Analysis and Synthesis Methods for the Appropriate Design of Parallel Mechanisms

by

Joshua K. Pickard

Bachelor of Science in Engineering, UNB, 2012

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy in Engineering

In the Graduate Academic Unit of Mechanical Engineering

Supervisor(s):	Juan A. Carretero, PhD, Dept. of Mechanical Engineering
Examining Board:	Andrew Gerber, PhD, Dept. of Mechanical Engineering
	Ben Newling, PhD, Dept. of Physics
	Joseph D. Horton, PhD, Faculty of Computer Science
External Examiner:	David Daney, PhD, Inria Bordeaux - Sud-Ouest, France

This dissertation is submitted to the

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

April, 2018

©Joshua K. Pickard, 2018

Abstract

Uncertainties are an inherent element of the manufacturing and control of mechanisms. Generally, uncertainties are neglected in the analysis and synthesis of mechanisms, as they can be difficult to account for. However, the performance of real mechanisms depends on the uncertainties; thus, there is a need for reliable methods which are able to provide bounds on the true performance of mechanisms.

Parallel mechanisms are considered in this work, where the design of a parallel mechanism accounting for uncertainties is termed an appropriate design. Reliable methods, based on interval analysis, are developed to analyze and synthesize appropriately designed parallel mechanisms, fittingly termed appropriate analysis and appropriate synthesis respectively. These appropriate methods are applied to the four-bar linkage and the 3-$\underline{\text{R}}$RR planar parallel mechanism. Appropriate analysis methods compute the associated workspaces of mechanisms with an appropriate design. These workspaces consider the reachability, self-collisions, singularities, and wrench-capabilities of the mechanisms, and are used to guarantee whether the desired task is able

to be accomplished by the mechanism. Appropriate synthesis methods are able to completely explore a continuous design parameter space to determine the complete set of appropriate designs of a mechanism which accomplish the desired task. Any design selected from this set is guaranteed to accomplish the desired task. The development of appropriate analysis and appropriate synthesis introduces a powerful new tool to mechanisms designers.

Dedication

To my family and friends,

who have continually encouraged

me to pursue my interests.

Learning without thought is labor lost; thought without learning is perilous.

– Confucius (551–479 BC)

Preface

The work presented in this thesis is a result of years of study. Arriving at the end result has not been exactly straight-forward. Feedback from the reviewers of various publications, as well as my participation in several conferences early in my studies, had a significant influence on the direction of this research.

Upon beginning this thesis, the primary objective was to develop methods which are capable of analyzing the wrench generating capabilities of parallel mechanisms, known as the wrench capabilities, and use these methods to determine task-optimized designs.

While attending a talk at the *Canadian Committee for the Theory of Machines and Mechanisms 2013 Symposium* I was introduced to *interval analysis*. Interval analysis provides a mathematical framework for producing reliable results over continuous spaces.

The issue with my original objective was that a task-optimized design solution considers the design to be exact; thus, the results are not reliable for any deviations from this exact design. As a result of this, a new objective was

formed. The goal was to incorporate the interval analysis framework into the previous work to be able to obtain reliable solutions for the task-optimized designs. The known uncertainties can be added to the analysis of parallel mechanisms to determine if a design successfully accomplishes the desired task under the uncertainties.

During a five-month research internship at Inria Sophia Antipolis under the supervision of Dr. Jean-Pierre Merlet, it was determined that there was a need for the development of methods for the synthesis of mechanisms with uncertainties. The development of both analysis and synthesis methods for mechanisms with uncertainties provides a significant contribution. During the internship, an interval-based synthesis routine was developed for the four-bar linkage. Rather than simply obtaining a single design solution, we wanted to be able to determine the complete set of design solutions which are appropriate for the desired task. In order to develop such a method, the four-bar linkage and the task would need to be described in a form appropriate for interval analysis. As well, the set of routines required for iterating over the continuous design space and classifying regions of the design space as solutions or non-solutions would need to be developed.

A significant number of publications have arisen from this thesis. These are listed as follows:

1. J. K. Pickard and J. A. Carretero. Wrench workspaces of redundantly actuated spatial parallel manipulators using minimum wrench capabilities. In *Proceeding of the 2013 CCToMM Mechanisms, Machines, and*

Mechatronics Symposium, Montreal, Canada, May 30–31 2013a;

2. J. K. Pickard and J. A. Carretero. A comparison of wrench-feasible workspaces using interval analysis and convex linear mapping techniques. In *Proceedings of the Canadian Society for Mechanical Engineering International Congress 2014*, Toronto, ON, Canada, June 1–4 2014;

3. J. K. Pickard, J. A. Carretero, and V. C. Bhavsar. Task-optimized cable-actuated planar parallel manipulator architecture and its concurrent implementation. In *Proceedings of the International Conference on High Performance Computing and Simulation (HPCS 2014)*, Bologna, Italy, July 21–25 2014;

4. J. K. Pickard and J. A. Carretero. An interval analysis method for wrench workspace determination of parallel manipulator architectures. In *Proceeding of the 2015 CCToMM Mechanisms, Machines, and Mechatronics Symposium*, Ottawa, Canada, May 28–29 2015a;

5. J. K. Pickard and J. A. Carretero. An interval analysis method for wrench workspace determination of parallel manipulator architectures. *Transactions of the Canadian Society for Mechanical Engineering*, 40 (2):139–154, November 2015c;

6. J. K. Pickard and J. A. Carretero. Design optimisation of the 3-RRR planar parallel manipulator via wrench capability analysis. In *Proceed-*

ings of the 14th World Congress in Mechanism and Machine Science, Taipei, Taiwan, October 25–30 2015b;

7. J. K. Pickard, J. A. Carretero, and V. C. Bhavsar. On the convergence and origin bias of the teaching-learning-based-optimization algorithm. *Applied Soft Computing*, 46(1):115–127, September 2016a;

8. J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Accounting for tolerances in the design parameters of the 3-RRR. In *Proceeding of Advances in Robot Kinematics*, Grasse, France, June 27–30 2016b;

9. J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Towards the appropriate synthesis of the four-bar linkage. In *Proceedings of the 2017 CCToMM Mechanisms, Machines, and Mechatronics Symposium*, Montreal, Canada, May 25–26 2017a;

10. J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Analysis methods for the 3-RRR with uncertainties in the design parameters. In Jadran Lenarčič and Jean-Pierre Merlet, editors, *Advances in Robot Kinematics 2016*, pages 109–118. Springer International Publishing, 2017b;

11. J. K. Pickard and J. A. Carretero. Towards the appropriate synthesis of the four-bar linkage. *Transactions of the Canadian Society for Mechanical Engineering*, Accepted November, 2017 ;

12. J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Appropriate analysis of the four-bar linkage. *Mechanism and Machine Theory*, Submitted

2018a;

13. J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Appropriate synthesis of the four-bar linkage. *Mechanism and Machine Theory*, Submitted 2018b.

Acknowledgements

I would like to acknowledge my supervisor Juan A. Carretero for his guidance and friendship during my studies. I would also like to acknowledge Jean-Pierre Merlet, whom provided invaluable insights into interval analysis and allowed me the opportunity to collaborate with him at INRIA Sophia-Antipolis, France.

Table of Contents

Abstract ii

Dedication iv

Preface v

Acknowledgments x

Table of Contents xvii

List of Tables xviii

List of Figures xxv

Abbreviations xxvi

1 Introduction to Parallel Mechanisms 1

1.1 Kinematics 5

1.2 Statics of Parallel Mechanisms 8

1.3 Singularities 8

1.4 Analysis and Synthesis of Parallel Mechanisms 10
1.5 Problem Statement 12
1.6 Motivation 13
1.7 Objectives 14
1.8 Major Contributions of the Work 16
1.9 Outline of Thesis 17

2 Interval Analysis Methods **19**
2.1 Reliable Computations 19
2.2 Background 21
2.3 Interval Arithmetic 24
2.4 Interval Vectors and Matrices 27
2.5 Interval Nonlinear System of Equations 27
2.6 Lipschitz Interval Extension 30
2.7 Interval Analysis Solving Routine 32
2.7.1 Simplification 33
2.7.2 Existence 34
2.7.3 Bisection 35

3 Appropriate Design **38**
3.1 Background 40
3.2 Modelling Uncertainties 42
3.3 Kinematics of an Appropriate Design 44
3.3.1 Inverse Displacement Problem 45

3.3.2 Forward Displacement Problem 46
3.4 Interval Evaluation of the Jacobian Matrix 48
3.5 Singularities of an Appropriate Design 49
3.6 Appropriate Analysis of the 3-RRR 51
3.6.1 Inverse Kinematics 52
3.6.2 Forward Kinematics 58
3.6.3 Kinematic Singularities 61

4 Self-Collision Detection for Appropriate Designs 63
4.1 Background 64
4.2 Self-Collisions of an Appropriate Design 66
4.3 Partial Collision Test 68
4.4 Full Collision Test 71
4.5 Case Studies 72

5 Wrench Capability Analysis 77
5.1 Background 79
5.2 Interval Methods for the Wrench Capabilities of Appropriate Designs 82
5.2.1 System of Interval Linear Equations with Bounded Solutions 84
5.2.1.1 Vertex Matrices and Vertex Vectors 85
5.2.1.2 Strong Feasibility 86
5.2.1.3 Strong Solvability 87

5.2.2 Wrench Workspace – Inside Box Classification 88
5.2.3 Wrench Workspace – Outside Box Classification 88
5.2.4 Preconditioning 90
5.3 Wrench Workspaces of the 3-RPR Parallel Mechanism 91
5.3.1 Methods for Exact Designs 92
5.3.2 Interval Methods 93
5.4 Wrench Workspaces of the 3-RRR Parallel Mechanism 95
5.4.1 Methods for Exact Designs 96
5.4.2 Interval Methods 97

6 Appropriate Synthesis: The Four-Bar Linkage 103

6.1 The Four-bar Linkage Description 104
6.2 Kinematic Equations 109
6.3 Linkage Classifications 112
6.4 Branches and Circuits 114
6.5 Solving the Kinematics 119
6.6 Coupler Curve of an Appropriate Design 121
6.7 Desired Response: Description and Verification Methods . . . 134
6.7.1 Precision Points 135
6.7.2 Trajectories 135
6.7.3 Verification Test for Precision Points 139
6.7.4 Verification Test for Trajectories 142
6.8 Appropriate Synthesis 147

6.9 Case Studies 151
6.9.1 Obtaining Allowable Regions 151
6.9.2 Finding Appropriate Design Solutions 157
6.9.3 Constraint Violation 159
6.9.4 Reordering the Design List 160
6.9.5 Identifying Regions of Interest using Global Optimization 161

7 Appropriate Synthesis: The 3-RRR Planar Parallel Mechanism 166

7.1 A Prototype Mechanism and its Associated Constraints 167
7.2 Appropriate Design Solutions for Actuator Locations 168
7.3 Case Studies 171
7.3.1 Allowable Regions for the Placement of a Single Actuator 171
7.3.2 Allowable Regions for the Placement of All Actuators . 176
7.4 Identifying Design Solutions from High-Dimensional Design Spaces 180
7.5 Sampling the Appropriate Design Solutions 182

8 Conclusions 187

8.1 Appropriate Design 188
8.2 Collision Detection for Appropriate Designs 189
8.3 Wrench Capabilities 190
8.4 Appropriate Synthesis 191
8.5 Future Work 192

Bibliography **193**

A Screw Theory **215**

A.1 Twists 216

A.2 Wrenches 217

A.3 Reciprocity of Screws 218

A.4 Screw-based Jacobian 219

A.5 Statics 221

B Interval Analysis: Additional Routines **223**

B.1 Extended Interval Arithmetic 223

B.2 Regularity of Interval Matrices 224

B.3 Simplification Methods 225

B.3.1 2B Filtering 225

B.3.2 3B Filtering 226

B.4 Existence Methods 227

B.4.1 Interval Newton Method 227

B.4.2 Kantorovitch Method 228

B.4.3 Krawczyk Method 231

C Inverse Kinematics and Jacobians of the 3-<u>R</u>RR **234**

C.1 Inverse Kinematics Formulation 234

C.2 Jacobian Formulation 236

D Wrench Capabilities of Exact Designs **238**

D.1 Describing the Desired Wrench Capabilities 241

D.2 Discretization 244

E Jacobians of the 3-RPR Parallel Mechanism **245**

F Task-Optimized Parallel Mechanisms **247**

F.1 Global Optimization 248

F.2 Objectives of Task-Optimized Mechanisms 250

F.2.1 Determining if the Task Workspace is a Subset of the Wrench Workspace 252

F.2.2 Approximating the Task Workspace 253

F.2.3 Evaluating the Differences of Workspaces 254

F.3 Optimising the Planar Cable-Driven Parallel Mechanism . . . 255

F.3.1 Concurrent Implementation 260

F.4 Task-Optimized Design of the 3-RRR Planar Parallel Mechanism 267

F.4.1 Optimization Problem for a Grid Discretization 268

F.4.2 Task-Optimized Design Solutions 270

G The Canonical Differential Evolution Algorithm **273**

Vita **276**

List of Tables

1.1 Summary of exact and appropriate terminology 15

6.1 Classifications of four-bar linkages with an appropriate design. 113

6.2 Appropriate solutions for parameters v, p and q of the four-bar linkage given three precision points. 164

6.3 The corresponding values of x, y, ψ and θ of the four-bar linkage for solution 1 of Table 6.2. 164

6.4 Exact design solutions for all nine design parameters of the four-bar linkage given three precision points. 165

6.5 Appropriate design solutions for all nine design parameters given three precision points. 165

List of Figures

1.1 Parallel structure proposed by Pollard (Pollard, 1940) (from US Patent No. 2213108). 3

1.2 Gough platform (Gough and Whitehall, 1962). 4

1.3 Stewart platform (Stewart, 1965). 5

1.4 Delta robot (Clavel, 1990). 6

2.1 a) A linear transformation applied to an interval; b) a linear transformation applied to a bisected interval. 32

2.2 Flowchart of the interval analysis general solving routine. . . . 33

3.1 The 3-RRR architecture with 3-DOF (x, y, and ψ) (adapted from (Pickard and Carretero, 2015c)). 51

3.2 The reachable workspace of the 3-RRR planar parallel mechanism. 58

3.3 The two solutions for the forward kinematics of the 3-RRR planar parallel mechanism at $\alpha_1 = 5.0$ rad, $\alpha_2 = 6.0$ rad and $\alpha_3 = 0.5$ rad. 60

3.4 The singularity-free workspace of the 3-RRR. 62

4.1 Modelling self-collisions for the 3-RRR planar parallel mechanism. 68

4.2 Generating the set of all line segments corresponding to the limbs for the partial collision test. 70

4.3 Generating the vertex representation for set of all line segments corresponding to the limbs for the full collision test. . . 71

4.4 Constant orientation and collision-free workspaces of the 3-RRR planar parallel mechanism 75

4.5 Constant orientation and collision-free workspaces of the 3-RRR planar parallel mechanism – accounting for self-collisions with the proximal links and platform in layer 1 and the distal links in layer 2. 76

5.1 Kinematic diagram of the 3-RPR planar parallel mechanism with fixed actuator locations A_1, A_2, A_3. 92

5.2 Wrench workspace of the CDPM with 2-DOF and 3 cables. . . 94

5.3 Wrench workspaces of the CDPM with 2-DOF and 3 cables using interval methods. 96

5.4 Wrench workspace of the 3-RRR parallel mechanism with exact design described in equation (3.16) and orientation $\psi = 1.72908$. 97

