_n\|\bar{\Phi}\|^2 s] + s^{\mathsf{T}}[-k_2\mathrm{sgn}(s) + \kappa]\end{aligned} \tag{4.17}$$

which, together with Lemma 4.1, implies

$$\dot{V}_1 \leq -k_1\|s\|^2 - k_n\Big[\|\bar{\Phi}\|\|s\| - \frac{1}{2k_n}\|\tilde{\Theta}\|\Big]^2 + \frac{1}{4k_n}\|\tilde{\Theta}\|^2$$

$$\leq -2\varphi V_1 + \frac{1}{4k_n}\|\tilde{\Theta}\|^2 \quad \text{where} \quad \varphi = \frac{k_1}{\lambda_M}. \tag{4.18}$$

Eq. (4.18) further implies

$$V_1(t) \leq V_1(0)\exp^{(-2\varphi t)} + \frac{1}{8\varphi k_n}\sup_{0\leq\tau\leq t}\|\tilde{\Theta}(\tau)\|^2. \tag{4.19}$$

With Property 4.1,

$$\|s(t)\|^2 \leq \frac{\mu_M}{\mu_m}\|s(0)\|^2 e^{(-2\varphi t)} + \frac{1}{4\mu_m\varphi k_n}\sup_{t_0\leq\tau\leq t}\|\tilde{\Theta}(\tau)\|^2. \tag{4.20}$$

By Young's Inequality, (4.20) further implies

$$\|s(t)\| \leq \sqrt{\frac{\mu_M}{\mu_m}}\|s(0)\|e^{-\varphi t} + \frac{1}{2\sqrt{\mu_m\varphi k_n}}\sup_{0\leq\tau\leq t}\|\tilde{\Theta}(\tau)\|. \tag{4.21}$$

Further (4.21) together with Theorem 3.1 together imply (ii) and (iii). The proof is thus completed. ■

4.3 Summary

This chapter presents a control architecture for MRRs. Opposite to existing procedures that push the MRRs to require model-free control designs, this work reveals that a decentralized adaptive control depending on the bound on the dynamical terms is achievable without user intervention when the robot is reconfigured.

Chapter 5

Low Level Model Reference Adaptive Control

The main objective of this chapter is to design a model reference adaptive control (MRAC) scheme for modular and reconfigurable robots, with stability and convergence characteristics robust to reconfigurations, system uncertainties, unmodeled dynamics, and disturbances. The developed MRAC scheme tested on the Franka robot in the Gazebo simulation environment to verify the proposed control performance in comparison with other controllers.

5.1 The Reference Model and Control Design

The dynamic model introduced in Eq. (2.4) can be rewritten in state-space as follows:

$$\mathsf{S_i}: \begin{cases} \dot{x}_i(t) = & \bar{A}_i x_{i-1}(t) + A_i x_i(t) + \bar{\bar{A}}_i x_{i+1}(t) + B_i u_i(t) \\ y_i(t) = & C x_i(t), \end{cases} \tag{5.1}$$

where $x_i = [x_{i,1}, x_{i,2}]^\mathsf{T} = [q_i, \dot{q}_i]^\mathsf{T}$ stands for the state, and $y_i = q_i$ and $u_i = \tau_i$ are output and control input of the ith joint. $A_i \in \mathbb{R}^{2\times 2}$ is the state matrix; $\bar{A}_i \in \mathbb{R}^{2\times 2}$ and $\bar{\bar{A}}_i \in \mathbb{R}^{2\times 2}$ are the interconnection matrices, $B_i \in \mathbb{R}^{2\times 1}$ is input matrix, the output matrix is $C = [1, 0]$ and the control input vector is $u_i \in \mathbb{R}$. We choose the input vector $u = \tau = [\tau_1, \tau_2, \ldots, \tau_n]^\mathsf{T} \in \mathbb{R}^n$ such that all signals in the system are bounded. The state $x_i \in \mathbb{R}^2$ follows a reference model state $x_{ir} \in \mathbb{R}^2$.

We assert the following assumptions to have the control problem tractable.

Assumption 5.1 $(A_i, B_i), (\bar{A}_i, B_i), (\bar{\bar{A}}_i, B_i)$ pairs are all controllable.

Assumption 5.2 There exist control gain vectors $K_i, \bar{K}_i, \bar{\bar{K}}_i \in \mathbb{R}^{1\times 2}$ such that $A_i - B_i K_i = A_{ir}$, $\bar{A}_i - B_i \bar{K}_i = \bar{A}_{ir}$, $\bar{\bar{A}}_i - B_i \bar{\bar{K}}_i = \bar{\bar{A}}_{ir}$, where $A_{ir}, \bar{A}_{ir}, \bar{\bar{A}}_{ir}$ are pre-defined Hurwitz model reference system matrices.

Assumption 5.3 The following $2n \times 2n$ matrix is Hurwitz:

$$\underline{\mathrm{A}}_r = \begin{bmatrix} A_{1r} & \bar{\bar{A}}_{1r} & 0 & \ldots & 0 \\ \bar{A}_{2r} & A_{2r} & \bar{\bar{A}}_{2r} & \ddots & \vdots \\ 0 & \bar{A}_{3r} & A_{3r} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \bar{A}_{nr} & A_{nr} \end{bmatrix}. \tag{5.2}$$

The input vector is chosen $u = \tau = [\tau_1, \tau_2, \ldots, \tau_n]^\mathsf{T} \in \mathbb{R}^n$ such that all signals in the system are bounded. The state $x_i \in \mathbb{R}^2$ follows a reference model state $x_{ir} \in \mathbb{R}^2$. The reference model is described by

$$\dot{x}_{ir} = A_{ir}x_{ir} + \bar{A}_{ir}x_{(i-1)r} + \bar{\bar{A}}_{ir}x_{(i+1)r} + B_{ir}r_i \tag{5.3}$$

where $r_i \in \mathbb{R}$ is bounded reference input and $x_{ir}, x_{(i-1)r}$, and $x_{(i+1)r}$ represent desired trajectories that $x_i, x_{(i-1)}$, and $x_{(i+1)}$ have to follow.

Assumption 5.4 There exists a non-zero scalar L_i such that $B_iL_i = B_{ir}$. $\text{sgn}(L_i)$ is known.

The following decentralized MRAC scheme is proposed:

$$u_i(t) = -\hat{K}_i x_i(t) - \hat{\bar{K}}_i x_{(i-1)}(t) - \hat{\bar{\bar{K}}}_i x_{(i+1)}(t) + \hat{L}_i r_i(t) \tag{5.4}$$

where $\hat{K}_i$, $\hat{\bar{K}}_i$, $\hat{\bar{\bar{K}}}_i$, and $\hat{L}_i$ are the estimate of K_i, $\bar{K}_i$, $\bar{\bar{K}}_i$, and L_i respectively, generated by the following adaptive laws

$$\begin{aligned}
\dot{\hat{K}}_i(t) =& B_{ir}^\mathsf{T} P e_i x_i^\mathsf{T} \text{sgn}(L_i), \\
\dot{\hat{\bar{K}}}_i(t) =& B_{ir}^\mathsf{T} P e_{i-1} x_{(i-1)}^\mathsf{T} \text{sgn}(L_i), \\
\dot{\hat{\bar{\bar{K}}}}_i(t) =& B_{ir}^\mathsf{T} P e_{i+1} x_{(i+1)}^\mathsf{T} \text{sgn}(L_i), \\
\dot{\hat{L}}_i(t) =& -B_{ir}^\mathsf{T} P e_i r_i \text{sgn}(L_i),
\end{aligned} \tag{5.5}$$

where $P = P^\mathsf{T} > 0$ is a pre-defined positive definite matrix, $e_i = x_i - x_{ir} \in \mathbb{R}^2$ is model reference tracking error of joint i.

5.2 Stability Analysis

Subtracting and adding the desired control input, $-B_{ir}(-K_i x_i(t) - \bar{K}_i x_{(i-1)}(t) - \bar{\bar{K}}_i x_{(i+1)}(t) + L_i r_i)$ in (5.1), and considering Assumption 5.2, it yields

$$\begin{aligned} \dot{x}_i = \bar{A}_{ir} x_{(i-1)} + A_{ir} x_i + \bar{\bar{A}}_{ir} x_{(i+1)} + B_i r_i \\ + B_i(K_i x_i + \bar{K}_i x_{(i-1)} + \bar{\bar{K}}_i x_{(i+1)} - L_i r_i + u_i). \end{aligned} \tag{5.6}$$

Substituting (5.4) into (5.6) and subtracting (5.3), the tracking error $e_i = x_i - x_{ir}$ and parameter errors $\tilde{K}_i = \hat{K}_i - K_i$, $\tilde{\bar{K}}_i = \hat{\bar{K}}_i - \bar{K}_i$, $\tilde{\bar{\bar{K}}}_i = \hat{\bar{\bar{K}}}_i - \bar{\bar{K}}_i$, $\tilde{L}_i = \hat{L}_i - L_i$ satisfy

$$\dot{e}_i = \bar{A}_{ir} e_{i-1} + A_{ir} e_i + \bar{\bar{A}}_{ir} e_{i+1} + B_i(-\tilde{\bar{K}}_i x_{(i-1)} - \tilde{K}_i x_i - \tilde{\bar{\bar{K}}}_i x_{(i+1)} + \tilde{L}_i r_i), \tag{5.7}$$

which can be written as

$$\dot{e} = \underline{\mathrm{A}}_r e - \underline{\mathrm{B}}\ \tilde{\underline{\mathrm{K}}} x + \underline{\mathrm{B}}\ \tilde{\underline{\mathrm{L}}} r, \tag{5.8}$$

where $\tilde{\underline{\mathrm{K}}} = \begin{bmatrix} \tilde{K}_1 & \tilde{\bar{\bar{K}}}_1 & 0 & \dots & 0 \\ \tilde{\bar{K}}_2 & \tilde{K}_2 & \tilde{\bar{\bar{K}}}_2 & \ddots & \vdots \\ 0 & \tilde{\bar{K}}_3 & \tilde{K}_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{\bar{K}}_n & \tilde{K}_n \end{bmatrix}_{n \times 2n}$, $\underline{\mathrm{B}} = \begin{bmatrix} B_1 & & 0 & \dots & 0 \\ 0 & B_2 & 0 & \ddots & \vdots \\ 0 & 0 & B_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & B_n \end{bmatrix}_{2n \times n}$,

$\tilde{\underline{\mathrm{L}}} = \begin{bmatrix} \tilde{L}_1 & & 0 & \dots & 0 \\ 0 & \tilde{L}_2 & 0 & \ddots & \vdots \\ 0 & 0 & \tilde{L}_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & \tilde{L}_n \end{bmatrix}_{2n \times 2n}$,

$r = [r_1^\mathsf{T}, r_2^\mathsf{T}, \cdots, r_n^\mathsf{T}]^\mathsf{T}$, $e = [e_1^\mathsf{T}, e_2^\mathsf{T}, \cdots, e_n^\mathsf{T}]^\mathsf{T}$, $x = [x_1^\mathsf{T}, x_2^\mathsf{T}, \cdots, x_n^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{2n}$. The error dynam-

ics becomes

$$\dot{e} = \underline{\mathrm{A}}_r e + \underline{\mathrm{B}}_r \underline{\mathrm{L}}^{-1}(-\tilde{\underline{\mathrm{K}}} x + \tilde{\underline{\mathrm{L}}} r) \quad \text{where} \quad \underline{\mathrm{B}}_r \underline{\mathrm{L}}^{-1} = \underline{\mathrm{B}}. \tag{5.9}$$

Consider the Lyapunov function candidate

$$V(\tilde{\underline{\mathrm{K}}}, \tilde{\underline{\mathrm{L}}}, e) = e^{\mathsf{T}} \underline{\mathrm{P}} e + \mathrm{tr}(\tilde{\underline{\mathrm{K}}}^{\mathsf{T}} \tilde{\underline{\mathrm{K}}} + \tilde{\underline{\mathrm{L}}}^{\mathsf{T}} \tilde{\underline{\mathrm{L}}}), \tag{5.10}$$

where $\underline{\mathrm{P}} = \underline{\mathrm{P}}^{\mathsf{T}} = \mathrm{diag}\{P, \cdots, P\} \in \mathbb{R}^{2n \times 2n} > 0$ satisfies the Lyapunov equation

$$\underline{\mathrm{P}}\ \underline{\mathrm{A}}_r + \underline{\mathrm{A}}_r^{\mathsf{T}}\ \underline{\mathrm{P}} = -\underline{\mathrm{Q}} \tag{5.11}$$

for some $\underline{\mathrm{Q}} = \underline{\mathrm{Q}}^{\mathsf{T}} > 0$. The matrix $\underline{\mathrm{B}}_r^{\mathsf{T}} \underline{\mathrm{P}}$ behaves as an adaptive gain matrix, where $\underline{\mathrm{P}}$ is calculated by solving (5.11). Then, the time derivative of V in (5.10) is given by

$$\dot{V} = -e^{\mathsf{T}} \underline{\mathrm{Q}} e - 2e^{\mathsf{T}} \underline{\mathrm{P}} \underline{\mathrm{B}}_r \underline{\mathrm{L}}^{-1} \tilde{\underline{\mathrm{K}}} x + 2e^{\mathsf{T}} \underline{\mathrm{P}}\ \underline{\mathrm{B}}_r \underline{\mathrm{L}}^{-1} \tilde{\underline{\mathrm{L}}} r + 2\mathrm{tr}(\tilde{\underline{\mathrm{K}}}^{\mathsf{T}} \dot{\tilde{\underline{\mathrm{K}}}} + \tilde{\underline{\mathrm{L}}}^{\mathsf{T}} \dot{\tilde{\underline{\mathrm{L}}}}). \tag{5.12}$$

Noting that $\dot{\tilde{\underline{\mathrm{K}}}} = \dot{\underline{\mathrm{K}}}$, $\dot{\tilde{\underline{\mathrm{L}}}} = \dot{\underline{\mathrm{L}}}$, and

$$\begin{aligned} e^{\mathsf{T}} \underline{\mathrm{P}} \underline{\mathrm{B}}_r \underline{\mathrm{L}}^{-1} \tilde{\underline{\mathrm{K}}} x =& \mathrm{tr}(x^{\mathsf{T}} \tilde{\underline{\mathrm{K}}}^{\mathsf{T}} \underline{\mathrm{B}}_r^{\mathsf{T}} \underline{\mathrm{P}} e) \mathrm{sgn}(\underline{\mathrm{L}}) = \mathrm{tr}(\tilde{\underline{\mathrm{K}}}^{\mathsf{T}} \underline{\mathrm{B}}_r^{\mathsf{T}} \underline{\mathrm{P}} e x^{\mathsf{T}}) \mathrm{sgn}(\underline{\mathrm{L}}), \\ e^{\mathsf{T}} \underline{\mathrm{P}} \underline{\mathrm{B}}_r \underline{\mathrm{L}}^{-1} \tilde{\underline{\mathrm{L}}} r =& \mathrm{tr}(\tilde{\underline{\mathrm{L}}}^{\mathsf{T}} \underline{\mathrm{B}}_r^{\mathsf{T}} \underline{\mathrm{P}} e r^{\mathsf{T}}) \mathrm{sgn}(\underline{\mathrm{L}}). \end{aligned} \tag{5.13}$$

substitution of the adaptive laws (5.5) into (5.12) yields

$$\dot{V} = -e^{\mathsf{T}} \underline{\mathrm{Q}} e\ \leq 0.$$

Thus, $\hat{\underline{\mathrm{K}}}(t)$, $\hat{\underline{\mathrm{L}}}(t)$ and $e(t)$ are bounded and $e(t) \to 0$ as $t \to \infty$. Hence, we have proven the following result:

Theorem 5.1. *Consider the nominal state-space MRR dynamic model (5.1) and the reference model (5.3). Under Assumptions 5.1-5.4, the MRAC scheme (5.4), (5.5) guarantees that all the closed loop signals and the parameter estimates are bounded, and the reference tracking error $e(t)$ converges to zero as $t \to \infty$.*

5.3 Summary

This chapter has presented a state space MRAC scheme for MRRs. As different from existing procedures that push the MRRs to require model-free control designs, a decentralized MRAC scheme has been designed, whose performance depends on the bound on the dynamical terms is achievable and which does not require user intervention when the robot is reconfigured arbitrarily.

Chapter 6

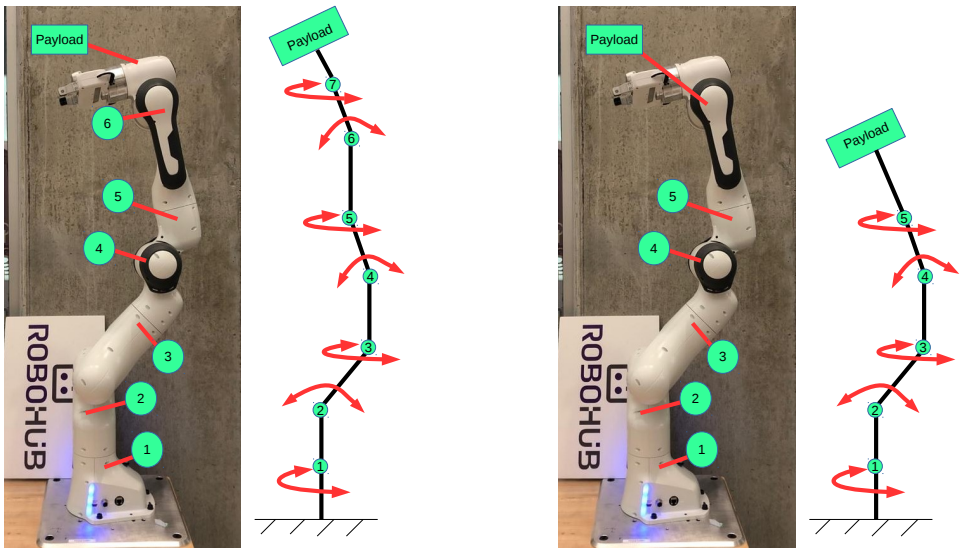
Simulations

6.1 Description of Franka Emika Panda Robot

The configurations are considered for the simulations are demonstrated in Figs. 6.1 (a), (b) and (c). The set of the available modules consists of a base module, five joint modules, and link modules, and one end-effector. Detailed information regarding the Panda robot can be found in [21]. The proposed control algorithms are developed in the ROS environment via C++ and the parametric model to estimate the dynamic parameters is generated by an algorithm built-in Python.

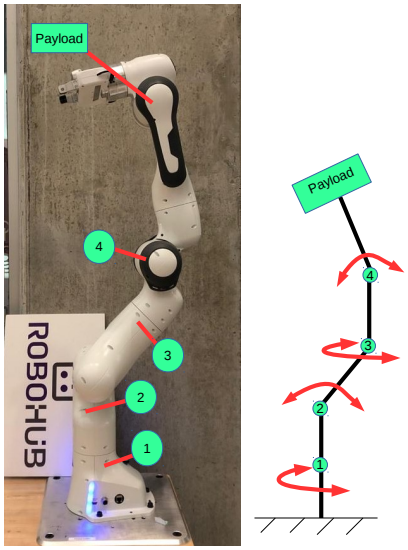
6.2 Simulation Results

In this section, a simulation analysis of the tracking performance of the Franka Panda arm under varying loading conditions is presented. In the simulations, we test and evaluate four controller structures, namely, PD, robust PD (RPD), MRAC, ASM, and their tracking



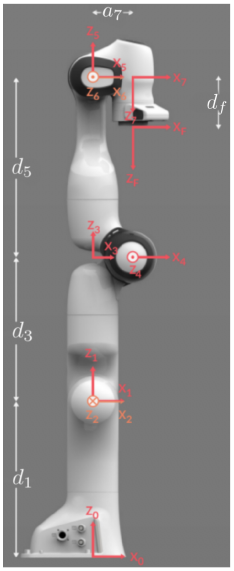
(a)

(b)



(c)

Figure 6.1: The Franka Emika Panda robot configurations. a) Configuration 1, b) Configuration 2, and c) Configuration 3 actuated with (i) no load, (ii) 1.5 Kg load, and (iii) 3 Kg load.



i	a_i	α_i	d_i	θ_i
1	0	0	d_1	q_1
2	0	$-\pi/2$	0	q_2
3	0	$\pi/2$	d_3	q_3
4	a_4	$\pi/2$	0	q_4
5	a_5	$-\pi/2$	d_5	q_5
6	0	$\pi/2$	0	q_6
7	a_7	$\pi/2$	0	q_7
8	0	0	d_f	0

Figure 6.2: DH parameters for the Franka Emika Panda robot. $d_1 = 0.333$m, $d_3 = 0.316$m, $d_5 = 0.384$m, $d_f = 0.107$m, $a_4 = 0.0825$m, $a_5 = -0.0825$m, $a_7 = 0.088$m [21].

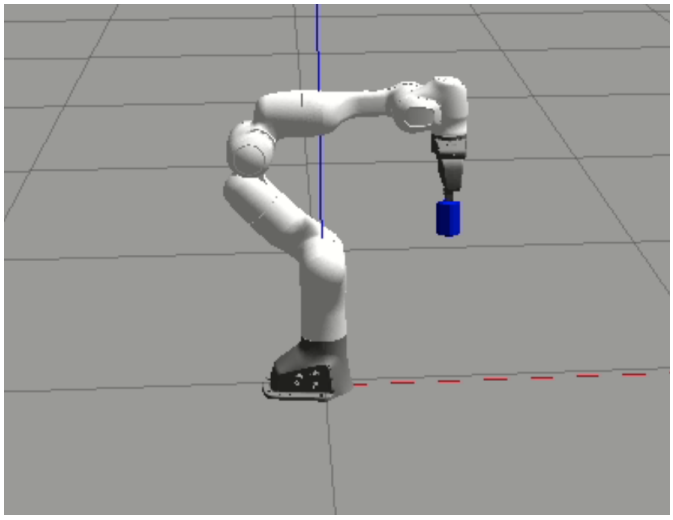


Figure 6.3: Pick a payload and place to the desired position in the joint space.

performances are compared for three robot configurations. The effect of a payload variation on the tracking error of three configurations is also analyzed. To compare the performance of the controllers, mean squared error (MSE) and percentage (%) improvement are defined as follows:

$$MSE = \frac{\sum_{i=1}^{N} e_i^2}{b} \tag{6.1}$$

where N is the number of sampled points in the simulation, e_i denotes i-th joint position error, and

$$\%\text{improvement via adaptation} = \frac{MSE_{controller} - MSE_{PD}}{MSE_{PD}} \times 100. \tag{6.2}$$

In this set of simulations, three robot configurations are actuated with (i) no load, (ii) 1.5 Kg load, and (iii) 3 Kg load (see Fig. 6.3) and the following operating cycles are tested:

Operation 1: The first operation is executed based on a pick and place task. The robot joint values are set to accomplish the desired task such that it picks and places a payload from one point to the other. For the first operation, the following joint positions are given to the robot arm:

- To reach the first point A, the desired position is set to be $q_A = [1, 0.5, 0, -2, 0, 2.5, 0]^\mathsf{T}$;
- To move the second position B, the desired joint positions are set to be $q_B = [0, 0.2, 0, -1, 0, 1.2, 0]^\mathsf{T}$ at $t = 6$s;
- To reach the third position C, the desired joint positions are given to be $q_C = [-1, 0.5, 0, -1.2, 0, 1.6, 0]^\mathsf{T}$ at $t = 12$s ;
- To complete the cycle and back to the first position, the desired joint positions are firstly set to be q_B at $t = 18$s, and the desired position is set again to be q_A at $t = 24s$.

Operation 2: For the second operation, the following desired joint positions are selected according to a pick and place task from position A to the other position C:

- The reach the first position A, the desired position is set to be $q_A = [1, 0.5, 0, -2, 0, 2.5, 0]^\mathsf{T}$ at t=0s;
- To move the second joint positions C, the joint values are commented to be $q_C = [-1, 0.5, 0, -1.2, 0, 1.6, 0]^\mathsf{T}$ at $t = 8$s;
- The desired position is adjusted to be q_A to complete the second operation.

Operation 3: For the third operation, the following desired joint positions are selected according to a pick and place task from position A to the other position B:

- To move the robot from the initial position to the first position A, the desired joint positions are chosen to be $q_A = [1, 0.5, 0, -2, 0, 2.5, 0]^\mathsf{T}$;
- To place the robot to the central position B, the desired joint values are given to be $q_B = [0, 0.2, 0, -1, 0, 1.2, 0]^\mathsf{T}$ at $t = 8$s;
- At $t = 16$s, the desired position is set again to q_A to complete the cycle.

In this section, the simulation results are presented to demonstrate the effect of trajectory/payload variations changes on the robot arm. First, the results of the controllers' performances with payload variations are summarized in Tabs. 6.1-6.5 in terms of MSE. Besides, for the first operation, the actual joint positions of the robot configurations with different payload conditions are demonstrated in Figs. 6.4-6.12. For three robot configurations with payload variations, to show the joint position performances of the proposed control structures during the second operation, Figs. 6.13-6.21 are introduced. Further, to demonstrate the actual positions of the joints during the third operation, Figs. 6.22-6.30 are introduced for three different robot configurations.

Configuration 1

Operation 1: For the first operating condition, using configuration 1 shown in Figure 6.1 (a) with a load of 1.5 Kg, the maximum tracking error of the first joint increased by 32.29%, 39.02%, 15.29% and 4.77% for PD, Robust PD, MRAC, ASMC, respectively. After the payload was attached, the maximum tracking error of the second joint rose by 49.4%, 26.11%, 6.9% for PD, RPD, and MRAC, respectively. Joint 3, 4, 5, 6 were also affected and showing an increase in the maximum tracking error when the payload was attached. Furthermore, the maximum position error of the first joint with a load of 3 Kg increased by 90%, 98.7%, 20.82%, and 11.48% for PD, RPD, MRAC, ASMC, respectively. The max-

imum tracking error for joint 2 increased by 49.4%, 45.4%, 23.29%, 8.19% when a load of 3 Kg is attached. Joint 3, 4,5,6 were also affected and tracking error for the controller increased by 27%,32%,33% in when the load was added.

Operation 2: The controller gains were kept fixed to see the tracking position error when the trajectory was changed under different loading conditions. The maximum tracking error under 3 Kg load for first joint increased by 33%. The maximum tracking error for first joint under the first operation is more than the second operation. When the load is added, the maximum tracking error increased by 31.86% for the second joint. When the load was attached for the second operation, joint 3 showed a 38.76% increase in the maximum tracking error. Under the varying payload condition, the tracking error of the second operation is less than the first operation for joint 3,4,5,6.

Table 6.1: The results for the first configuration of Joint 1.

	Operation 1	Joint 1	Operation 2	Joint 1
Controller/Payload	MSE	%	MSE	%
PD/no payload	0.96	-	0.886	-
PD/1.5 KG	1.27	-	1.18	-
PD/3 KG	1.824	-	1.71	-
Robust PD /no payload	0.82	14.58	0.84	5.19
Robust PD/ 1.5 KG	1.14	10.23	1.06	11.32
Robust PD/ 3 KG	1.63	10.63	1.48	13.4
MRAC/no payload	0.654	31.875	0.62	30.02
MRAC/ 1.5 KG	0.754	40.629	0.705	40.25
MRAC/ 3 KG	0.7902	56.677	0.75	56.14
ASMC/no payload	0.566	41.04	0.54	39.05
ASMC/1.5 KG	0.593	53.30	0.573	51.44
ASCM/3 KG	0.631	65.24	0.593	65.32

Configuration 2

Operation 1: For the first operation, using the configuration 2 shown in Figure 6.1 (b), for the first operating condition, when a load of 1.5 Kg was added, the tracking error of the first joint increased by 40.8%, 34.48%, 14.65% and 5.06% for PD, Robust PD, MRAC, ASMC, respectively. The maximum tracking error for the second joint increased by 33.97%, 43.18%, 4.96% for PD, RPD, and MRAC respectively. When the load was attached, Joint 3, 4,5,6 were also affected and the maximum tracking error increased. Furthermore, the maximum tracking error for the first joint increased by 91.40%, 92.7%, 7.927% and 4.912% for PD, RPD, MRAC, ASMC, respectively when a load of 3 Kg was attached. The max-

Table 6.2: The results for the first configuration of Joint 2 .

	Operation 1	Joint 2	Operation 2	Joint 2
Controller/Payload	MSE	%	MSE	%
PD/no payload	0.89	-	0.8635	-
PD/1.5 KG	1.23	-	1.12	-
PD/3 KG	1.33	-	1.19	-
Robust PD /no payload	0.674	6.15	0.781	8.25
Robust PD/ 1.5 KG	0.85	22.83	0.831	25.803
Robust PD/ 3 KG	0.98	26.31	0.889	25.294
MRAC/no payload	0.601	25.16	0.583	32.48
MRAC/ 1.5 KG	0.643	41.23	0.625	44.19
MRAC/ 3 KG	0.741	48.34	0.647	45.63
ASMC/no payload	0.5174	41.86	0.57	33.98
ASMC/1.5 KG	0.57	44.51	0.603	46.160
ASCM/3 KG	0.66	50.69	0.631	46.974

imum error of the second joint under a load of 3 Kg increased by 47.2%, 42.7%, 20.13% , 5.15%. Joint 3, 4,5,6 were also affected when the same load was retained and their maximum tracking errors increased.

Operation 3: For the second operation under a load of 3 Kg, the maximum error of the first joint is more than the tracking error during the third operation. The maximum position error of the second joint increased by 37.4% increased when the load was attached. Under the presence of the same load for the second joint, the maximum tracking error of the third task is less than the first task. The maximum error of the second operation is less than the first operation for joint 3,4,5,6.

