

Design Synthesis & Prototype Implementation of Parallel Orientation Manipulators for Optomechatronic Applications

by

© Taufiqur Rahman

A thesis submitted to the
School of Graduate Studies
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Faculty of Engineering & Applied Science
Memorial University of Newfoundland

January 2016

St. John's

Newfoundland

জোহরা বেগম এবং আবদুর রহমান, আমার মা-বাবা

ছোটখাট গড়নের মানুষ তাঁরা, তবু কী প্রকান্ড ছায়ায় আমাদের আগলে রেখেছেন সবসময়!

মিলা, আমার বাড়ি ফেরার ঠিকানা

তুমি পাশে ছিলে বলেই ফেলে আসা পথের ক্লান্তি আমায় কখনো গ্রাস করতে পারেনি।

Preface

The work described in this dissertation has been commissioned by the Intelligent Sensor Platforms for Remotely Piloted Vehicles (INSPIRUS) project at the Memorial university of Newfoundland. Correspondingly, the thesis author has collaborated with the project team during its development. Parts of the research presented here have appeared as multi-author publications in peer-reviewed journals and conferences. However, the principal research ideas of these publications were solely developed by the thesis author. The plurality in the author list reflect collaborative contributions in the form of expert critique, assisting in experiments, and prototyping designs synthesized by the thesis author. The copyright owners of these publications, namely the Institute of Electrical and Electronics Engineers (IEEE) and the American Society of Mechanical Engineers (ASME), have granted their permissions to reuse them in this thesis. The textual and the graphical contents of these publications were exclusively composed by the author of this dissertation.

The designs ideas involving the electromechanical system described in Chapter 3 and the spherical joint documented in Section 7.2 were conceived by Dr. Nicholas Krouglicof, the principal investigator of the INSPIRUS project. The thesis author along with the project team have contributed towards the development of these designs into functional prototypes. With the aforementioned exceptions, the engineering designs, the formulations, and the analyses presented here are claimed to be the intellectual product of the thesis author.

Abstract

This thesis documents a research endeavor undertaken to develop high-performing designs for parallel orientation manipulators (POM) capable of delivering the speed and the accuracy requirements of a typical optomechatronic application. In the course of the research, the state of the art was reviewed, and the areas in the existing design methodologies that can be potentially improved were identified, which included actuator design, dimensional synthesis of POMs, control system design, and kinematic calibration. The gaps in the current art of designing each of these POM system components were addressed individually. The outcomes of the corresponding development activities include a novel design of a highly integrated voice coil actuator (VCA) possessing the speed, the size, and the accuracy requirements of small-scale parallel robotics. Furthermore, a method for synthesizing the geometric dimensions of a POM was developed by adopting response surface methodology (RSM) as the optimization tool. It was also experimentally shown how conveniently RSM can be utilized to develop an empirical quantification of the *actual* kinematic structure of a POM prototype. In addition, a motion controller was formulated by adopting the active disturbance rejection control (ADRC) technology. The classic formulation of the ADRC algorithm was modified to develop a resource-optimized implementation on control hardware based on field programmable gate arrays (FPGA).

The practicality and the effectiveness of the synthesized designs were ultimately demonstrated by performance benchmarking experiments conducted on POM proto-

types constructed from these components. In specific terms, it was experimentally shown that the moving platforms of the prototyped manipulators can achieve high-speed motions that can exceed 2000 degrees/s in angular velocity, and 5×10^5 degrees/s² in angular acceleration.

Acknowledgments

I would like to express my deepest gratitude to my research supervisor, Dr. Nicholas Krouglicof, an engineering genius and an even better man. Without his versatile expertise, continual support, and witty leadership, this work would have not been completed. His relentless enthusiasm for research was fiercely contagious, and it delightfully inspired excellence in everyone around him. It has been an absolute privilege to have him as my mentor. I am eternally thankful for all he has done for me throughout my graduate studies. Simply put, a doctoral candidate cannot possibly ask for a better thesis supervisor.

I would like to thank Dr. Geoff Rideout and Dr. Leonard Lye for their invaluable advice and technical help. Their expertise has been instrumental in developing the key ideas of this thesis. I also thank Dr. Andrew Vardy and Dr. James Yang for their service in the committee for the doctoral comprehensive examinations.

I must acknowledge the technical assistance I received from the brilliant technologists at the Faculty of Engineering & Applied Science, Memorial University of Newfoundland. Regardless of how difficult or mundane my requests were, Messrs. Tom Pike, Steve Steele, Brian Pretty, and Don Taylor have always been welcoming and friendly. Their skillful services were absolutely crucial in implementing the design ideas of this thesis. I sincerely thank them for their help.

I am forever indebted to Dr. Kaaren May and Mr. Stephen Reddin, past managers of the INSPIRUS project. I will always fondly remember how awesomely nice Dr. May

was to me whenever I needed help with technical writing. I am extremely thankful to Mr. Reddin for bearing the nightmares of procurement and logistics in my stead, which enabled me to concentrate on the research, and remain carefree about the real world.

The generous support offered by the amazing people of the Center for Sustainable Aquatic Resources (CSAR) at Marine institute, Newfoundland is greatly appreciated. Particularly, Dr. Paul Winger, Ms. Kelly Moret, Ms. Caludene Hartley, and Mr. George Legge extended their helping hands in times of need. I express my sincere gratitude to them.

I would like to take this opportunity to gratefully acknowledge the financial support contributed by Memorial University of Newfoundland, Atlantic Canada Opportunities Agency (ACOA), Research and Development Corporation of Newfoundland (RDC), and the Boeing Company.

I thank my fellow graduate students Messrs. Raju Hossain, Migara Liyanage, Dennis Fifield, Dion Hicks, Michael Morgan, and Teng Wang for their friendship. The debates and the conversations that we shared have greatly enriched my learning, and I thank all of them for that.

I express the sincerest appreciation for my mother, Mrs. Johora Begum, and for my father, Mr. Abdur Rahman. Sacrifice is an inadequate word to convey how they always have put my future and success ahead of their own dreams and comfort. Without their love and support, I would have never been in a position to undertake this doctoral research.

Finally, I express my heartfelt gratitude to my lovely wife, Mrs. Mobash Akhter. Her strength and patience have driven me forward in this journey. The thesis would have never come to existence without her support.

Contents

Preface	i
Abstract	ii
Acknowledgments	iv
Contents	vi
List of Tables	xii
List of Figures	xiv
1 Introduction	1
1.1 Motivation	3
1.2 Statement of Co-Authorship	5
1.3 Contributions of the Thesis	6
1.4 Organization of the Thesis	8
2 Literature Review	10
2.1 Introduction	10
2.2 Kinematic Architectures of Parallel Orientation Manipulators	13
2.3 Dimensional Synthesis of Parallel Manipulators	17
2.4 Motion Control of Parallel Manipulators	21

2.5	Conclusion	24
3	Design & Prototyping of a Novel Voice Coil Actuator	26
3.1	Introduction	27
3.2	Proposed Design of a Voice Coil Actuator	28
3.2.1	Magnetics & PCB Coil Design	29
3.2.2	Position Feedback Mechanism	33
3.2.3	Electronic Drive Circuit	39
3.3	Specifications & Open-loop Performance	40
3.4	Conclusion	41
4	Dimensional Synthesis of Parallel Orientation Manipulators	42
4.1	Introduction	43
4.2	Kinematic Structures of the Candidate POM Architectures	45
4.3	Kinematic Analysis of 3- <u>P</u> SS/S Manipulators	47
4.3.1	Inverse Kinematics	48
4.3.2	Direct Kinematics	49
4.3.3	Inverse Kinematic Jacobian	53
4.3.4	Kinematic Singularities	55
4.4	Kinematic Analysis of 3- <u>S</u> PS/S Manipulators	56
4.4.1	Inverse Kinematics	56
4.4.2	Direct Kinematics	57
4.4.3	Differential Kinematics	58
4.5	Representation of Manipulator Geometry	59
4.5.1	Geometric Parameterization of 3- <u>P</u> SS/S Manipulators	59
4.5.2	Geometric Parameterization of 3- <u>S</u> PS/S Manipulators	61
4.6	Kinematic Performance Characteristics	62

4.6.1	Parameterization of an Orientation Workspace	62
4.6.2	Specification of Dexterity of an Orientation Workspace	65
4.6.3	Objective Function	65
4.7	Quantification of Kinematic Performance	66
4.7.1	Numerical Characterization of an Orientation Workspace	66
4.7.2	Estimating the Dexterity of an Orientation Workspace	68
4.8	Optimization Procedure	70
4.8.1	3-PSS/S Manipulator (Parallel Configuration)	73
4.8.1.1	Reduction of the Search Space	73
4.8.1.2	Data Analysis	74
4.8.1.3	Optimum Solution	76
4.8.2	3-PSS/S Manipulator (Slanted Configuration)	77
4.8.3	3-SPS/S Manipulator	77
4.9	Optimal Kinematic Performance	78
4.10	Discussion	82
4.11	Conclusion	86
5	Simulation & Control of Parallel Orientation Manipulators	88
5.1	Introduction	89
5.2	Model Construction	92
5.2.1	Eulerian Junction Structure & Articulation Points	93
5.2.2	Coordinate Transformation	95
5.2.3	Modeling of the Kinematic Joints	96
5.2.4	Complete Model	99
5.3	Active Disturbance Rejection Control	100
5.3.1	Transient Profile	103

5.3.2	Nonlinear PD Controller	105
5.3.3	Tuning of ADRC	106
5.4	Dynamic Performance Evaluation	108
5.4.1	Transient Motion Profile Generation	110
5.4.2	Performance Indices	112
5.4.3	Model Verification	113
5.4.4	Experimental Observations	114
5.5	Discussion	116
5.6	Conclusion	117
6	FPGA Implementation of ADRC Technology	118
6.1	Introduction	119
6.2	Resource Optimized Formulation of the ADRC Algorithm	122
6.3	FPGA Implementation of the ADRC Algorithm	125
6.3.1	Transient Profile Generator	127
6.3.2	ESO & Nonlinear Control Law	129
6.3.3	Discrete ADRC Tuning	131
6.3.4	Resource Cost Optimization	131
6.4	Experimental Evaluation of the Resource Optimized ADRC	132
6.4.1	Resource Cost of Implementation	132
6.4.2	Transient Response Performance	133
6.4.3	Frequency Response Performance	135
6.4.4	Experimental Results	137
6.5	Resource Optimized ADRC for a Parallel Orientation Manipulator . .	138
6.5.1	Performance Evaluation	140
6.6	Discussion	142

6.7	Conclusion	143
7	Prototype Implementation & Performance Characterization	144
7.1	Introduction	145
7.2	Design of an Anti-Backlash Spherical Joint	147
7.2.1	Prototype Development	149
7.2.2	Experimental Evaluation	151
7.2.2.1	Static Performance Characterization	151
7.2.2.2	Dynamic Motion Transmission Characteristics	154
7.3	Mechanical Design of POM Prototypes	156
7.3.1	Post Optimality Analysis	157
7.4	Dynamic Performance Characterization	162
7.4.1	Transient Response Performance	163
7.4.2	Demonstration of High Speed Angular Motion	168
7.5	Kinematic Calibration	174
7.5.1	Parameterization of Empirical Kinematic Models	176
7.5.2	Design of Experiment	177
7.5.3	Data Acquisition & Model Construction	179
7.5.4	Evaluation of Model Accuracy	181
7.6	Design of a Torsion-Restricted POM	184
7.6.1	Kinematic Analysis	185
7.6.2	Differential Kinematics	187
7.6.3	Inverse Kinematic Singularities	189
7.6.4	Direct Kinematic Singularities	189
7.6.5	Decoupling the Degrees of Freedom	190
7.6.6	Dimensional Synthesis	191

7.7	Conclusion	193
8	Conclusion	194
8.1	Review of the Contributions	195
8.2	Limitations of the Reported Research	198
8.3	Recommendations for Future Research	199
8.4	Final Remarks	200
	Bibliography	201
	Appendices	224
A	Orthonormality of Numerically Estimated Rotation Matrices	224

List of Tables

3.1	General specifications of the proposed voice coil actuator design . . .	40
4.1	Geometric parameterization of 3- <u>P</u> SS/S and 3- <u>S</u> PS/S manipulators .	61
4.2	Coefficients of multiple determination for the regression models of the kinematic performance metrics	76
4.3	Optimal Kinematic Performance	78
5.1	Model parameters for the actuator	109
5.2	Inertia parameters for the candidate manipulators	109
5.3	Controller tuning parameters for each kinematic loop of the candidate manipulators	109
5.4	Dynamic performance of the candidate manipulators while executing the test maneuvers	114
6.1	Mathematical complexity of the classic ADRC algorithm & the proposed resource optimized algorithm	125
6.2	Resource cost of FPGA implementation of the classic algorithm & the proposed resource optimized algorithm	134
6.3	ADRC controller tuning parameters for the nominal plant	134
6.4	Control performance obtained by the proposed resource optimized algorithm and the classic algorithm	135

6.5	Tuning parameters for the ADRC controller	141
6.6	Performance of the controller in the co-simulation study	141
7.1	Revised POM Geometry	158
7.2	Kinematic Performance Metrics Provided by the Optimal and the Revised POM Geometries	158
7.3	Variability of Kinematic Performance Metrics Observed in the Post Optimality Study	158
7.4	Transient response performance provided by the two configurations of the 3-PSS/S manipulator under different payload conditions	168
7.5	Details of the empirical direct kinematic model	180
7.6	Details of the empirical inverse kinematic model	180
7.7	Accuracy of the empirical direct kinematic model	183

List of Figures

3.1	Magnetics of the proposed VCA design.	30
3.2	Directions of the current in the coil and the corresponding resultant Laplace force for an applied voltage.	31
3.3	Experimental determination of the force constant of a prototype of the proposed VCA design.	32
3.4	Position sensing mechanism of the proposed VCA design.	34
3.5	Performance of the integral controller in the position sensing circuit.	35
3.6	Output of the position feedback circuit against actual displacement of the coil.	37
3.7	Detailed construction of the proposed VCA design.	38
3.8	Open-loop motion against gravity exhibited by a prototype of the proposed VCA design in response to step inputs of different magnitudes.	39
4.1	Kinematic structures of the candidate orientation manipulators.	46
4.2	A few examples of the fitted response surfaces of the 3- <u>P</u> SS/S manipulator (parallel configuration).	74
4.3	Optimal reachable workspace of the 3- <u>P</u> SS/S manipulator (parallel configuration).	79
4.4	Optimal reachable workspace of the 3- <u>P</u> SS/S manipulator (slanted configuration).	79

4.5	Optimal reachable workspace of the 3-SPS/S manipulator.	81
4.6	The projected orientation workspaces of the three candidate architectures in terms of azimuth and tilt angles.	81
4.7	Local dexterity index as a function of the joint coordinates.	83
4.8	Numerical stability of different methods estimating the dexterity characteristics of the optimal 3-PSS/S manipulator (parallel configuration).	84
4.9	Estimations of the optimal workspace characteristics provided by the 3-SPS/S manipulator exhibit asymptotic convergence.	86
5.1	The Eulerian Junction Structure and an articulation point.	94
5.2	Calculating the Euler angles from the angular velocity.	94
5.3	Bond graph model of kinematic joints.	97
5.4	Simple prismatic joint model.	98
5.5	Bond graph model of the voice coil actuator.	99
5.6	Complete bond graph model of a single kinematic loop of the parallel and the slanted configurations of the 3-PSS/S manipulator.	99
5.7	Structure of a linear extended state observer.	102
5.8	Cycloidal motion for a unit displacement in unit time.	104
5.9	Characteristics of the nonlinear weighting function $fal(\cdot)$	105
5.10	Topology of ADRC for a single kinematic loop.	106
5.11	Model verification through the application of the inverse kinematics models of the candidate manipulators.	113
5.12	System responses in a test maneuver executed by the parallel configuration of the 3-PSS/S manipulator.	115
5.13	System responses in a test maneuver executed by the slanted configuration of the 3-PSS/S manipulator.	115

5.14	Angular motion of the moving platform in a representative maneuver executed by the candidate architectures.	116
6.1	Cycloidal motion profile generation on an FPGA.	126
6.2	Accuracy of the motion profile as calculated by the profile generator module with respect to a double precision floating point calculation. The displacement corresponds to the entire range of a 16 bit position sensor.	126
6.3	FPGA implementation of the $falk(k_x, e_x, \alpha_x, \delta_x)$ function.	128
6.4	Accuracy analysis of FPGA evaluation of the $falk(k_x, e_x, \alpha_x, \delta_x)$ function through piece-wise linear approximation.	130
6.5	Recorded controller variables provided by the resource optimized algorithm in a representative trial displacement.	136
6.6	Frequency response of the experimental plant.	136
6.7	Block diagrams of the actuator model and the sensor model.	140
6.8	Architecture of the co-simulation experiment.	140
7.1	Design of the magnetically loaded, zero backlash spherical joint.	148
7.2	Experimental evaluation of the proposed spherical joint.	150
7.3	Experimental observations in the static repeatability test.	153
7.4	Experimental demonstration of position sensor dynamics.	155
7.5	Experimental results obtained from the slider-crank mechanism.	156
7.6	Virtual prototypes of the parallel and the slanted configurations of the 3-PSS/S architecture.	159
7.7	Prototypes of the parallel and the slanted configurations of the 3-PSS/S architecture.	159

7.8	Histograms of workspace related kinematic performance metrics sampled in the post optimality study.	160
7.9	Histograms of dexterity related kinematic performance metrics sampled in the post optimality study.	161
7.10	Test payload employed for quantifying the transient response performance of the manipulator prototypes.	163
7.11	Joint space responses observed in a representative test maneuver. . .	165
7.12	Tilt and azimuth angles observed in a representative test maneuver. .	166
7.13	Torsion angles and magnitudes of the angular velocity of the moving platform observed in a representative test maneuver executed by the manipulator prototypes.	167
7.14	Joint space responses observed in test motion A.	171
7.15	Joint space responses observed in test motion B.	171
7.16	Workspace responses observed in test motion A.	172
7.17	Workspace responses observed in test motion B.	172
7.18	Magnitudes of angular velocity vector observed in test motion A. . . .	173
7.19	Magnitudes of angular velocity vector observed in test motion B. . . .	173
7.20	Magnitudes of angular velocity vector observed in test motion B. . . .	174
7.21	Data acquisition for kinematic calibration.	178
7.22	A few examples of response surfaces generated from the empirical inverse kinematic model.	181
7.23	A few examples of response surfaces generated from the empirical direct kinematic model.	181
7.24	Kinematic structure of the 2P-S-S/U parallel orientation manipulator.	185
7.25	Kinematic performance of the torsion-restricted POM.	192

Chapter 1

Introduction

The revolutionary advancement of semiconductor and computing technologies in the last few decades has led to their widespread application in all disciplines of engineering. Following this trend, modern mechanical systems have been pushing the envelope of performance and system capacity by integrating electrical, electronic, and software components into their architectures. As a result, these systems have become more autonomous and more aware of their environments than ever. Even new functionalities have been created by combining multi-domain engineering knowledge. For instance, the transformation of the classic design of an internal combustion engine from a strictly mechanical system into its present-day hybrid construction can be considered. The efficiency of its primary function of converting the chemical energy of fossil fuel into mechanical energy has been improved by incorporating a variety of sensors, actuators, electronic control units, and communication modules into the contemporary design. In addition, new functionalities such as early fault detection and remote monitoring have been developed. Indeed, the disciplinary boundaries have become increasingly vague in recent years. Mechatronics engineering has appropriately availed of this opportunity to develop a holistic approach towards the practice of mechanical engineering and consequently has driven conventional

mechanical systems to achieve unprecedented performance and autonomy. Although the scope of mechatronics originally encompassed electrical, electronics, and mechanical hardware interfaced with intelligent software, the recent trend of integrating optical elements and technologies into mechatronic systems has given rise to a new engineering paradigm termed as “optomechatronics” or “optomechatronic technology” [1]. In [2, pp. 8-9], optomechatronics has been characterized as a subset of mechatronics engineering that has aided the evolution of engineering systems towards a state of greater precision, reliability, and intelligence.

The major functions of optical elements in optomechatronic systems include illuminating, sensing, actuating, etc. [2, p. 13]. In order to perform these functions, high speed manipulation of optical detectors and sensors (e.g., laser projectors, cameras, mirrors) is required in many applications; examples include laser scanning [3, 4], beam steering [5, 6], image stabilization [7], camera orientation [8–11], tracking [12, 13], free space optical communication [14], etc. Although galvanometer mirrors have been used in beam steering applications (e.g., [15]), their range of motion is generally limited. Alternatively, many optomechatronic applications utilize kinematic mechanisms for orientating optical payloads (i.e., *orientation manipulation*). Development of a suitable orientation manipulator involves a number of design tasks that include kinematic synthesis and analysis, prototype implementation, formulation of a suitable motion control application, etc. Performing the aforementioned synthesis, analysis, and implementation exercises requires expert application of engineering knowledge from several disciplines including robotics, electronics, precision manufacturing, and control engineering. Correspondingly, this thesis focuses on the theoretical and the practical aspects of designing and implementing prototypes of orientation manipulators that can be employed in small-scale, remotely deployable optomechatronic applications.

1.1 Motivation

General requirements for orientation manipulators in optomechatronic applications include speed, accuracy, reliability, and large range of motion. In addition, remotely deployable manipulators must possess appropriate SWaP (size, weight, and power) characteristics. It is generally difficult to adhere to these specifications when an orientation manipulator is constructed using off-the-shelf components, since off-the-shelf actuators, sensors, or control hardware that are designed for general purpose use cannot address the application-specific requirements of speed, size, reliability, or accuracy. Admittedly, it may happen that a commercially available component delivers the desired functionalities well at the cost of little or no modification to the overall system design. In such a case, it must be incorporated into the system architecture because of the efficiency it can provide in terms of development resources. However, such an occurrence is rare in practice. In contrast, a more coherent development strategy where each component is designed purposefully with a strong focus on the end goal is considered to be more effective in realizing the design specifications. Thus, this thesis is principally motivated by the need for a design approach that addresses the performance requirements of orientation manipulators for optomechatronic applications from the ground up. Specifically, this thesis focuses on parallel orientation manipulators because they offer potentially superior performance than the conventional Gimbal mechanism in terms of speed and accuracy.

The structural synthesis of parallel orientation manipulators [16, 17] have been studied extensively in the literature. Although it is yet to be established analytically, it can be remarked with cautious reservation that all kinematic structures for parallel orientation manipulators have been synthesized. Correspondingly, this thesis employs the published architectures for designing small-scale orientation manipulators. How-

ever, the existing art involving the determination of the optimal geometry of a parallel orientation manipulator is still developing. The published optimization methods are often computationally expensive and unnecessarily complicated (see Section 2.3). In view of these limitations, this thesis formulates an efficient method for synthesizing the optimal geometry of a given kinematic architecture, which is based on the response surface methodology [18].

Many motion control technologies for robotic manipulators have been proposed in the literature (see Section 2.4). Most of these control algorithms attempt to obtain robust performance by evaluating a complex dynamical model of the target system. Such an approach is not feasible for remotely deployable applications, since the corresponding control hardware is usually too limited in terms of computation capacity to deliver real-time performance. In recognition of this difficulty, this thesis proposes a resource optimized formulation of the active disturbance rejection control (ADRC) [19] technology. Unlike many modern model based controllers, ADRC can deliver robust motion control performance without the requirement of a model of the system, which allows this control technology to be particularly well-suited for embedded applications.

Although evaluating a control algorithm in simulation constitutes an important and necessary step in the development cycle of control applications, similar performance cannot be guaranteed to be replicated in a physical implementation. It is because unmodeled dynamics, occurrence of environmental noise, and perturbation of dynamic parameters (e.g., mass, compliance, inductance, resistance, etc.) can render even the most comprehensive dynamic model to be a weak representation of the actual system. Thus, a proper evaluation of a control algorithm must involve a comprehensive in-situ testing under realistic operating conditions. Among the control hardware platforms that are remotely deployable, field programmable gate arrays (FPGA) are preferred

in many embedded control applications because they can provide superior speed, power efficiency, and greater reliability [20]. In addition, they can mitigate obsolescence, which is important in view of the ever changing technological landscape in today's world. Since implementing a control application on an FPGA hardware is an exercise in designing digital circuits, it is yet to be adopted widely by the practitioners of control engineering, who are accustomed to developing control applications for the conventional control platforms in the form of software codes. Correspondingly, this thesis proposes efficient designs of digital circuits that implement the ADRC algorithm on an FPGA hardware.

Due to manufacturing tolerances, a robotic manipulator can never be constructed to conform exactly to the design specifications. In practice, the departure from the specified dimensions is compensated by performing a robot calibration exercise, which involves experimental localization of the articulation points and the joint axes. The geometric information thus obtained is subsequently employed to evaluate the kinematic model so that controlled robotic maneuvers can be performed in an accurate manner. Since a parallel manipulator generally possesses a greater number of articulation points than its serial counterpart, the conventional approach towards the kinematic calibration of parallel robots can be cumbersome. This thesis adopts an alternative approach where an empirical relationship between the actuated joint coordinates and the workspace coordinates is established.

1.2 Statement of Co-Authorship

This thesis has been undertaken within the framework of the Intelligent Sensor Platforms for Remotely Piloted Vehicles (INSPIRUS) project at the Memorial University of Newfoundland. The primary focus of this project includes the development of au-

onomous operation of unmanned vehicles. As a part of the project, the thesis author has collaborated with other researchers, and the subsequent research outcomes have been published in peer-reviewed journals and conferences. Among those publications, this thesis includes content from [21–23], which were principally written by the thesis author. As is the norm, permission for reusing each of these articles in an academic dissertation was obtained from the appropriate copyright owners.

The contributions of the thesis author in these publications include conceptual development of the central ideas and the theoretical treatises, designing the numerical and the physical experiments, data analysis and interpretation, and literary compositions of the articles. The co-authors have contributed by providing expert critique on the research approach and the theoretical foundation. In addition, they have assisted in executing the experiments, and have implemented the designs devised by the thesis author.

With the exceptions of Chapter 3 and Section 7.2, the author claims total intellectual ownership of the engineering designs, the formulations, and the analyses presented in this thesis. The design ideas of the electromechanical system described in Chapter 3 and the spherical joint documented in Section 7.2 were primarily conceived by Dr. Nicholas Krouglicof, the principal investigator of the INSPIRUS project. The thesis author, along with other project personnel contributed towards the implementation, refinement, and revision of the aforementioned design ideas.

1.3 Contributions of the Thesis

The research discoursed in this thesis revolves around two axes: design synthesis and design implementation of small-scale, remotely deployable parallel orientation manipulators. Correspondingly, the contributions of this thesis include both theoretical

and practical aspects of parallel robotics, which are elaborated in the following list.

Design, prototyping, and experimental evaluation of a novel voice coil actuation system: A voice coil actuator with a large actuation capacity was designed to address the limitations of the existing art involving small-scale electromechanical actuators. The novelty of this system originates from a highly accurate position sensing mechanism and an electronic drive circuit, both of which are seamlessly integrated into its architecture to provide a turn-key solution. Experimental evaluation of the proposed system confirmed that the design goals of accurate and high speed operation were achieved.

Dimensional synthesis of parallel orientation manipulators through the application of response surface methodology: Because of the complex relationship between the kinematic performance provided by a robotic manipulator and its geometry, a corresponding mathematical model that explicitly defines the underlying function is generally difficult to formulate. Alternatively, this thesis employed response surface methodology to develop an empirical estimation of this function. The implicit model thus obtained was subsequently used to determine the optimal manipulator geometry that maximizes an application-relevant set of kinematic performance features.

Development of a resource efficient formulation of ADRC algorithm and experimental evaluation on FPGA hardware: Although the computational complexity of ADRC is generally lower than that of a model-based controller, a remotely deployable application demands further simplification. Therefore, this thesis developed a resource efficient formulation of the ADRC algorithm. Its performance was experimentally validated using a corresponding implementation on an FPGA hardware.

Prototype implementation of high speed orientation manipulators for op-

tomechatronic applications and performance characterization: Employing the novel voice coil actuators described above as motion generators, three different prototypes of orientation manipulators were constructed. A series of performance characterization experiments were performed to evaluate the kinematic and the dynamic performances.

Empirical kinematic calibration of parallel orientation manipulators: This thesis proposes a simplified calibration method wherein a multivariate polynomial model is developed in order to empirically represent the relationship between workspace coordinates and actuated joint coordinates. This is in stark contrast with the conventional calibration methods where the goal is to determine a geometry that minimizes the discrepancy between some kinematic quantities obtained experimentally and the same provided by an analytic kinematic model. The accuracy of the proposed method was ascertained experimentally.

1.4 Organization of the Thesis

The remainder of the thesis is organized as follows.

Chapter 2 briefly reviews the existing literature relevant to the structural synthesis, the dimensional synthesis, and the control problem involving parallel orientation manipulators.

Chapter 3 details the proposed design of a novel voice coil actuator. It also reports the experimental evaluation of its performance.

Chapter 4 documents the procedure of determining the optimal geometries of a set of orientation manipulators featuring different kinematic architectures.

Chapter 5 constructs appropriate dynamic models of the candidate manipulators in order to conduct simulation studies.

Chapter 6 implements a resource efficient implementation of the ADRC technology on an FPGA hardware.

Chapter 7 develops the prototypes of the candidate manipulators. The dynamic performance of these prototypes were comprehensively evaluated. In addition, an empirical approach was proposed to perform the kinematic calibration of parallel orientation manipulators.

Chapter 8 offers the concluding remarks. In addition, possible avenues for future research are also discussed.

Chapter 2

Literature Review

2.1 Introduction

An orientation manipulator constraints its payload to only spherical motion [24, p. 28] about a point fixed in three dimensional (3D) space. In terms of kinematic topology, such manipulators are characterized either by a serial architecture or a parallel architecture. The classic Gimbal mechanism [25, 26] is the most intuitive and the most common embodiment of a serial orientation manipulator. In addition, many parallel kinematic architectures (PKM) featuring three rotational degrees of freedom (3 DOF) have been reported in the literature [17, 27]. Although it is widely claimed that the parallel manipulators are generally superior to their serial counterparts in terms of accuracy, speed, and stiffness, only a few comparative analyses are reported in the literature; examples include the case-specific studies in [28, 29]. Even in their limited scope, these studies do not unanimously support the aforementioned general claims. However, the perceived superiorities of a parallel mechanism should be recognized as potential advantages [30]. It should be noted that the moving platform in a parallel architecture is actuated by multiple kinematic chains as opposed to a single kinematic chain in a serial architecture. As a result, the kinematic structure of a PKM is

generally more conducive to better performance that can be attained at the cost of greater mechanical and control complexity. Nonetheless, careful engineering design and execution is necessary in order to realize the potentials of a parallel manipulator. Paul Sheldon, the designer of the Variax machine-tool appropriately remarks [27, p. 301]: “*The Variax, which is now over 10 years old, still stands as an existence proof of PKM potential. For instance, it is 3 to 6 times stiffer than a typical good conventional machining center. But the many PKMs erroneously conceived and poorly executed since then have proven inferior to conventional approaches and have deterred the advance of the art... This sort of thing certainly does not instill confidence in the minds of potential customers, or encourage researchers to explore the technology.*”

The motion generated by the proximal link (i.e., closest to the mechanical ground) in a general Gimbal mechanism is inherently slower than those generated by the distal links, because the corresponding inertial load, which is an aggregation of the payload, all the distal links, and their actuators, is relatively large. Hence, it is difficult to achieve high accelerations utilizing the limited capacity of the actuator that drives the proximal link. The high inertial load of a Gimbal mechanism also causes large deflections, which ultimately results in inaccurate motion. This thesis focuses on small-scale orientation manipulators that can generate high speed, backlash-free, and accurate spherical motion of optomechatronic payloads. In view of these design requirements, the serial architecture (i.e., the Gimbal mechanism) is henceforth eliminated from consideration, and only parallel architectures are reviewed in this chapter.

Different nomenclatures have been used in the literature to refer to PKMs that provide only rotational motion; examples include *spherical parallel manipulator* [17, 31, 32], *parallel wrist* [33, 34], *rotational parallel manipulator* [35], *spatial orientation mechanism* [24, p. 129] etc. The definitions provided in [36, p. 12] and [24, p. 28]

characterize a *spherical mechanism* as the one where all moving links are also restricted to only spherical motion. According to these definitions, the orientation PKMs in [8, 21, 37, 38] do not qualify as *stricto sensu* spherical mechanisms, since only the moving platforms of these manipulators are constrained to rotational motion. In order to avoid ambiguity, this thesis uses the term “parallel orientation manipulator” (POM) to refer to PKMs that restrict the moving platform to only rotational degrees of freedom. In addition, the widely used convention of representing a PKM architecture by a string of alphanumeric characters is adopted. The composition of a kinematic chain is expressed by alphabetic characters, while a preceding numeral indicates the number of kinematic chains present in a PKM architecture. Each alphabetic character indicates a joint, and the order of the characters represents the actual joint arrangement starting from the mechanical ground. The different types of kinematic joints are encoded as follows: R for revolute joints, P for prismatic joints, U for universal joint, S for spherical joints, C for cylindrical joints, and H for helical joints. An underlined character indicates an actuated joint and a character preceded by a slash (e.g., “/X”) denotes a passive joint. For example, the 2-PSS/U manipulator in [9] possesses two kinematic chains, where each chain is composed of an actuated prismatic joint and two spherical joints. In addition, the presence of a passive universal joint is also indicated in the adopted representation.

Synthesis of a parallel manipulator is composed of two distinct, yet closely related tasks: (a) *structural (type) synthesis*, and (b) *dimensional synthesis*. Systematic generation of all parallel kinematic structures that allow the moving platform to perform a specified motion pattern (e.g., only translational, only rotational, or any other combination thereof) is defined as structural synthesis. In addition, dimensional synthesis aims to determine the joint locations and the links lengths of a given PKM architecture so that some kinematic performance can be achieved. As far as

kinematic performance is concerned, the geometry (i.e., dimensions) of a parallel robot is as important as its kinematic structure, because a poorly dimensioned parallel manipulator, whose kinematic structure is apparently more appropriate for a given function, may exhibit inferior performance than an alternative PKM with a well-designed geometry [27, p. 25]. When executed properly, structural and subsequent dimensional synthesis yields a PKM design that is capable of high performance. An accurate and robust motion controller must complement the mechanical design to actualize this potential. For the sake of a systematic discourse, all these different aspects of PKM design are discussed in the remainder of this chapter. After presenting an extensive list of POM architectures proposed in the literature, Section 2.2 qualitatively evaluates their feasibility for the desired application. In addition, the existing arts on the dimensional synthesis, and the motion control of PKMs are reviewed in Section 2.3 and Section 2.4 respectively. Finally, Section 2.5 offers the concluding remarks.

2.2 Kinematic Architectures of Parallel Orientation Manipulators

Structural synthesis of POMs is performed in [16, 17, 35, 39]. In addition, an extensive catalog of POM structures can be found in [27, pp. 35–43]. Among these POM architectures, the *passive constraint mechanisms* [27] restrict the moving platform to the mechanical ground by an unactuated spherical joint, which allows only rotational motion of the moving platform. Examples of these kinematic structures are characterized by these limb configurations: 3-UPS/S [37], 3-SPS/S [24, p. 130], 3-PSS/S [8], 3-PUS/S, and 3-RRRS/S [38]. It should be noted that the 3-SPS/S and the 3-UPS/S manipulators are virtually similar, as are the 3-PSS/S and the 3-PUS/S manipulators.

However, the difference between the limbs of each of these manipulator pairs arises from the type of the unactuated joint that is located away from the moving platform. Since it can be either a universal or a spherical joint, each has different practical implications, especially for the 3-SPS/S and the 3-UPS/S manipulators. Nevertheless, the choice of this joint does not affect the kinematic constraints on the moving platform. When a universal joint is used, the passive degree of freedom of each limb, which enables it to rotate about its own axis, becomes restricted. This significantly minimizes the possibility of link interference, especially when the limbs are *flatter* (i.e., a large width to height ratio) in shape. On the other hand, the mechanical construction of a universal joint is generally more complex. Regardless of the composition of the limbs, the passively constrained POMs possess three limbs as motion generators to actuate the moving platform. However, a fourth limb was introduced in [32, 40] to obtain a redundantly actuated system. Although enhanced performance in terms of increased workspace, elimination of singularity configurations, and improved dexterity may be achieved because of redundant actuation, the kinematic model and the corresponding motion control problem becomes increasingly complex.

In contrast to the passive constraint mechanisms with prismatic joints, the following POM architectures employ rotary actuation as motion input: 3-RRR [11], 3-R(2R/2S)S [33], 3-RUU [34], 3-URU [41], 3-RSR [42]. The 3-RRR architecture was employed to construct the “*Agile eye*” camera orientation device in [11]. Here the axes of all the revolute joints are concurrent and they coincide at the mechanism center. This architecture can provide a “theoretically unlimited and undivided orientation workspace” [43]. Gosselin and St-Pierre documented the performance of a prototype of the Agile eye in [44].

A relatively complex limb configuration represented by the 3-R(2R/2S)S structure was employed to build the “*Argos*” manipulator [33]. Each limb of this manipulator

is attached to the mechanical ground by an actuated revolute joint with its axis coinciding with the mechanism center. In addition, the moving platform is constrained by a spherical joint located at the distal end of each limb. These two terminal joints in each limb are connected by a planar parallelogram equipped with two revolute and two spherical joints (i.e., 2R/2S). Although the structure of the Argos manipulator is theoretically intriguing, the large number of joints renders its kinematic structure unsuitable for small-scale applications.

The 3-URU manipulator in [41] has an ingenious architecture that can operate in any of the following modes: purely translational, purely rotational, or mixed DOF. When certain geometric constraints involving the different joint axes of the limbs are met, this architecture provides spherical motion. Moreover, it allows transition from a translational mode to an orientation mode without disassembly [27, p. 39]. Similar to the 3-URU architecture, the 3-RUU manipulator in [34] provides both translational and rotational motion. However, it is unclear whether a transition between the two operational modes is possible without disassembly for the 3-RUU manipulator.

Without specifying the actuated joints, Fang and Tsai synthesized the following POM structures in [35]: 3-RRS, 3-CRU, 3-UPC, and 3-CRC. These POM architectures along with the ones presented above are considered symmetrical, since they are composed of identical limbs. However, a few *asymmetrical* POM architectures are proposed in [31], where the limbs are structurally different. The diversely composed limbs in an asymmetrical POM render its prototyping task difficult.

Since the synthesis of a robotic manipulator is a developing field [30], analytical tools for conducting a quantifiable comparison of the available choices for kinematic architectures are scarce in the literature. As a result, the exercise of identifying a suitable kinematic topology often relies on qualitative and philosophical reasoning [36, p. 3]. Nonetheless, an evaluation scheme for POM architectures proposed in [45] combines

quantitative and qualitative analysis. Although the design study in [45] primarily focused on the kinematic characteristics of the candidate architectures, dynamic performance characteristics such as speed and accuracy were not evaluated. Thus, the scope of the study in [45] is limited to applications where the requirement of dynamic performance is not as stringent as in an optomechatronic application. In contrast, this thesis quantitatively evaluates the kinematic and the dynamic performance of a set of *feasible* POM architectures. To this end, the feasible POMs (i.e., candidate architectures) are selected by a qualitative analysis of their potential to achieve the application-specific requirements. Since accurate movement and compact size are two major prerequisites of a successful prototype, suitable architectures must be composed of a small number of simple joints that can provide backlash-free motion. In these regards, the revolute and the prismatic joints are favored because of their simple mechanical construction. If precision manufacturing is employed to fabricate these joints, accurate relative motion with minimal friction can be achieved. Although the conventional construction of a spherical joint does not encourage backlash-free motion, it is also considered suitable for the desired application, because this limitation of the conventional spherical joint is addressed in Chapter 7 by proposing an appropriate mechanical design that provides accurate motion over a large range. Correspondingly, the available kinematic structures are screened on the basis that a suitable architecture must possess any combination of the three preferred joint types. The following manipulators are subsequently selected from the extensive list of POM architectures presented here: 3-PSS/S, 3-SPS/S, 3-RRRS/S, 3-RRR, 3-R(2R/2S)S, 3-RSR, and 3-RRS. Because of the large number of joints, the 3-RRRS/S and the 3-R(2R/2S)S manipulators are considered infeasible for the desired application. From the remainder of the list of prospective architectures, the 3-RRR, the 3-RSR, and the 3-RRS manipulators require all their joint axes to coincide with the mechanism center

in order to restrict the translational motion of the moving platform. Departure from this strict geometrical constraint results in parasitic motion [46], which compromises the manipulation accuracy and the stiffness of the moving platform [47]. Since the 3-RRR manipulator (i.e., Agile eye) is overconstrained, mechanical assembly of the manipulator is not possible when the joint axes are not concurrent. As a result, the designers of the Agile eye replaced the passive revolute joints with self-aligning pin joints, which is kinematically equivalent to replacing the passive revolutes with spherical joints [47]. Thus, a practical 3-RRR manipulator is actually an embodiment of the 3-RRS or the 3-RSR kinematic structure. In contrast, the 3-PSS/S and the 3-SPS/S manipulators are much more forgiving to imperfect manufacturing precision. In the absence of joint backlash, they are structurally incapable of any parasitic motion. However, these manipulators cannot accommodate the payload to be mounted at the mechanism center. Since a typical optomechatronic payload (e.g., camera, laser projector, etc.) only requires its viewing axis to be concurrent with the mechanism center instead of being mounted on it, this limitation is not considered critical. Thus, these two manipulators are regarded to be feasible for the desired application. However, quantitative performance analyses are performed later in this thesis in order to identify the most suitable of the two architectures.

2.3 Dimensional Synthesis of Parallel Manipulators

Despite the potential advantages of speed, accuracy and stiffness, the limiting factors that may deter the performance of a parallel manipulator include workspace volume, presence of multiple singularities in the workspace, limited range of the link lengths,

range of the available motion of the joints, and possible link interference [48, 49]. Dimensional synthesis of a parallel manipulator refers to the systematic determination of the optimum geometry that minimizes these limitations so that a set of application relevant kinematic performance characteristics can be achieved. The kinematic characteristics that are commonly studied in the related literature include workspace volume [50–54], dexterity [21, 55], accuracy [56], stiffness [57], etc. It should be noted that the geometry of a given PKM architecture is generally defined by the locations of the joints and the lengths of the constituent links.

A workspace is said to be well-conditioned or dexterous if the inverse kinematic Jacobian of the manipulator remains strictly nonsingular over the entire workspace during all possible robotic maneuvers. Since an ill-conditioned inverse Jacobian implies transformation of a relatively small displacement of the input link to a large displacement of the moving platform, the manipulator is no longer able to function accurately. A frequently cited dexterity index is the reciprocal of the Euclidean norm condition number of the inverse Jacobian matrix (e.g., [32, 48, 58]), which measures only the local dexterity of the point at which the inverse Jacobian is evaluated. The quality of the entire workspace can be quantified by the global conditioning index (GCI) [59], which is an integral of the local dexterity index over the entire workspace. Besides GCI, the manipulability index [60] is also employed to measure the quality of a workspace. It quantifies the motion transmission quality in a PKM in terms of how effortlessly the actuators can maneuver the moving platform.

Except for very simple robots, an explicit mathematical model of the kinematic characteristics (i.e., workspace volume, dexterity, etc.) in terms of the kinematic parameters is extremely difficult to derive. Hence, a gradient-based approach is not readily applicable because of the unavailability of a gradient matrix. However, this challenge was negotiated in [32] by numerically estimating the gradient of local

dexterity in order to synthesize the geometry of a redundantly actuated 4-SPS/S architecture. In general, a gradient-based optimization method is not well suited for dimensional synthesis of parallel robots. As an alternative, parameter search approaches have been conveniently adopted as the preferred optimization methodology in many previous publications. For example, [56] employed an iterative parameter search in order to optimize a 3-PRS hybrid mechanism (one translational and two rotational degrees of freedom) for three objectives: predefined workspace volume, force generation capacity and maximal positioning error. A similar approach based on the controlled random search (CRS) algorithm can be found in [48], which attempts to optimize the translational workspace of a Stewart-Gough platform for a given orientation. An iterative parameter search was also employed in [55] in order to optimize an objective function that combines workspace volume and other dexterity indices for a 3-UPU and a 3-UPS mechanism. These parameter search approaches generally lack efficiency as the search space is either randomly or exhaustively explored to find a solution.

Besides parameter search, many other solution approaches are also reported in the literature. In [61], both discrete and continuous optimization approaches have been adopted to optimize the position of the redundant actuator of a planar parallel manipulator. Although most of the related work focus on maximizing a set of kinematic performance metrics, an alternative objective function was considered in [62] that determines the geometry of an n degrees of freedom (DOF) parallel robot for a given workspace and predefined manufacturing tolerances so that the moving platform can be positioned with minimal error. In contrast, Chapter 4 aims to find optimal geometries of the candidate architectures so that maximum kinematic performance can be achieved when a preferred actuator is employed as the motion generator. This actuator [63] has been custom designed to address several application-specific

requirements, such as compactness, dynamics and positioning accuracy.

Optimization techniques inspired by the Genetic Algorithms (GA) are also found in the literature. For example, a GA was employed in [54] in order to maximize the workspace of a 3 DOF spherical manipulator. Other kinematic characteristics were not considered. However, a multi-objective synthesis problem was solved in [57] by employing a GA. The geometry of a 2 DOF parallel robot was optimized in [57] by maximizing a set of kinematic characteristics including the workspace volume, the dexterity, and the stiffness characteristics. In order to find an optimum solution through the application of a GA, it is usually necessary to sample the objective function at a large number of trial points. Thus, such solution approaches are generally inefficient.

In [8], following a parameter variation study, an implicit filtering algorithm was employed to maximize an interesting objective function that incorporates the dimension ratio of the workspace volume to package volume of a 3-PSS/S manipulator. With regards to this objective function, it is worth mentioning that the workspace of an orientation mechanism is scale invariant and the packaging volume is directly proportional to the scale of the manipulator. However, a parameter variation study can provide, at a minimum, a qualitative sense of how kinematic parameters influence desirable kinematic characteristics. The insight gained from parameter variation studies can lead to the development of several criteria that can be employed to objectively reduce the search space.

In a related work, Shin *et al.* adopted the Taguchi method for maximizing stiffness and the workspace volume of a planar 3-RRR manipulator in [53]. Although the Taguchi method is more systematic than other solution approaches, within the statistics community it has been criticized for being unnecessarily inefficient and complicated [64]. An efficient alternative is the response surface methodology (RSM). It can provide an optimum solution by systematically probing the search space at a small number

of points. However, conventional RSM optimization techniques (e.g., Box-Benken design) are not well suited for negotiating the high complexity of the response space associated with the dimensional synthesis of parallel robots. It was shown in [65] that an enhanced RSM technique that incorporates Latin hypercube design, Kriging interpolation, and neural network training outperforms a conventional RSM solution in terms of synthesizing a robust design of a 4 DOF hybrid robot. Correspondingly, Chapter 4 employs a Latin hypercube sampling of an arbitrarily chosen parameter space in order to localize a region of interest. In a subsequent step, an IV-Optimal (integrated variance) experiment design is used to explore the reduces search space.

2.4 Motion Control of Parallel Manipulators

The dynamics of a general PKM is time-varying because the poses of the moving links determine the inertial load acting on its actuators. Although a quantified identification is scarce in the literature, qualitative reasoning suggests the existence of strong cross-coupling among the kinematic chains of a parallel manipulator. Except for the specially designed decoupled PKMs (e.g., [66–68]), these nonlinearities further degrade the controllability of a PKM. Even a simple parallel motion stage was shown to possess significant cross-coupling effects in [69]. Because of the nonlinear, time-varying and coupled dynamics of a PKM, formulating an appropriate controller is difficult. Indeed, the “*design for control*” [70] philosophy was adopted in [71] in recognition of this challenge. Nonetheless, several control technologies have been reported in the literature in order to solve the motion control problem of parallel manipulators.

Recognizing the simplicity of implementation offered by the PID controller, it has been employed in [9, 68, 72–74] for controlling the motion of different parallel manipulators. Although a PI controller in its classic form was used in [9, 72, 73] to

implement joint space control, the controllers in [68, 74] incorporate advanced control concepts to obtain improved performance despite the nonlinear plant dynamics. A feedforward term calculated from the input-output relationship of the actuators was used along with the classic PID algorithm in [68] so that robust and accurate tracking performance can be achieved. The high degree of cross-coupling nonlinearity in a planar 3-PRR manipulator was addressed in [74] by incorporating a saturated PI controller with a classic PD feedback loop. To this end, the concept of saturated control was used in order to minimize the synchronization errors among the kinematic chains of the manipulator. Although the damped nature of the piezo-electric actuators used in [9, 72, 73] may justify the use of a PI controller, reported benchmarking studies show that the classic PID controller cannot provide acceptable accuracy in the control performance of a PKM [69, 74–79]. Correspondingly, model-based controllers have been used for PKM operation as an alternative to the PID technology.

A model-based computed torque controller (CTC) employs a dynamic PKM model to predict future system responses so that control efforts can be adjusted at the present time accordingly. For example, accurate positioning maneuvers of a translational 3 DOF Cartesian PKM were obtained by a model-based CTC in [76]. The corresponding dynamic model was formulated by employing the Lagrange-D'Alembert method. Similarly, the computed torque controller in [77] was developed by employing the Lagrange-D'Alembert formulation to obtain the inverse dynamic model of a redundantly actuated 2 DOF parallel manipulator. In addition, a modified CTC that employs desired joint coordinates to compute the inverse dynamic model was used in [78] for a 6 DOF parallel manipulator called Hexaglide. Although no specialized technique was reportedly used in [76–78] for determining the controller gains, a neural network approach was adopted by [80] in order to obtain the controller parameters of a CTC. Besides PID and CTC, other control technologies have been

adopted for PKM operation including fuzzy logic controller [81,82], nonlinear adaptive controller [79], model predictive controller [83], robust H_∞ controller [84,85], sliding mode controller [86,87], LQG controller [88], etc.

Although the controllers referenced above are unique in many aspects, the most elemental difference arises from whether the underlying algorithm requires a model of the plant in order to calculate the control efforts. This has significant implications with regards to any high speed PKM. Since the model of a parallel manipulator has to account for the multi-body dynamics resulting from a closed-loop kinematic structure, a corresponding implementation is computationally cumbersome. Consequently, achieving real-time performance is rendered difficult. If the control application requires remote deployment, obtaining a sufficiently high sampling frequency of the control algorithm becomes even more challenging. This is because typical embedded control hardware does not possess the computing capacity required for real-time evaluation of a complex mathematical model. Thus, a successful implementation of these controllers requires the corresponding mathematical models to observe two competing constraints: (a) being able to provide sufficiently accurate estimation of the actual dynamics, and (b) being computationally simple enough to be evaluated in the control hardware at an adequate speed. As a compromise, model simplification becomes necessary to an extent that the simplified model may no longer represent the physical system accurately enough for the controller to be effective. In such a case, active disturbance rejection control (ADRC) offers an attractive solution that promises robust performance without the requirement of a system model.

ADRC is a novel control paradigm proposed by Han [19] that packages the best features of classical and modern control theory in a single architecture. From classical control theory, it borrows the idea of error driven, rather than model-based, control law. In addition, it employs an extended state observer (ESO) [89] that estimates

the aggregated effect of plant dynamics and external disturbances in order to adjust the error driven control effort accordingly [90]. Thus, ADRC establishes itself to be a powerful control technology that is applicable to n-th order, nonlinear, time-varying, MIMO systems [91]. Although ADRC has been successfully implemented in many control problems [92], a survey of the related literature suggests that its application in the domain of PKMs has been extremely limited. However, an exception can be found in [93] where a high-precision ADRC motion controller was developed for a Stewart platform.

Because an ADRC controller is composed of a nonlinear PD feedback loop along with other components, its implementation complexity is higher than a classical PID controller. Nonetheless, its architecture is generally simpler than any other similar model-based controller, which indicates a relatively low computational cost of a corresponding implementation. Thus, it can be regarded extremely suitable for control hardware that are remotely deployable. It was shown in the simulation study presented in [23] that an ADRC controller can provide robust performance for a 3-PSS/S manipulator. Correspondingly, this thesis identifies ADRC as the most appropriate controller for optomechatronic applications of parallel orientation manipulators.

2.5 Conclusion

The existing literature on the structural synthesis of parallel orientation manipulators is reviewed in this chapter. In addition, prior art on the dimensional synthesis of PKMs are also discussed. By reviewing the proposed kinematic structures for POMs, the potentials of these architectures to satisfy application-specific requirements were objectively assessed to identify a set of feasible structures for further analysis. The

two kinematic structures that were selected for subsequent kinematic and dynamic performance benchmarking studies are characterized by a 3-PSS/S and a 3-SPS/S limb configuration. After discussing the existing motion control technologies adopted for PKMs, the active disturbance rejection control (ADRC) algorithm was selected for prototyping a controller for the desired application.

Chapter 3

Design & Prototyping of a Novel Voice Coil Actuator

Since commercially available linear actuators cannot address the requirements of a remotely deployable optomechatronic application, this chapter adopted a ground-up design approach in order to construct a voice coil actuator that provides a large actuation capacity in a compact form factor. To this end, the conventional coil was replaced by a multi-layered PCB. The current conducting traces on the PCB generate the actuation force by interacting with an augmented stationary magnetic field provided by a modified Halbach array. In addition, an optical position sensing mechanism providing feedback signals with sub-micron accuracy was integrated into the construction of the actuator in order to obtain a turn-key, closed-loop positioning solution. Furthermore, the power electronics of the actuator was purposely embedded on the existing PCB coil to obtain a compact design. Physical experiments were conducted in order to confirm the suitability of the proposed design for the desired application.

3.1 Introduction

The 3-PSS/S and the 3-SPS/S manipulators employ linear actuation in order to obtain spherical motion of the moving platform. Although several off-the-shelf linear actuators are commercially available, their general purpose design acts as a deterrent to satisfying the specific requirements of the desired application. Consequently, a linear actuator was designed from the ground up in order to address the application-specific requirements of speed, compactness, and positioning accuracy. In view of these requirements, an electro-magnetic actuator is preferred over a pneumatic or a hydraulic motion generator. Moreover, electro-magnetic actuators are more suitable for remote deployment. Among the various types of electro-magnetic motion generators, the voice coil actuator (VCA) is regarded as the most appropriate choice with regards to the desired application. It is because the non-commutative mode of operation of a VCA offers the potential of achieving high speed performance. Correspondingly, this design is selected for constructing the custom actuator for the desired application.

Voice coil actuators are simple electro-mechanical systems that are employed as motion generators in many optomechatronic applications involving image stabilization or high speed manipulation of optical devices (e.g., mirror, laser projector, camera, etc.) [94–96]. General requirements for such applications include high actuation capacity, fast response, and precise movement in a compact form factor. However, the classic embodiment of a VCA featuring a cylindrical coil embedded in a hollow cylindrical magnet is not conducive to these requirements. Although commercially available VCA units offer several improvements over the classic design, they are usually equipped with neither a position sensor nor an electronic drive system, which are necessary to build a closed-loop positioning device. As a result, incorporation of a commercial VCA into a size-constrained application becomes difficult, since it must

be fitted with additional components to obtain the desired functionality. These issues were addressed in [63,97] by adopting a novel VCA design that replaced the traditional coil with a printed circuit board (PCB), an integrated position feedback circuit, and an augmented stationary magnetic field provided by a modified configuration of planar Halbach arrays. However, this chapter proposes several improvements over the previous design in [63] including a highly integrated electronic drive system implemented on the existing PCB coil, a refined position feedback system, and a higher motor constant yielding superior thermo-electric characteristics.