5.5 Separated regions of the wrench workspace of the 3-RRR parallel mechanism obtained with the α-shape routine. 98
5.6 Wrench workspace, obtained with interval methods, of the 3-RRR parallel mechanism with exact design described in equation (3.16) and orientation $[\psi] = [1.72908]$. 99
5.7 Wrench workspace of the 3-RRR parallel mechanism with appropriate design described in equation (4.12) and orientation $[\psi] = [0.0]$. 100
5.8 Wrench workspaces for the 3-RRR parallel mechanism. 102

6.1 Four-bar linkage description. 105
6.2 The effect of geometrical uncertainties in the four-bar linkage. 108
6.3 The coupler curve corresponding to an appropriate design using: a) the Kantorovitch existence method; b) Newton's method to determine two non-intersecting solutions. 123
6.4 A detailed view showing the exact coupler curves corresponding to the lower bound, midpoint, and upper bound of the design parameters from the appropriate design. 125
6.5 The coupler curves corresponding to appropriate designs with different classifications. 131
6.5 Continued. The coupler curves corresponding to appropriate designs with different classifications. 132

6.6 The coupler curves corresponding to appropriate designs with multiple classifications (folding linkages). 133
6.7 The desired trajectory corresponding to $y = x^2$. 138
6.8 A plot of the desired task elements described in equation (6.36). 152
6.9 Appropriate design solutions for p and q for precision points response elements described in equation (6.36). 154
6.10 a) Zoomed-in view of one of the disconnected allowed regions from Figure 6.9; b) the coupler curve corresponding to a solution from the allowed region. 155
6.11 A subset of the design solutions for p and q for the first trajectory response element described in equation (6.36). 155
6.12 Design solutions for p and q for all trajectory response elements described in equation (6.36). 156
6.13 a-b) Coupler curves corresponding to a design from the allowed region. 156
6.14 Design solutions for p and q for all response elements described in equation (6.36). 158
6.15 The coupler curve and desired response elements corresponding to the first appropriate design solution from Table 6.5. . . 163

7.1 Prototype design of the 3-$\underline{\text{R}}$RR with relocatable actuators. . . 168
7.2 The desired trajectory selected for the 3-$\underline{\text{R}}$RR. 172

7.3 Search space for $[\mathbf{a}_3]$ considering trajectory separation for $\delta_{trajectory} = 0.025$ m. 173
7.4 Search space for $[\mathbf{a}_3]$ considering trajectory and actuator separations for $\delta_{trajectory} = 0.025$m and $\delta_{actuator} = 0.05$m. 173
7.5 Design solutions for $[\mathbf{a}_3]$ which yield a reachable trajectory ensuring no self-collisions. 175
7.5 Continued. Design solutions for $[\mathbf{a}_3]$ which yield a reachable trajectory ensuring no self-collisions. 176
7.6 Design solutions for $[\mathbf{a}_3]$ which yield a wrench-capable trajectory. 177
7.7 Projected actuator location solutions. 179
7.7 Continued. Projected actuator location solutions. 180
7.8 Projected actuator location solutions after selecting a desired location $\mathbf{a}_3 = (-0.402, 0.2)$. 182
7.9 Projected actuator location solutions after selecting a desired locations $\mathbf{a}_3 = (-0.402, 0.2)$ and $\mathbf{a}_2 = (-0.002, 0.35)$. 183
7.10 The projected actuator location solutions after selecting a desired locations $\mathbf{a}_3 = (-0.3005, -0.4)$, then $\mathbf{a}_2 = (-0.0005, 0.26)$. 184
7.11 Collision-free wrench workspace corresponding to the design described in equation (7.6). 185
7.12 The collision-free wrench workspace corresponding to the design described in equation (7.7). 186

B.1 Visualization of 2B filtering applied to $f([u]) = [u]^2 - 2[u] + 1 = 0$ with $[u] = [-10, 10]$. 226

B.2 Visualization of 3B filtering applied to $f([u]) = [u]^2 - 2[u] + 1 = 0$ with $[u] = [0.5, 4.358899]$ with 3B width of $\delta_{3B} = 0.1$. 227

B.3 Visualization of one iteration of Interval Newton applied to $f([u]) = [u]^2 - 2[u] + 1 = 0$ with $[u] = [0.6, 4.258899]$. Extended interval analysis is applied. 230

D.1 Actuator force workspace $[\boldsymbol{\tau}]$ (left) formed from the torque limits of three actuators (τ_1, τ_2, τ_3) mapped to form the wrench capabilities $\mathcal{F}$ (right) using equation (D.3). 240

F.1 An axis-aligned approximation of the task workspace (approximated task workspace) which is guaranteed to be a subset of the wrench workspace. 254

F.2 Design variables of the planar CDPM. 256

F.3 The objective function value for each grid coordinate in the feasible search-space (top) and the area of the approximated task workspace for each grid coordinate in the infeasible search-space (bottom). 257

F.4 Task-optimized workspaces of the planar CDPM. 261

F.5 Convergence results for test 4 of the best and worst individuals in the population. The vertical dashed lines signify a change from infeasible to feasible. 262

F.6 Computational time and speed-up for test 4 for various numbers of processing cores. 267
F.7 Task-optimized 3-RRR results. 272

List of Symbols, Nomenclature and Abbreviations

Symbol/Name	Description
$\mathcal{D}$	set of exact design parameters
$[\mathcal{D}]$	set of appropriate design parameters
$\mathcal{L}$	list of elements
$[\mathbf{k}]$	vector of known appropriate design parameters
$[\mathbf{u}]$	vector of unknown appropriate design parameters
$\cap$	intersection
$\cup$	union
$\square$	interval hull
width	interval width
mid	interval midpoint
det	matrix determinant
diag	diagonal matrix
conv	convex hull

dist	minimum distance
α-shape	α-shape
grid	grid discretization
count	count the number of points
volume	generalized volume
simplification	interval simplification
existence	interval existence
bisection	interval bisection
$\mathbb{P}$	set of end-effector poses
$\mathbb{P}_{RW}$	reachable workspace
$\mathbb{P}_{\widetilde{RW}}$	approximation of reachable workspace
$\mathbb{P}_{SFW}$	singularity-free workspace
$\mathbb{P}_{WW}$	wrench workspace
$\mathbb{P}_{\widetilde{WW}}$	approximation of wrench workspace
$\mathbb{P}_{TW}$	task workspace
$\mathbb{P}_{\widetilde{TW}}$	approximation of task workspace
$\mathcal{P}$	discretized set of end-effector poses
$\mathcal{P}_{RW}$	discretized reachable workspace
$\mathcal{P}_{WW}$	discretized wrench workspace
$\mathcal{P}_{TW}$	discretized task workspace
m	dimension of the joint space
n	dimension of the task space
n_{limbs}	number of limbs

$\mathbf{q}$	exact joint variables
$[\mathbf{q}]$	appropriate joint variables
$\mathbb{Q}$	set of joint variables
$\mathbf{p}$	exact end-effector pose
$[\mathbf{p}]$	appropriate end-effector pose
$\mathbb{P}$	set of end-effector poses
$\mathbf{C}(\mathbf{q}, \mathbf{p}, \mathcal{D})$	exact system of kinematic constraints
$\mathbf{C}([\mathbf{q}], [\mathbf{p}], [\mathcal{D}])$	appropriate system of kinematic constraints
$\$,\ \hat{\$},\ \hat{\$}_r$	screw, unit screw, and unit reciprocal screw
$\mathbf{V},\ t$	twist and twist intensity
$\mathbf{F},\ w$	wrench and wrench intensity
$\mathbf{v},\ \boldsymbol{\omega}$	linear and angular velocity
$\dot{\mathbf{x}}$	end-effector velocities
$\dot{\mathbf{q}}$	joint velocities
$\mathbf{f}, \mathbf{m}$	force and moment
$\mathbf{J},\ \mathbf{J}_{\mathbf{x}},\ \mathbf{J}_{\mathbf{q}}$	Jacobian, direct Jacobian, and inverse Jacobian
$\mathbf{H}$	Hessian
$\mathcal{V}$	vertex representation
$\mathcal{H}$	hyperplane representation
$\boldsymbol{\tau}$	actuator forces/torques
$[\boldsymbol{\tau}]$	actuator force/torque workspace
$\mathcal{T}_{\mathcal{V}}$	actuator force/torque workspace (vertex representation)
$\mathcal{F}$	wrench capabilities (actual)

$\mathcal{F}_{\mathcal{V}}$	wrench capabilities (vertex representation)
$\mathcal{F}_{desired}$	wrench capabilities (desired)

Chapter 1

Introduction to Parallel Mechanisms

Serial mechanisms (*e.g.*, robotic arms) have a single open-loop kinematic chain, and therefore require all joints to be active to fully constrain the mechanism. Parallel mechanisms, on the other hand, are a type of mechanism consisting of a moving platform which is connected to a fixed base through multiple serial kinematic chains (limbs) forming a closed-loop architecture. This allows for some of the joints in the mechanism to be passive. Serial mechanisms typically require the heavy actuators to be mounted at or near the active joints, thus an actuator must accommodate the weight of the other actuators along the serial chain. Parallel architectures are typically constrained using passive joints near to the moving platform; thus, the active joints containing the heavy actuators can be located near the fixed platform,

potentially allowing the achievement of greater accelerations, high structural stiffness and high accuracy, as well as larger payload to weight ratios compared to their serial counterparts. The main drawbacks are that they tend to have relatively smaller workspaces (except for cable-driven parallel mechanisms (CDPMs)) and are more affected by internal singular configurations. A parallel mechanism must have a certain number of degrees-of-freedom (DOF), which are associated with the number of independent parameters required to exactly define the position and orientation (together referred to as pose) of the end-effector. The closed-loop architecture of parallel mechanisms allows n-DOF motions using m active joints, where m must be equal to or greater than n. All remaining joints are passive. The naming convention used for symmetric parallel mechanism architectures is:

$$l - \underline{J}JJ$$

where l is the number of limbs (n_{limbs}) and each J denotes the joint types consecutively outward from the fixed base[1]. An underlined joint indicates that the joint is actuated. For example, the 3-$\underline{\text{R}}$RR parallel mechanism is a planar 3-DOF mechanism which consists of three limbs, each with an active revolute joint ($\underline{\text{R}}$) at the base and two additional passive revolute joints (R) consecutively along the limb.

In the parallel kinematics community, Pollard's parallel robot (Pollard, 1940)

[1]For this example three joints are considered, although more are possible

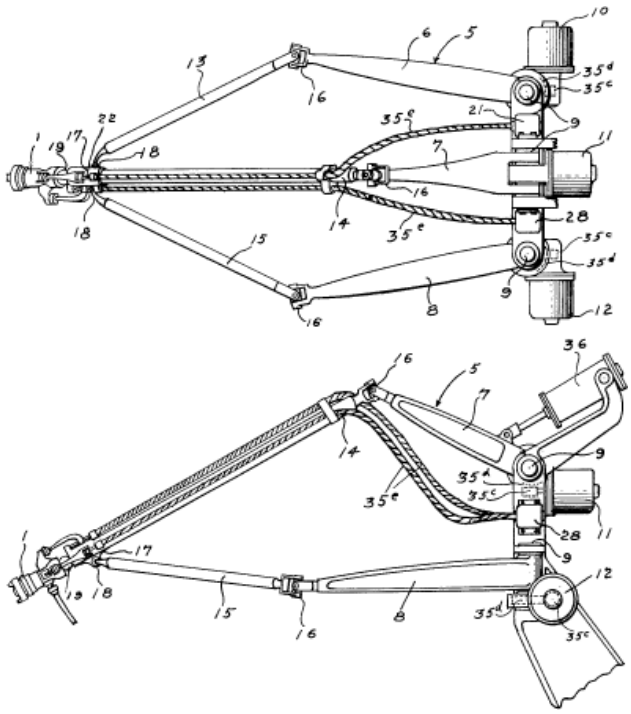


Figure 1.1: Parallel structure proposed by Pollard (Pollard, 1940) (from US Patent No. 2213108).

(see Figure 1.1), which is a 5 degrees-of-freedom (DOF) three-limbed parallel robot, is well known as the first industrial parallel robot design.

In 1947, Gough established the basic principles of a mechanism with a closed-loop architecture for the positioning and orientation of a moving platform so as to test tire wear (Gough, 1956, Gough and Whitehall, 1962) (see Figure 1.2). Stewart described the use of a parallel mechanism with 6-DOF as a flight simulator (Stewart, 1965) (see Figure 1.3). The architecture used in both instances is now generally termed the Gough-Stewart Platform, which has had significant influence in the academic world.

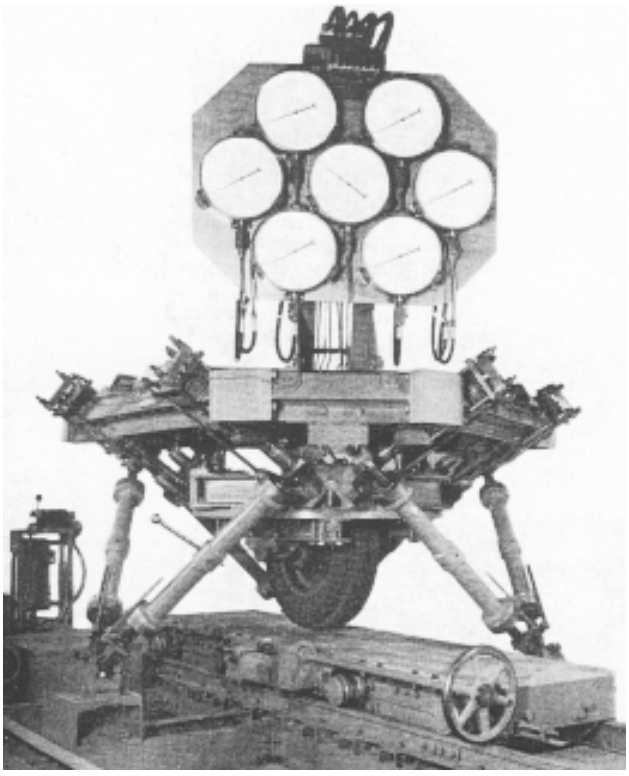


Figure 1.2: Gough platform (Gough and Whitehall, 1962).

Since their introduction, parallel mechanisms have been designed for a multitude of tasks. One of the most commercially successful parallel mechanism architectures is Reymond Clavel's 1985 invention, for which he obtained a patent (Clavel, 1990), known as the Delta robot. The Delta robot was one of the worlds first high-speed parallel robots thanks to its lightweight structure. The basic idea behind the Delta robot design is the use of parallelograms which allow the output link to remain at a fixed orientation with respect to the input link. These characteristics made the Delta robot a perfect tool for the industrial packaging industry for use in pick-and-place operations.