Configuration 3

Operation 1: For the first operation, using the configuration 3 shown in Figure 6.1 (c) under a load of 1.5 Kg, the maximum joint position error for joint 1 increased by 45.17%, 18.34%, 5.622% and 5.01% for PD, Robust PD, MRAC, ASMC, respectively. The maximum tracking error of the second joint increased by 33.97%, 43.18%, 4.96% for PD, RPD, and MRAC respectively when the payload is added. Joint 3, 4,5,6 were also affected and their maximum joint errors increased when the load were added. Furthermore, the maximum error of joint 1 under a load of 3 Kg increased by 90.68%, 25.43%, 13.62% and 8.908% for PD, RPD, MRAC, ASMC, respectively. The maximum tracking error of the second joint increased by 80.0%, 30.4%, 9.96% , 3.95%. The maximum tracking error of Joint 3, 4,5,6 were also affected and increased when the payload was attached.

Operation 3: For the third operation, the controller gains were kept the same to see the effect of the trajectory change on the maximum error under different payloads. Under 3 Kg load, the maximum tracking error increased for the first joint. Hence, the maximum

Table 6.3: The results for the second configuration of Joint 1.

	Operation 1	Joint 1	Operation 3	Joint 1
Controller/Payload	MSE	%	MSE	%
PD/no payload	0.93	-	0.89	-
PD/1.5 KG	1.31	-	1.281	-
PD/3 KG	1.63	-	1.533	-
Robust PD /no payload	0.83	10.75	0.797	10.44
Robust PD/ 1.5 KG	1.11	15.26	1.07	16.47
Robust PD/ 3 KG	1.37	15.95	1.28	16.50
MRAC/no payload	0.621	33.22	0.617	30.67
MRAC/ 1.5 KG	0.712	45.64	0.709	44.65
MRAC/ 3 KG	0.732	55.09	0.7410	51.66
ASMC/no payload	0.572	38.49	0.546	38.65
ASMC/1.5 KG	0.601	54.12	0.579	54.8
ASCM/3 KG	0.6402	60.72	0.587	60.7

Table 6.4: The results for the second configuration of Joint 2.

	Operation 1	Joint 2	Operation 3	Joint 2
Controller/Payload	MSE	%	MSE	%
PD/no payload	0.942	-	0.89	-
PD/1.5 KG	1.462	-	1.15	-
PD/3 KG	1.803	-	1.313	-
Robust PD /no payload	0.859	8.81	0.865	
Robust PD/ 1.5 KG	1.16	15.86	1.04	
Robust PD/ 3 KG	1.57	12.9	1.51	
MRAC/no payload	0.6421	31.80	0.631	
MRAC/ 1.5 KG	0.674	53.89	0.654	
MRAC/ 3 KG	0.693	60.5	0.663	
ASMC/no payload	0.572	39.27	0.538	
ASMC/1.5 KG	0.5831	56.7	0.569	
ASCM/3 KG	0.6001	61.2	0.584	

tracking error for joint 1 under the third operation is more than the error for the second operation.

Table 6.5: The results for the third configuration of Joint 1.

	Operation 2	Joint 1	Operation 3	Joint 1
Controller/Payload	MSE	%	MSE	%
PD/no payload	0.923	-	0.967	-
PD/1.5 KG	1.34	-	1.39	-
PD/3 KG	1.76	-	1.81	-
Robust PD /no payload	0.7103	23.04	0.733	24.27
Robust PD/ 1.5 KG	1.08	19.40	1.1653	16.54
Robust PD/ 3 KG	1.31	25.56	1.31	27.62
MRAC/no payload	0.6012	34.86	0.6130	36.60
MRAC/ 1.5 KG	0.635	52.61	0.6491	53.3
MRAC/ 3 KG	0.6831	61.18	0.7213	60.1
ASMC/no payload	0.5343	42.11	0.541	44.05
ASMC/1.5 KG	0.5611	58.12	0.5671	59.20
ASCM/3 KG	0.5919	61.2	0.6193	59.2

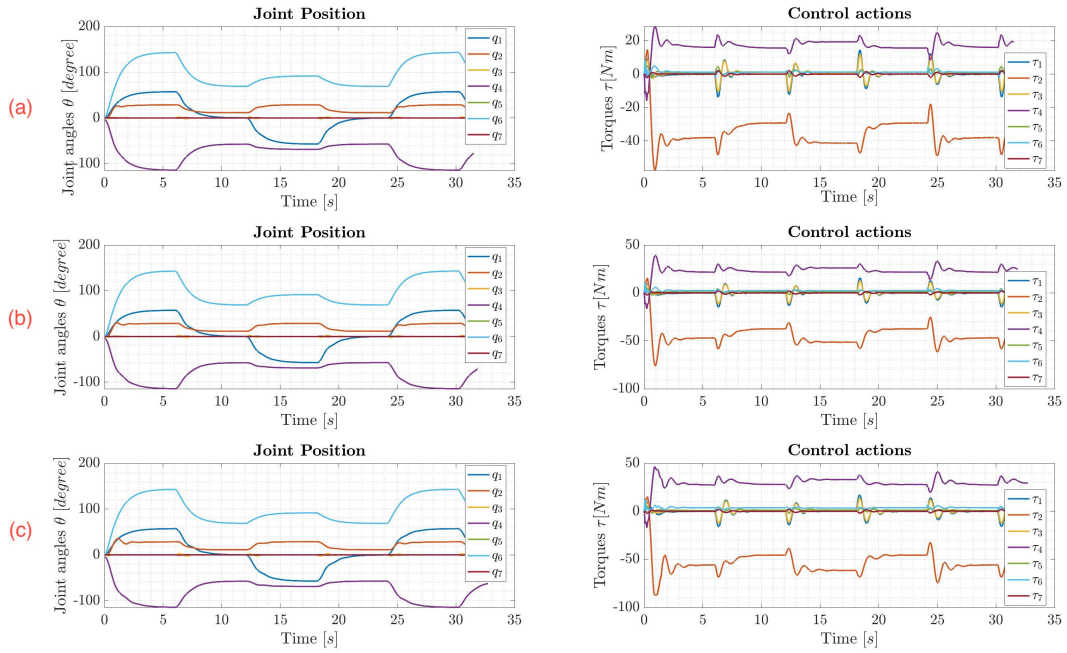


Figure 6.4: Adaptive sliding mode controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1.

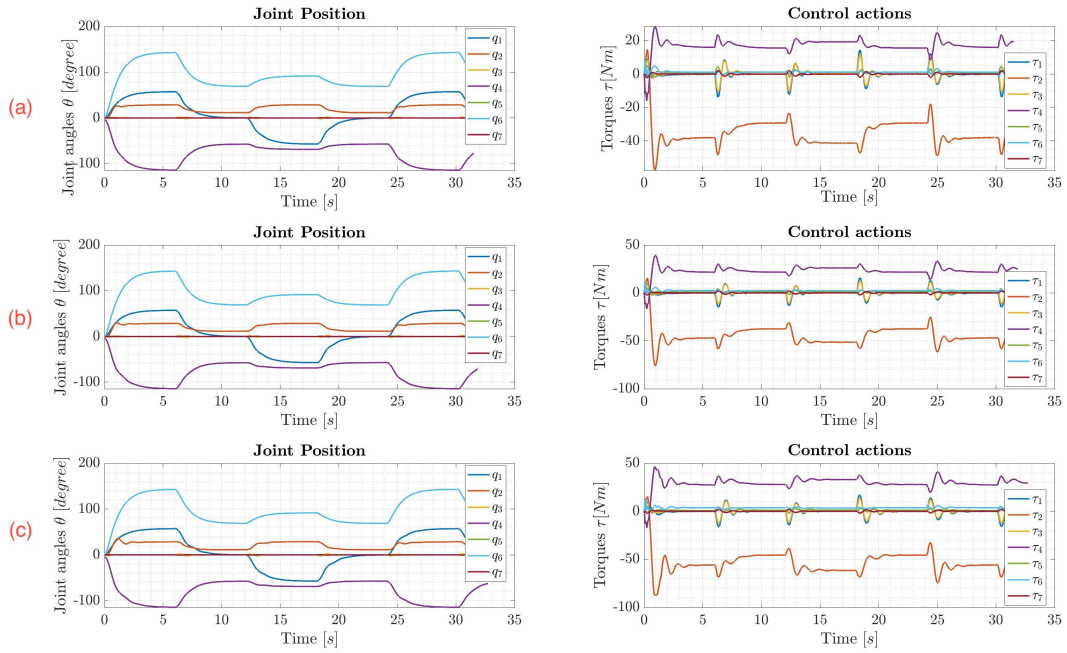


Figure 6.5: Adaptive model reference controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1.

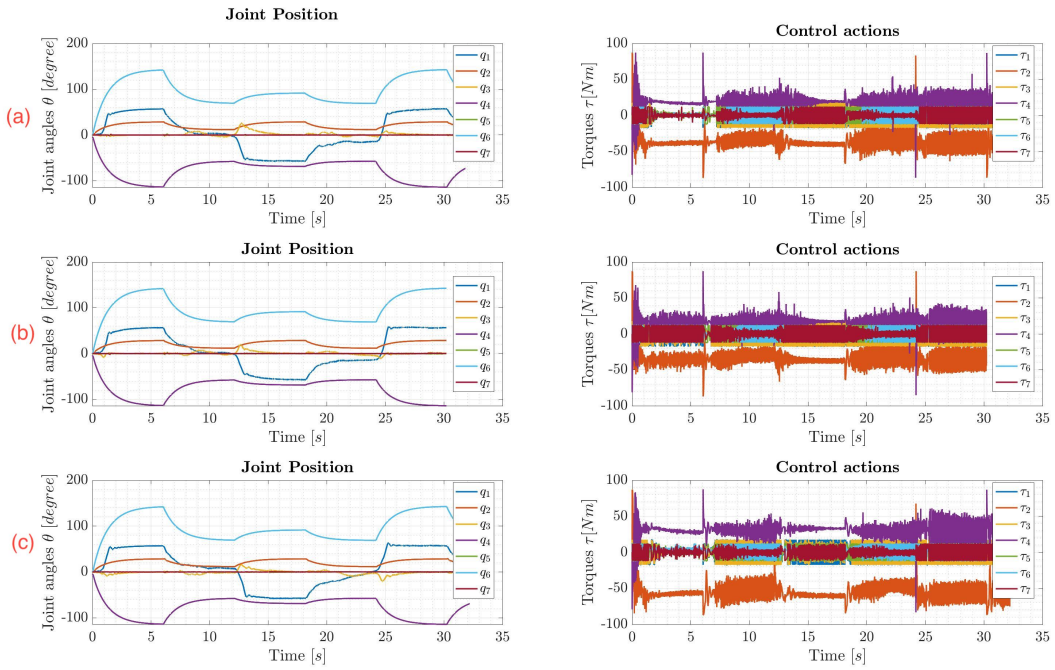


Figure 6.6: Robust PD controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1.

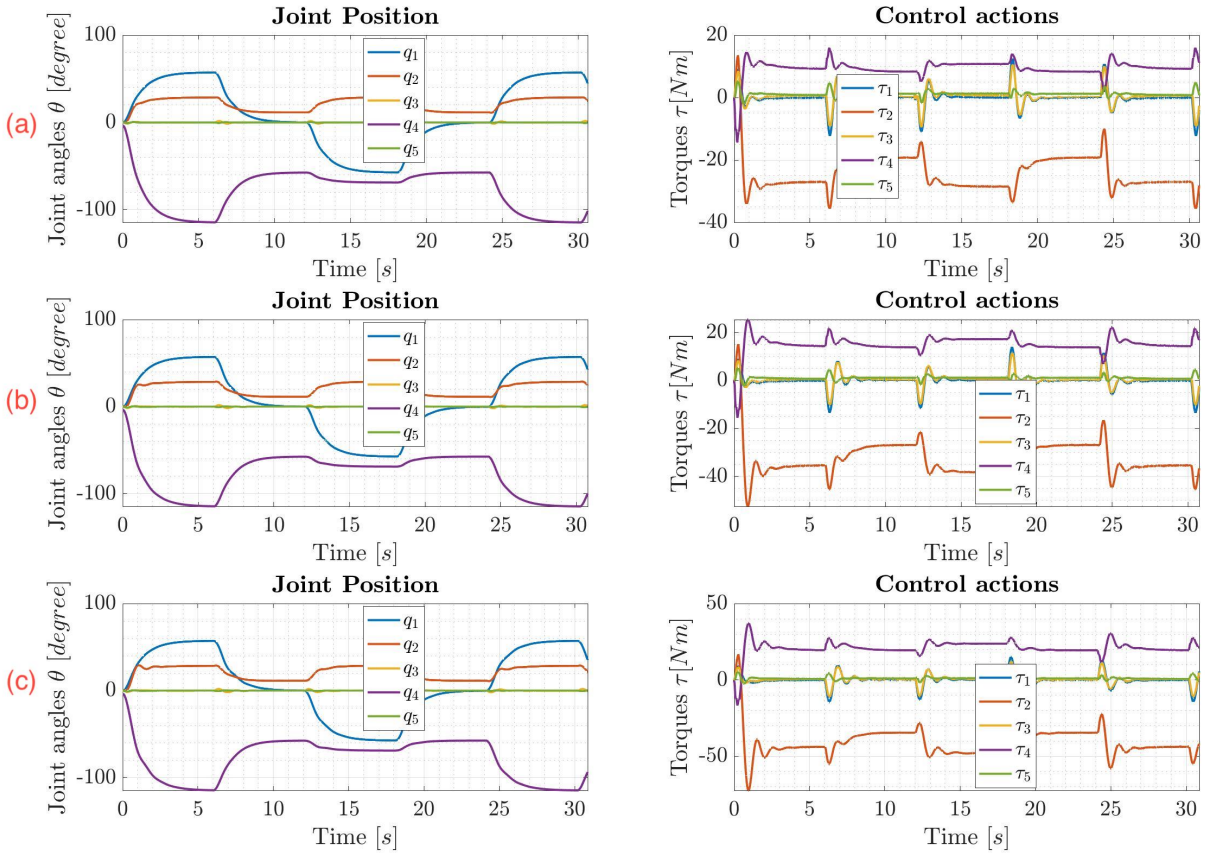


Figure 6.7: Adaptive sliding mode controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

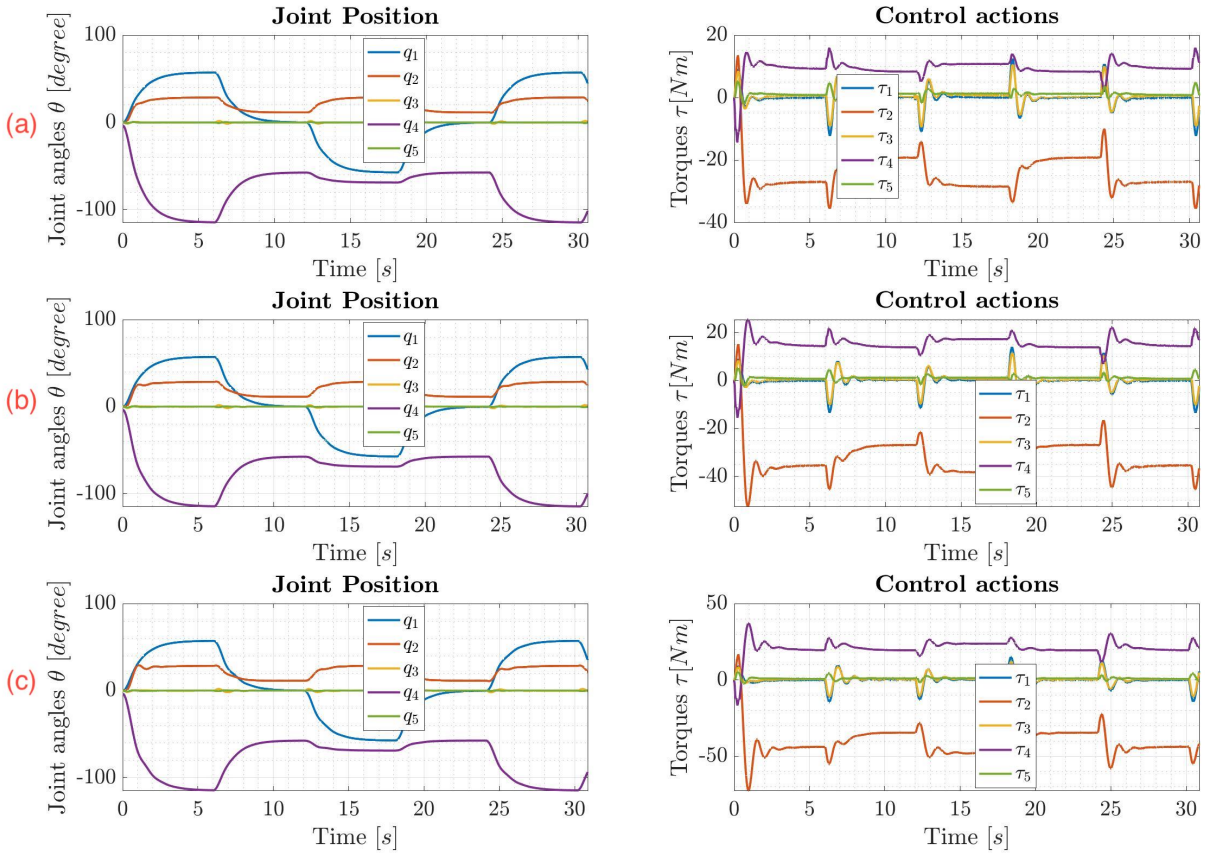


Figure 6.8: Adaptive model reference controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

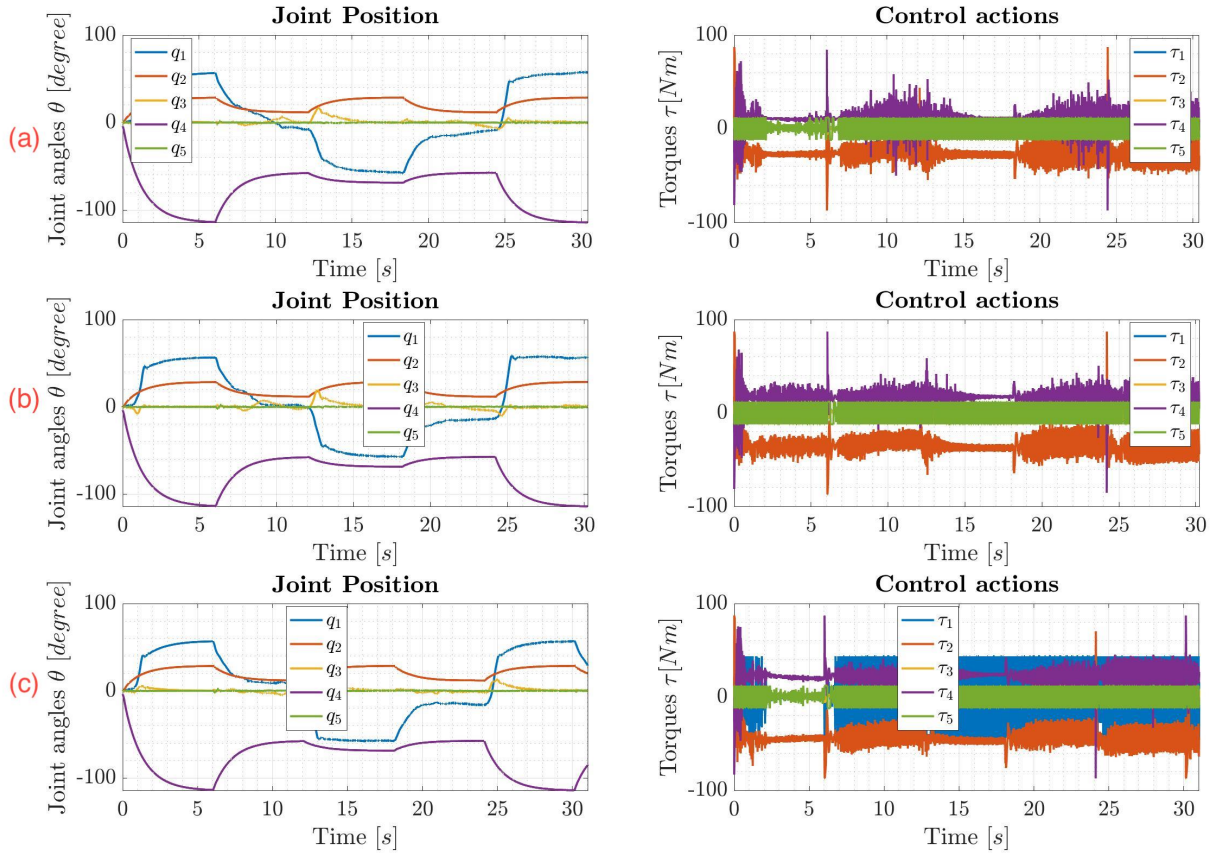


Figure 6.9: Robust PD controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

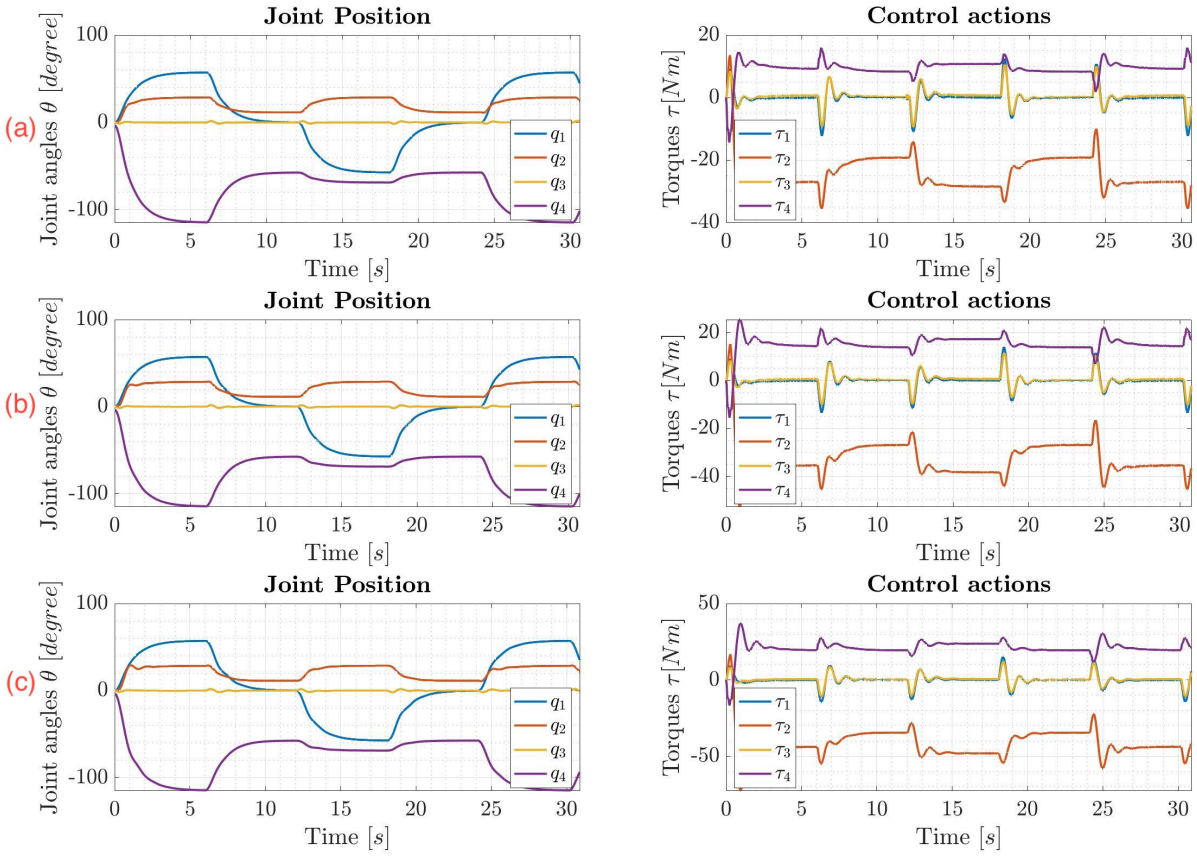


Figure 6.10: Adaptive sliding mode controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 3.

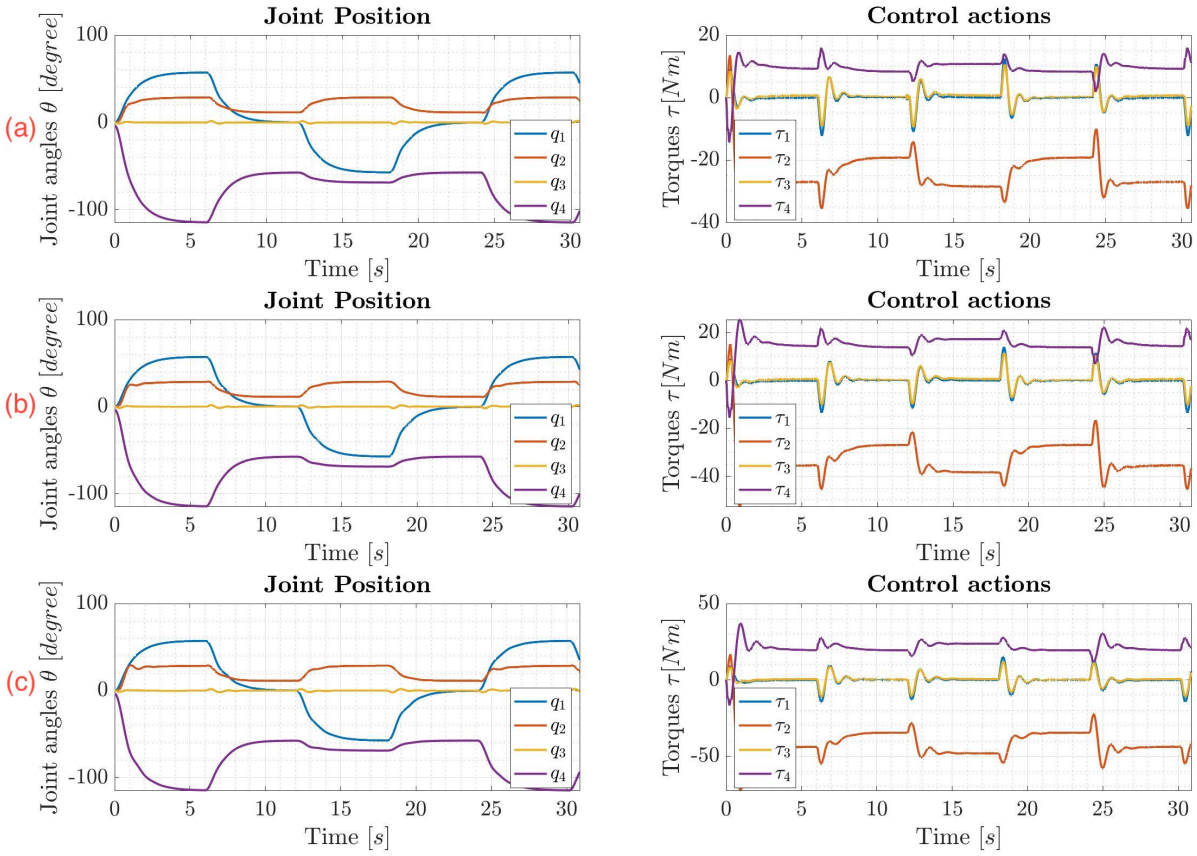


Figure 6.11: Adaptive model reference controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 3.

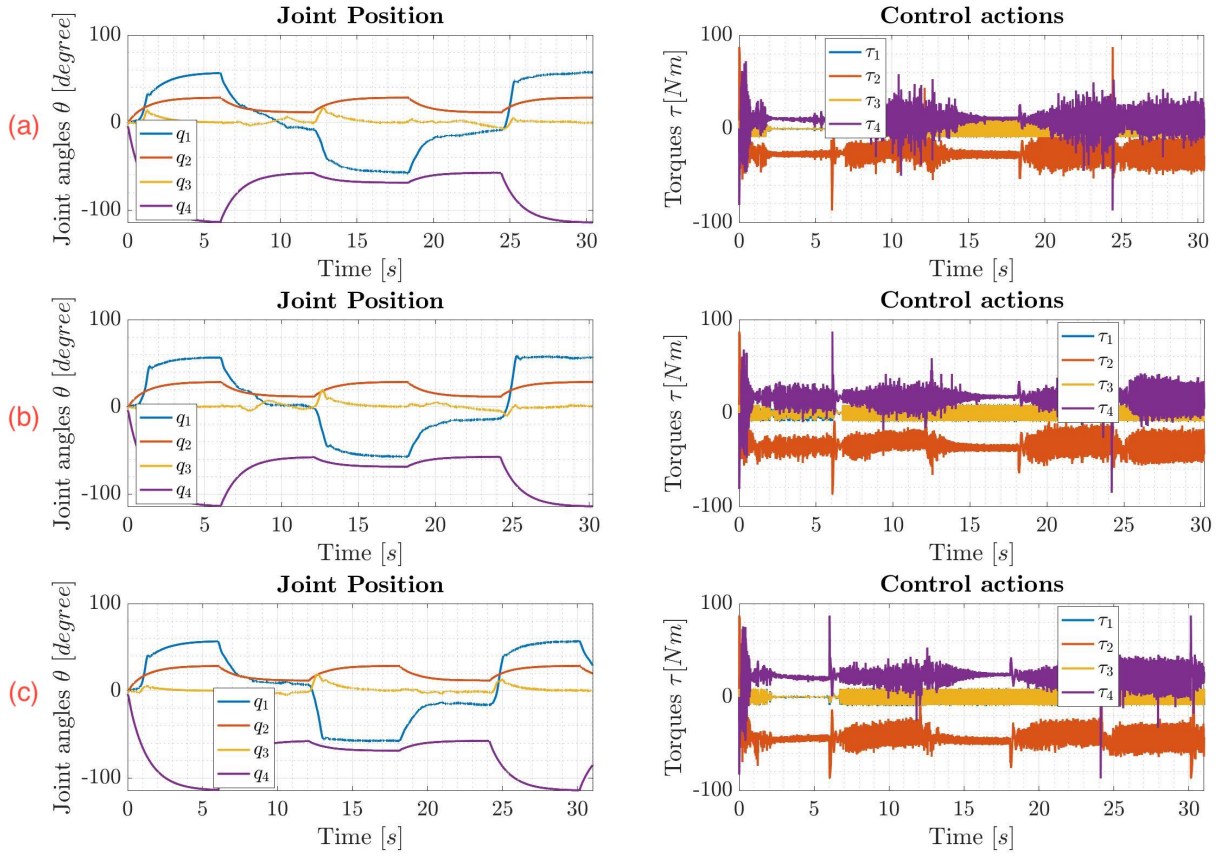


Figure 6.12: Robust PD controller's performance of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 3.