The aforementioned ideas are elaborated in the remainder of this chapter, which is organized as follows. Section 3.2 discusses different innovative aspects of the proposed VCA design. The performance achieved by the proposed design is experimentally evaluated in Section 3.3. Finally, the concluding remarks are offered in Section 3.4.

3.2 Proposed Design of a Voice Coil Actuator

When the reluctance effect caused by the variation of coil inductance with the position of a VCA is negligible, the generated force is principally contributed by the Laplace force [98]. Consequently, a sufficiently large Laplace force is necessary to improve the actuation capacity of a VCA. Since the Laplace force F_L is the product of the conductor length l , the coil current i , and the magnetic field B (i.e., $F_L = Bli$), increasing the conductor length l and/or the current i generally improve the yield in Laplace force. However, overheating and power consumption concerns render this proposition unattractive. Alternatively, augmenting the stationary magnetic field improves the actuation capacity without penalizing the thermo-electric characteristics of the VCA. This idea is adopted in many VCA designs by incorporating an iron core at the center of the cylindrical coil to purposely focus the magnetic field into the coil and

to prevent magnetic flux leakage. However, the improvement in VCA dynamics thus obtained is countervailed by the undesirable nonlinearities arising from the presence of an iron core including Eddy current, reluctance effect, and magnetic saturation of the iron core [98]. Alternatively, Halbach arrays [99] can be used to augment the fixed magnetic field of a VCA; examples include [100,101]. In contrast to the classic cylindrical embodiment of the aforementioned designs, a compact construction is proposed in [102] that employs planar magnets. Although the stationary magnetic field in [102] is augmented by placing a flat coil in the the gap between several pairs of planar magnets with alternating pole orientation, a Halbach configuration was not adopted. Since bar magnets of equal size and strength are arranged in a preferred configuration in the classic planar Halbach array, the concentrated magnetic field exhibits substantial variation in magnitude over a short distance, which limits the stroke of the non-commutated VCA. As a result, planar Halbach arrays in their classic form are generally difficult to be adopted in a flat VCA design. Although increasing the size of the magnet elements may alleviate this limitation, a flat VCA design is no longer tenable. This thesis overcomes this design problem by adopting a modified planar Halbach configuration that does not increase the compact form factor of the proposed VCA (see Fig. 3.7).

3.2.1 Magnetics & PCB Coil Design

This design uses five neodymium magnets featuring two different strengths to construct a single Halbach array [Fig. 3.1(a) and Fig. 3.1(d)]. Although a pair of relatively stronger N52 magnets principally contribute the magnetic field in which the coil operates, three smaller N42 magnets are used in a particular pole orientation to focus the magnetic field in one side of the array (active side) and to cancel or diminish the

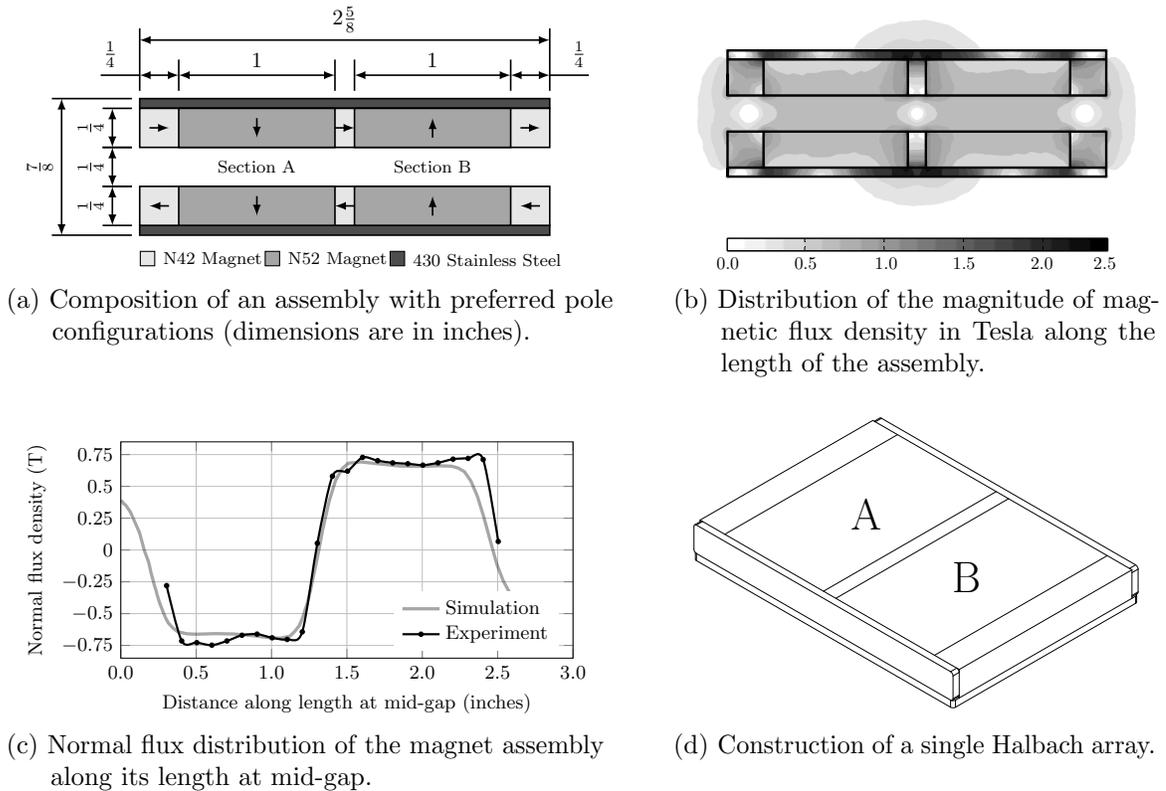


Figure 3.1: Magnetics of the proposed VCA design.

field on the other side (passive side) [Fig. 3.1(b)]. A magnetic stainless steel plate shields all sides of the array except for the active side [Fig. 3.1(a) and Fig. 3.1(d)] to further prevent the leakage of magnetic flux. In order to create a strong stationary magnetic field, two such Halbach arrays are assembled together so that their active sides face each other with a small gap of $\frac{1}{4}$ inch [Fig. 3.1(a) and Fig. 3.1(b)]. The coil of the proposed VCA operates in this small gap. The strong stationary magnetic field in the gap features two regions [i.e., Section A and Section B in Fig. 3.1(a)] of near-uniform flux density approaching 0.75 Tesla, each over a length of 1 inch. However, the magnetic pole orientations of these two regions are reversed [Fig. 3.1(c)]. Since reversal of the pole orientation in these two regions cannot be avoided without disjointing the Halbach configuration, the coil of the VCA must be designed accordingly

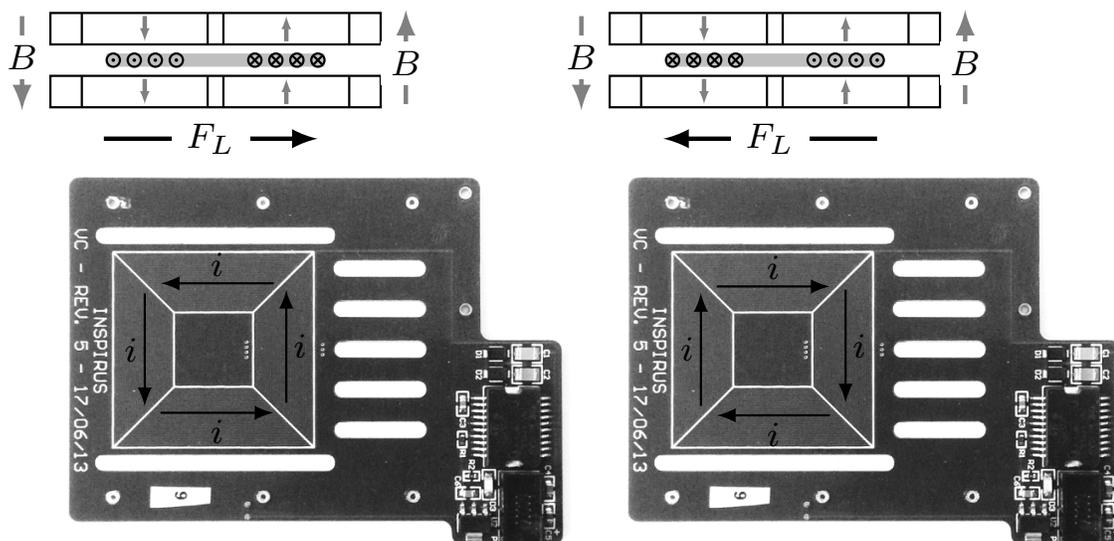


Figure 3.2: Directions of the current in the coil and the corresponding resultant Laplace force for an applied voltage.

to utilize both these regions for Laplace force generation. Correspondingly, current conducting traces on a PCB laid out in a spiraling square pattern constitute the coil of the proposed VCA (Fig. 3.2). As shown in Fig. 3.2, this pattern allows for reversal of the current in the left and right side of the square, which compliments the reversing pole orientation in the stationary magnetic field. Consequently, the Laplace forces generated by the traces on the right and left side of the square act in the same direction regardless of the direction of the voltage applied to the coil. In addition, the Laplace forces generated by the top and bottom side of the square cancel each other and do not contribute towards the motion of the actuator. The moment generated by all the Laplace forces in different sections of the square is canceled by the linear bearings that act as a motion guide for the actuator.

In addition to accommodating intricate trace patterns to form the coil of a VCA, the PCB technology offers a number of advantages over the conventional coil design. A multi-layered PCB can accommodate the spiraling square trace design in each of

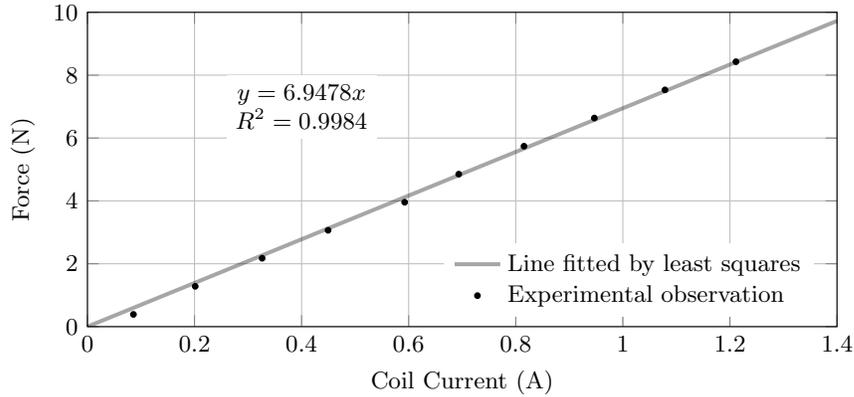


Figure 3.3: Experimental determination of the force constant of a prototype of the proposed VCA design.

its layers, which multiplies the length of the current carrying conductor drastically. An eight layered construction of the PCB coil yields an impressive 10.88 meters long current conducting trace over an area of 2.62 square inches. However, only half of that (i.e., 5.44 meters) contributes to the generation of the useful Laplace force. With respect to the previous design iteration in [63], the number of layers in the PCB coil is doubled in this design. Consequently, the length of the current conducting traces is also doubled, which improves the motor constant of the present design by a factor of two (i.e., $\frac{F}{i} = Bl$). Fig. 3.3 provides the force constant of the proposed VCA design. Assuming the resistivity of the current conducting traces does not change with the number of layers in a PCB coil, the coil resistance of the present design is also increased by a factor of two. However, with respect to the design in [63], this design requires only half the current to generate a certain force. As a result, heat dissipation due to the resistive load of the coil is reduced by a factor of two in comparison to the previous design.

Since the fiberglass material of a PCB allows for thin construction without compromising the strength of the moving coil (nominally 62 mils in this case), the Halbach arrays forming the magnet assembly can be placed in close proximity with minimal

spacing to prevent the loss of flux density. The PCB technology also offers convenient, cost-effective mass production of the coil using conventional manufacturing techniques that can yield accurate production units with a high degree of repeatable quality.

3.2.2 Position Feedback Mechanism

In order to develop an accurate position sensing mechanism capable of providing a high resolution signal approaching sub-micron accuracy, a 1D position sensitive device (PSD) is used as the sensor in the proposed VCA design. In addition, a vertical-cavity surface-emitting laser (VCSEL) diode is used to generate photocurrent in the PSD by striking it with a high photon density circular spot. The photocurrent generated from this light spot (diameter $< 200 \mu\text{m}$) is divided into two components and each component is collected at each of the two anodes of the PSD [Fig. 3.4(a)]. The active area of the PSD acts as a current divider and the generated photocurrent is divided in inverse proportion to the distance between the point of incidence and the respective anode [103]. If the photocurrents collected at the two anodes are denoted by I_1 and I_2 and the length of the active area of the PSD is L , the distance x of the point of incidence from the center of the PSD is provided by [103],

$$x = \frac{L(I_2 - I_1)}{2(I_1 + I_2)}. \quad (3.1)$$

When (3.1) is evaluated in a digital computer (e.g., a microcontroller), the analog photocurrents are converted to digital signals, which are then transmitted to the computer for subsequent processing. The sampling rate thus achieved may not be sufficient enough for the high acceleration of the proposed VCA (in excess of 30 g at no-load condition [63]). While an analog circuit implementing (3.1) can mitigate this issue, the division operation poses a challenge. It should be noted that the sum of the

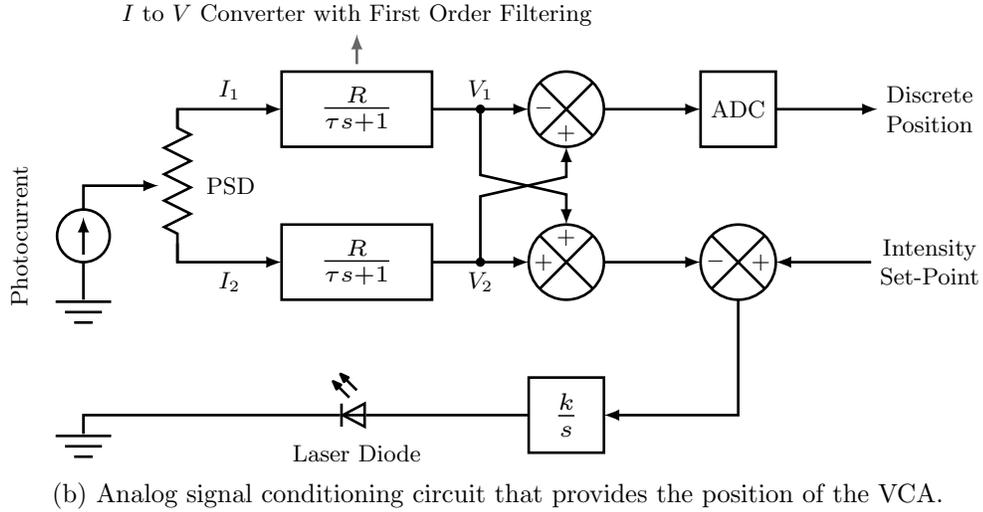
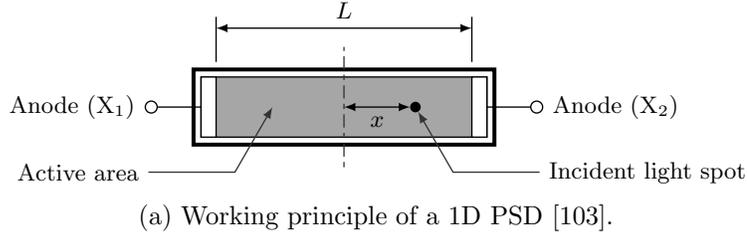
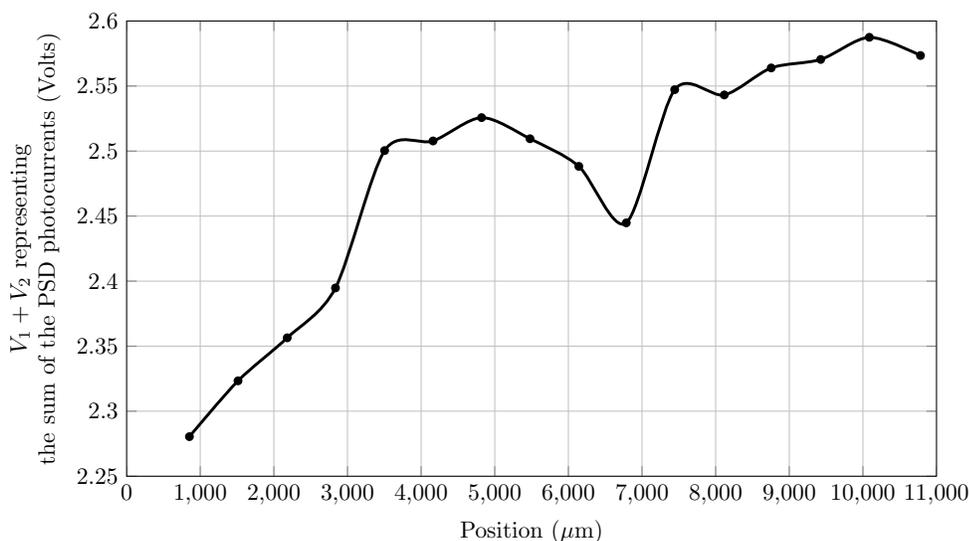


Figure 3.4: Position sensing mechanism of the proposed VCA design.

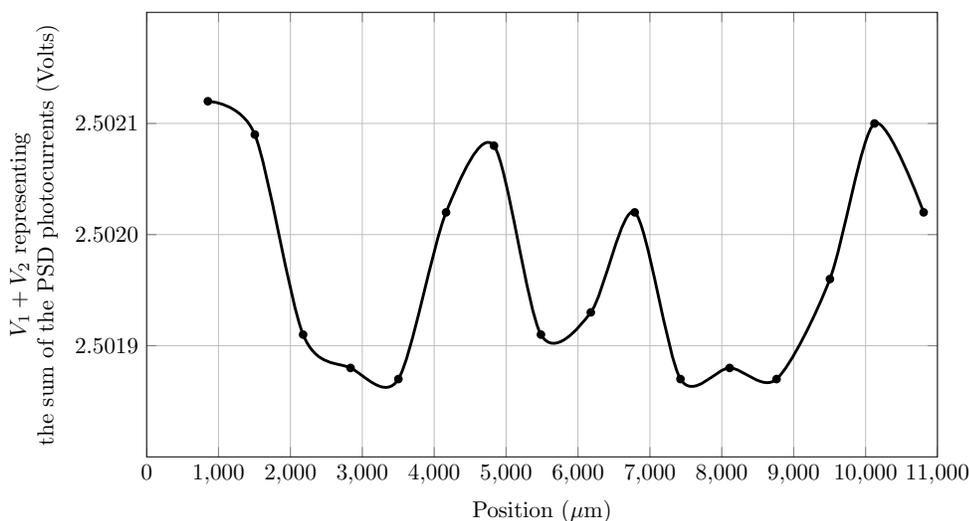
photocurrents collected at the two anodes (i.e., $I_1 + I_2$) is proportional to the intensity of the incident light spot. If the intensity observed by the PSD remains constant, (3.1) reduces to,

$$x = K(I_2 - I_1), \quad (3.2)$$

where, K is proportionality constant estimated by a sensor calibration exercise. Thus, the division operation in (3.1) can be avoided by actively controlling the current through the laser diode so that the light energy received by the PSD remains constant. Correspondingly, (3.2) is implemented using an analog circuit represented by a block diagram illustrated in Fig. 3.4(b). In this circuit, the photocurrents I_1 and I_2 collected at the two anodes are converted to corresponding voltage signals V_1 and V_2 . The differential signal ($V_2 - V_1$) is then fed to a 16 bit analog to digital converter (ADC)



(a) Open loop observations.



(b) Closed loop observations.

Figure 3.5: Performance of the integral controller in the position sensing circuit.

to provide a discrete quantification of the position. To this end, an ADC chip with a high sampling rate of 100 KHz was chosen so that the chip does not impose a bottleneck for the system performance. When the laser diode draws a fixed current, the intensity of the light incident on the PSD is a function of the normal distance between the laser diode and the PSD. Only a geometrically *perfect* assembly can

ensure that this distance strictly remains constant over the entire stroke, which is not practically achievable. As an alternative, an integral controller [see Fig. 3.4(b)] was incorporated into the position sensing circuit. By regulating the voltage across the laser diode, this controller maintains $(I_1 + I_2)$ at a constant level. A benchmarking study was conducted in order to evaluate the performance of the integral controller. This experimental study involved the open-loop and the closed-loop responses of the PSD, which were recorded as functions of the actuator position. The sum of the two photocurrents I_1 and I_2 represents the response of the PSD. In the open-loop trial, the laser diode was powered by a constant current. The experimental data presented in Fig. 3.5 shows that the open-loop response exhibits unfavorable variability over the actuator stroke. In contrast, the recorded closed-loop response in Fig. 3.5 demonstrates that the integral controller performs well. In specific terms, the standard deviation of the recorded 16 samples of the closed-loop response was observed to be $92.55 \mu\text{V}$.

The position feedback system presented above offers two major improvements over the previous design in [63] involving the sensor dynamics and the integral controller. The time constant of the analog filter in Fig. 3.4(b) was reduced by a factor of three in the current design so that the dynamics of the position sensing circuit becomes more conducive to the high acceleration of the VCA. It was experimentally confirmed that the reduced time constant did not adversely affect the noise margin of the feedback system. In addition, the set-point of the integral controller was changed in this design so that the laser diode can be operated within a linear region in its current-voltage characteristic curve. As a result, the integral controller performs better because the behavior of the laser diode is more uniform in this operational region.

Since each of the active regions in the stationary magnetic field is 1 inch in length [Fig. 3.1(a)] and the square trace pattern in the coil is 0.5 inch wide (Fig. 3.2), the proposed VCA design can provide a stroke of 0.5 inch without the requirement of

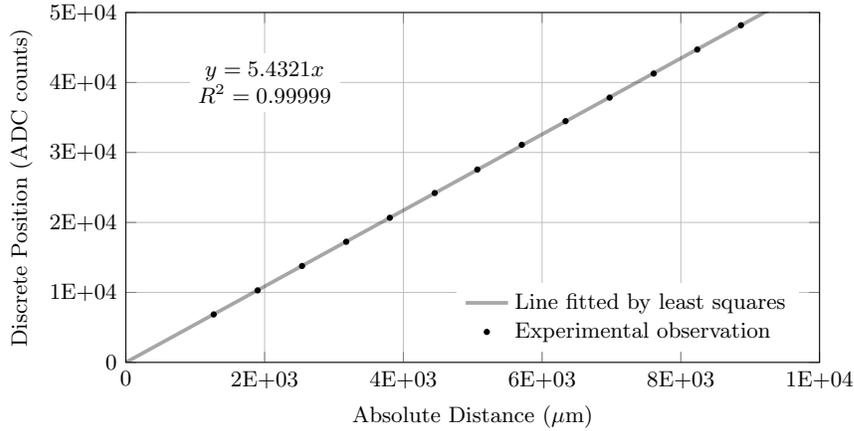
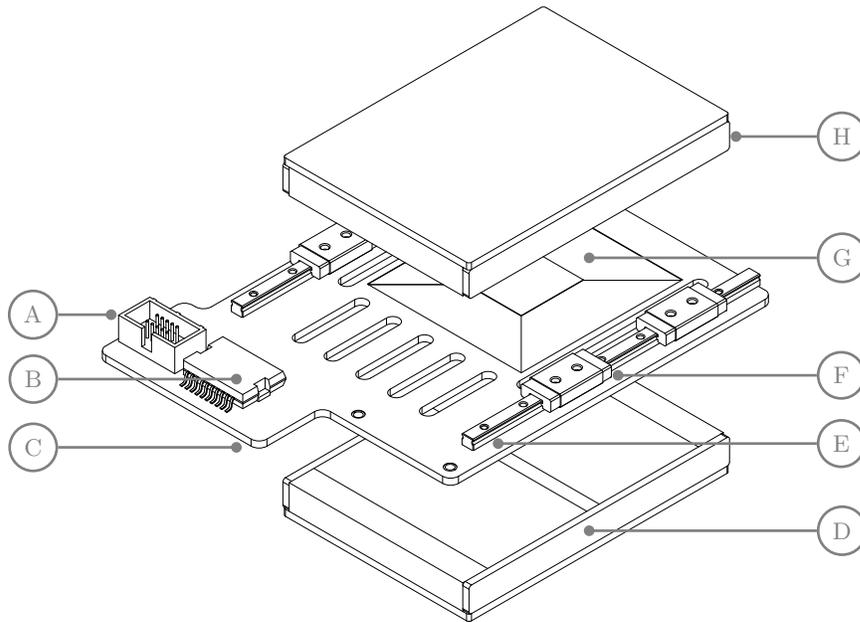
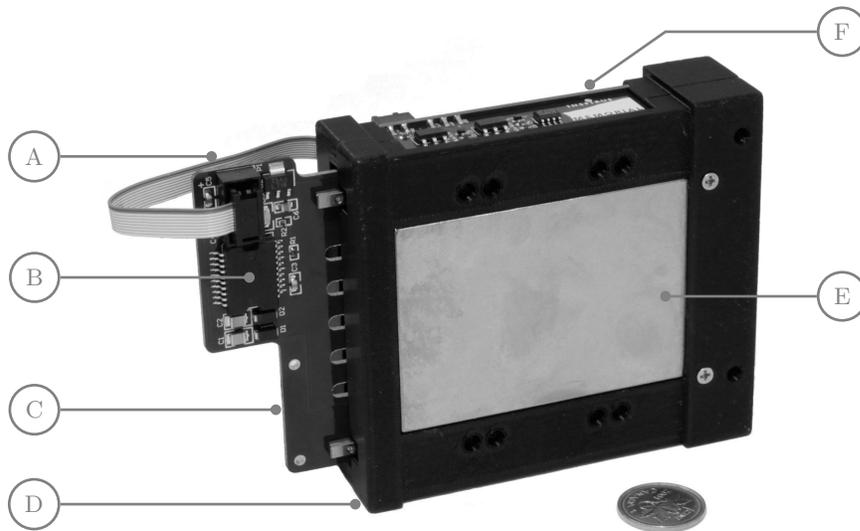


Figure 3.6: Output of the position feedback circuit against actual displacement of the coil.

commutation. Correspondingly, the chosen PSD has an active area of 12 mm (≈ 0.5 inch), which provides a theoretical resolution of $0.1831 \mu\text{m}$ [i.e., $12 \text{ mm}/(2^{16}-1)$]. The laser diode is installed on the PCB coil and the signal conditioning circuit is installed on the stator of the actuator. A prototype of the position feedback circuit was calibrated with reference to a sub-micron precision linear gauge. In this calibration exercise, the discrete position provided by the feedback circuit was recorded at different positions of the coil. The linear gauge provided the corresponding absolute positions. The data thus obtained was then fitted to a linear model to calculate the gain of the position sensor (Fig. 3.6). Specifically, the gain was estimated to be 5.4321 ADC counts per micron displacement which corresponds to a resolution is $0.1841 \mu\text{m}$. At each position of the coil, the output of the position sensor was sampled 2047 times (size of the designed FIFO of the data acquisition system) to minimize measurement error. For each data point, the standard deviation of all sampled data was also recorded to assess the statistical measurement error. In the worst case scenario, the recorded standard deviation for 2047 samples was 2.7219 ADC counts, which corresponds to a measurement error of $\pm 1.0 \mu\text{m}$ with 95% C.I.

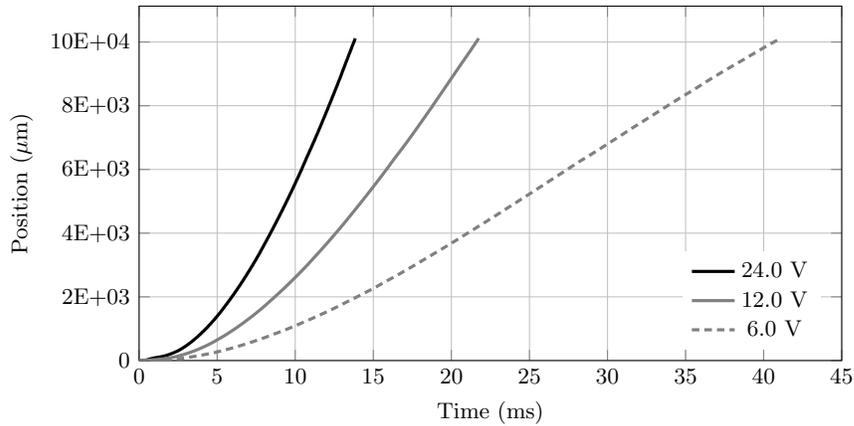


(a) Exploded view of the VCA assembly showing different components. A: Ribbon cable connector. B: H-bridge chip. C: Voice coil PCB. D: Bottom Halbach array. E: Linear bearing rail. F: Linear bearing carriage (part of the stator). G: Coil traces. H: Top Halbach array.

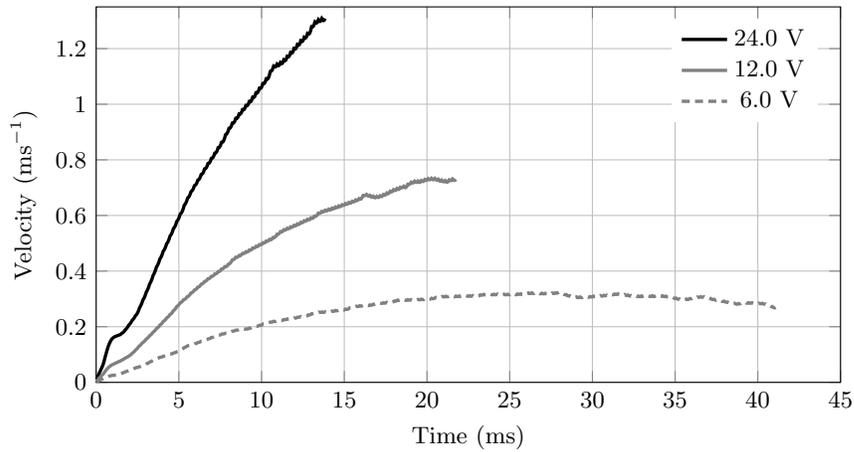


(b) A prototype of the proposed VCA design. A: Ribbon cable for control signals, data signals and power. B: H-bridge chip. C: voice coil PCB. D: Magnet holder rapid prototyped from thermo-plastics (stator of the VCA). E: Halbach array. F: Signal conditioning circuit for position sensing.

Figure 3.7: Detailed construction of the proposed VCA design.



(a) Observed position response.



(b) Estimated velocity.

Figure 3.8: Open-loop motion against gravity exhibited by a prototype of the proposed VCA design in response to step inputs of different magnitudes.

3.2.3 Electronic Drive Circuit

In a typical closed-loop application, a digital controller determines the input voltage to be supplied to the VCA to achieve or maintain a desired position. Subsequently, a PWM (pulse width modulation) signal representing the calculated control effort is dispatched to the VCA. Since the low voltage logic circuitry of the controller cannot provide the power necessary to drive the VCA, a DC-DC converter composed of a H-bridge IC was used to power the VCA based on the input provided by the PWM signal

from the controller. In order to obtain a compact design consisting of a self-contained drive system, the H-bridge chip was installed on the existing PCB coil [Fig. 3.7(a) and Fig. 3.7(b)]. A ribbon cable was used to transmit the PWM signals from the controller to the H-bridge chip. This cable also transmitted the electrical power to the VCA, and streamed out the position data provided by the signal conditioning circuit. Thus, the proposed design yielded a highly integrated VCA with a large actuation capacity in a compact form factor.

Table 3.1: General specifications of the proposed voice coil actuator design

Parameter	Value
Height (fully retracted)	113.97 mm
Stroke	12.00 mm (nominal)
Width	82.80 mm
Depth	25.40 mm
Total mass	520 g
Moving mass	33 g
Force constant	6.9478 N/A
Back EMF constant	6.9478 V-s/m
Magnetic flux density	0.704 T (typical)
Coil resistance	20.44 Ω (typical)
Coil inductance	1.27 mH (typical)

3.3 Specifications & Open-loop Performance

The general specifications of the proposed VCA design is provided in Table 3.1. In order to quantify the open-loop performance of the VCA design, step inputs of different magnitudes were provided and the resultant motion was recorded (Fig. 3.8). It should be noted that the moving mass of the VCA in this exercise was accelerated against gravity. Although the proposed VCA is designed to provide a stroke of 12 mm, a slightly shorter stroke is obtainable from a physical prototype because of some practical reasons. Due to the unavoidable manufacturing tolerances, it cannot be guaranteed

that the laser diode mounted on the PCB coil aligns with the mid-position of the PSD at the mid-stroke of the actuator. This misalignment results in two unequal half strokes measured from the mid-position of the PSD in both directions. Therefore, it is possible that the laser diode may travel beyond the active area of the PSD. Consequently, the position signal provided by the PSD will no longer be reliable in such a case. In order to avoid this occurrence, it is advisable to utilize a maximum of 5 mm displacement from the mid-position of the PSD in both directions, which provides a total stroke of 10 mm.

3.4 Conclusion

This chapter details the design of a low inertia, compact voice coil actuator that can be employed as a turn-key solution in applications requiring fast and accurate linear positioning. The experimental results show that a maximum velocity of 1.3 m/s can be achieved despite using a remotely deployable, low-voltage power source. Furthermore, the constructed prototype provides a large actuation capacity, which is quantified by its high force constant (≈ 7 N/A). Thus, the corresponding ground up design approach can be regarded successful in addressing the requirements of an optomechatronic application. However, a few ideas can be explored to further improve the current design. For example, a highly precise PCB prototyping technique can be employed to decrease the coil resistance, which will ultimately enhance the thermo-electric characteristics of the designed actuator.

Chapter 4

Dimensional Synthesis of Parallel Orientation Manipulators

Dimensional synthesis of a given POM architecture is an optimization problem where the objective function is composed of certain kinematic performance metrics that encapsulate application-specific requirements. Additional constraints (e.g., choice of an actuator) limit the parameter space and thus force dimensional synthesis to find a local optimum that is consistent with all design requirements. The volume and the dexterity of the workspace characterize the kinematic performance of an orientation manipulator requiring a small form factor. In this chapter, the optimum geometries of two prospective POM architectures are synthesized through the application of the efficient and statistically robust Response Surface Methodology (RSM). To this end, the direct and the inverse kinematics of both architectures were solved in order to estimate their kinematic performances. An iterative technique is employed to solve the direct kinematics problem of both architectures. The optimization procedure presented in this chapter begins with an arbitrarily chosen initial parameter space. A hybrid approach consisting of a space-filling and an IV-Optimal (integrated variance) experiment design is employed in order to reduce the initial search space and to find appropriate regression models that adequately fit the objective function. Subsequently, the empirical models thus determined are employed to find an optimum parameter set that maximizes the objective function. This solution approach efficiently identifies the optimal manipulators for both architectures.

4.1 Introduction

Precision manufacturing can eliminate many mechanical deficiencies that may occur in the physical prototypes of parallel mechanisms; examples include joint backlash, parasitic motion, etc. Nonetheless, even an excellently manufactured prototype will not be able to perform in an accurate manner if the underlying geometry induces poor kinematic characteristics. The locations of the joints and the lengths of the links characterize the geometry of a PKM. Thus, the geometry determines the inertial load and the motion transmission characteristics of a manipulator. A low inertial load with favorable motion transmission quality ensures that the limited actuation capacity is utilized to its full potential in order to achieve high speed robotic maneuvers. Furthermore, the controllability of a PKM can suffer if its geometry introduces a high degree of coupling among the multiple kinematic chains. Thus, it can be inferred that the kinematic and the dynamic performance of a manipulator are both functions of the PKM geometry. An ideal geometric optimization, therefore, accounts for both of these aspects of manipulator performance. However, practical realization of such a proposition is difficult, if not impossible. A feasible alternative is to determine the geometry of the manipulator for optimal kinematic performance. Since a PKM is inherently conducive to high speed manipulation due to the presence of multiple kinematic chains in its structure, aiming for optimal kinematic performance in a dimensional synthesis exercise is justified. It is relevant to mention that the volume of the workspace, a universally important kinematic performance measure, is generally smaller for a PKM, while its dynamic performance is typically better than what a similar serial architecture can provide.

Although workspace volume has been frequently employed as a performance metric in many dimensional synthesis methodologies reported in the literature [50–54], fo-

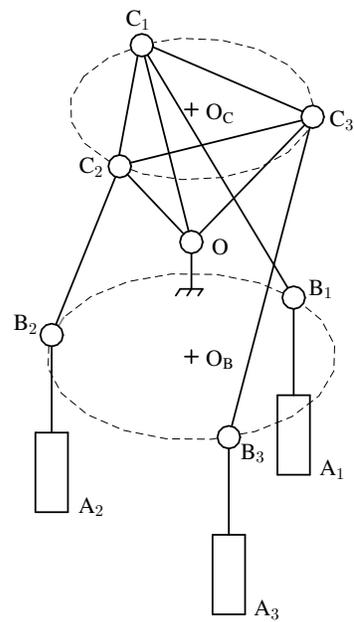
cusing solely on maximizing its size can be a catastrophically myopic approach, since desirable kinematic features cannot be guaranteed at each point of the workspace regardless of its size. In addition to the workspace volume, other kinematic performance metrics that are cited in the literature include dexterity [21, 55], stiffness [57], manipulation accuracy [56], etc. Despite such a large variety, the objective function in a geometric optimization exercise is comprised of only those performance features that are relevant to the desired application. For example, high stiffness is an important design requirement for manipulators that handle large payloads, while it is not as significant for small-scale manipulators. Besides specific kinematic performance requirements, additional design constraints further restrict the search space of a dimensional synthesis problem. For instance, if a particular actuator is selected because of the speed or the size requirements of an application, the manipulator geometries that can accommodate this design decision constitute the appropriate search space. Therefore, a practice-oriented dimensional synthesis exercise cannot be approached as a global optimization problem. Rather, its goal is to determine a locally optimum geometry that is consistent with all application-specific requirements. A dimensional synthesis exercise, therefore, can be characterized as a multi-objective local optimization problem wherein the workspace of a robotic manipulator is maximized without compromising its application relevant kinematic performance characteristics.

After performing a qualitative analysis on the feasibility for the desired application, the 3-PSS/S and the 3-SPS/S manipulators were selected in Chapter 2 for further design study. In this chapter, these two POM structures are dimensionally optimized for kinematic performance. To this end, the kinematic analysis of the candidate POMs are presented in Sections 4.3 and 4.4. Subsequently, the parameter spaces of the corresponding dimensional synthesis problems are defined in Section 4.5. In Section 4.6, appropriate parameterizations are selected for a limited number of kinematic perfor-

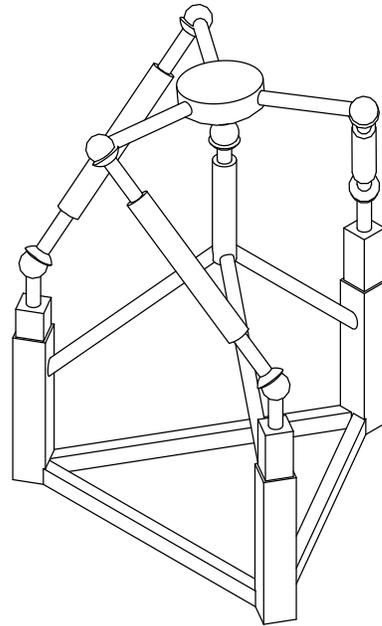
mance characteristics that are determined to be relevant to the desired application. The corresponding estimation methods that quantify these performance metrics are described in Section 4.7. Next, Section 4.8 employs response surface methodology in order to synthesize the optimal dimensions of the candidate architectures. The kinematic performances of the optimal manipulators are then analyzed in Section 4.9. In addition, the estimation accuracy of the selected performance metrics is evaluated in Section 4.10. Finally, the concluding remarks are offered in Section 4.11.

4.2 Kinematic Structures of the Candidate POM Architectures

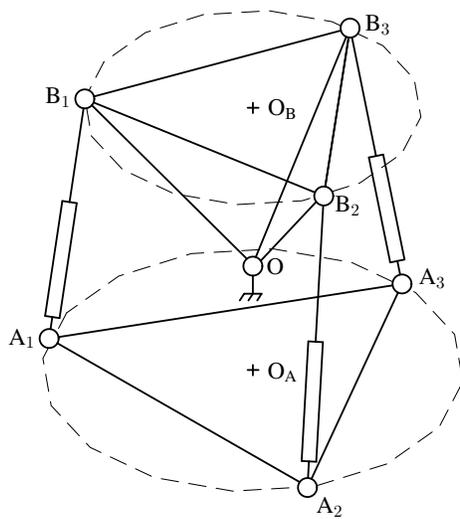
The kinematic structures of the candidate POM architectures are shown in Fig. 4.1. The 3-PSS/S limb configuration implies that the first link of each of the three limbs is an actuated prismatic joint A_iB_i [see Fig. 4.1(a)]. The intermediate link B_iC_i is connected to the moving platform and the piston of the prismatic actuator by two spherical joints at the two end points C_i and B_i , respectively. An additional spherical joint at the mechanism center O ensures that the moving platform $C_1C_2C_3$ is capable of spherical motion only. On the other hand, the 3-SPS/S architecture is comprised of three extensible limbs A_iB_i (i.e., prismatic actuators) that are connected to the moving platform $B_1B_2B_3$ by three spherical joints located at points B_i [see Fig. 4.1(c)]. In addition, three other spherical joints located at points A_i constraint the limbs to the mechanical ground. Another spherical joint at point O allows the moving platform to possess only three degrees of rotational freedom.



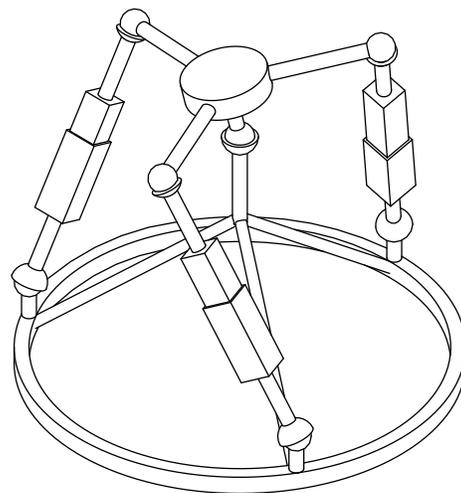
(a) Truss model of the 3-PSS/S manipulator.



(b) CAD model: 3-PSS/S.



(c) Truss model of the 3-SPS/S manipulator.



(d) CAD model: 3-SPS/S.

Figure 4.1: Kinematic structures of the candidate orientation manipulators.

4.3 Kinematic Analysis of 3-PSS/S Manipulators

A brief discussion on the inverse kinematics of the 3-PSS/S architecture can be found in [8]. The direct kinematics problem is solved iteratively in [21]. In addition, the differential kinematics of an equivalent architecture characterized by a 3-PUS/S limb configuration is provided in [27, pp. 159-160]. Despite the aforementioned literature, a detailed kinematic analysis of the 3-PSS/S manipulator is provided hereafter for the sake of completeness.

Two Cartesian coordinate frames A (the global reference frame, G-RF) and C (the local reference frame, L-RF) are introduced in order to facilitate the formulation of the kinematic analysis. The origins of both coordinate frames are located at point O (i.e., the mechanism center). The L-RF is attached rigidly to the moving platform. The G-RF is fixed to the mechanical ground. Since the position vectors of the articulation points (i.e., terminal points of the links) can be defined with respect to either coordinate frame, it is necessary to unambiguously specify the coordinate frame in which a vector is referenced. To this end, a preceding superscript is used only when a vector is expressed in the L-RF; e.g., ${}^x\mathbf{a}$ denotes the vector \mathbf{a} expressed in the X coordinate frame. When a vector is expressed with respect to the G-RF, this superscript is omitted. The rotation matrix that defines the relative orientation of the Y coordinate frame with respect to the X coordinate frame is denoted by ${}^x\mathbf{R}_y$. If a vector is denoted by ${}^x\mathbf{a}$ and ${}^y\mathbf{a}$ with respect to two separate coordinate frames, the rotation matrix ${}^x\mathbf{R}_y$ provides the following transformation: ${}^x\mathbf{a} = {}^x\mathbf{R}_y \times {}^y\mathbf{a}$. Unless otherwise specified, this convention of specifying vectors and the corresponding reference frames will be used in the remainder of this thesis. Let the relative orientation of the L-RF with respect to the G-RF be provided by the 3×3 rotation matrix ${}^a\mathbf{R}_c$. The vector $\overline{B_i C_i}$ is denoted by \mathbf{x}_i with respect to the G-RF. In reference to Fig. 4.1(a),

let the position vectors of the points A_i , B_i and C_i in the G-RF be represented by \mathbf{a}_i , \mathbf{b}_i , and \mathbf{c}_i . Further, let $\|\mathbf{a}_i\| = a_i$, $\|\mathbf{b}_i\| = b_i$, $\|\mathbf{c}_i\| = \|\mathbf{c}_i\| = c_i$, $\|\overline{A_i B_i}\| = d_i$, and $\|\mathbf{x}_i\| = x_i$. The unit direction vector along $\overline{A_i B_i}$ is denoted by $\hat{\mathbf{n}}_i$.

4.3.1 Inverse Kinematics

The position vector of the articulation point B_i can be written as,

$$\mathbf{b}_i = \mathbf{a}_i + d_i \hat{\mathbf{n}}_i. \quad (4.1)$$

In addition, the loop-closure equation for the i -th limb can be written as,

$$\mathbf{x}_i = \mathbf{c}_i - \mathbf{b}_i. \quad (4.2)$$

The magnitudes of the vectors in (4.1) and (4.2) can be found by dot multiplying each vector with itself, as shown in the following two equations.

$$b_i^2 = a_i^2 + d_i^2 + 2d_i a_i^T \hat{\mathbf{n}}_i \quad (4.3)$$

$$x_i^2 = c_i^2 + b_i^2 - 2\mathbf{c}_i^T \mathbf{b}_i \quad (4.4)$$

After substituting (4.1) and (4.3) into (4.4), algebraic simplification provides,

$$d_i^2 + d_i (2a_i^T \hat{\mathbf{n}}_i - 2\mathbf{c}_i^T \hat{\mathbf{n}}_i) + (c_i^2 + a_i^2 - x_i^2 - 2\mathbf{c}_i^T \mathbf{a}_i) = 0. \quad (4.5)$$

Let $p_i = 2a_i^T \hat{\mathbf{n}}_i - 2\mathbf{c}_i^T \hat{\mathbf{n}}_i$ and $q_i = c_i^2 + a_i^2 - x_i^2 - 2\mathbf{c}_i^T \mathbf{a}_i$. Substituting these expressions into (4.5) results in a quadratic equation as in (4.6).

$$d_i^2 + p_i d_i + q_i = 0 \quad (4.6)$$

Writing (4.6) for $i = 1, 2$ and 3 produces three equations that describe the kinematic constraints on the moving platform. In an inverse kinematics problem, the prismatic link lengths d_i are determined provided that the orientation of the moving platform is known. Correspondingly, the position vectors \mathbf{c}_i can be determined by applying a simple coordinate transformation. From the geometry of the manipulator, the vectors \mathbf{a}_i , $\hat{\mathbf{n}}_i$ and the scalar quantities x_i^2 , a_i^2 and c_i^2 can be determined. Subsequently, solving the quadratic equation in (4.6) yields two general solutions for each actuated joint. However, only one of these two solutions provides a physically achievable geometric configuration.

$$d_i = \frac{-p_i \pm \sqrt{p_i^2 - 4q_i}}{2} \quad (4.7)$$

4.3.2 Direct Kinematics

A closed-form analysis of the direct kinematics problem of the 3-PSS/S manipulator was not found. Alternatively, an iterative solution was formulated in [21]. However, this solution degenerates when all four spherical joints on the moving platform are coplanar. In order to address this limitation, an improved solution scheme that does not degenerate for any geometric configuration of the manipulator is presented hereafter. The proposed solution is computed iteratively from a system of four nonlinear equations in an efficient manner, which contrasts with the nine nonlinear equations in the iterative solution provided in [21].

In order to formulate a suitable kinematic constraint, (4.2) is dot multiplied with itself and subsequently rearranged to obtain,

$$\mathbf{c}_i^T \mathbf{b}_i - \gamma_i = 0, \quad (4.8)$$

where $\gamma_i = \frac{c_i^2 + b_i^2 - x_i^2}{2}$. In a direct kinematics problem, \mathbf{b}_i is known and the vector ${}^c\mathbf{c}_i$

is provided by the geometry of the moving platform. Substituting $\mathbf{c}_i = {}^a\mathbf{R}_c \times {}^c\mathbf{c}_i$ in (4.8) yields,

$$f_i := ({}^a\mathbf{R}_c \times {}^c\mathbf{c}_i)^T \mathbf{b}_i - \gamma_i = 0. \quad (4.9)$$

A direct kinematics problem determines the rotation matrix ${}^a\mathbf{R}_c$ that satisfies (4.9).

A minimum of three parameters are sufficient to represent 3D rotations. Consequently, the nine elements of a rotation matrix are subjected to six orthonormality constraints to account for the overparameterization. Directly estimating the nine elements in a nonlinear search requires these constraints to be maintained explicitly, which is difficult to achieve [104]. Since no overparameterization constraints are involved with the non-redundant representations (e.g., Euler angles), they offer an attractive alternative. However, it was shown that no three dimensional parameterization can be both global and nonsingular [105]; i.e., every rotation determines some finite values of the parameters but these values are not uniquely defined. Therefore, a nonsingular representation of 3D orientation must be defined by more than three parameters. The *unit quaternion* [104, 106–109] is such a nonsingular representation that is also minimally redundant because of its four parameter composition. Correspondingly, formulating the problem of relative orientation in terms of unit quaternions is particularly convenient, since it needs to account for only a single constraint arising from the parameterization redundancy (e.g., [109–111]). For a unit quaternion $\mathbf{q} = [d \ a \ b \ c]^T$, this constraint enforces $\|\mathbf{q}\| = 1$ (i.e., $d^2 + a^2 + b^2 + c^2 = 1$), which can be rewritten as,

$$f_c := K(d^2 + a^2 + b^2 + c^2 - 1) = 0. \quad (4.10)$$

The arbitrary gain $K \gg 1$ in (4.10) ensures that a nonlinear search imposes greater emphasis on the unit quaternion constraint. An inaccurate solution may be found if this constraint is not maintained strictly.

Three nonlinear equations are obtained by writing (4.9) for $i = 1, 2,$ and 3 . When these three equations are grouped with the unit quaternion constraint in (4.10), a system of equations as in (4.11) is constructed that involves the four elements of a unit quaternion as the unknown parameter vector.

$$F := \begin{bmatrix} ({}^a\mathbf{R}_c \times {}^c\mathbf{c}_1)^T \mathbf{b}_1 - \gamma_1 \\ ({}^a\mathbf{R}_c \times {}^c\mathbf{c}_2)^T \mathbf{b}_2 - \gamma_2 \\ ({}^a\mathbf{R}_c \times {}^c\mathbf{c}_3)^T \mathbf{b}_3 - \gamma_3 \\ K(d^2 + a^2 + b^2 + c^2 - 1) \end{bmatrix} = \mathbf{0}. \quad (4.11)$$

Solving the system of nonlinear equations in (4.11) for the unknown parameter vector $[d \ a \ b \ c]^T$ by an iterative technique (e.g., nonlinear least squares) provides the relative orientation of the moving platform. In order to algebraically determine the corresponding Jacobian matrix, the following mathematical discourse is presented.

The rotation matrix corresponding to a general unit quaternion $\mathbf{q} = [d \ a \ b \ c]^T$ is provided by,

$${}^x\mathbf{R}_y = \begin{bmatrix} d^2 + a^2 - b^2 - c^2 & 2ab - 2cd & 2ca + 2bd \\ 2ab + 2cd & d^2 - a^2 + b^2 - c^2 & 2bc - 2ad \\ 2ca - 2bd & 2bc + 2ad & d^2 - a^2 - b^2 + c^2 \end{bmatrix}. \quad (4.12)$$

Let $\mathbf{a}_i = [a_{x_i} \ a_{y_i} \ a_{z_i}]^T$, $\mathbf{b}_i = [b_{x_i} \ b_{y_i} \ b_{z_i}]^T$, and $\mathbf{c}_i = [c_{x_i} \ c_{y_i} \ c_{z_i}]^T$. With respect to the L-RF, these vectors are denoted by ${}^c\mathbf{a}_i$, ${}^c\mathbf{b}_i$ and ${}^c\mathbf{c}_i$ respectively. By substituting (4.12) into (4.9) and letting ${}^c\mathbf{c}_i = [c_{cx_i} \ c_{cy_i} \ c_{cz_i}]^T$, an expanded expression for (4.9) can be

obtained, which is provided by,

$$\begin{aligned}
f_i := & ((d^2 + a^2 - b^2 - c^2)c_{cx_i} + (2ab - 2cd)c_{cy_i} + (2ca + 2bd)c_{cz_i}) b_{x_i} \\
& + ((2ab + 2cd)c_{cx_i} + (d^2 - a^2 + b^2 - c^2)c_{cy_i} + (2bc - 2ad)c_{cz_i}) b_{y_i} \\
& + ((2ca - 2bd)c_{cx_i} + (2bc + 2ad)c_{cy_i} + (d^2 - a^2 - b^2 + c^2)c_{cz_i}) b_{z_i} - \gamma_i = 0. \quad (4.13)
\end{aligned}$$

For the unknown parameter vector $\mathbf{q} = [d \ a \ b \ c]^T$, the Jacobian matrix of the system of equations in (4.11) is formulated as,

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial d} & \frac{\partial f_1}{\partial a} & \frac{\partial f_1}{\partial b} & \frac{\partial f_1}{\partial c} \\ \frac{\partial f_2}{\partial d} & \frac{\partial f_2}{\partial a} & \frac{\partial f_2}{\partial b} & \frac{\partial f_2}{\partial c} \\ \frac{\partial f_3}{\partial d} & \frac{\partial f_3}{\partial a} & \frac{\partial f_3}{\partial b} & \frac{\partial f_3}{\partial c} \\ \frac{\partial f_c}{\partial d} & \frac{\partial f_c}{\partial a} & \frac{\partial f_c}{\partial b} & \frac{\partial f_c}{\partial c} \end{bmatrix}. \quad (4.14)$$

The partial derivatives of (4.13) with respect to the unknown parameter vector $\mathbf{q} = [d \ a \ b \ c]^T$ provide the general expressions of the elements in the first three rows of the Jacobian matrix as shown in (4.15), (4.16), (4.17), and (4.18).

$$\begin{aligned}
\frac{\partial f_i}{\partial d} = & (2dc_{cx_i} - 2cc_{cy_i} + 2bc_{cz_i})b_{x_i} + (2cc_{cx_i} + 2dc_{cy_i} - 2ac_{cz_i})b_{y_i} \\
& + (-2bc_{cx_i} + 2ac_{cy_i} + 2dc_{cz_i})b_{z_i} \quad (4.15)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial f_i}{\partial a} = & (2ac_{cx_i} + 2bc_{cy_i} + 2cc_{cz_i})b_{x_i} + (2bc_{cx_i} - 2ac_{cy_i} - 2dc_{cz_i})b_{y_i} \\
& + (2cc_{cx_i} + 2dc_{cy_i} - 2ac_{cz_i})b_{z_i} \quad (4.16)
\end{aligned}$$

$$\begin{aligned} \frac{\partial f_i}{\partial b} = & (-2bc_{cx_i} + 2ac_{cy_i} + 2dc_{cz_i})b_{x_i} + (2ac_{cx_i} + 2bc_{cy_i} + 2cc_{cz_i})b_{y_i} \\ & + (-2dc_{cx_i} + 2cc_{cy_i} - 2bc_{cz_i})b_{z_i} \end{aligned} \quad (4.17)$$

$$\begin{aligned} \frac{\partial f_i}{\partial c} = & (-2cc_{cx_i} - 2dc_{cy_i} + 2ac_{cz_i})b_{x_i} + (2dc_{cx_i} - 2cc_{cy_i} + 2bc_{cz_i})b_{y_i} \\ & + (2ac_{cx_i} + 2bc_{cy_i} + 2cc_{cz_i})b_{z_i} \end{aligned} \quad (4.18)$$

The fourth row of the Jacobian matrix is provided by,

$$\frac{\partial f_c}{\partial d} = 2Kd, \quad \frac{\partial f_c}{\partial a} = 2Ka, \quad \frac{\partial f_c}{\partial b} = 2Kb, \quad \frac{\partial f_c}{\partial c} = 2Kc. \quad (4.19)$$

The Jacobian matrix can be numerically evaluated by employing the above equations. Subsequently, a solution to the system of equations in (4.11) can be obtained iteratively under a nonlinear least squares scheme. Since the parameters thus estimated do not represent a *stricto sensu* unit quaternion, the corresponding rotation matrix is not guaranteed to be orthonormal. Although normalizing the estimated quaternion provides a simple way to deal with this discrepancy, the significance of such an operation in reference to 3D rotations is not clear. Alternatively, a geometrically elaborate technique described in Appendix A is employed so that a near-unit quaternion can be converted to the closest orthonormal rotation matrix.