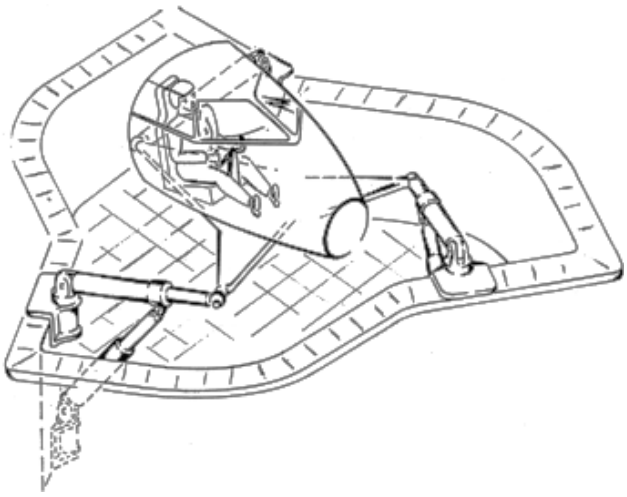


Figure 1.3: Stewart platform (Stewart, 1965).

Today, parallel mechanisms are also used as: high precision machine tools, designed by companies such as Giddings & Lewis, Ingersoll, Hexel, Geodetic and Toyoda (Patel and George, 2012); cable-driven mechanisms such as cable cameras (*e.g.*, Skycam (Brown, 1984)) and assistive devices (*e.g.*, Marionet-Assist (Merlet, 2010)); space applications such as the Canterbury satellite tracker (Jones and Dunlop, 2003); and reconfigurable robots such as that developed at the *National Research Council of Canada* (Xi et al., 2001). A good overview of the completed and remaining work on parallel mechanisms is given in Merlet et al. (2016).

1.1 Kinematics

Kinematics concerns the motion of points, bodies, and systems of bodies without considerations for the masses of each or the forces that produced

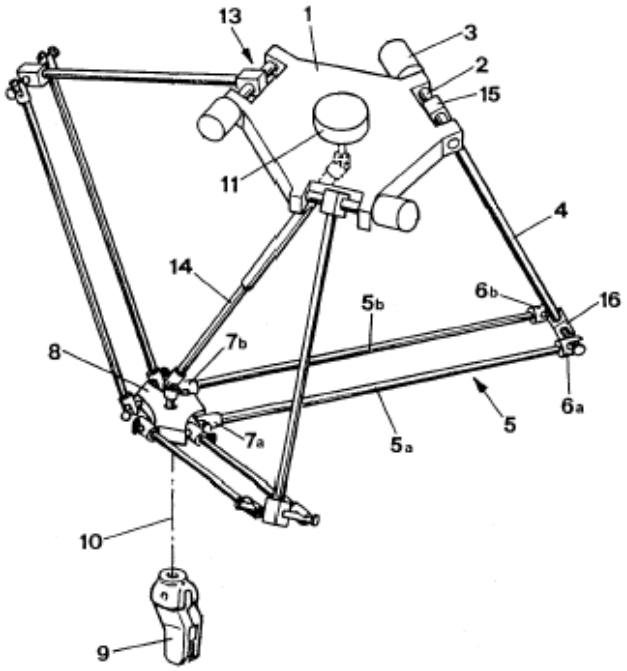


Figure 1.4: Delta robot (Clavel, 1990).

that motion. The study of kinematics is separated into two fundamental problems. These are the *inverse kinematics* and the *forward kinematics* which convert between the joint space and task or end-effector space. In this work, positions and displacements, commonly referred to as the inverse and forward displacement problems, are of primary interest. The inverse displacement problem converts from end-effector space to joint space, whereas the forward displacement problem converts from joint space to end-effector space. Generally, for parallel mechanisms, the forward displacement problem is a much more complex problem (see (Merlet, 2004)) than the inverse displacement problem. Throughout the remainder of this thesis, the terms

displacement and kinematics will be used interchangeably.

The inverse and forward displacement problems convert between the active joint variables $\mathbf{q}$ and end-effector poses $\mathbf{p}$. The inverse kinematics concern finding the set of active joint variables which correspond to the pose $\mathbf{p}$. The resulting set of active joint variables will be denoted by $\mathbb{Q}$. The forward kinematics concern finding the set of end-effector poses which correspond to the joint variables $\mathbf{q}$. The resulting set of end-effector poses will be denoted by $\mathbb{P}$.

To solve the kinematics, a mathematical description of the motions of the mechanism must be formulated. These motions may be described by a system of constraints, described in terms of the joint variables, end-effector pose, and design parameters of the mechanism. Let $\mathcal{D}$ represent the set of design parameters of a parallel mechanism. This system of constraints is denoted by $\mathbf{C}(\mathbf{q}, \mathbf{p}, \mathcal{D})$. Generally, it is desirable to formulate a closed-form description for the inverse and forward kinematics problems to avoid the use of numerical methods. This is not possible for all mechanisms, as the kinematics of some mechanisms are unable to be described in a closed-form.

Screw theory is a useful framework for describing motions of a rigid body. It is commonly applied in parallel mechanism research to formulate descriptions of the motions and statics of parallel mechanisms. For kinematics researchers, screw theory is an invaluable tool. An overview of screw theory applied to parallel mechanisms is provided in Appendix A.

1.2 Statics of Parallel Mechanisms

The study of the static capabilities of parallel mechanisms is termed, *wrench capability analysis*. Wrench capability analysis is essential for the design and performance evaluation of parallel mechanisms. At a given pose, the end-effector may be required to sustain an external wrench. The investigation into the joint torques that produce such conditions are necessary to ensure that the a mechanism is able to accomplish a desired task.

The forward force problem may be derived as

$$\mathbf{F} = \mathbf{J}^T \boldsymbol{\tau} \tag{1.1}$$

where $\mathbf{J} = {\mathbf{J_q}}^{-1}\mathbf{J_x}$, such that $\mathbf{J_x}$ is the $m \times n$ direct Jacobian matrix, $\mathbf{J_q}$ is the $m \times m$ diagonal matrix, $\boldsymbol{\tau}$ is the vector of m joint torques, and $\mathbf{F}$ is a wrench of dimension n.

The derivation of equation (1.1) is provided in Appendix A. The forward force problem provides an important relationship which describes a linear transformation between the joint torque and the end-effector wrench.

1.3 Singularities

Singularities are a major issue for parallel mechanisms. In these configurations the mechanism cannot be fully controlled, and there may be infinite forces/torques in its joints, possibly leading to a breakdown (Merlet, 2007b).

Such a configuration must usually be avoided. There are some scenarios in which singularities may be exploited. For example, the DexTAR robot (Campos et al., 2010) can increase the workspace of the robot by crossing certain singularities to change the working mode.

For non-redundant (*i.e.*, square) systems, poses in which the determinant of a Jacobian is close to zero are problematic. A *singular pose* is defined as a pose in which a Jacobian matrix becomes rank deficient or singular. There are three types of singularities for parallel mechanisms: *direct kinematic singularities*, *inverse kinematic singularities*, *and combined singularities* (Gosselin and Angeles, 1990).

Direct kinematic singularities occur when $\mathbf{J_x}$ is singular, namely when

$$det(\mathbf{J_x}) = 0 \tag{1.2}$$

This type of singularity causes the mechanism to gain one or more DOF, which means that the moving platform can possess infinitesimal motion in some direction when all of the actuators are locked. The mechanism is therefore unable to sustain certain wrenches in this pose. The resultant force of a parallel mechanism is a function of the forces in each limb generated by the actuators. If the resultant forces do not span the system of forces to be applied or sustained by the mechanism and the structure is unable to sustain the external forces, then the mechanism is termed degenerate and is force-unconstrained. These singular configurations are often referred to as

force-unconstrained poses or force-degenerate configurations (Firmani and Podhorodeski, 2004).

Inverse kinematic singularities occur when $\mathbf{J_q}$ is singular, namely when

$$det(\mathbf{J_q}) = 0 \tag{1.3}$$

These singularities cause the mechanism to lose one or more DOF, resulting in velocity degeneracy. Inverse kinematic singularities typically occur at the workspace boundary due to one or more limbs being fully extended or retracted. In such cases, the mechanism can theoretically sustain infinite loads in certain directions because these loads act through the structure of the mechanism.

Combined singularities occur when both $\mathbf{J_x}$ and $\mathbf{J_q}$ are singular. Generally, this type of singularity only occurs for mechanisms with special kinematic architectures (Tsai, 1999).

1.4 Analysis and Synthesis of Parallel Mechanisms

The analysis of a mechanism is a *detailed examination of its elements, structure and performance*. Analysis serves as a basis for discussion and interpretation. For parallel mechanisms, analysis may concern the determination of relevant workspaces, the force and moment generating capabilities

at the end-effector, singularities, and self-collisions and external collisions, to name a few. Synthesis, on the other hand, is the *determination of the constituent elements of a mechanism in order to satisfy a goal or desired task*. Traditionally, the development of parallel mechanisms to satisfy a desired goal relies on the knowledge, experience and ingenuity of the designer (*e.g.*, Gough-Stewart platform, the Delta robot, the Agile eye (Gosselin and Hamel, 1994)). For linkages, design aids such as the four-bar linkage coupler curve atlas (see (Hrones and Nelson, 1951, Mullineux, 2011)) are useful in allowing the designer to make well-informed design decisions. Design formalization of mechanisms is sometimes regarded as a mixture of art and science. Systematics, in the context of mechanisms, consists in generating the entire set of solutions defined by certain structural parameters. Systematics attempts to offer a gift of creativity to the engineering community in the form of formalized procedures and methods. *Type synthesis* (Freudenstein and Dobrjanskyj, 1966, Fang and Tsai, 2004, Kong and Gosselin, 2007, Gogu, 2009) is identified as the main approach in the systematics of mechanisms. Type synthesis of parallel mechanisms involves the determination of the type of mechanical components (*e.g.*, prismatic or revolute joint, gears, cams, screws, ratchets) in order to determine all parallel mechanisms realizing a specified motion. Certain rules must be followed, such as the Chebychev-Grübler-Kutzbachs mobility criterion. Depending on the number of selected actuators a mechanism may be non-redundant, redundantly-actuated, or underactuated. Müeller (2008) provides a detailed introduction and discussion of

the modelling and control of redundant parallel mechanisms. Some advantages of redundancy include larger reachable workspaces, avoidance of kinematic singularities, dexterity improvement, and fault tolerance (Ebrahimi et al., 2008). Merlet (1996) also states the importance of redundancy in solving the forward kinematics, avoiding singular configurations, and improving obstacle avoidance, calibration, and force control. Type synthesis considers only the desired types of motions of the mechanism and does not regard the dimensions of the components.

Dimensional synthesis (Gosselin and Guillot, 1991, Boudreau and Gosselin, 1999, Gallant and Boudreau, 2002) considers the determination of the dimensions of the components in order to achieve a desired performance. It is considered as one of the final steps in the design process with regards to kinematics and statics. Dimensional synthesis of a parallel mechanism may consider: reachability (Laribi et al., 2007), singularities (Arsenault and Boudreau, 2004), force/motion transmissibility (Huang T, 2014) and wrench capabilities (Pickard and Carretero, 2015b), to name a few.

1.5 Problem Statement

An important and often neglected element in all mechanisms are the uncertainties arising from the fabrication and operation of mechanisms. Machining operations are able to achieve a known manufacturing tolerance. Sensors in a mechanism are able to yield data with a known tolerance and control al-

gorithms may perform with known uncertainties. These uncertainties can significantly affect the performance of the mechanism. The actual performance of a mechanism may differ drastically from the expected performance, as conventional analysis methods are unable to account for uncertainties in the mechanism. Unaccounted-for uncertainties in a mechanism can lead to undesirable and unexpected performance.

This thesis considers two main problems:

1. There is a need for methods capable of analyzing parallel mechanisms while considering the inherent uncertainties in the mechanism. Such methods should reliably predict the expected performance of an actual mechanism.

2. Dimensional synthesis methods should be able to automatically account for the uncertainties present in parallel mechanisms. A synthesized design for a parallel mechanism should reliably achieve an expected performance.

1.6 Motivation

The problem of selecting a parallel mechanism which is appropriate for a given task is a difficult one. Conventional synthesis techniques, which rely mainly on discretized optimization-based approaches, provide a reasonable guess for a design solution. The issue with conventional techniques is that

these design solutions provide no guarantee of actual performance. The uncertainties present in the manufacturing and operation of the mechanism prevent the ideal design solution from being physically achievable. There will always exist a deviation between the ideal design and the achievable design. It is therefore not possible to reliably predict the performance of a mechanism using conventional techniques.

The motivation behind this work is that there exists a need for methods which are able to reliably determine a mechanism's performance under uncertainties. Allow the term *appropriate design* to describe a parallel mechanism with uncertainties. Appropriate methods should be able to reliably analyze and synthesize parallel mechanisms described by an appropriate design. The appropriate analysis methods should be able to operate over continuous domains in order to avoid approximation issues associated with discretization. Interval analysis provides the mathematical framework for operating with continuous domains. The appropriate synthesis methods should be able to completely explore the continuous design space and return the complete set of design solutions. The appropriate terminology is summarized in Table 1.1. Development of these appropriate methods would provide a very useful tool for mechanism designers.

1.7 Objectives

The objectives of this work are summarized as follows:

Table 1.1: Summary of exact and appropriate terminology

Term	Description
Exact Design	The conventional design description of a mechanism. All parameters (link dimensions, angles, etc.) are exact values.
Analysis	A procedure which considers the performance of an exact design of a mechanism.
Synthesis	A procedure which uses a desired performance to determine one or more exact designs of a mechanism for a specific task.
Appropriate Design	The design description of a mechanism which accounts for uncertainties in one or more parameters.
Appropriate Analysis	A procedure which considers the performance of an appropriate design of a mechanism.
Appropriate Synthesis	A procedure which uses a desired performance to determine the complete set of appropriate designs of a mechanism for an exact task or a task with uncertainties.

- Develop appropriate analysis routines capable of handling appropriate designs of parallel mechanisms for a reliable analysis of kinematics, wrench-capabilities, singularities, and self-collisions.

- Develop appropriate synthesis routines capable of handling appropriate designs of parallel mechanisms for a reliable and complete synthesis of parallel mechanisms.

1.8 Major Contributions of the Work

The major contributions of the research work detailed in this thesis are:

- Incorporating uncertainties resulting from the fabrication and operation of a mechanism in the appropriate design description, and developing reliable methods for solving the kinematics and statics problems of parallel mechanisms.

- The development of a collision detection routine for appropriate designs which is capable of reliably determining poses of a mechanism which always or never result in collisions.

- Extending wrench capability methods to appropriate designs and demonstrating the performance of the methods on several parallel mechanisms.

- The development of appropriate synthesis techniques which are able to determine the complete set of appropriate design solutions of a mechanism which satisfy a desired task.

- Solving the four-bar linkage synthesis problem using appropriate synthesis techniques in order to determine the complete set of appropriate design solutions for desired coupler curves described by precision-point and trajectories.

- Solving the placement problem or the actuated base joint of the 3-$\underline{\text{R}}$RR using appropriate synthesis techniques in order to determine the

complete set of appropriate design solutions which satisfy a desired task workspace and desired wrench capabilities.