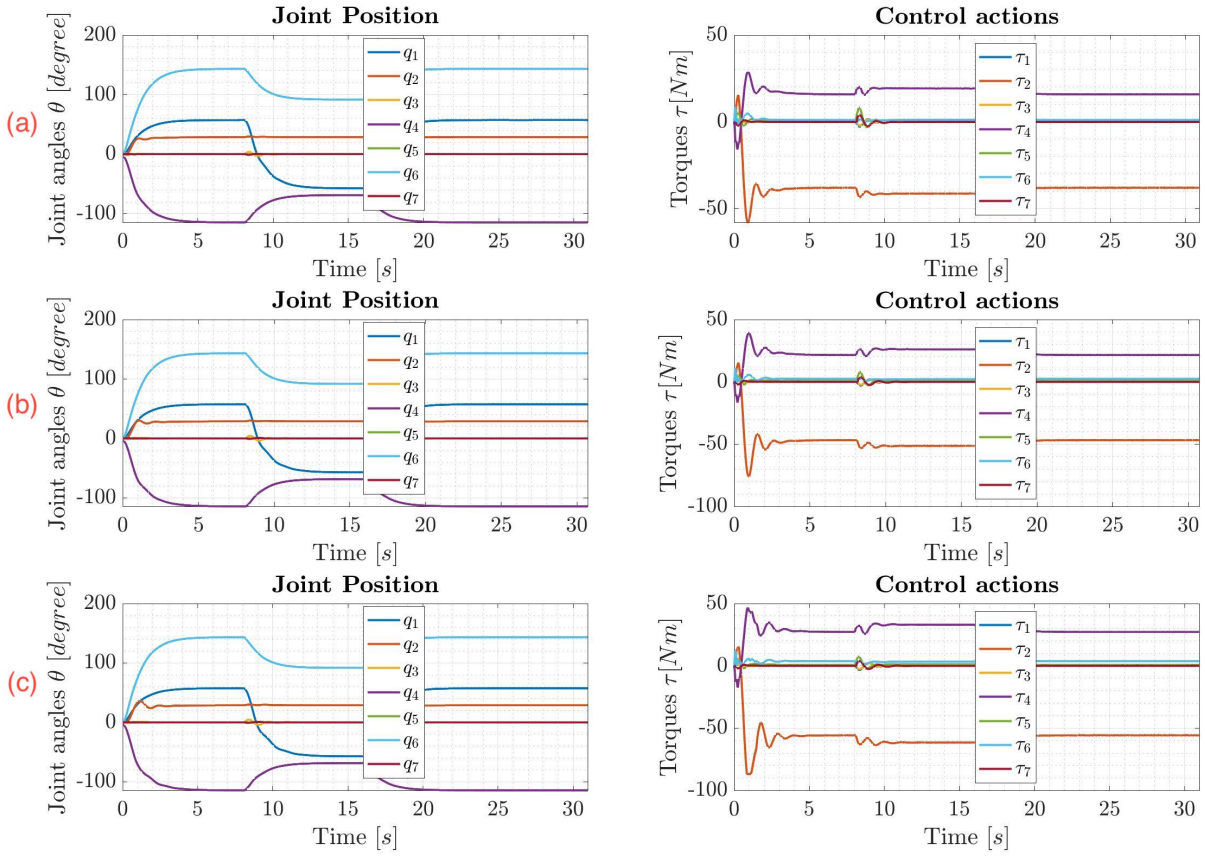


Figure 6.13: Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1.

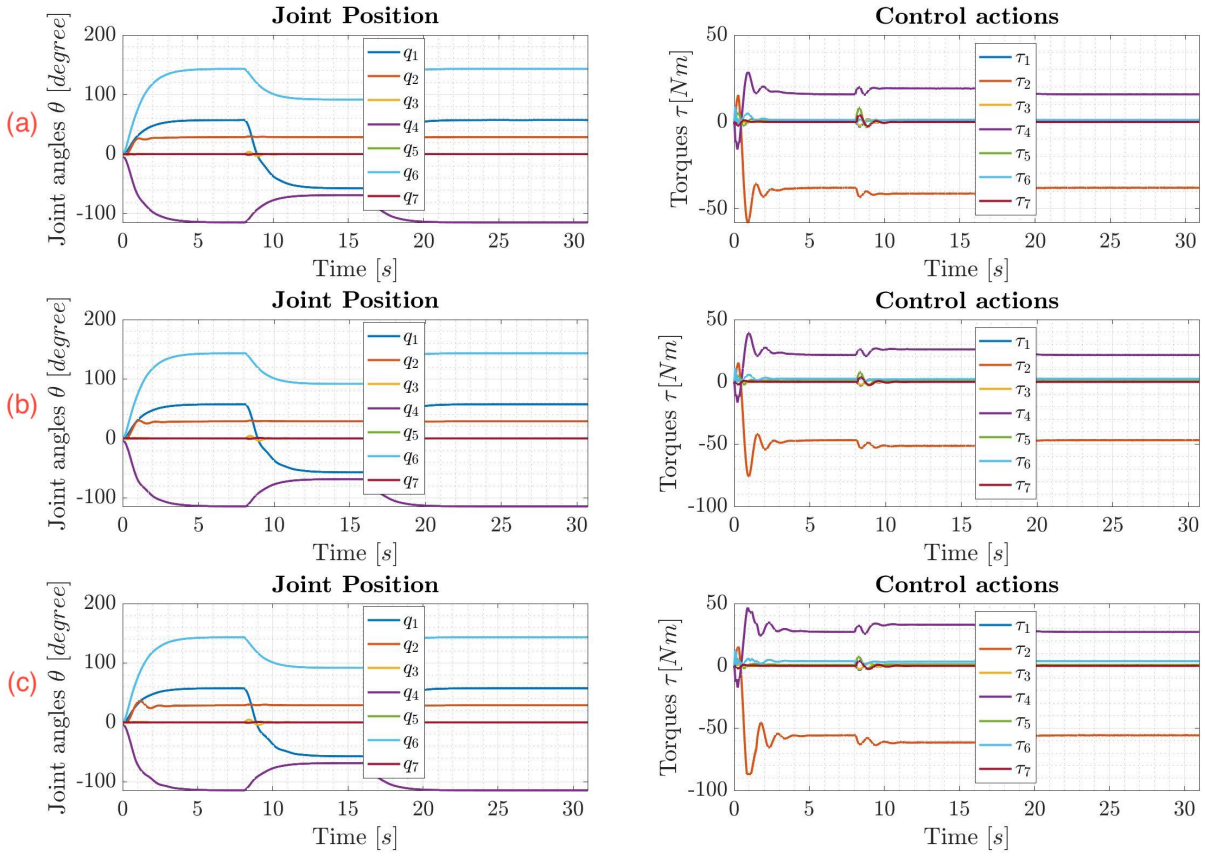


Figure 6.14: Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1.

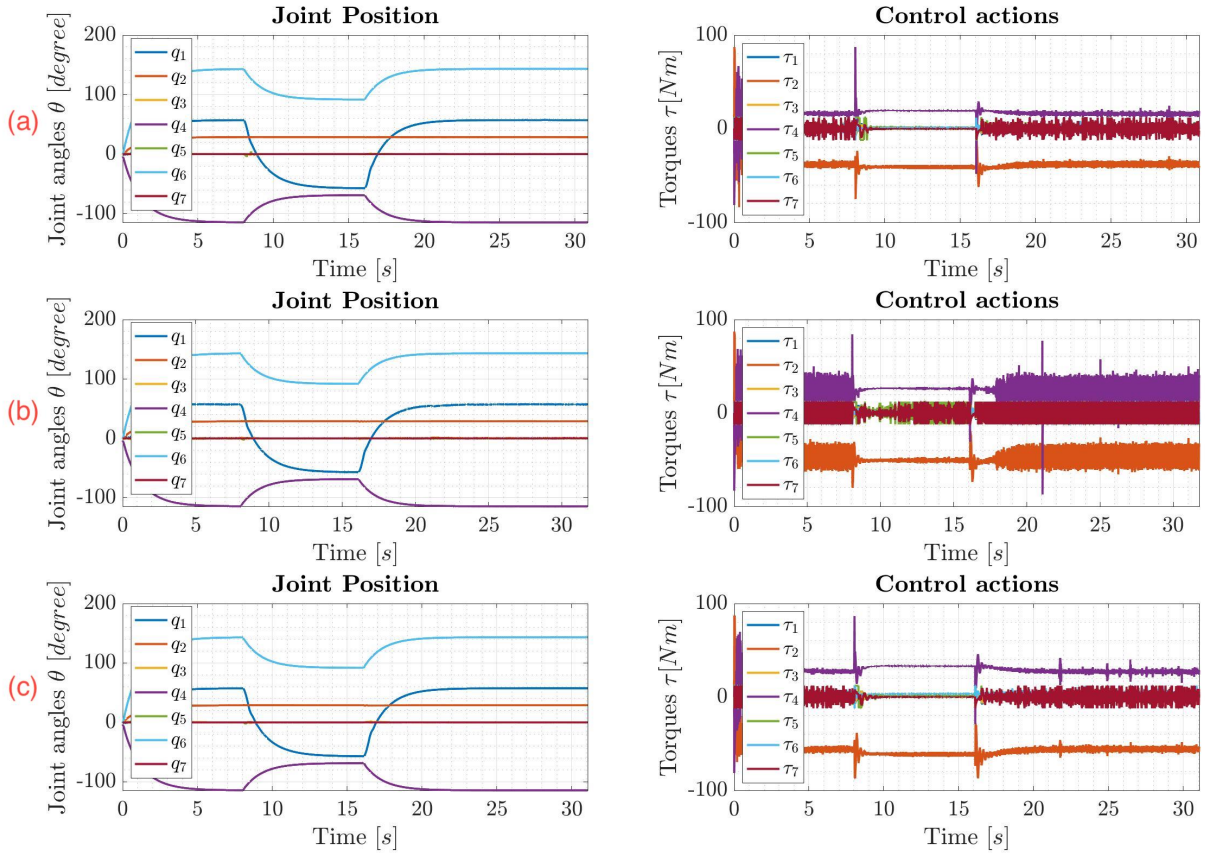


Figure 6.15: Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 1.