4.3.3 Inverse Kinematic Jacobian

Let the angular velocity of the intermediate link B_iC_i of the i -th limb and the moving platform be respectively denoted by $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_c$. The linear velocity \mathbf{v}_{ci} of the point C_i can be expressed either in terms of the angular velocity of the moving platform as in

(4.20) or in terms of the angular velocity of the intermediate link as in (4.21).

$$\mathbf{v}_{ci} = \boldsymbol{\omega}_c \times \mathbf{c}_i \quad (4.20)$$

$$\mathbf{v}_{ci} = \dot{d}_i \hat{\mathbf{n}}_i + \boldsymbol{\omega}_i \times \mathbf{x}_i \quad (4.21)$$

In (4.21), the elongation rate of the linear actuator is denoted by \dot{d}_i . Substituting (4.20) into (4.21) and dot multiplying both sides by \mathbf{x}_i yields,

$$(\mathbf{c}_i \times \mathbf{x}_i)^T \boldsymbol{\omega}_c = (\hat{\mathbf{n}}_i \cdot \mathbf{x}_i) \dot{d}_i. \quad (4.22)$$

Writing (4.22) three times for $i = 1, 2$ and 3 yields three scalar equations that can be arranged in a matrix form:

$$\mathbf{J}_{xx} \boldsymbol{\omega}_c = \mathbf{J}_{qx} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix}^T. \quad (4.23)$$

Here,

$$\mathbf{J}_{xx} = \begin{bmatrix} (\mathbf{c}_1 \times \mathbf{x}_1)^T \\ (\mathbf{c}_2 \times \mathbf{x}_2)^T \\ (\mathbf{c}_3 \times \mathbf{x}_3)^T \end{bmatrix}, \quad \mathbf{J}_{qx} = \begin{bmatrix} (\hat{\mathbf{n}}_1 \cdot \mathbf{x}_1) & 0 & 0 \\ 0 & (\hat{\mathbf{n}}_2 \cdot \mathbf{x}_2) & 0 \\ 0 & 0 & (\hat{\mathbf{n}}_3 \cdot \mathbf{x}_3) \end{bmatrix}. \quad (4.24)$$

The corresponding inverse kinematic Jacobian is defined by $\mathbf{J}_x = \mathbf{J}_{qx}^{-1} \mathbf{J}_{xx}$, which provides input link velocities in terms of the moving platform angular velocity; i.e., $[\dot{d}_1 \quad \dot{d}_2 \quad \dot{d}_3]^T = \mathbf{J}_x \boldsymbol{\omega}_c$.

4.3.4 Kinematic Singularities

The 3-PSS/S architecture is at a *singular configuration* when either of the matrices \mathbf{J}_{xx} or \mathbf{J}_{qx} are singular. Since \mathbf{J}_{qx} is a diagonal matrix, it is singular only when at least one of the diagonal entries is zero; i.e., $\hat{\mathbf{n}}_i \cdot \mathbf{x}_i = 0$. Therefore, when the intermediate link B_iC_i is normal to the axis of the prismatic joint A_iB_i , the manipulator is in an inverse kinematic singular configuration. When \mathbf{J}_{qx} is singular and its null space is not empty, there exists some non-zero \dot{d}_i for which $\boldsymbol{\omega}_c$ is zero. This signifies that certain infinitesimal motion of the moving platform at this singular configuration (i.e., $A_iB_i \perp B_iC_i$) cannot be achieved despite applying actuation efforts.

When \mathbf{J}_{xx} is singular and its null space is not empty, there exists some non-zero $\boldsymbol{\omega}_c$ that yields zero \dot{d}_i ; i.e., despite the absence of motion in the actuators, the moving platform of the mechanism can possess infinitesimal motion in some directions. In other words, the manipulator cannot resist forces or moments in certain limb configurations that are identified in the following cases:

- Case 1: One of the three vectors $(\mathbf{c}_i \times \mathbf{x}_i)^T$ vanishes; i.e., points B_i , C_i and O are collinear.
- Case 2: Two of the three vectors $(\mathbf{c}_i \times \mathbf{x}_i)^T$ are linearly dependent; i.e., $\triangle B_iOC_i$ and $\triangle B_jOC_j$ are coplanar with $i \neq j$.
- Case 3: The three vectors $(\mathbf{c}_i \times \mathbf{x}_i)^T$ are linearly dependent; i.e., the planes containing $\triangle B_1OC_1$, $\triangle B_2OC_2$ and $\triangle B_3OC_3$ intersect in a common line.

4.4 Kinematic Analysis of 3-SPS/S Manipulators

Closed-form solutions to the inverse and the direct kinematics problem of 3-SPS/S manipulators were formulated by Tsai in [24, pp. 129-134]. After presenting a detailed Jacobian analysis, the kinematic singularities of this POM were identified in [24, pp. 231-234]. In order to avoid redundancy, this chapter provides only a brief review of the inverse and the differential kinematics. However, an alternative solution to the direct kinematics problem is presented in Section 4.4.2.

For the purpose of analysis, two coordinate frames A (G-RF) and B (L-RF) are introduced and they are fixed to the ground and the moving platform respectively. In addition, the origins of both frames coincide at the mechanism center O [see Fig. 4.1(c)]. Let the position vectors of the points A_i and B_i be denoted by ${}^b\mathbf{a}_i$ and ${}^b\mathbf{b}_i$ with respect to the L-RF. In addition, these vectors are denoted by \mathbf{a}_i and \mathbf{b}_i with respect to the G-RF. The unit direction vector $\hat{\mathbf{x}}_i$ is expressed in the G-RF and points along $\overline{A_iB_i}$. Let $\overline{A_iB_i} = d_i\hat{\mathbf{x}}_i$, $\|\mathbf{a}_i\| = a_i$, and $\|{}^b\mathbf{b}_i\| = \|\mathbf{b}_i\| = b_i$. The rotation matrix ${}^a\mathbf{R}_b$ provides the relative orientation so that $\mathbf{b}_i = {}^a\mathbf{R}_b \times {}^b\mathbf{b}_i$.

4.4.1 Inverse Kinematics

In reference to Fig. 4.1(c), the loop-closure equation for the 3-SPS/S architecture can be written as,

$$\mathbf{b}_i - \mathbf{a}_i = d_i\hat{\mathbf{x}}_i. \quad (4.25)$$

Dot multiplying the above equation with itself yields,

$$d_i^2 = a_i^2 + b_i^2 - 2\mathbf{b}_i^T \mathbf{a}_i. \quad (4.26)$$

The vectors \mathbf{a}_i and ${}^b\mathbf{b}_i$ are provided by the manipulator geometry. In an inverse kinematics problem, the relative orientation (i.e., ${}^a\mathbf{R}_b$) and, by extension, \mathbf{b}_i are known and the coordinates of the actuated joints are to be determined. Between the two roots of the above quadratic equation, the positive one in (4.27) provides the solution to the inverse kinematics problem.

$$d_i = \pm \sqrt{a_i^2 + b_i^2 - 2\mathbf{b}_i^T \mathbf{a}_i} \quad (4.27)$$

4.4.2 Direct Kinematics

The relative orientation of the moving platform in a 3-SPS/S manipulator is solved in closed-form from a polynomial of degree eight [24, pp. 129-134]. Although each of the eight solutions of this polynomial is feasible in a mathematical sense, only a single orientation is relevant for a physical prototype, because real geometric constraints, such as motion range of the joints, link collisions etc., render the other solutions unachievable without the disassembly of the manipulator. Identifying this *relevant solution* requires further analysis. However, this section provides an alternative solution to the direct kinematics problem that iteratively obtains the relevant solution.

The kinematic constraint in (4.26) can be rearranged as,

$$\mathbf{b}_i^T \mathbf{a}_i - \gamma_i = 0, \quad (4.28)$$

where $\gamma_i = \frac{a_i^2 + b_i^2 - d_i^2}{2}$. In a direct kinematics problem, \mathbf{a}_i is known from the manipulator geometry and d_i is given. Substituting $\mathbf{b}_i = {}^a\mathbf{R}_b \times {}^b\mathbf{b}_i$ in (4.28) provides,

$$f_i := ({}^a\mathbf{R}_b \times {}^b\mathbf{b}_i)^T \mathbf{a}_i - \gamma_i = 0. \quad (4.29)$$

Similar to Section 4.3.2, three nonlinear equations can be obtained by writing (4.29) for $i = 1, 2,$ and 3 . If a unit quaternion $\mathbf{q} = [d \ a \ b \ c]^T$ is employed to determine the rotation matrix ${}^a\mathbf{R}_b$ as shown in (4.12), these three equations must be complimented by the unit quaternion constraint in (4.10) so that an exactly determined system of equations involving the unknown parameter vector $[d \ a \ b \ c]^T$ can be constructed as in,

$$F := \begin{bmatrix} ({}^a\mathbf{R}_b \times {}^b\mathbf{b}_1)^T \mathbf{a}_1 - \gamma_1 \\ ({}^a\mathbf{R}_b \times {}^b\mathbf{b}_2)^T \mathbf{a}_2 - \gamma_2 \\ ({}^a\mathbf{R}_b \times {}^b\mathbf{b}_3)^T \mathbf{a}_3 - \gamma_3 \\ K(d^2 + a^2 + b^2 + c^2 - 1) \end{bmatrix} = \mathbf{0}. \quad (4.30)$$

Since the system of nonlinear equations in (4.30) is algebraically equivalent to (4.11), its iterative solution can be determined by the Jacobian matrix presented in Section 4.3.2.

4.4.3 Differential Kinematics

The inverse kinematic Jacobian maps the angular velocity $\boldsymbol{\omega}_b$ of the moving platform to the elongation rates \dot{d}_i of the extensible limbs; i.e., $[\dot{d}_1 \ \dot{d}_2 \ \dot{d}_3]^T = \mathbf{J}_y \boldsymbol{\omega}_b$. The analytical expression of the inverse kinematic Jacobian \mathbf{J}_y was derived in [24, p. 232], as shown in the following equation.

$$\mathbf{J}_y = \begin{bmatrix} (\mathbf{b}_1 \times \hat{\mathbf{x}}_1)^T \\ (\mathbf{b}_2 \times \hat{\mathbf{x}}_2)^T \\ (\mathbf{b}_3 \times \hat{\mathbf{x}}_3)^T \end{bmatrix} \quad (4.31)$$

An analysis of \mathbf{J}_y provided in [24] identifies the kinematic singular configurations of the 3-SPS/S architecture.

4.5 Representation of Manipulator Geometry

A set of dimensional parameters define the lengths of the links and the locations of the joints of a manipulator. If the link lengths and the joint locations are chosen arbitrarily, a large number of kinematic parameters are required in order to sufficiently specify the geometry of a manipulator. Optimization over such a large dimension is impractical from a computational point of view. Hence, geometric constraints must be imposed in order to reduce this large set of kinematic parameters to a manageable set of *design kinematic parameters* (DKP) [48]. These geometric constraints often reflect practical design considerations such as symmetry of certain links, similarity of the actuators, coplanarity of certain joints, etc. Suitable geometric parametrization of a manipulator yields a set of DKPs that define the search space for the optimization problem.

4.5.1 Geometric Parameterization of 3-PSS/S Manipulators

The actuator in each kinematic chain of the 3-PSS/S architecture is fixed, while the intermediate link B_iC_i undergoes spatial motion [see Fig. 4.1(a)]. Consequently, the axis of the actuator and the adjacent intermediate link do not always coincide. When these two axes are aligned, the force generated by each actuator is entirely transmitted through the intermediate link to create motion in the moving platform. Otherwise, only a component of the actuation force that acts along the link B_iC_i is utilized. If this component is designed to have a large magnitude, the dynamics of the moving platform is likely to be improved. This can be achieved by a geometrical configuration where the fixed axis of each actuator is tilted towards the axis of the corresponding adjacent link. Thus, the angle between the two axes remains small over the reachable workspace.

However, the potential gain in the dynamics of the moving platform can also possibly be countervailed by an increase in the dominance of the cross-coupling nonlinearities arising from such a *slanted* configuration. The aforementioned conjectures must be investigated comprehensively in order to determine whether the potential advantages of designing slanted actuator axes justify the obvious difficulty of physically prototyping this configuration. A contrasting geometric configuration that is more conducive to mechanical prototyping is designed by constraining all the actuator axes to be *parallel*. Both of these geometric configurations are optimized in this chapter.

In order to define the geometry of the 3-PSS/S architecture in terms of a suitable set of parameters, the following geometric constraints are imposed: (i) each link is identical to the corresponding link in other kinematic loops, and (ii) the spherical joints at points C_i form an equilateral triangle (circumcenter at point O_C). In addition, when all actuators are at mid-stroke, the manipulator is defined to be in the home position. The following constraints are defined for the home position: (i) the three planes defined by $\triangle A_1A_2A_3$, $\triangle B_1B_2B_3$, and $\triangle C_1C_2C_3$ are mutually parallel, (ii) $\triangle B_1B_2B_3$ is an equilateral triangle (circumcenter at point O_B), and (iii) the points O_B , O_C , and the spherical joint at point O are all collinear on a line normal to the plane of $\triangle A_1A_2A_3$. Furthermore, a case-specific constraint is defined for each of the geometric configurations. For the slanted configuration, the axis of each actuator is defined to be coincident with the corresponding intermediate link at the home position. On the other hand, the actuation axes are normal to the plane of $\triangle A_1A_2A_3$ for the parallel configuration. The aforementioned constraints lead to five dimensional parameters (DKP) that completely define the geometry of the manipulator. Definitions of these parameters are provided in Table 4.1.

Table 4.1: Geometric parameterization of 3-PSS/S and 3-SPS/S manipulators

DKP	Definition	
	3- <u>PSS</u> /S [see Fig. 4.1(a)]	3- <u>SPS</u> /S [see Fig. 4.1(c)]
r	Radius of the circumcircle of $\triangle C_1C_2C_3$.	Radius of the circumcircle of $\triangle B_1B_2B_3$.
R	Radius of the circumcircle of $\triangle B_1B_2B_3$ at the home position.	Radius of the circumcircle of $\triangle A_1A_2A_3$.
h	Distance of point O_C from point O at the home position.	Distance of point O_B from point O at the home position.
H	Distance of point O_B from point O at the home position.	–
θ	Angle between O_BB_1 and O_CC_1 at the home position.	Angle between O_BB_1 and O_AA_1 at the home position.

4.5.2 Geometric Parameterization of 3-SPS/S Manipulators

The following geometric constraints are imposed on the geometry of the 3-SPS/S architecture: (i) the actuators are identical, (ii) the two planes defined by $\triangle A_1A_2A_3$ and $\triangle B_1B_2B_3$ are parallel when all the actuators are at mid-stroke (home position), (iii) $\triangle A_1A_2A_3$ and $\triangle B_1B_2B_3$ are equilateral triangles with the circumcenters located at point O_A and O_B respectively, and (iv) at the home position, points O_A , O_B , and O are all collinear on a line normal to the plane of $\triangle A_1A_2A_3$. In order to define the geometry of this architecture under these constraints, a set of DKPs are provided in Table 4.1. Please note that four dimensional parameters are defined for the 3-SPS/S architecture, whereas five parameters are required to specify the geometry of the 3-PSS/S architecture. The four DKPs for a 3-SPS/S architecture in Table 4.1 must be complemented by the length parameter of the actuators in order to obtain a complete definition of the manipulator geometry. However, in a restricted optimization problem, where the choice of an actuator is already defined, the four parameters in Table 4.1 is sufficient. Because of speed and accuracy requirements, the dimensional synthesis

exercise aims to find geometries of the 3-SPS/S architecture that can accommodate a preferred actuator. A detailed discussion on the design of this actuator is provided in [63].

4.6 Kinematic Performance Characteristics

Concurrency of the joint axes in an Agile eye manipulator enforces spherical motion of the moving platform. Since there are no other additional joints besides the three parallel RRR chains, the Agile eye can have a theoretically *unlimited* workspace [43]. In contrast, a passive spherical joint allows the moving platforms of the 3-PSS/S and the 3-SPS/S architectures to possess only rotational degrees of freedom and three additional limbs actuate the payload to achieve desired orientations. Although the passive joint guards against parasitic motion, it limits the motion range of the moving platform. Maximizing the orientation workspace thus constitutes an important objective in the corresponding dimensional optimization problem. In addition, a typical optomechatronic application demands the orientation maneuvers to be executed with a high degree of accuracy. In this regard, the dexterity of the workspace is identified as another important kinematic performance feature for the desired application.

4.6.1 Parameterization of an Orientation Workspace

Although the orientation workspace has been parameterized by Euler angles [8,112,113], they are not an appropriate representation of 3D rotations. If the workspace of a POM is described by some geometric entity embedded in a space defined by three Euler angles, there may exist multiple points that specify only one rotation. In fact, any representation of 3D rotations involving only three parameters suffers from this limitation (see Section 4.3.2). On the other hand, a unit quaternion \mathbf{q} and its negative

counterpart $-\mathbf{q}$ specify the same rotation [32]. Thus, the space of 3D rotations encloses half of the 3D spherical surface defined by the unit hypersphere $\|\mathbf{q}\| = 1$. Any rotation on this 3D spherical surface is uniquely defined. However, because of the difficulty of visualizing the four dimensional composition of a unit quaternion, a more intuitive parameterization must be developed. Observing the limitations of Euler angles and unit quaternions, an alternative parameterization is provided in [21] that defines the workspace by the trajectory of the terminal point of a unit direction vector fixed in the moving platform during all possible robotic maneuvers. However, such a specification of the orientation workspace is incomplete, since the torsional capabilities of a manipulator (i.e., the ability to rotate about the pointing vector) cannot be defined. The design synthesis in [21] involves the 3-PSS/S and the 3-SPS/S manipulators. Since their kinematic structures are inherently conducive to torsional motion, it can be argued that the outcomes of the dimensional synthesis exercise in [21] were perturbed minimally because of the incomplete parameterization.

Bonev and Gosselin proposed the Tilt & Torsion (T&T) angles to represent the orientation workspace in [114]. In order to avoid the representational singularity involved with conventional Euler angles, the three T&T angles are defined over a subset of all possible values. It should be noted that a fraction of the space of rotations cannot be specified by the T&T angles without introducing singularity. Since the corresponding rotations are practically unachievable by any parallel manipulator, the T&T angles can be employed as a nonsingular, non-redundant representation of the orientation workspace. An azimuth angle ϕ , a tilt angle θ , and a torsion angle σ constitute the three parameters of the T&T angles. These parameters are defined over the following ranges: $\phi \in (-\pi, \pi]$, $\sigma \in (-\pi, \pi]$, and $\theta \in [0, \pi)$. Correspondingly, a cylindrical coordinate system can be constructed where the azimuth angle ϕ is represented by the angular coordinate, the tilt angle θ is represented by the radial

distance, and the torsion angle σ is represented by the z axis. With the exception of rotations corresponding to $\theta = \pi$, a bijective representation of the space of rotations is provided by the $\{\theta, \phi, \sigma\}$ cylindrical coordinate frame [115, p. 82]. Consequently, an orientation workspace can be represented by a 3D region defined in the cylindrical space.

The reachable workspace of a POM is defined as the set of the orientations that the moving platform can achieve without violating any physical constraint (e.g., motion range of the joints). This implies that the moving platform can move between any two orientations in the reachable workspace without requiring disassembly. The regular workspace of a manipulator is defined by the maximal geometric object (e.g., cube, sphere, hypersphere, etc.) that can be completely embedded in the reachable workspace [48]. When an orientation workspace is mapped in a $\{\theta, \phi, \sigma\}$ coordinate frame, the z axis provides the torsional range and the projection of the map on a horizontal plane provides the *pointing* capacity (i.e., the achievable range of the pointing vector). In [114], the pointing capacity of an orientation manipulator is defined as the *projected orientation workspace*. Since the pointing capacity of the candidate manipulators are generally smaller than their torsional ranges, it must be optimized as a unique kinematic performance feature. To this end, the radius r_w of the maximal circle that can be enclosed inside the projected orientation workspace is selected as a measure of pointing capacity. Essentially, r_w provides the *maximal regular tilt* of a workspace. In addition, the volume v_w of the mapped workspace provides another kinematic performance metric that specifies the achievable range of rotational motion.

4.6.2 Specification of Dexterity of an Orientation

Workspace

Since no practical position sensor provides infinite resolution, the positions of the input links can only be measured up to a certain accuracy. Thus, a positioning inaccuracy is generated, which propagates through the kinematic structure and causes the moving platform to be perturbed from its desired position. Dexterity reflects the error in the position of the moving platform due to the limited resolution of the feedback sensors. In a given configuration, the reciprocal of the Euclidean norm condition number of the inverse kinematic Jacobian matrix provides the local dexterity index. Integrating the local dexterity index over the entire workspace provides the global conditioning index ρ_w . Although the global conditioning index (GCI) provides an aggregated measure of the dexterity characteristics of a workspace, one of the drawbacks arises from its inability to indicate any poor local behaviour [32]. However, this limitation is mitigated by including the minimum local dexterity index d_m in the objective function. Thus, the information provided by the GCI metric is effectively complimented to provide a complete picture of the quality of the workspace.

4.6.3 Objective Function

Since dimensional synthesis is a multi-objective optimization problem, the objective function can be formulated as a weighted sum of the selected performance indices. In contrast, defining the objective function as a vector of the desirable kinematic characteristics is more advantageous. This approach ensures that the optimization methodology has access to all available information at the cost of added complexity. Fortunately, several well established methods in RSM [18] and corresponding software

implementations (e.g., [116]) exist to address the complexity of multiple response optimization. However, a detailed account is beyond the scope of this thesis. The objective function in this chapter is defined as a vector of the maximal regular tilt, the volume, the GCI, and the minimum dexterity of the reachable workspace; i.e., $[r_w \ v_w \ \rho_w \ d_m]^T$.

4.7 Quantification of Kinematic Performance

An exact mathematical model for evaluating the kinematic performance metrics was not found for the candidate architectures. As an alternative, the kinematic performance was estimated by employing a discrete evaluation scheme, which is derived from the well-known discretization method [27, p. 219]. Despite being based on the conventional discretization technique, this evaluation scheme is unique in many aspects. The adopted technique employs a bisection search [117] in order to determine the workspace boundary, which contrasts with the conventional method of representing the workspace using a regular grid. Moreover, this estimation method utilizes the direct kinematic analysis in order to obtain an efficient estimation of the dexterity characteristics.

4.7.1 Numerical Characterization of an Orientation Workspace

The volume of the reachable workspace can be estimated by solving the inverse kinematics under a Monte-Carlo integration scheme at points randomly sampled from a possible work envelope; e.g., [118]. Similar to any Monte-Carlo integration, obtaining an accurate estimate requires a large number of trials, which is computationally

intensive. On the other hand, the conventional discretization method represents the possible workspace by a regular grid. In a subsequent step, the workspace boundary is estimated by evaluating the corresponding inverse kinematic problem at the nodal points of this grid. Besides workspace, the dexterity characteristics may also be evaluated at the nodal points. The accuracy of such an estimation scheme is dependent on the *fineness* of the grid and the computational cost increases exponentially with a finer grid. In view of these limitations, this thesis modifies the discretization technique to devise the following method for workspace estimation:

- Step 1: If the geometric constraints described in Section 4.5 are imposed on the candidate manipulators, the largest torsional displacement occur when all the actuators are at the extremes of their strokes. The geometric constraints also enforce the extreme torsional points to be located on the z axis of the cylindrical T&T coordinate system. In order to localize these two points σ^+ and σ^- , two bisection searches are separately conducted over the two ranges $[0, \pi]$ and $[-\pi, 0]$.
- Step 2: Once the torsional range $[\sigma^+, \sigma^-]$ is determined, it is discretized into n_σ equal segments. Except for the two terminal points, the *constant torsion* workspace for each segment of the torsional range is determined by localizing the maximum tilt for a set of azimuth angles. The range of the azimuth angle $(-\pi, \pi]$ is divided into n_ϕ number of equidistant points to represent this set. For each trial value of the azimuth angle, a bisection algorithm operates over the range $[0, \pi)$ to find the maximum tilt (i.e., workspace boundary). Thus, the constant torsion workspace is represented by n_ϕ radial rays, where the length of each ray represents the maximal tilt. It should be noted that the constant torsion workspaces are *tomograms* determined across the torsional axis. Therefore,

they can be combined together, in the order of the corresponding torsional values, to represent the mapped orientation workspace. The data structure of this representation is very convenient for determining the projected orientation workspace. Subsequently, the maximal regular tilt r_w can be determined by finding the shortest radial distance between the boundary of the projected workspace and the origin.

- Step 3: Each of the constant torsion workspaces determined in the previous step is a collection of coplanar triangular elements with a common vertex (i.e., the origin defined by a zero tilt angle). Subsequently, the corresponding area can be estimated by evaluating the sum of all the triangular areas. Since the distances between any two adjacent constant torsion workspaces are equal, estimating the entire workspace volume v_w is trivial.

Although exactly determining the workspace extremity by a bisection algorithm may require a large number of iterations, a high resolution estimation can be reached within a few trials. Thus, this evaluation scheme efficiently estimates the workspace boundary with an accuracy specified by the search parameters. It should be noted that each bisection search in the above algorithm evaluates an the inverse kinematic function $f(\theta, \phi, \sigma)$ that returns a one when the given orientation yields a reachable configuration. Otherwise, a zero is returned.

4.7.2 Estimating the Dexterity of an Orientation Workspace

In a comprehensive review paper [58], Merlet criticized the use of the condition number of the inverse Jacobian matrix and the GCI as kinematic performance indices. For a robot featuring both translational and rotational degrees of freedom, the condition number suffers from being non-homogeneous in terms of the units that

specify the manipulator geometry. However, in this study, both manipulators have only rotational degrees of freedom. Consequently, the corresponding inverse Jacobian matrices are independent of the units. Since a closed-form solution of the GCI in terms of the kinematic parameters is very difficult to obtain, it is computed from the local dexterity characteristics of a finite number of points in the workspace. This method of computation gives rise to the possibility of inaccurate estimation of the GCI. Merlet in [58], therefore, suggests Monte-Carlo integration for a more reliable calculation. However, a Monte-Carlo integration may be computationally expensive when a large number of candidate geometries must be evaluated. Reliable estimations of the GCI and the minimum local dexterity can still be obtained efficiently if the kinematic characteristics are evaluated over a regular grid representing the actuated joint space by applying the direct kinematic analysis. A regular grid in the actuated joint space usually corresponds to a non-uniform grid in the workspace coordinates. However, as long as the sample points do not exhibit any obvious clustering and represent the entire workspace volume, uniform dispersion of the points is not a strict requirement for estimating the GCI. To this end, the stroke of each actuator was discretized into n divisions. Subsequently, a large regular grid in the joint space can be constructed, which contains n^3 number of nodes, each specifying a unique combination of input link lengths. By solving the direct kinematics problem, the dexterity features were evaluated at each of these nodal points. The mean of the local dexterity indices thus determined provides an estimate of the GCI. In addition, the minimum of the sampled dexterity indices provide the minimum dexterity of the workspace. It should be noted that the conventional discretization method performs this evaluation in the workspace coordinates employing the inverse kinematic analysis.

4.8 Optimization Procedure

In order to efficiently explore the search space in a systematic manner, RSM offers multiple methods for designing experiments. In contrast to the conventional methods, computer generated optimal designs are appropriate for situations where: (i) the search region is of irregular shape, (ii) the underlying model is nonstandard and unknown, and (iii) it is imperative to minimize the number of search region probes [18]. All these situations hold true for the dimensional synthesis problem. Since the IV-Optimal (integrated variance) design seeks to minimize the integral of the prediction variance across the search space [116], this design is considered to be the most pertinent to the problem being studied.

It is safe to assume that the objective function is not well behaved over a large search space. However, when *a priori* knowledge of a solution is unavailable, a large search space must be considered to ensure that it contains at least one or preferably multiple solutions. In a dimensional synthesis problem, the curvature of a response surface over a large parameter space is usually not suitable for a reliable regression analysis. Paradoxically, a RSM optimization performs best when the corresponding regression models closely fit the observed responses and such a fit is generally obtained for a smaller parameter space. In order to address this challenge, the optimization procedure in this chapter employs a large parameter space as a starting point so that it is virtually guaranteed to include multiple feasible solutions. The initial search space is then explored methodically in order to identify a region of interest (ROI) that exhibits favorable kinematic performance. Subsequently, an elaborate experiment design is employed to characterize this ROI.

A space-filling design [18] is the more appropriate choice of experiment formulation when a large search space must be explored in an efficient manner. It produces

experiment points that are distributed with a “loose” uniformity over the search region. In this chapter, a Latin hypercube (LHC) design [119] was adopted for this purpose. Although other space-filling designs exist, as long as a ROI can be identified reliably, the choice of the corresponding method is not critical. Subsequently, an IV-Optimal experiment design can be employed to explore the previously determined ROI for an optimal solution. With regards to the numerical experiment, it should be noted that the trials need not be replicated since the objective function is deterministically evaluated by the discretization method described in Section 4.7. Since the workspace of an orientation manipulator is scale invariant, the parameter space of the candidate manipulators were normalized with respect to the stroke of the actuator. Thus, all DKPs were conveniently transformed into dimensionless quantities.

Because the initial parameter space in the LHC experiments was large in size, a greater number of sample points were required to identify a suitable ROI. However, the high computing cost of sampling a large number of points must be reduced for the sake of efficiency. Since the objective of the initial LHC experiment is to determine a ROI, highly accurate estimations of the responses is not strictly required. Correspondingly, a coarse discretization of the workspace and the joint space was adopted for efficiently evaluating the objective function. In reference to the estimation methods described in Sections 4.7.1 and 4.7.2, the parameters $n_\sigma = 10$, $n_\phi = 25$, and $n = 10$ were used to obtain a coarse grid. Although the accuracy of the estimated responses suffered, it was considered sufficient for identifying a suitable ROI.

A fifth order model was assumed in the IV-Optimal experiment that characterizes the ROI determined by the LHC sampling. When a higher order model is assumed for a response surface experiment, the search space must be probed at a greater number of points, which adds to the computational cost. However, the probability of model underfitting is minimized. Since no *a priori* information is available regarding the

curvature of the response surface, the order of the assumed model was chosen to find a compromise between the computational cost and the probability of underfitting the responses. In addition, a lower order model can always be adopted if statistical analysis indicates overfitting. The workspace and the joint space were discretized into a fine grid so that the estimated responses are a close approximation of the true kinematic performance. In specific terms, the discretization parameters $n_\sigma = 50$, $n_\phi = 90$, and $n = 25$ were employed in order to evaluate the responses in the IV-Optimal experiment. Since not all sample points in the IV-Optimal experiment provided desirable kinematic performance, a rejection criterion specifying $d_m > 0.15$ was established. Whenever a trial point exhibited a minimum dexterity index less than 0.15, the unfavorable geometry was excluded from the subsequent analysis.

Direct kinematics of the candidate manipulators were solved iteratively in order to estimate the dexterity features. In most cases, it was observed that the iterative solution to the direct kinematics problem converged to a solution for all sample joint space points. However, when the geometry was not conducive to good kinematic performance, a fraction of the trial points did not yield a solution. If this failure rate exceeded a predefined threshold, the corresponding estimations of dexterity features were considered unreliable. Consequently, these geometries were excluded from the experimental data analysis. Given the large size of the initial search space, a lax failure threshold of 5% was selected for the LHC sampling; i.e., when a trial geometry did not converge to a solution in the direct kinematics problem for more than 5% of the nodal points representing the actuated joint space, it was rejected from analysis. In contrast, the rejection criterion for the IV-Optimal design was set at a much stricter 1% failure threshold.

The dimensional synthesis exercise for the parallel configuration of the 3-PSS/S architecture is described in details in the following section. The process of determining

the optimal dimensions are very similar for all the candidate architectures. For the sake of a concise discourse, the optimization procedures for the other two candidate architectures are discussed only briefly.

4.8.1 3-PSS/S Manipulator (Parallel Configuration)

The following observations are made prior to the numerical experiments: (i) larger values of R and H increase the size of the manipulator, (ii) since three spherical joints must be located on a circular circumference of radius r , a small value renders physical prototyping difficult, and (iii) a small θ reduces the possibility of link interference.

4.8.1.1 Reduction of the Search Space

The optimization procedure began with a relatively large and arbitrarily chosen search space that was defined by the following DKP intervals: $r \in [1.00, 1.30]$, $h \in [0.00, 0.25]$, $R \in [1.35, 1.75]$, $H \in [2.00, 2.50]$, and $\theta \in [50^\circ, 90^\circ]$. This large search space was probed at a total of 600 points. These sample points were determined by MATLAB[®] under a LHC design. Subsequently, the recorded responses in the LHC experiment were screened to identify a suitable ROI, where all sampled responses were contained within the following *feasible region*: $r_w > 30^\circ$, $\rho_w > 0.6$, and $d_m > 0.25$. In specific terms, this region of interest was defined by the following parameter intervals: $r \in [1.0001, 1.0370]$, $h \in [0.0463, 0.1862]$, $R \in [1.4769, 1.7366]$, $H \in [2.0886, 2.4060]$, and $\theta \in [79.8327^\circ, 89.8504^\circ]$. Since a very small range of the parameter r constitute the region of interest, it was correspondingly excluded from the formulation so that all subsequent search spaces only include $r = 1.00$.

Based on the heuristics gained from the observed responses of the LHC sampling, an IV-Optimal experiment was performed on a parameter space that is defined by

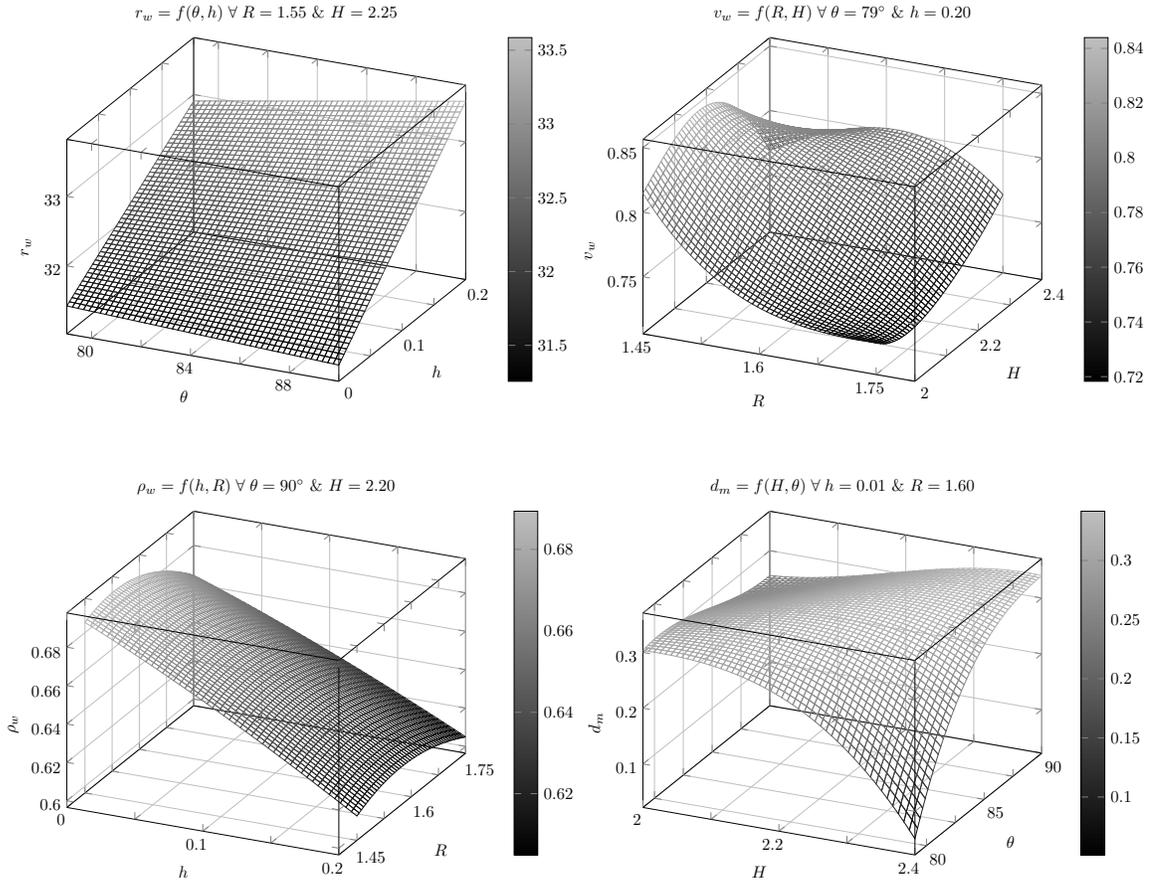


Figure 4.2: A few examples of the fitted response surfaces of the 3-PSS/S manipulator (parallel configuration). For all models, $r = 1.0$.

the following parameter intervals: $r \in [1.00, 1.00]$, $h \in [0.00, 0.20]$, $R \in [1.45, 1.75]$, $H \in [2.00, 2.40]$, and $\theta \in [79.00^\circ, 90.00^\circ]$.

4.8.1.2 Data Analysis

The reduced search space was probed at a total of 131 points for the IV-Optimal experiment. It was observed that about 30% of the sampled responses exhibited very poor dexterity characteristics (i.e., $d_m < 0.15$). Correspondingly, these trial points were excluded from the subsequent analysis. The experimental data from the remaining samples were then fitted to a series of models (linear, first order with

interaction, quadratic, cubic, etc.) in order to evaluate the goodness of fit. Based on the values of the adjusted and the predicted coefficient of multiple determination R^2 , a suitable model for each of the responses was selected. Subsequently, the selected standard model was modified based on the analysis of variance (ANOVA) study to exclude any statistically insignificant terms. A 95% confidence interval was chosen as the threshold of acceptance for the statistical significance testing. These fitted models characterized the response surfaces.

Optimization of a response surface will produce misleading results unless the model provides an adequate fit [18]. Checking for model adequacy ensures that the fitted model is indeed a good approximation of the real system and none of the regression assumptions (e.g., residuals are normally distributed with zero mean) are violated. Therefore, all the fitted models must be verified by residual analysis and lack of fit testing. The corresponding Box-Cox plots for the responses v_w and ρ_w indicated that transformations of these responses were required to maintain agreement with the least-squares assumptions. Accordingly, the response v_w was power transformed by a factor of $-\frac{1}{2}$. In addition, a logarithmic transform was applied for the response ρ_w . The transformed responses v_w and ρ_w were then fitted to a modified quartic and a modified cubic model respectively. In contrast, the remaining two responses r_w and d_m did not require any transformations. Although a reduced two factor interaction model was selected for the response r_w , a reduced quartic model was necessary to adequately fit observed values of the response d_m . A randomly selected subset of these fitted models are graphically represented in Fig. 4.2 as 3D response surfaces. It should be mentioned that all the fitted models were a polynomial function of the four geometric parameters h, R, H , and θ (r was fixed). Since a 4D graphical representation is not possible, two of the four DKPs were chosen as the function variables. In addition, the other two DKPs were fixed at some constant value. Although the choice of the function variables

Table 4.2: Coefficients of multiple determination for the regression models of the kinematic performance metrics

Architecture	Statistic	Kinematic Performance Metrics			
		r_w	v_w	ρ_w	d_m
3- <u>P</u> SS/S (parallel)	R^2	0.9797	0.9997	1.0000	0.9889
	Adjusted R^2	0.9780	0.9992	1.0000	0.9755
	Predicted R^2	0.9755	0.9976	1.0000	0.9316
3- <u>P</u> SS/S (slanted)	R^2	0.9990	1.0000	0.9997	0.9843
	Adjusted R^2	0.9988	1.0000	0.9994	0.9715
	Predicted R^2	0.9985	0.9999	0.9988	0.9292
3- <u>S</u> PS/S	R^2	0.9991	1.0000	1.0000	1.0000
	Adjusted R^2	0.9990	1.0000	1.0000	0.9999
	Predicted R^2	0.9988	1.0000	1.0000	0.9993

was arbitrary, the constant values of the other two DKPs were selected by trial and error so that the curvature of the response surfaces were prominent. In addition, the adequacy of the empirical models for each of the responses were confirmed since the predicted and the corresponding adjusted coefficients of multiple determination were in reasonable agreement as shown in Table 4.2.

4.8.1.3 Optimum Solution

By applying the simultaneous optimization technique in [120], multiple solutions with varying desirability scores [18] were formulated in order to satisfy a set of constraints. Besides the obvious constraint of maximizing the four responses (i.e., r_w , v_w , ρ_w , and d_m), an additional restriction was imposed so that the optimal solution includes a small value of the geometric parameter θ , because it was observed that a 3-PSS/S manipulator featuring a small θ is less susceptible to possess a link collision in the reachable workspace. After numerically exploring the fitted models, several solutions were found that satisfied the aforementioned constraints. Subsequently, the solution with the highest desirability score was chosen as the optimal geometry. The DKP

values $r = 1.00$, $h = 0.13$, $R = 1.71$, $H = 2.39$ and $\theta = 82.55^\circ$ define the optimal solution.

4.8.2 3-PSS/S Manipulator (Slanted Configuration)

A parameter space specified by the intervals $r \in [0.9, 1.50]$, $h \in [0.00, 0.50]$, $R \in [1.00, 1.80]$, $H \in [1.75, 2.75]$, and $\theta \in [65^\circ, 95^\circ]$ was selected arbitrarily for the initial evaluation. Assuming that this space contains multiple dimensional configurations that provide suitable kinematic performance, a LHC experiment sampled the objective function at 600 points. Based on these experimental observations, a ROI was identified where the recorded responses satisfied constituted the following feasible region: $r_w > 30^\circ$, $\rho_w > 0.65$, and $d_m > 0.35$. Subsequently, an IV-Optimal experiment was employed to explore the ROI assuming a fifth order model for the responses. Similar to the parallel configuration of the 3-PSS/S manipulator, the search for an optimal solution was restricted to include only those geometries that feature a small θ . The experimental data was then analyzed by [116] to obtain an optimal combination of DKPs that maximized the responses. The optimal geometry thus determined is specified by $r = 1.20$, $h = 0.07$, $R = 1.25$, $H = 1.91$ and $\theta = 89.88^\circ$.

4.8.3 3-SPS/S Manipulator

In order to optimize the geometry of the 3-SPS/S manipulator, an initial search space featuring the parameter intervals $r \in [1.0, 1.70]$, $h \in [0.0, 1.0]$, $R \in [4.50, 8.50]$, and $\theta \in [60^\circ, 120^\circ]$ was selected. However, the selection of the search space was not purely random, unlike the previous optimization exercises. Since an appropriate geometry of the 3-SPS/S manipulator must accommodate the preferred actuator over its entire stroke, an interval with large values of R was chosen to obtain a suitable

Table 4.3: Optimal Kinematic Performance

Architecture	Evaluation Method	Kinematic Performance Metrics			
		r_w	v_w	ρ_w	d_m
		degrees	rad ³	–	–
3-PSS/S (parallel)	Simulated	32.5195	0.7382	0.6618	0.2670
	Response surface	32.3227	0.7403	0.6622	0.2638
3-PSS/S (slanted)	Simulated	36.1450	1.0576	0.6597	0.3206
	Response surface	36.0101	1.0557	0.6596	0.3240
3-SPS/S	Simulated	33.7500	0.8954	0.6926	0.3486
	Response surface	33.7410	0.8957	0.6924	0.3543

parameter space. It should be noted that the mid-stroke length of the actuator was measured to be 11.1 times the range of its displacement. Similar to the previous cases, the initial search space was reduced by a LHC experiment in order to determine a region of interest. Subsequently, assuming a fifth order model for the responses, an IV-Optimal experiment was employed to obtain appropriate regression models that fit the experimental observations. A set of optimum solutions with varying desirability scores was then determined by [116]. The solution with the highest desirability score is specified by $r = 1.12$, $h = 0.10$, $R = 5.94$, and $\theta = 86.65^\circ$.

4.9 Optimal Kinematic Performance

The synthesized dimensions of the three candidate architectures are employed to estimate the kinematic performance metrics provided in Table 4.3. While the discretization method in Section 4.7 evaluates the optimal geometries employing the inverse and the direct kinematic analyses, the response surfaces fitted in the IV-Optimal experiments estimate the performance features from the underlying regression models. The data presented in Table 4.3 indicates a close agreement between the predicted and the estimates values of each kinematic performance measure. In addition, the reachable

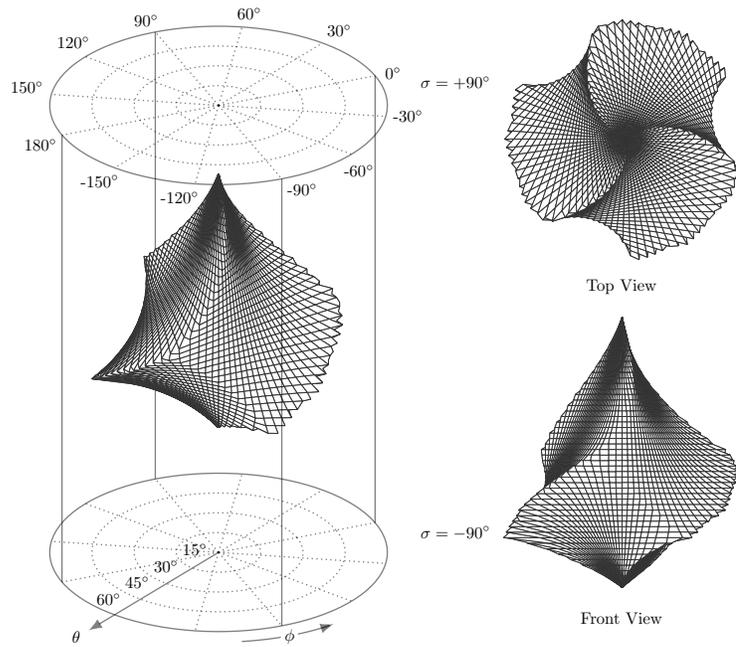


Figure 4.3: Optimal reachable workspace of the 3-PSS/S manipulator (parallel configuration).

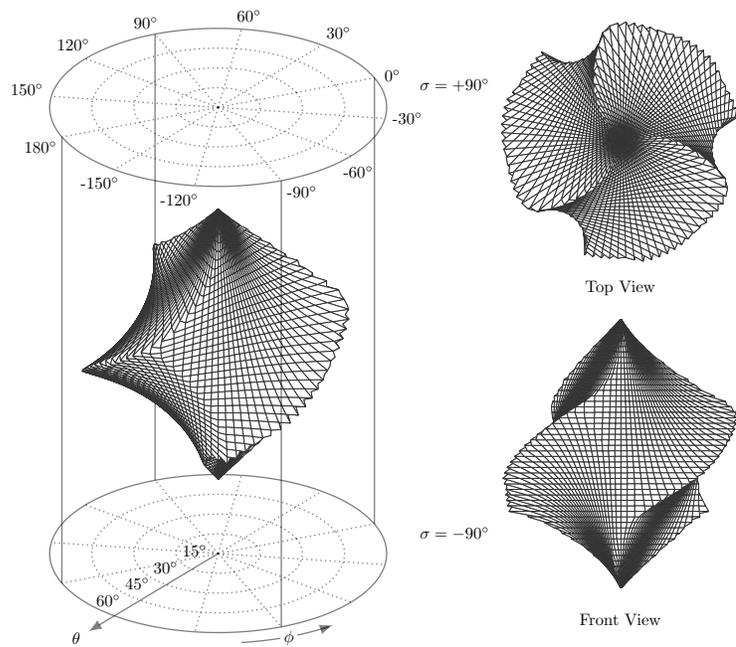


Figure 4.4: Optimal reachable workspace of the 3-PSS/S manipulator (slanted configuration).

workspaces of the three candidate architectures are mapped in Figures 4.3, 4.4, and 4.5. In addition, the optimal projected orientation workspaces are provided in Fig. 4.6. The reachable orientation workspaces, as mapped in Figures 4.3, 4.4, and 4.5, show certain interesting characteristics. Unlike the other two manipulators, it is easy to see that the workspace for the parallel configuration of the 3-PSS/S manipulator is not symmetrical about the zero torsion plane. In addition, its optimal dexterity characteristics are also slightly inferior than the candidates.

Dexterity characteristics of the optimal manipulators are presented as a function of the input link lengths in Fig. 4.7. A fairly uniform behaviour is exhibited in the dexterity maps that feature the third actuator positioned at its mid-stroke. Comparing the dexterity maps in Fig. 4.7, it can be seen that the 3-SPS/S architecture is the most dexterous manipulator of the three candidates. Furthermore, its three dexterity maps exhibit similar curvatures, which contrasts with the two configurations of the 3-PSS/S architecture. It can also be observed that the dexterity features of the poorly behaved regions of the workspace provided by the 3-PSS/S manipulator can be improved by slanting the prismatic actuator axes towards the intermediate links. However, the data presented in Table 4.3 suggests that such a modification may not always lead to an enhanced GCI of the workspace.

The geometry of the 3-SPS/S architecture must accommodate three prismatic actuators between the base and the moving platform. Therefore, it can be inferred that a suitable motion generator must possess a high stroke to length ratio in order to obtain a large workspace. The preferred actuator, which provides a low stroke to length ratio, lacks such a feature. Nonetheless, it was shown that favorable performance can still be obtained by applying the proposed RSM optimization. Despite the limited motion range of the actuators, a large workspace was obtained from the synthesized geometry. Although the size of this workspace is comparable to the other two candidate

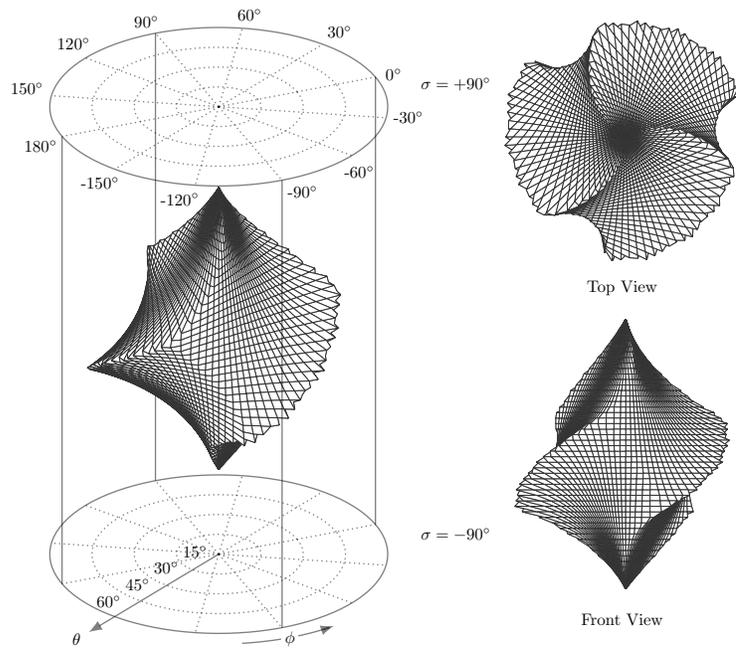


Figure 4.5: Optimal reachable workspace of the 3-SPS/S manipulator.

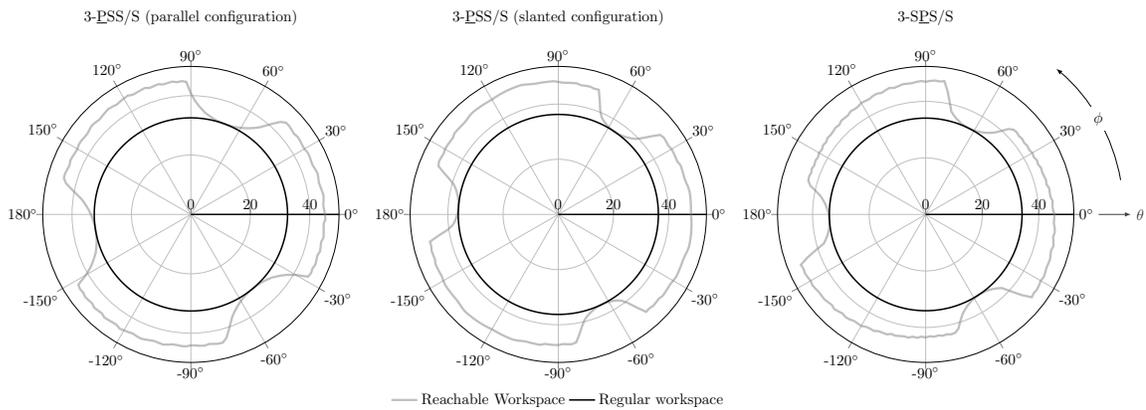


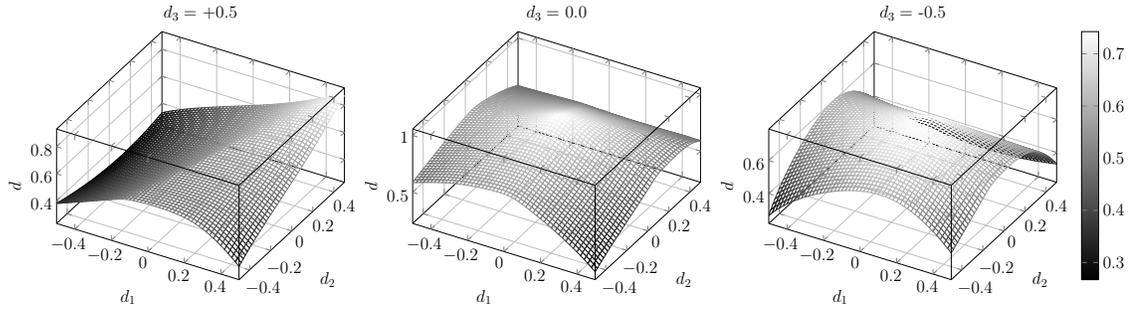
Figure 4.6: The projected orientation workspaces of the three candidate architectures in terms of azimuth and tilt angles.

architectures, the corresponding footprint is significantly large, which renders this architecture unsuitable for a size constrained application. Moreover, because of the two Halbach magnetic arrays, the cylinder of the actuator designed in Chapter 3 exhibits unfavorably high inertia (see Table 3.1). In contrast to the 3-PSS/S manipulator where the high inertia of the actuator cylinder is fixed to the ground, this architecture allows

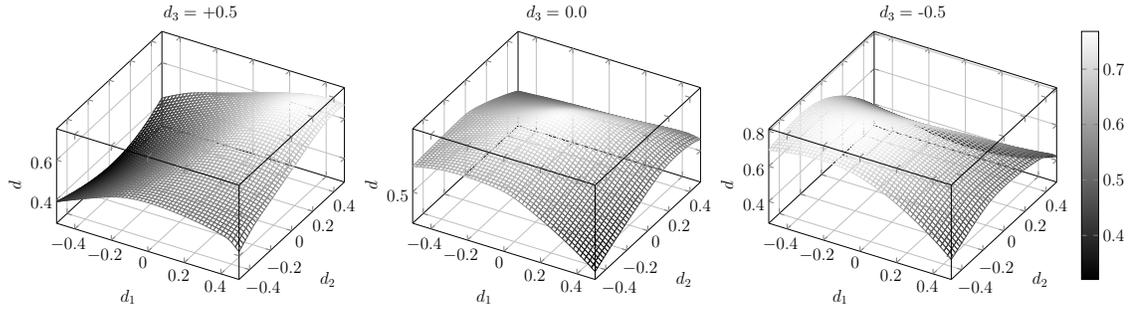
its actuators to possess constrained spatial motion. Since the high kinetic energy associated with moving these high inertia elements must be supplied by the available actuation capacity, this architecture cannot utilize the full potential of the designed actuator. Therefore, despite being similar to the other two candidate architecture in terms of kinematic performance, the 3-SPS/S manipulator is regarded unsuitable for the desired application.