1.9 Outline of Thesis

The rest of the thesis is outlined as follows. In Chapter 2, interval analysis methods are introduced and the various routines, including simplification, existence, and bisection, which are applied throughout this work, are presented. This chapter serves to introduce the reader to the fundamental routines of interval analysis, the mathematical tool used throughout the rest of this work. An understanding of these routines is required in the following chapters. In Chapter 3, the concept of an appropriate design for parallel mechanisms is introduced through the modelling of uncertainties in the geometric and actuation parameters in planar mechanisms. Methods for solving the inverse and forward kinematics problems, specifically the displacement problems, and identifying singular regions are then presented. In Chapter 4, a new method for the detection of self-collisions of parallel mechanism with an appropriate design is proposed. An appropriate design of the 3-$\underline{\text{R}}$RR planar parallel mechanism is selected and several case studies are presented. In Chapter 5, methods for evaluating the wrench capabilities of parallel mechanisms with appropriate designs are developed and presented. In Chapter 6, the appropriate synthesis problem is formulated for the four-bar linkage in order to synthesize the complete set of appropriate design solutions. Then in

Chapter 7, the appropriate synthesis problem is formulated for the 3-$\underline{\text{R}}$RR planar parallel mechanism. The appropriate design solutions account for reachability, self-collisions, wrench-capabilities, and other task requirements. Finally, the thesis is concluded in Chapter 8.

Chapter 2

Interval Analysis Methods

2.1 Reliable Computations

The *interval analysis* framework provide a means of performing reliable computations on computers. Reliability in terms of computations means that the true outcome of the computation is guaranteed to be contained in the actual outcome. It is a well known issue that the floating point representation used by computers has a finite set of representable values, while the other values can only be approximated by selecting the nearest representable value. An insightful quote regarding interval analysis was made by (Hayes, 2003): "*give a digital computer a problem in arithmetic, and it will grind away methodically, tirelessly, at gigahertz speed, until ultimately it produces the wrong answer*". Without careful consideration for floating point representation, conventional arithmetic cannot produce reliable results.

The following example from (Rump, 1988) demonstrates the importance of reliable computations. The example is updated in (Loh and Walster, 2002) for IEEE-754 computers.

Example 2.1.1. Consider the evaluation of $f(x, y)$ defined by

$$f(x, y) = (333.75 - x^2)y^6 + x^2(11x^2y^2 - 121y^4 - 2) + 5.5y^8 + \frac{x}{2y}$$

for $x = 77617.0$ and $y = 33096.0$.

With round-to-nearest, the default IEEE-754 arithmetic, the expression produces:

32-bit	$f(x, y) = 1.172604\ldots$
64-bit	$f(x, y) = 1.1726039400531786\ldots$
128-bit	$f(x, y) = 1.172603940053178631858834904520183 8\ldots$

It appears that the value of $f(x, y)$ converges close to 1.172604 as the precision increases. One would expect this value to be the solution, but it is incorrect. It is shown in (Loh and Walster, 2002) that the expression for $f(x, y)$ may be simplified at $x = 77617.0$ and $y = 33096.0$ using $x^2 = 5.5y^2+1$ to

$$f(x, y) = \frac{x}{2y} - 2$$

from which $f(x, y) = -0.827396059946821368141165095479816\ldots$ (128-bit), which is the correct answer.

Interval arithmetic arose from the need for reliable computations on computers. Uncertainties in the representation of the values on computers can

automatically be taken into account, such that a desired value is replaced by an interval whose bounds are the nearest representable values. Interval arithmetic does not directly improve the accuracy of calculation, but rather gives a certificate of accuracy by providing guaranteed bounds on the solutions. Interval analysis also enables the solution of certain problems that cannot be solved by non-interval methods, like the problem by (Rump, 1988) above. The primary example of this is for global optimization (see (Hansen and Walster, 2003)). Additionally, interval algorithms for solving non-linear systems and global optimization problems are naturally parallel and have been benefiting from the improvements in parallel computing software and hardware.

2.2 Background

Modern understanding of interval analysis is attributed to Moore's dissertation (Moore, 1963) and subsequent book (Moore, 1966). In "The dawning" (Moore, 1999), Moore describes his idea to use computers to perform arithmetic on closed intervals to create a viable tool for bounding the effect of errors. The popularity of interval analysis is due to the growing requirements to solve increasingly difficult non-linear problems, and to do so with guaranteed accuracy.

Let an interval be defined by $[x]$ such that the interval represents a closed

set as

$$[x] = [\underline{x}, \overline{x}] = \{x \mid \underline{x} \leq x \leq \overline{x}\} \tag{2.1}$$

where $\underline{x}$ and $\overline{x}$ are the lower and upper bounds, respectively, of the set.

Interval arithmetic, as introduced by (Moore, 1963), provides a framework for bounding error with guaranteed accuracy. The width of any interval provides a measure of accuracy. Overestimation is inherent in the interval analysis framework (*e.g.*, the *wrapping effect* and the *dependency problem*). Properly accounting for these sources of overestimation is required for effective use of interval analysis.

A few years after Moore's dissertation, interval analysis methods were being developed for computing bounds on the entire set of solutions to non-linear problems. One of the most important interval algorithms is the solving procedure utilising an interval version of Newton's method (discovered by (Moore, 1966)) for finding the roots of non-linear functions. Another much-studied algorithm for verifying the existence of solutions, known as the Krawcyzk method, first appeared in (Krawczyk, 1969).

Methods for bounding the solution set (the *hull*) of an interval linear system have been developed by many researchers (Alefeld and Herzberger, 1983, Rohn, 1989, Neumaier, 1985, Shary, 1995). It has been shown that finding the hull is NP-hard (Heindl et al., 1995). Theorems for the strong feasibility and strong solvability of interval linear systems were developed in (Rohn, 1981). Procedures for determining the feasibility and solvability of interval

linear systems are presented in (Rohn, 2006).

Skelboe (1974) introduced an efficient algorithm, based on refinements, for computing the upper and lower bounds on the range of values for non-linear problems. An interval global optimization algorithm was developed by Hansen for the one-dimensional case in (Hansen, 1979) and for the general case in (Hansen, 1980). In their book, Hansen and Walster (2003) present many sophisticated techniques for interval global optimization.

Early efforts to implement interval arithmetic relied on writing subroutine calls. Thanks to the work on the formalization of the mathematical foundations of finite interval analysis, many programming languages now support interval data types. Outwardly rounded interval arithmetic provides rigorous enclosures for the ranges of operations and functions. Outward rounding (directed rounding) is available on all computers supporting IEEE 754 floating-point standard (Stevenson, 1985). Interval arithmetic operations have been implemented in many software packages. The libraries Profil/BIAS (Programmer's Runtime Optimized Fast Interval Library, Basic Interval Arithmetic) and Gaol are two popular C++ interval arithmetic libraries. The Boost collection of C++ libraries also contains a template class for intervals. Interval analysis methods have been developed from the motivation of improving the accuracy of interval arithmetic computations. *Simplification procedures* (also called *consistency techniques*, *filtering techniques*, or *contractors*) produce narrower (more accurate) intervals (see (Benhamou and Granvilliers, 2006, Benhamou et al., 1999, Collavizza et al., 1999, Lhomme,

1993, Trombettoni and Chabert, 2007)).

There currently exist several well-developed software packages which provide high-level general purpose interval analysis solving routines using interval analysis.

- IBEX C++: (Gilles Chabert – IMT Atlantique, Nantes)
- ALIAS/ALIAS-Maple: (Jean-Pierre Merlet – INRIA)
- RealPaver: (Laurent Granvilliers – University of Nantes)
- INTLAB Matlab/Octave toolbox: (Siegfried M. Rump – Institute for Reliable Computing, Hamburg University of Technology)
- RSolver: (Stefan Ratschan – see Ratschan (2002))
- Intel ® Math Kernel Library

2.3 Interval Arithmetic

As stated in (Moore et al., 2009), the key point of arithmetic operations between intervals is that computing with intervals is computing with sets. The central property of interval arithmetic is the fundamental underlying principle of inclusion isotonicity of interval operations. That is

Theorem 2.3.1 (Inclusion isotonicity)**.** For all interval quantities $[a]$, $[b]$ and all arithmetic operations $\circ \in \{+, -, \times, /\}$, such that $a \circ b$ is defined for

$a \in [a]$, $b \in [b]$, it holds that

$$a \circ b \in [a] \circ [b] \text{ for all } a \in [a],\ b \in [b] \tag{2.2}$$

The basic operations are as follows. Interval addition is represented as

$$[x] + [y] = \{x + y \mid x \in [x], y \in [y]\} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}] \tag{2.3}$$

Interval subtraction is represented as

$$[x] - [y] = \{x - y \mid x \in [x], y \in [y]\} = [\underline{x} - \overline{y}, \overline{x} - \underline{y}] \tag{2.4}$$

Interval multiplication is represented as

$$\begin{aligned} &[x] \cdot [y] = \{x \cdot y \mid x \in [x], y \in [y]\} = [\min(s), \max(s)] \\ &\text{where } s = \{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\} \end{aligned} \tag{2.5}$$

Interval division is represented as

$$\begin{aligned} &[x]/[y] = [x] \cdot 1/[y] \\ &\text{where } 1/[y] = [1/\overline{y}, 1/\underline{y}] \\ &\text{and } 0 \notin [y] \end{aligned} \tag{2.6}$$

Extended interval arithmetic extends ordinary interval arithmetic to allow for divisions by intervals containing 0 (see Appendix B).

The width of an interval is given by

$$width([x]) = \overline{x} - \underline{x} \tag{2.7}$$

We say that $[x]$ is degenerate if $\overline{x} = \underline{x}$ since it has zero width. The value represented by a degenerate interval will be termed *exact*.

The midpoint of an interval is given by

$$mid([x]) = \underline{x} + width([x])/2 \tag{2.8}$$

The intersection of two intervals is an interval given by

$$[x] \cap [y] = [\max(\underline{y}, \underline{x}), \min(\overline{y}, \overline{x})] \tag{2.9}$$

The intersection is empty if either $\overline{x} < \underline{y}$ or $\overline{y} < \underline{x}$. This returns an empty set $\emptyset$.

The union of two intervals, $[x] \cup [y]$, is in general not an interval. However, the *interval hull* is always an interval and is given by

$$[x] \square [y] = [\min(\underline{y}, \underline{x}), \max(\overline{y}, \overline{x})] \tag{2.10}$$

2.4 Interval Vectors and Matrices

Let $[\mathbf{x}]$ denote the interval extension of a vector where each element $[x_i]$ in the vector is an interval. An interval vector is commonly termed a *box*, as it represents a Cartesian product of intervals.

Let $[\mathbf{X}]$ denote the interval extension of a matrix where each element $[x_{ij}]$ in the matrix is an interval.

Operations which apply to vectors and matrices generally still apply to interval vectors and matrices, although interval arithmetic is applied instead of standard arithmetic.

A square interval matrix $[\mathbf{X}]$ is said to be regular if each $\mathbf{X} \in [\mathbf{X}]$ is non-singular. The Baumann regularity theorem (Theorem B.2.1, see Appendix B) may be used to determine if an interval matrix is always non-singular.

2.5 Interval Nonlinear System of Equations

A powerful aspect of interval computations is tied to the Brouwer fixed point theorem:

Theorem 2.5.1 (Brouwer fixed point theorem)**.** If $\mathbb{X}$ is a closed convex subset of $\mathbb{R}^n$ and g maps $\mathbb{X}$ into itself, then g has a fixed point $\mathbf{x}^*$, $\mathbf{x}^* = g(\mathbf{x}^*),\ \mathbf{x}^* \in \mathbb{X}$.

The Brouwer fixed point theorem combined with interval arithmetic enables numerical computations to prove the existence of solutions to non-linear sys-

tems.

A non-linear algebraic or transcendental equation in terms of unknowns $\mathbf{x}$ is given by $f(\mathbf{x})$. The interval extension of the non-linear equation is given by $f([\mathbf{x}])$, such that $f([\mathbf{x}])$ yields an *inclusion function*, denoted $[f([\mathbf{x}])]$. The inclusion function has the property that $f([\mathbf{x}])$ is a subset of $[f([\mathbf{x}])]$. That is,

$$f([\mathbf{x}]) = \{f(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\} \subseteq [f([\mathbf{x}])]. \tag{2.11}$$

The inclusion function is obtained by applying the corresponding interval arithmetic operations. The bounds of the inclusion function are often inflated due to two well known properties of interval analysis. First, the *wrapping effect* is a result of the axis-aligned representation of intervals. The inclusion function $[f([\mathbf{x}])]$ will always be an axis-aligned *box* which contains $f([\mathbf{x}])$ and therefore introduces overestimation to the solution. For example, the smallest interval which can represent a circle in Cartesian coordinates is an axis-aligned circumscribed square. Overestimation is inherent in such a representation. Second, the *dependency problem* is a result of multiple occurrences of a variable appearing in the equation. This causes additional expansion of the solution since interval arithmetic considers each variable occurrence as independent. That is, from the point of view of interval arithmetic, $x \times x$ is the same as $x \times y$.

Consider the simple example.

Example 2.5.1. Let $x = [-1, 1]$ and compute $x \times x$. This gives

$$x \times x = [-1, 1] \times [-1, 1] = [-1, 1]$$

Alternatively, perform the same operation represented as x^2. Due to the single occurrence of x and the properties of exponents, this now gives

$$x^2 = [-1, 1]^2 = [0, 1]$$

It is possible to minimize the effects of the dependency problem by properly writing the equation in a form best suited for interval analysis. The ALIAS-Maple libraries implement several routines which make use of symbolic computations to rewrite the equations in a more optimal form for interval arithmetic.

Let a system of n non-linear algebraic or transcendental equations in terms of m unknowns $\mathbf{x} \in [\mathbf{x}] \subset \mathbb{R}^m$ be described by

$$\mathbf{f}([\mathbf{x}]) = \mathbf{0} = \begin{pmatrix} \mathrm{f}_1([\mathbf{x}]) = 0 \\ \mathrm{f}_2([\mathbf{x}]) = 0 \\ \cdots \ \cdots \ \cdots \\ \mathrm{f}_n([\mathbf{x}]) = 0 \end{pmatrix} \tag{2.12}$$

If the coefficients in the equations are exact, the system is described as being *thin*. The problem of solving the thin system is to find a solution $\mathbf{x}^* \in$

$[\mathbf{x}]$ which $\mathbf{f}$ maps into the origin $\mathbf{0}$. If the system (2.12) contains interval coefficients, such that the system describes a family of systems, then the system is described as being *thick*. The problem of solving a thick system is to find an approximate interval solution $[\mathbf{x}^*] \subset [\mathbf{x}]$ which $\mathbf{f}$ maps "close" to the origin $\mathbf{0}$ for the entire family of systems. Ideally, the bounds on the approximate interval solution should be tight on the set of solutions for each system in the family of systems.

A common criteria for accepting an interval $[\mathbf{x}^*]$ as a good approximation to the solution for a thin or thick system are a small *residual*

$$r = width([\mathbf{f}([\mathbf{x}^*])]) \tag{2.13}$$

or a small correction when calculating $[\mathbf{x}^*]$ as the limit of a sequence

$$\delta_k = width([\mathbf{x}^{(k+1)}] - [\mathbf{x}^k]). \tag{2.14}$$

2.6 Lipschitz Interval Extension

According to the fundamental theorem of interval analysis, we have

$$\mathbf{f}([\mathbf{x}]) \subseteq [\mathbf{f}([\mathbf{x}])]. \tag{2.15}$$

This means that the inclusion function $[\mathbf{f}([\mathbf{x}])]$ contains the range of values of the corresponding real function $\mathbf{f}$ when the real arguments lie in the interval

$[\mathbf{x}]$. The real solution $\mathbf{f}([\mathbf{x}])$ may be approximated as closely as desired by a finite union of intervals.