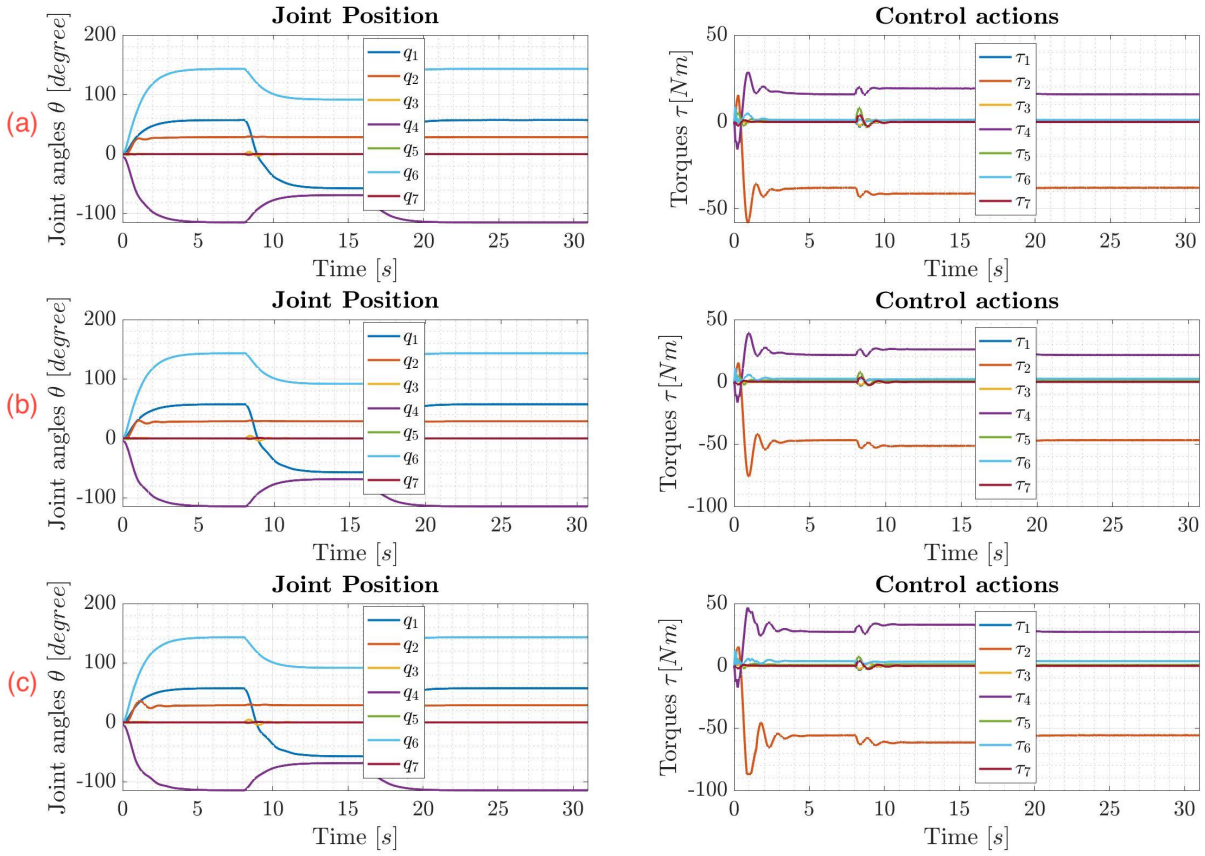


Figure 6.16: Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

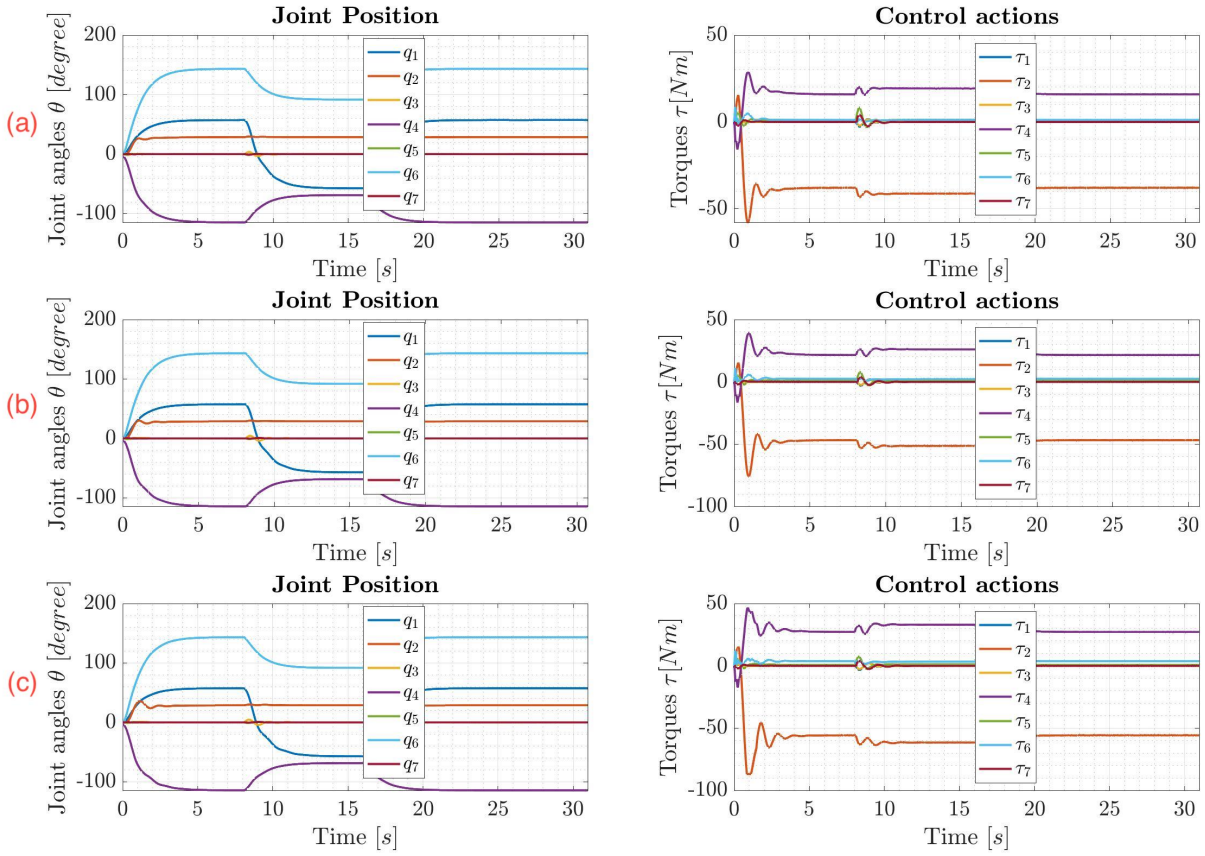


Figure 6.17: Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

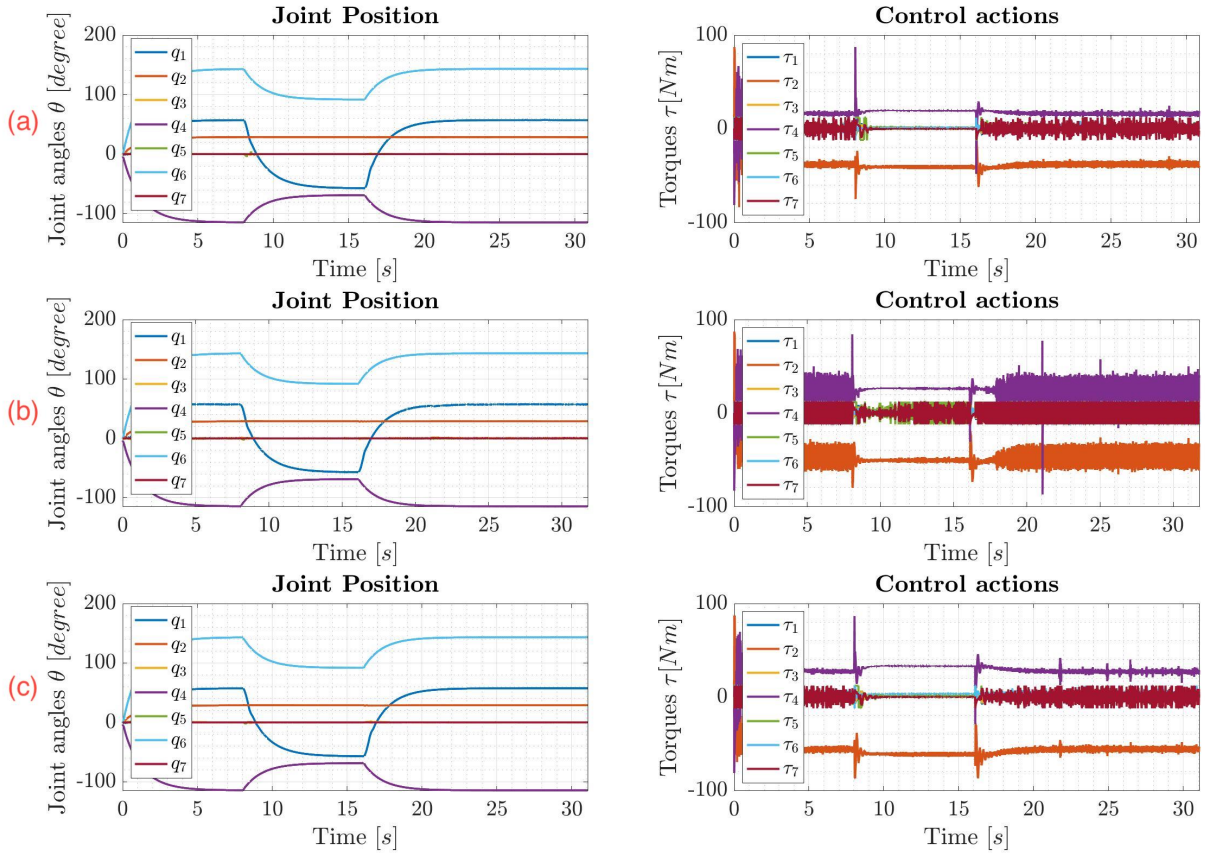


Figure 6.18: Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

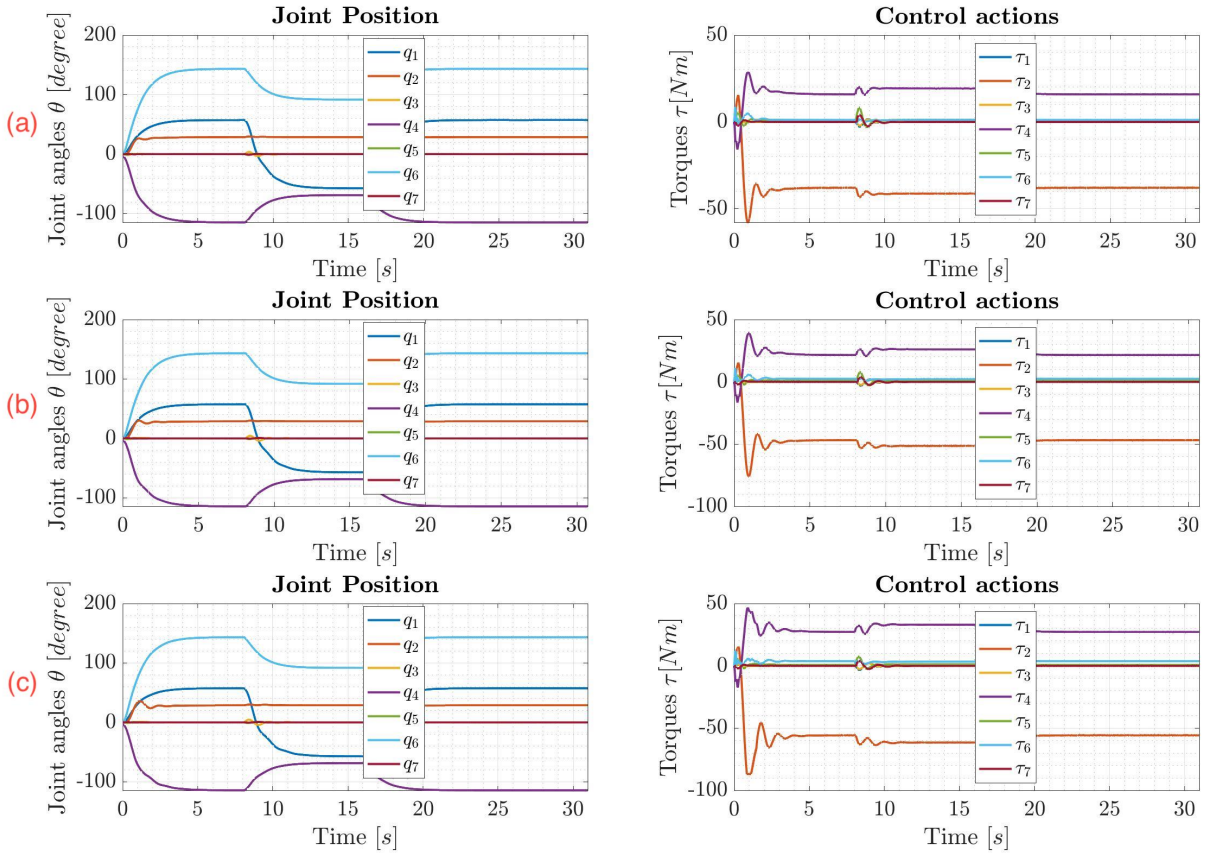


Figure 6.19: Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

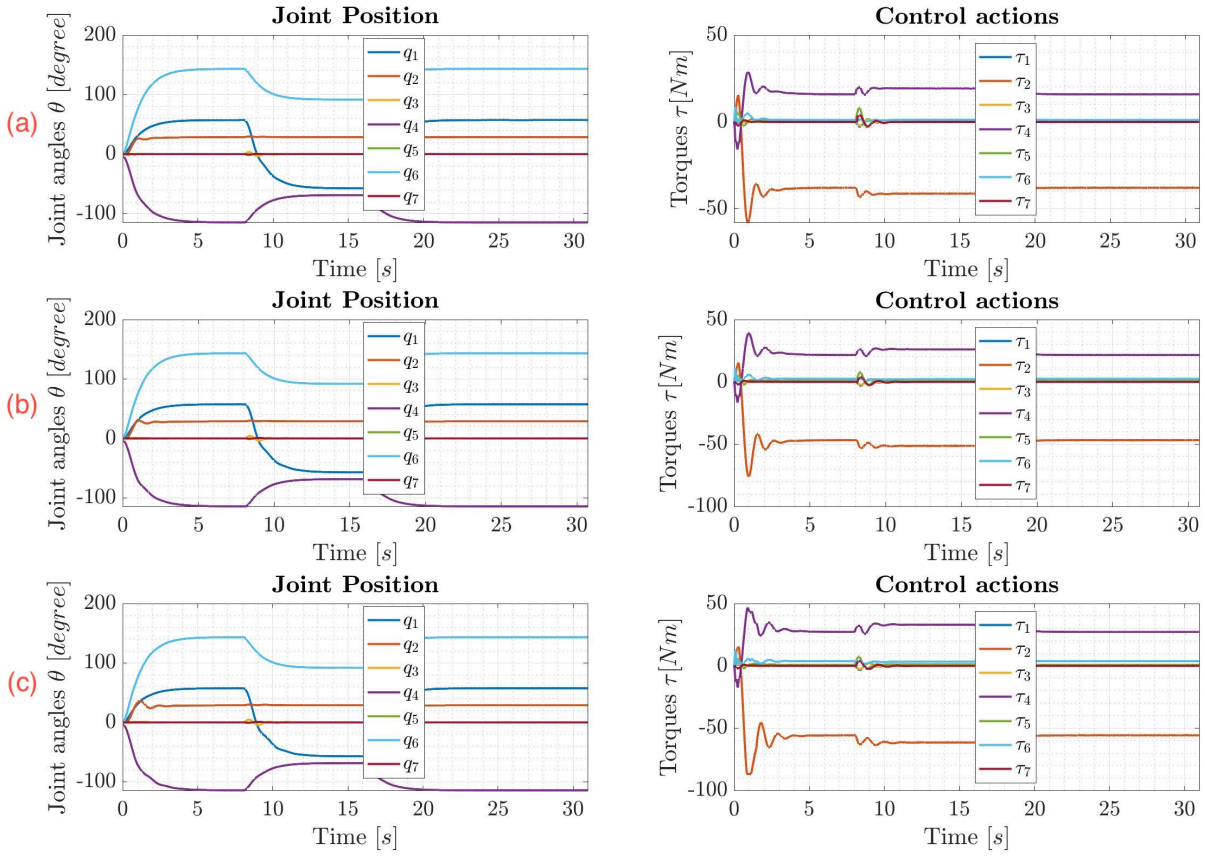


Figure 6.20: Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

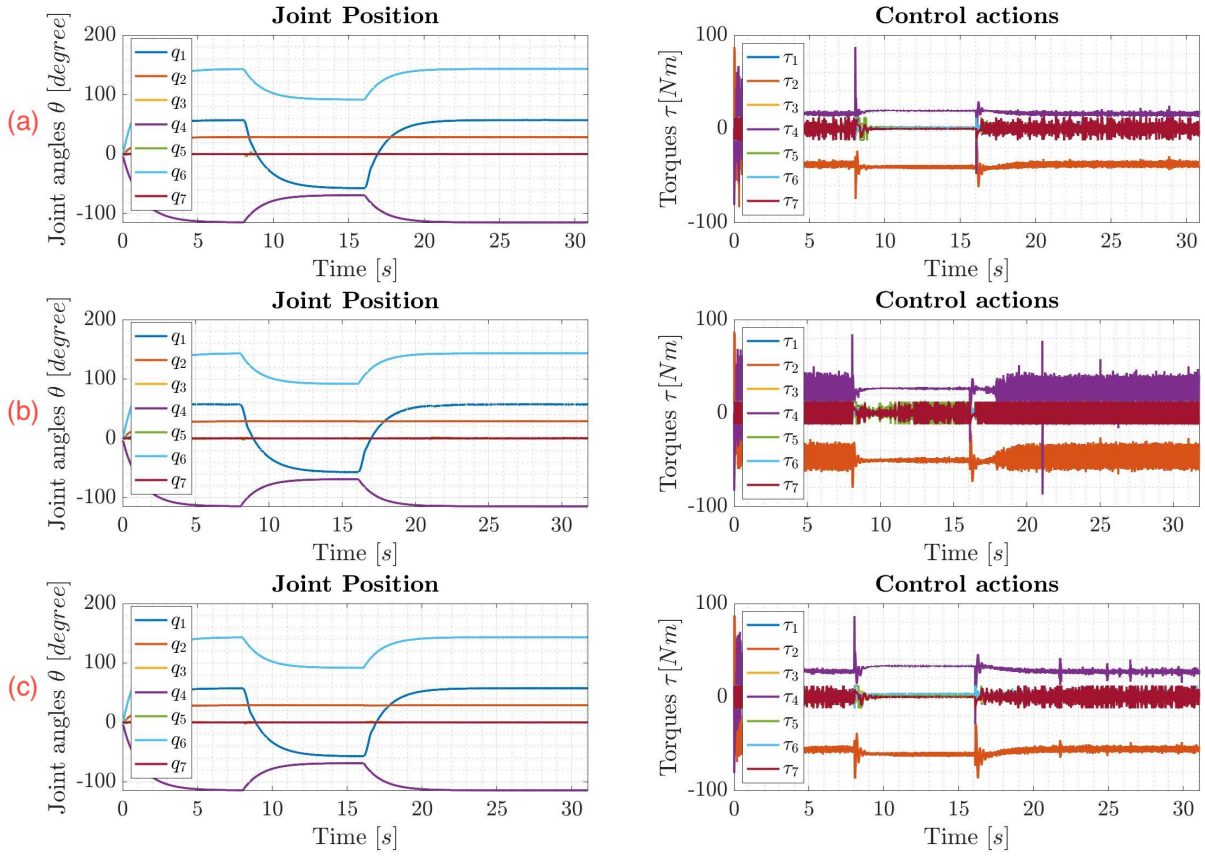


Figure 6.21: Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

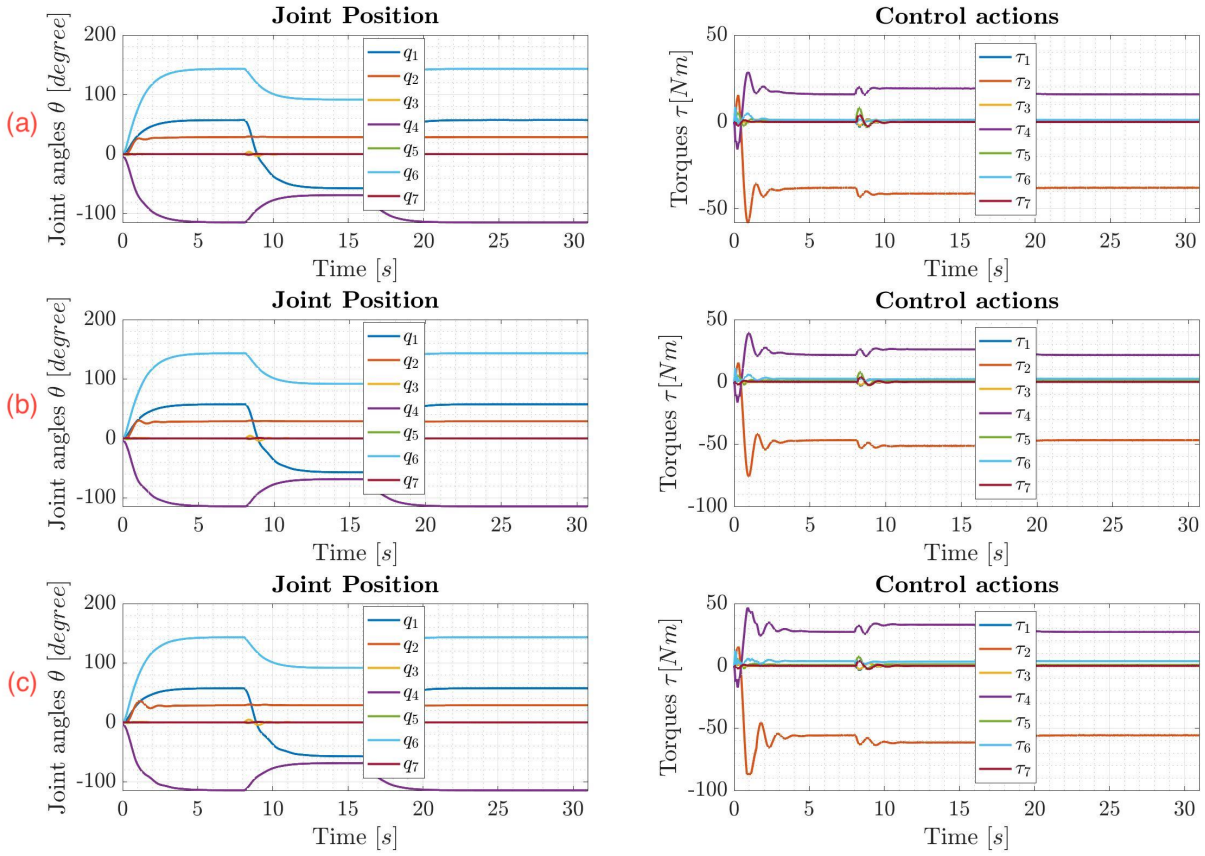


Figure 6.22: Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

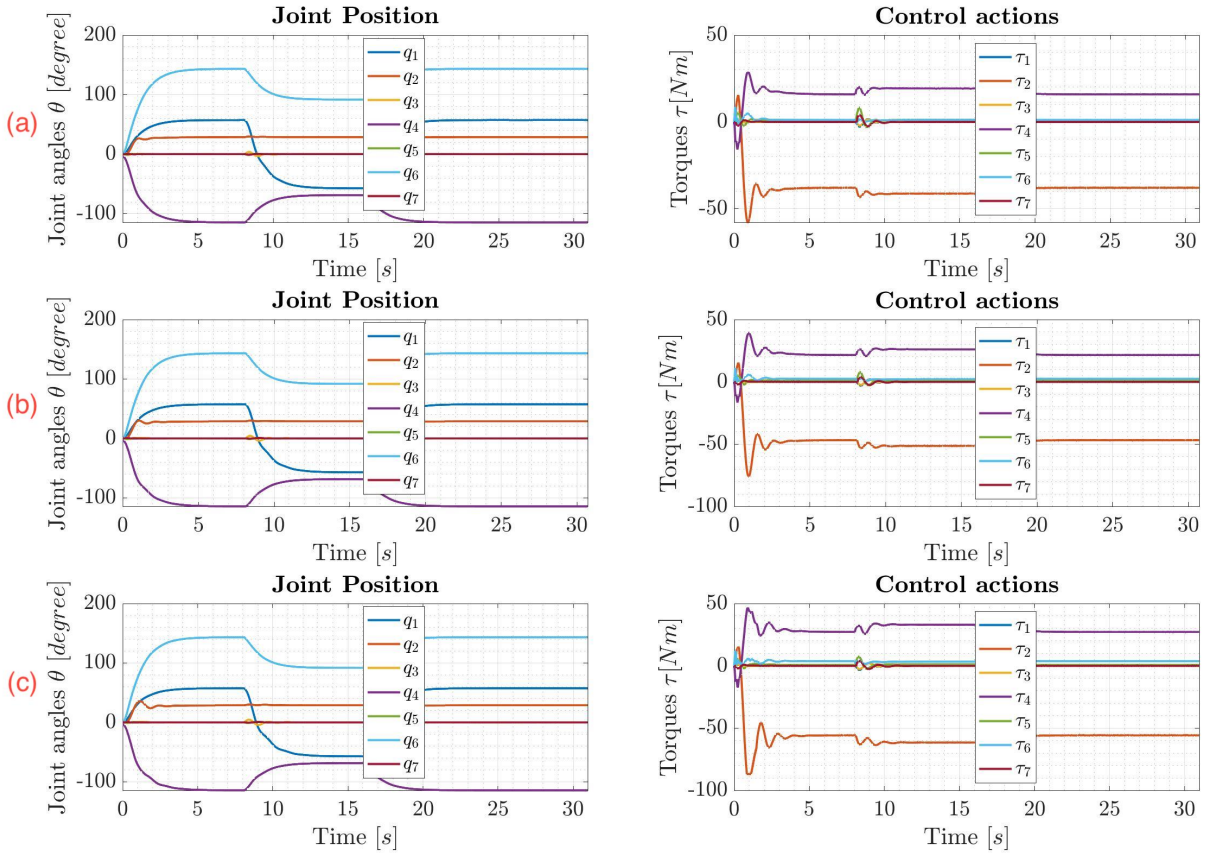


Figure 6.23: Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

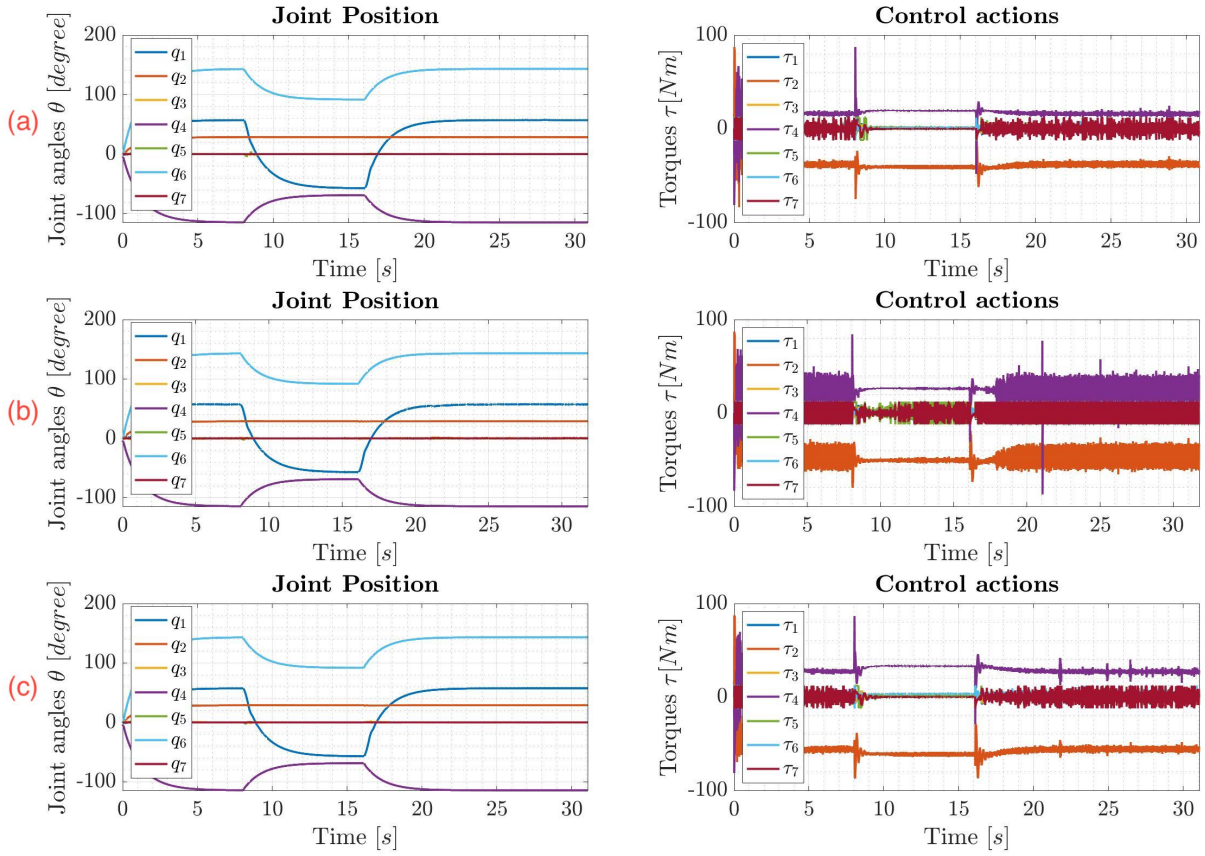


Figure 6.24: Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

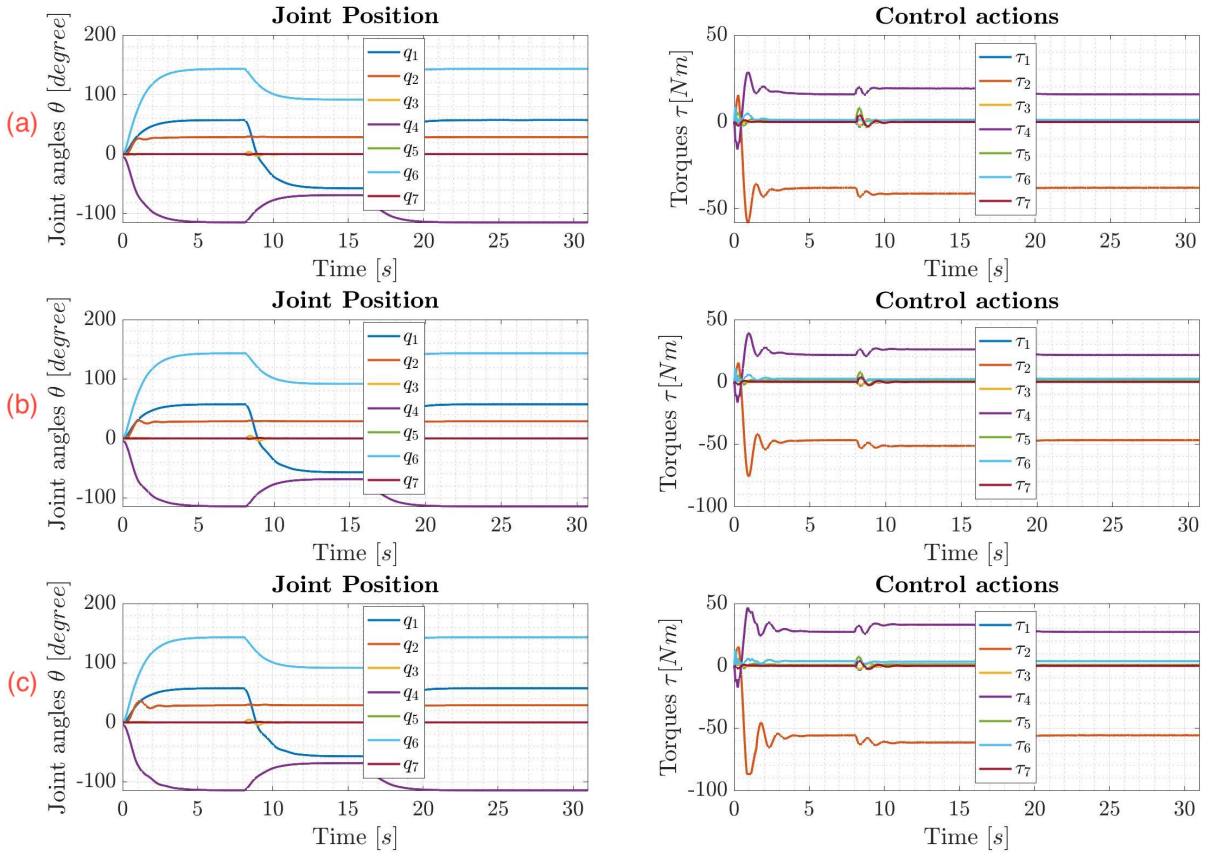


Figure 6.25: Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

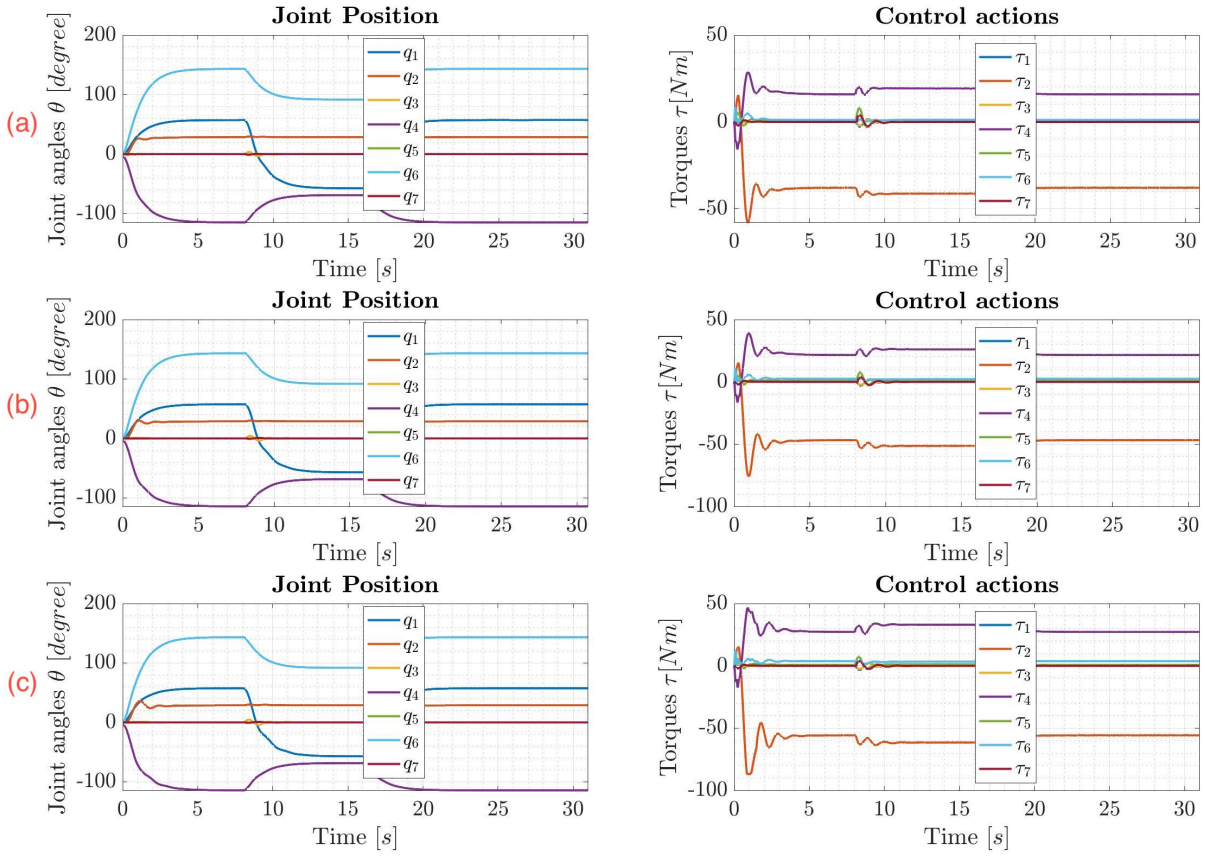


Figure 6.26: Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

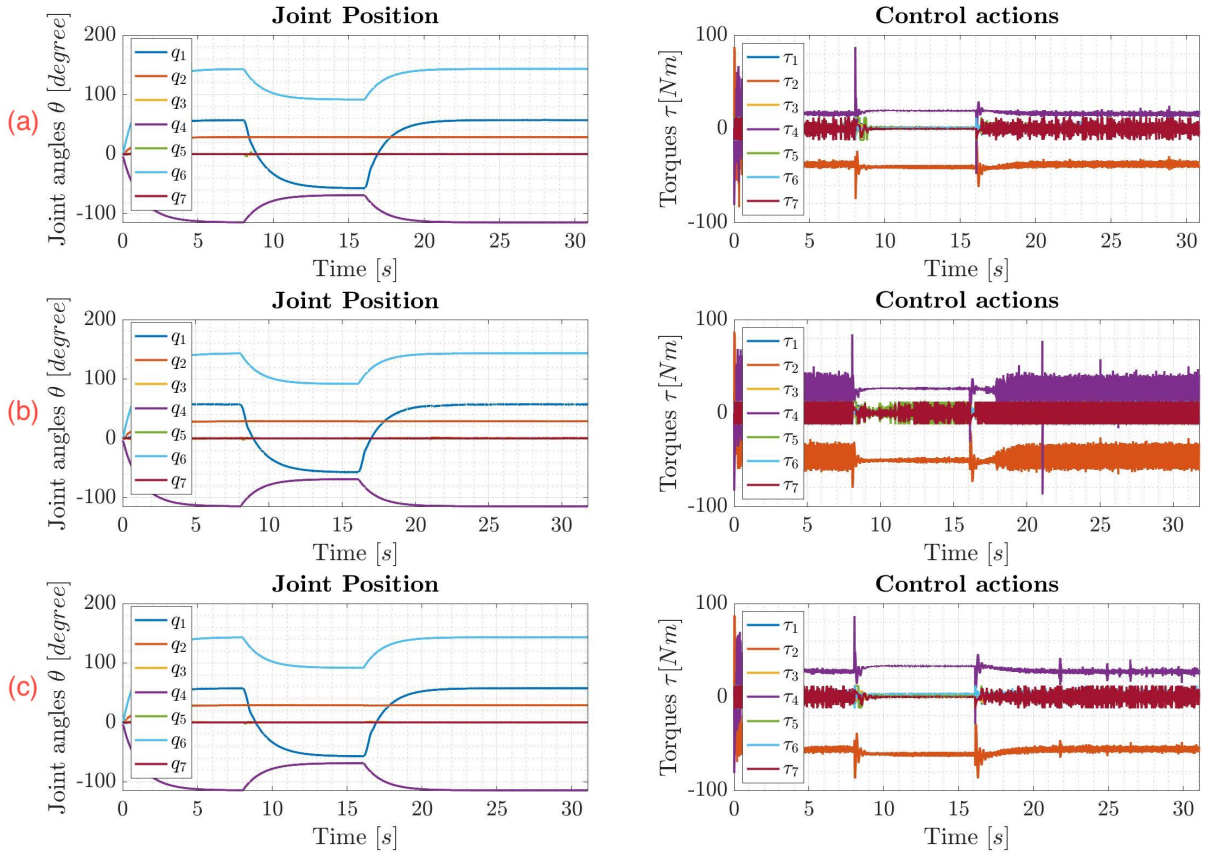


Figure 6.27: Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

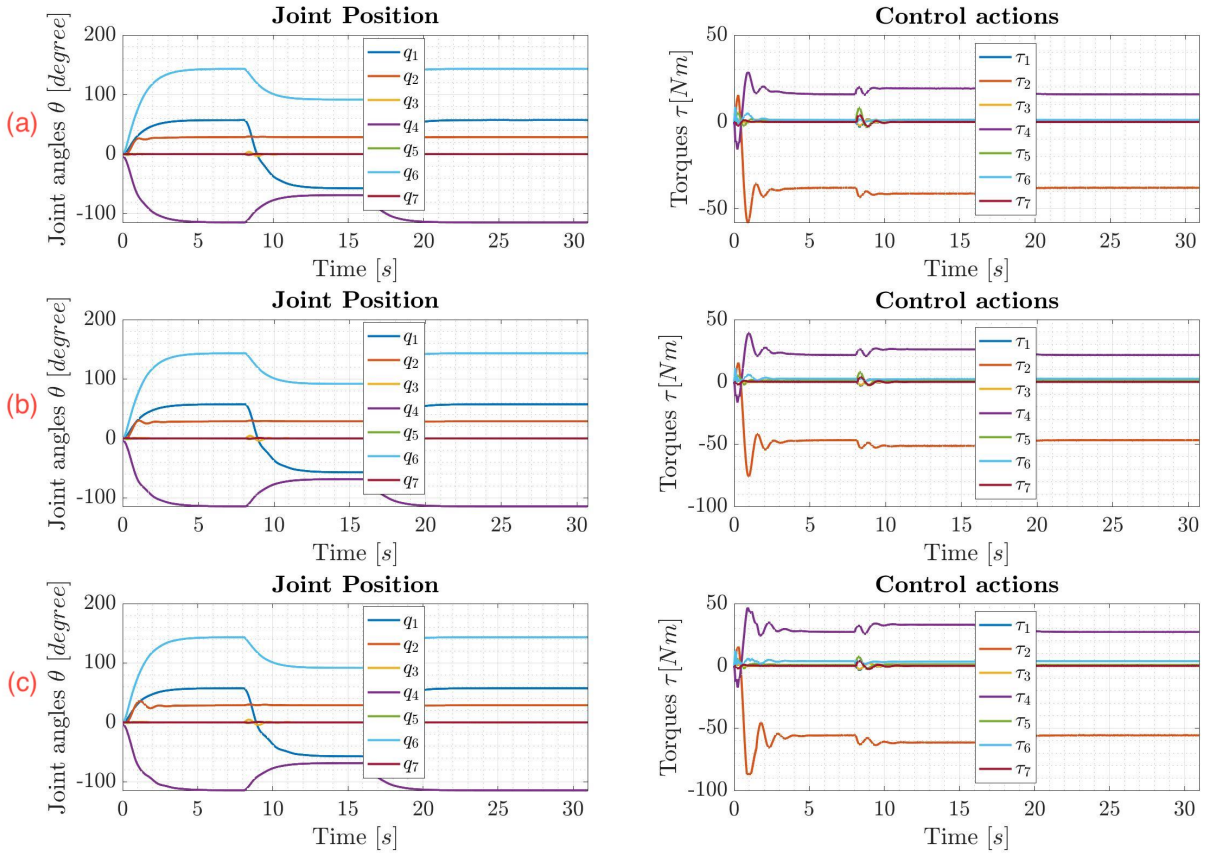


Figure 6.28: Adaptive sliding mode controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

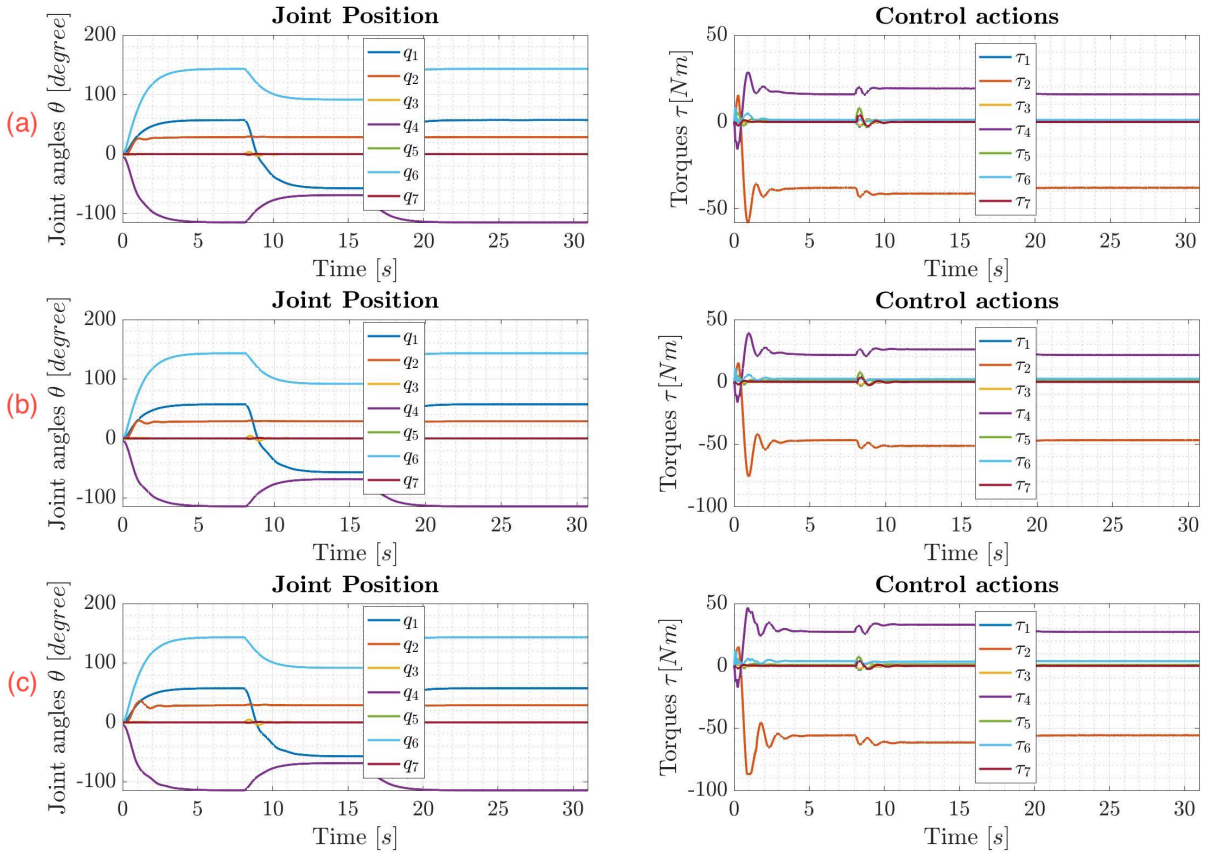


Figure 6.29: Adaptive model reference controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

The position errors of each joint during the operations under the payload variations are demonstrated in Figs. 6.31-6.61. For the first operation, Figs. 6.31-6.40 demonstrate the position errors of the joints for three robot configurations. The position errors of the joints for the second operation with the payload variations are shown in Figs. 6.41-6.50. For the last operation, the position errors of the joints with four different controllers are shown Figs. 6.51-6.61.

The proposed control schemes are effective in tracking the desired trajectories. From the results, it is concluded that the proposed ASMC and MRAC produce the best tracking output and they outperform PD and RPD controllers. It is also deduced that ASMC and MRAC can better handle the varying payloads compared with PID and RPD controllers. Furthermore, the developed ASMC and MRAC outperform the PD and RPD without increasing the control efforts. Furthermore, due to the varying payload, when the number of d.o.f increases, gravity plays an important role in amplifying the disturbance effects. Consequently, adaptive control structures are well implemented to perform a desired precise positioning under any operation of the robot.