4.10 Discussion

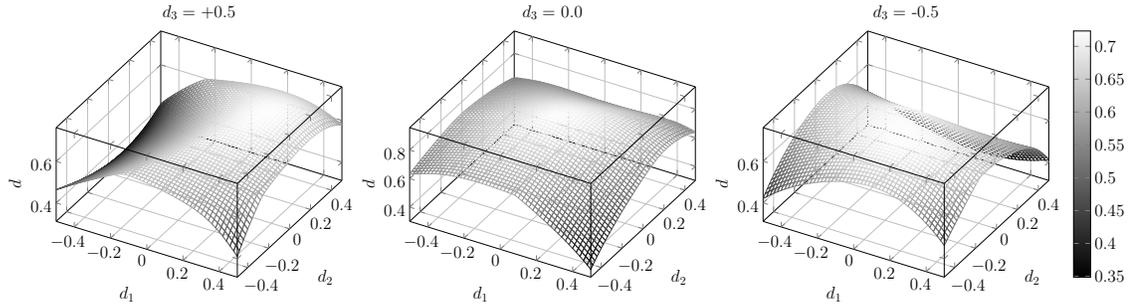
In [58], it was suggested that Monte-Carlo integration might be a more reliable choice of an estimation method for quantifying the dexterity characteristics. As described in Section 4.7.2, the dexterity characteristics are estimated by solving the direct kinematics problem at a finite set of points that were were uniformly distributed in the joint space coordinates covering the entire range of the actuator stroke. In order to evaluate the accuracy obtained by the estimation scheme in Section 4.7.2, a Monte-Carlo integration method was adopted as a benchmark. The optimal dimensions of the parallel configuration of the 3-PSS/S manipulator was chosen as the benchmarking platform, since it exhibits the most non-uniform distribution of the dexterity index. In the estimation method in Section 4.7.2, the stroke of each actuator was divided into n equidistant points in order to construct a sample set of n^3 elements. The test cases of the benchmarking study were constructed by varying the value of n . In each of the test cases, a Monte-Carlo simulation consisting of an identical number of trials was performed. The results of this benchmarking study is presented in Fig. 4.8. The estimates of the GCI metric provided by the Monte-Carlo simulation are higher than those evaluated by the direct kinematic analysis. This discrepancy can be explained by the dexterity maps presented in Fig. 4.7, where it was shown that subpar dexterity



(a) 3-PSS/S (parallel configuration).



(b) 3-PSS/S (slanted configuration).



(c) 3-SPS/S.

Figure 4.7: Local dexterity index d as a function of the joint coordinates of the first two actuators (i.e., d_1 and d_2), while the third actuator was fixed at three different positions; left: $d_3 = +0.5$, middle: $d_3 = 0.0$, and right: $d_3 = -0.5$. The entire stroke of each actuator is normalized over $[-0.5, 0.5]$ in order to represent the joint coordinates.

indices occur at the workspace boundary. Since the adopted estimation method sampled the boundary points more frequently than the Monte-Carlo simulation, the corresponding estimates of the GCI metric were smaller in magnitude. As the size of the sample set increased (i.e., n was increased), the ratio of the boundary points to the interior points in the sample set asymptotically approached the actual ratio

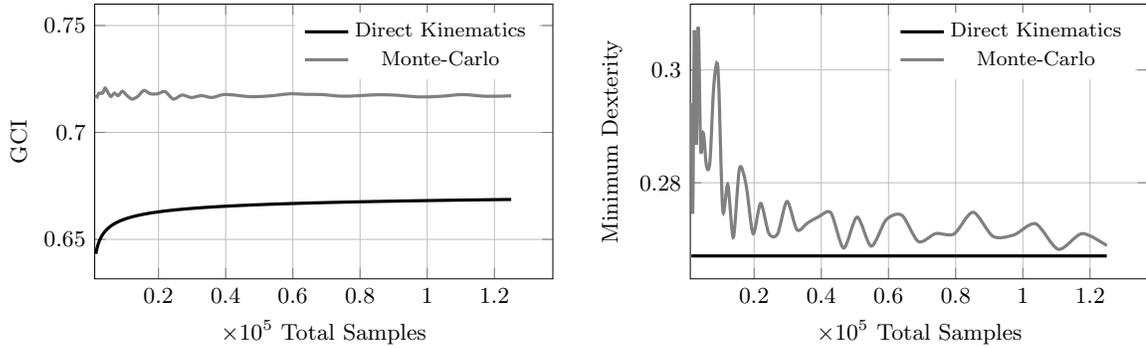


Figure 4.8: Numerical stability of different methods estimating the dexterity characteristics of the optimal 3-PSS/S manipulator (parallel configuration).

of the volume of the boundary to the volume of interior region. On the other hand, the Monte-Carlo method performed irregularly for smaller sample sets. However, the erratic behaviour ceased once the number of trials were sufficiently large. In the absence of *a priori* knowledge of the true value of the GCI metric, it is unclear from Fig. 4.8 which method provides a better estimate of GCI. Nonetheless, from a practical point of view, an underestimate of the GCI metric is less hazardous than an overestimate because a manipulator prototype, whose GCI was overestimated during the design phase, cannot meet the expected performance benchmark. In this regard, the adopted method can be considered a more appropriate estimation scheme for evaluating the GCI.

Since the minimum local dexterity index d_m indicates the presence of a kinematic singularity in the workspace, a sufficiently accurate estimate is more important than the GCI measure. Between the adopted method and the Monte-Carlo simulation scheme, the former is better equipped to explore the workspace boundary where inferior dexterity performance is usually observed. Correspondingly, the adopted method provided an estimate of the d_m metric that remained unchanged irrespective of the number of samples. Despite the unavailability of the true value of d_m , it can be confidently said that the adopted method efficiently provided a better estimate than

the Monte-Carlo simulation. It is because the true value of d_m is most certainly not greater than the estimate provided by the direct kinematic analysis.

The direct kinematics of a parallel robot is typically more difficult to solve because a closed-form solution is not readily available. Iterative solution methods are often adopted as an alternative. Although success is not always guaranteed, a solution is usually found when the iterative technique begins its search from within close proximity of a solution. In order to enhance the probability of converging to a solution, this initial approximation (i.e., starting point) is often obtained by specialized estimation schemes such as the Genetic Algorithm; e.g., [121]. These specialized estimation methods are generally computationally expensive. In this chapter, however, it was not necessary to compute an initial approximation. The iterative solutions presented in Sections 4.3.2 and 4.4.2 always begin their search from the corresponding home position. Subsequently, a solution is obtained within a few iterations. Despite the large number of design kinematic parameters in Table 4.1, the optimum geometries of the manipulators were synthesized in a computationally efficient manner. The respective parameter spaces were probed at a small number of points in order to synthesize the optimal manipulators.

Since the optimal workspace provided by the slanted configuration of the 3-PSS/S manipulator is the largest of the candidate architectures, it was selected for a numerical experiment to study the performance of the workspace characterization method described in Section 4.7.1. The corresponding experimental results presented in Fig. 4.9 confirm an obvious property of the discretization method: the estimation accuracy improves with a finer grid. However, these plots provide a graphical means of selecting suitable values for the discretization parameters n_ϕ and n_σ so that an appropriate compromise between the computational cost and the accuracy of the estimates can be found. Furthermore, the bisection method employed to determine the

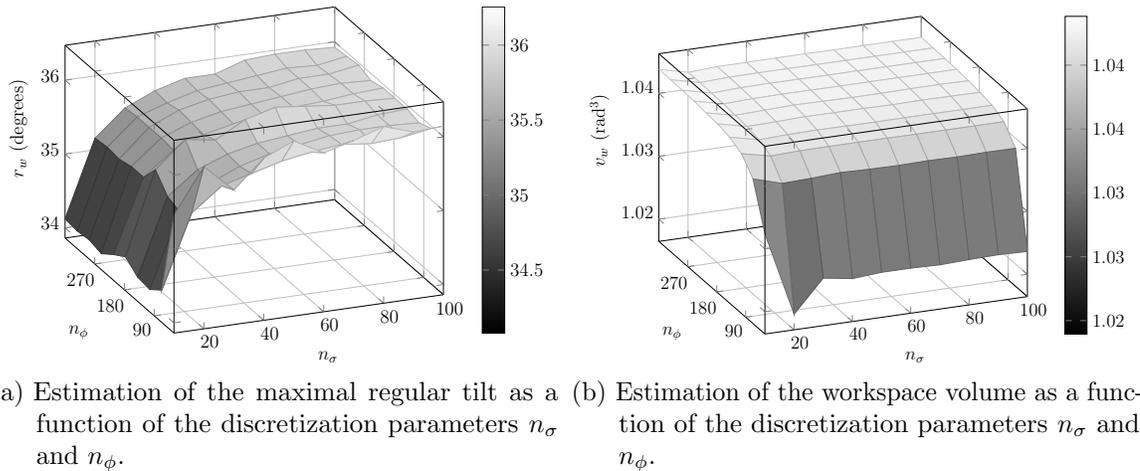


Figure 4.9: Estimations of the optimal workspace characteristics provided by the 3-SPS/S manipulator exhibit asymptotic convergence.

workspace boundary was observed to be particularly efficient. In specific terms, only 14 evaluations of the inverse kinematic problem was necessary in order to determine the maximum tilt along a radial direction with an accuracy of 0.1 mrad, which is in a stark contrast with the conventional discretization method where such a fine estimate generally involves a significantly greater computational cost. Despite these advantages, the bisection method in its implemented form is unable to detect any void region in the workspace. However, the workspaces of the candidate architectures were examined to be free of such features.

4.11 Conclusion

In general, dimensional synthesis of a parallel robot is not a trivial problem. Even for a fairly simple architecture, the kinematic performance metrics cannot be expressed as an explicit function of the geometric parameters in most cases. Because of the complex structural topology of parallel robots, the performance metrics are often ill-behaved over the corresponding search region. Hence, an optimum solution can

only be found when the optimization method overcomes these challenges. Response surface methodology offers greater flexibility and attractive efficiency in this regard, thanks to its strong statistical foundation. This chapter demonstrated how response surface methodology can be employed to synthesize the optimum geometry of a parallel orientation manipulator for maximum kinematic performance.

Chapter 5

Dynamic Simulation & Control of Parallel Orientation Manipulators

The slanted configuration of the 3-PSS/S manipulator can be characterized by the tilted axes of its actuators. In contrast, the parallel configuration features a vertical alignment of the actuator axes. Although implementing a prototype of the slanted configuration is not as simple as the parallel configuration, dimensional synthesis of the two manipulators showed that the optimal geometry of the former provides superior kinematic performance. In this chapter, the dynamic performance characteristics of the two configurations were numerically investigated employing appropriate dynamic models. Besides multibody mechanics, the scope of these models includes actuator dynamics and related mechanical phenomena (e.g., joint friction). In this regard, the bond graph modeling formalism is preferred, since a bond graph model can seamlessly incorporate multiple energy domains. In addition, it facilitates construction of multibody models with multiple kinematic loops at a graphic level. Correspondingly, bond graphs formalism was adopted to develop the dynamic models of the candidate manipulators. In order to evaluate the dynamic performance of the manipulators, a numerical experiment was performed where the constructed models were employed to execute a series of random robotic maneuvers. Since executing the test maneuvers required implementation of motion control, an active disturbance rejection controller (ADRC) was formulated for each manipulator model. In order to evaluate the robustness of the controller against the cross-coupling nonlinearities and the time-varying inertia, the system responses exhibited in the test maneuvers were recorded and were subsequently analyzed. The corresponding simulation results confirmed that the ADRC technology can deliver excellent dynamic performance despite the nonlinear dynamics of the candidate manipulators.

5.1 Introduction

In terms of kinetics, a parallel manipulator is essentially a constrained multibody system. A dynamic model of this system facilitates quantitative investigation into its dynamic characteristics. Because of the presence of multiple kinematic loops, a parallel manipulator generally possesses a greater number of moving bodies than a comparable serial architecture. As a result, the corresponding model is structurally and computationally more complex. However, several multibody dynamic modeling formalisms have been proposed in the literature. Examples include the Newton-Euler method [122], the Lagrangian approach [123], the Lagrange-D'Alembert formulation [76, 77], etc. In addition, component based system modeling approaches are also reported. Linear graph theory has been used in the analysis of multibody dynamics; examples include [124]. Besides linear graphs, bond graphs have also been used to model multibody dynamics; e.g., [22, 125–127].

Among the three manipulators that were dimensionally synthesized in Chapter 4, the 3-SPS/S architecture was considered unsuitable for the desired application because it cannot utilize the full potential of the preferred actuator. The dynamic performance of the remaining two optimized manipulators are assessed in this chapter through the application of multibody simulation. Constructing a 3D multibody model that consists of multiple kinematic loops and incorporates actuator dynamics and additional physics (e.g., frictional characteristics of the joints) is particularly straightforward in bond graph formalism. Moreover, this modeling formalism inherently accounts for the causal conflicts that can arise from the kinematics of the robot by imposing the kinematic constraints through stiff parasitic elements. The design study conducted in this chapter is accordingly based on multibody bond graph models. In addition, motion controllers based on ADRC technology were formulated for the candidate

manipulators in order to evaluate their robustness against the time-varying inertia and the cross-coupling nonlinearities.

Several performance metrics have been proposed in the literature in order to quantify the dynamic performance of robotic manipulators. A brief survey can be found in [128]. Ma and Angeles proposed the concept of dynamic isotropy as a measure of dynamic performance of manipulators in [129]. A manipulator is defined to be in dynamic isotropy when its generalized inertia matrix is diagonal and perfectly conditioned. Such a configuration refers to a completely decoupled dynamics of a manipulator. A performance metric called the dynamic conditioning index was defined in [129] in order to measure the dynamic isotropy. A weighted least-squares distance between the generalized inertia matrix of a manipulator and its nearest isotropic matrix provides the dynamic conditioning index. In a performance benchmarking study [130], the maximum actuation effort generated by the actuated joints to produce a certain motion of the moving platform was considered as a performance metric. It should be noted that this index was developed to measure the performance of two 6 DOF parallel manipulators. The condition number of the generalized inertia matrix was employed as a local measure of dynamic performance of a 2 DOF parallel manipulator in [131]. In order to estimate the global performance characteristics, the mean and the standard deviation of this local measure over the manipulation workspace were used. In a comparative study conducted in [132] the dynamic performance of two 3 DOF parallel manipulators (one translational and two rotational degrees of freedom) was measured in terms of the maximum joint force required to generate a unit acceleration of the moving platform. Besides the aforementioned dynamic performance metrics, several other measures were proposed in the literature; examples include the *generalized inertia ellipsoid* [133], the *dynamic manipulability ellipsoid* [134], etc.

The research works referenced above are geared towards quantifying the *absolute*

dynamical ability of a manipulator. Although the aforementioned performance metrics can facilitate design tasks involving synthesis and evaluation, they convey an incomplete picture for the practical case. Regardless of how well-designed a manipulator is, a sub-optimal controller may not be able to completely utilize its dynamic potential. Therefore, only the aggregated performance of a manipulator and its controller offers practical significance. Correspondingly, the notion of dynamic performance of a parallel manipulator was approached from a practical point of view in this chapter. To this end, the *practical* dynamical capability of a manipulator was quantified by the controlled responses obtained from the respective model. It is relevant to mention that the capacity of a controller in terms of utilizing the available dynamics is not infinite. Furthermore, for two different robotic maneuvers this capacity may vary for the same manipulator because the coupling nonlinearities are a function of the poses of the moving links. Correspondingly, a Monte-Carlo sampling scheme was adopted to estimate the practical performance of each candidate manipulator.

The remainder of the chapter is organized as follows to illustrate the aforementioned ideas. The dynamical models of the candidate manipulators were developed employing an algorithmic multibody bond graphs model construction procedure provided in Section 5.2. In addition, the active disturbance rejection control (ADRC) algorithm is reviewed in Section 5.3. A numerical experiment presented in Section 5.4 was conducted to quantify the performance of the candidate manipulators. The corresponding simulation results are discussed in Section 5.5. The concluding remarks are presented in Section 5.6.

5.2 Model Construction

Unlike Newton-Euler or Lagrangian formulation, constructing a multibody model with multiple kinematic loops in bond graphs does not require extensive analytic derivation of the kinematic constraints. This unique feature of bond graph formalism can be attributed to the significant contributions of the seminal research papers in [135–137]. In a pioneering work [135], Karnopp and Rosenberg developed the Eulerian Junction Structure (EJS) that represents the dynamics of a rigid body in bond graph formalism. In addition, bond graph representations of the kinematic constraints (i.e., joints) that characterize the relative motions of the bodies in a multibody system are provided in [136]. Favre and Scavarda in [137] developed the concept of *privileged frame* in order to systematize the construction of multibody models with kinematic loops in a graphic level. Employing a privileged frame minimizes the number of coordinate transformations required in a model.

Combining the ideas from [135–137] leads to a general algorithmic procedure for constructing a multibody model with kinematic loops. This procedure is comprised of the following steps:

1. Identify a privileged frame. In this case, the inertial frame is chosen as the privileged frame.
2. According to the EJS, construct the dynamic model of the center of mass (CM) of each rigid body. It should be noted that the EJS is expressed in the body fixed frame [125, p. 352].
3. Determine the *articulation points* on each body at which different joints are located according to the geometry of the manipulator. Derive the effort and the flow vectors associated with each articulation point from the EJS of the

corresponding body. In order to constrain these vectors according to the joint type, they need to be expressed in the privileged frame through appropriate coordinate transformation.

4. Construct the bond graph models of all the joint types present in the manipulator. Connect the transformed effort and flow vectors of each articulation point to the appropriate joint model through *multibonds* [126].
5. Incorporate additional dynamic systems to obtain a complete model of the manipulator; e.g., actuator dynamics, joint friction, etc.

In order to explain these steps in detail, three coordinate frames are specified; namely, the inertial, the body fixed, and the CAD coordinate frame, which are respectively denoted by A, B, and C. While the body fixed and the inertial coordinate frames are the intuitive ones, the CAD coordinate frame facilitates development of the geometries of the bodies and assists in visualizing the simulation results.

5.2.1 Eulerian Junction Structure & Articulation Points

The EJS provides the bond graph representation of the Newton-Euler rigid body dynamics model expressed in the body fixed frame [125, p. 352]. Conventionally the body fixed frame is characterized by its axes being coincident with the principal axes of inertia of the body. In addition, its origin is located at the CM of the corresponding body. Usually the geometry of the constituent bodies in a manipulator is developed in a suitable CAD application. The corresponding articulation points are defined with respect to the CAD application's native coordinate frame, which does not necessarily align with the body fixed frame. If the EJS is transferred to the CAD frame, the multibonds corresponding to the articulation points can be conveniently derived from

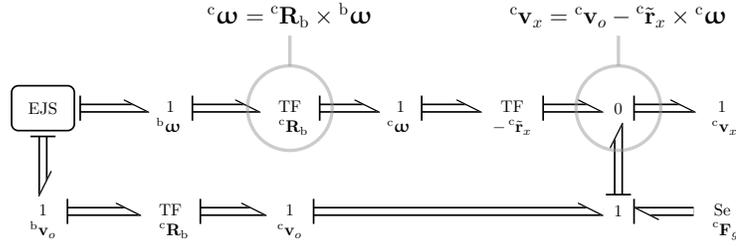


Figure 5.1: The Eulerian Junction Structure and an articulation point.

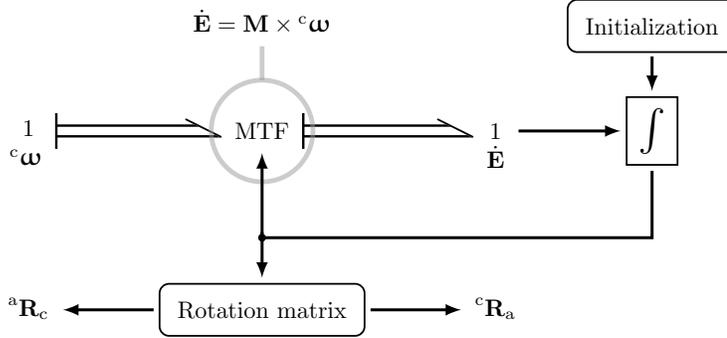


Figure 5.2: Calculating the Euler angles from the angular velocity.

the transformed EJS. Since the orientation of the body fixed frame relative to the CAD frame does not change, the appropriate coordinate transformation can be implemented by a transformer element whose modulus is provided by the fixed rotation matrix ${}^c\mathbf{R}_b$ as shown in Fig. 5.1.

Let the position vector of an articulation point x with respect to the corresponding CM be provided by ${}^c\mathbf{r}_x$. Its linear velocity is provided by

$${}^c\mathbf{v}_x = {}^c\mathbf{v}_o - {}^c\mathbf{r}_x \times {}^c\boldsymbol{\omega}. \quad (5.1)$$

Here, the linear velocity of the CM and the angular velocity of the body are respectively denoted by ${}^c\mathbf{v}_o$ and ${}^c\boldsymbol{\omega}$. The vector multiplication in (5.1) is implemented by transforming the flow vector ${}^c\boldsymbol{\omega}$ by a modulus of $-{}^c\tilde{\mathbf{r}}_x$, which is the skew-symmetric matrix corresponding to the position vector ${}^c\mathbf{r}_x$. In addition, a zero junction (see Fig. 5.1) is employed to execute the vector addition. The causal output of this zero

junction provides the linear velocity of the articulation point ${}^c\mathbf{v}_x$. In addition, the gravitational force ${}^c\mathbf{F}_g$ is appropriately augmented into the model as an effort source (see Fig. 5.1).

5.2.2 Coordinate Transformation

A typical multibody bond graph model incorporates many vectorial quantities that are expressed in different coordinate frames. For example, prior to augmenting the gravitational effort on an EJS, it must be transferred to the coordinate frame of the corresponding EJS, since the gravitational acceleration is expressed in the inertial frame. While the coordinate transformation between two frames with fixed relative orientation is simple, a more elaborate model is required when the relative orientation is continuously changing; e.g., coordinate transformation between a body fixed frame and the inertial frame. Such a coordinate transformation is implemented in bond graphs by a modulated transformer (MTF) element whose modulus is provided by the appropriate rotation matrix. A rotation matrix can be calculated from a given set of Euler angles. For this purpose, the ZYX Euler angles are employed in this chapter. These Euler angles $[\psi \ \theta \ \phi]^T$ are calculated from the angular velocity vector ${}^c\boldsymbol{\omega}$ of the body. The analytic relation between the angular velocity vector and the corresponding time derivatives of the ZYX Euler angles, as provided in (5.2), can be derived from the definition of the ZYX convention [125, p. 358].

$$\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \\ 0 & \cos \phi & -\sin \phi \\ 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \end{bmatrix} {}^c\boldsymbol{\omega} \quad (5.2)$$

Once the time derivatives of the Euler angles $\dot{\mathbf{E}} = [\dot{\psi} \ \dot{\theta} \ \dot{\phi}]^T$ are calculated from

the angular velocity vector, they can be integrated to obtain the corresponding Euler angles. It should be noted that this integration must be initialized properly so that the relative orientations of the constituent bodies are conducive to the kinematic structure. The aforementioned procedure of calculating the Euler angles and corresponding rotation matrices from the angular velocity is implemented in bond graphs in Fig. 5.2. The modulus of the MTF element in Fig. 5.2 is provided by the transformation matrix in (5.2).

5.2.3 Modeling of the Kinematic Joints

A kinematic joint defines the relative motion of the two paired bodies. In bond graph terms, a kinematic joint imposes a set of constraints on the effort and the flow vectors of the two articulation points that are contributed by each of the pairing bodies so that the desired relative motion can be achieved. The kinematic architectures being investigated feature only two types of joints; namely, spherical joints and prismatic joints. A spherical joint implies that the paired articulation points coincide and maintain identical linear velocities while the angular velocities of the joined bodies are independent of each other. Equating the corresponding linear velocities realizes the aforementioned constraints. However, assigning the linear velocity of one articulation point as a causal input to the other will result in an unfavorable causal structure in the model. In order to maintain integral causality throughout the model, a *pseudo-equalization* of the linear velocities is implemented by a stiff spring as shown in Fig. 5.3(a) where the terminal velocities of the spring corresponds to the linear velocities to be equalized. Since this parasitic spring deflects only negligibly under dynamic conditions because of its high stiffness, its terminal velocities are *effectively* equal. In order to dampen the high eigenfrequency associated with the high stiffness,

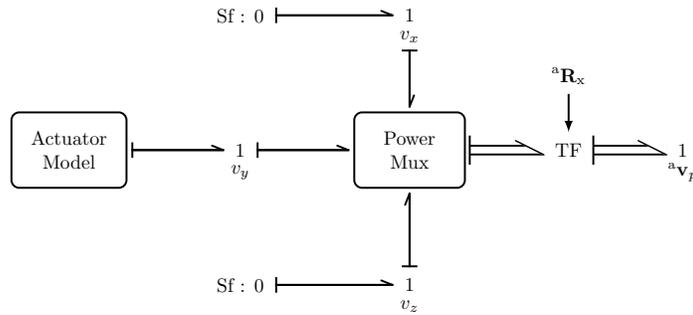


Figure 5.4: Simple prismatic joint model constraining the moving mass with respect to the mechanical ground. The joint axis is assumed to be along y axis of the X coordinate frame.

one of the axes of the CAD frame. Identical angular velocities of the joined bodies can be modeled in a manner similar to a spherical joint where two linear velocities are constrained [see Fig. 5.3(b)]. Without loss of generality, the flow vector corresponding to either of the articulation points can be chosen as the causal input to the joint model. From this causal input, as shown in Fig. 5.3(c), the multibond associated with the other articulation point can be derived. In Fig. 5.3(c), the y axis of the CAD frame of the joined bodies has been arbitrarily chosen as the joint axis and the effort source corresponds to the actuator input. In the case of a passive prismatic joint, this source provides zero effort. The modulus of the MTF element in Fig. 5.3(c) is provided by the skew-symmetric matrix associated with the relative position vector of the two articulation points. It should be noted that the general bond graph model of a prismatic joint is not strictly necessary for constructing the models of the candidate manipulators. The fact that the prismatic joint in the candidate manipulators constrains the moving coil with respect to the mechanical ground provided an opportunity for a simple modeling approach, which is illustrated in Fig. 5.4. The modulus ${}^a\mathbf{R}_x$ of the TF element in Fig. 5.4 provides the coordinate transformation between the X coordinate frame and the inertial coordinate frame (i.e., privileged frame). It can be used to model the tilting of the actuator axis.

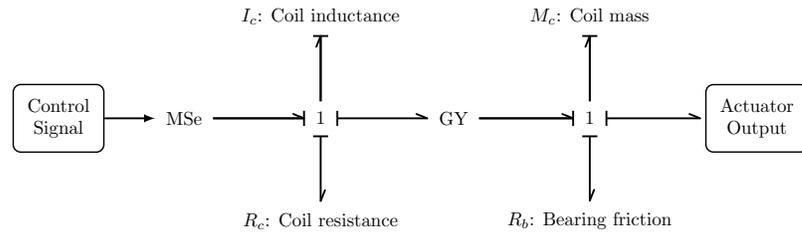


Figure 5.5: Bond graph model of the voice coil actuator.

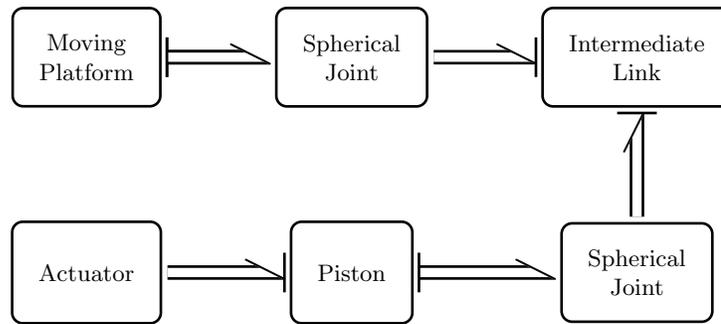


Figure 5.6: Complete bond graph model of a single kinematic loop of the parallel and the slanted configurations of the 3-PSS/S manipulator.

5.2.4 Complete Model

The linear voice coil actuator described in Chapter 3 was selected as the motion input device for the candidate manipulators. In order to obtain an elaborate bond graph representation of the actuator, the model of a generic voice coil motor in [139] was incorporated with coil resistance, mass, and bearing friction (see Fig. 5.5). Once the individual models of the joints, the bodies, and the actuators are constructed, they can be connected together according to the respective kinematic structure in order to obtain a complete dynamic model of each manipulator. A high level bond graph model of a representative kinematic loop of the candidate manipulators is shown in Fig. 5.6. Three such loops constitute the complete model of each architecture. It should be noted that the structures of the dynamic models of the parallel and the slanted configurations of the 3-PSS/S architectures are identical. Besides the model

parameters, they differ in the direction of the axis of the actuated joints.

5.3 Active Disturbance Rejection Control

In order to study the dynamic performance of the candidate manipulators, each dynamic model constructed in the previous section must be paired with an appropriate controller. A multi-loop motion controller based on the active disturbance rejection control (ADRC) technology is formulated to this end. An initial step in any controller prototyping exercise involves identification of a set of the system variables that are to be controlled (i.e., system output). In addition, a complementary set of system variables must be selected that are required to be manipulated in order to achieve the control objective (i.e., system input). In this case, the objective is to control the positions of the linear actuators by manipulating the corresponding actuator voltages so that the desired orientation of the payload can be achieved. Accordingly, the position sensor signals and the actuator input voltages respectively constitute the inputs and the outputs of the controller. Since each input to this MIMO system exhibits an obvious pairing with a unique output (i.e., input voltage and position of each actuator), the corresponding ADRC controller can be constructed by designing three individual SISO (single input single output) controllers for each input-output pair [140]. The position of the actuated joint in each of three kinematic loops is managed by a SISO controller.

In this chapter, the formulation of a general control problem in [19] is adopted in order to provide an overview of the ADRC algorithm. A highly abstract model of a representative kinematic loop of the candidate manipulators is a second order system

of the following structure,

$$\ddot{y}_i = f_i \left(y_i, \dot{y}_i, \zeta_i, \dot{\zeta}_i, w_i(t), t \right) + b_i u_i, \quad (5.3)$$

where y_i denotes output of the i -th loop (i.e., actuator position), t denotes time, b_i is a system parameter, and u_i is the input to the i -th loop ($i = 1, 2, 3$). In addition, ζ_i denotes a vector of a comprehensive set of dynamic variables (excluding y_i) originating from the system model. $f_i(\cdot)$ represents the cross-coupled, nonlinear loop dynamics including the external disturbance $w_i(t)$. It should be emphasized that $f_i(\cdot)$ does not decouple i -th loop dynamics from the other two kinematic loops. It is merely a lumped estimation of the aggregate effects of loop dynamics, cross-coupling, and external disturbance. It should also be noted that explicit knowledge of $f_i(\cdot)$ is not necessary for ADRC. The system in (5.3) can be rewritten in state-space as in (5.4) with an augmented state $x_{3i} = f_i(\cdot)$.

$$\begin{aligned} \dot{x}_{1i} &= x_{2i} \\ \dot{x}_{2i} &= x_{3i} + b_i u_i, \quad x_{3i} = f_i(\cdot) \\ \dot{x}_{3i} &= \dot{f}_i(\cdot) \\ y_i &= x_{1i} \end{aligned} \quad (5.4)$$

In the context of feedback control, the *total disturbance* $f_i(\cdot)$ needs to be overcome by the control effort u_i in order to achieve desired behavior of y_i [19]. To this end, an extended state observer (ESO) for the system in (5.4) can be constructed in the

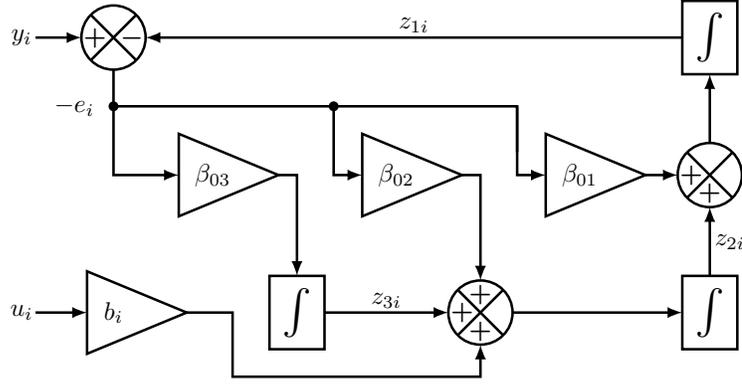


Figure 5.7: Structure of a linear extended state observer.

following form.

$$\begin{aligned}
 e_i &= z_{1i} - y_i \\
 \dot{z}_{1i} &= z_{2i} - \beta_{01}e_i \\
 \dot{z}_{2i} &= z_{3i} + b_i u_i - \beta_{02}\vartheta_{1i}(e_i) \\
 \dot{z}_{3i} &= -\beta_{03}\vartheta_{2i}(e_i)
 \end{aligned} \tag{5.5}$$

Here, z_{1i} , z_{2i} , and z_{3i} are respectively the observations of x_{1i} , x_{2i} , and x_{3i} . In addition, β_{01} , β_{02} , and β_{03} are the tuning parameters for the ESO. $\vartheta_{1i}(e_i)$ and $\vartheta_{2i}(e_i)$ are appropriate linear or nonlinear functions of the tracking error e_i . In this case, they were chosen as $\vartheta_{1i}(e_i) = \vartheta_{2i}(e_i) = e_i$ in order to linearize the ESO. The structure of the linear ESO is illustrated in Fig. 5.7.

The observation of $f_i(\cdot)$ provided by the ESO allows the control law $u_i = \frac{u_{0i} - f_i(\cdot)}{b_i} \approx \frac{u_{0i} - z_{3i}(\cdot)}{b_i}$ to be applied on the system in (5.4), which ultimately results in a reduced system of cascade integral form as in (5.6).

$$\dot{x}_{1i} = x_{2i}, \quad \dot{x}_{2i} = u_{0i}, \quad y_i = x_{1i} \tag{5.6}$$

This reduced system can be controlled by calculating u_{0i} as a function of the tracking error and its derivate (i.e., a PD controller). In addition to the central idea of

estimating and rejecting disturbances, ADRC topology also features a few other components that are discussed below.

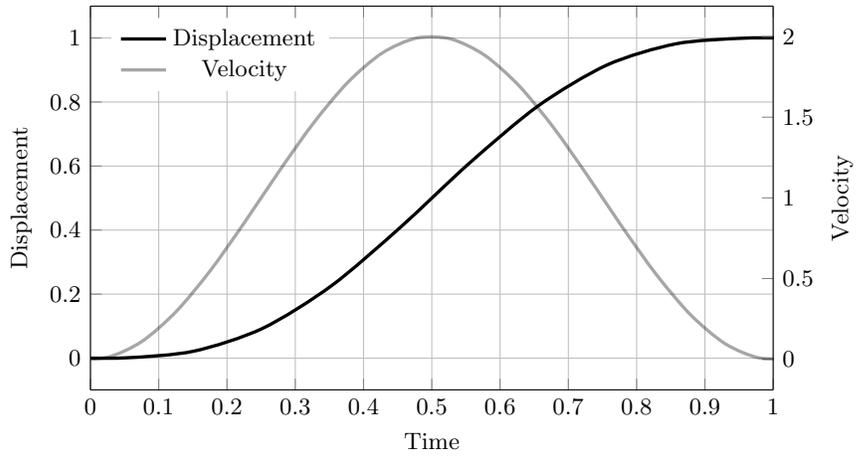
5.3.1 Transient Profile

Each SISO loop of the controller is augmented with a transient profile generator producing a trajectory that the actuators can follow without exceeding their actuation capacities. Although a number of methods for generating transient profiles are available in the literature, only the limited jerk (time derivative of acceleration) profiles are of interest in the context of parallel manipulators. The limited jerk transient motion profile improves tracking performance, and reduces mechanical impact [141]. It should be noted that this thesis does not implement trajectory planning in the workspace coordinates because the classic ADRC formulation involves an empirical model of system dynamics [i.e., see (5.3)] in terms of the input (e.g., actuator voltage) and the measured output of the plant (e.g., actuator position). Correspondingly, trajectory planning in the joint space coordinates is considered to be more conducive to the structure of the ADRC controller and is the approach adopted in this thesis.

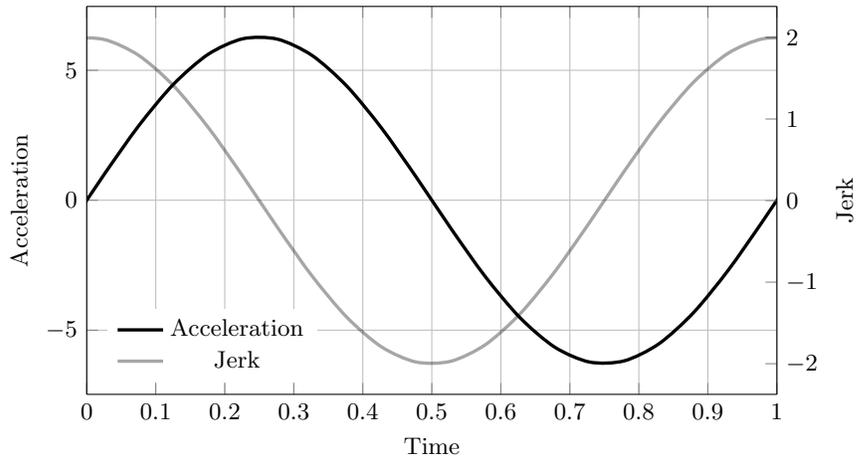
A transient trajectory generating limited jerk can be modeled by a sinusoidal acceleration profile that produces a cycloidal displacement [142] as shown in Fig. 5.8. The displacement, and the corresponding velocity of the cycloidal profile are provided as a function of time in (5.7) and (5.8).

$$y_p(t) = (y_f - y_j) \left(\frac{t}{t_f} - \frac{1}{2\pi} \sin 2\pi \frac{t}{t_f} \right) + y_j \quad (5.7)$$

$$v_p(t) = \frac{y_f - y_j}{t_f} \left(1 - \cos 2\pi \frac{t}{t_f} \right) \quad (5.8)$$



(a) Position and velocity profiles.



(b) Acceleration and jerk profiles.

Figure 5.8: Cycloidal motion for a unit displacement in unit time.

Here, y_f is the final position (i.e., desired set-point), y_j is the initial position, t is the current time ($t = 0$ at the onset of motion), and t_f is the *transient time* (i.e., at time $t = t_f$ the position trajectory reaches y_f).

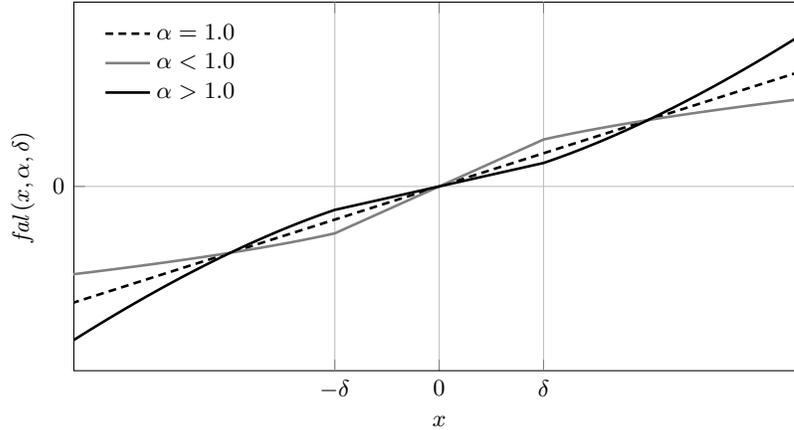


Figure 5.9: Characteristics of the nonlinear weighting function $fal(\cdot)$.

5.3.2 Nonlinear PD Controller

ADRC employs a nonlinear weighted sum (n-PD) of the position and velocity tracking errors in order to calculate u_{0i} as follows,

$$u_{0i} = k_p \text{fal}(e_{pi}, \alpha_p, \delta_p) + k_d \text{fal}(e_{vi}, \alpha_v, \delta_v). \quad (5.9)$$

Here, e_{pi} and e_{vi} are respectively the position and the velocity tracking error for the i -th kinematic loop. $fal(\cdot)$ is a nonlinear function provided by (5.10), and parameterized with α and $\delta > 0$.

$$\text{fal}(e, \alpha, \delta) = \begin{cases} \frac{e}{\delta^{1-\alpha}}, & |e| \leq \delta \\ |e|^\alpha \text{sign}(e), & |e| > \delta \end{cases} \quad (5.10)$$

Gao *et al.* in [143] provided a detailed explanation on how the $fal(\cdot)$ function improves controller performance by guarding against control signal saturation. It can be easily seen that the nonlinear PD control law in (5.9) becomes linear for $\alpha_p = \alpha_v = 1.0$. The characteristics of the $fal(\cdot)$ function is provided in Fig. 5.9. The nonlinear function $fal(\cdot)$ with $\alpha_p < 1.0$ provides a high gain for a small position tracking error ($|e_{pi}| \leq \delta_p$).

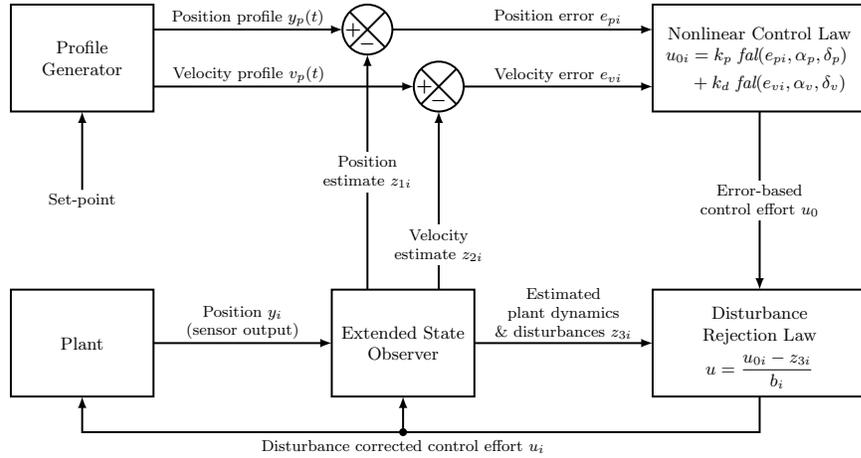


Figure 5.10: Topology of ADRC for a single kinematic loop.

This aids the minimization of steady state error without integral control. Furthermore, saturation of the control signal for a large position error ($|e_{pi}| > \delta_p$) is avoided by limiting the corresponding gain (Fig. 5.9). When the plant transitions from one set-point to another, the velocity tracking error e_{vi} may become unfavorably large. If the control signal is adjusted accordingly, a sudden spike in the plant input may result in unstable conditions. Such an occurrence is avoided by setting $\alpha_v > 1.0$ for the velocity tracking error.

5.3.3 Tuning of ADRC

Appropriately assembling the transient motion profile generator, the extended state observer, the nonlinear PD control law, and the disturbance rejection law together provides the ADRC that can be applied for the position control of the actuated joints. The different components of this controller are illustrated graphically in Fig. 5.10. Three such control loops constitute the complete ADRC that is suitable for each candidate manipulator. A general continuous time implementation of the ADRC controller was employed for the two study cases; namely, the parallel and the slanted

configurations of the 3-PSS/S manipulator.

A simple tuning method for second order ADRC in [144] determines the n-PD, and the ESO gains as a function of the controller bandwidth ω_c , and the observer bandwidth ω_o respectively [145]. In addition, the tuning procedure of the system parameter b_i in (5.3) is also provided in [144]. Since a large controller bandwidth ω_c demands fast dynamics from the system, the response tracks the set-point aggressively. However, this may lead to undesirable oscillations or instability because the system may be marginally capable or even completely incapable of delivering the dynamics demanded by the controller. Hence, the tuning exercise of ω_c attempts to find an acceptable compromise between requirements of performance and stability margin [92]. Similarly, a large observer bandwidth ω_o improves tracking performance of the ESO, which is achieved at the cost of a degraded noise tolerance. Thus, ω_o is tuned to balance tracking performance against the noise sensitivity of the ESO [90]. According to [145], the controller gains in (5.9) are determined as,

$$k_p = \omega_c^2, \quad k_d = 2\zeta\omega_c. \quad (5.11)$$

The damping ratio ζ in the above equation was introduced to avoid oscillations in the system response. In addition, the observer gains $[\beta_{01}, \beta_{02}, \beta_{03}]$ for a continuous time ESO are chosen as in (5.12). These tuning parameters place the poles of the observer at $-\omega_o$.

$$\beta_{01} = 3\omega_o, \quad \beta_{02} = 3\omega_o^2, \quad \beta_{03} = \omega_o^3. \quad (5.12)$$

5.4 Dynamic Performance Evaluation

In order to simulate realistic manipulator performance, appropriate model parameters were used to construct the dynamic models of the candidate manipulators. Most of the dynamic parameters for the actuator model were estimated from a physical prototype (see Table 5.1). In addition, the inertia parameters of the moving bodies were calculated from appropriate CAD models (refer to Table 5.2). The corresponding geometries of the moving bodies were primarily based the optimal dimensions determined in Chapter 4. The friction in the spherical and the universal joints were assumed to be negligible. However, a reasonable value for the linear bearing friction was considered. It is relevant to mention that the moving platform inertia in Table 5.2 includes a 50 g cylindrical payload of radius 5.0 mm and height 40.0 mm. Furthermore, the actuators were assumed to be powered by an energy source that provides a maximum voltage output of ± 24 V. The complete models with appropriate dynamic parameters were then used for tuning the corresponding controllers. All controller parameters in Table 5.3 were determined according to the tuning procedure described in Section 5.3.3.

Since the candidate manipulators use the same prismatic actuator, the benchmarking study was conducted in joint space coordinates. All the manipulator models were simulated to execute a series of two hundred robotic maneuvers. Each maneuver was defined by three set-points for the three actuated joints. It should be noted that the set-points were expressed in terms of a 16-bit ADC (analog to digital converter) scale that maps the bottom extreme position of the actuator to zero and the top extreme position to $2^{16} - 1 = 65535$. The set-points were sampled randomly from the experimental space [5000, 60000] ADC counts in order to minimize the possibility of hitting the two hard stops at either end of the actuator stroke during experimental trials. In addition, the displacement of an actuator in a test maneuver was constrained

Table 5.1: Model parameters for the actuator

Parameter	Value
Motor constant	6.9478 N/amp
Back EMF constant	6.9478 V-s/m
Coil resistance	20.44 Ω
Coil inductance	1.27E-03 H
Moving mass	33.0 g
Linear bearing friction ^a	9.81E-02 Ns/m
Sensor gain ^a	5.46125E+06 ADC counts/m

^aassumed for simulation.

Table 5.2: Inertia parameters for the candidate manipulators

Manipulator	Link	Mass (Kg)	Principal Moments of Inertia (Kg·m ²)		
			I_x	I_y	I_z
3-PSS/S (parallel)	Moving platform	6.50E-02	2.40E-06	1.44E-05	1.44E-05
	Intermediate link	5.00E-03	1.35E-08	1.13E-06	1.13E-06
	Actuator piston	3.30E-02	-	-	-
3-PSS/S (slanted)	Moving Platform	6.61E-02	3.09E-06	1.50E-05	1.50E-05
	Intermediate link	4.53E-03	1.29E-08	7.62E-07	7.62E-07
	Actuator piston	3.30E-02	-	-	-

Table 5.3: Controller tuning parameters for each kinematic loop of the candidate manipulators

Parameter	Unit	3-PSS/S (parallel)	3-PSS/S (Slanted)
Observer bandwidth, ω_o	rad/s	2600	2600
Controller bandwidth, ω_c	rad/s	15.0	20.0
Damping ratio, ζ	-	1.0	1.0
System parameter, b_i	m/(s ² V)	1.12E+07	1.05E+07
Nonlinear gain parameter, α_p	-	0.75	0.75
Nonlinear gain parameter, δ_p	ADC counts	500	500
Nonlinear gain parameter, α_v	-	1.45	1.75
Nonlinear gain parameter, δ_v	ADC counts/s	6.50E+04	6.50E+04

to be at least 1000 ADC counts in magnitude so that no trivial sample was included in the results. In order to ensure a fair comparison, the simulation exercises for all three candidate manipulators were conducted employing the same set of randomly generated maneuvers. The time interval between two consecutive maneuvers were chosen to be 200 ms in order to allow a sufficiently large monitoring window.

5.4.1 Transient Motion Profile Generation

Although motion planning algorithms providing a time-optimal trajectory that accounts for the PKM dynamics have been proposed in the literature (e.g., [146]), the potential gain in speed from an *in situ* implementation of such an algorithm on an embedded control hardware may not justify the associated resource cost. Alternatively, this thesis has adopted a heuristics based method to determine the transient time for a general robotic maneuver. To this end, a motion defined by a position profile oscillating about the mid-stroke of the actuator was prescribed on a randomly chosen joint controller. Because of the cross-coupling effects, active position control of the other two actuated joints was required to maintain their home positions (i.e., mid-stroke positions). A peak-to-peak sinusoidal displacement of 6 mm (i.e., half of the actuator stroke) characterized the position trajectory. As the frequency of the prescribed motion was increased monotonically, it was observed that the control effort required for the oscillating motion eventually exceeded the actuation capacity provided by the given power source (i.e., ± 24 V). The maximum frequency f of the sinusoidal motion that can be achieved by the actuator without saturating the voltage input at steady-state conditions was determined by a trial and error exercise. Since the instantaneous velocities are zero at the peaks, $t_{min} = 2f^{-1}$ was estimated to be the minimum time required for an actuator to move between the extremes of its

range (i.e., S_{max}) without exhausting its capacity. Correspondingly, the *achievable* transient time for a general displacement S is calculated as, $t_f = t_{min} \times (\frac{S}{S_{max}} + c)$. Here, the constant c accounts for the linear assumption made in estimating t_{min} and the cross-coupling effects of a general robotic maneuver. Although calculating c or t_f as a function of the dynamics of the manipulator constitutes an interesting research concept, it is deferred to a future work. However, a large value of c is not conducive to high speed performance. On the other hand, a low value of c may generate motion profiles that are not achievable with the limited actuation capacity. As a compromise between speed and performance, a reasonable value of $c = 0.1$ was selected for the two test cases. This empirical method of designing the transient motion profile is admittedly not ideal. Nevertheless, it was regarded to be sufficient and feasible for an embedded deployment of the controller. Correspondingly, the benchmarking study employed this empirical method to generate the transient motion profiles for the test maneuvers. It is also relevant to mention that a more intuitive amplitude providing a peak-to-peak displacement representing the entire actuator stroke was not chosen to eliminate the possibility of the moving coil hitting the mechanical stops at the two extremes. Since a robotic maneuver was considered to have been executed once all three actuators were settled, one actuator reaching its set-point before the other two provides no advantage. Therefore, the corresponding displacements were synchronized by adopting an identical transient time t_f for all three actuators. The largest of three displacements provided the transient time for all three actuators. Such a synchronized movement of the actuators minimized the cross-coupling nonlinearities [74].

5.4.2 Performance Indices

The transient responses of the candidate manipulators while executing the test maneuvers were recorded in order to benchmark their dynamic performances. Correspondingly, the speed of each manipulator was measured in terms of the settling time of each actuator response during the test maneuvers. The largest of the three settling times provided the execution time for an experimental maneuver. In order to measure the accuracy of the test displacements, the corresponding steady state errors served as an additional set of performance metrics. Since the energy consumption in the robotic maneuvers provides a comparative measure of effectiveness in terms of energy utilization, it was considered as an index for *manipulation efficiency* in the benchmarking study. Correspondingly, the expended energy in an experimental maneuver is provided by the sum of the energy consumed by all three actuators. The consumed energy by each actuator in an experimental trial was estimated by taking the integral of the effort and the flow vector associated with the modulated effort source in the actuator model (see Fig. 5.5) over the entire monitoring window.

The settling time is defined by the time elapsed from the initiation of a test displacement to the time when the system response (i.e., position of an actuator) has entered and remained within $\pm 30 \mu\text{m}$ of the steady state value. The method of moving variance in [147] was employed to detect steady state in the time-series of the system response. In this chapter, the system was considered to have entered steady state when the moving variance of the system response for a sampling window of 10 ms first becomes less than the variance threshold of 737.7 squared ADC counts and subsequently maintains this statistical property for the remainder of the monitoring time. According to this definition, all system responses in each sample window after the system has reached steady state should be within $\pm 10.0 \mu\text{m}$ of the respective mean

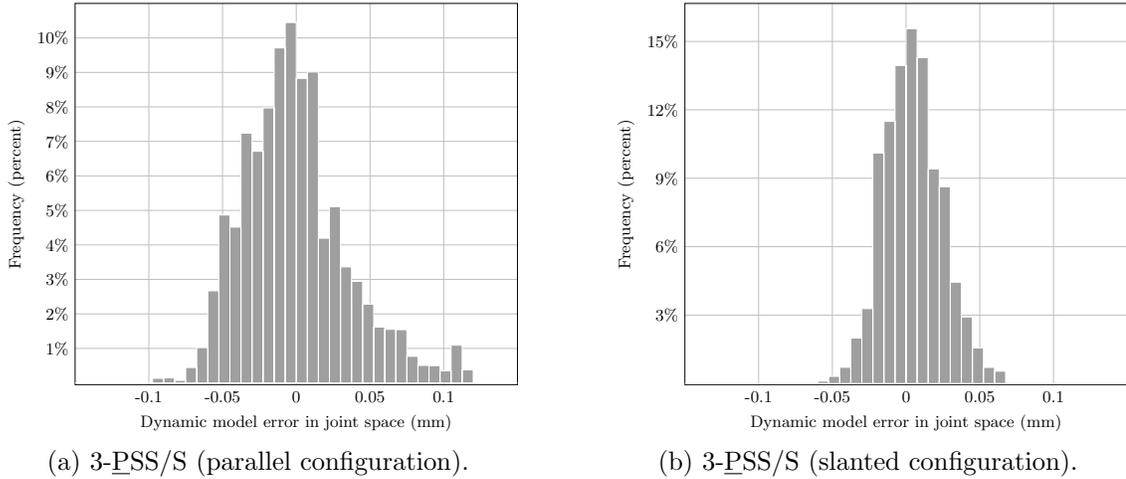


Figure 5.11: Model verification through the application of the inverse kinematics models of the candidate manipulators.

value with a 95% C.I., assuming the underlying statistical distribution is Gaussian. Correspondingly, the steady state value was determined by taking the mean of all system responses sampled from the time the system entered steady state to the time when window of monitoring expired. The steady state error is provided by the absolute difference between the steady state value and the desired final position. Based on the aforementioned definitions, the performance metrics provided in Table 5.4 were calculated from the experimental data.