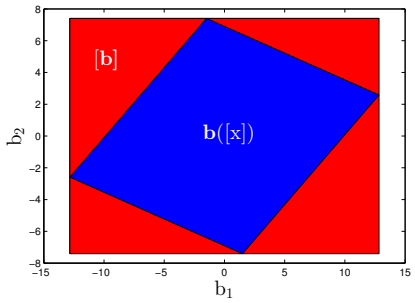
Definition 2.6.1. An interval extension of an equation, $\mathbf{f}([\mathbf{x}])$, is said to be Lipschitz over a domain $\mathbb{X}$ if there is a positive real constant K such that

$$width([\mathbf{f}([\mathbf{x}])]) \leq K\, width([\mathbf{x}]) \text{ for every } [\mathbf{x}] \in \mathbb{X} \tag{2.16}$$

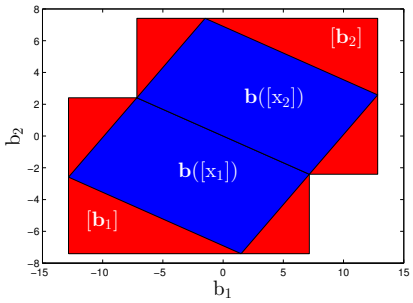
The width of the inclusion function $[\mathbf{f}([\mathbf{x}])]$ approaches zero at least linearly with the width of $[\mathbf{x}]$ (Moore et al., 2009). Therefore, the approximation of the solution to $\mathbf{f}([\mathbf{x}])$ becomes more accurate when the width of $[\mathbf{x}]$ is reduced.

Interval analysis solving routines make use of this property by bisecting the interval $[\mathbf{x}]$ into two new intervals $[\mathbf{x}_1]$ and $[\mathbf{x}_2]$. The union of the inclusion functions $[\mathbf{f}([\mathbf{x}_1])]$ and $[\mathbf{f}([\mathbf{x}_2])]$ better approximate the solution $\mathbf{f}([\mathbf{x}])$. The interval hull $[\mathbf{f}([\mathbf{x}_1])]\square[\mathbf{f}([\mathbf{x}_2])]$ will never be larger than $[\mathbf{f}([\mathbf{x}])]$ for a Lipschitz system.

Consider the following problem. A linear transformation $\mathbf{Ax} = \mathbf{b},\ \mathbf{x} \in [\mathbf{x}]$ is applied to an interval for $\mathbf{A} = (5.67, -7.16; 5.00, 2.41)$ with $[\mathbf{x}] = ([-1, 1], [-1, 1])$. Figure 2.1a shows the exact solution $\mathbf{b}([\mathbf{x}]) = \{\mathbf{b} \mid \mathbf{Ax} = \mathbf{b},\ \mathbf{x} \in [\mathbf{x}]\}$ and the interval solution $[\mathbf{b}] = \mathbf{A}[\mathbf{x}]$. Figure 2.1b shows the exact solution and interval solutions after bisecting $[\mathbf{x}]$ into $[\mathbf{x}_1]$ and $[\mathbf{x}_2]$. The exact solution does not change; however, the union of the interval solution $[\mathbf{b}_1] \cup [\mathbf{b}_2]$ is improved compared to $[\mathbf{b}]$.



(a) $[\mathbf{x}] = ([-1, 1], [-1, 1])$



(b) $[\mathbf{x}_1] = ([-1, 0], [-1, 1])$, $[\mathbf{x}_2] = ([0, 1], [-1, 1])$

Figure 2.1: a) A linear transformation applied to an interval; b) a linear transformation applied to a bisected interval.

2.7 Interval Analysis Solving Routine

An interval analysis solving routine generally contains three phases evaluated in a loop, *simplification*, *existence*, and *bisection* (see Figure 2.2). Let $[\mathbf{k}]$ denote the interval vector of known variables (*i.e.*, the coefficients) and $[\mathbf{u}]$ denote the interval vector of unknown variables in a problem and let $\mathbf{f}([\mathbf{u}], [\mathbf{k}]) = \mathbf{0}$ be the system being solved. The following subsections elaborate on each of the three phases and how an interval solving routine incorporating all three can be formulated.

The initial domain of $[\mathbf{u}]$ is first added to a list $\mathcal{L}_u$. Given the system $\mathbf{f}([\mathbf{u}], [\mathbf{k}]) = \mathbf{0}$, the simplification, existence and bisection phases are applied to each element in the list $\mathcal{L}_u$ consecutively. This creates a loop since a bisection phase is able to add new elements to $\mathcal{L}_u$. The loop repeats until the list $\mathcal{L}_u$ is empty. The goal of the solving routine is to determine the list of unique solutions $\mathcal{L}_{u^*}$ for the unknowns. The known variables $[\mathbf{k}]$ are

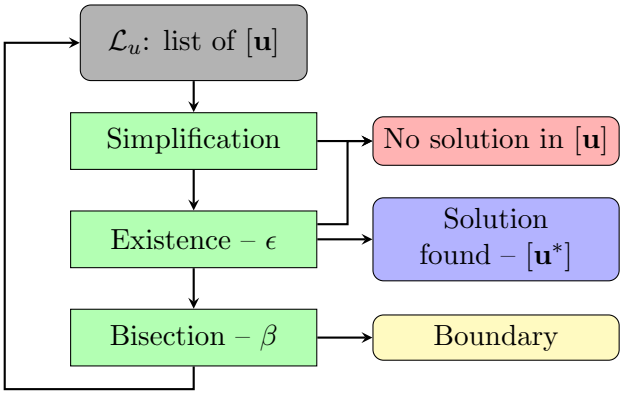


Figure 2.2: Flowchart of the interval analysis general solving routine.

considered as fixed during the solving routine. The method is summarized in Algorithm 1.

2.7.1 Simplification

In the simplification phase, various routines (*e.g.*, 2B (Lhomme, 1993) and 3B filtering (Collavizza et al., 1999), HC4 (Benhamou et al., 1999), ACID (Neveu and Trombettoni, 2013)) can be applied to attempt to reduce the width of [**u**] such that it becomes more consistent with [**k**] for the system $\mathbf{f}([\mathbf{u}], [\mathbf{k}]) = \mathbf{0}$. Consistency is described by the tightness of the unknowns to the actual solution. An unknown $[u_i]$ is considered to be simplified if the width is reduced. If any $[u_i]$ is simplified to an empty set $\emptyset$, then no solution exists for the unknowns. A simplification procedure returns the outcome of the simplification as:

Algorithm 1: General solving routine

```
function GENERAL_SOLVE ([u], [k], ε, β);
Input  : unknowns [u], knowns [k], desired solution resolution ε, and
         desired bisection resolution β
Output: the list of solutions L_u* for the unknowns
L_u ← [u] ;                                  // Add unknowns to list L_u
while L_u is not empty do
    Pop unknowns [u] from L_u;
    simp = simplification([u], [k]) ;  // Apply filtering routines
    if simp ≠ −1 then
        exist = existence([u], [k], ε) ; // Apply existence routines
        if exist == 1 then
            L_u* ← [u];
        else if exist == 0 then
            L_u ← bisection([u], β) ;      // Apply bisection routine
```

-1 : no solution exists in $[\mathbf{u}]$;

1 : $[\mathbf{u}]$ has been simplified;

0 : $[\mathbf{u}]$ has not been simplified.

A function *simplification*$([\mathbf{u}], [\mathbf{k}])$ applies one of more simplification methods to the unknowns $[\mathbf{u}]$. Several simplification methods are provided in Appendix B. Simplification may be repeated as many times as desired, usually until significant simplification can no longer be achieved.

2.7.2 Existence

When the width of $[\mathbf{u}]$ is small enough, existence methods (*e.g.*, Interval Newton (Moore, 1966), Krawcyzk (Krawczyk, 1969), or Kantorovitch (Kantorovich and Forsythe, 1952)) may be applied to determine if a unique so-

lution $[\mathbf{u}^*]$ exists within the domains of $[\mathbf{u}]$ which correspond to $[\mathbf{k}]$ for the system $\mathbf{f}([\mathbf{u}], [\mathbf{k}]) = \mathbf{0}$. The existence methods are able to converge to the unique solution where a unique solution corresponds to a region of non-separable solutions. A tolerance of ϵ is used as the stopping criteria for the convergence (*i.e.*, the existence method stops when the current solution estimate cannot be improved by more than ϵ). The existence methods may also return that no solution exists. An existence method returns the outcome of the existence test as:

$$\begin{aligned} -1 &: \text{no solution exists in } [\mathbf{u}]; \\ 1 &: \text{a unique solution } [\mathbf{u}^*] \text{ is found;} \\ 0 &: \text{otherwise (additional bisection/analysis is required)} \end{aligned}$$

A return value of 0 means that the possibility of a solution exists in $[\mathbf{u}]$, but is not guaranteed and may be a consequence of overestimation as a result of the wrapping effect and dependency problem. Here, the function EXISTENCE$([\mathbf{u}], [\mathbf{k}], \epsilon)$ applies one or more existence methods to $[\mathbf{u}]$. Several existence methods are provided in Appendix B.

2.7.3 Bisection

In the bisection phase, an unknown $[u_i]$ is bisected such that the original unknowns $[\mathbf{u}]$ are split into two subintervals $[\mathbf{u}_1]$ and $[\mathbf{u}_2]$. The union of the two subintervals results in the original interval, thus no combination of unknowns is skipped. A common stopping criteria is to exit when the width of all unknowns are less than a desired threshold β. Generally, the bisection

resolution β is selected to be larger than the existence resolution ϵ. This is because existence methods are quite efficient at iteratively reducing the width of the $[\mathbf{u}]$, whereas bisection is only able to reduce the width once in order to generate the two new unknown domains. The benefit of bisecting the unknowns is that the simplification and existence phases may have greater success when the unknowns have smaller widths. Bisection requires that a list of the bisected interval vectors, denoted $\mathcal{L}_u$, be maintained. The algorithm completes when the list $\mathcal{L}_u$ is empty. Many bisection routines have been developed and while there is currently no optimal technique, the following techniques may usually be applied with good success.

- Largest-first – $[\mathbf{u}]$ is bisected along dimension i such that $[u_i]$ is the interval which has the largest width. The bisection usually splits $[\mathbf{u}]$ along the midpoint of the interval $[u_i]$. This technique suffers when the unknowns contains different units and therefore may require the parameters to be non-dimensionalized.

- Smear function (introduced by Kearfott and Manuel (1990)) – let $\mathbf{J}_{ji}$ be the ith column and jth row of the Jacobian evaluated over $[\mathbf{u}]$. The smear value s_i is evaluated for each unknown i as

$$s_i = \max(|\mathbf{J}_{ji}[u_i]|) \; \forall j \in [1, \ldots, n] \tag{2.17}$$

  The unknown $[u_i]$ with the largest smear value is selected for bisection, as it is considered to be the most sensitive variable.

The bisection of an unknown $[u_i]$ is only applied when $\text{width}([u_i]) > \beta$. Function BISECTION($[\mathbf{u}], \beta$) attempts to apply a bisection technique to the unknowns $[\mathbf{u}]$. Generally, the smear function is the method of choice for bisection, although some problems may benefit from the simpler largest-first bisection. Both methods are applied in this work. A successful bisection returns the two subintervals, while an unsuccessful bisection may return the original interval to save to a boundary list.

Chapter 3

Appropriate Design

There is need for an improved description for the design of mechanisms for the same reason that there is a need for interval analysis – reliable computations. It is not possible to manufacture a mechanism to exact specifications; there will always be some level of uncertainty present in the fabrication and operation of mechanisms. An exact design can provide a reasonable estimation of the real performance; however, it cannot provide any guarantee on real performance. The majority of analysis and synthesis methods developed for mechanisms consider only exact designs.

It is possible to generate a more appropriate description for the design by incorporating the uncertainties of a mechanism into its description. This can be achieved with the use of interval analysis. This type of design description is termed *appropriate design*.

Analysis routines can be formulated for appropriate designs so as to describe

the range of characteristics of the appropriate design. Such a routine will be termed *appropriate analysis*. For example, one may want to obtain the positioning error of a mechanism described by an appropriate design. A positioning error interval may be obtained whose bounds describe the worst-case positioning error over the appropriate design. If an exact design is selected from within the appropriate design, the positioning error of the exact design is guaranteed to be contained within the positioning error interval.

Synthesis routines can be formulated to obtain appropriate designs when an objective contains uncertainties. Such a routine will be termed *appropriate synthesis*. Appropriate synthesis differs from ordinary synthesis because the incorporation of uncertainties requires a specialized formulation which relies on interval analysis. An appropriate design solution can only be obtained from an appropriate synthesis routine, thus the appropriate terminology (summarized in Table 1.1) is used to distinguish between methods capable of handling uncertainties and methods not capable of handling uncertainties.

Consider that a particular mechanism needs to be synthesized in order to accomplish a desired trajectory, and that the desired trajectory is described with uncertainties. There may exist an infinitude of exact design solutions to the problem which presents a major issue for synthesis methods. With appropriate synthesis, every possible design solution can be described by a finite set of appropriate design solutions. Each appropriate design solution is guaranteed to accomplish the desired trajectory. Any exact design selected

from within the appropriate designs accomplishes the desired trajectory. The concept of appropriate synthesis is explored in Chapters 6 and 7.

3.1 Background

Wang and Masory (1993) considered the effects of manufacturing tolerances, installation error, and link offset deviations of the Gough-Stewart platform, and studied the effects of the uncertainties on pose accuracy. Rao (1998) developed one of the first applications of interval analysis for robotic mechanisms. They presented a method for solving the inverse kinematics of several serial manipulators by solving the non-linear system describing the kinematics using interval analysis. Ting et al. (2000) present an approach for identifying the worst-case positioning errors due to uncertainties in the joints of single closed loop mechanisms by applying rotatability laws. Han et al. (2002) address the problem of kinematic sensitivity of the 3-U$\underline{P}$U parallel mechanism. Small uncertainties in the universal joints of the mechanism are shown to cause significant undesirable self-motions. Caro et al. (2005) present approaches for dimensioning and tolerance synthesis of serial mechanisms for obtaining robust designs which attempt to minimize the sensitivity to uncertainties. Sensitivity ellipsoids are used to select the robust designs. Hao and Merlet (2005) propose an interval analysis routine to obtain valid design solutions for the synthesis of the INRIA active wrist. Their method is based on the satisfaction of compulsory requirements for the workspace

and accuracy of the mechanism. A feasible solution is found when either requirement is satisfied, while a valid solution is found when both requirements are satisfied. Merlet and Daney (2005) consider the *error synthesis* and *error analysis* problems applied to a Gough platform. For error synthesis, the authors give a prescribed workspace and use interval analysis techniques to synthesize the set of design parameters such that any pose in the workspace has a positioning error below some desired limit. For the error analysis problem, the authors determine the extremal positioning errors corresponding to some specific geometrical parameters which contain uncertainties. Kotlarski et al. (2009) present an approach based on interval analysis for determining the overall workspace, singularity-free regions, certified accuracy regions of several planar parallel manipulators. The term *appropriate design* was introduced by Merlet and Daney (2008) for the synthesis of parallel manipulators with uncertainties. In their work, appropriate design referred to the synthesis problem, where the appropriate design methodology involves computing *allowable regions* (the set of valid solutions). The end-user is then presented with the allowable regions for the design parameters and is free to select any valid design to achieve the desired performance. Chen et al. (2013) studied the accuracy performance of general planar parallel manipulators due to input uncertainties and joint clearances. A description of the output error solid is obtained in the kinematic image space. Ni et al. (2016) consider the error model and tolerance synthesis of a full-circle rotation parallel mechanism. A genetic algorithm is applied to determine the manufacturing tolerances for

each component given allowable end-effector orientation errors. Kaloorazi et al. (2016b) develop an approach based on interval analysis for determining the maximal singularity free ball inside the workspace of parallel manipulators. Ting et al. (2017) studied the effect of joint uncertainties in linkages containing revolute and prismatic joints.