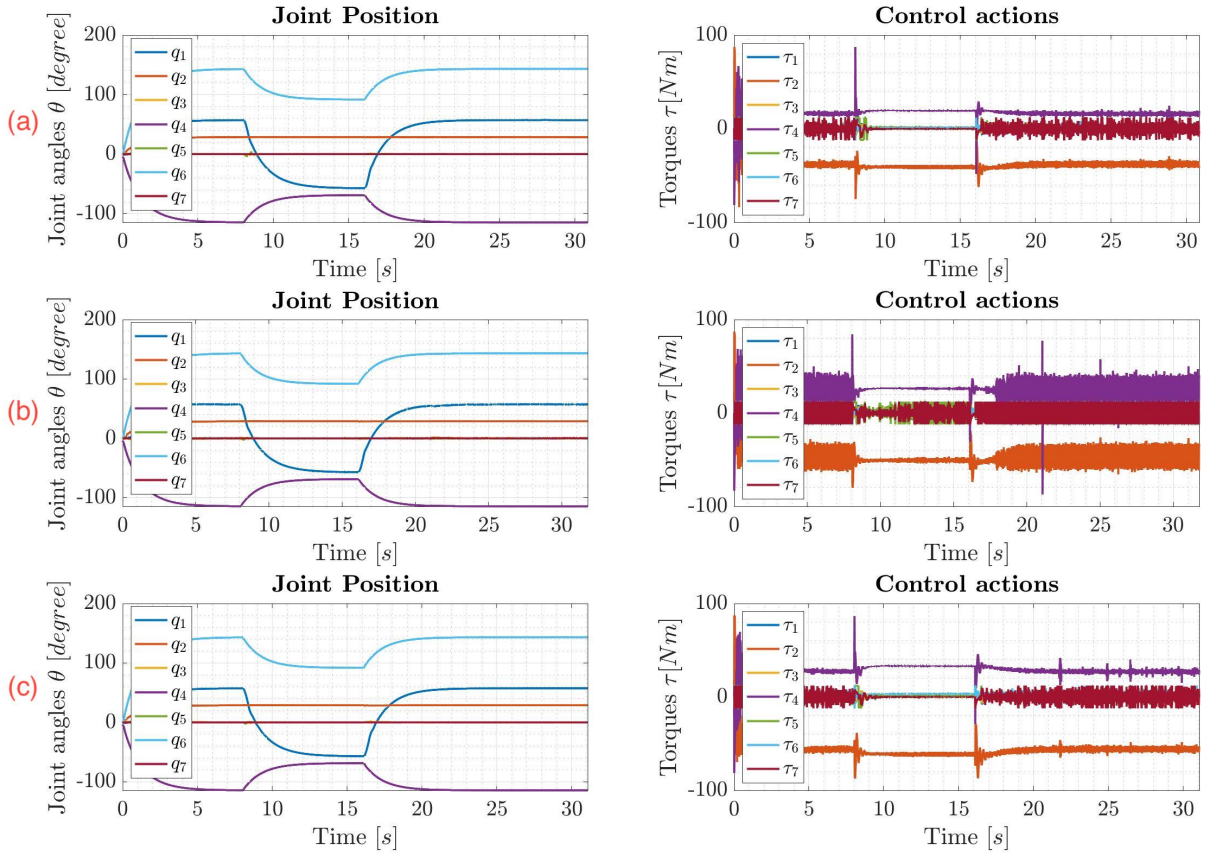


Figure 6.30: Robust PD controller's performance of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for Configuration 2.

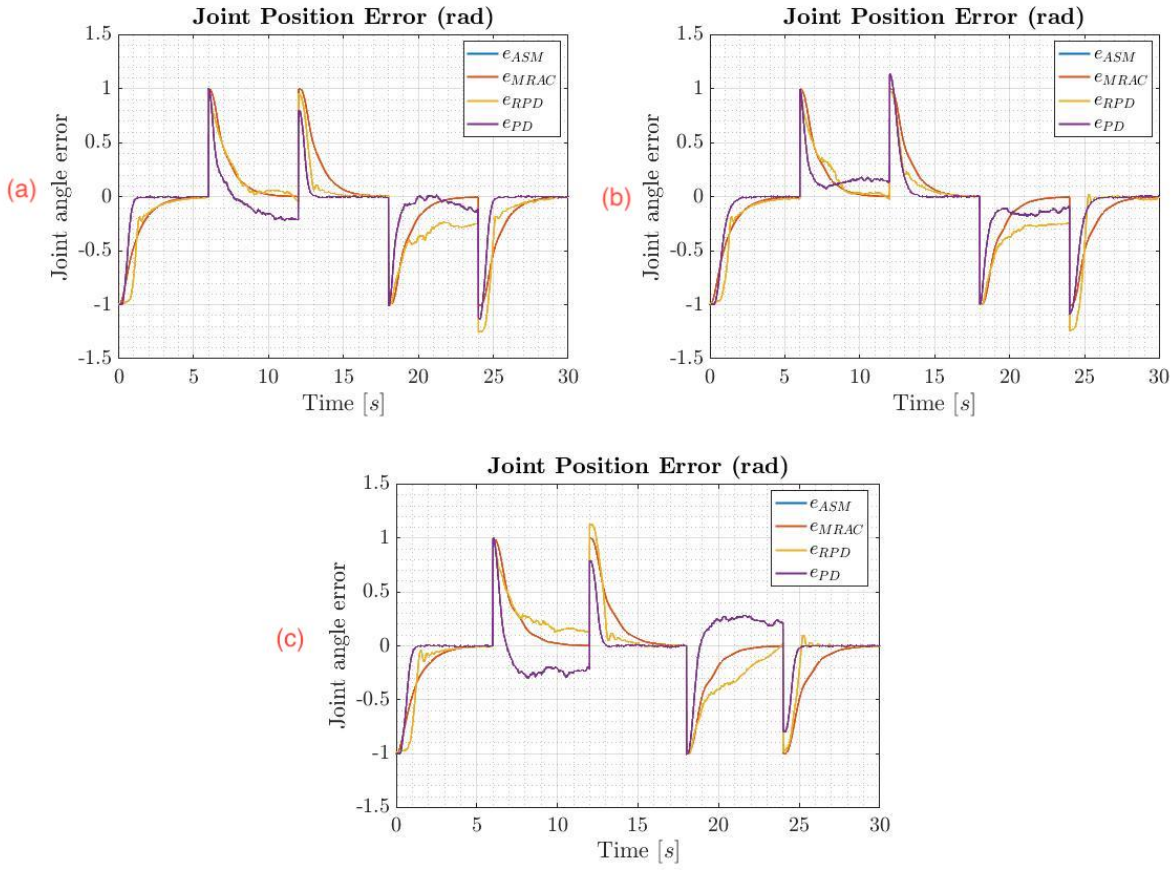


Figure 6.31: Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 1.

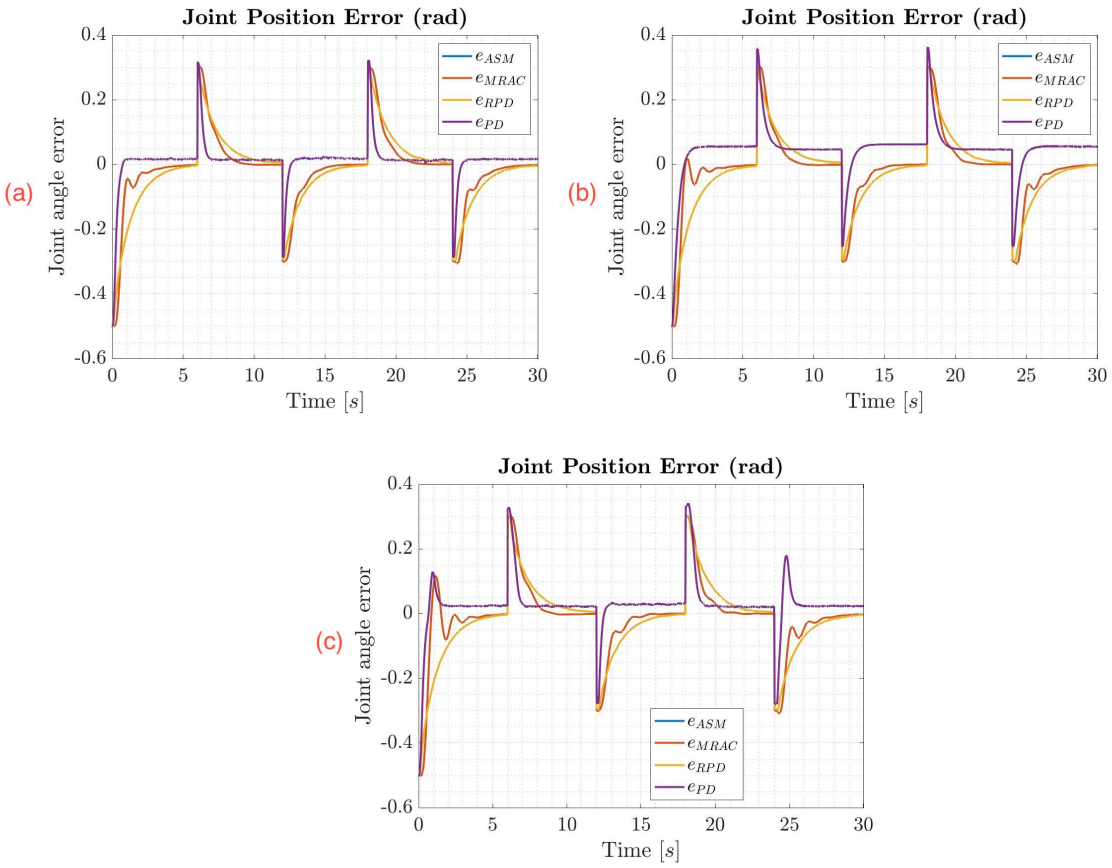


Figure 6.32: Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 1.

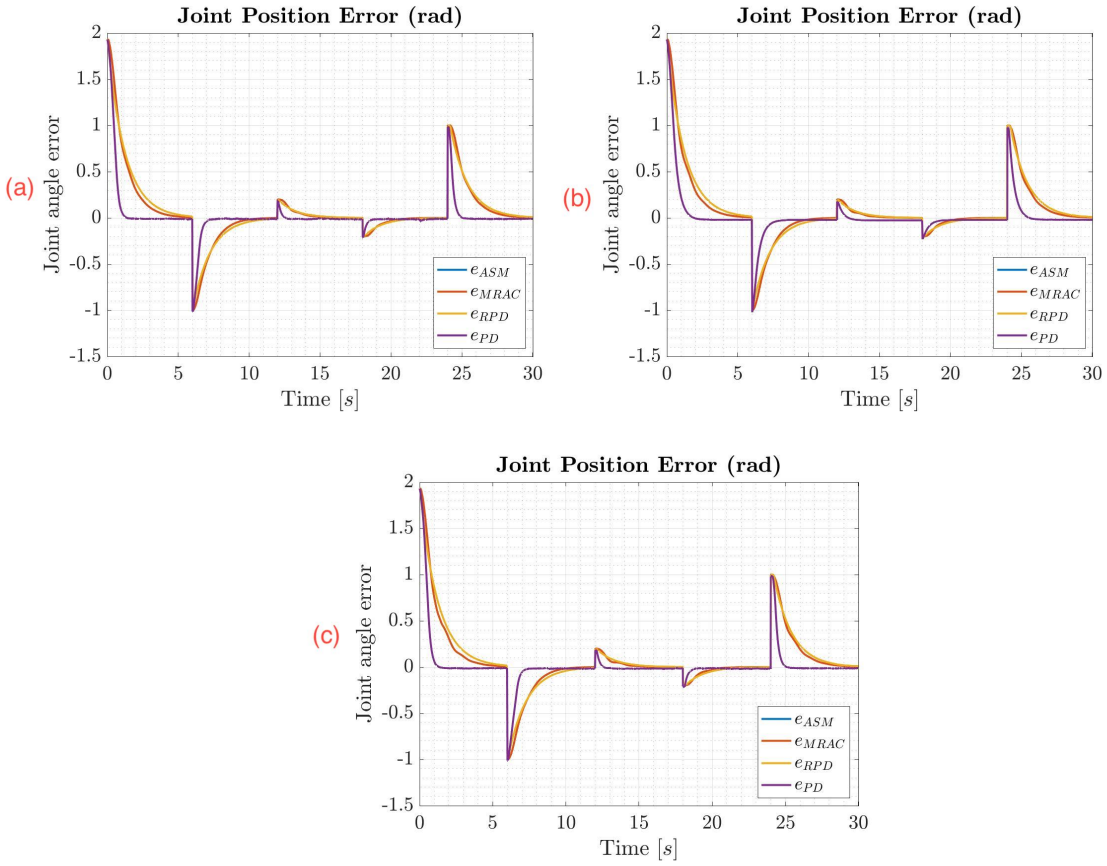


Figure 6.33: Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 1.

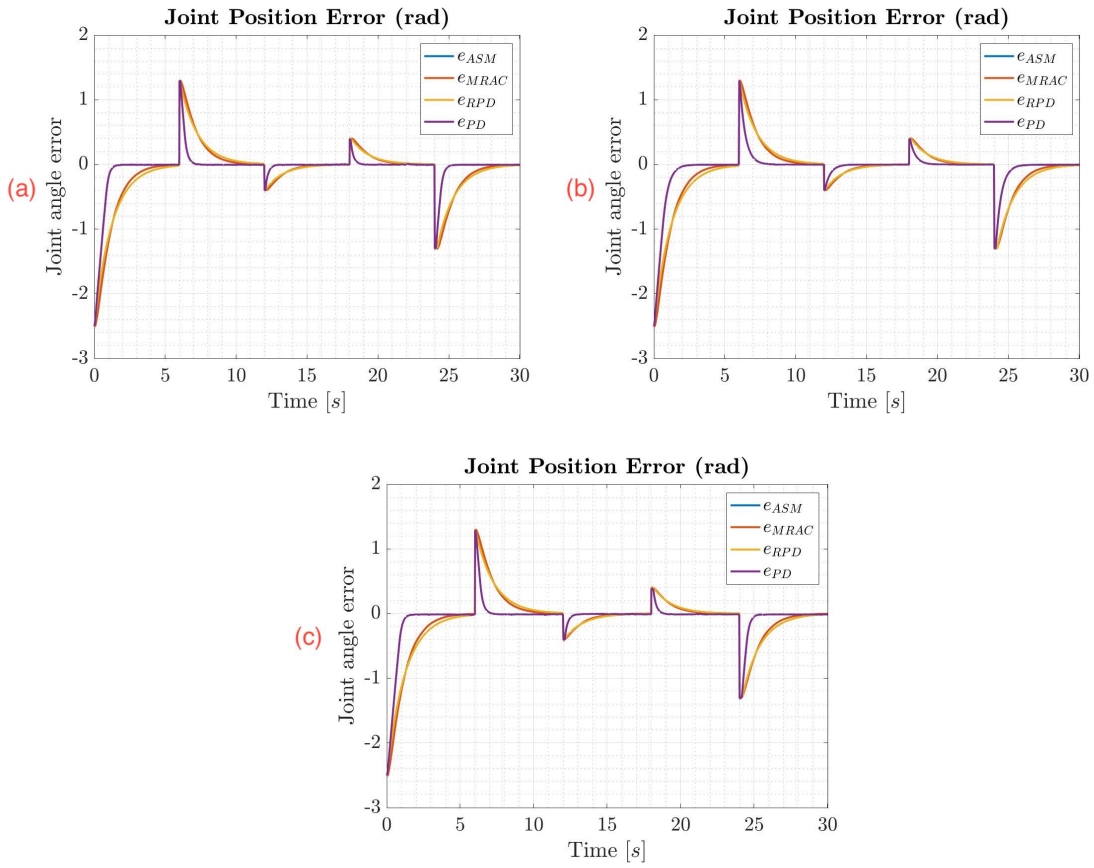


Figure 6.34: Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 1.

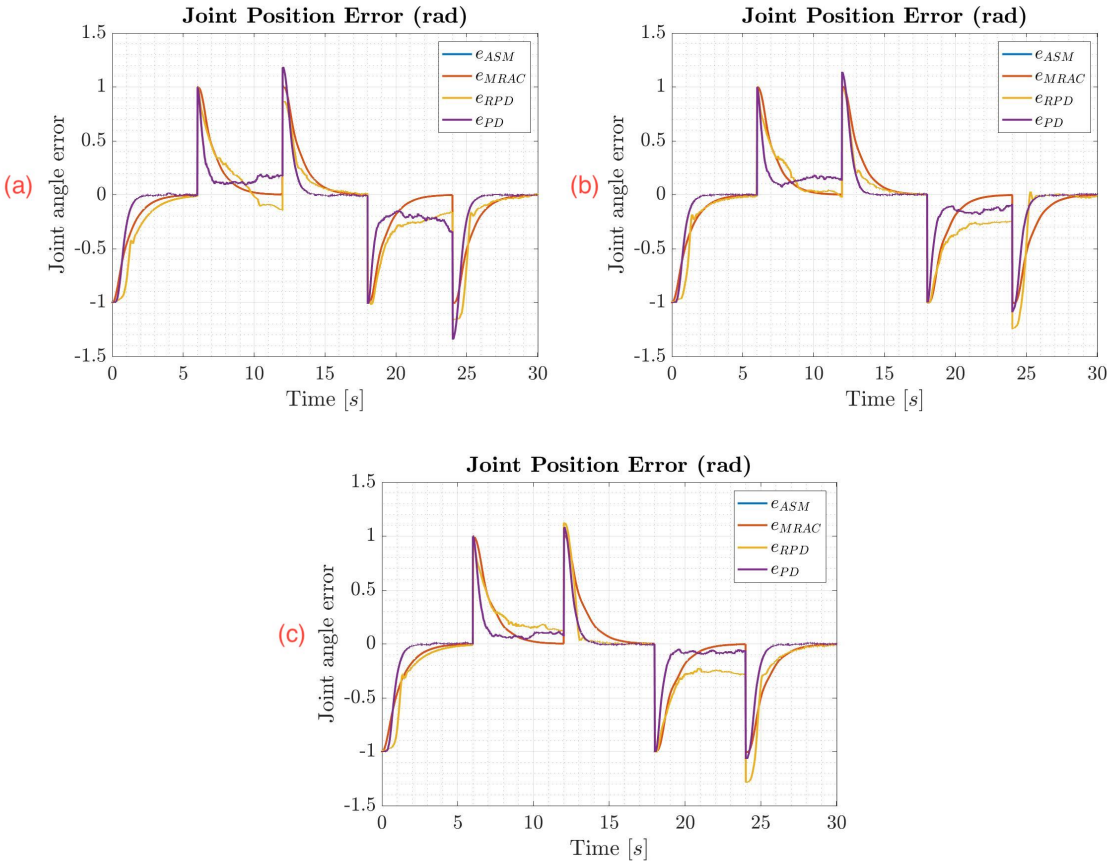


Figure 6.35: Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 2.

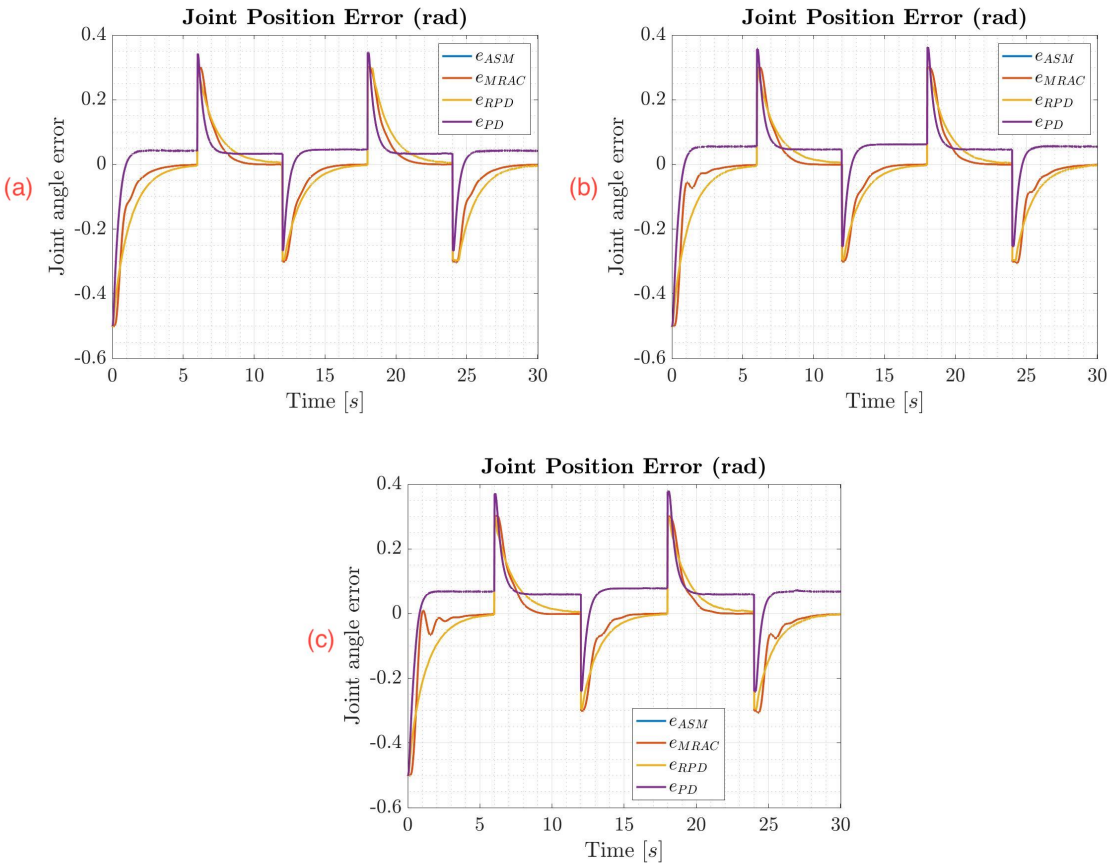


Figure 6.36: Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 2.

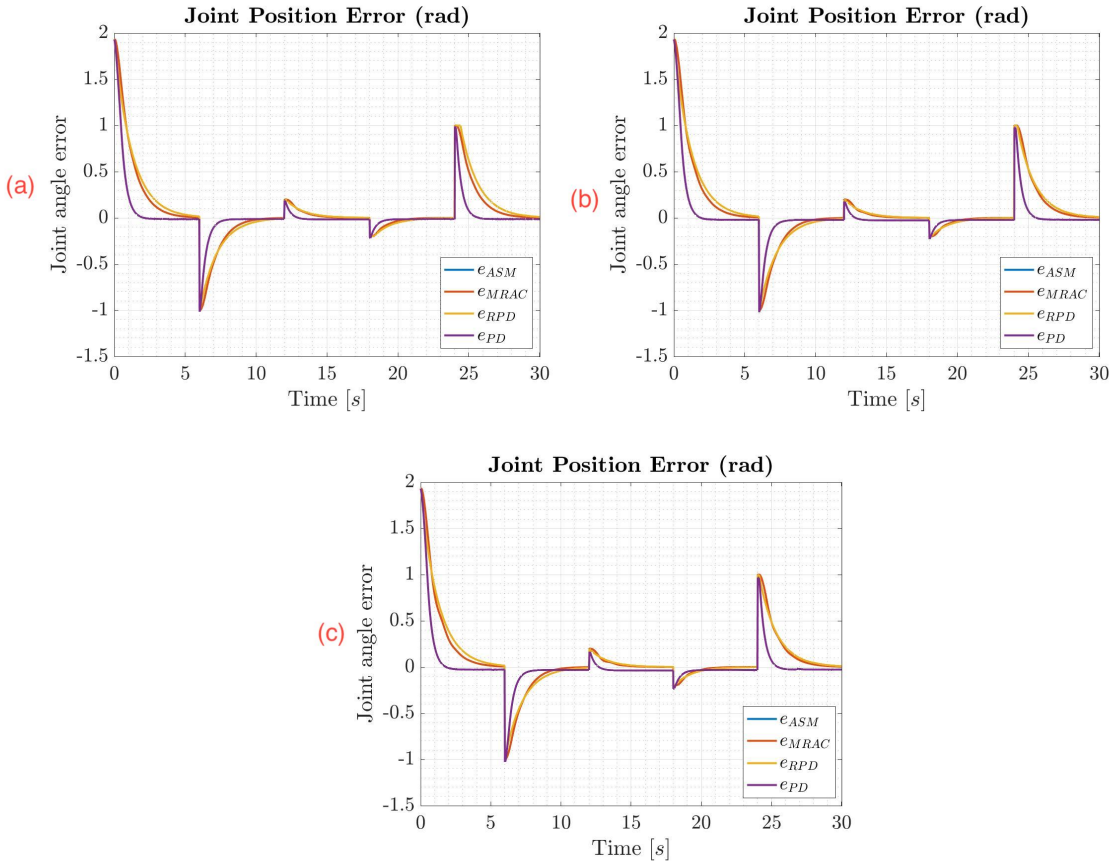


Figure 6.37: Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 2.

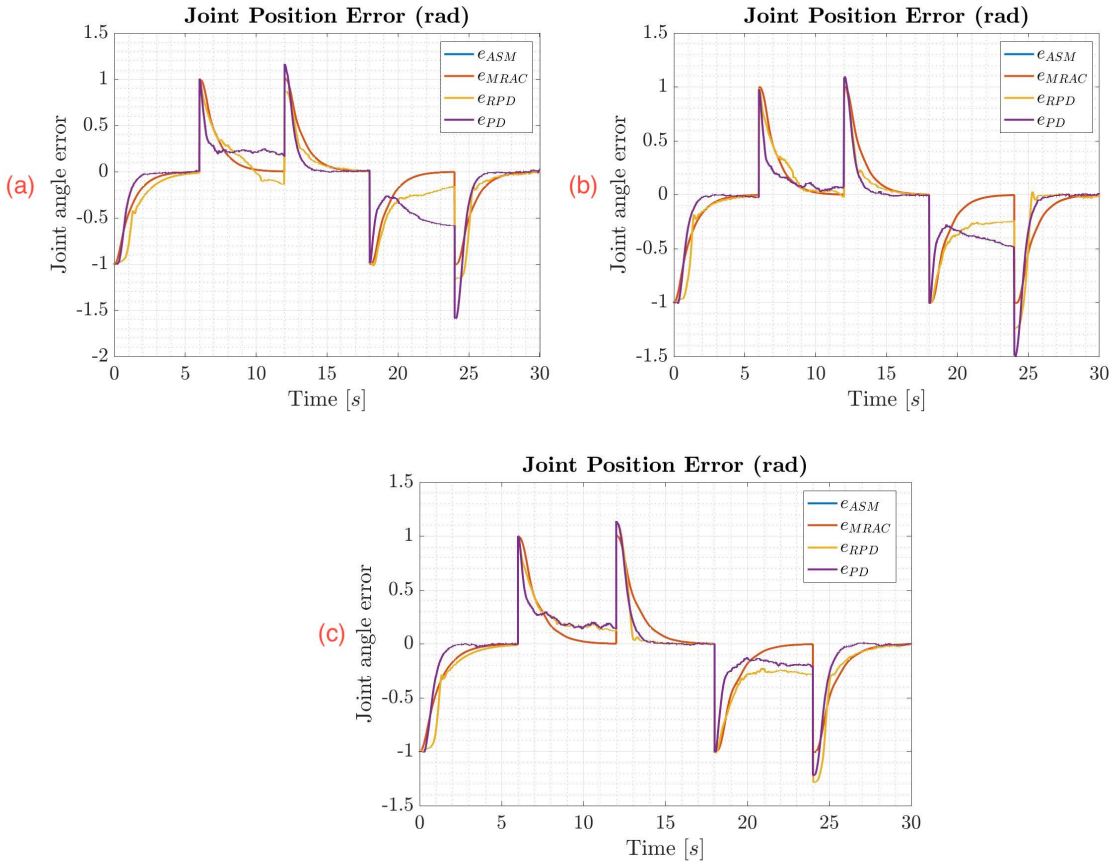


Figure 6.38: Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 3.

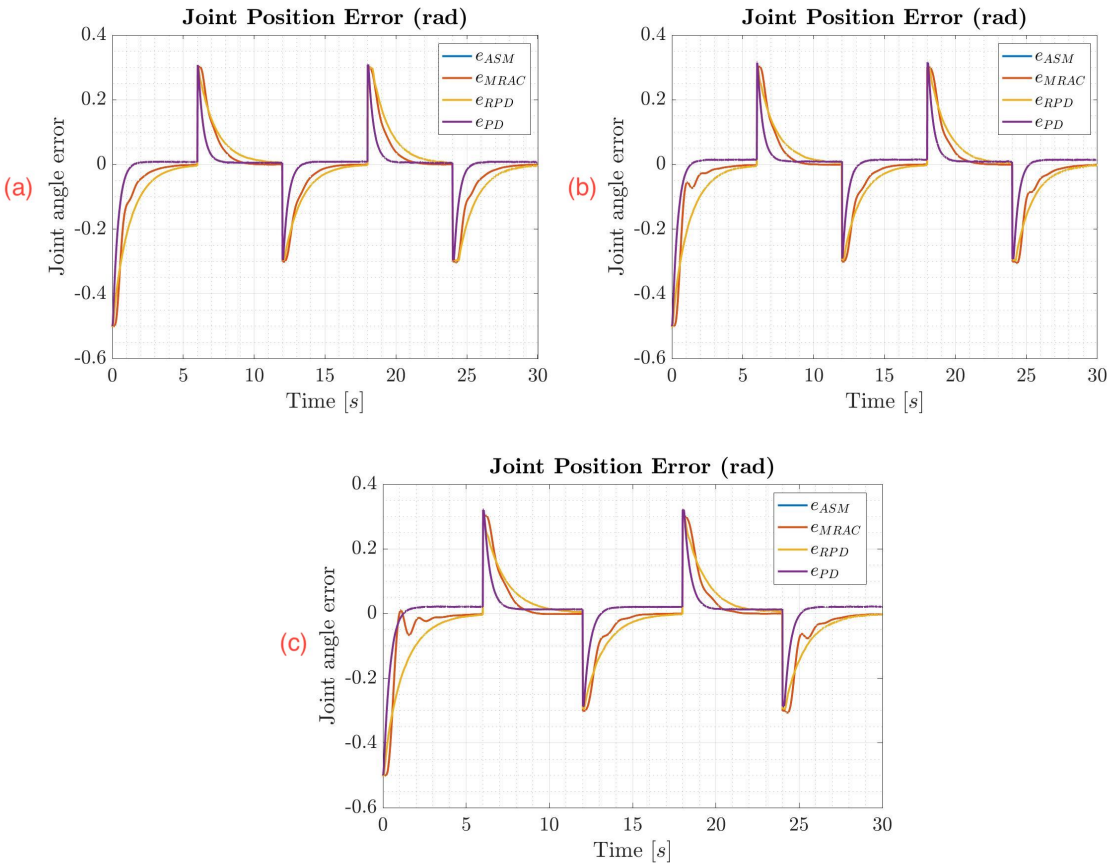


Figure 6.39: Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 3.