5.4.3 Model Verification

In order to ascertain the mathematical correctness of the dynamic models, the experimental data was evaluated against the kinematics of the candidate manipulators. To this end, the orientation of the moving platform, as provided by the dynamic simulation, was employed to determine the actuated joint coordinates using the appropriate inverse kinematics model. In a subsequent step, the joint coordinates thus obtained were compared against those provided by the dynamic simulation. The corresponding

Table 5.4: Dynamic performance of the candidate manipulators while executing the test maneuvers

Manipulator	Statistic	Execution time (ms)	Steady state errors (μm)			Energy (J)
			1	2	3	
3-PSS/S (parallel)	Mean	60.7703	0.7696	0.8387	0.8840	0.1609
	Max	99.5185	3.4335	4.4607	3.0466	0.6017
	Min	20.0383	0.0139	0.0006	0.0053	0.0380
	σ	18.9103	0.6682	0.7413	0.6786	0.0921
3-PSS/S (slanted)	Mean	66.5947	0.6993	0.7584	0.7987	0.1567
	Max	108.9394	3.8722	3.4039	3.8729	0.5668
	Min	21.8446	0.0009	0.0149	0.0019	0.0242
	σ	20.8116	0.6936	0.7131	0.6698	0.0959

difference provides a means of validating the models. Since the volume of the experimental data was massive, only 10000 random samples from each dynamic simulation data were considered for this exercise. The error histograms of these samples are provided in Fig. 5.11. The sources of the errors can be identified as the inclusion of the parasitic elements, integration error, imperfect estimation of the geometry and the center of masses of the moving bodies by the CAD application etc. Nonetheless, the dynamic models were regarded to be satisfactorily accurate, as indicated by the histograms provided in Fig. 5.11.

5.4.4 Experimental Observations

The system responses and the controller variables in an identical test maneuver executed by the parallel and the slanted configuration of the 3-PSS/S manipulator are respectively shown in Fig. 5.12 and Fig. 5.13. This test maneuver featured the largest execution time in the simulation study. The corresponding angular motion exhibited by the moving platform of each candidate manipulator is provided in Fig. 5.14. In addition, the statistics of the simulated performance metrics defined in Section 5.4.2 is presented in Table 5.4.

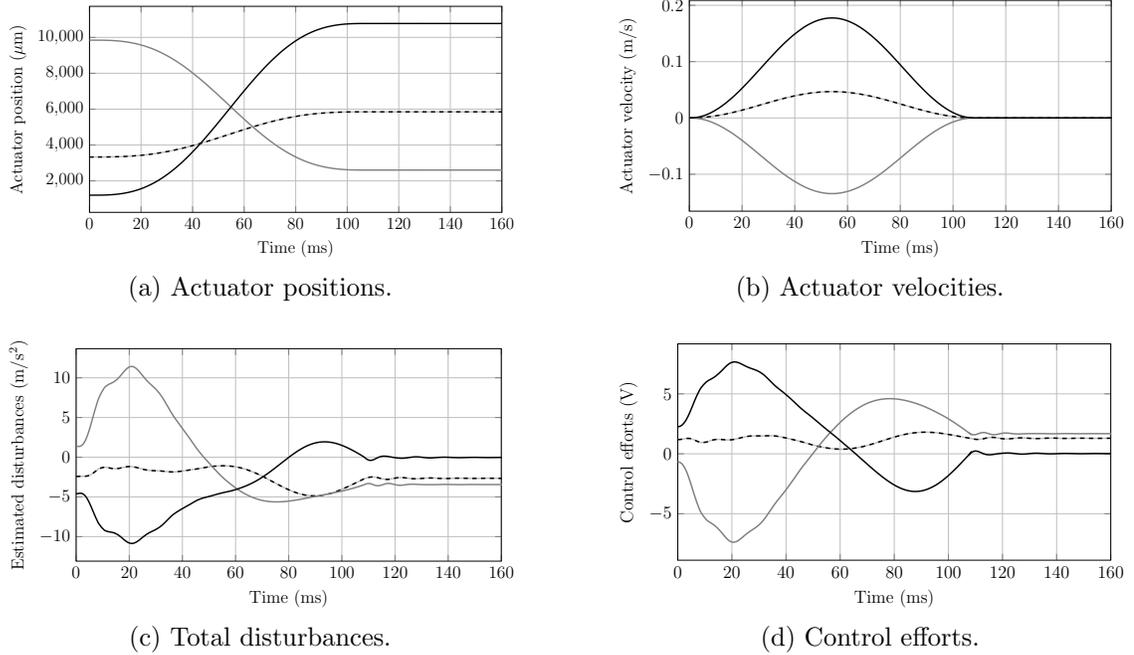


Figure 5.12: System responses in a representative maneuver executed by the parallel configuration of the 3-PSS/S manipulator (--- Actuator 1, — Actuator 2, — Actuator 3).

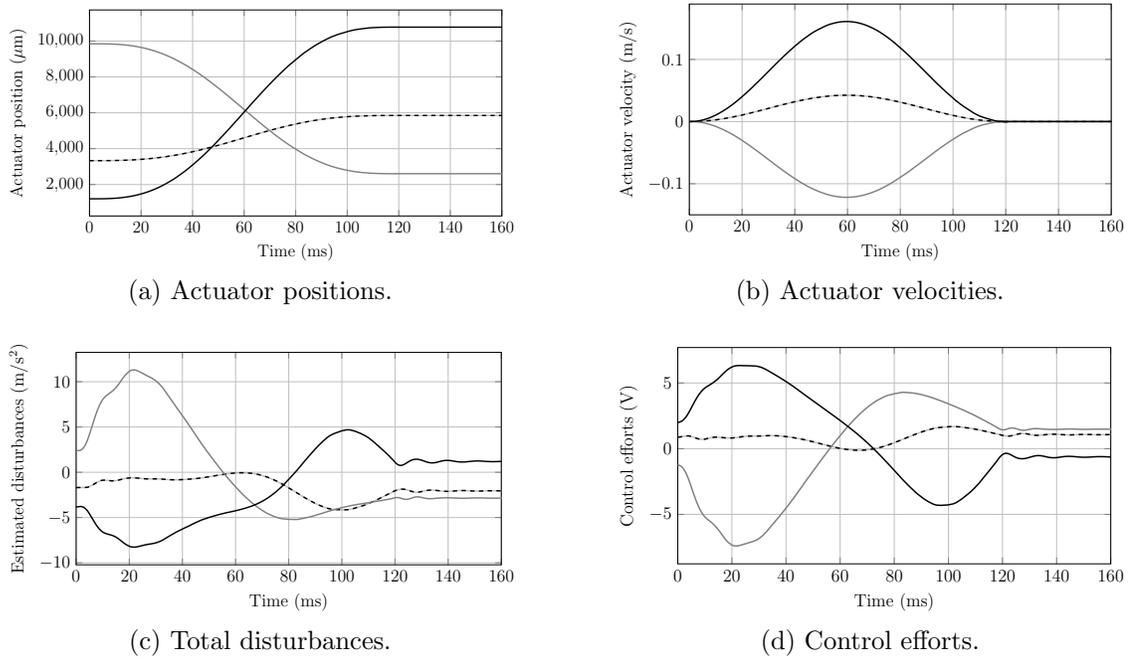


Figure 5.13: System responses in a representative maneuver executed by the slanted configuration of the 3-PSS/S manipulator (--- Actuator 1, — Actuator 2, — Actuator 3).

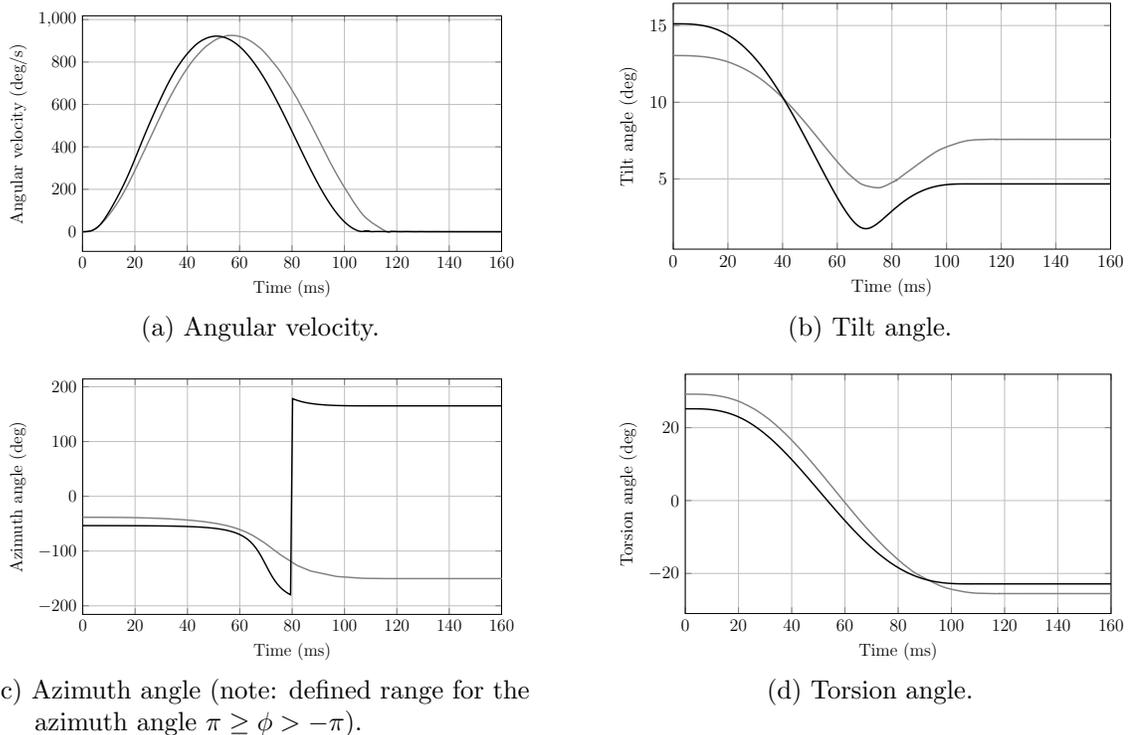


Figure 5.14: Angular motion of the moving platform in a representative maneuver executed by the candidate architectures (— slanted configuration, — parallel configuration).

5.5 Discussion

The empirical procedure in Section 5.4.1 was employed for designing the transient motion profiles for the test maneuvers. The minimum time t_{min} required for one actuator to move between the extremes of its range with zero terminal velocities was determined to be 120.12 ms for the parallel configuration and 131.58 ms for the slanted configuration. This discrepancy can be explained by the fact that the reachable workspace volume of the slanted configuration is more than 1.4 times larger than the parallel configuration. As a result, the corresponding moving platform goes through a larger angular motion in response to the full scale displacement of a single actuator. Nonetheless, the system responses shown in Fig. 5.12 and Fig. 5.13 indicate

excellent tracking performance provided by the ADRC controller. The data presented in Table 5.4 show that both manipulators provide comparable dynamic performance. A series of two sample t-test at the 5% significance level confirmed that the performance exhibited by the candidate manipulators in terms of the steady state errors in the test maneuvers and the associated energy expenses have equal means with equal variances. In other words, both controllers showed statistically equal performance in terms of accuracy and manipulation efficiency. However, the two samples of the execution times were observed to be statistically different in the corresponding t-test, which is consistent with the difference in the workspace volumes of the candidate manipulators.

5.6 Conclusion

Dynamic simulation plays an important role in the design flow of any modern engineering system. In this chapter, dynamic simulation was employed to quantify the dynamic characteristics of the two selected manipulators. To this end, a number of ideas related from previous work was amalgamated in the form of an algorithmic procedure for synthesizing a closed-loop multibody dynamic model in the bond graphs formalism. Subsequently the constructed multibody dynamic models were employed in a Monte-Carlo study, which confirmed that the ADRC technology can provide robust motion control performance for the candidate manipulators.

Chapter 6

FPGA Implementation of ADRC Technology

Due to its superior speed, parallelism and suitable SWaP (size, weight, and power) characteristics, the Field Programmable Gate Array (FPGA) is favored as a reliable control hardware platform for time-critical applications, especially in the military and the aerospace domain. However, deploying a modern control technology like active disturbance rejection control (ADRC) on FPGA hardware can be a resource intensive exercise that requires expensive, high logic density FPGA chips. In order to formulate a cost efficient solution, an alternative implementation of the ADRC algorithm is developed in this chapter, which can be deployed on digital hardware at a fraction of the resource cost. It was experimentally demonstrated that the resource efficient formulation does not compromise the control performance. A prototype of the proposed voice coil actuator (VCA) was employed as the experimental plant and the corresponding controller was composed of a single feedback loop. In contrast, three such feedback loops comprise the ADRC controller for the 3-PSS/S manipulator. The design of this controller was tested in a co-simulation approach so that the corresponding digital circuits can be verified prior to hardware deployment. To this end, the digital implementation of the controller and an appropriate dynamical model of the parallel manipulator were simulated concurrently employing two different simulators. A bi-directional communication link between the control hardware simulator and the multibody dynamics simulator ensured that the controller was being tested under realistic conditions. Subsequently, the results obtained from the co-simulation experiment confirmed the validity of the digital design.

6.1 Introduction

An active disturbance rejection controller (ADRC) for the 3-PSS/S manipulator features three input-output channels (i.e., the input voltage and the position sensor output of each actuator) that regulate the motion of the three limbs. Latency among the input channels is undesirable, since it may compound the adverse effect of inter-channel coupling further. Ideally a feasible control hardware platform must be able to execute the control algorithm at a sufficiently high frequency in such a manner that all inputs to the system are dispatched simultaneously. Among all the available choices, the field programmable gate array (FPGA) offers the speed and the parallelism required for a successful implementation in regards to the desired application. Moreover, an FPGA simplifies the prototyping process by integrating control circuitry for multiple input-output channels, ancillary communication modules, and soft processors for solving the kinematics problem on a single chip. Thus, the FPGA is identified as the most suitable platform for implementing ADRC for a high speed manipulator. In addition to these advantages of an FPGA hardware, it generally offers superior reliability and robustness without the requirement of any additional cooling. In contrast, while a powerful multi-core processor may overcome the challenge of achieving parallelism by evaluating the control algorithm at a high speed utilizing its multiple CPU cores, it is considered unsuitable for embedded applications because of reliability and robustness issues that originate from the software overhead (e.g., operating system) and the power requirements. However, developing a control system for an FPGA target is essentially an exercise in digital circuit design, which renders this task to be entirely different from developing computer codes. While software codes are typically executed on a general purpose processor in a sequential manner, a controller implemented on an FPGA chip may have a number of digital circuits

that function in parallel. As governed by the underlying design, each of these circuits is dedicated to a certain task, which ensures simultaneous execution of all functions regardless of their criticality in realizing the control objective. In consequence, the system stability is greatly enhanced, since no high-priority task (e.g., mathematical operations for evaluating the control algorithm, dispatching control signals) ever remains unattended to facilitate the execution of a low-priority task (e.g., data storage, communication). Correspondingly, an FPGA implementation is preferred to develop a highly reliable, time-critical control system that also meets the stringent SWaP (size, weight, and power) requirements for the military and the aerospace domains [148]. Previous examples of ADRC algorithm deployed on an FPGA include [90] and [149]. While [90] employed single precision floating point formats in order to accommodate the large magnitudes of the controller parameters, a fixed point implementation was used in [149] so that a high sampling rate of the ADRC algorithm can be achieved. In contrast, this paper proposes a resource optimized formulation of the ADRC algorithm, which allows multiple control loops to be populated on a single low logic density FPGA chip to provide a cost effective solution. In addition, the nonlinear control law in the ADRC algorithm was evaluated on an FPGA hardware in a resource efficient manner by adopting a piecewise linear approximation scheme. Although the proposed formulation was primarily developed for an FPGA platform, it can be adopted for any traditional control hardware.

Although mathematical correctness of an FPGA design can be verified in simulation under arbitrary test cases, the dynamics of the system, which is a function of the inputs, and the past states, is hardly random. Therefore, the controller must be verified and evaluated under dynamic conditions, which involves either the physical system, or a corresponding simulation model. Since hardware testing of an unverified controller is difficult without access to all the state variables, this chapter adopts a

co-simulation approach wherein the manipulator model and the FPGA implementation of the controller are respectively simulated in a conventional simulator (e.g., MATLAB-Simulink) and an HDL (hardware description language) simulator (e.g., Modelsim). Both of the simulators communicate with each other for synchronization and data exchange through the FLI (foreign language interface) feature of the HDL simulator. This approach enhances the visibility of the signals/quantities internal to the FPGA design, which is otherwise unobtainable from an FPGA and the real system. It should be noted that comprehensive access to these quantities is necessary for the sake of timing and mathematical verification, debugging, tuning, performance evaluation, etc. Moreover, HDL simulators can provide one to one correspondence in terms of input-output relations with the physical embodiment of the digital design on an FPGA chip provided that timing issues are completely resolved by the synthesizer (e.g., Quartus II). As a result, the controller running in an HDL simulator *exactly* represents the behavior of the FPGA hardware. However, the system model still suffers from uncertainties arising from unmodeled dynamics and inaccurate parameter estimations, which ADRC can handle [93, 144, 150].

The remainder of this chapter is organized as follows. The resource optimized formulation of the ADRC algorithm is developed in Section 6.2. Section 6.3 details the implementation procedure of the proposed ADRC algorithm on FPGA hardware. The control performance achieved by the proposed algorithm was evaluated in Section 6.4 employing a prototype of the proposed VCA as the experimental plant. In addition, a simulation study was conducted in order to assess the performance of the proposed ADRC algorithm in terms of regulating the motion of the parallel configuration of the 3-PSS/S manipulator in Section 6.5. The experimental and the simulation results are discussed in Section 6.6. Finally, the concluding remarks are offered in Section 6.7.

6.2 Resource Optimized Formulation of the ADRC Algorithm

The discrete implementation of a single ADRC loop provided in [19] serves as a point of reference for the derivation of the resource optimized formulation. The discrete algorithm from [19] is recalled in (6.1) and henceforth referred to as the classic implementation of ADRC.

$$\begin{aligned}
 e(k) &= z_1(k) - y(k), \quad \sigma(k) = T(z_3(k) + bu(k)) \\
 z_1(k+1) &= z_1(k) + Tz_2(k) - \beta_{01}e(k) \\
 z_2(k+1) &= z_2(k) + \sigma(k) - \beta_{02}e(k) \\
 z_3(k+1) &= z_3(k) - \beta_{03}e(k) \\
 y_p(k) &= f_p(k), \quad v_p(k) = f_v(k) \\
 e_p(k) &= y_p(k) - z_1(k), \quad e_v(k) = v_p(k) - z_2(k) \\
 u_0(k) &= k_p \text{fal}(e_p(k), \alpha_p, \delta_p) + k_v \text{fal}(e_v(k), \alpha_v, \delta_v) \\
 u(k) &= \frac{u_0(k) - z_3(k)}{b}
 \end{aligned} \tag{6.1}$$

In (6.1), $y(k)$ is the position feedback of the plant at any discrete time k , e is the tracking error, z_1 is an estimation of y provided by the ESO, z_2 is the estimated plant velocity, z_3 is the observed total disturbance and T is the sampling period of the controller. The tuning parameters of the ESO (i.e., β_{01} , β_{02} , β_{03} , and b) are specific to plant dynamics and control objectives. The nonlinear function f_p provides a transient position trajectory that the plant can reasonably follow to reach the desired position without exhausting its actuation capacity. In addition, v_p provides the transient velocity profile corresponding to f_p . In order to calculate the error-driven control effort

u_0 , a nonlinear weighted sum of the position error e_p and the velocity error e_v is used. The nonlinear PD controller is characterized by the nonlinear error weighting function $fal(\cdot)$ [see (5.10)] and the tuning parameters k_p , and k_d . Finally, the disturbance corrected control signal is denoted by u .

While addition and subtraction operations are relatively simple to implement on FPGA hardware, multiplication and division operations are significantly more resource intensive. Correspondingly, the tuning parameters,

$$\begin{aligned} b_{01} &= \beta_{01}, & b_{02} &= \beta_{02}T, & b_{03} &= \frac{\beta_{03}}{b}, \\ B &= bT^2, & \kappa_p &= \frac{k_p}{b}, & \kappa_v &= \frac{k_v}{b}, \end{aligned} \quad (6.2)$$

are defined to reduce the number of multiplication and division operations required to evaluate the ADRC algorithm. In addition, an alternative set of observer variables are introduced as,

$$\zeta_1 = z_1, \quad \zeta_2 = z_2T, \quad \zeta_3 = \frac{z_3}{b}, \quad (6.3)$$

where, ζ_1 is a simple substitute variable for z_1 , ζ_2 is the estimated plant velocity measured in displacement per unit sample period of the controller (i.e., T) instead of conventional units (e.g., displacement per second), and ζ_3 is the *scaled total disturbance*. Employing these new tuning parameters and variables in (6.2) and (6.3), the discrete ESO is reformulated in (6.4) to provide the estimation of the *total disturbance* in a scaled form so that it can be utilized to directly calculate the disturbance corrected control effort without the requirement of further scaling. In addition, this reformulation also reduces the number of multiplication operations required to evaluate the ESO by 33%. It should be noted that the parameters in (6.2) can be calculated directly from

the tuning parameters of the classic ADRC algorithm in (6.1).

$$\begin{aligned}
e(k) &= \zeta_1(k) - y(k), \quad \phi(k) = B(\zeta_3(k) + u(k)) \\
\hat{\zeta}_1(k+1) &= \zeta_1(k) + \zeta_2(k) - b_{01}e(k) \\
\zeta_2(k+1) &= \zeta_2(k) + \phi(k) - b_{02}e(k) \\
\zeta_3(k+1) &= \zeta_3(k) - b_{03}e(k)
\end{aligned} \tag{6.4}$$

The redefined ESO in (6.4) estimates the plant velocity in displacement per unit sample period. It is particularly convenient to choose an appropriate nonlinear function g_v that generates the transient velocity profile in the same unit. Since determining the transient position profile corresponding to g_v becomes a simple task of accumulating g_v once in each controller sample period as shown in (6.5), evaluating an additional nonlinear function to determine the transient position profile is no longer required.

$$\nu_p(k) = g_v(k), \quad y_p(k) = \sum_{k=0} \nu_p(k) + y(0) \tag{6.5}$$

Finally, the error-driven control law and the disturbance rejection law for the resource optimized implementation is provided in (6.6).

$$\begin{aligned}
e_p(k) &= y_p(k) - \zeta_1(k), \quad e_v(k) = \nu_p(k) - \zeta_2(k) \\
u_0(k) &= \mathit{falk}(\kappa_p, e_p(k), \alpha_p, \delta_p) + \mathit{falk}(\kappa_v, e_v(k), \alpha_v, \delta_v) \\
u(k) &= u_0(k) - \zeta_3(k)
\end{aligned} \tag{6.6}$$

The resource optimized algorithm evaluates the nonlinear control law in (6.6) by adopting an alternative nonlinear weighting function $\mathit{falk}(\cdot)$. It is evident from the definition provided in (6.7) that the mathematical complexity of directly evaluating the error-based control effort by the classic formulation in (6.1) and by using the

Table 6.1: Mathematical complexity of the classic ADRC algorithm & the proposed resource optimized algorithm

Criterion	Proposed	Classic
Nonlinear function evaluation	3	4
Addition/subtraction	13	11
Multiplication/division	4	9

$falk(\cdot)$ function in (6.6) is identical. However, the resource cost of implementing $falk(\cdot)$ on an FPGA is practically equal to that of implementing the $fal(\cdot)$ function, when the piece-wise linear approximation method in [23] is used. Thus, the two multiplication operations in evaluating the error driven control law in (6.1) can be avoided by using the $falk(\cdot)$ function to calculate u_0 as shown in (6.6).

$$falk(k_x, e_x, \alpha_x, \delta_x) = \begin{cases} k_x \frac{e_x}{\delta_x^{1-\alpha_x}}, & |e_x| \leq \delta_x \\ k_x |e_x|^{\alpha_x} \text{sign}(e_x), & |e_x| > \delta_x \end{cases} \quad (6.7)$$

A comparison of the mathematical complexities between the classic algorithm and the resource optimized algorithm provided in Table 6.1 shows that the number of required multiplication/division operations to evaluate the ADRC algorithm can be drastically reduced when the modified formulation is adopted. For the sake of simplicity, the fact that the nonlinear functions in both formulations may involve a varied number of arithmetic operations was ignored in this comparison.

6.3 FPGA Implementation of the ADRC

Algorithm

ADRC potentially can provide the framework for formulating a universal motion controller for the exactly actuated parallel manipulators. Nonetheless, examples of

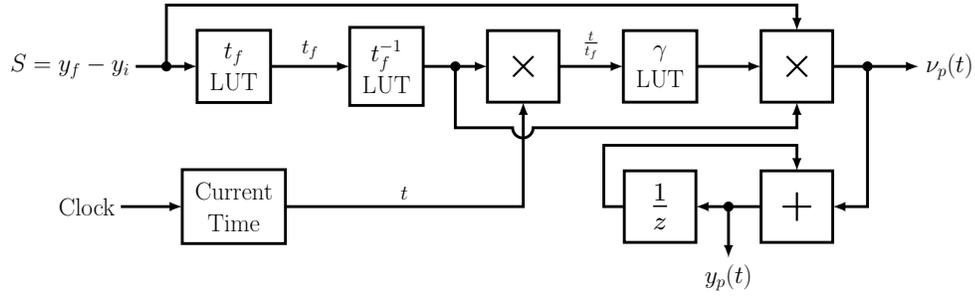


Figure 6.1: Cycloidal motion profile generation on an FPGA.

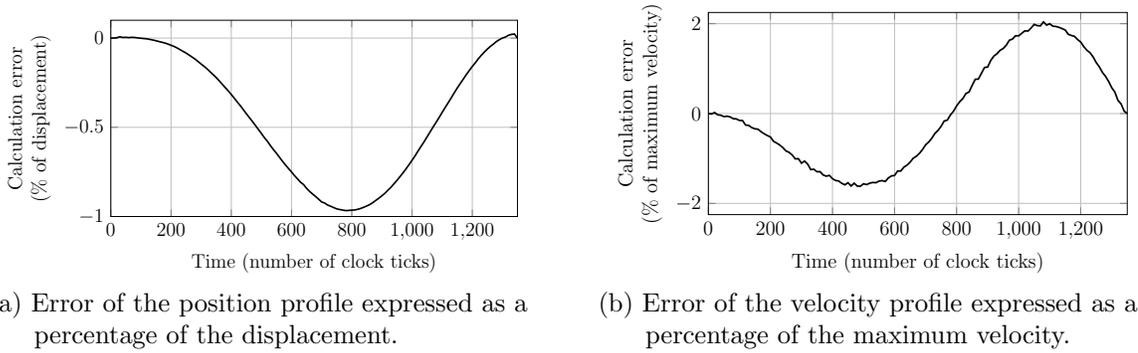


Figure 6.2: Accuracy of the motion profile as calculated by the profile generator module with respect to a double precision floating point calculation. The displacement corresponds to the entire range of a 16 bit position sensor.

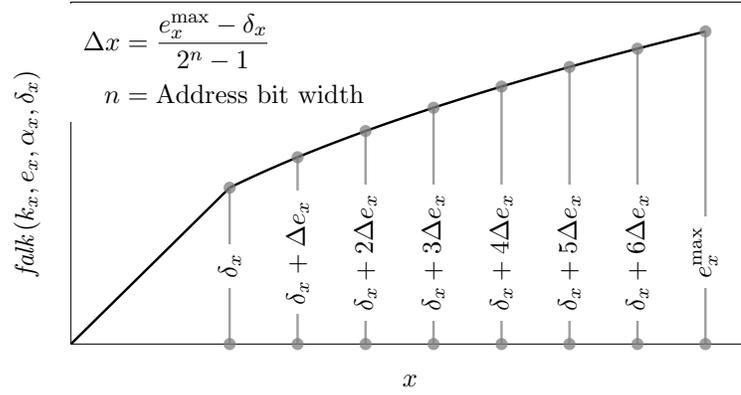
ADRC being implemented for such applications are scarce in the related literature; exceptions include [23, 93] that adopted ADRC for a 6 DOF and a 3 DOF PKM respectively. While the experimental validation in [93] involved an ADRC implemented on conventional control hardware, the simulation study conducted in [23] was based on a FPGA implementation of ADRC. Other applications that implemented ADRC systems on FPGA hardware include [90, 149, 151]. The resource cost of implementing the ADRC algorithm on an FPGA was optimized in [149] by adopting fixed point formats. It was reported that a high sampling rate of 22.25 MHz was achieved from a low logic density FPGA by using approximations and optimization techniques specific to fixed point format. Similar to [149], fixed point formats were used in [23]

to implement an ADRC controller on FPGA hardware for a 3-PSS/S manipulator. In contrast, the FPGA implementations in [90, 151] employed floating point format because floating point representation can provide superior accuracy and dynamic range. However, comparable performance is also obtainable from a fixed point format, if implemented carefully [152].

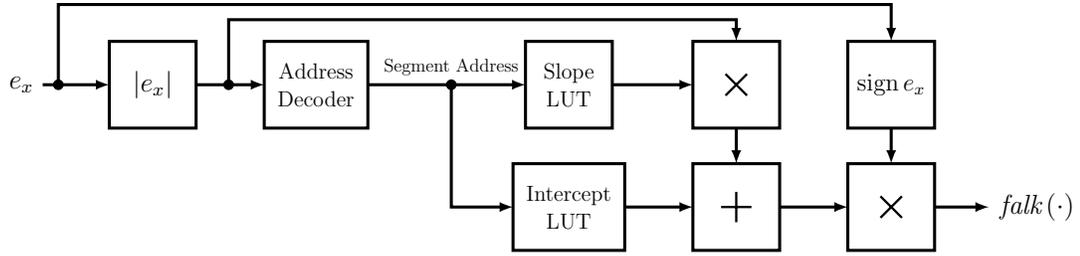
The FPGA implementation of ADRC algorithm in this thesis differs from the prior arts in many aspects, principally in the adoption of the resource optimized ADRC algorithm described in the previous section. In addition, it employs fixed point arithmetic to evaluate the necessary transcendental functions in a piece-wise linear manner. Since this design philosophy encourages efficient utilization of FPGA resources, it is possible to implement the design on a low-cost, low logic density FPGA. In this thesis, VHDL was chosen as the preferred design language to develop the digital circuits required for evaluating the ADRC algorithm. The design also utilizes the IEEE fixed point package extensively. In the following sections, FPGA implementation of the different components of the ADRC algorithm is discussed.

6.3.1 Transient Profile Generator

The transient time t_f in the cycloidal motion profile equations (5.7) and (5.8) determines how fast the desired position is reached. While a smaller t_f is preferred for high speed operation, the corresponding jerk may be too great for the limited actuation capacity. Determining the optimum t_f for a given maneuver (i.e., displacement in at least one actuator) constitutes an interesting problem that is beyond the scope of this thesis. However, this thesis employs a LUT (look up table) based approach in order to determine t_f for a given maneuver. To this end, the stroke of each actuator was arbitrarily divided into eight equal segments and different t_f values were assigned for



(a) Approximating $falk(k_x, e_x, \alpha_x, \delta_x)$ as a piece-wise linear function.



(b) Evaluation of $falk(\cdot)$ on an FPGA.

Figure 6.3: FPGA implementation of the $falk(k_x, e_x, \alpha_x, \delta_x)$ function.

each segment in order to construct the LUT. It should be noted that the t_f determined by the LUT for the largest of the three actuator displacements provides the global transient time for an arbitrary maneuver. Consequently, all three resulting motion profiles become synchronous in order to achieve the idea of *synchronous control* [74] without any extra effort. It is relevant to mention that this thesis adopts the empirical method of determining the transient time t_f described in Section 5.4.1 for simulation studies only. In the experimental cases, t_f was determined by the aforementioned LUT approach in order to optimize the resource cost.

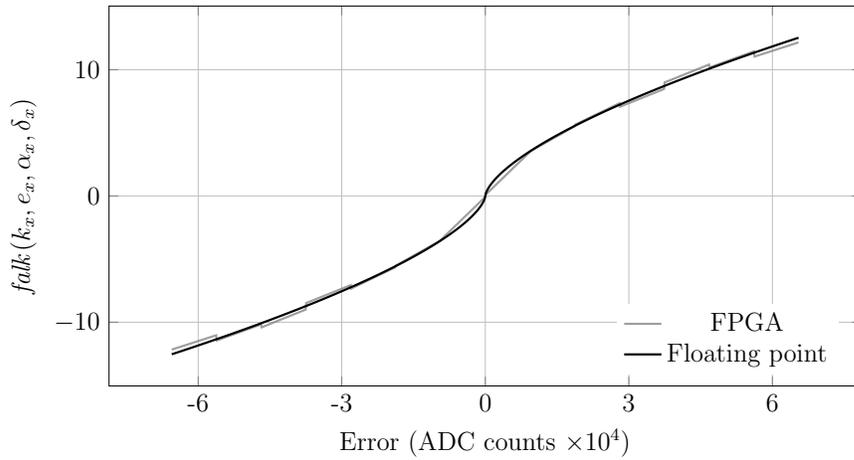
Since (5.7) can be obtained by integrating (5.8), only the velocity profile needs to be implemented for an efficient design. The nonlinear component $\gamma = (1 - \cos 2\pi \frac{t}{t_f})$ in (5.8) poses the greatest challenge, since FPGA architecture is not conducive to operations involving transcendental functions. The bounded nature of its argument

(i.e., $0 \leq \frac{t}{t_f} \leq 1$), however, permits implementation of the nonlinear component γ as a LUT. Since γ is symmetric about $\frac{t}{t_f} = \frac{1}{2}$, the LUT is constructed efficiently by evaluating γ for a set of values of $\frac{t}{t_f}$ uniformly distributed over the range $[0, \frac{1}{2}]$. Flexibility and scalability can be achieved by inferring this LUT analytically at compile time employing the IEEE real math package. A simplified block diagram of the corresponding implementation is provided in Fig. 6.1. In addition, Fig. 6.2 shows that the accuracy of the motion profile calculated in fixed point is adequate. Furthermore, the following remarks are deemed relevant:

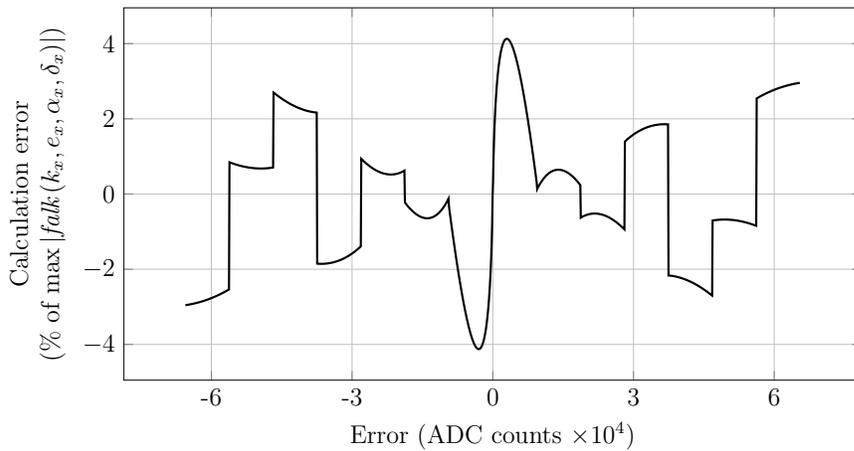
- a) Since the ADRC algorithm is evaluated once in each control clock cycle, it is natural to express the transient time t_f in terms of number of clock cycles. This allows the calculation of $\frac{t}{t_f}$ as a clock triggered accumulation of t_f^{-1} . It is also possible to store pre-calculated t_f^{-1} values in order to save the resources required for the inversion circuit.
- b) Wherever possible, dedicated silicon resources should be employed for implementing the design. This includes on-chip memory and multiplier blocks.

6.3.2 ESO & Nonlinear Control Law

Since the ESO in (6.4) involves only multiplication and addition/subtraction operations, implementing it on an FPGA is straightforward. The nonlinearity in the control law [i.e., (6.6)] originates from the $falk(k_x, e_x, \alpha_x, \delta_x)$ function in (6.7), which is nonlinear except when $|e_x| \leq \delta$. This observation inspires the idea of approximating it as a piece-wise linear function, where each linear piece is characterized by a slope and an intercept. Since $falk(\cdot)$ is an odd function, it is sufficient to characterize it for only positive valued arguments. To this end, a reasonable e_x^{\max} is chosen arbitrarily and the range $[0, e_x^{\max}]$ is subsequently divided into 2^n segments, as shown in Fig. 6.3(a).



(a) Calculation error of $falk(\cdot)$ with respect to a double precision evaluation.



(b) Relative error of $falk(\cdot)$ function evaluation.

Figure 6.4: Accuracy analysis of FPGA evaluation of the $falk(k_x, e_x, \alpha_x, \delta_x)$ function through piece-wise linear approximation. Realistic values of $k_x = 9.2829\text{E-}03$, $\alpha_x = 0.65$, and $\delta_x = 30.0$ were chosen for the calculations.

Here, n corresponds to the address bus width of the look up tables that store the linear characteristics of each segment. For simplicity, the slope and the intercept characterizing each segment are evaluated for their respective terminal points only. The block diagram representation of the corresponding digital circuit is presented in Fig. 6.3(b). In addition, Fig. 6.4 shows that the adopted piece-wise linearization approach provides acceptable accuracy.

6.3.3 Discrete ADRC Tuning

The tuning procedure described in Section 5.3 is applicable for a continuous time implementation of ADRC. However, the tuning method for a discrete implementation of ADRC is provided in [144, 145, 153]. It should be noted that the time scale of the velocity tracking error e_v in (6.6) must be consistent with the time scale of the controller bandwidth ω_c . Since e_v is expressed in ADC counts per clock period of the controller, the derivative gain for a discrete ADRC is also rescaled accordingly as,

$$k_d = \frac{2\zeta\omega_c}{T}. \quad (6.8)$$

In addition, the proportional gain k_p is determined by (5.11). According to [153], the observer gains $[\beta_{01}, \beta_{02}, \beta_{03}]$ for a discrete ESO can be calculated in terms of the observer bandwidth ω_o as follows,

$$\begin{aligned} \beta_{01} &= 1 - \beta^3, & \beta &= e^{-\omega_o T}, \\ \beta_{02} &= (1 - \beta)^2(1 + \beta) \frac{3}{2T}, \\ \beta_{03} &= (1 - \beta)^3 \frac{1}{T^2}. \end{aligned} \quad (6.9)$$

6.3.4 Resource Cost Optimization

When the permissible precision loss margin of a fixed point operation and the corresponding operational dynamic range are known *a priori*, it is possible to optimize the resource cost of implementing it on FPGA hardware by choosing appropriate fixed point formats (i.e., bit lengths) for the operands. The dynamic ranges of all variables in the ADRC algorithm are determined by the magnitudes of the controller tuning parameters, system input, and system feedback. Although the magnitudes of system

input and feedback are known by design, the tuning parameters must be determined experimentally. Once suitable values for all tuning parameters are determined, the resource cost of each arithmetic operation in the ADRC algorithm is optimized by assigning appropriate fixed point formats to all controller variables. However, it should be mentioned that large bit lengths should be allowed for the variables involving the modified ESO in (6.4). Because if these variables are represented by insufficient bit lengths, they may become susceptible to unacceptable precision loss and overflow due to the accumulative structure of the ESO.

6.4 Experimental Evaluation of the Resource Optimized ADRC Algorithm

A prototype of the VCA design proposed in Chapter 3 was used as the experimental platform for evaluating the performance of the modified algorithm. The controller for the experimental setup was prototyped on an EP3C40F484C6 FPGA chip manufactured by Altera Corporation. By design, the maximum input voltage for the VCA prototype was set at 24.0 volts.

6.4.1 Resource Cost of Implementation

In order to compare the resource costs of the modified and the classic ADRC algorithm, three controllers were prototyped. One of them adopted the classic algorithm, while the remaining two implemented the resource optimized algorithm. The latter two are henceforth referred to as the Proposed-A and the Proposed-B controllers. Although only fixed point formats were employed in the Proposed-A and the classic controllers, the modified ESO of the Proposed-B controller was prototyped using single precision

32-bit floating point formats. It should be noted that the other components of the Proposed-B controller are identical to those of the Proposed-A controller. In addition, the ESO variables of the Proposed-B controller was converted to fixed point formats to resolve the compatibility issue. With the exception of the underlying implementation of the ADRC algorithm, a fair comparison was ensured by making the three controllers identical in every aspect including the tuning parameters and the other auxiliary systems involving data acquisition. In addition, each fixed point arithmetic operation in the implementation of the classic formulation was optimized by determining the dynamic ranges of the operands. The synthesizing software Quartus II (64 bit web edition, build 13.0.1) was used to configure these controllers on the experimental control hardware. The corresponding resource costs are provided in Table 6.2. Since the memory resources of the FPGA chip were primarily used to implement a data acquisition system, the resource cost of the three implementations are identical in this category. The Proposed-B controller utilized the most number of logic elements and 3% more embedded multipliers than the Proposed-A controller, which is not unexpected because of the added resource cost arising from floating point implementation of the modified ESO. The presented data clearly shows that the modified formulation (i.e., the Proposed-A controller) can be implemented at a fraction of the resource cost of the classic algorithm. Specifically, the resource optimized algorithm of the Proposed-A controller utilizes 34% less logic elements and 69% less embedded multipliers than the classic algorithm.

6.4.2 Transient Response Performance

An experiment consisting of a hundred controlled displacements of the proposed VCA was conducted to evaluate the performance of the controller. Each of these

Table 6.2: Resource cost of FPGA implementation of the classic algorithm & the proposed resource optimized algorithm

Criterion	Proposed-A	Proposed-B	Classic
Logic elements	4,029 (10%)	15,648 (40%)	6,131 (15%)
Memory bits	315,392 (27%)	315,392 (27%)	315,392 (27%)
Multipliers	36 (14%)	44 (17%)	117 (46%)

*In the parentheses, utilized resources in each category is expressed as a percentage of total available resources on an EP3C40F484C6 chip.

Table 6.3: ADRC controller tuning parameters for the nominal plant

Parameter	Value	Unit
T	40.96	μs
k_p	390,625	volts/ADC counts
k_d	3.8147E+07	volts \times T/ADC counts
b	7.75E+07	-
β_{01}	0.2227	-
β_{02}	455.9597	-
β_{03}	3.1139E+05	-
α_p	0.68	-
δ_p	10.0	ADC counts
α_v	1.25	-
δ_p	5.0	ADC counts/T

displacements was defined by a randomly chosen initial position and final position. These positions, expressed in ADC counts, were sampled from the experimental space [5000, 60000] in order to minimize the possibility of hitting the two hard stops at either end of the VCA stroke during experimental trials. The purpose of the hard stops is to maintain the laser diode focused on the active area of the PSD at all times. Furthermore, they prevent the coil from advancing beyond the uniform sections of the magnetic field. In addition, each displacement was constrained to be at least 1000 ADC counts in magnitude to avoid any trivial sample point. The test displacements remained identical for each trial of this experiment. In addition, a cycloidal motion profile [23, 142] generator was used to provide the controller with a transient position and velocity profile to follow during the test displacements.

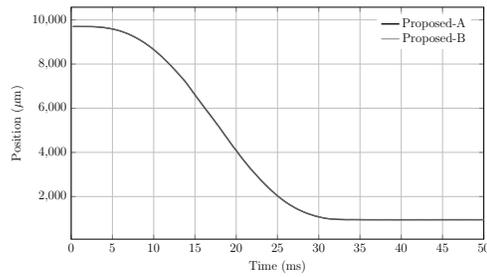
Table 6.4: Control performance obtained by the proposed resource optimized algorithm and the classic algorithm

Plant	Implementation	SSE (μm)		ST (ms)	
		Mean	σ	Mean	σ
Nominal	Proposed-A	2.5706	2.5993	21.7518	4.5709
	Proposed-B	2.7635	2.7055	21.8685	4.5718
	Classic	3.3200	3.3436	21.8206	4.6143
Perturbed	Proposed-A	3.2326	3.1361	24.0484	5.7246
	Proposed-B	2.7697	2.7600	23.6794	5.9473
	Classic	3.3852	2.7944	23.7072	5.7640

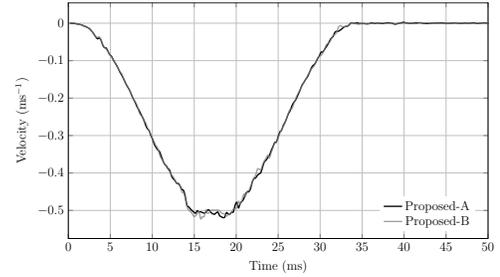
In a first trial, the Proposed-A controller carried out the test displacements on the nominal plant. The experimental VCA prototype with a moving mass of 33.30 grams at no load condition constituted the nominal plant, for which the tuning parameters in Table 6.3 were specifically determined. In order to assess the robustness of the controller against plant perturbation, the nominal plant was drastically changed by adding a load of 33.50 grams to the moving mass. Without retuning the controller parameters, a second trial was conducted using the perturbed plant. The Proposed-B controller and the classic controller were then employed to carry out the test displacements on both the nominal plant and the perturbed plant. Thus, a total of six trials were conducted to assess the transient response performance provided by the different controllers. As a reference, the recorded controller variables from the largest trial displacement are provided in Fig. 6.5.

6.4.3 Frequency Response Performance

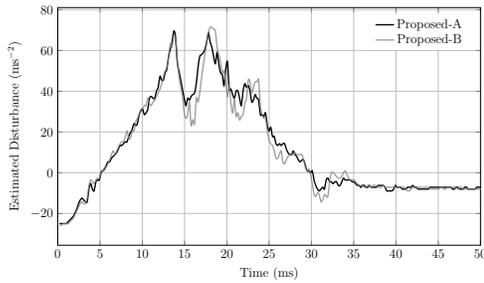
In order to determine the frequency response performance of the closed-loop system, the cycloidal motion profile generator in the controller was replaced with a sinusoidal position profile and a corresponding velocity profile generator. The sinusoidal position profile is characterized by an amplitude of 5000 ADC counts oscillating about the



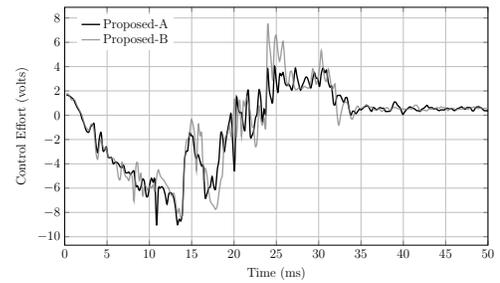
(a) Actuator position provided by the ESO.



(b) Estimated velocity provided by the ESO.

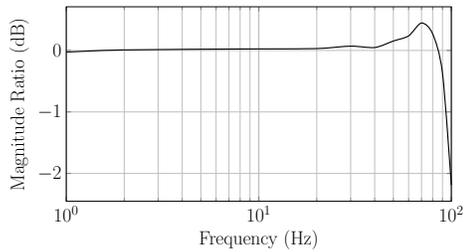


(c) Estimated total plant disturbances.

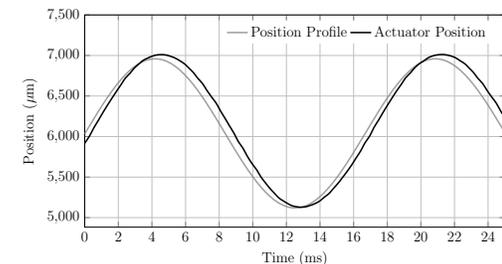
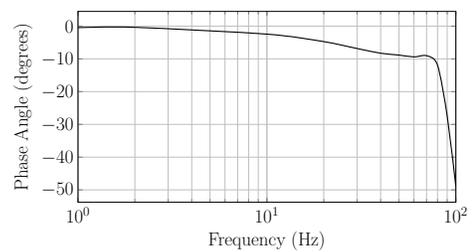


(d) Disturbance corrected control effort.

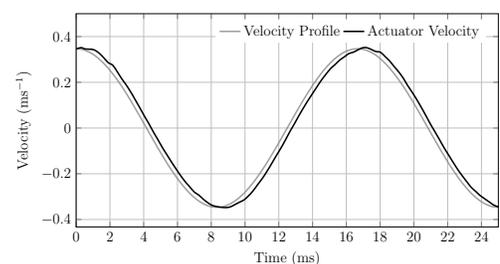
Figure 6.5: Recorded controller variables provided by the resource optimized algorithm in a representative trial displacement.



(a) Experimentally determined bode plot of the system for a sinusoidal position input with an amplitude of 5000 ADC counts about the mid stroke of the VCA.



(b) Position tracking performance of the controller at a frequency of 60 Hz.



(c) Velocity tracking performance of the controller at a frequency of 60 Hz.

Figure 6.6: Frequency response of the experimental plant.

mid stroke of the actuator. The tracking performance of the controller were recorded in terms of magnitude ratio and phase shift at different input frequencies. From the corresponding bode plot provided in Fig. 6.6(a), it can be seen that the controller provided excellent tracking performance up to a frequency of 60 Hz. However, the control performance rapidly deteriorated as the input frequency was increased further. The tracking performance in response to a 60 Hz input signal is shown in Fig. 6.6(b) and Fig. 6.6(c). It is relevant to mention that the Proposed-A controller was used in this exercise.

6.4.4 Experimental Results

Besides the controller, a data acquisition system was implemented on the experimental FPGA hardware to log and transmit interesting controller variables to a desktop computer during the experiment. Starting just after the initiation of a trial displacement, all interesting controller variables were sampled 2047 times (size of the data logging FIFO) at a frequency of 24.4 kHz, which corresponds to a monitoring window of 83.85 ms. The data collected in all four trials was subsequently analyzed to quantify the control performance in terms of settling time and steady state error. The definitions provided in Section 5.4.2 were adopted to estimate the settling time and the steady state error of a transient response. Since the data recorded here is of discrete nature, a sampling window of 8.1920 ms (i.e., 200 consecutive samples) was used to determine the steady state according to the method of moving variance in [147]. The performance metrics provided in Table 6.4 were calculated from the experimental data.

The most energetically active component of the plant (i.e., the moving mass) was increased by more than 100% in order to evaluate the robustness of the controller. Despite this drastic change, the experimental results presented in Table 6.4 show

that the steady state errors from all experimental trials are statistically similar in terms of sample means and their respective distributions, which is indicative of the robust performance the ADRC technology is capable of delivering. A series of t-tests performed on the performance metrics presented in Table 6.4 confirmed that the settling times and the steady state errors within each experiment group (i.e., nominal plant or perturbed plant) have equal means and equal variances at the 5% significance level. However, the settling times involving the perturbed plant exhibit an approximate increase of 9%, which is not unexpected considering the magnitude of plant perturbation. Besides experimentally verifying the robustness in control performance, the statistics in Table 6.4 provide no evidence that the classic implementation of the ADRC algorithm outperforms the proposed resource optimized formulation. Hence, it can be concluded that the proposed formulation delivers typical ADRC performance at a fraction of the resource cost of the classic algorithm.

6.5 Resource Optimized ADRC for a Parallel Orientation Manipulator

Prior to hardware deployment, the performance of the resource optimized ADRC formulation in terms of regulating the motion of a parallel orientation manipulator was evaluated in a co-simulation approach. To this end, the dynamical model of the parallel configuration of the 3-PSS/S manipulator was chosen arbitrarily to conduct the simulation study. Although bond graphs were employed to develop the dynamic models of the candidate manipulators in Chapter 5, a bond graph simulator that interfaces with an HDL simulator is not readily available. Alternatively, the multibody model was reconstructed employing the SimMechanics toolbox of Simulink. In contrast to

the bond graph model, the SimMechanics model of the test POM can be characterized as a black box model. Since one of the objectives of this chapter is to describe how an advanced controller can be developed for an FPGA hardware, a detailed account on system modeling is considered to be outside of its scope and is not discussed further.

Three identical linear voice coil actuators (VCA) drive the orientation manipulator. Accordingly, the causal structure of the multibody model was designed to accept actuation forces at the three prismatic joints. Since the multibody model already accounts for the moving mass of each actuator, only the electrical and the frictional dynamics of the actuators are required to be incorporated with the multibody manipulator model. Although the frictional characteristics of the spherical joints were ignored for the sake of simplicity, a friction model involving Coulomb and viscous friction for each of the three actuated prismatic joints was implemented [see Fig. 6.7(a)], which is provided by [154],

$$F_f = \text{sign}(V_a)F_c + \mu_k V_a. \quad (6.10)$$

In (6.10), F_f , V_a , F_c , and μ_k denote total friction force, actuator velocity, Coulomb friction force, and viscous friction coefficient respectively. The electrical dynamics of the actuators can be modeled as a RL (resistance-inductance) circuit. The corresponding transfer function (i.e., coil impedance) provides the coil current in terms of the coil voltage. As shown in Fig. 6.7(b), the actuator force and the actuator velocity respectively constitute the casual output and the causal input of the actuator sub-model.

An optical position sensor installed on each actuator provides the controller with the necessary feedback signals. The sensor acts as a first order filter with the transfer function $\frac{K_S}{1+\tau s}$, where K_S and τ denote the gain and the time constant of the sensor respectively. Because of the dynamics of the sensor, the feedback signal lags the actual

Table 6.5: Tuning parameters for the ADRC controller

Parameter	Value	Unit
Controller bandwidth, ω_c	870	rad/s
Observer bandwidth, ω_o	6500	rad/s
Damping factor, ζ	1.0	–
Controller sample period, T	40.96	μs
Nonlinear gain parameter, α_p	0.75	–
Nonlinear gain parameter, δ_p	100.0	ADC counts
Nonlinear gain parameter, α_v	1.75	–
Nonlinear gain parameter, δ_v	5.0	ADC counts/T
Plant parameter, b_i	1.50E+08	–
Transient motion profile parameter, t_{min}	125.00	ms
Transient motion profile parameter, c	0.15	–

Table 6.6: Performance of the controller in the co-simulation study

Statistic	Execution time (ms)	Steady state errors (μm)		
		1	2	3
Mean	66.8168	2.2623	2.5088	2.3886
Max	148.1523	5.6908	9.5931	6.3097
Min	25.6819	0.0250	0.0400	0.0014
σ	21.7730	1.3340	1.5564	1.4391

controller is difficult to achieve. Except for the linear bearing friction parameters, the manipulator and the actuator models were identical to those described in Chapter 5. The friction model employed in the simulation study is characterized by a coulomb friction force of 0.5 N and a viscous friction coefficient of 0.16 Ns/m. In addition, the position sensor time constant was determined to be 300 μs . Since the friction model adopted in this chapter is significantly different from that in Chapter 5, the controller parameters were accordingly tuned. These parameters are provided in Table 6.5.