3.2 Modelling Uncertainties

The uncertainties associated with the geometrical design parameters and operational parameters can be modelled by replacing the exact parameters with interval representations, such that the uncertainties of the mechanism are contained in the design description. To achieve reliable computations, each source of uncertainty should be properly represented in the appropriate design description.

Let $\mathcal{D}$ denote the usual (exact) design description, whose parameters are described exactly, and let $[\mathcal{D}]$ denote an appropriate design. Certain characteristics are associated with an appropriate design. An appropriate design has the property that each design $\mathcal{D} \in [\mathcal{D}]$ shares these characteristics. Consider the mechanism synthesis problem. If an appropriate design $[\mathcal{D}]$ of a mechanism is able to satisfy the requirements of a task, then any design $\mathcal{D} \in [\mathcal{D}]$ must also satisfy these requirements. This provides the user with a set of design solutions, from which any design may be chosen.

Considering planar mechanisms, an appropriate design may consist of planar

uncertainties arising from the following

- Errors in the distances between points:

  The distance between two points may be considered as the desired distance l inflated[1] by the uncertainty Δl. An appropriate distance is therefore: $[l] = l \pm \Delta l$.

- Errors in the joint connections between two bodies:

  The connection between two joints introduces uncertainties in the form of "play". The uncertainties in a joint connection are modelled by a worst-case error of $\Delta\mathbf{c}$. The appropriate location of a joint is therefore the desired joint location $\mathbf{c}$ inflated by the connection uncertainty $\Delta\mathbf{c}$ as: $[\mathbf{c}] = \mathbf{c} \pm \Delta\mathbf{c}$. Visually, $[\mathbf{c}]$ is a Cartesian product of intervals and may be interpreted as a box.

- Errors in the actuator control:

  Actuators introduce an uncertainty to the control of a mechanism. The combination of actuator, sensors, and controller results in an uncertainty in the real output of the actuator. A worst-case actuator uncertainty can be assumed to be $\Delta\theta$. The appropriate output of the actuator is therefore the desired output θ inflated by the actuator uncertainty $\Delta\theta$: $[\theta] = \theta \pm \Delta\theta$.

[1] Inflated means that the upper bounds of the interval are increased and the lower bounds are decreased

The uncertainty in the pose of a mechanism is a function of the uncertainties in the links, joints, and control of the actuators.

To accurately predict the performance of real mechanisms, spatial uncertainties need to be considered. Spatial uncertainties influence the performance of both planar and spatial mechanisms. The previously mentioned planar uncertainties may be extended to spatial uncertainties. As well, errors in the desired orientation of axes must now also be accounted for. However, spatial uncertainties are quite difficult to model and are not considered in the remainder of this work. They are considered as possible future extensions to the methods that are developed here.

3.3 Kinematics of an Appropriate Design

Let $\mathbf{C}([\mathbf{q}], [\mathbf{p}], [\mathcal{D}])$ be an interval system of constraints describing the kinematics of a mechanism, where $[\mathbf{q}]$ represent the joint variables, $[\mathbf{p}]$ represent the pose, and $[\mathcal{D}]$ is the appropriate design. The system describes a mechanism with uncertainties. The simplest interval system of constraints may not be the best formulation for an appropriate design. In many cases, it is more effective to add redundant constraints or use a representation which contains a non-minimal number of variables. For example, an interval system of constraints describing the kinematics of a mechanism may be more easily solved when represented in terms of Cartesian coordinates, as opposed to angles or Polar coordinates. Interval analysis generally requires a trade-off between

the complexity of the constraints and the number of variables.

3.3.1 Inverse Displacement Problem

Given an interval system of constraints $\mathbf{C}([\mathbf{q}], [\mathbf{p}], [\mathcal{D}])$, the inverse kinematics (displacement) problem is solved by finding the set of corresponding active joint variables $\mathbb{Q}$. The joint angle solutions $[\mathbf{q}]$ will be intervals. That is,

$$\mathbb{Q} \leftarrow \mathit{Solve}(\mathbf{C}([\mathbf{q}], [\mathbf{p}], [\mathcal{D}]), [\mathbf{q}]) \tag{3.1}$$

such that the system $\mathbf{C}([\mathbf{q}], [\mathbf{p}], [\mathcal{D}])$ is solved for the active joint variables $[\mathbf{q}]$ and the solutions are added to the set $\mathbb{Q}$.

An algorithm for computing the inverse kinematics solution is proposed in Algorithm 2 which makes use of the general solving routine described in Algorithm 1. The known variables $[\mathbf{k}]$ and the unknown variables $[\mathbf{u}]$ are problem dependant, but generally the desired pose $[\mathbf{p}]$ is considered as known and the joint variables are considered as unknown. Given the desired pose $[\mathbf{p}]$, the algorithm returns the set of joint angle solutions $\mathbb{Q}$ for the inverse kinematics. Additional constraints may be enforced by adding them to the constraint system $\mathbf{C}([\mathbf{q}], [\mathbf{p}], [\mathcal{D}])$. The constraint system is used inside the general solving routine to generate the simplification, existence, and bisection routines.

Algorithm 2: Compute the inverse kinematics with interval analysis

function COMPUTE_INVERSE_KINEMATICS $([\mathbf{u}], [\mathbf{k}], \epsilon, \beta)$;
Input : unknowns $[\mathbf{u}]$, knowns $[\mathbf{k}]$, desired solution resolution ϵ, and desired bisection resolution β
Output: the list of joint variable solutions $\mathbb{Q}$ for the unknowns
$\mathbb{Q} \leftarrow$ *GENERAL_SOLVE*$([\mathbf{u}], [\mathbf{k}], \epsilon, \beta)$; `/* Call the general solving routine */`

3.3.2 Forward Displacement Problem

Given an interval system of constraints $\mathbf{C}([\mathbf{q}], [\mathbf{p}], [\mathcal{D}])$, the forward kinematics (displacement) problem is solved by finding the set of corresponding end-effector poses $\mathbb{P}$. That is,

$$\mathbb{P} \leftarrow \mathit{Solve}(\mathbf{C}([\mathbf{q}], [\mathbf{p}], [\mathcal{D}]), [\mathbf{p}]) \tag{3.2}$$

such that the system $\mathbf{C}([\mathbf{q}], [\mathbf{p}], [\mathcal{D}])$ is solved for the end-effector pose $[\mathbf{p}]$ and the solutions are added to the set $\mathbb{P}$.

Similar to the inverse kinematics solver, an algorithm for computing the forward kinematics solution is proposed (Algorithm 3) which makes use of the general solving routine. The known variables $[\mathbf{k}]$ and the unknown variables $[\mathbf{u}]$ are again problem dependant, but generally the desired joint variables $[\mathbf{q}]$ are considered as known and the pose $[\mathbf{p}]$ is considered as unknown. Given the desired joint variables $[\mathbf{q}]$, the algorithm returns the set of corresponding pose solutions $\mathbb{P}$ for the forward kinematics. Additional constraints may also be enforced by adding them to the constraint system $\mathbf{C}([\mathbf{q}], [\mathbf{p}], [\mathcal{D}])$. The

constraint system is used inside the general solving routine to generate the simplification, existence, and bisection routines.

Algorithm 3: Compute the forward kinematics with interval analysis

function COMPUTE_FORWARD_KINEMATICS $([\mathbf{u}], [\mathbf{k}], \epsilon, \beta)$;
Input : unknowns $[\mathbf{u}]$, knowns $[\mathbf{k}]$, desired solution resolution ϵ, and desired bisection resolution β
Output: the list of pose solutions $\mathbb{P}$ for the unknowns
$\mathbb{P} \leftarrow$ *GENERAL_SOLVE*$([\mathbf{u}], [\mathbf{k}], \epsilon, \beta)$; `/* Call the general solving routine */`

The forward kinematics are necessary for following trajectories. For the trajectory problem, the initial search domain for the poses may be narrowed, such that only solutions from a particular region are found. A safe-starting search domain[2] must be used with existence methods to ensure that the forward kinematics solver returns the correct solution, based on the previous pose.

Consider the following trajectory problem. Assume that the current pose $[\mathbf{p}_0]$ is known and the that the corresponding joint variables are $[\mathbf{q}_0]$. The joint variables are perturbed such that the new joint variables are obtained as $[\mathbf{q}_1]$. The problem is to determine the new pose $[\mathbf{p}_1]$. The Kantorovitch method, called using the new joint variables $[\mathbf{q}_1]$ with the estimation of the new pose as $[\mathbf{p}_0]$, is able to verify if a unique solution exists for the new pose and tightly converges to it. Let the design parameters be represented by the interval vector $[\mathbf{d}]$. Assume that the interval system of constraints

[2] A safe-starting search domain is a domain which has been determined to have a single unique solution (*i.e.*, the Kantorovitch condition is satisified).

is able to be formulated for a mechanism with the unknowns selected as $[\mathbf{u}] = [\mathbf{p}_0]$ and the knowns selected as $[\mathbf{k}] = ([\mathbf{q}_1], [\mathbf{d}])$. The Kantorovitch method is called. If it can be found, the existence ball will be centred at $[\mathbf{p}_0]$ and the solution for the unknowns will exist inside the existence ball. This solution will correspond to the new pose $[\mathbf{p}_1]$. The difference between the previous joint variables and new joint variables must be small in order for Kantorovitch to be successful.

3.4 Interval Evaluation of the Jacobian Matrix

Since the Jacobian matrix, $\mathbf{J} = {\mathbf{J}_\mathbf{q}}^{-1}\mathbf{J}_\mathbf{x}$ is pose-dependent, each element J_{ij} of $\mathbf{J}$ is interval evaluated over the pose $[\mathbf{p}]$, thereby yielding an interval $[J_{ij}]$. The $m \times n$ interval matrix $[\mathbf{J}]$ whose elements are the intervals $[J_{ij}]$ has the fundamental property

$$\forall\ \mathbf{p} \in [\mathbf{p}],\ \mathbf{J}(\mathbf{p}) \in [\mathbf{J}]. \tag{3.3}$$

In other words, for every pose $\mathbf{p} \in [\mathbf{p}]$, the Jacobian matrix obtained for $\mathbf{p}$ belongs to $[\mathbf{J}]$. Consequently, the interval matrix $[\mathbf{J}]$ overestimates the set $\{\mathbf{J}(\mathbf{p}) \mid \mathbf{p} \in [\mathbf{p}]\}$. There exists some matrices $\mathbf{J}_0 \in [\mathbf{J}]$ where $\forall\ \mathbf{p} \in [\mathbf{p}],\ \mathbf{J}_0 \neq \mathbf{J}(\mathbf{p})$; this is due to the wrapping effect and the dependency problem. Note that the Jacobian matrix is also a function of the appropriate design $[\mathcal{D}]$,

i.e., $\mathbf{J}([\mathcal{D}], [\mathbf{p}])$, such that

$$\forall\ \mathbf{p} \in [\mathbf{p}],\ \forall\ \mathcal{D} \in [\mathcal{D}],\ \mathbf{J}(\mathcal{D}, \mathbf{p}) \in [\mathbf{J}] \tag{3.4}$$

3.5 Singularities of an Appropriate Design

Certifying the absence of singularities within a prescribed workspace is essential for the practical use of parallel mechanisms (Merlet, 2007b). An appropriate design will always yield interval matrices for the Jacobian matrix. For non-redundant mechanisms (*i.e.*, square systems), singular poses of an appropriate design can be determined by considering the determinants of the interval Jacobian matrices. For a given pose $[\mathbf{p}]$, the mechanism is non-singular when

$$0 \notin det([\mathbf{J_x}([\mathcal{D}], [\mathbf{p}])]) \text{ and } 0 \notin det([\mathbf{J_q}([\mathcal{D}], [\mathbf{p}])]) \tag{3.5}$$

This does not imply that a mechanism is singular when

$$0 \in det([\mathbf{J_x}([\mathcal{D}], [\mathbf{p}])]) \text{ or } 0 \in det([\mathbf{J_q}([\mathcal{D}], [\mathbf{p}])]) \tag{3.6}$$

Due to the overestimation present in the Jacobians, the existence of a singularity cannot be guaranteed by (3.6). If the interval evaluation of the determinants at a given pose do not have constant signs, either the workspace will include singularities or it cannot be stated that the workspace is singularity-

free without using a more accurate arithmetic.

The proximity of a pose to a singularity can be determined by considering the bounds on the determinant of the Jacobians (Merlet, 2007b). Let a positive constant α be chosen in such a way that the a pose interval $[\mathbf{p}]$ with an absolute value of the determinants lower than α is unsafe from a control viewpoint. A pose interval $[\mathbf{p}]$ is considered safe if

$$\underline{|det(\mathbf{J_x}(\mathcal{D}, [\mathbf{p}]))|} > \alpha \text{ and } \underline{|det(\mathbf{J_q}(\mathcal{D}, [\mathbf{p}]))|} > \alpha \tag{3.7}$$

An additional test may be introduced for improving the detection of singularity-free regions of the workspace based on the regularity of the interval matrices of the Jacobians. The Baumann regularity theorem (see Chapter 2) may be incorporated to determine if the Jacobians are regular. If the Jacobians are regular, then they are guaranteed to be non-singular.

Connected singularity-free regions can be found by considering the signs of the determinant of the interval Jacobian matrices (Merlet, 2009). Assume that at a pose $\mathbf{p}_1$ the determinant is positive. If it is possible to determine a pose $\mathbf{p}_2$ at which the determinant is negative, then it is guaranteed that any path path joining $\mathbf{p}_1$ and $\mathbf{p}_2$ has to cross a singular pose. Conversely, the set of adjacent non-singular poses with identical determinant signs are connected singularity-free regions.

3.6 Appropriate Analysis of the 3-RRR

The appropriate design $[\mathcal{D}]$ of a 3-RRR can be described in terms of the uncertainties on the design parameters. Illustrated in Figure 3.1, the appropriate design parameters include

proximal link lengths	$[r_i]$
distal link lengths	$[l_i]$
moving platform geometry	$[\mathbf{d}_i]$
fixed base geometry	$[\mathbf{a}_i]$

The result of the appropriate design is that it is no longer possible to obtain an exact end-effector pose (position and orientation). Let the actuator angles be given by the intervals $[\alpha_i] = \alpha_i \pm \Delta\alpha_i$. The pose can be represented as an interval vector $[\mathbf{p}] = ([x], [y], [\psi])$, such that

$$\forall \alpha_1 \in [\alpha_1],\ \forall \alpha_2 \in [\alpha_2],\ \forall \alpha_3 \in [\alpha_3],\ \forall \mathcal{D} \in [\mathcal{D}],\ \exists \mathbf{p} \in [\mathbf{p}] \tag{3.8}$$

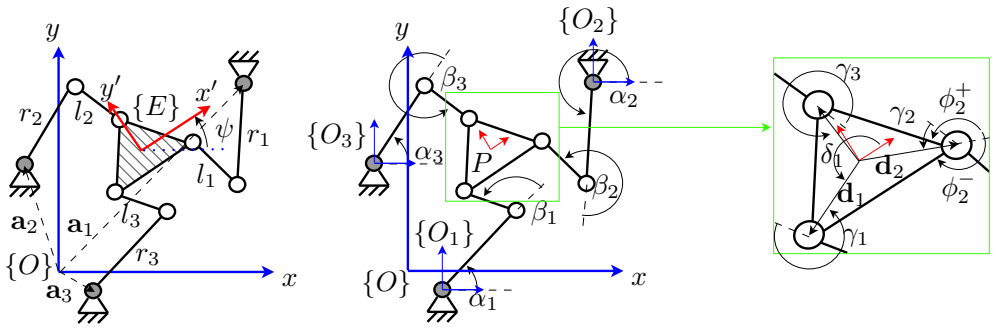


Figure 3.1: The 3-RRR architecture with 3-DOF (x, y, and ψ) (adapted from (Pickard and Carretero, 2015c)).