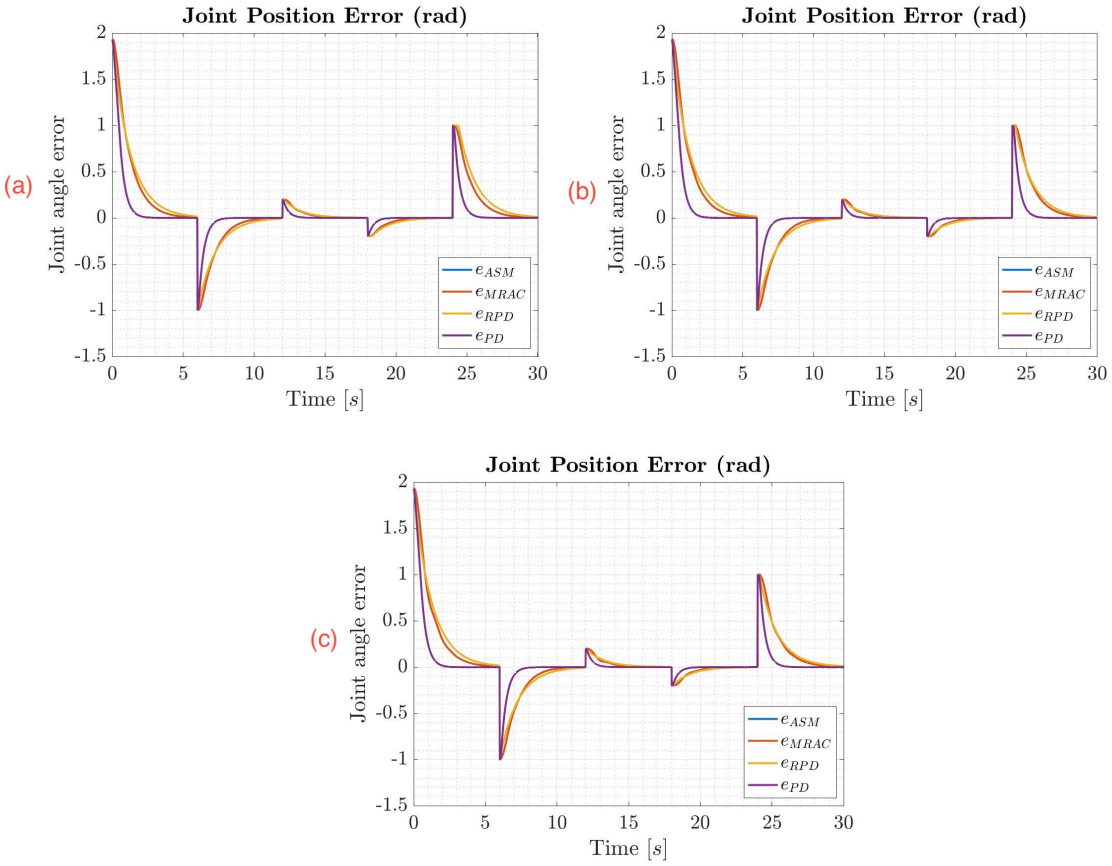


Figure 6.40: Position error of the first operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 3.

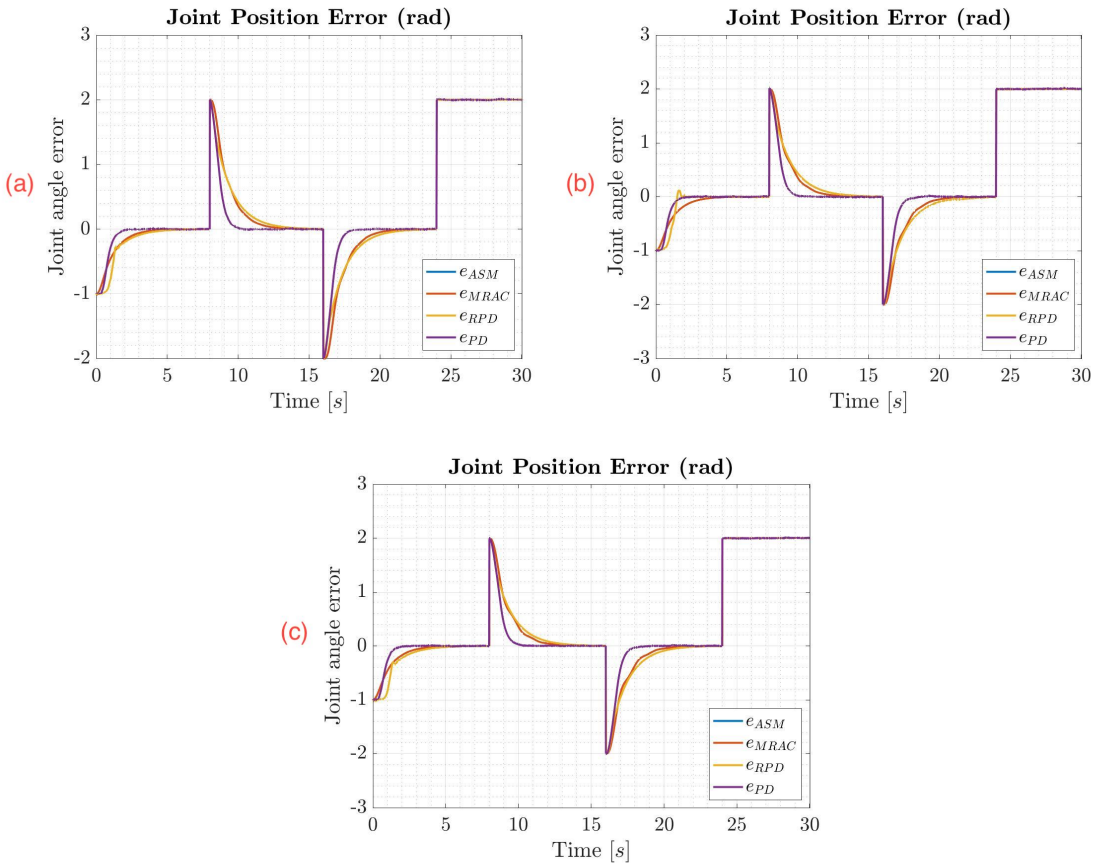


Figure 6.41: Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 1.

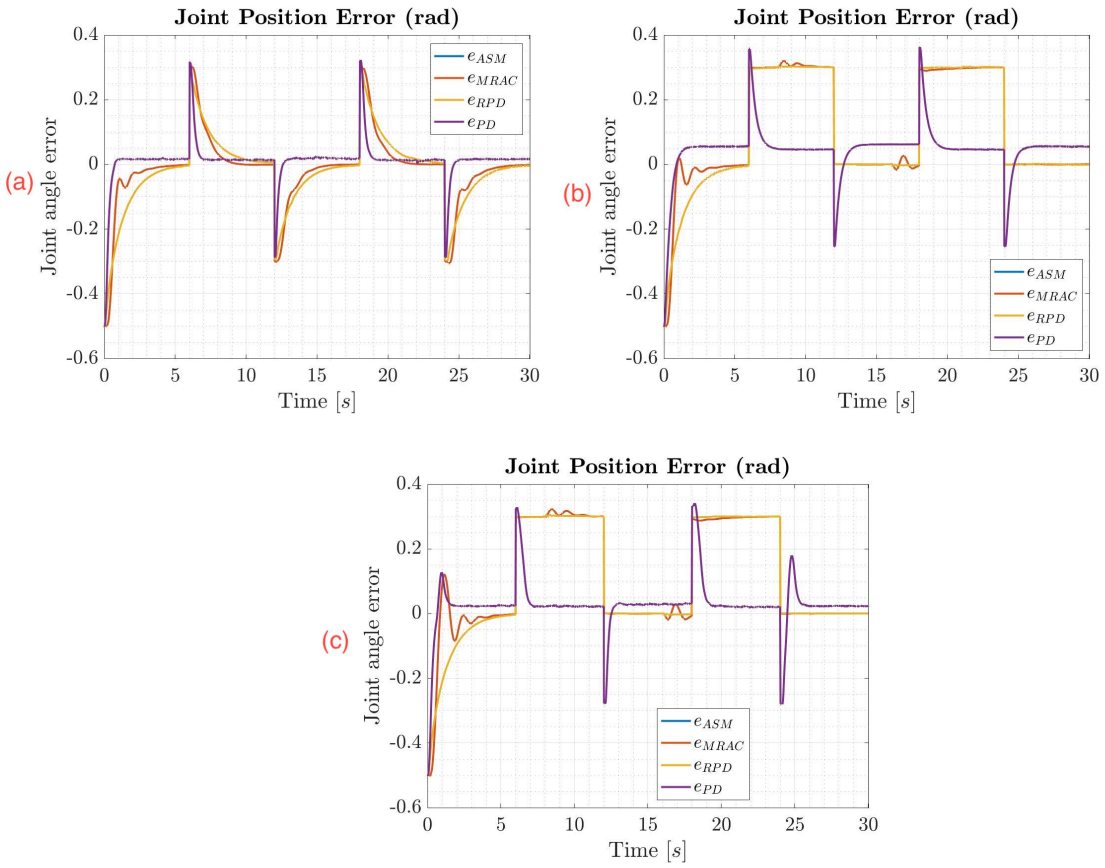


Figure 6.42: Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 1.

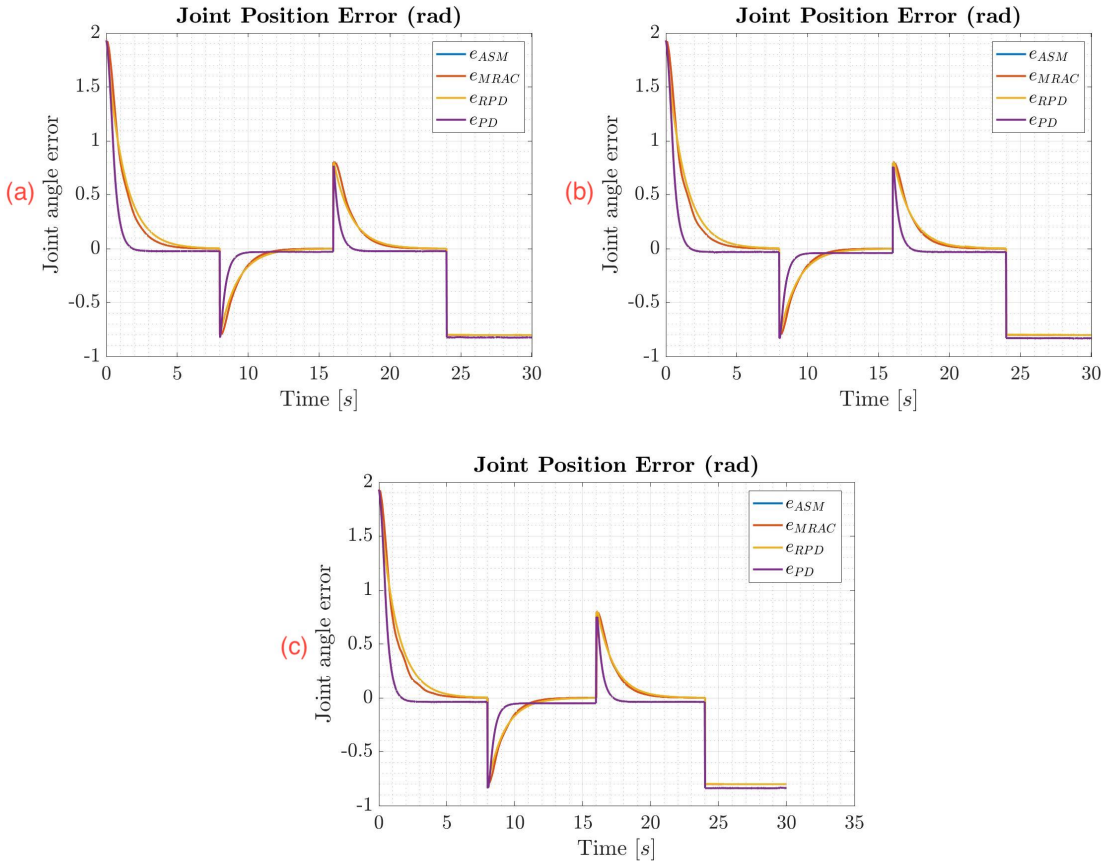


Figure 6.43: Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 1.

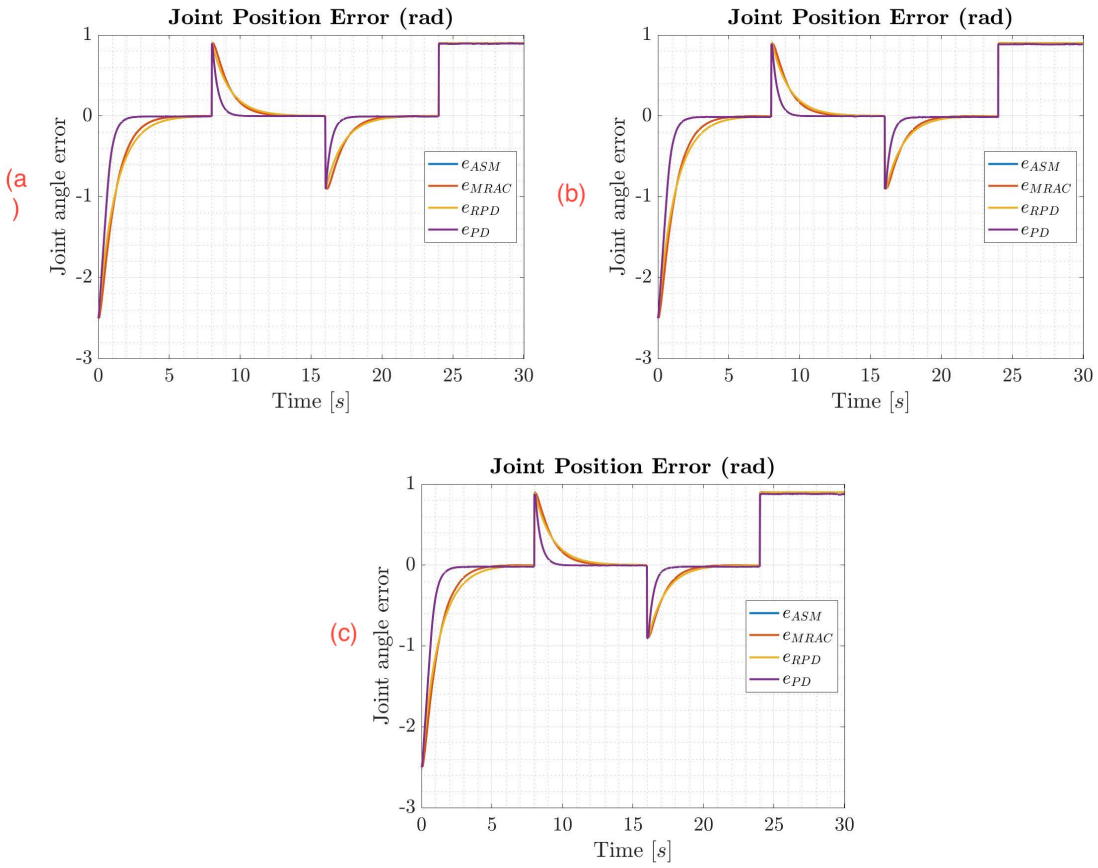


Figure 6.44: Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 1.

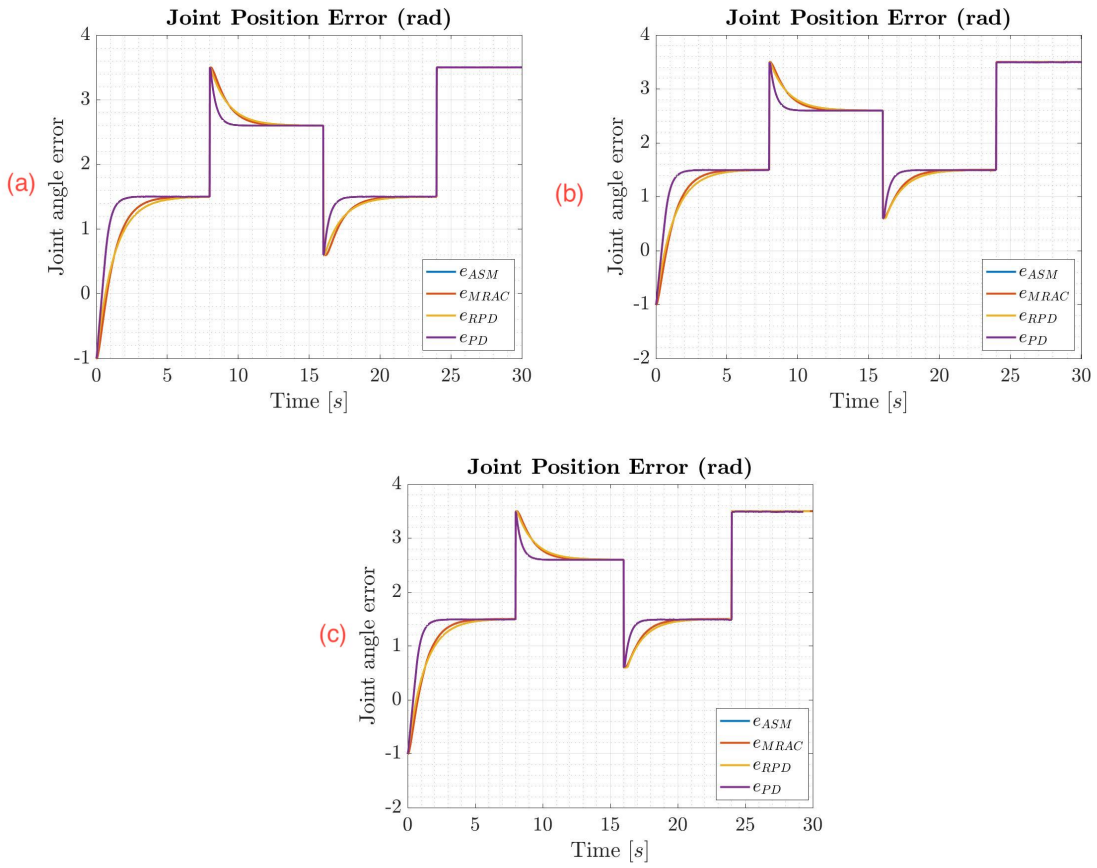


Figure 6.45: Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 2.

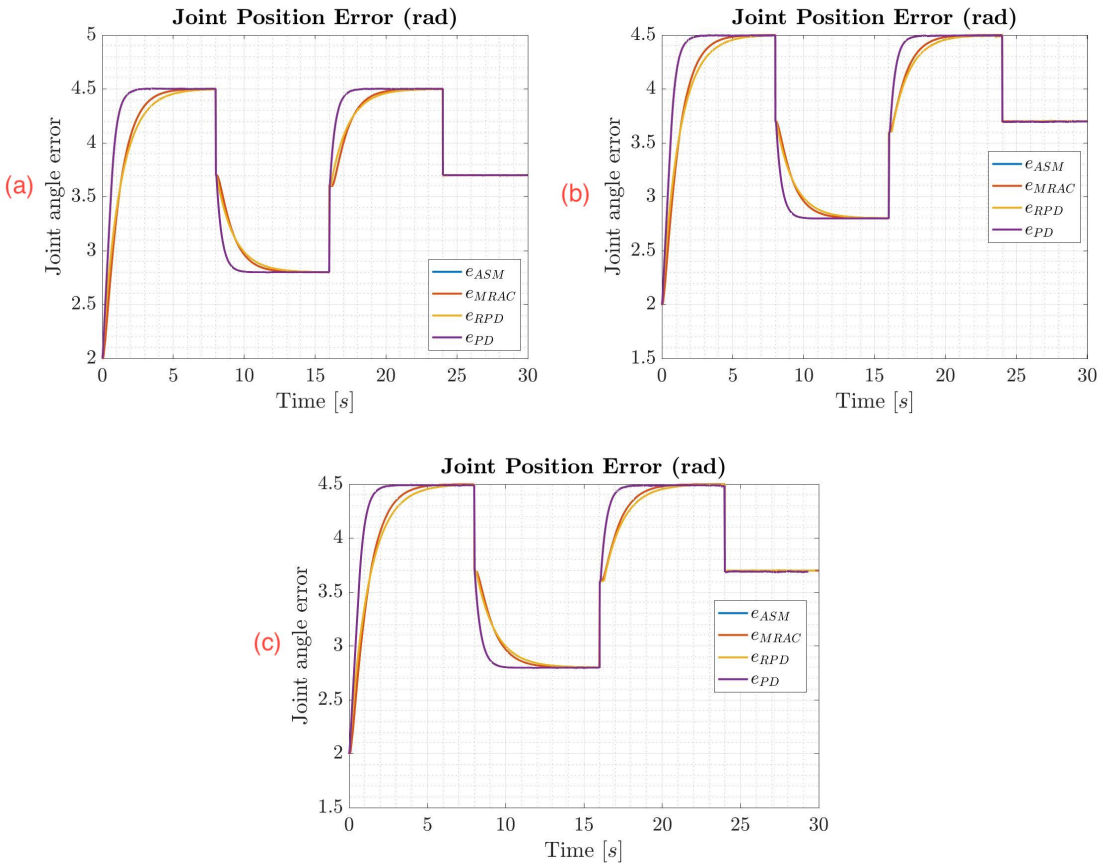


Figure 6.46: Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 2.

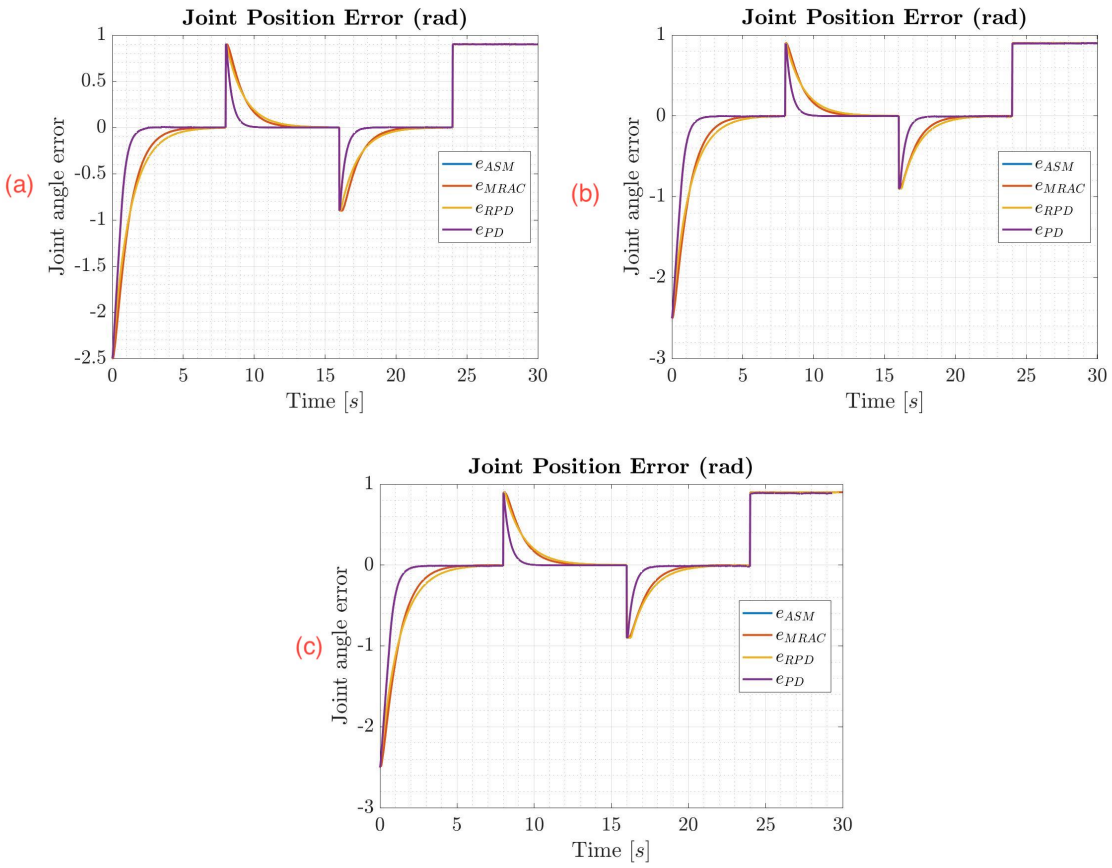


Figure 6.47: Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 2.

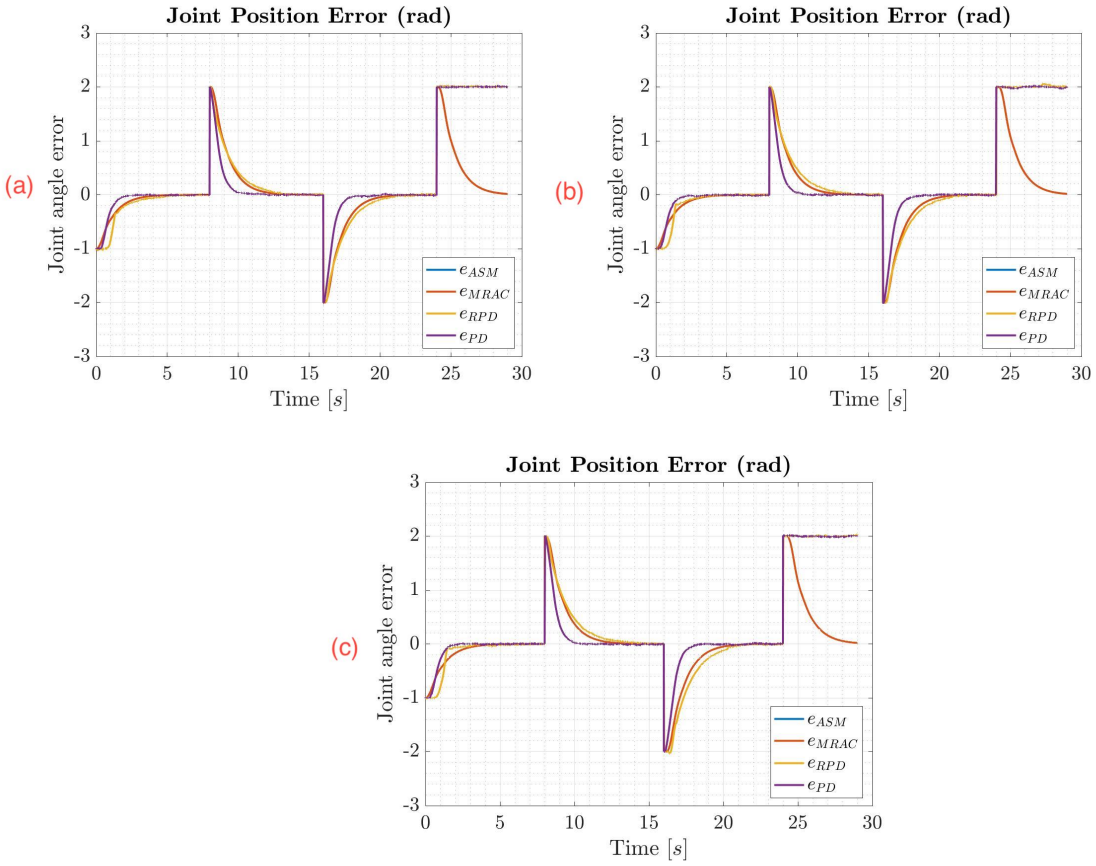


Figure 6.48: Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 3.

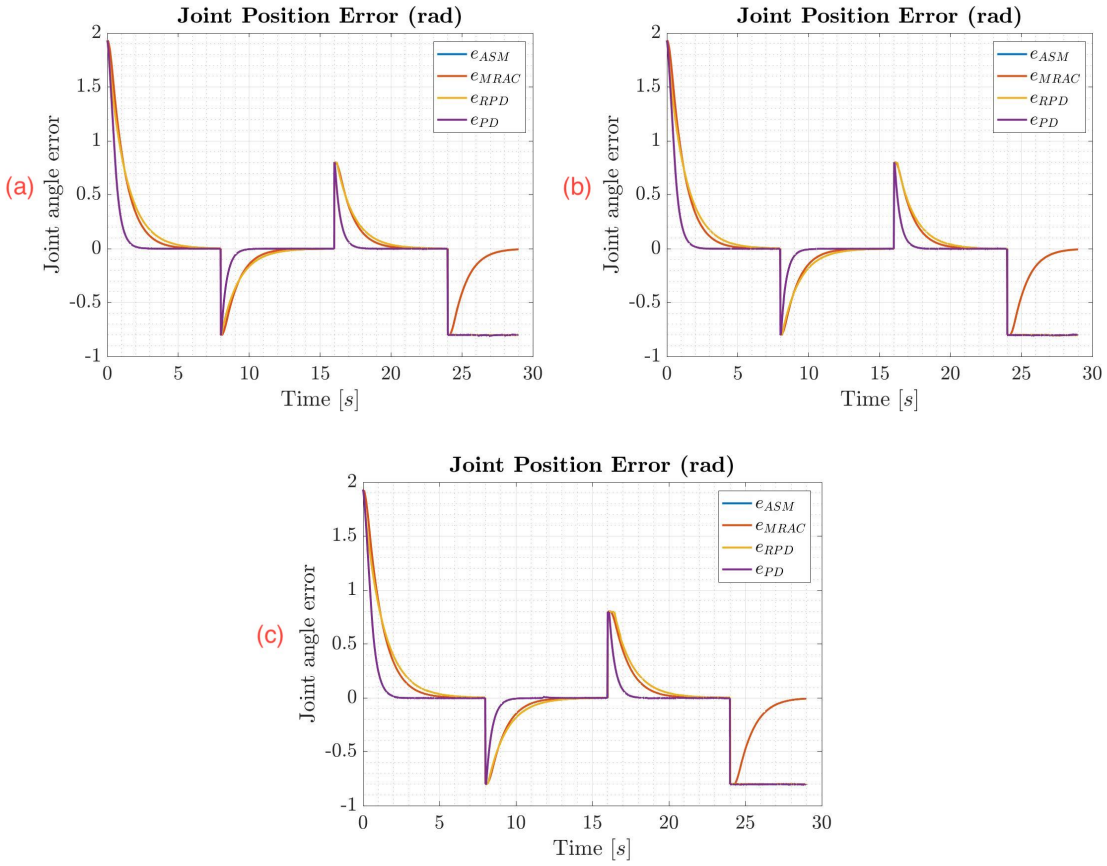


Figure 6.49: Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 3.