In the simulation experiment, the designed ADRC controller was evaluated by an HDL simulator and the manipulator model in Simulink provided the ADRC controller with the simulated position data of all three actuators to calculate appropriate control actions. These control actions were then fed back to the Simulink model (i.e., co-simulation approach, refer to Fig. 6.8) to complete the control loop. The test maneuvers

of this co-simulation experiment were identical to those in Chapter 5. Statistics of the simulated performance metrics are provided in Table 6.6. Please note that a uniform random noise with a range of ± 4 ADC counts was introduced in each input channel of the controller to simulate the occurrence of noise in a practical sensor.

6.6 Discussion

Since the validation of a complex digital circuit is not a trivial exercise, only a subset of all possible test cases was considered [155]. Because the number of possible test cases exponentially increases with the complexity of the circuit, an exhaustive validation is often impractical. Nonetheless, the co-simulation study validates the controller in terms of functionality and structural accuracy of the digital design under simulated operating conditions. As a result, the risk of damaging an experimental setup with a controller that would have been otherwise validated for only random inputs is greatly minimized. It also offers better debugging capacity by providing comprehensive access to quantities internal to the controller. Thus, a co-simulation study appreciably streamlines the prototyping process of a complex controller on digital hardware by reducing development time and resources.

Admittedly the data presented in Tables 6.6 and 6.4 are not absolute measures of control performance, since these performance metrics depend on the steady state detection algorithm. However, it was confirmed through simulation that active disturbance rejection control can perform well for a nonlinear, time-varying, highly coupled dynamic system. In addition, the data presented in Table 6.4 suggest that the controller delivers similar performance for both the nominal and the perturbed plant despite an increase in the moving mass by more than 100% in the perturbed plant. This observation is indicative of the robustness in control performance, which

is a prerequisite for handling the time-varying inertia of the orientation manipulator.

The control performance of the resource optimized ADRC algorithm provided in Table 6.6 is very similar to the simulation results obtained in Chapter 5. However, the minor discrepancy can be explained by the elaborate friction model and the dynamical model of the sensor that were adopted in the co-simulation study. The data presented in Table 6.6 validates the designs of the digital circuits that implement the resource optimized ADRC algorithm. In addition, it can also be claimed that the resource optimized formulation of the ADRC algorithm does not affect the control performance.

6.7 Conclusion

This chapter details the digital design of an ADRC controller that involves transcendental functions employing the efficient fixed point format. Besides demonstrating how the co-simulation approach can simplify the development flow of control applications that regulate physical systems, this chapter has experimentally established the robustness of the controller in terms of managing varying inertia. In addition, the classic formulation of the ADRC technology was algebraically manipulated to obtain a resource optimized algorithm that can be efficiently implemented on FPGA hardware. Despite the reduced resource cost, this chapter has experimentally demonstrated that the control performance achieved by the proposed algorithm is identical to that provided by the classic formulation of ADRC technology. In applications requiring multiple control loops (e.g., motion control of a multi-DOF manipulator), the proposed algorithm provides the opportunity to implement a plurality of ADRC loops on a single low logic density FPGA chip to obtain a cost effective solution.

Chapter 7

Prototype Implementation & Performance Characterization

The previous chapters of this thesis have synthesized designs of actuation systems, POM dimensions, and motion control systems for coupled MIMO plants, all of which individually constitute an important component for a parallel orientation manipulation system. Some of these designs were already implemented in the physical domain, and the corresponding prototypes were extensively tested to confirm their ability to deliver the desired function. Nonetheless, all of these engineering development exercises were geared towards the ultimate goal of constructing POM prototypes for optomechatronic applications. To the end of fulfilling that goal, this chapter employed the previously synthesized designs, and proposed an alternative construction of spherical joints that was specifically designed to possess anti-backlash characteristics in a small form factor. All these system components were employed to implement the prototypes of the two candidate POM designs. The underlying kinematic structures of these prototypes were characterized by the parallel and the slanted configurations of the 3-PSS/S architecture. The dynamic capacities of the prototypes were experimentally determined to confirm that the design goal of high speed manipulation was achieved. In addition, this chapter proposed an empirical approach towards the kinematic calibration of parallel orientation manipulators. Numerical and physical experiments showed that the performances of these empirical models in terms of prediction accuracy were comparable to those of the conventional geometric kinematic models.

7.1 Introduction

Prior to investing significant development resources to construct a physical system, it is customary to evaluate the candidate designs in the virtual domain first. Since this approach provides a means to identify infeasible designs early in the development cycle, available resources can be assigned to the most promising design solutions. Despite the efficiency obtained in terms of resource utilization, virtual prototypes can never be *exact* representations of the physical world. Because assumptions and simplifications must be made in order to manage the computational complexity of the underlying mathematical models, there is a limit to how closely these models can simulate real world behavior. Nonetheless, virtual prototyping can bring immense value to engineering development, especially when it is not cost-sensible to construct physical prototypes for evaluation purposes. The past chapters of this thesis embraced the philosophy of virtual prototyping, and conducted numerical experiments to evaluate the kinematic and the dynamic performance of a set of candidate POM designs. However, these numerical experiments constituted the early stages of development. The current chapter further advances the development cycle by constructing physical prototypes of the synthesized designs.

The candidate POM designs in this thesis utilize two types of kinematic joints; namely, prismatic joints and spherical joints. Since the prismatic joints are an integral part of the linear actuators developed in Chapter 3, they are not discussed further in the current chapter. However, an improved design for spherical joints is proposed in order to address the limitations of the commercially available units. The proposed design was evaluated extensively, and was shown to possess anti-backlash characteristics.

Since a synthesized PKM design can only be implemented in the physical domain within a specified manufacturing tolerance, the kinematic models based on its nominal

geometry provide limited accuracy. In practice, accuracy in this regard is generally improved by identifying the actual geometry; i.e., kinematic calibration. In a conventional calibration method, constraint equations are derived from the kinematic structure of the PKM. Experimental observations (i.e., actuated joint coordinates and the corresponding poses of the moving platform) are then employed to find a solution that fits the constraint equations best. This chapter proposes that the requirement of formulating the constraint equations can be dissolved if the relationships between the joint space coordinates and the workspace coordinates are treated as response surfaces defined by empirical polynomial models. Correspondingly, the response surfaces estimated by standard statistical algorithms can serve as an alternative to the kinematic models. Regardless of whether a kinematic structure possesses a closed-form solution to the direct or the inverse kinematic models, the response surfaces can be defined either to estimate the workspace coordinates as functions of the joint space coordinates or vice versa. Furthermore, calibration of the joint sensors to absolute units is no longer necessary since the response surfaces can be parameterized by uncalibrated sensor readings. Encouraged by these potential advantages, this chapter experimentally showed that the 3-PSS/S manipulator can be conveniently calibrated employing response surface models.

The aforementioned ideas are elaborated in the remainder of this chapter. Section 7.2 documents the design and the experimental evaluation of the proposed spherical joint. The prototypes of the synthesized POM designs are developed in Section 7.3. In addition, a post optimality study was conducted to examine the robustness of the kinematic performances provided by the candidate POM designs. The corresponding dynamic performances are experimentally evaluated in Section 7.4. The proposed kinematic calibration method for the 3-PSS/S manipulator is described and experimentally evaluated in Section 7.5. Since the torsional degree of freedom

of a POM is not necessary in applications involving axis-symmetric payloads, a torsion-restricted POM is synthesized in Section 7.6. Finally, Section 7.7 provides the concluding remarks.

7.2 Design of an Anti-Backlash Spherical Joint

A spherical joint is typically composed of two major components, namely a ball and a socket. In a conventional spherical joint, the ball is embedded in a spherical or conical cavity of the socket to allow only relative angular motion with three degrees of freedom. For the sake of low friction movement and easy assembly, a small gap is necessary between the mating surfaces of these two parts. However, this gap introduces backlash, which ultimately results in inaccurate spherical motion. In order to eliminate the backlash, alternative designs have been proposed in the literature; examples include spherical rolling joints [156] and dynamic preload adjustment ball joint [157]. Although a spherical rolling joint can provide relative motion that is virtually free of backlash, its range is generally limited. The preload adjustment ball joint requires a dedicated preload actuator and a corresponding controller. Therefore, its construction is not suitable for small-scale applications. Furthermore, commercially available spherical joints (e.g., rod-end bearings) can offer neither backlash-free operation nor a large range of motion. This thesis addresses these issues by proposing an improved design that offers a large range of motion with practically zero-backlash movement. An implementation of this design is illustrated in Fig. 7.1. It should be mentioned that a similar design can be found in a commercial 3D printer [158].

Similar to the conventional construction of spherical joints, the proposed design includes a ball and a socket [see Fig. 7.1(c)]. The cylindrical socket features a continuous axial cavity between its two planar surfaces [see Fig. 7.1(a)]. This cavity

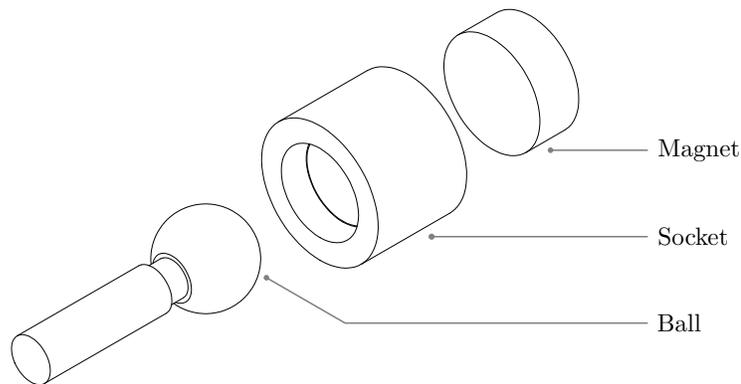
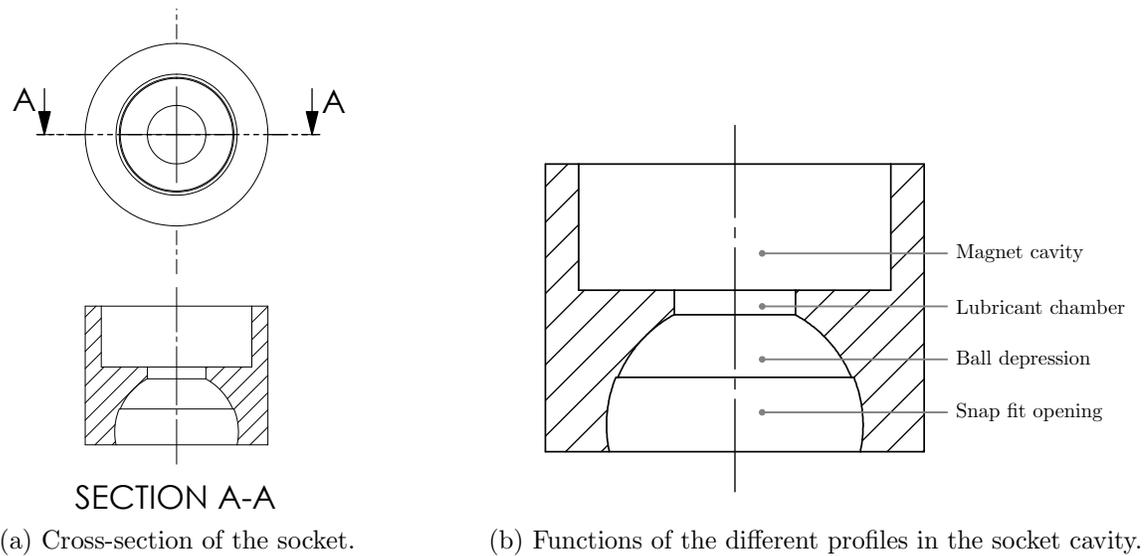


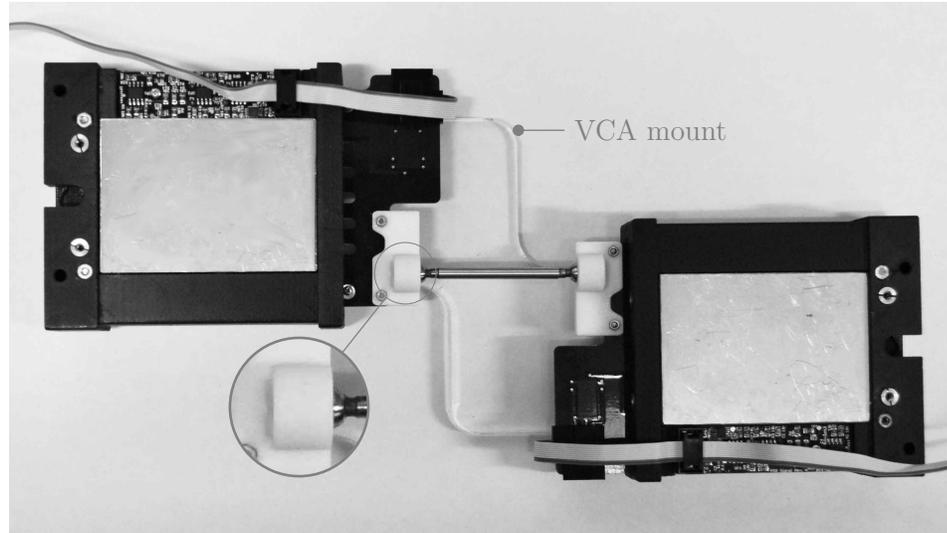
Figure 7.1: Design of the magnetically loaded, zero backlash spherical joint.

has four distinct geometric profiles, each serving a certain function [see Fig. 7.1(b)]. At one end, there are two concentric spherical profiles. The outer spherical cavity features a slightly larger diameter than its adjacent cavity. Within a very tight geometric tolerance, the diameters of the ball and the inner spherical cavity are identical. It is also relevant to mention that the diameter of the circular opening corresponding to the outer spherical cavity is slightly smaller than the diameter of the ball, which creates a snap fit between the socket and the ball. The snap fit design increases the capacity of the joint in terms of the maximum load it can support; i.e., the force

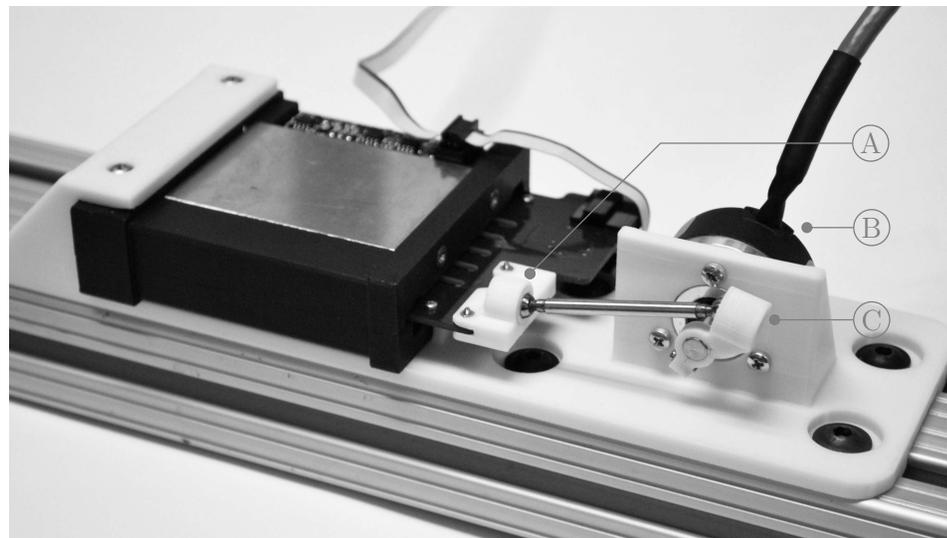
needed to completely separate the ball and the socket assembly. The corresponding breakaway force was experimentally confirmed to be sufficient for the dynamic loads that may occur in a small-scale orientation manipulator. Of the two cylindrical cavities at the other end of the socket, the outer cavity accommodates an axially magnetized cylindrical magnet and the inner cavity serves as a lubricant chamber. The attractive force between the magnet and the ferromagnetic ball acts as an anti-backlash preloading. Thus, backlash-free motion is obtained as long as the load on the joint does not exceed the magnetic preloading. Although a stronger magnet may seem desirable to extend the joint's capacity, the friction from a large magnetic preload results in poor dynamic performance. Therefore, the strength of the magnet should be chosen as a compromise between the achievable range of backlash-free motion and the maximum allowable friction in the joint.

7.2.1 Prototype Development

Delrin is a thermoplastic polymer that offers excellent abrasion resistance, low coefficient of friction, and high heat resistance. Because of these favorable mechanical properties, the socket of the proposed spherical joint was manufactured from this material. A precision CNC machining process was employed to ensure the geometrical accuracy of the spherical cavities in the prototyped socket. The inner spherical profile of the socket cavity was dimensioned to mate with a commercially available *tooling ball* [see Fig. 7.1(c)] made of magnetic stainless steel. Since a tooling ball is typically used to establish a reference point by measuring hole centerlines in precision machining tasks, it is manufactured to a tight tolerance. Specifically, the tooling ball used in this prototype features a diameter of $\frac{1}{4}$ inch with a tolerance ± 0.2 mil. It is obvious that the construction of the proposed design offers the full range of the torsional motion.



(a) Experimental setup for quantifying anti-backlash characteristics under static conditions. Inset: proposed spherical joint prototype.



(b) Experimental setup for determining motion transmission characteristics under dynamic conditions.
A: slider, B: encoder, C: crank.

Figure 7.2: Experimental evaluation of the proposed spherical joint.

In addition, it can provide a maximum tilt angle of 54° while covering the entire range of the azimuth angle.

7.2.2 Experimental Evaluation

In order to quantify the performance of the proposed design, two separate experiments were performed. The purpose of the first experiment was to demonstrate the anti-backlash properties under static loading. In contrast, the second experiment was conducted to quantify the dynamic motion transmission characteristics.

7.2.2.1 Static Performance Characterization

The experimental setup corresponding to the first experiment is shown in Fig. 7.2(a), which was comprised of two voice coil actuators (VCA) from Chapter 3. These two VCAs were mounted on a rigid platform so that their axes were aligned. The integrated position sensor with sub-micron resolution was proved to be immensely useful for identifying the backlash characteristics of the proposed design. In this experiment, one actuator was arbitrarily chosen to be the driver VCA. The other VCA (i.e., follower) remained constrained to the driver by a shaft. At each end of the shaft, a spherical joint provided the connecting interface between the actuator piston and the shaft body. The ball and the socket of the the spherical joint adjacent to the driver VCA were rigidly bonded together by using epoxy. Consequently, the other joint alone compensated for any misalignment between the axes of the actuators. Furthermore, restricting the motion of one spherical joint ensured that any backlash in the system was contributed by a single joint. Although the coil of the follower VCA did not receive any power, its position sensor remained active to provide its location. In addition, the position of the driver VCA was controlled in order to characterize the

performance of the proposed spherical joint.

In each of the experimental trials, the driver VCA traveled a distance of 1.84 mm (≈ 10000 ADC counts) from either direction (i.e., inward or outward motion) to reach and maintain a predefined target position. The mid-stroke of the driver VCA was arbitrarily chosen as the target position. Sufficient time was allowed after each experimental movement so that the driver VCA could completely come to a rest. After the driver VCA was settled, the positions of the two actuators were sampled for 1000 times. Since the position controller of the driver VCA failed to reach the target position with a minimum steady state error in some trials, they were correspondingly discarded from the post-experiment analysis. In specific terms, the acceptance criterion in this regard dictated that the standard deviation of the 1000 position samples must be less than $0.18 \mu\text{m}$ (≈ 1.0 ADC counts), and the corresponding mean must be within $0.37 \mu\text{m}$ (≈ 2.0 ADC counts) of the target position. A total of 100 such successful trials were recorded. The first fifty trials were conducted by performing inward motion, and outward motion was employed to execute the remaining fifty trials. These trials constituted the treatment group of the experiment. In addition, a control group was established by recording 100 more successful trials with no preload magnet in the spherical joint being examined.

The recorded experimental data is graphically presented in Fig. 7.3. The zero positions of the two actuators in each experiment group were provided by respective group means. The presented data exhibits four distinct clusters. It can be observed that the data points in each individual cluster were obtained by approaching the target position from the same direction (i.e., outward or inward motion). Furthermore, a comparison between the distributions of the data points from the two experiment groups indicates that the magnetic preloading in the proposed design indeed minimizes backlash. In specific terms, the mean positions of the follower VCA from the treatment

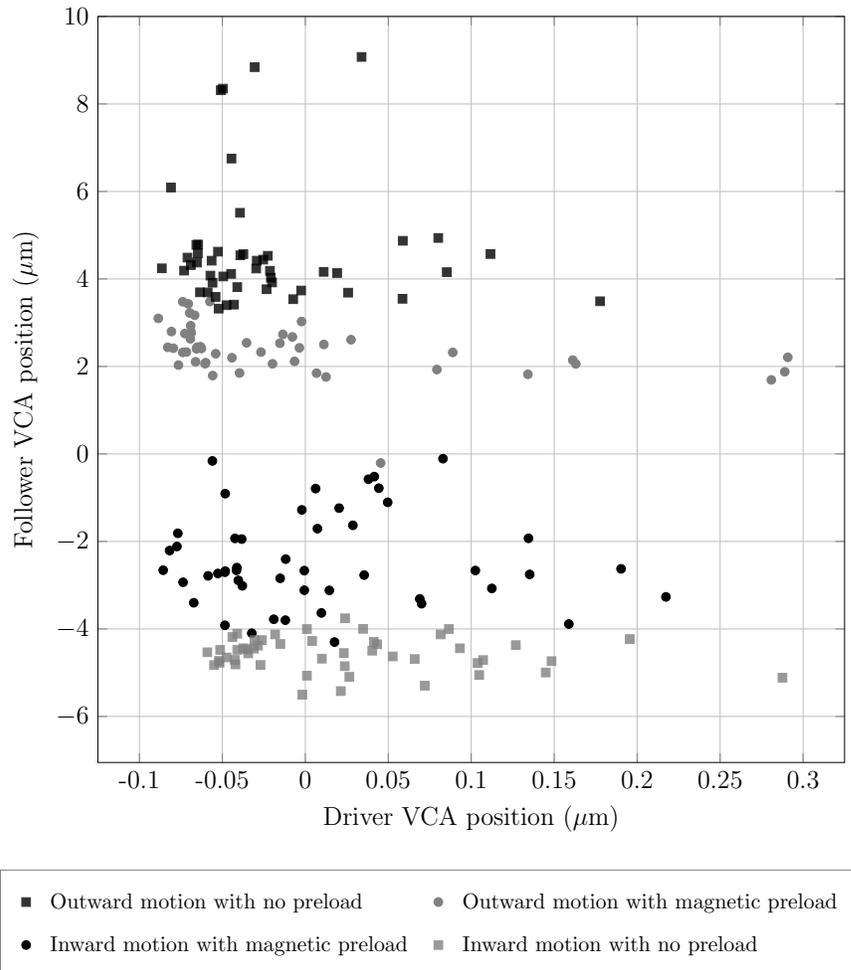


Figure 7.3: Experimental observations in the static repeatability test.

group and the control group are respectively $\pm 2.3750 \mu\text{m}$ and $\pm 4.5759 \mu\text{m}$. Since this experiment did not account for other sources of error such as measurement error or the compliance of the experimental setup, it is unclear whether the small magnitude of the backlash observed in the treatment group was entirely caused by the spherical joint itself. Nonetheless, the proposed design was shown to provide nearly backlash-free motion under static conditions.

7.2.2.2 Dynamic Motion Transmission Characteristics

The proposed spherical joint was subjected to a negligible static load in the previous experiment. In order to determine the performance of the proposed design under dynamic loading, a second experiment was conducted employing the experimental setup shown in Fig. 7.2(b). This experimental setup can be characterized as a classic slider-crank mechanism, where the two revolute joints were replaced by two proposed spherical joints. In addition, the proposed VCA from Chapter 3 served as the slider and the motion generator.

An encoder was used to determine the crank angle and the integrated position sensor provided the slider position. For the sake of this experiment, it was necessary to obtain a mathematical model that provides the slider position as a function of the crank angle. To this end, the corresponding analytical model defined by the geometric parameters (e.g., crank radius, connecting rod length, offset, etc.) of this slider-crank mechanism can be employed. However, the cited geometric parameters must be estimated by metrological means with acceptable accuracy. Alternatively, a simpler solution can be formulated by employing an empirical model derived from a finite set of experimental observations. Specifically, a ninth order polynomial model estimated by least squares curve fitting was commissioned to provide an accurate representation of motion of the experimental setup.

It should be mentioned that the position sensor of the VCA is a first order filter with a time constant of $100 \mu\text{s}$. Due to its dynamics, the signal provided by the sensor lags the true position by a small amount, especially when the VCA is in motion. In order to demonstrate the effects of sensor dynamics, the slider was powered by a step input with a magnitude of 24.0 Volts. The corresponding slider positions and the encoder positions were recorded. From these recorded data, two time series of crank

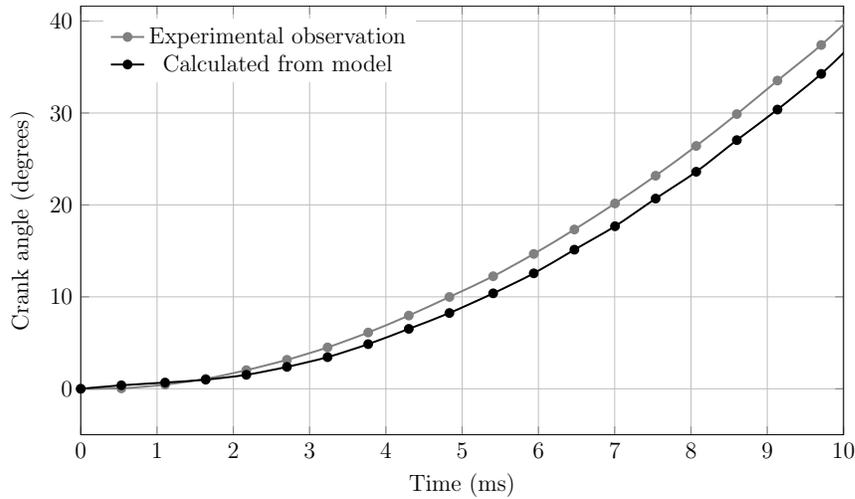


Figure 7.4: Experimental demonstration of position sensor dynamics.

angles are presented in Fig. 7.4. The *experimental observation* in Fig. 7.4 refers to the crank angles obtained from the encoder. In addition, the second time series was calculated from the recorded slider positions employing the aforementioned empirical model. It is relevant to mention that the encoder can be regarded as a *pure gain* without any lag between its input and output. The time lag between the two time series of crank angles demonstrates the dynamics of the position sensor. The entire stroke of the VCA was employed for this exercise.

In order to impose dynamic loading on the two spherical joints in the experimental setup, the slider (i.e., VCA) followed a sinusoidal position signal oscillating about its mid-stroke. The corresponding position profile is characterized by an amplitude of $920.45 \mu\text{m}$ (≈ 5000 ADC counts). The system responses for three different frequencies were recorded and are graphically presented in Fig. 7.5. When the input frequency was low (i.e., 1.0 Hz), the inertial forces on the two spherical joints were negligible. Furthermore, the dynamics of the position sensor did not significantly effect the experimental observations because the corresponding slider velocity was relatively low. However, at higher frequencies, the crank angles showed obvious hysteresis-

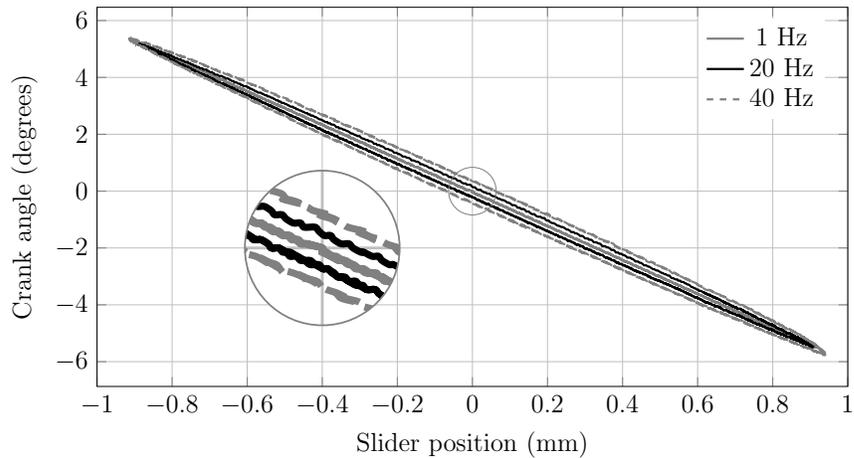


Figure 7.5: Experimental results obtained from the slider-crank mechanism.

like behavior. While it was partly caused by the dynamics of the position sensor, the inertial forces associated with the oscillating motion at high frequencies also contributed by overcoming the magnetic preloads in the two spherical joints. Therefore, it can be concluded that the anti-backlash capabilities of the proposed design is not infinite. However, since the proposed spherical joint design was primarily developed for constructing small-scale PKMs, its performance, as quantified by experimental means, is considered to be sufficient for the desired application.

7.3 Mechanical Design of POM Prototypes

The task of synthesizing the dimensions of the candidate POM architectures described in Chapter 4 was regarded as an optimization exercise constrained by the kinematics of the manipulators. The issues involving link interference were consciously excluded from the problem formulation because the corresponding computational complexity would have been unmanageably high. Nonetheless, prior to implementation, each the synthesized designs must be thoroughly examined in order to ensure that no link interference exists in their respective reachable workspace. To this end, virtual proto-

types of the two candidate POMs were developed in a CAD environment. Subsequent evaluation of these CAD models identified several link interferences, especially at the boundaries of the reachable workspace. In order to eliminate these occurrences of link collision, the optimal geometries of the two manipulators were slightly adjusted by trial & error. The optimal and the revised geometries are presented in Table 7.1. In addition, the corresponding kinematic performances are reported in Table 7.2. It can be seen from the data presented in Table 7.2 that the revised geometries provide near optimal performance in all aspects except the minimum dexterity of the reachable workspace. However, it was observed that the least dexterous regions of the workspace can be reached only when the actuators are fully extended or fully retracted (see Fig. 4.7). Because the two extremes of the actuator stroke are not usable due to practical reasons (see Section 3.3), the POMs never operate in these regions of low dexterity. Therefore, the unfavorable low value of the minimum dexterity metrics provided by the revised geometries were considered to be inconsequential from a practical point of view. It should be noted that some of the data shown in Table 7.2 were previously cited in Table 4.3. The virtual prototypes of the manipulators employing the revised geometries are presented in Fig. 7.6. In addition, fully functional physical prototypes are shown in Fig. 7.7.

7.3.1 Post Optimality Analysis

Studying the effects of small perturbations in the geometry of a parallel robot is referred to as *post optimality analysis* in [27, p. 307]. Since manufacturing tolerances cannot be avoided when a design is implemented in the physical domain, a well executed post optimality study ensures that a physical prototype, whose geometry is only an *approximation* of the synthesized dimensions, can indeed provide the

Table 7.1: Revised POM Geometry

3-PSS/S Configuration	Geometry	Design Kinematic Parameters ¹				
		r	h	R	H	θ
		mm	mm	mm	mm	degrees
Parallel	Optimal	12.00	1.56	20.52	28.68	82.55
	Revised	12.50	1.20	19.20	26.04	76.64
Slanted	Optimal	14.40	0.84	15.00	22.92	89.88
	Revised	14.40	2.40	15.00	24.00	78.51

¹Refer to Table 4.1

Table 7.2: Kinematic Performance Metrics Provided by the Optimal and the Revised POM Geometries

3-PSS/S Configuration	Geometry	Kinematic Performance Metrics ¹				
		Regular Tilt	Workspace Volume	Workspace Dexterity	Minimum Dexterity	Torsional range
		degrees	rad ³	–	–	degrees
Parallel	Optimal	32.5195	0.7382	0.6618	0.2670	102.2666
	Revised	30.6738	0.6167	0.6736	0.1286	102.7926
Slanted	Optimal	36.1450	1.0576	0.6597	0.3206	108.9753
	Revised	34.8926	0.9926	0.6315	0.3217	116.5199

¹Refer to Section 4.6

Table 7.3: Variability of Kinematic Performance Metrics Observed in the Post Optimality Study

Kinematic Performance Metric	Parallel Configuration		Slanted Configuration	
	Mean	σ	Mean	σ
Regular tilt (degrees)	30.2794	0.2497	34.7721	0.1267
Workspace volume (rad ³)	0.6007	0.0139	0.9920	0.0080
Workspace dexterity	0.6778	0.0020	0.6314	0.0016
Minimum dexterity	0.2497	0.0107	0.3147	0.0054
Torsional range (degrees)	95.4451	0.9928	116.4965	0.5533

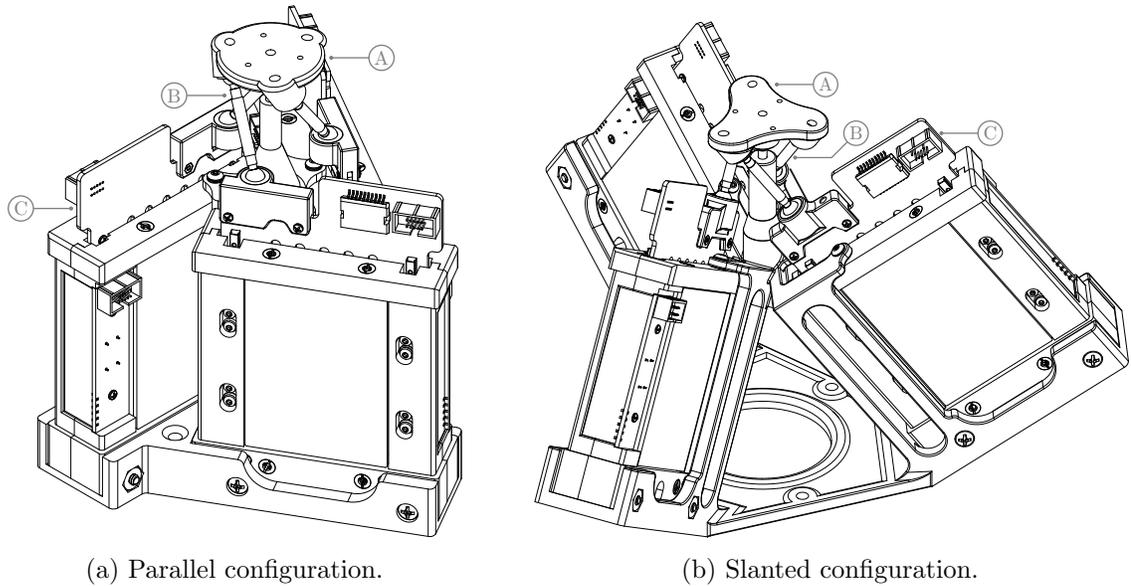


Figure 7.6: Virtual prototypes of the parallel and the slanted configurations of the 3-PSS/S architecture (not drawn to scale). A: moving platform, B: intermediate link, C: linear actuator.

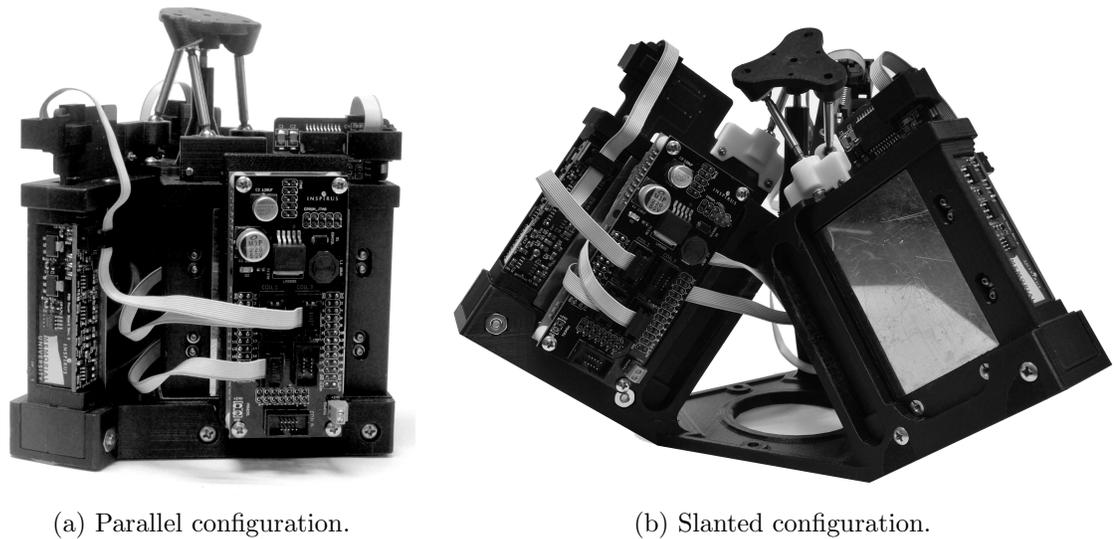
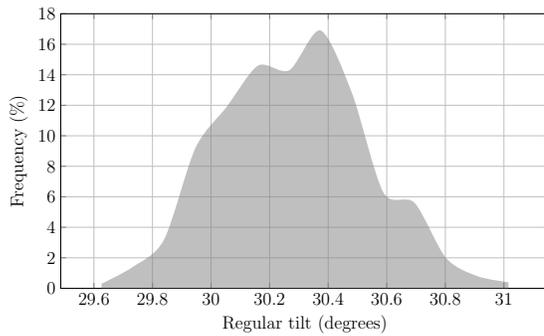
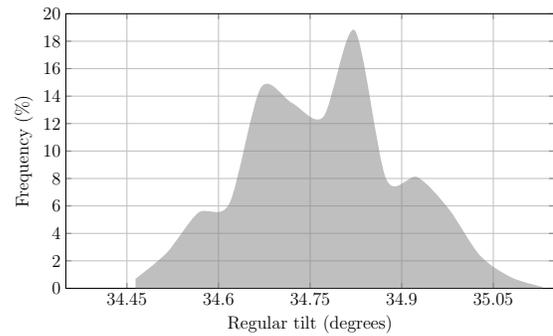


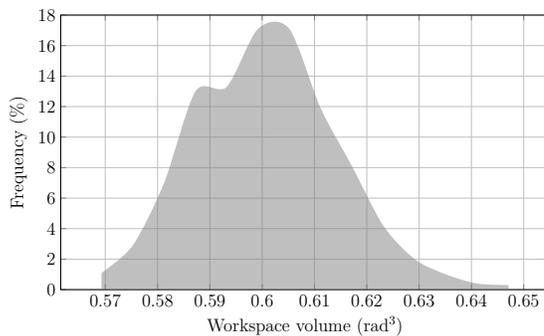
Figure 7.7: Prototypes of the parallel and the slanted configurations of the 3-PSS/S architecture (not shown to scale).



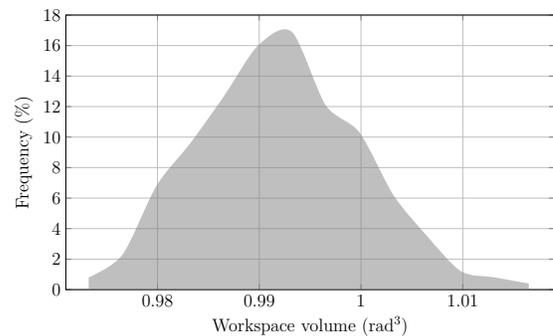
(a) Regular tilt (parallel configuration).



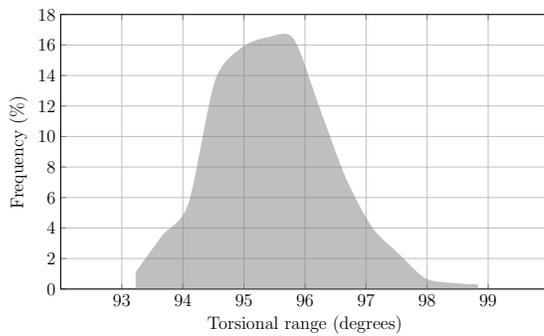
(b) Regular tilt (slanted configuration).



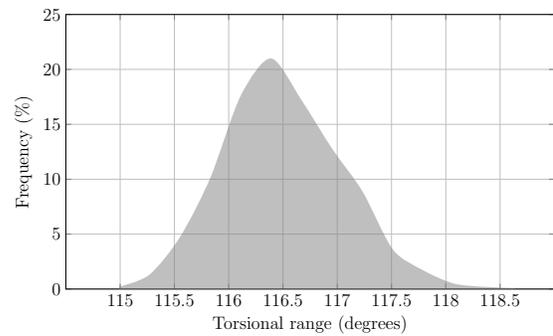
(c) Workspace volume (parallel configuration).



(d) Workspace volume (slanted configuration).



(e) Torsional range (parallel configuration).



(f) Torsional range (slanted configuration).

Figure 7.8: Histograms of workspace related kinematic performance metrics sampled in the post optimality study.

specified kinematic performance within an acceptable margin. In order to simulate the geometrical variations that may occur during manufacturing, the articulation points of the manipulators (i.e., joint locations) and the actuated joint axes were perturbed by a small magnitude in each trial of the post optimality study. A total

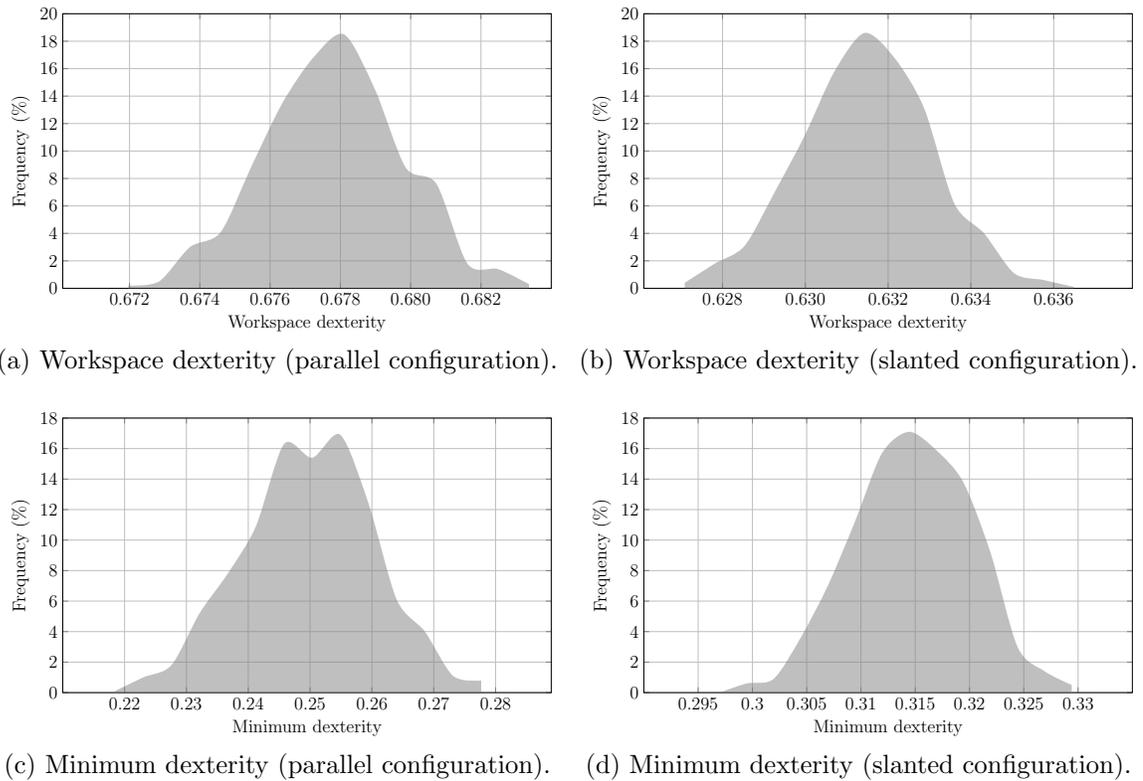


Figure 7.9: Histograms of dexterity related kinematic performance metrics sampled in the post optimality study.

of 1000 such trials were conducted to evaluate each POM design. In these trials, the random errors in all three coordinates of the articulation points were characterized by a uniform probability distribution with a range of ± 0.1 mm. In addition, the tilt angle between the perturbed actuation axis and the nominal axis was randomly selected from a uniform probability distribution defined over the range $[0, 2]$ degrees. The corresponding azimuth angle was chosen arbitrarily. It should be noted that the manufacturing processes employed to construct the POM prototypes (e.g., CNC machining, precision 3D printing) are capable of delivering manufacturing tolerances superior than those used in this post optimality study.

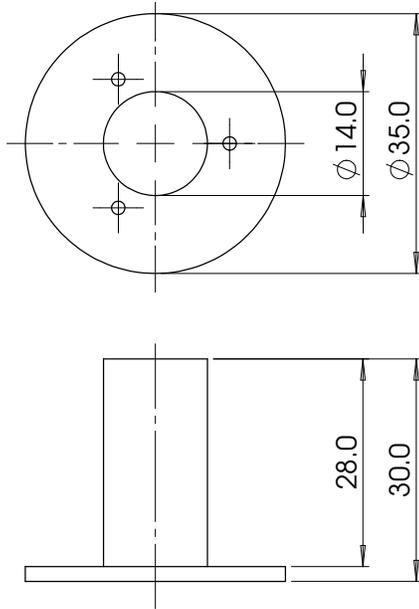
The statistics of the data recorded in the post optimality study are presented in Table 7.3. In addition, the corresponding histograms are shown in Figures 7.8

and 7.9. It can be concluded from the presented data that the revised geometries of the two manipulators exhibit robust performance in the presence of lax manufacturing tolerances. However, the kinematic performance metrics provided by the slanted configuration of the 3-PSS/S manipulator generally exhibited smaller variability. Obviously, the slanted configuration of the actuators contributed towards its robustness against geometric tolerances.

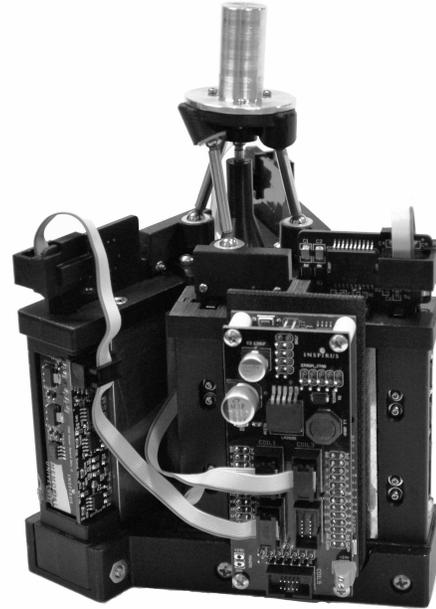
7.4 Dynamic Performance Characterization

In order to orientate the moving platform as desired, an ADRC controller was designed to regulate the actuated joint coordinates through three SISO control loops. The disturbance rejection feature of the controller compensated for the cross-coupling effects originating from the MIMO structure of the manipulators. Detailed discussions on the formulation and the implementation of the controller can be found in Chapter 6. Although the controller architecture remained identical for different manipulator configurations and payloads, it was tuned specifically for each test case. The tuning procedure described in Section 6.3.3 provided a set of approximate values for the controller parameters. In order to address the specific requirements of the plant (e.g., oscillations in the control signal, unacceptably large steady state errors, overshoot, etc.), these values were slightly adjusted in a subsequent step. It is relevant to mention that this tuning exercise emphasized the speed of manipulation over the steady state performance. Superior performance in terms of smaller steady state errors can be achieved by tuning the controller more conservatively. However, such an improvement is obtained at the cost of reduced manipulation speed.

It should be mentioned that the ADRC controller was designed to implement setpoint control. Since the controller employed transient motion profiles for the



(a) Mechanical drawing of the test payload (all dimensions are in mm).



(b) Test payload mounted on the moving platform.

Figure 7.10: Test payload employed for quantifying the transient response performance of the manipulator prototypes.

actuators to follow during a maneuver, it may appear that achieving tracking control was the design objective. In fact, the motion profiles were incorporated into the control architecture in order to keep the actuators from saturating their capacities when a maneuver was executed.

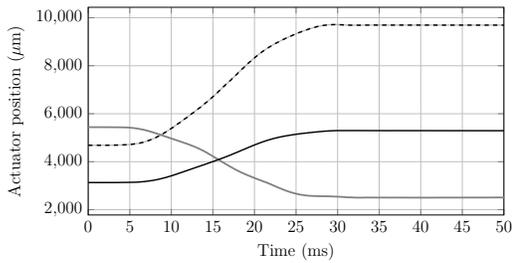
7.4.1 Transient Response Performance

The dynamic performance metrics *settling time* and *steady state error*, as defined in Section 5.4.2, have been adopted for quantifying the dynamic performance in the joint space. In order to specify the dynamic performance in the workspace, this thesis defines two additional quantities, namely the *axis error* E_θ and the *torsion error* E_σ . The axis error of a test maneuver defines the absolute angular distance between

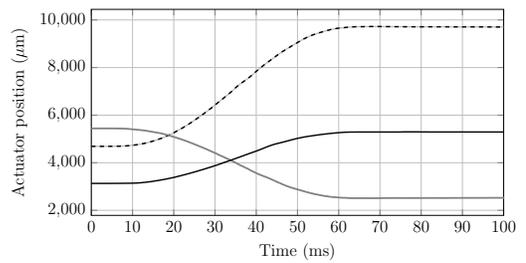
the torsion axis provided by the steady state orientation and the torsion axis of the desired orientation. In addition, the torsion error is the absolute difference between the steady state torsion angle and the torsion angle of the desired orientation. The axis about which the torsion angle is measured is defined as the torsion axis. The direct kinematic model and the nominal geometry of each POM were employed to estimate the axis error and the torsion error.

An experiment comprised of 200 test maneuvers was designed to quantify the transient response performance. Each test maneuver began with the three actuators resting at predefined initial positions. The ADRC controller regulated the actuators in order to reach the desired final positions. The initial and the final positions were expressed in terms of ADC counts and were randomly chosen from the experimental range [5000, 60000]. Although the entire stroke of the actuator is theoretically represented by the range $[0, 65535=2^{16}-1]$, the experimental space was truncated due to reasons explained in Section 3.3. Two trials of the designed experiment were conducted for each prototype. Only the masses of the moving components of the manipulator (e.g., actuator pistons, intermediate links, and the moving platform) constituted the inertial load in the first trial. However, a payload of 53.80 g was mounted on the moving platform for the second trial (see Fig. 7.10). The principal moments of inertia $[3.39, 6.62, 6.62] \times 10^{-6}$ kgm² of the test payload were estimated by a CAD application. Table 7.4 reports the transient performance observed in all four experimental trials.

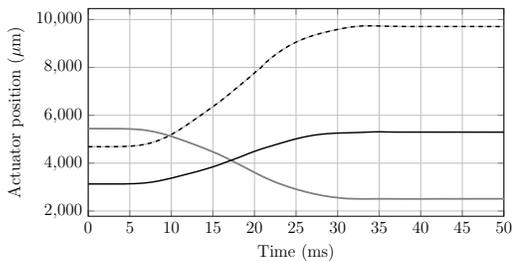
It can be seen from the data presented in Table 7.4 that the performance metrics for both manipulators are closely comparable when no test payload was involved. However, with the test payload, the parallelly configured manipulator convincingly outperformed the slanted configuration. It is highly unlikely that suboptimal controller parameters could cause performance disparity of such a large magnitude. In order to develop a



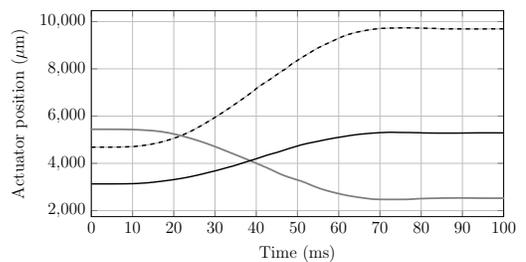
(a) Actuator positions (no payload, parallel configuration).



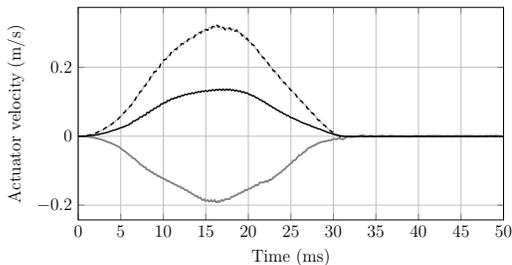
(b) Actuator positions (with payload, parallel configuration).



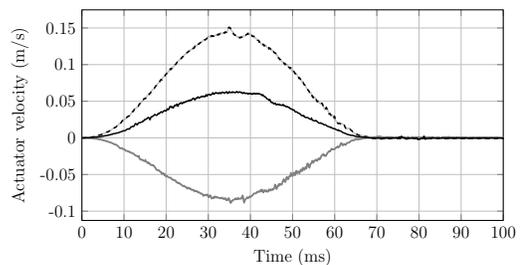
(c) Actuator positions (no payload, slanted configuration).



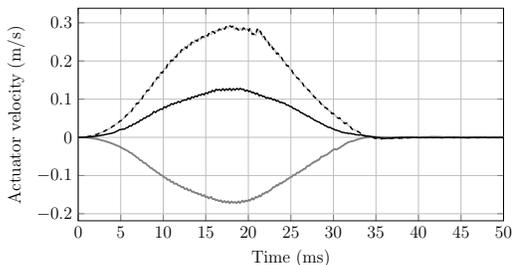
(d) Actuator positions (with payload, slanted configuration).



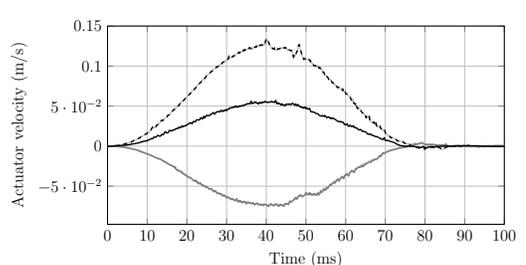
(e) Actuator velocities (no payload, parallel configuration).



(f) Actuator velocities (with payload, parallel configuration).

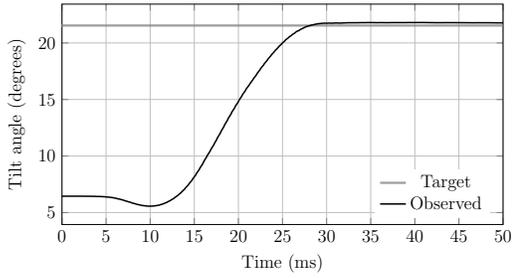


(g) Actuator velocities (no payload, slanted configuration).

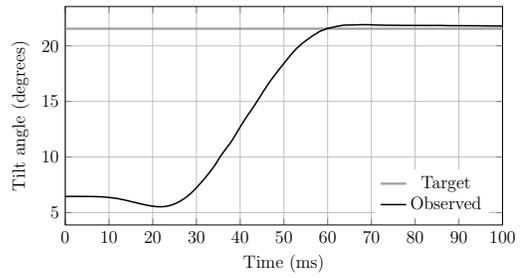


(h) Actuator velocities (with payload, slanted configuration).