That is, for every angle and every design the resulting pose $\mathbf{p}$ is contained inside $[\mathbf{p}]$, although $[\mathbf{p}]$ may be overestimated.

The 3-$\underline{\text{R}}$RR architecture (see Figure 3.1) consists of three limbs, each consisting of three revolute joints and two links, attaching the moving platform to the fixed base. The joint angles are described by α_i, β_i, γ_i, for each limb $i = 1, \ldots, 3$. Each limb is given an individual reference frame $\{O_i\}$ located at the limb's fixed base position with the same orientation as the base frame $\{O\}$. The origin of each reference frame is denoted by Point A_i. Let point B_i be the location of the joint between the proximal and distal links. Let point C_i be the location of the joint between the distal link and platform. Let point P be the location of the end-effector. The mechanism's complete pose is defined as $\mathbf{p} = (P_x, P_y, \psi)$.

3.6.1 Inverse Kinematics

A closed-form solution for the inverse kinematics of the 3-$\underline{\text{R}}$RR is given in Appendix C. The angles α and β for each limb i may be obtained as:

$$\begin{aligned}
\cos(\beta_i) &= \frac{C_{xi}^2 + C_{yi}^2 - r_i^2 - l_i^2}{2 r_i l_i} \\
\sin(\beta_i) &= \pm\sqrt{1 - \cos(\beta_i)^2} \\
\cos(\alpha_i) &= (C_{xi}(r_i + l_i \cos(\beta_i)) + C_{y_i} l_i \sin(\beta_i))/({C_{x_i}}^2 + {C_{y_i}}^2) \\
\sin(\alpha_i) &= (C_{y_i}(r_i + l_i \cos(\beta_i)) - C_{xi} l_i \sin(\beta_i))/({C_{x_i}}^2 + {C_{y_i}}^2)
\end{aligned} \tag{3.9}$$

The computed values for the angles will therefore be significantly overestimated when any uncertainties are present. For this reason, it is useful to introduce additional equations which can aid in reducing the overestimation. The following trigonometric identity may be added as an additional equation to aid in reducing overestimation on $\cos(\alpha_i)$ and $\sin(\alpha_i)$.

$$\cos(\alpha_i)^2 + \sin(\alpha_i)^2 - 1 = 0 \tag{3.10}$$

Also, (C.2) and (C.3) may be used. However, they will be converted to the form

$$\begin{aligned} r_i \cos(\alpha_i) + l_i(\cos(\alpha_i)\cos(\beta_i) - \sin(\alpha_i)\sin(\beta_i)) - C_{xi} = 0 \\ r_i \sin(\alpha_i) + l_i(\sin(\alpha_i)\cos(\beta_i) + \cos(\alpha_i)\sin(\beta_i)) - C_{y_i} = 0 \end{aligned} \tag{3.11}$$

Lastly, equations for the angles γ_i will be added, which may be utilized to determine collisions (see Chapter 4). These are:

$$\begin{aligned} \cos(\gamma_i) = \cos(\delta_i - \text{atan2}(\sin(\alpha_i), \cos(\alpha_i)) - \text{atan2}(\sin(\beta_i), \cos(\beta_i))) \\ \sin(\gamma_i) = \sin(\delta_i - \text{atan2}(\sin(\alpha_i), \cos(\alpha_i)) - \text{atan2}(\sin(\beta_i), \cos(\beta_i))) \end{aligned} \tag{3.12}$$

where, atan2 is the quadrant-corrected inverse tangent.

The constraint system used to solve the inverse kinematic for each limb i may be written as

$$\mathbf{C}([\mathbf{q}],[\mathbf{p}],[\mathcal{D}])_i = \left\{ \begin{array}{l} \cos(\beta_i) = ({C_{x_i}}^2 + {C_{y_i}}^2 - (r_i^2 + l_i^2))/(2 r_i l_i); \\ \sin(\beta_i) - \mathrm{conf}_i \sqrt{1 - \cos(\beta_i)^2} = 0; \\ \cos(\alpha_i) = (C_{xi}(r_i + l_i \cos(\beta_i)) + C_{y_i} l_i \sin(\beta_i))/({C_{x_i}}^2 + {C_{y_i}}^2); \\ \sin(\alpha_i) = (C_{y_i}(r_i + l_i \cos(\beta_i)) - C_{xi} l_i \sin(\beta_i))/({C_{x_i}}^2 + {C_{y_i}}^2); \\ r_i \cos(\alpha_i) + l_i(\cos(\alpha_i)\cos(\beta_i) - \sin(\alpha_i)\sin(\beta_i)) - C_{xi} = 0; \\ r_i \sin(\alpha_i) + l_i(\sin(\alpha_i)\cos(\beta_i) + \cos(\alpha_i)\sin(\beta_i)) - C_{y_i} = 0; \\ \cos(\alpha_i)^2 + \sin(\alpha_i)^2 = 1; \\ \cos(\gamma_i) = \cos(\delta_i - \mathrm{atan2}(\sin(\alpha_i), \cos(\alpha_i)) \\ \qquad - \mathrm{atan2}(\sin(\beta_i), \cos(\beta_i))); \\ \sin(\gamma_i) = \sin(\delta_i - \mathrm{atan2}(\sin(\alpha_i), \cos(\alpha_i)) \\ \qquad - \mathrm{atan2}(\sin(\beta_i), \cos(\beta_i))); \end{array} \right\} \tag{3.13}$$

where the term conf_i is used to select the limb configuration ($\mathrm{conf}_i = 1$ selects the elbow-right configuration, and $\mathrm{conf}_i = -1$ selects the elbow-left configuration).

The inverse kinematics may be solved by ensuring that a pose $[\mathbf{p}]$ yields consistent interval domains for the joint variables. Oetomo et al. (2008) proposed a method for solving the inverse kinematics of the 3-$\underline{\mathrm{R}}$RR by taking the sines and cosines of the angles as variables. That is, the full constraint system, for all three limbs, has the following interval variables

$$[C_{xi}] \text{ for } i = 1, \ldots, 3$$

$$[C_{y_i}] \text{ for } i = 1, \ldots, 3$$

$$[r_i] \text{ for } i = 1, \ldots, 3$$

$$[l_i] \text{ for } i = 1, \ldots, 3$$

$$\text{conf}_i \text{ for } i = 1, \ldots, 3$$

$$[\sin(\beta_i)] \text{ for } i = 1, \ldots, 3$$

$$[\cos(\beta_i)] \text{ for } i = 1, \ldots, 3$$

$$[\sin(\alpha_i)] \text{ for } i = 1, \ldots, 3$$

$$[\cos(\alpha_i)] \text{ for } i = 1, \ldots, 3$$

$$[\sin(\gamma_i)] \text{ for } i = 1, \ldots, 3$$

$$[\cos(\gamma_i)] \text{ for } i = 1, \ldots, 3$$

where $[C_{xi}]$ and $[C_{y_i}]$ are functions of the pose $[\mathbf{p}]$.

Each sine and cosine of the joint variables must be consistent with the domain $[-1, 1]$ for the pose to be achievable. The allowable domain of each joint variable can also be narrowed to limit the allowable range of motion of a joint.

The interval analysis inverse kinematic solving routine (Algorithm 2) may be called, where the knowns for each limb i are considered to be

$$[\mathbf{k}] = \big([C_{xi}], [C_{y_i}], [r_i], [l_i], \text{conf}_i\big)$$

and the unknowns are considered to be

$$[\mathbf{u}] = ([\sin(\beta_i)], [\cos(\beta_i)], [\sin(\alpha_i)], [\cos(\alpha_i)], [\sin(\gamma_i)], [\cos(\gamma_i)]) \,.$$

The existence method applied here must ensure that the joint variables are consistent with the domain $[-1, 1]$, thus three classifications may be achieved:

1. *Inside*: Every pose $\mathbf{p} \in [\mathbf{p}]$ is guaranteed to be reachable. The inside classification is obtained if every joint variable interval is a subset of $[-1, 1]$ (or the desired narrowed domain).

2. *Outside*: Every pose $\mathbf{p} \in [\mathbf{p}]$ is guaranteed to be unreachable. The outside classification is obtained if one or more joint variables return an empty intersection with $[-1, 1]$ (or the desired narrowed domain).

3. *Boundary*: There exists a pose $\mathbf{p} \in [\mathbf{p}]$ which may be unreachable. If either the inside or outside classifications are not obtained, then the boundary classification is given. Bisection of a domain which was previously classified as boundary may yield a different classification.

The 3-$\underline{\text{R}}$RR planar parallel manipulator with an appropriate design $[\mathcal{D}]$, de-

scribed by the following interval design parameters, is analysed.

$$[\mathcal{D}] = \left\{ \begin{array}{l} [\mathbf{a}_1] = ([0.0996, 0.1008], [0.1076, 0.1088])^T \text{ m}; \\ [\mathbf{a}_2] = ([-0.2420, -0.2407], [0.0542, 0.0554])^T \text{ m}; \\ [\mathbf{a}_3] = ([0.0727, 0.0738], [-0.2370, -0.2358])^T \text{ m}; \\ [\mathbf{d}_1] = ([0.0489, 0.0511], [-0.0001, 0.0001])^T \text{ m}; \\ [\mathbf{d}_2] = ([-0.0256, -0.0244], [0.0424, 0.0442])^T \text{ m}; \\ [\mathbf{d}_3] = ([-0.0256, -0.0244], [-0.0442, -0.0424])^T \text{ m}; \\ [r_1] = [0.1648, 0.1650] \text{ m}; \\ [r_2] = [0.1350, 0.1352] \text{ m}; \\ [r_3] = [0.1479, 0.1481] \text{ m}; \\ [l_1] = [0.2411, 0.2413] \text{ m}; \\ [l_2] = [0.2955, 0.2957] \text{ m}; \\ [l_3] = [0.2895, 0.2897] \text{ m}. \end{array} \right\} \tag{3.14}$$

A constant orientation is selected with a platform orientation specified as $[\psi] = [0.0]$ rad. The corresponding set of inside, outside, and boundary pose intervals (boxes) are depicted in Figure 3.2. A resolution of $\beta = 0.001$m is used as a stopping criteria for the bisection. Each pose $\mathbf{p} \in [\mathbf{p}]$ for each inside box is guaranteed to be reachable with a platform orientation of $[\psi] = [0.0]$ rad for each design $\mathcal{D} \in [\mathcal{D}]$.

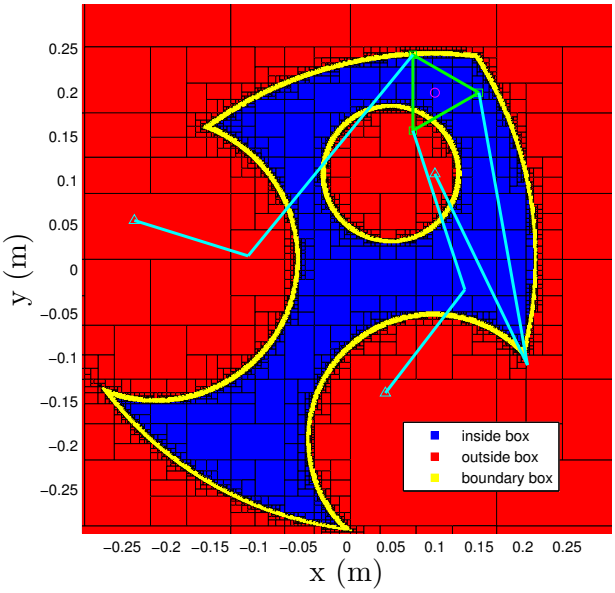


Figure 3.2: The reachable workspace of the 3-RRR planar parallel mechanism.

3.6.2 Forward Kinematics

Without the need to consider the passive joint angles β_i and γ_i, the forward kinematics may be written from distance constraints for each limb i as

$$\begin{aligned}
\mathbf{C}([\mathbf{q}],[\mathbf{p}],[\mathcal{D}])_i &= (C_{xi} - B_{xi})^2 + (C_{yi} - B_{yi})^2 = l_i^2 \\
\mathbf{C}([\mathbf{q}],[\mathbf{p}],[\mathcal{D}])_i &= ((g_i \cos(\psi + \delta_i) + P_x) - (A_{xi} + r_i \cos(\alpha_i)))^2 + \\
&\quad ((g_i \sin(\psi + \delta_i) + P_y) - (A_{y_i} + r_i \sin(\alpha_i)))^2 = l_i^2;
\end{aligned} \tag{3.15}$$

where g_i is the distance from the origin of frame E to joint B_i. It has been shown that the system (3.15) may be converted to a sixth-degree univariate polynomial which may then be solved numerically (Merlet, 2005), such that

there are at most six possible solutions to the forward kinematics.

The interval analysis forward kinematics solving routine (Algorithm 3) may be called with the knowns

$$[\mathbf{k}] = ([\alpha_i], [\delta_i], [A_{xi}], [A_{y_i}], [r_i], [l_i])$$

and the unknowns

$$[\mathbf{u}] = ([P_x], [P_y], [\psi]).$$

The following exact design is selected (Eq. (3.16)), such that the exact design is a subset of the previously described appropriate design.

$$\mathcal{D} = \left\{ \begin{array}{l} \mathbf{a}_1 = (0.0996777, 0.107694)^T \text{ m}; \\ \mathbf{a}_2 = (-0.241939, 0.0542745)^T \text{ m}; \\ \mathbf{a}_3 = (0.0726504, -0.23703)^T \text{ m}; \\ \mathbf{d}_1 = (0.0489999, 0.0000086)^T \text{ m}; \\ \mathbf{d}_2 = (-0.0244927, 0.0424394)^T \text{ m}; \\ \mathbf{d}_3 = (-0.0245070, -0.0424312)^T \text{ m}; \\ r_1 = 0.1648 \text{ m}; \\ r_2 = 0.1350 \text{ m}; \\ r_3 = 0.1479 \text{ m}; \\ l_1 = 0.2411 \text{ m}; \\ l_2 = 0.2955 \text{ m}; \\ l_3 = 0.2895 \text{ m}. \end{array} \right\} \tag{3.16}$$

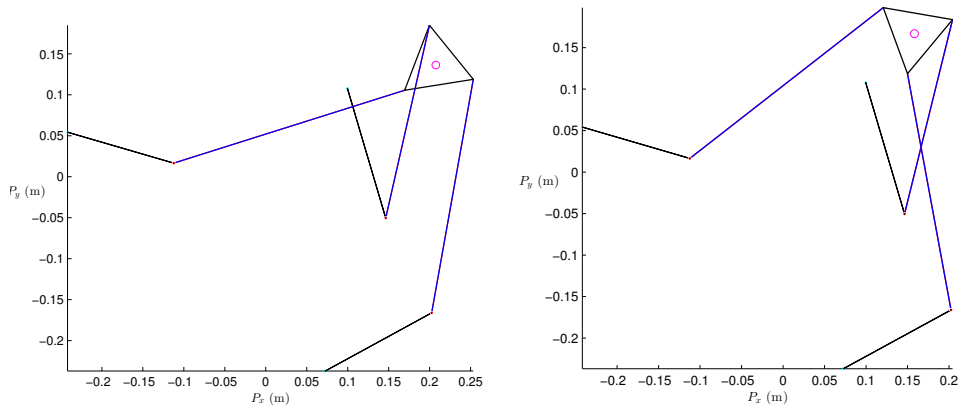


Figure 3.3: The two solutions for the forward kinematics of the 3-RRR planar parallel mechanism at $\alpha_1 = 5.0$ rad, $\alpha_2 = 6.0$ rad and $\alpha_3 = 0.5$ rad.