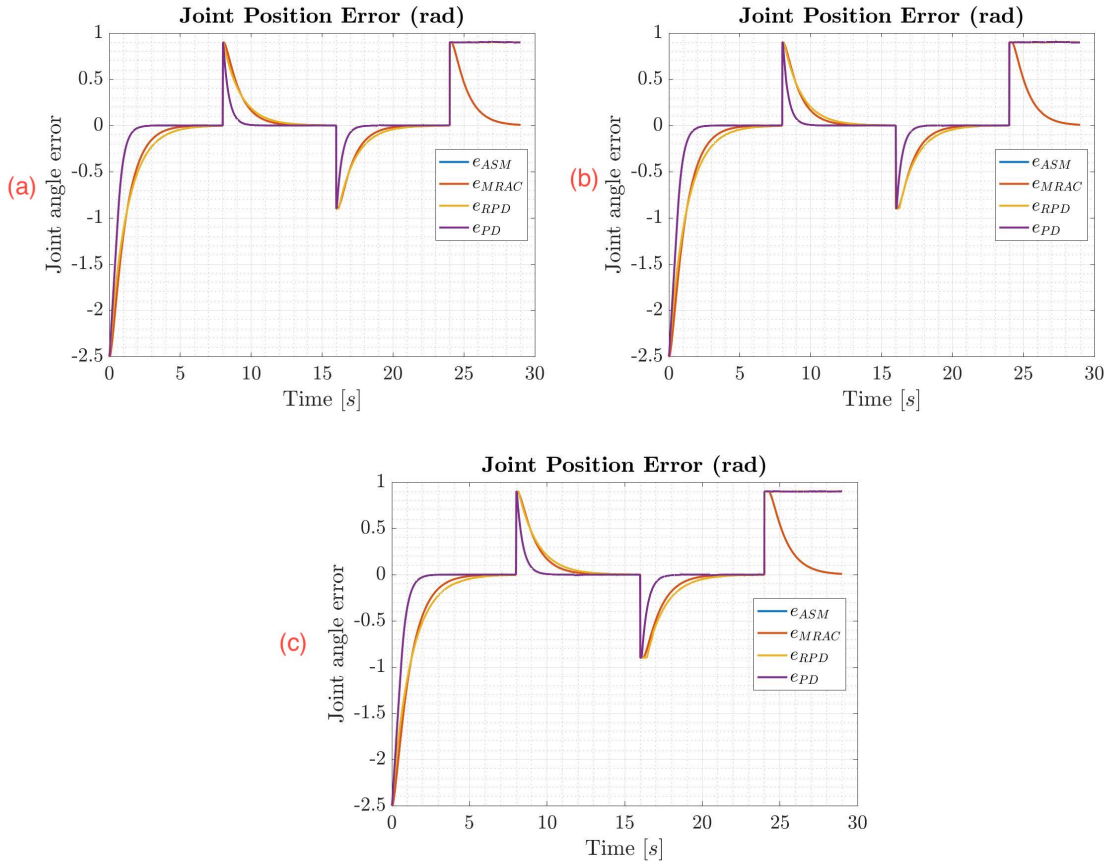


Figure 6.50: Position error of the second operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 3.

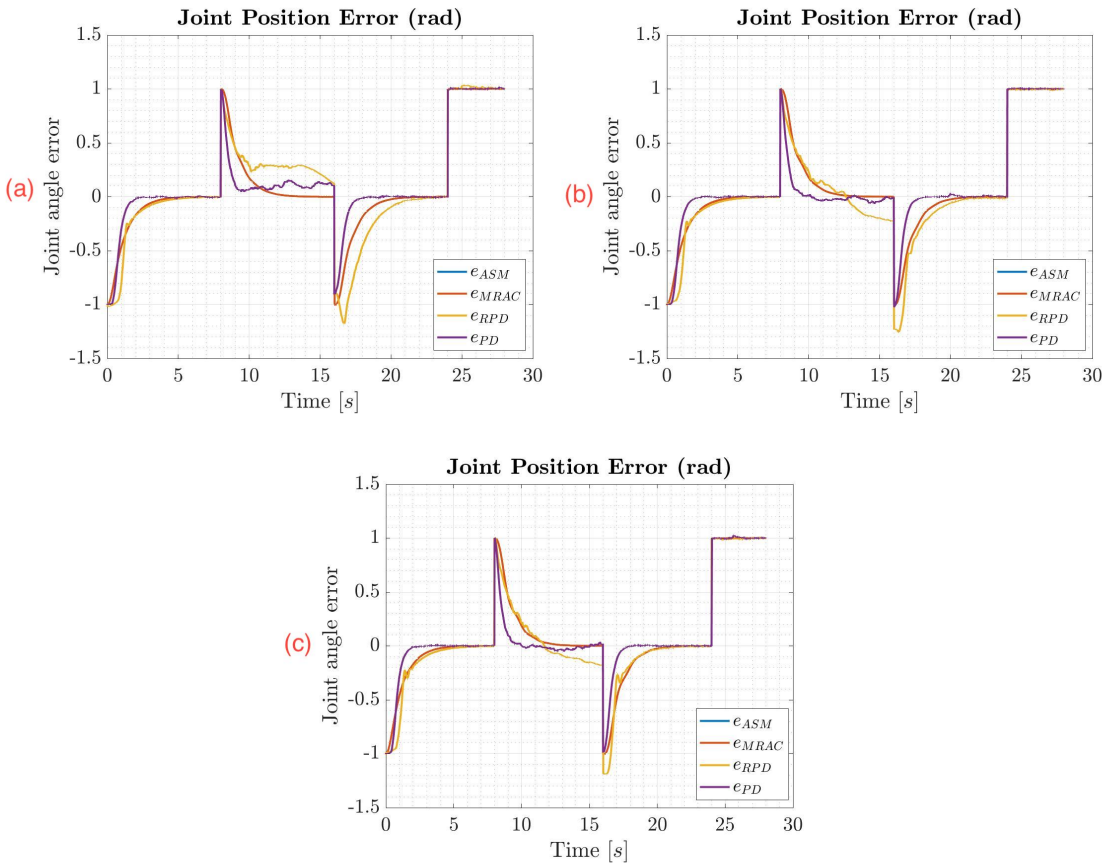


Figure 6.51: Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 1.

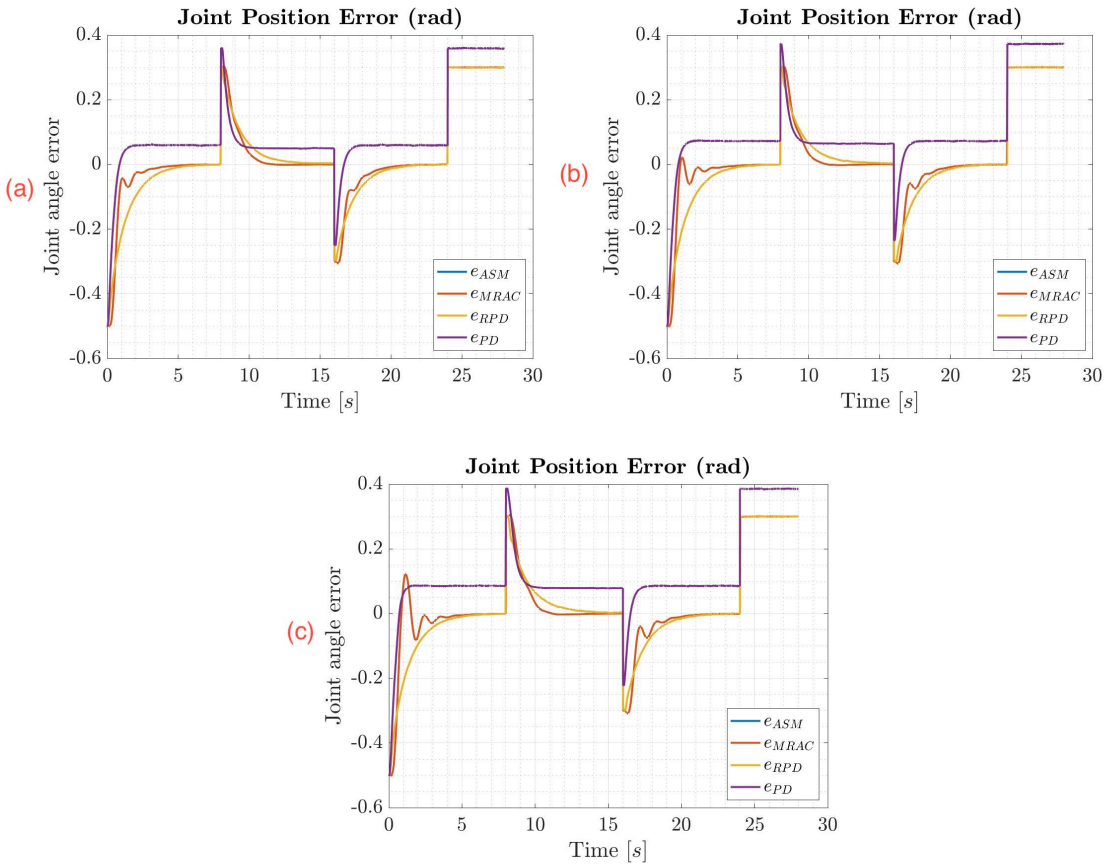


Figure 6.52: Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 1.

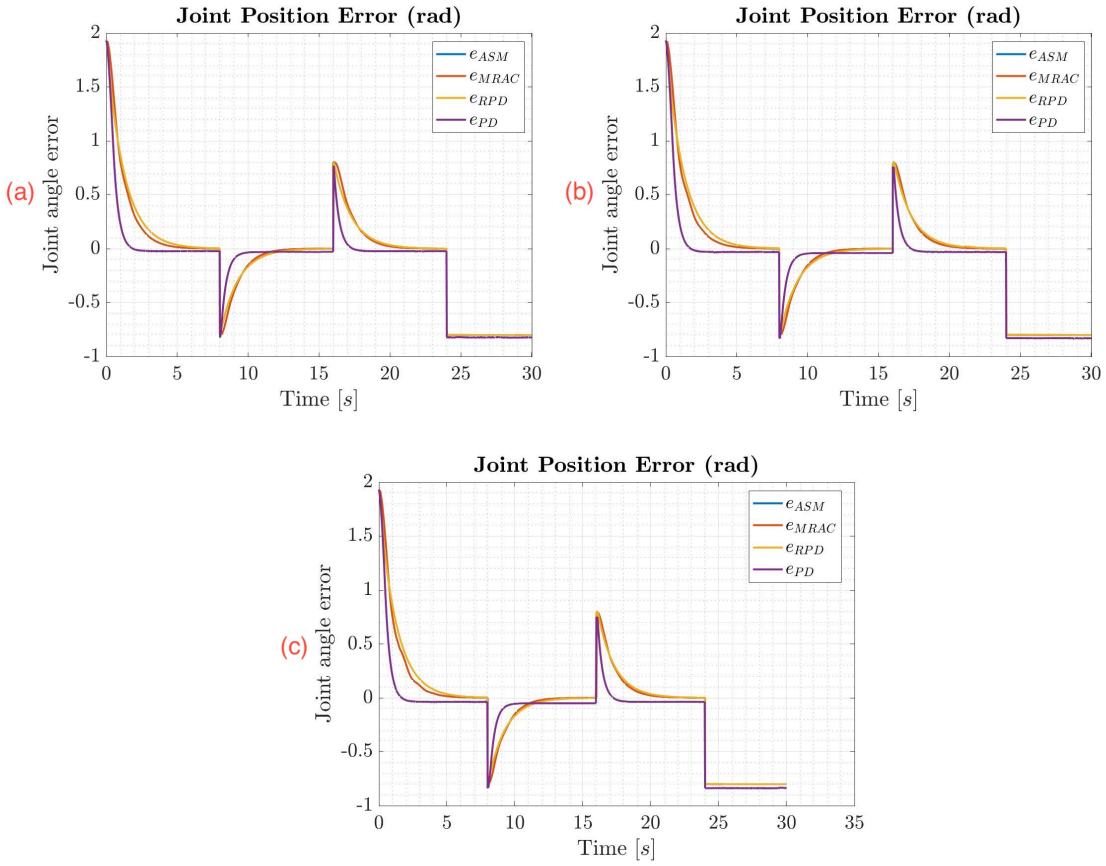


Figure 6.53: Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 1.

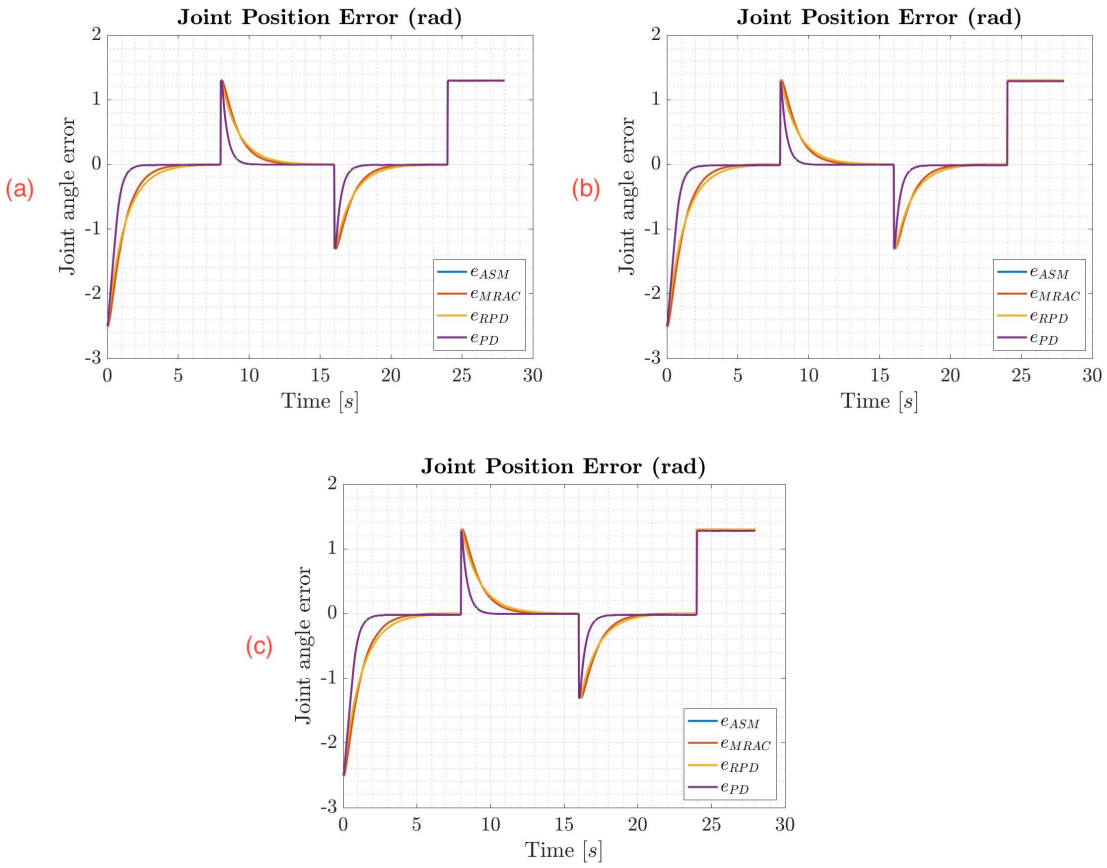


Figure 6.54: Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 1.

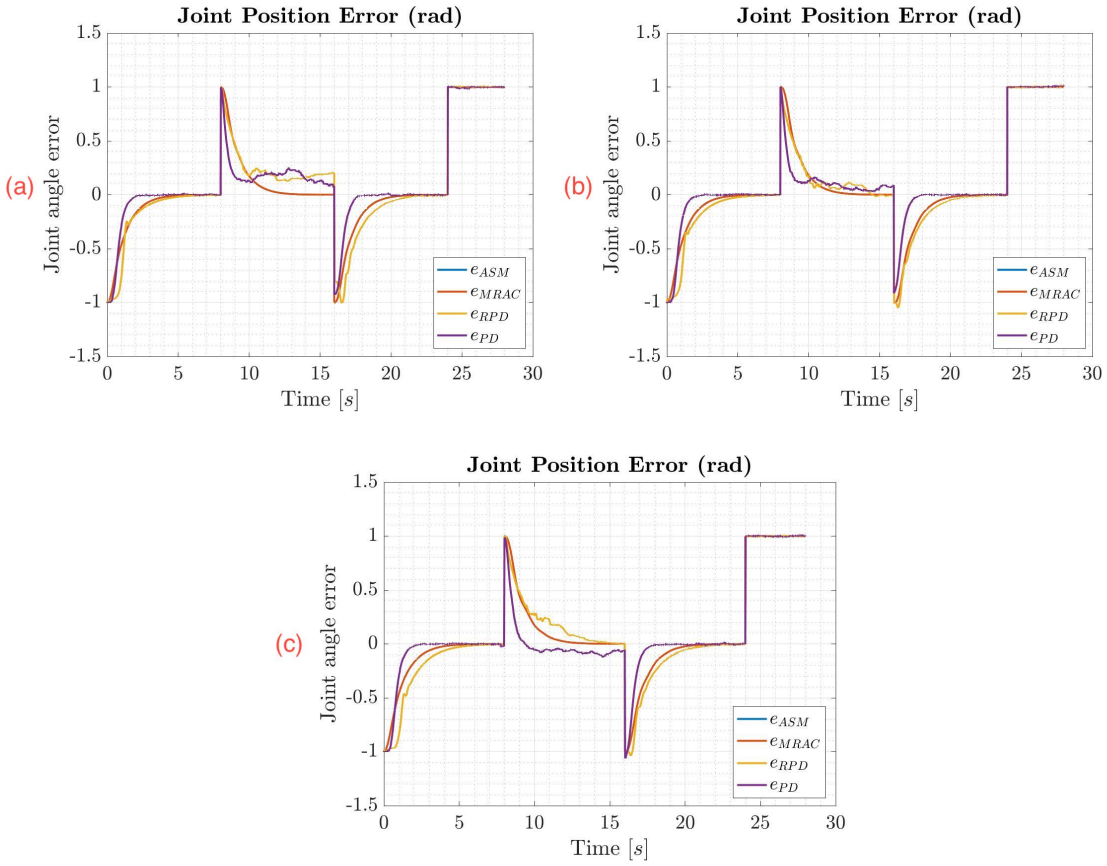


Figure 6.55: Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 2.

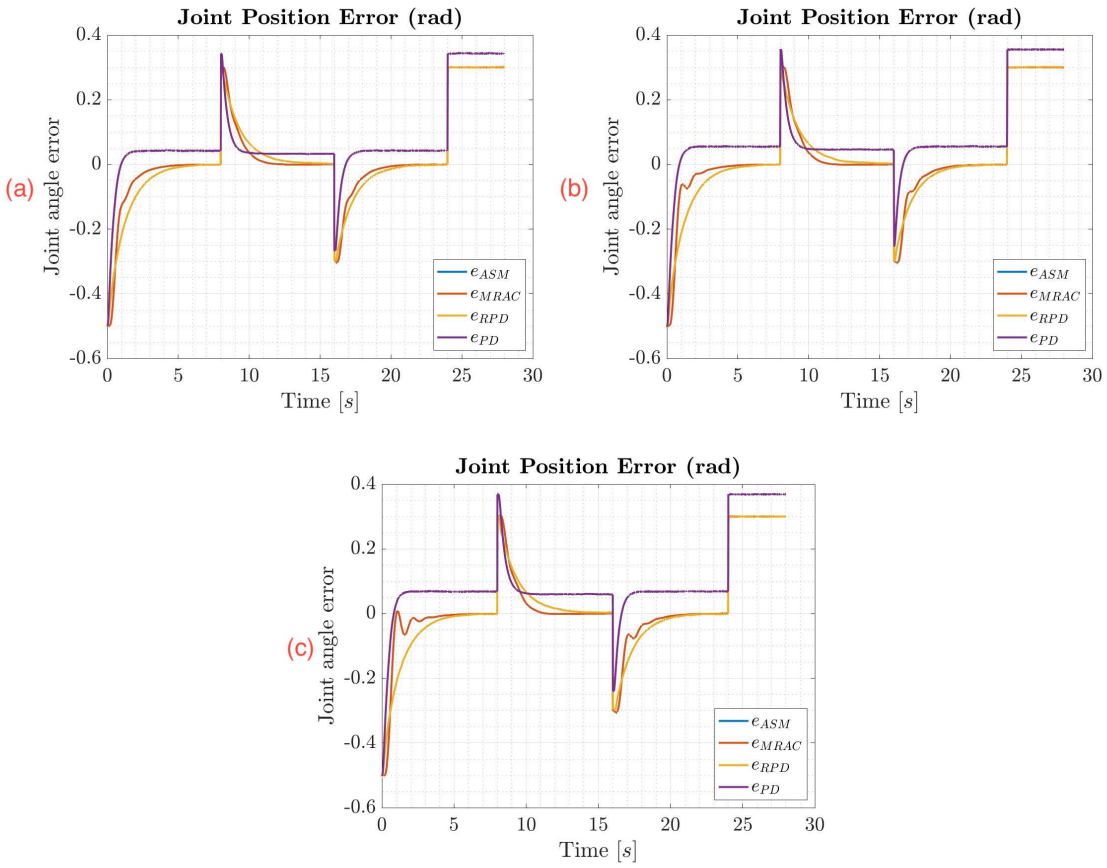


Figure 6.56: Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 2.

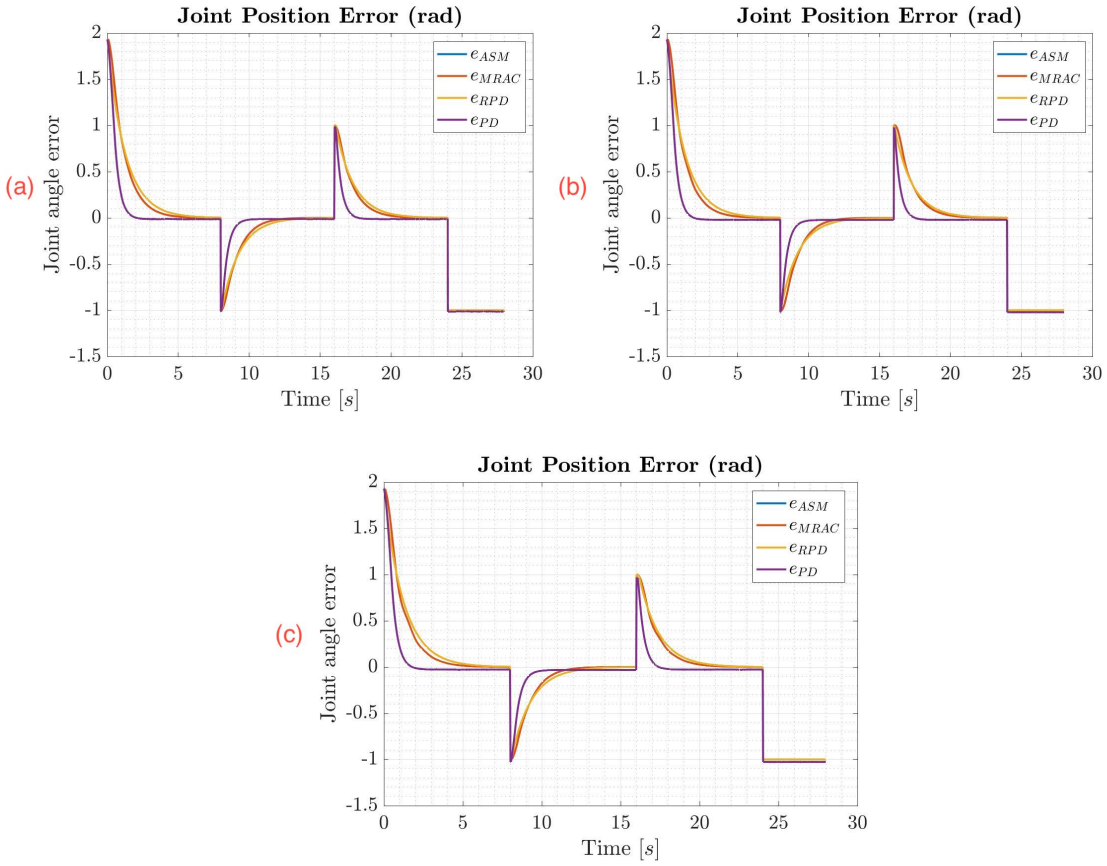


Figure 6.57: Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 2.

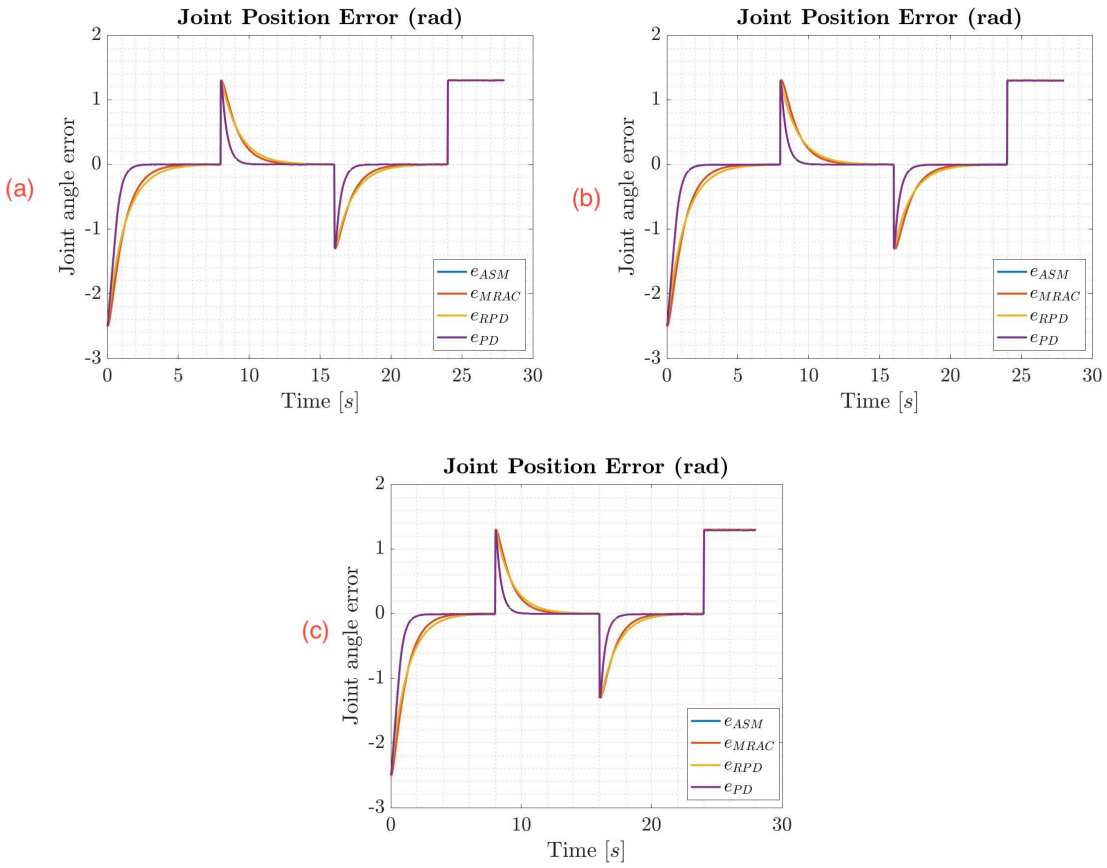


Figure 6.58: Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the sixth joint of Configuration 2.

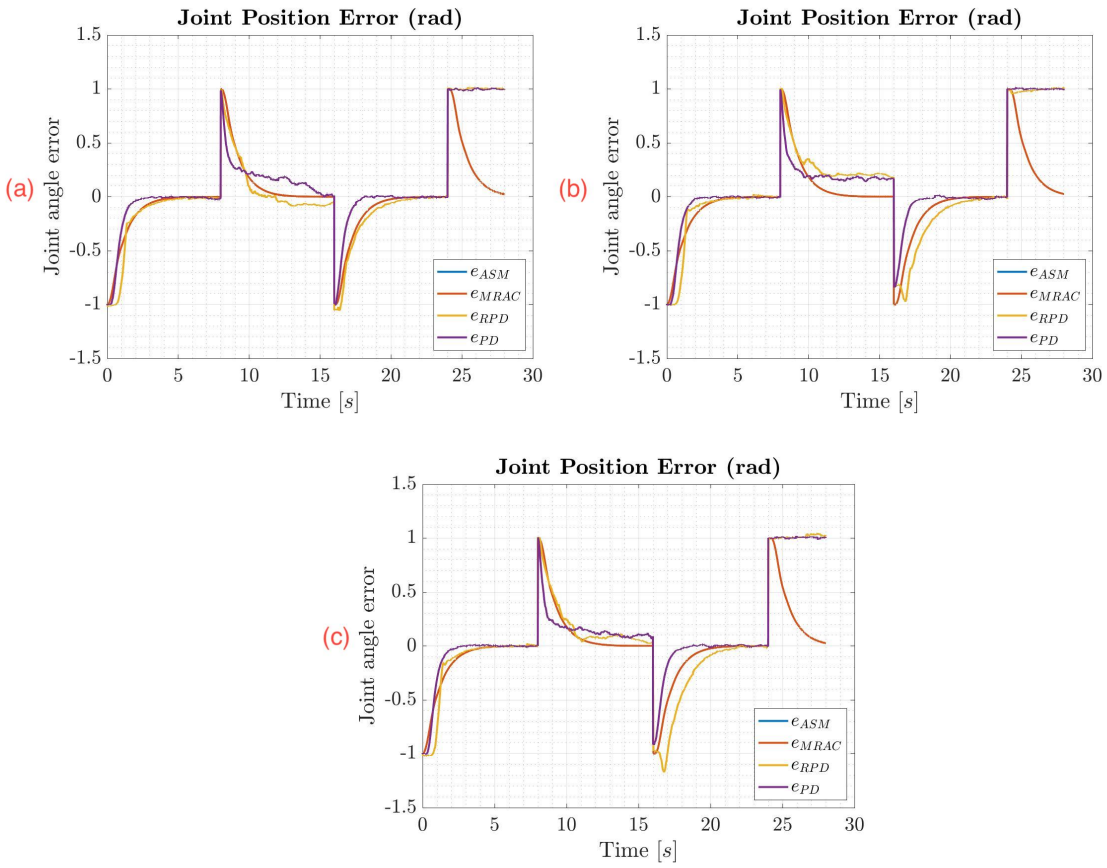


Figure 6.59: Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the first joint of Configuration 3.