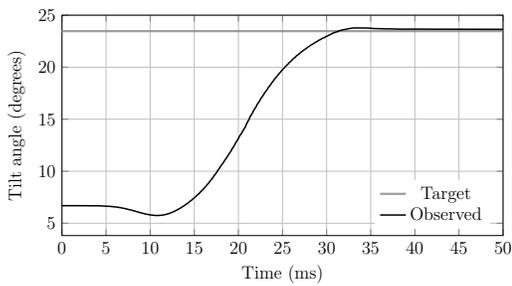
Figure 7.11: Joint space responses observed in a representative test maneuver (--- Actuator 1, — Actuator 2, — Actuator 3).



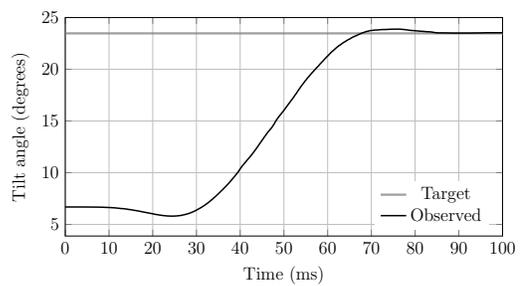
(a) Tilt angle (no payload, parallel configuration).



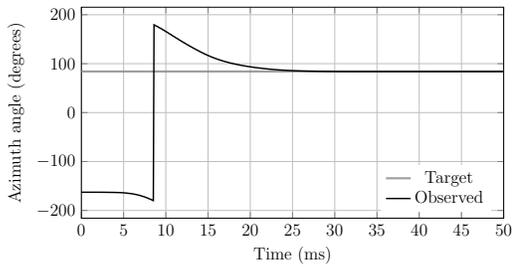
(b) Tilt angle (with payload, parallel configuration).



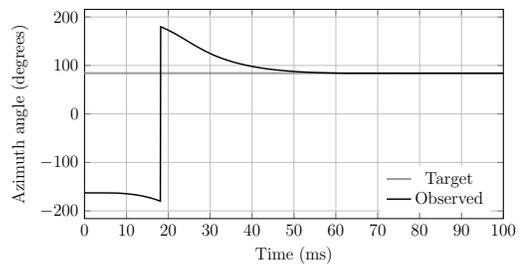
(c) Tilt angle (no payload, slanted configuration).



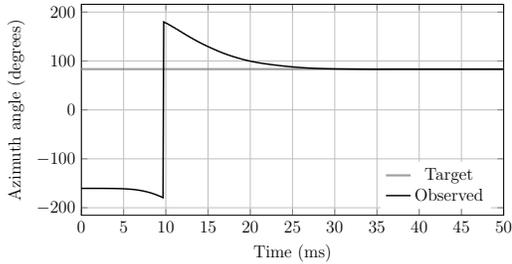
(d) Tilt angle (with payload, slanted configuration).



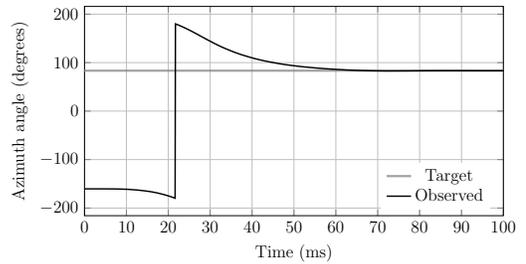
(e) Azimuth angle (no payload, parallel configuration).



(f) Azimuth angle (with payload, parallel configuration).

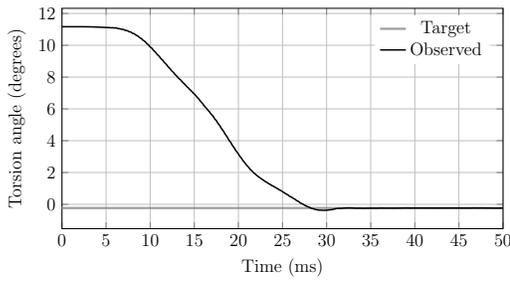


(g) Azimuth angle (no payload, slanted configuration).

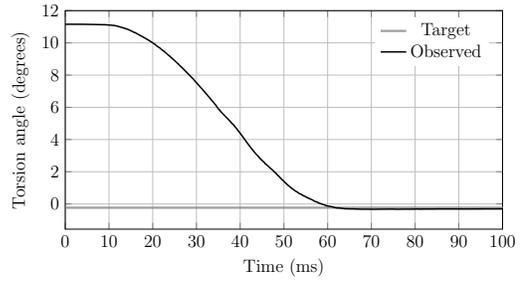


(h) Azimuth angle (with payload, slanted configuration).

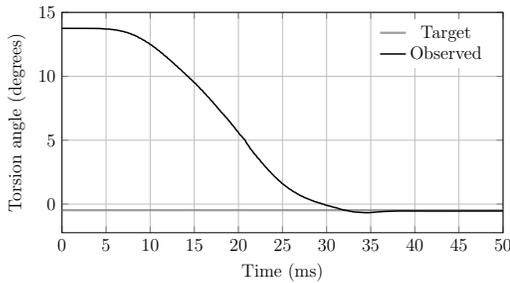
Figure 7.12: Tilt and azimuth angles observed in a representative test maneuver.



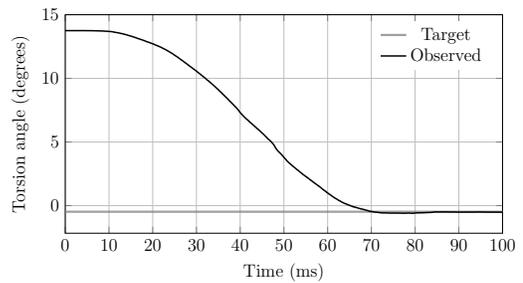
(a) Torsion angle (no payload, parallel configuration).



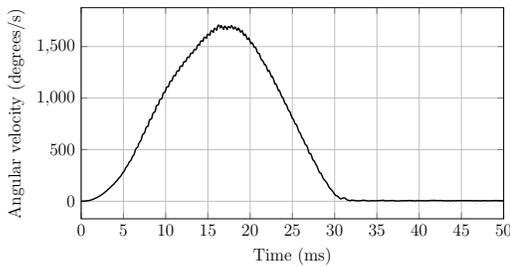
(b) Torsion angle (with payload, parallel configuration).



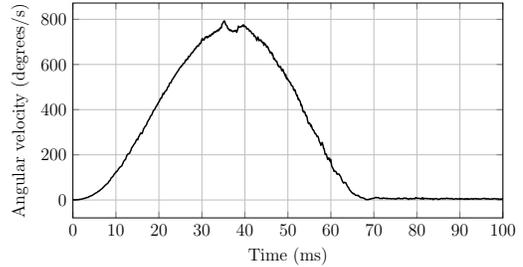
(c) Torsion angle (no payload, slanted configuration).



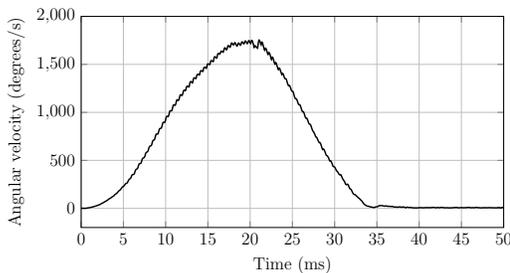
(d) Torsion angle (with payload, slanted configuration).



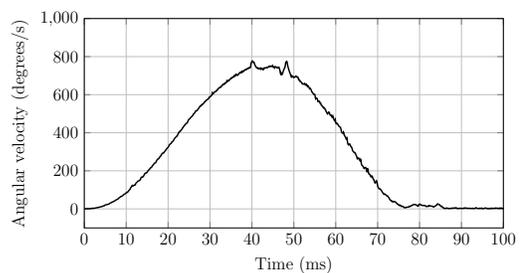
(e) Magnitude of the angular velocity (no payload, parallel configuration).



(f) Magnitude of the angular velocity (with payload, parallel configuration).



(g) Magnitude of the angular velocity (no payload, slanted configuration).



(h) Magnitude of the angular velocity (with payload, slanted configuration).

Figure 7.13: Torsion angles and magnitudes of the angular velocity of the moving platform observed in a representative test maneuver executed by the manipulator prototypes.

Table 7.4: Transient response performance provided by the two configurations of the 3-PSS/S manipulator under different payload conditions

Manipulator configuration	Performance metric	No payload		With payload	
		Mean	σ	Mean	σ
Parallel	ST (ms)	35.0085	11.6729	65.1858	11.1295
	SSE ₁ (μm)	5.1068	3.3521	8.5107	6.5298
	SSE ₂ (μm)	4.0947	2.3555	6.4681	4.1990
	SSE ₃ (μm)	5.4614	3.2657	8.5742	6.2145
	E_θ (mrad)	0.5163	0.2373	0.8575	0.4130
	E_σ (mrad)	0.3949	0.2762	0.6387	0.4852
Slanted	ST (ms)	36.0407	9.6142	80.6539	18.1918
	SSE ₁ (μm)	4.6410	3.3793	6.4221	4.0339
	SSE ₂ (μm)	4.8337	2.7570	7.7822	5.0932
	SSE ₃ (μm)	4.8466	2.8366	5.6327	3.6405
	E_θ (mrad)	0.6104	0.3278	0.8018	0.4281
	E_σ (mrad)	0.3761	0.2774	0.6403	0.4396

ST = Settling time, SSE_{*n*} = Steady state error in actuator *n*
 E_θ = Axis error, E_σ = Torsion error

conjecture as to why this discrepancy happened, the geometric structures of the two manipulator prototypes must be examined closely. Because of the large workspace volume, the payload of the slantedly configured manipulator undergoes an angular displacement greater than that obtained from identical actuator displacements in the parallelly configured manipulator. As a result, the energy requirement for executing identical displacements in the joint space is greater for the slantedly configured manipulator. Since this energy must be supplied by the limited capacities of the actuators, they require a longer time period to supply the demanded energy. The recorded responses in a representative test maneuver from all four experimental trials are graphically presented in Figures 7.11, 7.12, and 7.13.

7.4.2 Demonstration of High Speed Angular Motion

The cross-coupling effects in the 3-PSS/S manipulator are generally observed to become increasingly prominent with the speed of the actuators. Since the limited

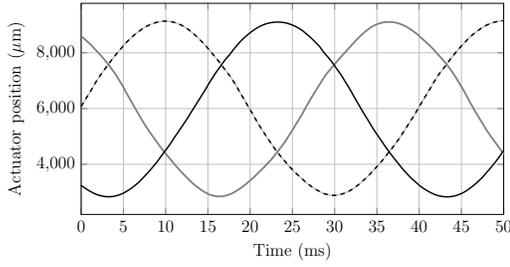
actuation capacity is exhausted to accelerate the actuators to high speeds, little to no capacity remains available to compensate for the corresponding high degree of cross-coupling nonlinearities. As a result, the controller is no longer able to perform as desired. In order to avoid such situations, the transient motion profiles employed in the ADRC controller must be designed conservatively so that an acceptable compromise between the speed of manipulation and the global controllability can be obtained. In other words, the transient motion profiles must ensure that the actuators are able to follow the prescribed motion without creating uncontrollable cross-coupling effects. The ADRC controllers employed in the transient response experiment were formulated accordingly. Although globally robust performance was obtained by adopting this design philosophy, the available dynamics of the manipulator prototypes were not fully utilized. Correspondingly, this section aims to showcase the dynamic capacity of the prototyped manipulators, especially when achieving globally robust performance is no longer a constraint. To this end, two different test motions were prescribed for each of the manipulators. For the sake of nomenclature, these test motions are referred to as test motion A and test motion B.

Since the manipulators were allowed to perform the prescribed motions without any external payload, the controller parameters corresponding to the no payload trials from the transient response experiment were adopted. In addition, the cycloidal motion profiles of the ADRC controller were replaced with sinusoidal motion profiles. Correspondingly, the trial motions were sinusoidal in nature, and they were identical in terms of amplitude, frequency, and offset. In both trial motions, the amplitude of all transient motion profiles was chosen as 16384 ADC counts ($\approx \frac{1}{4}$ of actuator stroke). In addition, an offset of 32768 ADC counts was included in all three position profiles so that the actuators oscillated about their respective mid-stroke position during the test motions. Furthermore, a phase angle ϕ_i was assigned for each individual

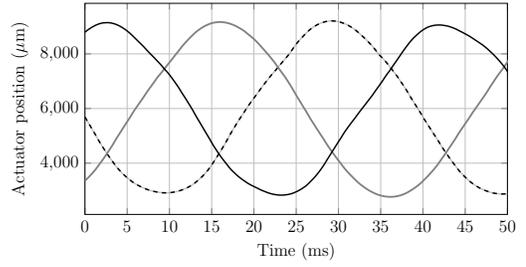
transient position profile. In test motion A, the phase angles were purposely chosen as $\phi_1 = 0^\circ$, $\phi_2 = 120^\circ$, and $\phi_3 = 240^\circ$ so that each prescribed sinusoidal motion maintained a phase difference of 120° with respect to the other two. In contrast, the transient profiles for test motion B were designed to be in phase; i.e., $\phi_1 = \phi_2 = \phi_3 = 0^\circ$.

The frequencies of the sinusoidal profiles were kept identical for all three actuators in each experimental trial. An input frequency of 25 Hz was chosen for all experimental trials as a compromise between tracking performance and maximum angular speed attained by the moving platforms. The corresponding joint space responses shown in Figures 7.14 and 7.15 confirm that the prescribed motions were followed well by the two manipulators. The motion of the moving platforms are presented in Figures 7.16 and 7.17. Since the sinusoidal displacements of the three actuators in test motion B were all in-phase with identical amplitude and offset values, the corresponding motion of the moving platform was primarily torsional. Correspondingly, Fig. 7.17(a) and Fig. 7.17(b) do not show the tilt and the azimuth angles. In addition, empirical models for these torsional motions were developed by fitting each set of recorded data to a sum of sines model of order 4. These models were later employed in Fig. 7.20 to estimate the angular velocity and the corresponding angular acceleration of the moving platforms. It should be noted that these empirical models are a close approximation of the true angular motion of the moving platform because the components of the corresponding angular velocity vector along any axis other than the vertical axis of the inertial frame were negligible.

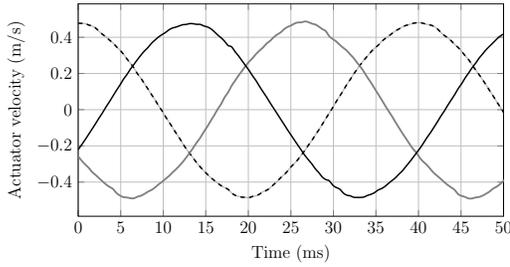
Fig. 7.18 graphically presents the magnitudes of the angular velocity vector of the moving platform observed in test motion A. It can be clearly seen that the parallelly configured manipulator exhibited angular velocities with magnitudes exceeding 2000 degrees/s. However, the slanted configuration of the manipulator was observed to attain angular velocities of relatively higher magnitudes. In specific terms, the



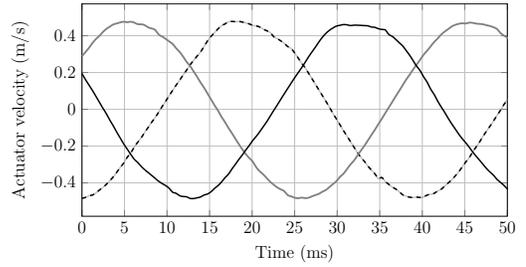
(a) Actuator positions (parallel configuration).



(b) Actuator positions (slanted configuration).

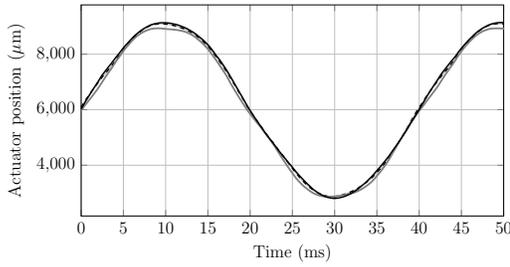


(c) Actuator velocities (parallel configuration).

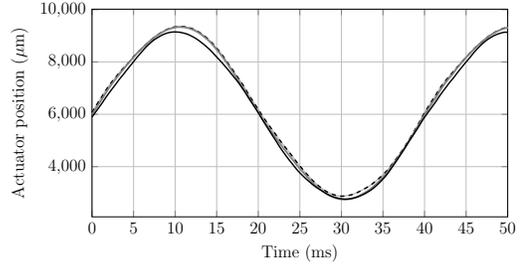


(d) Actuator velocities (slanted configuration).

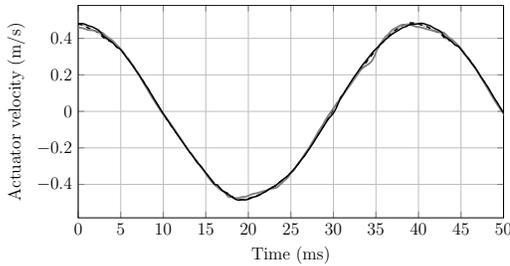
Figure 7.14: Joint space responses observed in test motion A (--- Actuator 1, — Actuator 2, — Actuator 3).



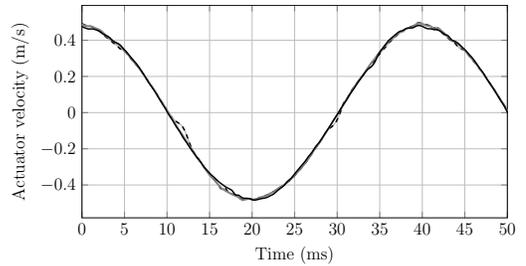
(a) Actuator positions (parallel configuration).



(b) Actuator positions (slanted configuration).

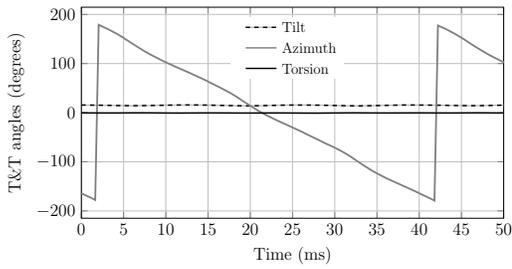


(c) Actuator velocities (parallel configuration).

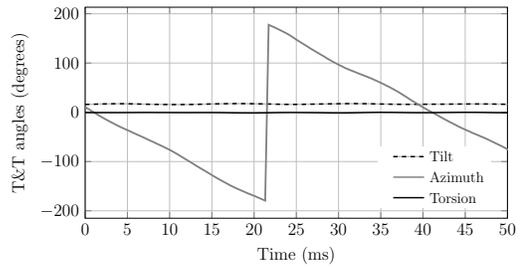


(d) Actuator velocities (slanted configuration).

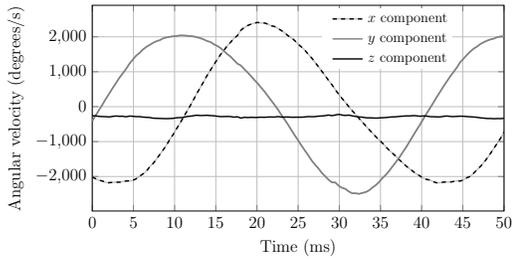
Figure 7.15: Joint space responses observed in test motion B (--- Actuator 1, — Actuator 2, — Actuator 3).



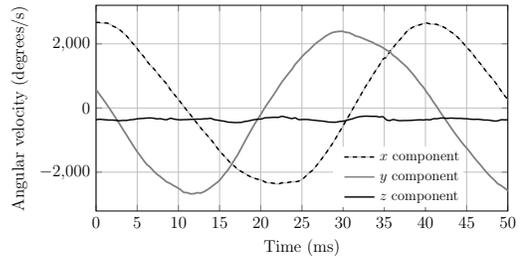
(a) T&T angles (parallel configuration).



(b) T&T angles (slanted configuration).

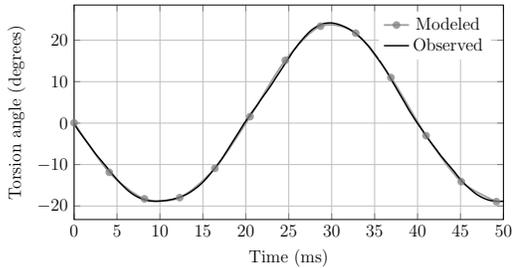


(c) Angular velocity vector of the moving platform (parallel configuration).

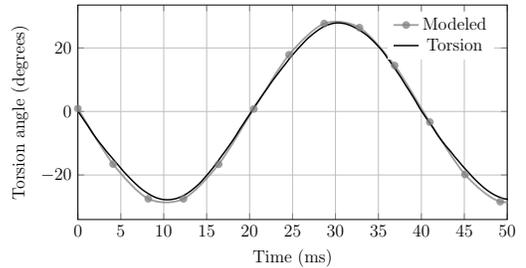


(d) Angular velocity vector of the moving platform (slanted configuration).

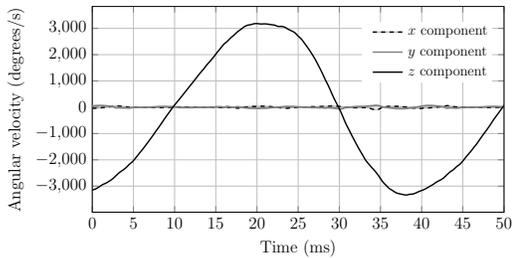
Figure 7.16: Workspace responses observed in test motion A.



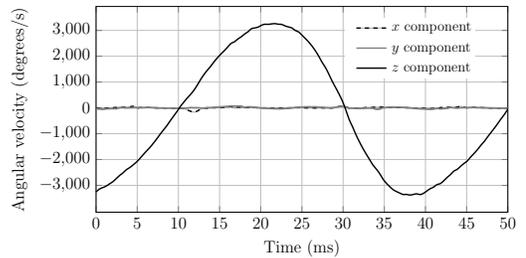
(a) Torsion angle (parallel configuration).



(b) Torsion angle (slanted configuration).

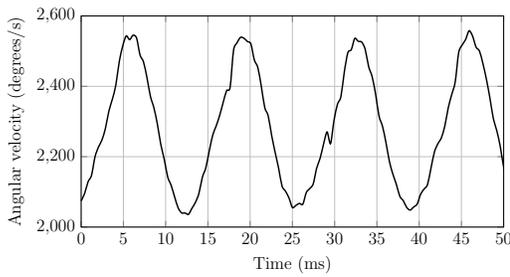


(c) Angular velocity vector of the moving platform (parallel configuration).

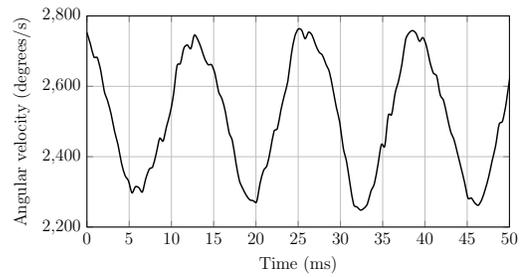


(d) Angular velocity vector of the moving platform (slanted configuration).

Figure 7.17: Workspace responses observed in test motion B.

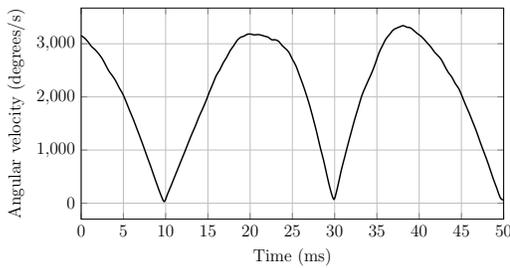


(a) Magnitude of the angular velocity vector of the moving platform (parallel configuration).

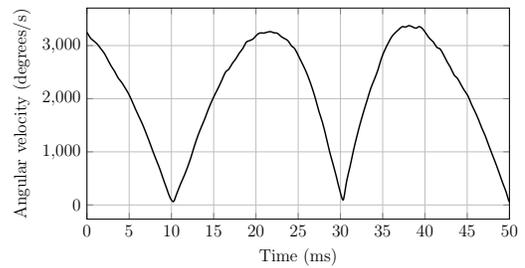


(b) Magnitude of the angular velocity vector of the moving platform (slanted configuration).

Figure 7.18: Magnitudes of angular velocity vector observed in test motion A.



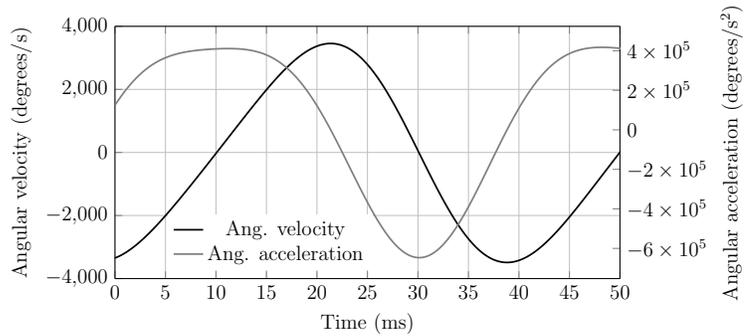
(a) Magnitude of the angular velocity vector of the moving platform (parallel configuration).



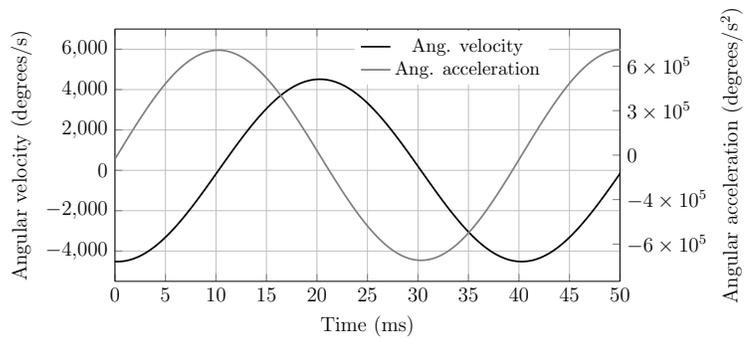
(b) Magnitude of the angular velocity vector of the moving platform (slanted configuration).

Figure 7.19: Magnitudes of angular velocity vector observed in test motion B.

magnitude of the the angular velocity vector never fell below 2200 degrees/s [see Fig. 7.18(b)]. In general, test motion A demonstrated that both of the manipulators can achieve and maintain angular velocities exceeding 2000 degrees/s. However, the purpose of test motion B was to showcase the high angular acceleration that the manipulators can generate. Correspondingly, the angular accelerations estimated from the recorded data are presented in Fig. 7.20. Both manipulators were shown to generate angular acceleration of magnitudes greater than 5×10^5 degrees/s².



(a) Angular velocity and angular acceleration of the moving platform estimated from empirical models (parallel configuration).



(b) Angular velocity and angular acceleration of the moving platform estimated from empirical models (slanted configuration).

Figure 7.20: Magnitudes of angular velocity vector observed in test motion B.

7.5 Kinematic Calibration

Because of the manufacturing tolerances associated with the fabrication processes that are employed to implement a synthesized PKM design, the specified geometry cannot be reproduced *exactly*. As a result, when the direct or the inverse kinematic models are defined by the nominal geometry, the corresponding solutions provide limited accuracy. Therefore, it is a common practice to perform a kinematic calibration of the manipulator in order to identify the actual geometry [27, p. 289] so that reliable and accurate kinematic solutions can be obtained. The related literature describes a number of kinematic calibration procedures. Detailed surveys can be found in [27, 159].

In [27, p. 290], Merlet classified the calibration methods into three major types; namely, *external calibration*, *constrained calibration*, and *self calibration*. In all three types of calibration methods, virtual or real constraints are imposed on the poses on the moving platform. Obtaining these constraints at a large number of poses of the moving platform facilitates the formulation of a system of constraint equations. Solving these constraint equations provides the geometry of the manipulator that satisfies them *best* in some mathematical sense. From the above discussion, it is clear that the conventional approach towards kinematic calibration generally begins with formulating the problem in terms of constraint equations that are derived from the kinematic model of the robot. In the data acquisition phase, the pose of the moving platform and the corresponding actuated joint coordinates are obtained. Finally, a suitable optimization method utilizes the obtained data to determine the actual geometry. Sometimes *a priori* knowledge of the geometry is necessary in order to successfully determine a solution. In addition, sensor calibration may be required to estimate the corresponding gain, so that the actuated joint coordinates can be expressed in terms of a preferred absolute unit.

It was shown in Chapter 4 that response surface methodology (RSM) can be effectively employed to obtain empirical models of kinematic performance metrics such as workspace volume, GCI, and minimum dexterity. Inspired by this success, this thesis proposes a simplified approach where RSM is employed to formulate an empirical model of the 3-PSS/S kinematic structure. Such a model requires neither *a priori* knowledge of the kinematics nor calibrated sensors. This approach completely eliminates the need for formulating constraint equations. In addition, the empirical model can provide solutions to both the direct and the inverse kinematic problem. If the actuated joint coordinates were defined as independent variables, the empirical model provides a solution to the direct kinematics problem. On the other hand, defining the

workspace coordinates as independent variables provides an empirical solution to the inverse kinematics problem. It should be mentioned that the calibration data used in conventional methods (i.e., pose of the moving platform and the corresponding joint coordinates) is sufficient for the proposed approach. Although the proposed approach was developed for the 3-PSS/S kinematic architecture, it remains to be seen whether it can be employed to calibrate parallel robots of higher complexity. However, this avenue of research is deferred to future work. It is relevant to report that the parallel configuration of the 3-PSS/S manipulator was arbitrarily chosen to experimentally verify the practicality of the proposed calibration method.

7.5.1 Parameterization of Empirical Kinematic Models

In order to obtain an empirical model of the direct kinematics of the 3-PSS/S manipulator, the actuated joint coordinates d_1 , d_2 , and d_3 were chosen as the independent variables. Three additional variables that define the orientation of the moving platform constituted the response parameters. Although the tilt-torsion angles $[\theta, \phi, \sigma]^T$ from Chapter 4 were an intuitive choice in this regard, it was found to be difficult for a multivariate polynomial to represent the transcendental relationship that the tilt-torsion angles exhibit with respect to the actuated joint coordinates, especially at the moving platform poses where the azimuth angle is discontinuous. In specific terms, these poses occur when the azimuth angle approaches $+\pi$ or $-\pi$. As an alternative, the parameters in (7.1) are derived from the tilt-torsion angles to serve as the responses of the empirical direct kinematic model. For the sake of nomenclature, the parameters

are referred to as the *modified workspace coordinates* hereinafter.

$$\begin{aligned}x &= \sin \theta \cos \phi \\y &= \sin \theta \sin \phi \\z &= \sigma\end{aligned}\tag{7.1}$$

Here, θ , ϕ , and σ respectively refers to the tilt, the azimuth, and the torsion angle of a given orientation. It should be noted that the parameters x and y refers to the projection of the *torsion axis* on the xy plane of the inertial coordinate frame. It was experimentally confirmed that each of the three responses in (7.1) can be successfully modeled as a polynomial function of the actuated joint coordinates d_1 , d_2 , and d_3 .

In the empirical model of the inverse kinematics of the 3-PSS/S manipulator, the modified workspace coordinates defined in (7.1) are selected as the independent variables. Naturally, the corresponding responses are provided by the actuated joint coordinates.

7.5.2 Design of Experiment

In order to estimate the direct kinematic model, an IV-Optimal (integrated variance) experiment was designed. The experimental points were determined by the *coordinate exchange* algorithm [160] under the assumption of a fifth order model. Each of the 66 experiment points thus determined is defined by a set of three actuated joint coordinates. Although the choice of the model order is arbitrary, an experiment designed for estimating a higher order model minimizes the chances of inadequate representation of an unexplored response surface. On the other hand, an experiment designed for a higher order model involves a greater number of sample points, which adds to the resource cost for data acquisition. Therefore, the model order should be

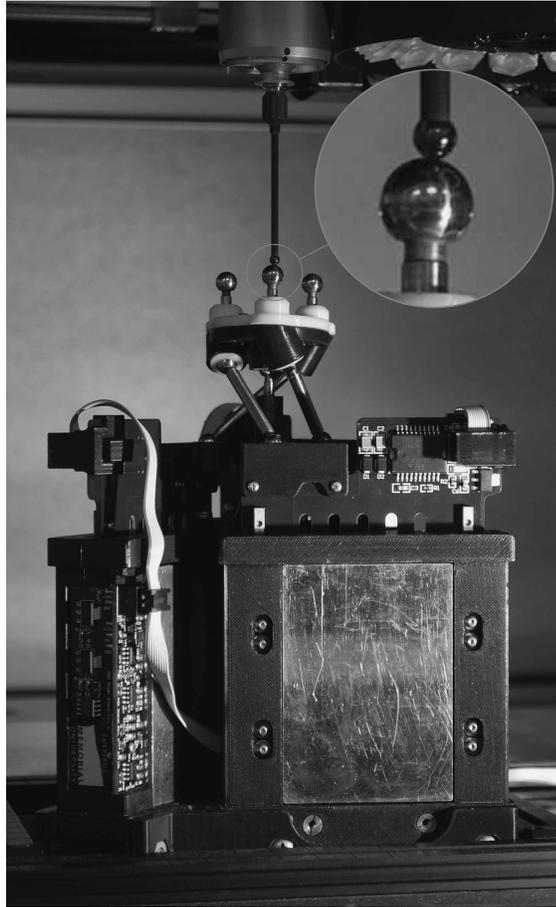


Figure 7.21: Data acquisition for kinematic calibration. Inset: CMM probe measuring the location of the sphere center.

selected as a compromise between model adequacy and experiment cost. Nonetheless, the analytic kinematic model provides a qualitative sense of the complexity of the response surface. Without any modification, the input-output data obtained for estimating the empirical direct kinematics model was employed for estimating the empirical inverse kinematics model.

7.5.3 Data Acquisition & Model Construction

An ADRC controller was employed to obtain the desired actuated joint coordinates defined by the experimental points. Although some steady state errors were observed in the actuated joint coordinates, their magnitudes were small enough to be considered insignificant to the overall sampling of the experiment space. The actuated joint coordinates were provided by the integrated position sensors. In a subsequent step, a coordinate measurement machine (CMM) was employed to determine the orientation of the moving platform. To this end, the spherical surfaces of three tooling balls mounted rigidly on the moving platform were probed by the CMM (see Fig. 7.21) so that the corresponding spherical centers could be localized. These centers represented three points that defined the body-fixed coordinate frame. The native coordinate frame of the CMM was selected as the inertial frame. It should be noted that the body-fixed coordinate frame and the inertial coordinate frame did not necessarily coincide at the home position of the manipulator.

As seen in Fig. 7.21, the tooling balls were designed to be equidistant from the mechanism center. In addition, they were purposely patterned to form an equilateral triangle. Such an arrangement ensured that the measurement error in each spherical center did not cause any imbalanced bias in the experimental observations. It should be emphasized that employing a CMM for measurements is not an absolute necessity for successfully performing the proposed kinematic calibration method. As long as the orientation of the moving platform can be determined with acceptable accuracy, the choice of the measurement method is not significant. For example, vision based measurement techniques similar to [159] can also be used, provided that the corresponding measurement errors are not too high to obtain a RSM model that provides adequate accuracy.

Table 7.5: Details of the empirical direct kinematic model

Response	Model order	Coefficient of determination, R^2		
		Actual	Adjusted	Predicted
x	Reduced quartic	1.0000	1.0000	0.9999
y	Reduced quartic	1.0000	1.0000	1.0000
z	Reduced fifth	1.0000	1.0000	0.9999

Table 7.6: Details of the empirical inverse kinematic model

Response	Model order	Coefficient of determination, R^2		
		Actual	Adjusted	Predicted
d_1	Reduced fifth	1.0000	1.0000	0.9999
d_2	Reduced quartic	1.0000	1.0000	1.0000
d_3	Reduced cubic	1.0000	1.0000	1.0000

After acquiring the actuated joint coordinates and the corresponding moving platform orientation for each trial point, the modified workspace coordinates for the RSM models were calculated from (7.1). It should be noted that the empirical direct kinematic model provides the modified workspace coordinates as functions of the actuated joint coordinates. On the other hand, the empirical inverse kinematic model provides the actuated joint coordinates as functions of the modified workspace coordinates. It is also relevant to mention that the empirical model can be different for manipulators with identical kinematic architecture depending on the measurement accuracy, the dimensions of the PKM, and the resolution of the actuated joint sensors. Since the empirical models were constructed employing standard RSM algorithms, the corresponding procedures are not discussed further for the sake of brevity. However, details of the empirical models are provided in Tables 7.5 and 7.6. In addition, a few examples of the response surfaces generated from the empirical kinematic models are shown in Figures 7.22 and 7.23.

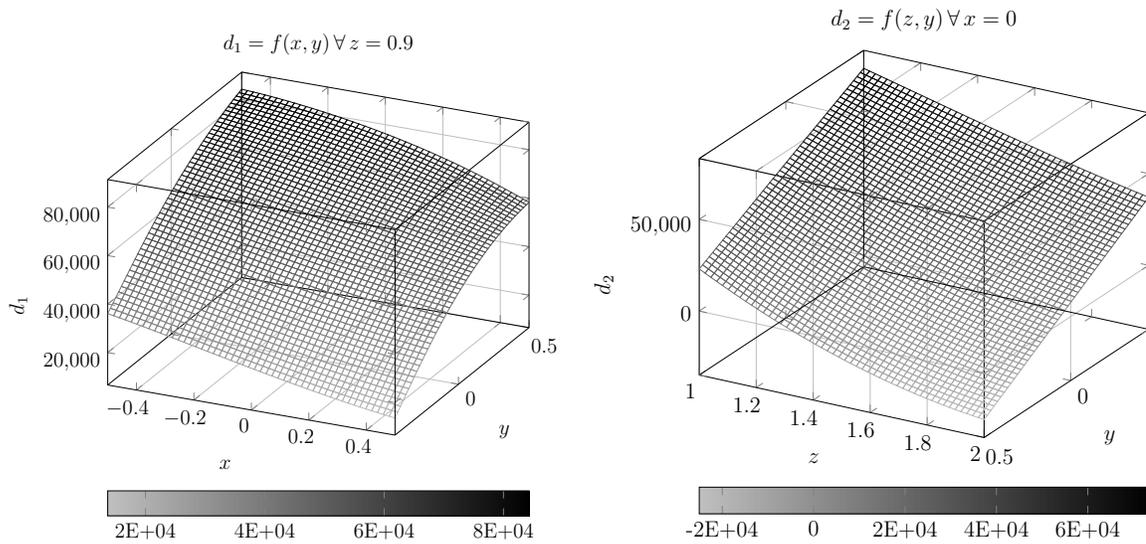


Figure 7.22: A few examples of response surfaces generated from the empirical inverse kinematic model.

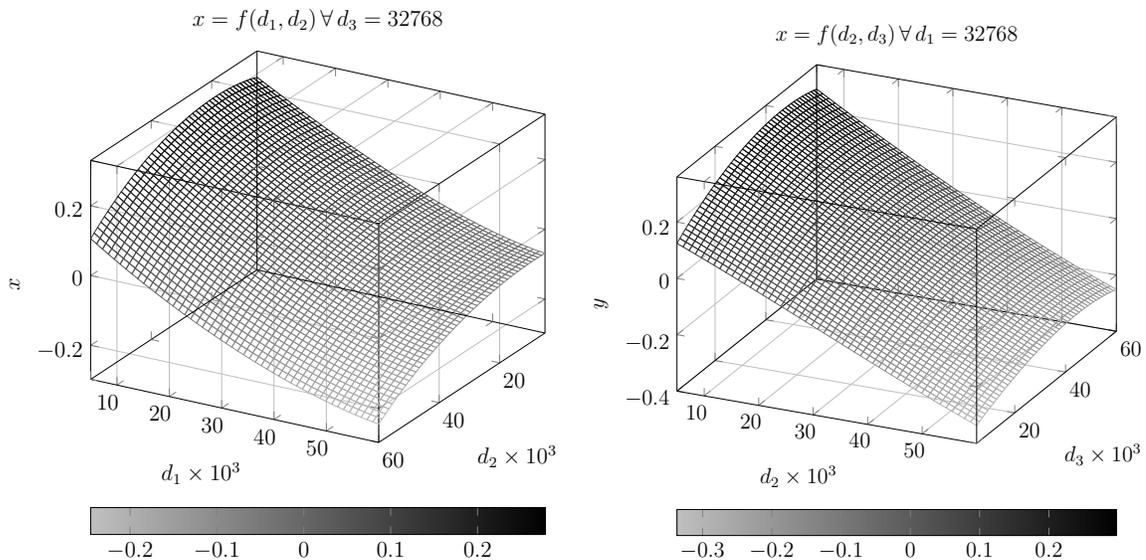


Figure 7.23: A few examples of response surfaces generated from the empirical direct kinematic model.

7.5.4 Evaluation of Model Accuracy

In the absence of any knowledge about the actual geometry of the manipulator, the accuracy of the empirical kinematic models must be assessed in terms of the residual

errors obtained from the least squares fitting of the experimental observations. To this end, the empirical direct kinematic model was arbitrarily selected for further evaluation. In order to quantify the residual errors, the axis error E_θ and the torsion error E_σ between two orientations, which are respectively obtained from the model and from experimental observations at identical joint space coordinates, were employed. It should be noted that these quantities were defined in Section 7.4.1. The statistics of the residuals in terms of the axis error and the torsion error are reported in Table 7.7. Although an analysis of the residuals can provide a means to evaluate model accuracy, it cannot benchmark the performance of the empirical model against a geometric model provided by conventional kinematic calibration methods. However, a ground truth for benchmarking purposes can still be established by performing a numerical experiment on an analytic 3-PSS/S manipulator in order to simulate the performance of a geometric model. In this experiment, it was assumed that the nominal geometry of the manipulator featuring the parallel configuration characterized the actual dimensions of the virtual POM. It should be noted that these *true* dimensions can never be known exactly in practice. However, conventional kinematic calibration methods are employed to obtain estimations of these values within a tolerance. For the sake of the numerical experiment, the accuracy of these estimations must be specified. To this end, it was assumed that the articulation points (i.e., joint locations) provided by the calibrated dimensions were within ± 0.0254 mm (= 1 mil) of their *actual* locations at the home position (i.e., all actuators at mid-stroke). In other words, the estimation error of the calibration method was arbitrarily quantified by a uniform probability distribution over the specified range (i.e., ± 0.0254 mm). Although the actuation axes of the nominal geometry were defined to be exactly vertical, the estimations of these vectors, as provided by a conventional calibration method, are subject to measurement error. The magnitude of this angular measurement error

Table 7.7: Accuracy of the empirical direct kinematic model

Model	Axis error (mrad)		Torsion error (mrad)	
	Mean	σ	Mean	σ
Empirical	1.0749	0.5129	0.3154	0.3257
Geometric	0.4818	0.3077	0.3019	0.2559

for each actuation axis was characterized by a tilt angle sampled from a uniform probability distribution defined over the range $[0, 0.1^\circ]$. The corresponding azimuth angle was determined randomly.

The analytic dimensions perturbed by the measurement errors defined above provided the *calibrated* geometry of the manipulator in each trial of the numerical experiment. For a given set of actuated joint coordinates, the direct kinematic model defined by the calibrated geometry provided the *estimated* tilt-torsion angles. The corresponding *true* tilt-torsion angles were calculated from the nominal geometry. The axis error and the torsion error between these two sets of tilt-torsion angles quantified the accuracy of the calibrated model in terms of estimating the workspace coordinates. In each trial, the axis error and the torsion error were determined at actuated joint coordinates randomly chosen from the range $[5000, 60000]$ ADC counts. A total of 10^5 such trials were conducted. The statistics of the performance provided by the geometric model in terms of the axis errors and the torsion errors are presented in Table 7.7.

Axis error and torsion error between two orientations were employed as measures of accuracy in this exercise. While these accuracy metrics represented the discrepancy between the actual and the calculated orientations of the moving platform in the simulation study, they conveyed an entirely different connotation for the empirical model. Specifically, they represented the disagreement between the observed and the model predicted orientations because information regarding the actual orientation

was not available. Furthermore, the accuracy metrics for the geometric model were estimated from a sample space whose volume was several orders of magnitude larger than that of the empirical model. These factors, which were deemed unavoidable due to the nature of the experiment, must be acknowledged when conclusions are drawn from the experimental observations. Although the torsion errors from both models were statistically very similar, the sampled axis errors from the empirical model were relatively larger than those from the geometric model. The discrepancy in this regard can be explained by the data acquisition apparatus (i.e., the attachment with three tooling balls) employed for developing the empirical model. Since the surface normal of the plane containing the spherical centers of the three tooling balls provided the torsion axis, the measurement errors impacted its estimated value to a greater degree than that of the torsion angle. However, further examination of this conjecture is deferred to future work. Despite the inferior accuracy quantified by the large magnitude of the axis errors, the empirical direct kinematic model is considered to have performed well.

7.6 Design of a Torsion-Restricted POM

When a payload in an orientation manipulation application is axis symmetric (e.g., a laser projector), the torsional degree of freedom is no longer required. In such cases, a torsion-restricted POM is more preferable because of its relatively simple architecture. The related literature provides a number of examples of these torsion-restricted orientation manipulators; examples include [9, 161, 162]. However, this thesis focuses on the 2-PSS/U architecture from [9] because it is the torsion-restricted variant of the 3-PSS/S architecture.

The kinematic structure of the 2-PSS/U POM is presented in Fig. 7.24. The

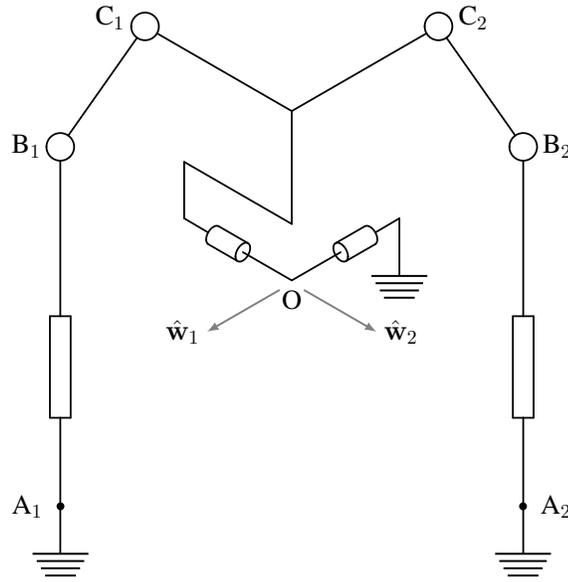


Figure 7.24: Kinematic structure of the $2\underline{P}$ -S-S/U parallel orientation manipulator.

universal joint that constraints the moving platform to the mechanical ground is decomposed into two revolute joints that are defined by their respective joint axes $\hat{\mathbf{w}}_i$ ($i = 1, 2$). The joint axes $\hat{\mathbf{w}}_i$ are not constrained in any way except that they intersect at the mechanism center O, and they are perpendicular to each other. The link A_iB_i is a linearly extensible limb (i.e., prismatic joint). Furthermore, the articulation points B_i and C_i refer to the locations of the spherical joints. The link C_1C_2 constitutes the moving platform of the manipulator.

7.6.1 Kinematic Analysis

The kinematic model of the $2\underline{PSS}$ /U architecture has been analyzed in [9, 163]. It should be mentioned that the inverse model is very similar to that of the $3\underline{PSS}$ /S architecture. However, a closed-form solution to the direct kinematics of the $2\underline{PSS}$ /U architecture is not found in the literature. The iterative solution to the direct kinematics problem provided in [9] numerically estimates one of the two

workspace coordinates, and the remaining coordinate is analytically determined in a subsequent step. In contrast, this thesis formulates the direct kinematics problem as an exactly defined system of nonlinear equations, which is iteratively solved to obtain both workspace coordinates simultaneously. In order to facilitate the formulation of the direct kinematics problem, two coordinate frames are established first. Let the coordinate frame A be fixed in space (i.e., inertial frame) and the coordinate frame B (i.e., body-fixed frame) be embedded in the moving platform. The origins of these two coordinate frames are coincident at the mechanism center O.

Without loosing any generality, let the x axis of the inertial frame A point along $\hat{\mathbf{w}}_1$ and the y axis point along $\hat{\mathbf{w}}_2$ at the home position of the manipulator. The z axis of the inertial frame A is determined by the right hand rule. In order to define the orientation of the body-fixed frame B, it is assumed that frames A and B coincide initially. The final orientation of the frame B is reached by rotating the body-fixed frame about the x axis of the inertial frame A by an angle ψ in a first rotation. A second rotation of the body-fixed frame about the rotated y axis (i.e., $\hat{\mathbf{w}}_2$ axis) of the frame B by an angle ϕ provides the final orientation of frame B. Following the aforementioned Euler angles convention, the rotation matrix ${}^a\mathbf{R}_b$ is provided by,

$${}^a\mathbf{R}_b = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ \sin \psi \sin \phi & \cos \psi & -\sin \psi \cos \phi \\ -\cos \psi \sin \phi & \sin \psi & \cos \psi \cos \phi \end{bmatrix}. \quad (7.2)$$

Geometry of the moving platform provides the position vectors of points C_i with respect to the body-fixed frame B; i.e.,

$${}^b\mathbf{c}_i = \begin{bmatrix} c_{iu} & c_{iv} & c_{iw} \end{bmatrix}^T.$$

In order to obtain ${}^a\mathbf{c}_i$,

$${}^a\mathbf{c}_i = {}^a\mathbf{R}_b \times {}^b\mathbf{c}_i = \begin{bmatrix} \cos \phi c_{iu} + \sin \phi c_{iw} \\ \sin \psi \sin \phi c_{iu} + \cos \psi c_{iv} - \sin \psi \cos \phi c_{iw} \\ -\cos \psi \sin \phi c_{iu} + \sin \psi c_{iv} + \cos \psi \cos \phi c_{iw} \end{bmatrix}. \quad (7.3)$$

Let, $\overline{B_i C_i} = \mathbf{x}_i$, $\|\mathbf{x}_i\| = x_i$, $\|\mathbf{b}_i\| = b_i$, and $\|\mathbf{c}_i\| = c_i$. Since $\mathbf{x}_i = \mathbf{b}_i - \mathbf{c}_i$, it can be written that $\|\mathbf{b}_i - \mathbf{c}_i\|^2 = x_i^2$. The equation in (7.4) can be written by expressing the vectors in coordinate frame A.

$$f_i := \|{}^a\mathbf{b}_i - {}^a\mathbf{c}_i\|^2 - x_i^2 = 0 \quad (7.4)$$

Writing (7.4) for $i = 1, 2$ provides a system of two nonlinear equations; i.e.,

$$F := \begin{bmatrix} \|{}^a\mathbf{b}_1 - {}^a\mathbf{c}_1\|^2 - x_1^2 \\ \|{}^a\mathbf{b}_2 - {}^a\mathbf{c}_2\|^2 - x_2^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (7.5)$$

Since ${}^a\mathbf{b}_i$ is known in a direct kinematics problem, and ${}^a\mathbf{c}_i$ can be obtained from (7.3), F in (7.5) becomes a system of equations in two unknowns $[\psi \ \phi]^T$. Solving (7.5) iteratively for the unknowns ψ and ϕ provides the solution for the direct kinematics problem. A nonlinear least squares analysis can be employed to this end. The corresponding Jacobian matrix can be conveniently obtained from a computer algebra system (CAS).

7.6.2 Differential Kinematics

In order to facilitate the study of the differential kinematics, the angular velocity of the link $B_i C_i$ is denoted by $\boldsymbol{\omega}_i$ and the angular velocity of the moving body is denoted

by $\boldsymbol{\omega}_c$. In addition, the angular displacements about the axes $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_2$ are denoted by ψ and ϕ respectively. The linear velocity of point C_i is provided by,

$$\mathbf{v}_{ci} = \boldsymbol{\omega}_c \times \mathbf{c}_i. \quad (7.6)$$

In terms of the the velocity of the actuated joints (i.e., \dot{d}_i), \mathbf{v}_{ci} can also be calculated as,

$$\mathbf{v}_{ci} = \dot{d}_i \hat{\mathbf{n}}_i + \boldsymbol{\omega}_i \times \mathbf{x}_i. \quad (7.7)$$

Equating (7.6) and (7.7) provides,

$$\boldsymbol{\omega}_c \times \mathbf{c}_i = \dot{d}_i \hat{\mathbf{n}}_i + \boldsymbol{\omega}_i \times \mathbf{x}_i. \quad (7.8)$$

Dot multiplying both sides of (7.8) by \mathbf{x}_i and subsequent rearranging using the vector triple product rule yields,

$$(\mathbf{c}_i \times \mathbf{x}_i) \cdot \boldsymbol{\omega}_c = (\hat{\mathbf{n}}_i \cdot \mathbf{x}_i) \dot{d}_i. \quad (7.9)$$

Writing (7.9) for $i = 1, 2$ provides two scalar equations that can be arranged in matrix format as,

$$\begin{bmatrix} (\mathbf{c}_1 \times \mathbf{x}_1)^T \\ (\mathbf{c}_2 \times \mathbf{x}_2)^T \end{bmatrix} \times \boldsymbol{\omega}_c = \begin{bmatrix} (\hat{\mathbf{n}}_1 \cdot \mathbf{x}_1) & 0 \\ 0 & (\hat{\mathbf{n}}_2 \cdot \mathbf{x}_2) \end{bmatrix} \times \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \end{bmatrix} \quad (7.10)$$

The angular velocity vector $\boldsymbol{\omega}_c$ can be written as,

$$\begin{aligned} \boldsymbol{\omega}_c &= \dot{\psi} \hat{\mathbf{w}}_1 + \dot{\phi} \hat{\mathbf{w}}_2 \\ &= \begin{bmatrix} \hat{\mathbf{w}}_1 & \hat{\mathbf{w}}_2 \end{bmatrix} \times \begin{bmatrix} \dot{\psi} \\ \dot{\phi} \end{bmatrix}. \end{aligned} \quad (7.11)$$

Substituting (7.11) into (7.10) and subsequent rearranging provides,

$$\begin{aligned} \begin{bmatrix} (\mathbf{c}_1 \times \mathbf{x}_1) \cdot \hat{\mathbf{w}}_1 & (\mathbf{c}_1 \times \mathbf{x}_1) \cdot \hat{\mathbf{w}}_2 \\ (\mathbf{c}_2 \times \mathbf{x}_2) \cdot \hat{\mathbf{w}}_1 & (\mathbf{c}_2 \times \mathbf{x}_2) \cdot \hat{\mathbf{w}}_2 \end{bmatrix} \times \begin{bmatrix} \dot{\psi} \\ \dot{\phi} \end{bmatrix} \\ = \begin{bmatrix} (\hat{\mathbf{n}}_1 \cdot \mathbf{x}_1) & 0 \\ 0 & (\hat{\mathbf{n}}_2 \cdot \mathbf{x}_2) \end{bmatrix} \times \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \end{bmatrix}. \end{aligned} \quad (7.12)$$

Equation (7.12) can be rewritten as,

$$J_x \times \begin{bmatrix} \dot{\psi} & \dot{\phi} \end{bmatrix}^T = J_q \times \begin{bmatrix} \dot{d}_1 & \dot{d}_2 \end{bmatrix}^T. \quad (7.13)$$

The orientation manipulator is said to be in a *singular configuration* when at least one of the two matrices J_x and J_q is singular.