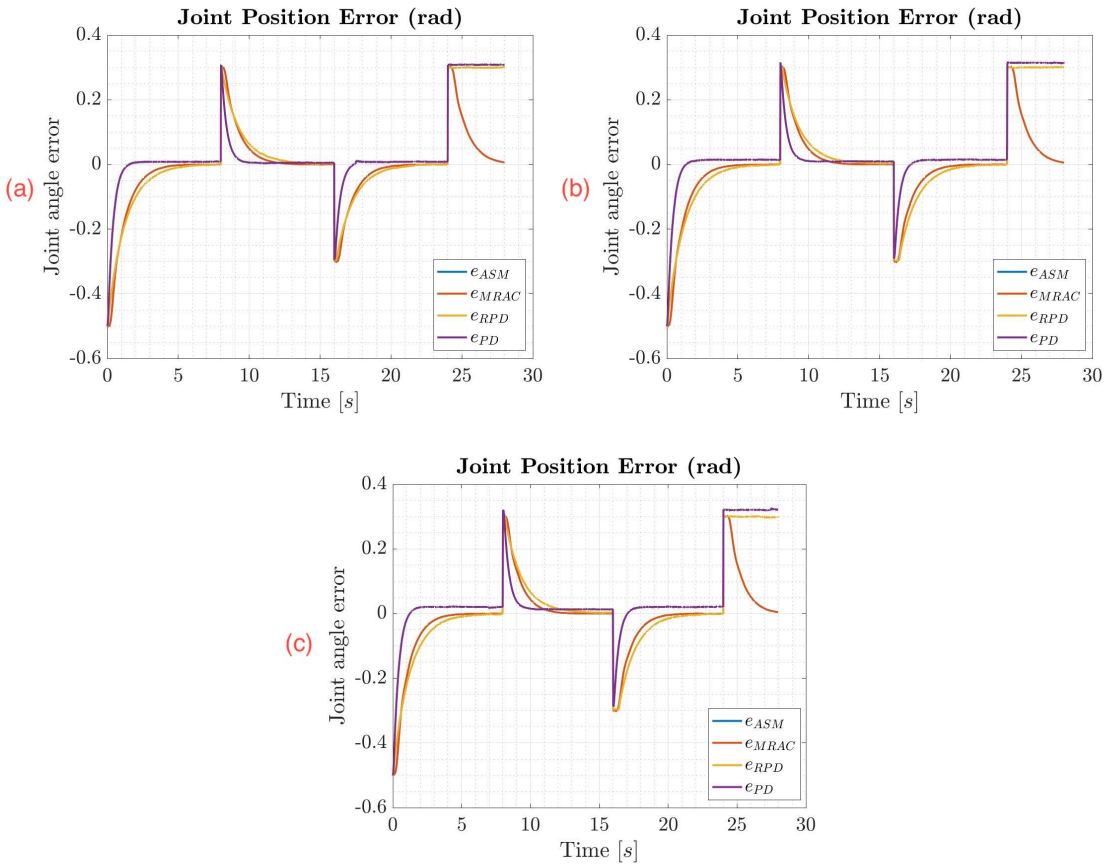


Figure 6.60: Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the second joint of Configuration 3.

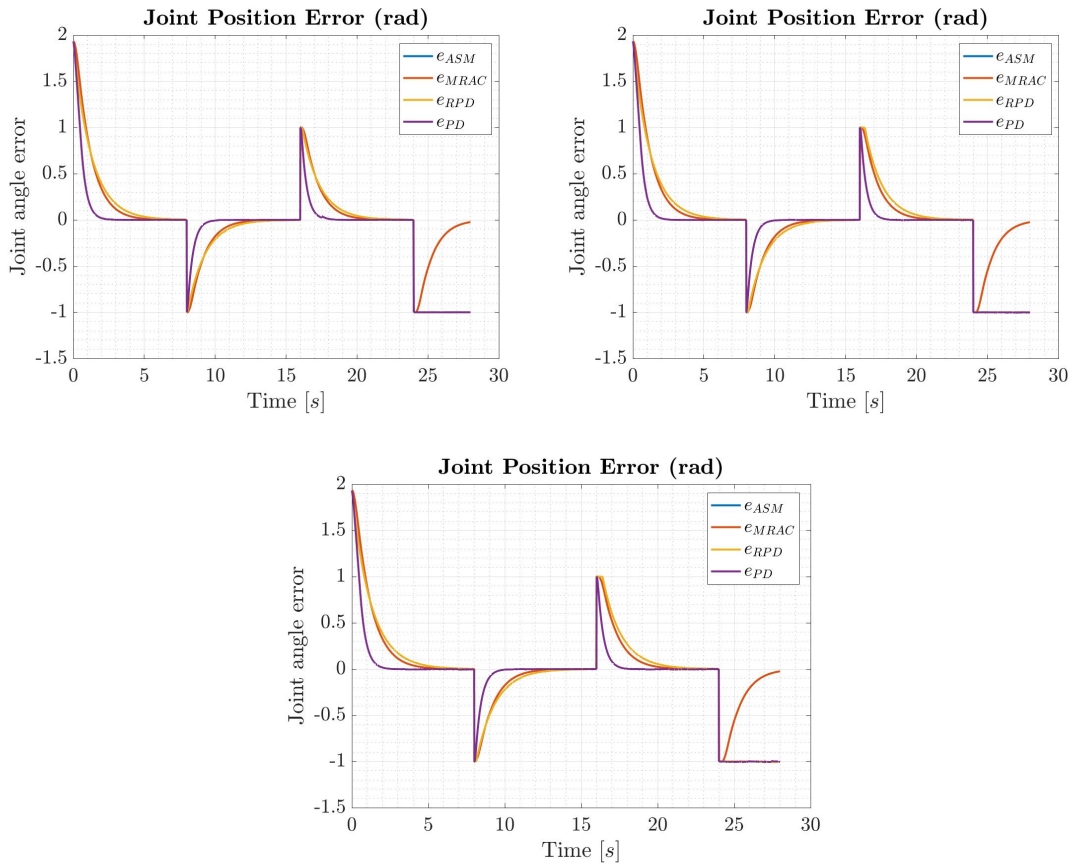


Figure 6.61: Position error of the third operation with (a) no load, (b) 1.5 KG, and (c) 3 KG for the fourth joint of Configuration 3.

6.3 Summary

From results obtained from three different robot configurations, we can say that even as small as a 1.5 Kg payload can significantly decrease the tracking performance of the robot arm. Moreover, due to the varying payload, when the number of d.o.f increases, gravity plays an important role in amplifying the disturbance effects. Besides, the proposed adaptive control structures can better handle the parametric uncertainties and disturbances compared with the PD and robust PD controllers. We also show that the adaptive control frameworks introduced in Chapters 4 and 5 to accomplish the desired positioning under different operating conditions.

Chapter 7

Conclusion and Future Directions

7.1 Conclusion

The proposed control architecture provides a systematic approach for designing decentralized adaptive controllers. This architecture allows the use of a two-layer control structure for the MRR system whose modules can be reassembled to reconfigure the configuration. The proposed two-layer architecture is effective to apply stable adaptive robust control designs when the modules are changed to configure different morphologies, a payload is added to the end-effector, and when the robot have uncertainties. A hierarchical adaptive control architecture has been proposed and the results show the applicability of the architecture for MRRs.

7.2 Future Directions

An extension of this research may include the utilization of the architecture proposed for a decentralized adaptive robust control design to the adaptive impedance control designs. Moreover, the difficulty of designing adaptive control schemes for possible redundancies and self collisions may be examined for future investigation. The proposed control architecture may be also applied for optimal adaptive control problems of MRR platforms.

References

[1] C. J. J. Paredis, H. B. Brown, and P. K. Khosla. A rapidly deployable manipulator system. In *Proc.of IEEE International Conference on Robotics and Automation*, volume 2, pages 1434–1439 vol.2, 1996.

[2] P. K. Khosla D. E. Schmitz and T. Kanade. The CMU reconfigurable modular manipulator system. In *Proc. IEEE International Symposium and Exposition on Robots*, pages 473–488, 1979.

[3] S. Tabandeh. *Development of Novel Task-Based Configuration Optimization Methodologies for Modular and Reconfigurable Robots Using Multi-Solution Inverse Kinematic Algorithms*. PhD thesis, University of Waterloo, December 2010.

[4] Traclabs mobile rmm. https://traclabs.com/projects/rmm, 2020.

[5] J. Kereluk and R. Emami. A new modular, autonomously reconfigurable manipulator platform. *International Journal of Advanced Robotic Systems*, 12, 06 2015.

[6] Kinova robotics, kinova jaco research edition. https://www.kinovarobotics.com/products/jaco-researchedition. Accessed: 2020-04-30.

[7] Kinova robotics, kinova movo robot. https://www.kinovarobotics.com/sites/default/files/OP-03_2019-05_R01.pdf. Accessed: 2020-04-30.

[8] Kinova robotics, kinova gen3 ulr. https://www.kinovarobotics.com/en/products/gen3-robot. Accessed: 2020-04-30.

[9] Mara cobot - the first modular robotic arm. https://acutronicrobotics.com/docs/products/robots/mara. Accessed: 2020-04-30.

[10] Schunk gmbh and co. kg. powerball lightweight arm lwa 4p. http://schunk-microsite.com/en/produkte/produkte/powerball-lightweight-arm-lwa-4p. Accessed: 2020-04-30.

[11] e.do robot. https://edo.cloud/edo-robot. Accessed: 2020-04-30.

[12] The dlr light weight robot iii (lwr iii). https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-12464/#gallery/29165. Accessed: 2020-04-30.

[13] The humanoid walking robot toro (torque-controlled humanoid robot). https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-11678/#gallery/28603. Accessed: 2020-04-30.

[14] M. Biglarbegian, W. W. Melek, and J. M. Mendel. Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: Theory and experiments. *IEEE Transactions on Industrial Electronics*, 58(4):1371–1384, 2011.

[15] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer Science and Business Media, 2012.

[16] L. Kelmar and P. K. Khosla. Automatic generation of kinematics for a reconfigurable modular manipulator system. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 663–668 vol.2, 1988.

[17] B. Benhabib, G. Zak, and M.G. Lipton. A generalized kinematic modeling method for modular robots. *Journal of Robotic Systems*, 6(5):545–571, 1989.

[18] A. Giusti and M. Althoff. Automatic centralized controller design for modular and reconfigurable robot manipulators. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3268–3275, 2015.

[19] A. Giusti and M. Althoff. On-the-fly control design of modular robot manipulators. *IEEE Transactions on Control Systems Technology*, 26(4):1484–1491, 2018.

[20] G. Gungor, B. Fidan, and W. Melek. Hierarchical decentralized robust control design for modular and reconfigurable robots. In *Proc. IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pages 1006–1011, 2018.

[21] Franka panda robot. https://frankaemika.github.io/docs/control_parameters, 2020.

[22] M. W. Spong, S. A. Hutchinson, and M. Vidyasagar. Robot modeling and control. *IEEE Control Systems*, 26(6):113–115, 2006.

[23] J. Luh. An anatomy of industrial robots and their controls. *IEEE Transactions on Automatic Control*, 28(2):133–153, 1983.

[24] H.A. ElMaraghy and B. Johns. An investigation into the compliance of scara robots. part i: Analytical model. *Journal of Dynamic Systems, Measurement, and Control*, 110(1):18–22, 1988.

[25] R. C. Beecher. Puma: Programmable universal machine for assembly. In *Computer Vision and Sensor-based Robots*, pages 141–152. Springer, 1979.

[26] G. Gungor, B. Fidan, and W. W. Melek. Reconfigurable robot manipulators: adaptation, control, and mems applications. In *Advanced Mechatronics and MEMS Devices II*, pages 169–194, 2017.

[27] S. Tabandeh, W. W. Melek, M. Biglarbegian, S. P. Won, and C. Clark. A memetic algorithm approach for solving the task-based configuration optimization problem in serial modular and reconfigurable robots. *Robotica*, 34(9):1979–2008, 2016.

[28] L. Bartevyan. Industry 4.0–summary report. *DLG-Expert report*, 5:1–8, 2015.

[29] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, and M. Harnisch. Industry 4.0: The future of productivity and growth in manufacturing industries. *Boston Consulting Group*, page 14, 2015.

[30] White House. Revitalizing american manufacturing. *National Economic Council*, 2016.

[31] I-M. Chen and M. Yim. Modular robots. In *Siciliano B., Khatib O. (eds) Springer Handbook of Robotics*, pages 531–542, 2016.

[32] J.D.G. Fernandez, H. Sprengel, M. Mallwitz, M. Zipper, B. Yu, and V. Bargsten. Designing modular series-elastic actuators for safe human-robot collaboration in industrial settings. In *Proc.of The 19th International Conference On Cooperative Robotics*, page 135. World Scientific, 2016.

[33] I. Maurtua, I. Fernández, A. Tellaeche, J. Kildal, L. Susperregi, A. Ibarguren, and B. Sierra. Natural multimodal communication for human-robot collaboration. *International Journal of Advanced Robotic Systems*, 14(4):1729881417716043, 2017.

[34] K. H. Wurst. The conception and construction of a modular robot system. In *Proc. of International Conference on Industrial Robots*, pages 37–44. IFS, 1986.

[35] T. Matsumaru. Design and control of the modular robot system: TOMMS. In *Proc. of IEEE International Conference on Robotics and Automation*, volume 2, pages 2125–2131, 1995.

[36] R. Hui, N. Kircanski, A. Goldenberg, C. Zhou, P. Kuzan, J. Wiercienski, D. Gershon, and P. Sinha. Design of the iris facility-a modular, reconfigurable and expandable robot test bed. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 155–160 vol.3, 1993.

[37] G. Liu, Y. Liu, and A. A. Goldenberg. Design, analysis, and control of a spring-assisted modular and reconfigurable robot. *IEEE/ASME Transactions on Mechatronics*, 16(4):695–706, 2011.

[38] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schäffer. Overview of the torque-controlled humanoid robot toro. In *Proc. of IEEE-RAS International Conference on Humanoid Robots*, pages 916–923, 2014.

[39] N. Lopez, Y. Nuin, E. Moral, L. Juan, A. Solano, V. Vilches, and R. Kojcev. gym-gazebo2, a toolkit for reinforcement learning using ros 2 and gazebo, 03 2019.

[40] M. G. Catalano, G. Grioli, M. Garabini, F. Bonomo, M. Mancini, N. Tsagarakis, and A. Bicchi. VSA-CubeBot: A modular variable stiffness platform for multiple

degrees of freedom robots. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 5090–5095, 2011.

[41] S. Tabandeh, W. W. Melek, and M. C. Clark. An adaptive niching genetic algorithm approach for generating multiple solutions of serial manipulator inverse kinematics with applications to modular robots. *Robotica*, 28(4):493–507, 2010.

[42] C. J. J. Paredis and P. K. Khosla. Kinematic design of serial link manipulators from task specifications. *The International Journal of Robotics Research*, 12(3):274–287, 1993.

[43] I-M. Chen and J. W. Burdick. Determining task optimal modular robot assembly configurations. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 132–137 vol.1, 1995.

[44] E. Icer, A. Giusti, and M. Althoff. A task-driven algorithm for configuration synthesis of modular robots. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 5203–5209, 2016.

[45] W. W. Melek and A. A. Goldenberg. Neurofuzzy control of modular and reconfigurable robots. *IEEE/ASME Transactions on Mechatronics*, 8(3):381–389, 2003.

[46] X. Pan, H. Wang, Y. Jiang, and J. Xiao. Automatic kinematic modelling of a modular reconfigurable robot. *Transactions of the Institute of Measurement and Control*, 35(7):922–932, 2013.

[47] Y. Tang, M. Tomizuka, G. Guerrero, and G. Montemayor. Decentralized robust control of mechanical systems. *IEEE Transactions on Automatic Control*, 45(4):771–776, April 2000.

[48] G. Liu. Decomposition-based friction compensation of mechanical systems. *Mechatronics*, 12(5):755 – 769, 2002.

[49] G. Liu, S. Abdul, and A. A. Goldenberg. Distributed modular and reconfigurable robot control with torque sensing. In *Proc. of IEEE International Conference on Mechatronics and Automation*, pages 384–389, 2006.

[50] M. Zhu and Y. Li. Decentralized adaptive fuzzy control for reconfigurable manipulators. In *Proc. of IEEE International Conference on Robotics, Automation and Mechatronics*, pages 404–409, Sep. 2008.

[51] Z. Li, W. W. Melek, and C. Clark. Decentralized robust control of robot manipulators with harmonic drive transmission and application to modular and reconfigurable serial arms. *Robotica*, 27(2):291–302, 2009.

[52] A. Giusti. *Automatic Design of Controllers for Modular Reconfigurable Robot Manipulators*. PhD thesis, Technical University of Munich, 2018.

[53] R.H. Middletone and G.C. Goodwin. Adaptive computed torque control for rigid link manipulators. In *Proc. of IEEE International Conference on Decision and Control*, pages 68–73, 1986.

[54] P. Hsu, M. Bodson, S. Shankar, and B. Paden. Adaptive identification and control for manipulators without using joint accelerations. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 1210–1215. IEEE, 1987.

[55] P. Ioannou and B. Fidan. *Adaptive control tutorial*. SIAM, 2006.

[56] P. V. Kokotovic. The joy of feedback: nonlinear and adaptive. *IEEE Control Systems Magazine*, 12(3):7–17, 1992.

[57] S. Marx, Y. Chitour, and C. Prieur. Stability analysis of dissipative systems subject to nonlinear damping via lyapunov techniques. *IEEE Transactions on Automatic Control*, pages 1–1, 2019.

[58] W. E. Dixon, M. S. de Queiroz, D. M. Dawson, and T. J. Flynn. Adaptive tracking and regulation of a wheeled mobile robot with controller/update law modularity. *IEEE Transactions on Control Systems Technology*, 12(1):138–147, 2004.

[59] A. Loría, E. Panteley, and M. Maghenem. Strict lyapunov functions for model reference adaptive control: Application to lagrangian systems. *IEEE Transactions on Automatic Control*, 64(7):3040–3045, 2019.

[60] A. Loria, E. Panteley, D. Popovic, and A. R. Teel. /spl delta/-persistency of excitation: a necessary and sufficient condition for uniform attractivity. In *Proc. of the 41st IEEE Conference on Decision and Control, 2002.*, volume 3, pages 3506–3511 vol.3, 2002.

[61] F. Mazenc, M. de Queiroz, and M. Malisoff. Uniform global asymptotic stability of a class of adaptively controlled nonlinear systems. *IEEE Transactions on Automatic Control*, 54(5):1152–1158, 2009.

[62] F. Mazenc, M. Malisoff, and O. Bernard. A simplified design for strict lyapunov functions under matrosov conditions. *IEEE Transactions on Automatic Control*, 54(1):177–183, 2009.

[63] F. Ghorbel, B. Srinivasan, and M. W. Spong. On the uniform boundedness of the inertia matrix of serial robot manipulators. *Journal of Robotic Systems*, 15(1):17–28, 1998.

[64] G. Gungor, B. Fidan, and W. W. Melek. Decentralized model reference adaptive control design for modular and reconfigurable robots. In *Advances in Motion Sensing and Control for Robotic Applications*, pages 109–125. Springer, 2019.

[65] M. Gautier and W. Khalil. Direct calculation of minimum set of inertial parameters of serial robots. *IEEE Transactions on Robotics and Automation*, 6(3):368–373, 1990.

[66] L. Shi and S. K. Singh. Decentralized adaptive controller design for large-scale systems with higher order interconnections. *IEEE Transactions on Automatic Control*, 37(8):1106–1118, 1992.

[67] M. Gautier and W. Khalil. On the identification of the inertial parameters of robots. In *Proc. of the 27th IEEE Conference on Decision and Control*, pages 2264–2269 vol.3, 1988.

[68] V. Mata, F. Benimeli, N. Farhat, and A. Valera. Dynamic parameter identification in industrial robots considering physical feasibility. *Advanced Robotics*, 19(1):101–119, 2005.

[69] V. Bargsten, P. Zometa, and R. Findeisen. Modeling, parameter identification and model-based control of a lightweight robotic manipulator. In *2013 IEEE International Conference on Control Applications (CCA)*, pages 134–139, 2013.

[70] W. Khalil and E. Dombre. *Modeling, identification and control of robots*. Butterworth-Heinemann, 2004.

[71] M. Krstic and P. V. Kokotovic. Adaptive nonlinear design with controller-identifier separation and swapping. *IEEE Transactions on Automatic Control*, 40(3):426–440, 1995.

[72] M. S. de Queiroz , D. M. Dawson, and M. Agarwal. Adaptive control of robot manipulators with controller/update law modularity. *Automatica*, 35(8):1379–1390, 1999.

[73] E. Panteley, A. Loria, and A. Teel. Relaxed persistency of excitation for uniform asymptotic stability. *IEEE Transactions on Automatic Control*, 46(12):1874–1886, 2001.

[74] P. A. Ioannou and P. V. Kokotovic. Instability analysis and improvement of robustness of adaptive control. *Automatica*, 20(5):583–594, 1984.

[75] C. G. Atkeson, C. H. An, and J. M. Hollerbach. Estimation of inertial parameters of manipulator loads and links. *The International Journal of Robotics Research*, 5(3):101–119, 1986.

[76] E. D. Sontag and Y. Wang. On characterizations of the input-to-state stability property. *Systems and Control Letters*, 24(5):351–359, 1995.

[77] M. Neubauer, H. Gattringer, and H. Bremer. A persistent method for parameter identification of a seven-axes manipulator. *Robotica*, 33(5):1099–1112, 2015.

[78] S-H. Hsu and L-C. Fu. A fully adaptive decentralized control of robot manipulators. *Automatica*, 42(10):1761–1767, October 2006.

[79] S-H. Hsu and L-C. Fu. Globally adaptive decentralized control of time-varying robot manipulators. In *Proc. of IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 1, pages 1458–1463 vol.1, 2003.

[80] G-J. Liu and A. A. Goldenberg. On robust saturation control of robot manipulators. In *Proc. of 32nd IEEE Conference on Decision and Control*, pages 2115–2120 vol.3, 1993.

[81] J.J.E. Slotine and W. Li. On the adaptive control of robot manipulators. *The international Journal of Robotics Research*, 6(3):49–59, 1987.

[82] J.J.E. Slotine and W. Li. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, 1991.

[83] J. J. Craig. *Introduction to robotics: mechanics and control*. Pearson Education, 2009.

[84] P. A. Ioannou and J. Sun. *Robust adaptive control*. Courier Corporation, 2012.

[85] I-M. Chen and G. Yang. Automatic model generation for modular reconfigurable robot dynamics. *Journal of Dynamic Systems, Measurement, and Control*, 120(3):346–352, 1998.

[86] H. Liu, H. Wang, S. Li, and L. He. Research on the topology description and modeling method for reconfigurable modular robots. In *Proc. of ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots*, pages 129–135, 2009.

[87] Y-Q. Fei , G-T. Fen, X-F. Zhao, and W-L. Xu. Automatic generation of the reconfigurable robot forward and inverse kinematics. *Journal-Shanghai Jiaotong University-Chinese Edition*, 34(10):1430–1433, 2000.

[88] J. Baca, A. Yerpes, M. Ferre, J. A. Escalera, and R. Aracil. Modelling of modular robot configurations using graph theory. In *International Workshop on Hybrid Artificial Intelligence Systems*, pages 649–656. Springer, 2008.

[89] C. Pozna. The modular robots kinematics. *Acta polytechnica hungarica*, 4(2):5–18, 2007.

[90] E. Yoshida, S. Murata, A. Kamimura, K. Tomita, H. Kurokawa, and S. Kokaji. Reconfiguration planning for a self-assembling modular robot. In *Proc. of of IEEE International Symposium on Assembly and Task Planning (ISATP). Assembly and Disassembly in the Twenty-first Century. (Cat. No.01TH8560)*, pages 276–281, 2001.

[91] F. Hou and W-M. Shen. Mathematical foundation for hormone-inspired control for self-reconfigurable robotic systems. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 1477–1482, 2006.

[92] M. Park, S. Chitta, A. Teichman, and M. Yim. Automatic configuration recognition methods in modular robots. *The International Journal of Robotics Research*, 27(3-4):403–421, 2008.

[93] D. Kohli and A.H. Soni. Kinematic analysis of spatial mechanisms via successive screw displacements. *Journal of Engineering for Industry*, 97(2):739–747, 1975.

[94] R. P. Paul. *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Richard Paul, 1981.

[95] G.R. Pennock and A.T. Yang,. Application of dual-number matrices to the inverse kinematics problem of robot manipulators. *Journal of Mechanisms, Transmissions, and Automation in Design*, 107(2):201–208, 1985.

[96] An-Tzu Yang and F. Freudenstein. Application of dual-number quaternion algebra to the analysis of spatial mechanisms. *Journal of Applied Mechanics*, 31(2):300–308, 1964.

[97] J.J. Uicker, J. Denavit, and R.S. Hartenberg. An iterative method for the displacement analysis of spatial mechanisms. *Journal of Applied Mechanics*, 31(2):309–314, 1964.

[98] G. Legnani, F. Casolo, P. Righettini, and B. Zappa. A homogeneous matrix approach to 3d kinematics and dynamics—i. theory. *Mechanism and Machine Theory*, 31(5):573–587, 1996.

[99] G. Legnani, F. Casolo, P. Righettini, and B. Zappa. A homogeneous matrix approach to 3d kinematics and dynamics—ii. applications to chains of rigid bodies and serial manipulators. *Mechanism and Machine Theory*, 31(5):589–605, 1996.

[100] L-W Tsai and A. P. Morgan. Solving the kinematics of the most general six-and five-degree-of-freedom manipulators by continuation methods. *Journal of Mechanisms, Transmissions, and Automation in Design*, 107(2):189–200, 1985.

[101] I-Ming Chen. Rapid response manufacturing through a rapidly reconfigurable robotic workcell. *Robotics and Computer-Integrated Manufacturing*, 17(3):199–213, 2001.

[102] A. Goldenberg, B. Benhabib, and R. Fenton. A complete generalized solution to the inverse kinematics of robots. *IEEE Journal on Robotics and Automation*, 1(1):14–20, 1985.

[103] I-Ming Chen and Guilin Yang. Inverse kinematics for modular reconfigurable robots. In *Proc. of IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 2, pages 1647–1652, 1998.

[104] C. Baur and D. Wee. Manufacturing's next act. *McKinsey Quarterly, Jun*, 2015.

[105] D. W. Havas. Facing manufacturing challenges. *The Bent of Tau Beta PI, New York Nu*, 2009.

[106] K. J. Åström. Theory and applications of adaptive control—a survey. *Automatica*, 19(5):471–486, 1983.

[107] F. L. Lewis, D.M. Dawson, and C. T. Abdallah. Control of robot manipulators. 1993.

[108] H. K. Khalil. *Nonlinear systems*. Upper Saddle River, 2012.

[109] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[110] W. E. Dixon, M.S. De Queiroz, D. M. Dawson, and T. J. Flynn. Adaptive tracking and regulation of a wheeled mobile robot with controller/update law modularity. *IEEE Transactions on Control Systems Technology*, 12(1):138–147, 2004.

[111] W. W. Melek and H. Najjaran. Study of the effect of external disturbances on the position control of iris modular and reconfigurable manipulator. In *Proc. of IEEE International Conference Mechatronics and Automation*, volume 1, pages 144–147, 2005.

[112] P. Tomei. Adaptive pd controller for robot manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):565–570, 1991.

[113] G. Yang and I-M. Chen. Task-based optimization of modular robot configurations: minimized degree-of-freedom approach. *Mechanism and Machine Theory*, 35(4):517–540, 2000.

[114] E. Icer and M. Althoff. Cost-optimal composition synthesis for modular robots. In *Proc. of IEEE International Conference on Control Applications*, pages 1408–1413, 2016.

[115] M. Zhu and Y. Li. Decentralized adaptive fuzzy control for reconfigurable manipulators. In *Proc. of IEEE International Conference on Robotics, Automation and Mechatronics*, pages 404–409, 2008.

[116] A. Castano and P. Will. Representing and discovering the configuration of conro robots. In *Proc. of ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 4, pages 3503–3509. IEEE, 2001.

[117] Mark Yim, Kimon Roufas, David Duff, Ying Zhang, Craig Eldershaw, and Sam Homans. Modular reconfigurable robots in space applications. *Autonomous Robots*, 14(2-3):225–237, 2003.

[118] J. Baur, J. Pfaff, H. Ulbrich, and T. Villgrattner. Design and development of a redundant modular multipurpose agricultural manipulator. In *Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 823–830, 2012.

[119] M. Biglarbegian, W. W. Melek, and J. M. Mendel. Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: theory and experiments. *IEEE transactions on industrial electronics*, 58(4):1371–1384, 2011.