7.6.3 Inverse Kinematic Singularities

Since J_q is a diagonal matrix, it is singular only when at least one of the diagonal entries is zero; i.e., $\hat{\mathbf{n}}_i \cdot \mathbf{x}_i = 0$ or $\hat{\mathbf{n}}_i \perp \mathbf{x}_i$. Therefore, when the passive link B_iC_i is perpendicular with the axis of the prismatic joint A_iB_i , the mechanism is in an inverse kinematic singular configuration. When J_q is singular and its null space is not empty, there exist some non-zero \dot{d}_i for which $\begin{bmatrix} \dot{\psi} & \dot{\phi} \end{bmatrix}^T$ is zero; i.e., certain infinitesimal motion of the moving platform at the singular configuration (i.e., $A_iB_i \perp B_iC_i$) cannot be achieved despite the application of actuation forces.

7.6.4 Direct Kinematic Singularities

When J_x is singular, there exists some non-zero $\begin{bmatrix} \dot{\psi} & \dot{\phi} \end{bmatrix}^T$ that yields zero \dot{d}_i ; i.e., even though the actuators are fixed, the moving platform of the mechanism can exhibit

infinitesimal motion in some directions.

Case 1: When one of the rows of J_x vanishes, it becomes singular. This occurs when $(\mathbf{c}_i \times \mathbf{x}_i)$ lies in the same plane defined by $\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2$. Physically, when points B_i and C_i ($i = 1, 2$) coincides with the plane $\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2$ this direct kinematic singularity occur.

Case 2: The Jacobian matrix J_x becomes deficient in column rank when the four articulation points C_1, C_2, B_1 and B_2 are coplanar with either $\hat{\mathbf{w}}_1$ or $\hat{\mathbf{w}}_2$.

Case 3: Each row of the Jacobian matrix J_x defines a vector in a plane defined by the universal joint axes $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_2$. When the two row vectors coincide, J_x becomes singular. In geometric terms, when the planes defined by $(\mathbf{c}_1 \times \mathbf{x}_1)$, $(\mathbf{c}_2 \times \mathbf{x}_2)$ and $(\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2)$ intersect on a single line, the mechanism is at a direct kinematic singularity configuration.

7.6.5 Decoupling the Degrees of Freedom

Equation (7.12) provides the necessary condition for obtaining a configuration that fully decouples the degrees of freedom of the mechanism. If either the major or the minor diagonal of direct kinematic Jacobian J_x becomes zero, there is a one to one mapping of each actuated joint to a single degree of freedom. For the two diagonals of J_x , two such decoupled configuration exists, one of which occurs when $\mathbf{c}_i \times \mathbf{x}_i$ is coplanar with $\hat{\mathbf{w}}_j$ ($i \neq j$). Geometrically, the degrees of freedom are decoupled when the points C_i and B_i remains coplanar with $\hat{\mathbf{w}}_j$ ($i \neq j$) over the workspace of the mechanism. The other decoupled configuration occurs when $\mathbf{c}_i \times \mathbf{x}_i$ is coplanar with $\hat{\mathbf{w}}_i$. In geometric terms, this configuration is constrained by the co-planarity of the points C_i and B_i with the axis $\hat{\mathbf{w}}_i$ over the entire workspace. The condition for decoupling is fulfilled when each of the two actuated joint axes coincides with either of the two

universal joint axes. However, the spherical joints C_i must not be allowed to become collinear with the corresponding joint axis in order to avoid a singularity configuration. The aforementioned condition for producing decoupled motion is consistent with [164].

7.6.6 Dimensional Synthesis

The geometry of the mechanism was constrained to facilitate dimensional synthesis of the mechanism in the following manner. When the two identical actuators were at mid-stroke, the manipulator was defined to be at its home position. In addition, both actuators were constrained to operate in the vertical direction. At the home position, points A_1 and A_2 lie in a horizontal plane. The moving platform articulation points C_1 and C_2 also lie in a different horizontal plane at the home position. The plane defined by the points A_1 , B_1 and C_1 contain the revolute joint axis \hat{w}_2 at the home position. Correspondingly, the axis \hat{w}_1 is coplanar with the plane defined by the points A_2 , B_2 and C_2 at the home position. Finally, the distance of each point of the pairs (C_1, C_2) and (A_1, A_2) from the mechanism center O is constrained to be equal to that of their paired point. Under these constraints, the geometry of the manipulator can be defined by four parameters (r, R, h, H) . At the home position, r and R are the horizontal distances between the points O and C_i and B_i respectively. In addition, h and H are the vertical distances between the points O and C_i and B_i respectively. It should be noted that the aforementioned geometric configuration does not allow decoupled motion.

Since the coupling between the two kinematic loops of the manipulator was observed to be insignificantly small, the motion of each loop can be approximated as an independent spatial slider-crank mechanism of trivial complexity. In order to synthesize the dimensions of this manipulator, two such models must be optimized for kinematic

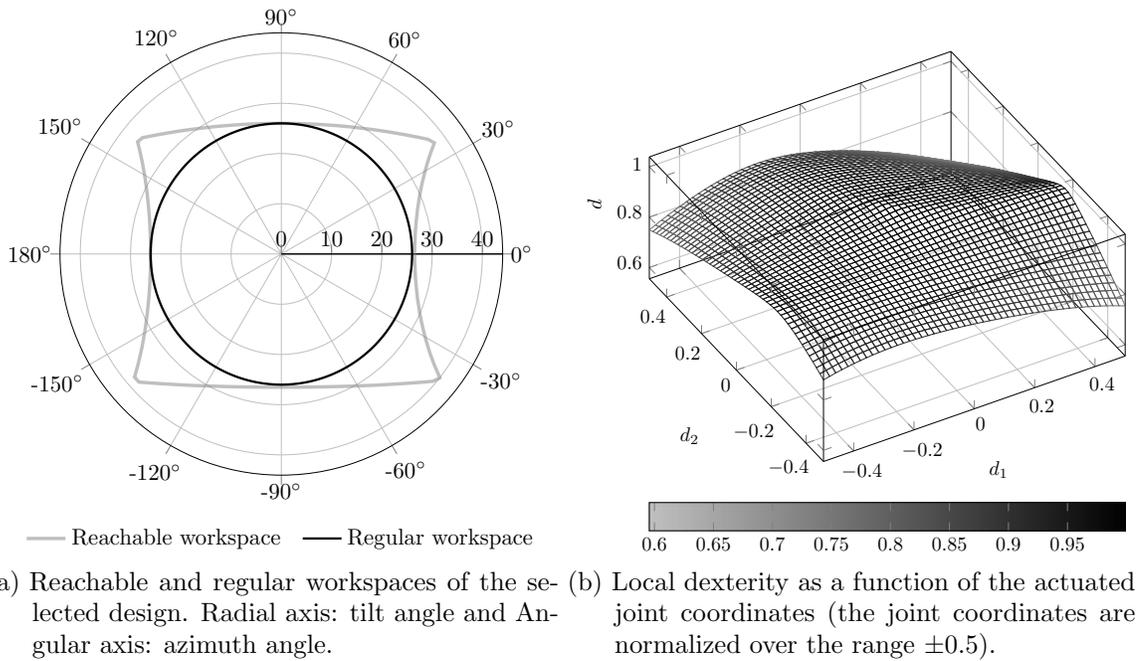


Figure 7.25: Kinematic performance of the torsion-restricted POM.

performance. However, application of RSM as an optimization method was considered to be unnecessary because interaction between the two models was *practically* nonexistent. As an alternative, the parameter space defined by the aforementioned geometric parameters was explored under a Latin hypercube sampling scheme, and the kinematic performances estimated at these sample points were employed to choose the preferred dimensions of the manipulator. In order to quantify the performance of the candidate designs, the GCI, the minimum dexterity d_m and the maximum tilt angle θ_T of the regular workspace served as performance metrics. Subsequently, the design that was chosen for its high kinematic performance is characterized by the following geometric parameters: $r = 1.1$, $R = 1.6$, $h = 0.2$, and $H = 4.0$. For the sake of computational convenience, the values of these parameters were normalized with respect to the stroke of the actuator. The kinematic performance metrics corresponding to the selected design are graphically presented in Fig. 7.25. Specifically, these metrics were estimated

as $GCI = 0.9183$, $d_m = 0.6592$, and $\theta_T = 26.0156^\circ$.

7.7 Conclusion

Virtual prototypes generally rely on a simplified representation (i.e., mathematical models) of the real world. In such cases, whether an engineering design fulfills its promise to deliver the very performance it was synthesized for must be confirmed by implementing it in the physical domain. Motivated by this argument, this chapter has constructed POM prototypes by amalgamating designs synthesized in the previous chapters. Each of these designs constitutes an important system component such as actuation systems, geometric dimensions, and motion control systems. Furthermore, the constructed POM prototypes were tested extensively to evaluate their dynamic capacities. Another contribution of this chapter is claimed as the formulation of an empirical kinematic calibration method that was developed to conveniently obtain superior positioning accuracy in a POM operation. Physical and numerical experiments were performed to validate the proposed calibration method. In addition, the knowledge gained from the design synthesis exercises of the 3-PSS/S manipulators was applied to formulate a design of a torsion-restricted POM.

Chapter 8

Conclusion

This thesis set out to explore the design synthesis methodologies for parallel orientation manipulation systems. Upon reviewing the related literature, it was decided that a ground-up design philosophy is better suited to actualize the specified goals as opposed to constructing a prototype by commissioning general purpose system components that may or may not function well together. Indeed, this design strategy enhances the chances of success by demanding a fierce focus on the design goals in all development activities. Admittedly, such an approach is not without its pitfalls, since it may happen that existing designs that are proven to be technically sound and practically robust might be overlooked in favor of inferior alternatives. Nonetheless, this thesis remained ever watchful not to reinvent the proverbial wheel by employing established design methodologies wherever they were applicable. At the same time, it purposefully utilized every opportunity to explore new ideas in hopes of extending the existing knowledge. This allowed the thesis to forge a philosophical identity that thrived to bridge the gap between academic curiosity and engineering practicality. Correspondingly, the novel designs presented in this thesis were not only evaluated in simulation environments, but also implemented in the physical domain. The extensive empirical evidence thus obtained demonstrated the effectiveness of the ground-up design philosophy.

8.1 Review of the Contributions

The contributions of this thesis, as claimed in Section 1.3, pertain to synthesizing and implementing the designs of a number of POM system components including actuators, kinematic dimensioning, and control systems. Each of these components is discussed in the following in order to qualitatively review the relevance of the contributions to the art of parallel robotics.

The design of the VCA presented in Chapter 3 addresses the need for a highly integrated and accurate actuation system for small-scale parallel manipulators. Experimental evaluation showed that the proposed design can generate motions of high acceleration. Correspondingly, the proposed VCA can be utilized as a turn-key solution for motion control applications requiring linear positioning with sub-micron accuracy. Although the stroke of the proposed actuator is relatively small, it must be acknowledged that the non-commutated mode of operation compromises stroke length in favor of high dynamics and design simplicity. However, the innovative design ideas such as the low-inertia construction achieved by a single PCB implementation and the integrated position sensing mechanism can be adopted for commutated linear motors with larger stroke lengths.

This thesis stipulates that attempting to determine the globally optimal dimensions of a parallel manipulator is generally impractical. Even if it is conceded, only for the sake of argument, that such an optimality exists, the corresponding dimensions might not satisfy all application-specific requirements. Therefore, dimensional synthesis of PKMs must be approached as an exercise in local optimization over a truncated parameter space that is conducive to all specified requirements. This is in stark contrast with exploring the entire solution space to find the *best* geometry that may or may not exist to encapsulate all the design specifications. In view of these

arguments, Chapter 4 employed response surface methodology (RSM) for synthesizing the dimensions of parallel orientation manipulators. Although the related literature reports to have employed a number of different optimization methods (e.g., genetic algorithm, parameter variation, controlled random search etc.), RSM was selected because it can develop an empirical model of the relationship between the objective function and its parameters from a small number of samples of the solution space. The computational efficiency and simplicity thus obtained is very attractive from an engineering application point of view.

Inspired by the success of modeling kinematic performance metrics in terms of the geometric parameters of parallel robots, RSM was employed by this thesis to develop empirical kinematic models of POMs. Although the compositions of the data required for a conventional and the proposed calibration methods are very similar (i.e., sample points of actuated joint coordinates and the corresponding pose of the moving platform), the advantages of the proposed method originate from its formulation of the problem. Since the empirical kinematic models of the proposed method take the form of a polynomial defined in either joint space or workspace coordinates, there is no need for deriving constraint equations from the kinematic structure of the robot. Furthermore, the empirical models can be defined in both *directions* to estimate either the inverse or the direct kinematic model. Therefore, whether a closed-form solution exists for either model is of no consequence to the proposed method. Experimental evaluation showed that its accuracy is comparable to the geometric models obtained by conventional calibration methods. Correspondingly, the aforementioned conveniences of the empirical kinematic models, especially for field applications, can be very appealing to the practitioners of parallel robotics.

The lack of robustness of the PID technology in terms of managing the time-varying inertial load and the nonlinear dynamics of a robotic manipulator has been

experimentally demonstrated by many past studies. Although advanced controllers can offer robust performance in such applications, the disparity between their complexities and the potential improvements in performance has deterred the industrial practitioners from favoring them over the simpler PID controllers. As an alternative, the ADRC technology was employed to bridge this gap by achieving superior performance at a computation cost comparable to a PID controller. The corresponding experimental evaluations performed in this thesis confirmed the robust control performance ADRC can provide for time varying and nonlinear MIMO systems such as parallel orientation manipulators. In addition, optimization of the classic ADRC algorithm, as proposed by this thesis, was shown to reduce the computational complexity further, which enables hardware with limited capacity to implement the ADRC technology. In this regard, the optimized ADRC algorithm can potentially have profound implications for embedded control systems.

When a motion control system is implemented on a remotely deployable hardware, the application-specific requirements of small size, low power consumption, high speed, large computational capacity, and robust reliability must be satisfied. Although an FPGA-based control hardware encapsulates all these characteristics, the unconventional nature of its design methodology (i.e., digital hardware design) poses a challenge. Correspondingly, this thesis formulated digital hardware designs that efficiently implemented a multi-loop ADRC controller on an FPGA chip. Since the architecture of an ADRC controller does not change for different plants, these designs can be employed to potentially formulate a universal controller for robotic manipulators. However, such a controller must be individually tuned for each application. In addition to the conveniences of this controller, the FPGA implementation is guarded against obsolescence because of its digital design.

As a result of utilizing software tools to formulate and subsequently to evaluate

designs in the virtual domain, modern engineering products are being developed more efficiently than it was ever possible. Nonetheless, it must be acknowledged that it is difficult for these software tools to exhaustively model all real world constraints. Therefore, practicality of a design is convincingly demonstrated when its physical prototype can be shown to have achieved all design goals. To this end, this thesis constructed POM prototypes as a means to definitively evaluate the synthesized designs. Corresponding experimental results showed that the prototypes achieved the goals of accurate and high speed orientation manipulation.

8.2 Limitations of the Reported Research

As is the case in any research, conducting it effectively requires a predefined threshold so that issues that fall outside can be deferred to future work. Although this threshold reflects the research goals, being too ambitious might result in complexities unmanageable by the available resources. On the other hand, a conservative approach may leave the goal of generating new knowledge unfulfilled. Therefore, it is necessary to set the threshold realistically in order to maximize the likelihood of success. Nonetheless, even a well-balanced threshold introduces limitations to the conducted research. Correspondingly, these limitations must be put into the context of the findings of the study.

RSM can be characterized as a collection of techniques that largely involve the design of the experiment and the analysis of the experimental data. This thesis employed it for performing dimensional synthesis and kinematic calibration of POMs. However, RSM does not give any direction regarding the parameterization of the problem being studied. For example, one of the responses in a dimensional synthesis exercise can be the volume of the workspace. Despite the obviousness of this kinematic

performance metric, it is not always clear how to parameterize this quantity in order to provide RSM the best chance of success. Correspondingly, it is up to the creativity of the designer to devise a parametrization appropriate for the polynomial approximation models of RSM. However, the existing literature can be extended by conducting further research to develop a comprehensive database of such parameterizations because the problem formulations are largely identical for PKMs with similar motion patterns.

The designs synthesized in this thesis were evaluated by physical experiments performed in laboratory environments. Since the POM prototypes are constructed for remote deployment, these experiments fail to ascertain how well they would function over the extended industrial temperature range. Nonetheless, these experiments provide proof of functionality of the formulated designs.

8.3 Recommendations for Future Research

Although instances of RSM being employed for dimensional synthesis of PKMs are scarce in the existing literature, this thesis demonstrated how effectively it can be utilized in this regard. However, whether its application can be extended to PKMs of higher complexity remains an open question. Correspondingly, it constitutes an interesting avenue for future research.

The notion of manipulation dexterity was employed in this thesis in order to evaluate geometric design alternatives. Although the existing literature provides numerical methods for estimating it, eminent researchers [58] have criticized them for being inconsistent and ill-equipped to numerically represent its qualitative definition. Correspondingly, alternative parameterizations were proposed in the literature (e.g., [58]). However, they have struggled to gain traction within the parallel robotics community because they lack the simplicity of the conventional estimation methods.

Since addressing these drawbacks is beyond the scope of this thesis, the topic was deferred to future work. Nonetheless, potentially impactful research can be conducted if the goal is to devise a robust and simple metric for measuring dexterity.

This thesis employed CMM measurements for the application of RSM models of kinematic calibration. Admittedly, the accuracy of CMM is hard to replicate by other measurement methods. As a result, it is unclear whether the expectedly larger error of the corresponding measurements would render the proposition of performing an empirical kinematic calibration unachievable. A possible future research can explore this question. In addition, it would be interesting to experimentally evaluate the feasibility of RSM calibration for PKMs of higher complexity.

8.4 Final Remarks

The dynamic capacities of the POM prototypes constructed in this thesis were demonstrated by their individual abilities to achieve angular velocities and angular accelerations exceeding 2000 degrees/s and 5×10^5 degrees/s², respectively. Furthermore, they were shown to provide angular positioning of a payload with an accuracy in the order of 1.0 mrad while maintaining globally robust control performance. Besides these tangible performance metrics, this thesis successfully utilized RSM for the tasks of dimensional synthesis and kinematic calibration of parallel orientation manipulators. In addition, a resource-optimized formulation of the ADRC controller was developed, which drastically reduced the hardware cost of implementation. In view of these achievements, it is claimed that this thesis has fulfilled its original goal of devising practical designs of parallel orientation manipulators that can deliver the speed and the accuracy requirements of a typical optomechatronic application.

Bibliography

- [1] H. Cho and M. Y. Kim, “Optomechatronic technology: the characteristics and perspectives,” *Industrial Electronics, IEEE Transactions on*, vol. 52, no. 4, pp. 932–943, 2005.
- [2] H. Cho, *Optomechatronics: Fusion of optical and mechatronic engineering*. CRC press, 2005.
- [3] M. Hafez, T. Sidler, R. Salathe, G. Jansen, and J. Compter, “Design, simulations and experimental investigations of a compact single mirror tip/tilt laser scanner,” *Mechatronics*, vol. 10, no. 7, pp. 741–760, 2000.
- [4] J. Sofka, V. Skormin, V. Nikulin, and D. Nicholson, “Omni-wrist iii-a new generation of pointing devices. part i. laser beam steering devices-mathematical modeling,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 42, no. 2, pp. 718–725, 2006.
- [5] E. Onillon, L. Lisowski, P. Spanoudakis, and E. Gilson, “Titp-tilt mechanism controlled by an hbrisc2 space grade processor,” in *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*. IEEE, 2005, pp. 1239–1244.

- [6] S. Woody and S. Smith, “Design and performance of a dual drive system for tip-tilt angular control of a 300mm diameter mirror,” *Mechatronics*, vol. 16, no. 7, pp. 389–397, 2006.
- [7] T. Shimizu, S. Nagata, S. Tsuneta, T. Tarbell, C. Edwards, R. Shine, C. Hoffmann, E. Thomas, S. Sour, R. Rehse *et al.*, “Image stabilization system for hinode (solar-b) solar optical telescope,” *Solar Physics*, vol. 249, no. 2, pp. 221–232, 2008.
- [8] T. Villgrattner and H. Ulbrich, “Optimization and dynamic simulation of a parallel three degree-of-freedom camera orientation system,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct. 2010, pp. 2829–2836.
- [9] —, “Design and control of a compact high-dynamic camera-orientation system,” *Mechatronics, IEEE/ASME Transactions on*, vol. 16, no. 2, pp. 221–231, 2011.
- [10] Y.-C. Lee, C.-C. Lan, C.-Y. Chu, C.-M. Lai, and Y.-J. Chen, “A pan-tilt orienting mechanism with parallel axes of flexural actuation,” *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 3, pp. 1100–1112, 2013.
- [11] C. M. Gosselin and J.-F. Hamel, “The agile eye: a high-performance three-degree-of-freedom camera-orienting device,” in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*. IEEE, 1994, pp. 781–786.
- [12] G. Dunlop and T. Jones, “Position analysis of a two dof parallel mechanismthe canterbury tracker,” *Mechanism and Machine Theory*, vol. 34, no. 4, pp. 599–614, 1999.

- [13] S. Hrabar, P. Corke, and V. Hilsenstein, "Ptz camera pose estimation by tracking a 3d target," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 240–247.
- [14] T. Yamashita, M. Morita, M. Shimizu, D. Eto, K. Shiratama, and S. Murata, "The new tracking control system for free-space optical communications," in *Space Optical Systems and Applications (ICSOS), 2011 International Conference on*. IEEE, 2011, pp. 122–131.
- [15] Z. Zhakypov, E. Golubovic, and A. Sabanovic, "Galvanometric optical laser beam steering system for microfactory application," in *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*. IEEE, 2013, pp. 4138–4143.
- [16] X. Kong and C. M. Gosselin, "Type synthesis of three-degree-of-freedom spherical parallel manipulators," *The International Journal of Robotics Research*, vol. 23, no. 3, pp. 237–245, 2004.
- [17] X. Kong and C. Gosselin, *Type Synthesis of Parallel Mechanisms*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2010.
- [18] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, 3rd ed. Wiley, 2009.
- [19] J. Han, "From PID to active disturbance rejection control," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 3, pp. 900–906, March 2009.

- [20] E. Monmasson and M. N. Cirstea, “Fpga design methodology for industrial control systemsa review,” *Industrial Electronics, IEEE Transactions on*, vol. 54, no. 4, pp. 1824–1842, 2007.
- [21] T. Rahman, N. Krouglicof, and L. Lye, “Kinematic synthesis of nonspherical orientation manipulators: Maximization of dexterous regular workspace by multiple response optimization,” *Journal of Mechanical Design*, vol. 134, no. 7, p. 071009, 2012. [Online]. Available: <http://link.aip.org/link/?JMD/134/071009/1>
- [22] T. Rahman, G. Rideout, and N. Krouglicof, “Evaluation of dynamic performance of nonspherical parallel orientation manipulators through bond graph multi-body simulation,” in *10th International Conference on Bond Graph Modeling and Simulation (ICBGM’2012)*, July 2012.
- [23] T. Rahman, D. Hicks, M. Hossain, and N. Krouglicof, “Digital hardware implementation of an active disturbance rejection controller for a highly dynamic parallel orientation manipulator,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 5750–5757.
- [24] L.-W. Tsai, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. John Wiley & Sons, Inc., 1999.
- [25] J. Osborne, G. Hicks, and R. Fuentes, “Global analysis of the double-gimbal mechanism,” *Control Systems, IEEE*, vol. 28, no. 4, pp. 44–64, Aug 2008.
- [26] W. Appleberry, “Gimbal mechanism,” US Patent 4 318 522, Mar 9, 1982.
- [27] J. Merlet, *Parallel Robots*, 2nd ed., ser. Solid mechanics and its applications. Dordrecht, The Netherlands: Springer, 2006.

- [28] J. Tlustý, J. C. Ziegert, and S. Ridgeway, “A comparison of stiffness characteristics of serial and parallel machine tools,” *Journal of Manufacturing Processes*, vol. 2, no. 1, pp. 67–76, 2000.
- [29] S. Briot, I. Bonev *et al.*, “Are parallel robots more accurate than serial robots?” *CSME Transactions*, vol. 31, no. 4, pp. 445–456, 2007.
- [30] J. Merlet, “Still a long way to go on the road for parallel mechanisms,” in *Proceedings of the ASME 27th Biennial Mechanisms and Robotics Conference*, Montreal, Quebec, 2002.
- [31] M. Karouia and J. M. Hervé, “Asymmetrical 3-dof spherical parallel mechanisms,” *European Journal of Mechanics-A/Solids*, vol. 24, no. 1, pp. 47–57, 2005.
- [32] R. Kurtz and V. Hayward, “Multiple-goal kinematic optimization of a parallel spherical mechanism with actuator redundancy,” *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 644–651, Oct 1992.
- [33] P. Vischer and R. Clavel, “Argos: a novel 3-dof parallel wrist mechanism,” *The International Journal of Robotics Research*, vol. 19, no. 1, pp. 5–11, 2000.
- [34] R. Di Gregorio, “A new parallel wrist using only revolute pairs: the 3-ruu wrist,” *Robotica*, vol. 19, no. 03, pp. 305–309, 2001.
- [35] Y. Fang and L.-W. Tsai, “Structure synthesis of a class of 3-dof rotational parallel manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 117–121, 2004.
- [36] L. Tsai, *Mechanism Design: Enumeration of Kinematic Structures According to Function*, ser. Mechanical and Aerospace Engineering Series. Taylor & Francis, 2000.

- [37] I. A. Bonev and C. M. Gosselin, “Singularity loci of spherical parallel mechanisms,” in *IEEE International Conference on Robotics and Automation*, vol. 3. IEEE; 1999, 2005, p. 2957.
- [38] J. Gallardo-Alvarado, M. García-Murillo, and L. Pérez-González, “Kinematics of the 3rrrs+ s parallel wrist: A parallel manipulator free of intersecting revolute axes,” *Mechanics Based Design of Structures and Machines*, vol. 41, no. 4, pp. 452–467, 2013.
- [39] X. Kong and C. M. Gosselin, “Type synthesis of 3-dof spherical parallel manipulators based on screw theory,” *Journal of Mechanical Design*, vol. 126, no. 1, pp. 101–108, 2004.
- [40] V. Hayward and R. Kurtz, “Preliminary study of serial-parallel redundant manipulator,” in *NASA Conference on Space Telerobotics*, Pasadena, January 31 1989, pp. 39–48.
- [41] D. Zlatanov, I. A. Bonev, and C. M. Gosselin, “Constraint singularities as c-space singularities,” in *Advances in Robot Kinematics*. Springer, 2002, pp. 183–192.
- [42] R. Di Gregorio, “Kinematics of the 3-rsr wrist,” *Robotics, IEEE Transactions on*, vol. 20, no. 4, pp. 750–753, 2004.
- [43] I. A. Bonev, D. Chablat, and P. Wenger, “Working and assembly modes of the agile eye,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2317–2322.

- [44] C. M. Gosselin and É. St-Pierre, “Development and experimentation of a fast 3-dof camera-orienting device,” *The International Journal of Robotics Research*, vol. 16, no. 5, pp. 619–630, 1997.
- [45] T. A. Hess-Coelho, “Topological synthesis of a parallel wrist mechanism,” *Journal of Mechanical Design*, vol. 128, no. 1, pp. 230–235, 2006.
- [46] J. Carretero, R. Podhorodeski, M. Nahon, and C. M. Gosselin, “Kinematic analysis and optimization of a new three degree-of-freedom spatial parallel manipulator,” *Journal of Mechanical Design*, vol. 122, no. 1, pp. 17–24, 2000.
- [47] K. Al-Widyan, X. Q. Ma, and J. Angeles, “The robust design of parallel spherical robots,” *Mechanism and Machine Theory*, vol. 46, no. 3, pp. 335–343, 2011.
- [48] Y. Lou, G. Liu, N. Chen, and Z. Li, “Optimal design of parallel manipulators for maximum effective regular workspace,” in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, Aug. 2005, pp. 795 – 800.
- [49] J. P. Merlet, “Determination of the orientation workspace of parallel manipulators,” *Journal of Intelligent & Robotic Systems*, vol. 13, pp. 143–160, 1995, 10.1007/BF01254849. [Online]. Available: <http://dx.doi.org/10.1007/BF01254849>
- [50] A. Rao, P. Rao, and S. Saha, “Dimensional design of hexaslides for optimal workspace and dexterity,” *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 444–449, June 2005.
- [51] K. Miller, “Maximization of workspace volume of 3-dof spatial parallel manipulators,” *Journal of Mechanical Design*, vol. 124, no. 2, pp. 347–350, 2002.

- [52] S. Bai, “Optimum design of spherical parallel manipulators for a prescribed workspace,” *Mechanism and Machine Theory*, vol. 45, no. 2, pp. 200–211, 2010.
- [53] H. Shin, S. Lee, W. In, J. I. Jeong, and J. Kim, “Kinematic optimization of a redundantly actuated parallel mechanism for maximizing stiffness and workspace using taguchi method,” *Journal of Computational and Nonlinear Dynamics*, vol. 6, no. 1, p. 011017, 2011. [Online]. Available: <http://link.aip.org/link/?CND/6/011017/1>
- [54] T. Li and S. Payandeh, “Design of spherical parallel mechanisms for application to laparoscopic surgery,” *Robotica*, vol. 20, no. 02, pp. 133–138, 2002. [Online]. Available: <http://dx.doi.org/10.1017/S0263574701003873>
- [55] M. Badescu and C. Mavroidis, “Workspace optimization of 3-legged upu and ups parallel platforms with joint constraints,” *Journal of Mechanical Design*, vol. 126, no. 2, pp. 291–300, 2004. [Online]. Available: <http://link.aip.org/link/?JMD/126/291/1>
- [56] J.-P. Merlet, “Optimal design for the micro parallel robot mips,” in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 2, 2002, pp. 1149–1154.
- [57] S.-D. Stan, V. Maties, and R. Balan, “Genetic algorithms multiobjective optimization of a 2 dof micro parallel robot,” in *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on*, June 2007, pp. 522–527.
- [58] J. P. Merlet, “Jacobian, manipulability, condition number, and accuracy of parallel robots,” *Journal of Mechanical Design*, vol. 128, no. 1, pp. 199–206, 2006. [Online]. Available: <http://link.aip.org/link/?JMD/128/199/1>

- [59] C. Gosselin, “Kinematic analysis, optimization and programming of parallel robotic manipulators,” Ph.D. dissertation, Université de Sherbrooke, 1985.
- [60] T. Yoshikawa, “Manipulability of robotic mechanisms,” *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.
- [61] J. Kotlarski, T. D. Thanh, B. Heimann, and T. Ortmaier, “Optimization strategies for additional actuators of kinematically redundant parallel kinematic machines,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, may 2010, pp. 656 –661.
- [62] J.-P. Merlet and D. Daney, “Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, april 2005, pp. 942 – 947.
- [63] N. Krouglicof, M. Morgan, N. Pansare, T. Rahman, and D. Hicks, “Development of a novel pcb-based voice coil actuator for opto-mechatronic applications,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013.
- [64] G. Box, “Signal-to-noise ratios, performance criteria, and transformations,” *Technometrics*, vol. 30, no. 1, pp. pp. 1–17, 1988. [Online]. Available: <http://www.jstor.org/stable/1270311>
- [65] Y. Hu and B. Li, “Robust design and analysis of 4pus1rpu parallel mechanism for a 5-degree-of-freedom hybrid kinematics machine,” *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 225, no. 5, pp. 685–698, 2011.

- [66] M. Carricato and V. Parenti-Castelli, "A novel fully decoupled two-degrees-of-freedom parallel wrist," *The International Journal of Robotics Research*, vol. 23, no. 6, pp. 661–667, 2004.
- [67] Z. Fan and L. Yan, "Type synthesis of a fully decoupled parallel wrist manipulator," in *Proceedings of the 2011 Third International Conference on Measuring Technology and Mechatronics Automation-Volume 03*. IEEE Computer Society, 2011, pp. 1068–1071.
- [68] Y. Li, J. Huang, and H. Tang, "A compliant parallel xy micromotion stage with complete kinematic decoupling," *Automation Science and Engineering, IEEE Transactions on*, vol. 9, no. 3, pp. 538–553, 2012.
- [69] S. Bashash and N. Jalili, "Robust adaptive control of coupled parallel piezoflexural nanopositioning stages," *Mechatronics, IEEE/ASME Transactions on*, vol. 14, no. 1, pp. 11–20, 2009.
- [70] Q. Li, W. Zhang, and L. Chen, "Design for control-a concurrent engineering approach for mechatronic systems design," *Mechatronics, IEEE/ASME Transactions on*, vol. 6, no. 2, pp. 161–169, 2001.
- [71] P. R. Ouyang, W.-J. Zhang, and F.-X. Wu, "Nonlinear pd control for trajectory tracking with consideration of the design for control methodology," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 4. IEEE, 2002, pp. 4126–4131.
- [72] T. Villgrattner, E. Schneider, P. Andersch, and H. Ulbrich, "Compact high dynamic 3 dof camera orientation system: Development and control," *Journal of System Design and Dynamics*, vol. 5, pp. 819–828, 2011.

- [73] H. Shin and J.-H. Moon, "Design of a double triangular parallel mechanism for precision positioning and large force generation," 2014.
- [74] Y. Su, D. Sun, L. Ren, and J. Mills, "Integration of saturated PI synchronous control and PD feedback for control of parallel manipulators," *Robotics, IEEE Transactions on*, vol. 22, no. 1, pp. 202 – 207, feb. 2006.
- [75] F. Paccot, N. Andreff, and P. Martinet, "A review on the dynamic control of parallel kinematic machines: Theory and experiments," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 395–416, 2009.
- [76] P.-L. Yen and C.-C. Lai, "Dynamic modeling and control of a 3-dof cartesian parallel manipulator," *Mechatronics*, vol. 19, no. 3, pp. 390–398, 2009.
- [77] H. Cheng, Y.-K. Yiu, and Z. Li, "Dynamics and control of redundantly actuated parallel manipulators," *Mechatronics, IEEE/ASME Transactions on*, vol. 8, no. 4, pp. 483–491, 2003.
- [78] M. Honegger, R. Brega, and G. Schweiter, "Application of a nonlinear adaptive controller to a 6 dof parallel manipulator," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 1930–1935.
- [79] G. S. Natal, A. Chemori, F. Pierrot, and O. Company, "Nonlinear dual mode adaptive control of PAR2: a 2-DOF planar parallel manipulator, with real-time experiments," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 2114–2119.

- [80] Z. Yang, J. Wu, and J. Mei, "Motor-mechanism dynamic model based neural network optimized computed torque control of a high speed parallel manipulator," *Mechatronics*, vol. 17, no. 7, pp. 381–390, 2007.
- [81] O. Linda and M. Manic, "Uncertainty-robust design of interval type-2 fuzzy logic controller for delta parallel robot," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 4, pp. 661–670, nov. 2011.
- [82] P. Begon, F. Pierrot, and P. Dauchez, "Fuzzy sliding mode control of a fast parallel robot," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 1. IEEE, 1995, pp. 1178–1183.
- [83] A. Vivas and P. Poignet, "Model based predictive control of a fully parallel robot," in *7th IFAC Symposium on Robot Control (SYROCO03)*, September 2003, pp. 277–282.
- [84] J. Dong, S. Salapaka, and P. Ferreira, "Robust MIMO control of a parallel kinematics nano-positioner for high resolution high bandwidth tracking and repetitive tasks," in *Decision and Control, 2007 46th IEEE Conference on*, dec. 2007, pp. 4495–4500.
- [85] P.-L. Yen, "A two-loop robust controller for compensation of the variant friction force in an over-constrained parallel kinematic machine," *International Journal of Machine Tools and Manufacture*, vol. 48, no. 12, pp. 1354–1365, 2008.
- [86] H. Guo, Y. Liu, G. Liu, and H. Li, "Cascade control of a hydraulically driven 6-DOF parallel robot manipulator based on a sliding mode," *Control Engineering Practice*, vol. 16, no. 9, pp. 1055–1068, 2008.

- [87] S. Flottmeier, S. Olma, and A. Trächtler, “Sliding mode and continuous estimation techniques for the realization of advanced control strategies for parallel kinematics,” in *World Congress*, vol. 19, no. 1, 2014, pp. 182–190.
- [88] K. S. Grewal, R. Dixon, and J. Pearson, “LQG controller design applied to a pneumatic Stewart-Gough platform,” *International Journal of Automation and Computing*, vol. 9, no. 1, pp. 45–53, 2012.
- [89] A. Radke and Z. Gao, “A survey of state and disturbance observers for practitioners,” in *American Control Conference, 2006*. IEEE, 2006, pp. 6–pp.
- [90] Q. Zheng, L. Dong, D. H. Lee, and Z. Gao, “Active disturbance rejection control for MEMS gyroscopes,” *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 6, pp. 1432–1438, Nov. 2009.
- [91] Z. Gao, “Active disturbance rejection control: a paradigm shift in feedback control system design,” in *American Control Conference*, June 2006, p. 7.
- [92] Z. Qing and G. Zhiqiang, “On practical applications of active disturbance rejection control,” in *Control Conference (CCC), 2010 29th Chinese*, July 2010, pp. 6095–6100.
- [93] Y. X. Su, B. Y. Duan, C. H. Zheng, Y. F. Zhang, G. D. Chen, and J. W. Mi, “Disturbance rejection high-precision motion control of a stewart platform,” *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 364–374, May 2004.
- [94] C. H. Tee, J. H. Lee, S. H. Kim, and Y. K. Kwak, “Development of beam rotating actuator based on voice coil motor type for multi-beam optical disc

- system,” in *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 3. IEEE, 1999, pp. 1884–1889.
- [95] P. Gandhi and S. Deshmukh, “A 2d optomechanical focused laser spot scanner: analysis and experimental results for microstereolithography,” *Journal of Micromechanics and Microengineering*, vol. 20, no. 1, p. 015035, 2010.
- [96] M.-G. Song, Y.-J. Hur, N.-C. Park, K.-S. Park, Y.-P. Park, S.-C. Lim, and J.-H. Park, “Design of a voice-coil actuator for optical image stabilization based on genetic algorithm,” *Magnetics, IEEE Transactions on*, vol. 45, no. 10, pp. 4558–4561, 2009.
- [97] N. Krouglicof, T. Rahman, and D. Hicks, “Parallel kinematic mechanism and bearings and actuators thereof,” U.S. Patent and Trademark Office Patent Application 61 700 080, September, 2012.
- [98] M. Remy, G. Lemarquand, B. Castagnede, and G. Guyader, “Ironless and leakage free voice-coil motor made of bonded magnets,” *Magnetics, IEEE Transactions on*, vol. 44, no. 11, pp. 4289–4292, 2008.
- [99] K. Halbach, “Design of permanent multipole magnets with oriented rare earth cobalt material,” *Nuclear Instruments and Methods*, vol. 169, no. 1, pp. 1–10, 1980.
- [100] M. B. Binnard, J.-M. Gery, and A. J. Hazelton, “High efficiency voice coil motor,” May 6 2008, US Patent 7,368,838.
- [101] J. Choi, H. Lee, S. Yoo, and M. D. Noh, “Analysis and modeling of a voice-coil linear vibration motor using the method of images,” *Magnetics, IEEE Transactions on*, vol. 48, no. 11, pp. 4164–4167, 2012.

- [102] M. Godkin, “Linear voice coil actuator with planar coils,” Sep. 7 2004, US Patent 6,787,943.
- [103] “One-dimensional PSD S3931, S3932, S3270,” White Paper, Hamamatsu Photonics K.K, Solid State Division, April 2011, www.hamamatsu.com.
- [104] B. K. Horn, “Tsai’s camera calibration method revisited,” 2004. [Online]. Available: http://people.csail.mit.edu/bkph/articles/Tsai_Revisited.pdf
- [105] J. Stuelpnagel, “On the Parametrization of the Three-Dimensional Rotation Group,” *SIAM Review*, vol. 6, no. 4, pp. 422–430, Oct 1964. [Online]. Available: <http://www.jstor.org/stable/2027966>
- [106] C. D. Crane and J. Duffy, *Kinematic Analysis of Robot Manipulators*, 1st ed. The Press Syndicate of The University of Cambridge, 1990, pp. 13–15.
- [107] N. Krouglicof, “Rigid-body pose measurement from a single perspective view,” in *Intelligent Autonomous Systems: Proceedings of the International Conference IAS-3*, F. C. A. Groen, S. Hirose, and C. E. Thorpe, Eds. Washington: IOS Press, 1993, pp. 368–377.
- [108] J. Schmidt and H. Niemann, “Using quaternions for parametrizing 3D rotations in unconstrained nonlinear optimization,” in *Vision, Modeling, and Visualization 2001*. AKA/IOS Press, 2001, pp. 399–406.
- [109] B. K. P. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *J. Opt. Soc. Am. A*, vol. 4, no. 4, pp. 629–642, 1987.
- [110] T. Rahman and N. Krouglicof, “An efficient camera calibration technique offering robustness and accuracy over a wide range of lens distortion,” *Image Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 626–637, 2012.

- [111] M. L. Husty, “An algorithm for solving the direct kinematics of general Stewart-Gough platforms,” *Mechanism and Machine Theory*, vol. 31, no. 4, pp. 365–379, 1996.
- [112] E. Ottaviano and M. Ceccarelli, “Optimal design of capaman (cassino parallel manipulator) with a specified orientation workspace,” *Robotica*, vol. 20, no. 02, pp. 159–166, 2002.
- [113] F. Xie, X.-J. Liu, and C. Wang, “Design of a novel 3-dof parallel kinematic mechanism: type synthesis and kinematic optimization,” *Robotica*, pp. 1–16, 2014, (accepted, to be published).
- [114] I. A. Bonev and C. M. Gosselin, “Analytical determination of the workspace of symmetrical spherical parallel mechanisms,” *Robotics, IEEE Transactions on*, vol. 22, no. 5, pp. 1011–1017, 2006.
- [115] I. A. Bonev, “Geometric analysis of parallel mechanisms,” Ph.D. dissertation, University of Laval, 2002.
- [116] Design-Expert version 8.0.6. Minneapolis, Minnesota: Stat-Ease, Inc., 2011. [Online]. Available: <http://www.statease.com/>
- [117] G. Corliss, “Which root does the bisection algorithm find?” *Siam Review*, vol. 19, no. 2, pp. 325–327, 1977.
- [118] R. Stamper, L.-W. Tsai, and G. Walsh, “Optimization of a three dof translational platform for well-conditioned workspace,” in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, apr 1997, pp. 3250–3255 vol.4.

- [119] M. D. McKay, R. J. Beckman, and W. J. Conover, “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, vol. 21, no. 2, pp. pp. 239–245, May, 1979. [Online]. Available: <http://www.jstor.org/stable/1268522>
- [120] G. Derringer and R. Suich, “Simultaneous optimization of several response variables,” *Journal of Quality Technology*, vol. 12, pp. 214–219, 1980.
- [121] X.-S. Wang, M.-L. Hao, and Y.-H. Cheng, “On the use of differential evolution for forward kinematics of parallel manipulators,” *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 760 – 769, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0096300308003548>
- [122] W. Khalil and S. Guegan, “Inverse and direct dynamic modeling of gough-stewart robots,” *Robotics, IEEE Transactions on*, vol. 20, no. 4, pp. 754–761, 2004.
- [123] K.-M. Lee and D. K. Shah, “Dynamic analysis of a three-degrees-of-freedom in-parallel actuated manipulator,” *Robotics and Automation, IEEE Journal of*, vol. 4, no. 3, pp. 361–367, 1988.
- [124] J. J. McPhee, “On the use of linear graph theory in multibody system dynamics,” *Nonlinear Dynamics*, vol. 9, no. 1, pp. 73–90, 1996.
- [125] D. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System Dynamics: Modeling and Simulation of Mechatronic Systems*, 4th ed. John Wiley and Sons, Inc., Hoboken, New Jersey, 2005.
- [126] W. Borutzky, *Bond Graph Methodology: Development and Analysis of Multidisciplinary Dynamic System Models*. Springer, 2010, ch. 8, p. 354.

- [127] D. Karnopp, “Understanding multibody dynamics using bond graph representations,” *Journal of the Franklin Institute*, vol. 334, no. 4, pp. 631 – 642, 1997.
- [128] R. Di Gregorio, “Dynamic model and performances of 2-dof manipulators,” *Robotica*, vol. 24, no. 01, pp. 51–60, 2006. [Online]. Available: <http://dx.doi.org/10.1017/S0263574705001839>
- [129] O. Ma and J. Angeles, “The concept of dynamic isotropy and its applications to inverse kinematics and trajectory planning,” in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*. IEEE, 1990, pp. 481–486.
- [130] Y. Zhao and F. Gao, “Dynamic performance comparison of the 8psr redundant parallel manipulator and its non-redundant counterpart the 6psr parallel manipulator,” *Mechanism and Machine Theory*, vol. 44, no. 5, pp. 991–1008, 2009.
- [131] J. Wu, J. Wang, T. Li, L. Wang, and L. Guan, “Dynamic dexterity of a planar 2-dof parallel manipulator in a hybrid machine tool,” *Robotica*, vol. 26, no. 01, pp. 93–98, 2008.
- [132] M. Li, T. Huang, J. Mei, X. Zhao, D. G. Chetwynd, and S. J. Hu, “Dynamic formulation and performance comparison of the 3-dof modules of two reconfigurable pm—the tricept and the trivariant,” *Journal of Mechanical Design*, vol. 127, no. 6, pp. 1129–1136, 2005.
- [133] H. Asada, “A geometrical representation of manipulator dynamics and its application to arm design,” *Journal of dynamic systems, measurement, and control*, vol. 105, no. 3, pp. 131–142, 1983.

- [134] T. Yoshikawa, “Dynamic manipulability of robot manipulators,” in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2. IEEE, 1985, pp. 1033–1038.
- [135] D. Karnopp and R. C. Rosenberg, *Analysis and Simulation of Multiport Systems – The Bond Graph Approach to Physical System Dynamics*. MIT Press, Cambridge, MA, 1968.
- [136] A. Zeid and C.-H. Chung, “Bond graph modeling of multibody systems: a library of three-dimensional joints,” *Journal of the Franklin Institute*, vol. 329, no. 4, pp. 605 – 636, 1992. [Online]. Available: [http://dx.doi.org/10.1016/0016-0032\(92\)90076-S](http://dx.doi.org/10.1016/0016-0032(92)90076-S)
- [137] W. Favre and S. Scavarda, “Bond graph representation of multibody systems with kinematic loops,” *Journal of the Franklin Institute*, vol. 335, no. 4, pp. 643 – 660, 1998. [Online]. Available: [http://dx.doi.org/10.1016/S0016-0032\(96\)00143-3](http://dx.doi.org/10.1016/S0016-0032(96)00143-3)
- [138] J. Wang, C. M. Gosselin, and L. Cheng, “Modeling and simulation of robotic systems with closed kinematic chains using the virtual spring approach,” *Multibody System Dynamics*, vol. 7, no. 2, pp. 145–170, 2002.
- [139] D. Karnopp, “Energetically consistent bond graph models in electromechanical energy conversion,” *Journal of the Franklin Institute*, vol. 327, no. 5, pp. 677 – 686, 1990.
- [140] Z. Gao, “Controllers, observers, and applications thereof,” US Patent US 8 060 340 B2, Nov, 2011.

- [141] S. Macfarlane and E. Croft, “Jerk-bounded manipulator trajectory planning: design for real-time applications,” *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 1, pp. 42 – 52, feb 2003.
- [142] R. L. Norton, *CAM Design and Manufacturing Handbook*. Industrial Press Inc., 2009.
- [143] Z. Gao, Y. Huang, and J. Han, “An alternative paradigm for control system design,” in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 5, 2001, pp. 4578 –4585 vol.5.
- [144] C. Xing, L. Donghai, G. Zhiqiang, and W. Chuanfeng, “Tuning method for second-order active disturbance rejection control,” in *Control Conference (CCC), 2011 30th Chinese*, July 2011, pp. 6322 –6327.
- [145] Z. Gao, “Scaling and bandwidth-parameterization based controller tuning,” in *American Control Conference, 2003. Proceedings of the 2003*, vol. 6, June 2003, pp. 4989 – 4996.
- [146] H. Abdellatif and B. Heimann, “Adapted time-optimal trajectory planning for parallel manipulators with full dynamic modelling,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 411–416.
- [147] M. Kim, S. H. Yoon, P. A. Domanski, and W. V. Payne, “Design of a steady-state detector for fault detection and diagnosis of a residential air conditioner,” *International Journal of Refrigeration*, vol. 31, no. 5, pp. 790 – 799, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0140700707002186>

- [148] “Addressing SWaP challenges in military platforms with 65-nm FPGAs and structured ASICs,” White Paper, Altera Corporation, October 2007.
- [149] S. Zhao, N. Usher, D. Morris, and J. Vincent, “Fixed-point implementation of active disturbance rejection control for superconducting radio frequency cavities,” in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 2693–2698.
- [150] A. Martini, F. Leonard, and G. Abba, “Robust nonlinear control and stability analysis of a 7DOF model-scale helicopter under vertical wind gust,” in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept. 2008, pp. 354–359.
- [151] Z. Ping and Z. Gao, “An FPGA-based digital control and communication module for space power management and distributed systems,” in *2005 American Control Conference*, 2005.
- [152] F. de Dinechin, J. Detrey, O. Cret, and R. Tudoran, “When fpgas are better at floating-point than microprocessors,” in *Proceedings of the 16th international ACM/SIGDA symposium on Field programmable gate arrays*, ser. FPGA ’08. New York, NY, USA: ACM, 2008, pp. 260–260. [Online]. Available: <http://doi.acm.org/10.1145/1344671.1344717>
- [153] R. Miklosovic, A. Radke, and Z. Gao, “Discrete implementation and generalization of the extended state observer,” in *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, June 2006.
- [154] S. T. Karris, *Introduction to Simulink with Engineering Applications*. Orchard Publication, 2011, ch. Appendix E, pp. E–18.

- [155] R. Ho and M. Horowitz, "Validation coverage analysis for complex digital designs," in *Computer-Aided Design, 1996. ICCAD-96. Digest of Technical Papers., 1996 IEEE/ACM International Conference on*, 1996, pp. 146–151.
- [156] S. Noguchi, H. Aramaki, and T. Kanada, "The development of pivot bearing with multiple degrees of freedom: 1st report: Prototype of pivot bearing with two degrees of freedom," *Precision Engineering*, vol. 30, no. 4, pp. 373 – 380, 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0141635906000067>
- [157] R. A. Kendall, "Ball joint with dynamic preload adjustment," Oct. 1 1991, uS Patent 5,052,844.
- [158] "deltaRocket - fast and accurate 3D printing," retrieved on March 20, 2015. [Online]. Available: <http://deltarocket.com/>
- [159] P. Renaud, N. Andreff, J.-M. Lavest, and M. Dhome, "Simplifying the kinematic calibration of parallel mechanisms using vision-based metrology," *Robotics, IEEE Transactions on*, vol. 22, no. 1, pp. 12–22, 2006.
- [160] R. K. Meyer and C. J. Nachtsheim, "The coordinate-exchange algorithm for constructing exact optimal experimental designs," *Technometrics*, vol. 37, no. 1, pp. 60–69, 1995.
- [161] B. D. Adelstein and M. J. Rosen, "Design and implementation of a force reflecting manipulandum for manual control research," in *Advances in Robotics*, H. Kazerooni, Ed., vol. 42. American Society of Mechanical Engineers, New York, 1992, pp. 1–12.

- [162] C. M. Gosselin and F. Caron, “Two degree-of-freedom spherical orientating device,” US Patent 5 966 991, October 19, 1998.
- [163] R. Di Gregorio, “Analytic determination of workspace and singularities in a parallel pointing system,” *Journal of Robotic Systems*, vol. 19, no. 1, pp. 37–43, 2002.
- [164] M. Carricato and V. Parenti-Castelli, “A novel fully decoupled two-degrees-of-freedom parallel wrist,” *The International Journal of Robotics Research*, vol. 23, no. 6, pp. 661–667, 2004.

Appendix A

Orthonormality of Numerically Estimated Rotation Matrices

The orthonormalization method presented here is an extension of the technique proposed by Horn in [104]. The rotation matrix \mathbf{R} is composed of three row vectors as in,

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{bmatrix}.$$

The orthonormality constraint dictates: (a) $\mathbf{r}_i \cdot \mathbf{r}_j = 0$ ($i \neq j$), (b) $\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = \|\mathbf{r}_3\| = 1$, and (c) $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$. A numerically estimated rotation matrix $\tilde{\mathbf{R}} = [\tilde{\mathbf{r}}_1 \ \tilde{\mathbf{r}}_2 \ \tilde{\mathbf{r}}_3]^T$ will not maintain the aforementioned properties strictly. This discrepancy can be mitigated by determining three mutually orthogonal unit vectors that replace the three estimated rows $\tilde{\mathbf{r}}_1$, $\tilde{\mathbf{r}}_2$, and $\tilde{\mathbf{r}}_3$. Each row vector in the orthonormal approximation of the matrix $\tilde{\mathbf{R}}$ must be as close as possible to the the corresponding vector \mathbf{r}_i . To this end, let k be a scalar that provides,

$$\mathbf{r}'_1 = \tilde{\mathbf{r}}_1 + k\tilde{\mathbf{r}}_2 \quad \text{and} \quad \mathbf{r}'_2 = \tilde{\mathbf{r}}_2 + k\tilde{\mathbf{r}}_1.$$

Dot multiplying the above two equations and subsequent simplification yields,

$$\mathbf{r}'_1 \cdot \mathbf{r}'_2 = \tilde{\mathbf{r}}_1 \cdot \tilde{\mathbf{r}}_2 + k(\tilde{\mathbf{r}}_1 \cdot \tilde{\mathbf{r}}_1 + \tilde{\mathbf{r}}_2 \cdot \tilde{\mathbf{r}}_2) + k^2(\tilde{\mathbf{r}}_1 \cdot \tilde{\mathbf{r}}_2) = 0.$$

Since $\tilde{\mathbf{R}}$ is very close to being orthonormal, the approximation that $\tilde{\mathbf{r}}_1 \cdot \tilde{\mathbf{r}}_1 \approx \tilde{\mathbf{r}}_2 \cdot \tilde{\mathbf{r}}_2 \approx 1$ is reasonable. In addition, the scalar k and the dot product $\tilde{\mathbf{r}}_1 \cdot \tilde{\mathbf{r}}_2$ are small because the row vectors $\tilde{\mathbf{r}}_1$ and $\tilde{\mathbf{r}}_2$ are nearly orthogonal. Therefore, $k^2(\tilde{\mathbf{r}}_1 \cdot \tilde{\mathbf{r}}_2) \approx 0$. These observations lead an approximate solution that can be formulated as,

$$k \approx -\frac{1}{2}\tilde{\mathbf{r}}_1 \cdot \tilde{\mathbf{r}}_2.$$

Subsequently, the row vectors \mathbf{r}'_1 and \mathbf{r}'_2 can be estimated. In a next step, they are normalized to obtain these two unit vectors,

$$\mathbf{p}'_1 = \frac{\mathbf{r}'_1}{\|\mathbf{r}'_1\|} \quad \text{and} \quad \mathbf{p}'_2 = \frac{\mathbf{r}'_2}{\|\mathbf{r}'_2\|}.$$

Finally, the orthonormalized approximation of the numerically estimated rotation matrix $\tilde{\mathbf{R}}$ is provided by,

$$\mathbf{R}' = \begin{bmatrix} \mathbf{p}'_1 \\ \mathbf{p}'_2 \\ \mathbf{p}'_1 \times \mathbf{p}'_2 \end{bmatrix}.$$