The forward kinematics, solved at $\alpha_1 = 5.0$ rad, $\alpha_2 = 6.0$ rad and $\alpha_3 = 0.5$ rad for (3.15) yields two pose solutions in the domain $P_x = [-10, 10]$, $P_y = [-10, 10]$, $\psi = [0, 2\pi]$:

1. $P_x = 0.15832\ldots, P_y = 0.166846\ldots, \psi = 0.351451\ldots,$

2. $P_x = 0.207523\ldots, P_y = 0.13639\ldots, \psi = 1.72908\ldots.$

All other poses are inconsistent with the system (3.15). The two solutions are plotted in Figure 3.3.

The forward kinematics may also be solved, without modification, using appropriate design parameters. The appropriate lengths of the proximal and distal links from (3.14) are substituted into the exact design (3.16). Solving the forward kinematics at $\alpha_1 = 5.0$ rad, $\alpha_2 = 6.0$ rad and $\alpha_3 = 0.5$ rad yields the two interval pose solutions:

1. $P_x = [0.15786, 0.158783], P_y = [0.166757, 0.167133], \psi = [0.34392, 0.354937],$

2. $P_x = [0.20736, 0.207966], P_y = [0.136158, 0.136673], \psi = [1.72574, 1.73711]$.

Since this appropriate design contains the previous exact design, the interval pose solutions also contain the pose solutions of the exact design. If the uncertainties on the design parameters are too large, the existence methods applied to the system may likely not succeed. This is because the system (3.15), as well as its Jacobian and Hessian, contain multiple occurrences of variables. The effect of the dependency problem is less significant when the uncertainties are small.

3.6.3 Kinematic Singularities

The singularity-free workspace may be obtained for an appropriate design of the 3-$\underline{\text{R}}$RR by checking (3.7) and the regularity of the Jacobians (Theorem B.2.1) for each pose interval. The direct and inverse Jacobians of the 3-$\underline{\text{R}}$RR are formulated in Appendix C. The singularity-free workspace with an unsafe constant of $\alpha = 0.0$ and a bisection resolution of $\beta = 0.001$ is provided in Figure 3.4. The selected appropriate design consists of the exact design (3.16) with appropriate lengths of the proximal and distal links from (3.14). The blue regions are guaranteed to be singularity free, the red regions are unreachable, and the yellow regions are near-singular. The interval-based singularity detection method is quite effective and it able to obtain classifications of pose intervals with relatively large widths.

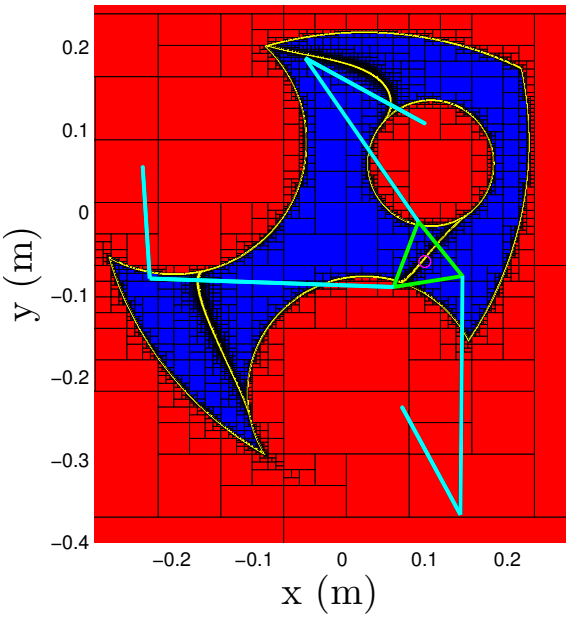


Figure 3.4: The singularity-free workspace of the 3-RRR.

Chapter 4

Self-Collision Detection for Appropriate Designs

Self-collisions are typically prevalent throughout the workspace of parallel mechanisms. For an exact design, the description of a self-collision is quite straight-forward – *the intersection of two members of the mechanism.* It is simple enough to ensure that two members do not collide by verifying that the distance between the members is always greater than some minimum allowable distance. When uncertainties in the design parameters are considered, conventional methods for determining self-collisions cannot be applied. For an appropriate design, the determination of a self-collision is significantly more complicated than for an exact design. It is necessary to determine if there is *always a self-collision* or *never a self-collision* for every design $\mathcal{D} \in [\mathcal{D}]$. Furthermore, the pose may be an interval, such that each

$\mathbf{p} \in [\mathbf{p}]$ must also always, or never, result in a self-collision. This chapter presents a method for detecting self-collisions for appropriate designs. The methods are derived for planar mechanisms, although these methods may be extended for spatial mechanisms.

4.1 Background

Collision detection plays an important role in the trajectory validation and motion planning of parallel mechanisms. In parallel mechanism literature, several techniques have been proposed to deal with self-collisions for exact designs. Merlet (1994) presented an algorithm for detecting the self-collisions of a Gough-Stewart platform based on an analysis of the algebraic inequalities describing the workspace constraints. Their method accounts for the limited range of motion of the prismatic and passive joints and the interference between links, where all links are approximated by cylinders. Chablat and Wenger (1998) introduced the notion of free aspects for fully parallel mechanisms with several inverse and forward kinematic solutions in the presence of collisions. They define a free aspect as the maximal singularity-free domain without self-collisions or collisions with external objects. The projection of a free aspect onto the pose-space or joint-space yields the respective collision-free workspace. Cortés and Siméon (2003) developed Probabilistic RoadMap methods to approximate the topology of the collision-free configuration space. Merlet and Daney (2006) applied interval analysis techniques to determine

the minimum distance between cylindrical members to detect self-collisions in the Gough-Stewart platform. Ketchel and Larochelle (2008) exploited the geometry of right circular cylindrical objects to facilitate the detection of collisions from three dimensional motions. They developed necessary and sufficient conditions to determine if a pair of infinite length cylinders collide. If the actual finite length rigid bodies collide, then it is necessary that the infinite length cylinder models also collide. A second stage of the algorithm then determines if the finite length cylinders collide. Blanchet and Merlet (2014) developed an interval analysis based algorithm for detecting collisions of CDPMs. The method is able to detect collisions between pairs of cables and collisions between a cable and an external object. Uncertainties resulting from the cable sag and interval pose are considered in the method in order to determine always valid, always invalid, or undetermined poses. Kaloorazi et al. (2016a) presented an interval analysis based algorithm for detecting collisions with external objects for a special class of parallel mechanisms with prismatic actuation.

These works cannot be directly applied to general parallel mechanisms described by an appropriate design, aside from the work by Blanchet and Merlet (2014) which has limited use for mechanisms other than CDPMs. The problem of detecting self-collisions for appropriately designed parallel mechanisms which contain more than one link per limb is considered here. A geometric technique which utilizes interval analysis is developed for the determination of collision-free poses.

4.2 Self-Collisions of an Appropriate Design

A self-collision detection method, for appropriately designed parallel mechanisms, consists of a *partial collision test* and a *full collision test*. The partial collision test first determines if a self-collision is possible. The full collision test is then applied to determine if a self-collision will always occur.

In addition to detecting self-collisions, the method here is also capable of modelling the layered construction of planar mechanisms. That is, planar mechanisms typically consist of several non-colliding layers which aid in enlarging the collision-free workspace. It is not necessary to check for collisions between every member during each pose test; instead, it is useful to track which members are guaranteed not to collide and pass this information along in the bisection. Only the members which have the possibility of colliding must be rechecked.

Consider a symmetric parallel mechanism with n_{limbs} limbs, each with n_{links} links. Let $\mathbf{a}_i$ describe the location of the base joint in limb i, and let $\mathbf{b}_i$, $\mathbf{c}_i$, ... describe each consecutive joint along the limb. A line segment, denoted $\overline{\mathbf{a}_i\mathbf{b}_i}$, $\overline{\mathbf{b}_i\mathbf{c}_i}$, ..., provides a representation of a slender link. The effect of uncertainties on the mechanism are that the joint locations become intervals, *e.g.*, $[\mathbf{a}_i]$, $[\mathbf{b}_i]$, $[\mathbf{c}_i]$.

The workspace of the ith limb, denoted L_i, of a mechanism with an appropriate design can be represented by the union of segments as:

$$L_i = \overline{\mathbf{a}_i\mathbf{b}_i} \cup \overline{\mathbf{b}_i\mathbf{c}_i} \cup \ldots, \ \mathbf{a}_i \in [\mathbf{a}_i], \mathbf{b}_i \in [\mathbf{b}_i], \mathbf{c}_i \in [\mathbf{c}_i], \ldots \tag{4.1}$$

The width of link k on limb i is represented by w_{ik}. Two links will collide when they are in the same layer and the distance (*dist*) between the links is less than or equal to the sum of the half-widths of the corresponding links. For convenience, L_{ik} will denote the kth link on limb i and L_{jh} will denote the hth link on limb j. Assuming that two limbs i and j are in the same layer, a collision will always occur when

$$\begin{aligned} &\forall \mathbf{a}_i \in [\mathbf{a}_i],\ \forall \mathbf{a}_j \in [\mathbf{a}_j],\ \forall \mathbf{b}_i \in [\mathbf{b}_i],\ \forall \mathbf{b}_j \in [\mathbf{b}_j],\ \forall \mathbf{c}_i \in [\mathbf{c}_i],\ \forall \mathbf{c}_j \in [\mathbf{c}_j], \ldots, \\ &\quad dist(L_{ik}, L_{jh}) \leq \left(\frac{w_{ik}}{2} + \frac{w_{jh}}{2} \right),\ \text{for } k = 1, \ldots, n_{links},\ h = 1, \ldots, n_{links} \end{aligned} \tag{4.2}$$

such that each pair of links is checked for collisions. Alternatively, k and h in Eq. (4.2) may be modified to only consider links in the same layer.

As an example, consider an appropriate design of 3-$\underline{\text{R}}$RR planar parallel mechanism, as depicted in Figure 4.1. The proximal link in limb i is modelled by the set of line segments $\overline{\mathbf{a}_i\mathbf{b}_i}$, $\mathbf{a}_i \in [\mathbf{a}_i], \mathbf{b}_i \in [\mathbf{b}_i]$. The distal link is modelled by the set of line segments $\overline{\mathbf{b}_i\mathbf{c}_i}$, $\mathbf{b}_i \in [\mathbf{b}_i], \mathbf{c}_i \in [\mathbf{c}_i]$. If (4.2) is true, then for every pose $\mathbf{p} \in [\mathbf{p}]$ limbs i and j collide.

For limbs consisting of two or more links it is also necessary to account for collisions between the first $n_{links} - 1$ links and the platform. The last link shares a joint with the platform and restrictions to angle between the link and platform can be used to prevent collisions. The following method applies to limbs consisting of two or more links. Let L_p denote the set of line segments corresponding to the platform. For the 3-$\underline{\text{R}}$RR, this is $L_p = \overline{\mathbf{c}_1\mathbf{c}_2} \cup \overline{\mathbf{c}_2\mathbf{c}_3} \cup \overline{\mathbf{c}_3\mathbf{c}_1}$.

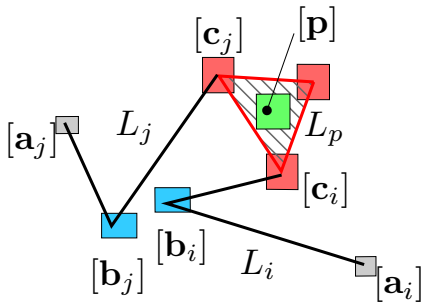


Figure 4.1: Modelling self-collisions for the 3-RRR planar parallel mechanism.

The edges of the platform are located at a distance of w_p from L_p. A collision will always occur between limb i and the platform when

$$\begin{aligned} \forall \mathbf{a}_i \in [\mathbf{a}_i],\ \forall \mathbf{b}_i \in [\mathbf{b}_i],\ \forall \mathbf{c}_1 \in [\mathbf{c}_1],\ \forall \mathbf{c}_2 \in [\mathbf{c}_2],\ \forall \mathbf{c}_3 \in [\mathbf{c}_3], \\ dist(L_{ik}, L_p) \leq \left(\frac{w_{ik}}{2} + w_p\right),\ \text{for } k = 1 \end{aligned} \tag{4.3}$$

4.3 Partial Collision Test

In computational geometry, the *convex hull* of a set of points $\mathbb{X}$ is the smallest convex set (polytope) that contains those points. In a convex combination, each point $X_i \in \mathbb{X}$ is assigned a weight λ_i in such a way that the weights are all non-negative and sum to one. The convex hull is described by the set

$$conv(\mathbb{X}) = \left\{ \sum_{i=1} (\lambda_i X_i) \;\middle|\; \sum_{i=1} \lambda_i = 1,\ \lambda_i \geq 0 \right\}. \tag{4.4}$$

For two points, the convex hull is the straight line connecting the points. For three points, the convex hull is the triangle with the three points as vertices.

The set of line segments corresponding to a link can be modelled by applying the convex hull routine to respective joint intervals of the link. That is, set of line segments describing the first link in limb i can be modelled by the convex hull as

$$\{L_{i1}\} = \left\{\overline{\mathbf{a}_i\mathbf{b}_i} \mid \mathbf{a}_i \in [\mathbf{a}_i], \mathbf{b}_i \in [\mathbf{b}_i]\right\} = conv([\mathbf{a}_i], [\mathbf{b}_i]) \tag{4.5}$$

The entire limb can be modelled by taking the union of the convex hulls corresponding to each link.

For the 3-$\underline{\text{R}}$RR, the set of all line segments for the proximal link ($\{L_{i1}\}$), distal link ($\{L_{i2}\}$), and platform ($\{L_p\}$) are represented as

$$\begin{aligned} \{L_{i1}\} &= conv([\mathbf{a}_i], [\mathbf{b}_i]) \\ \{L_{i2}\} &= conv([\mathbf{b}_i], [\mathbf{c}_i]) \\ \{L_p\} &= conv([\mathbf{c}_1], [\mathbf{c}_2], [\mathbf{c}_3]) \end{aligned} \tag{4.6}$$

The set of line segments which model the entire limb (see Figure 4.2) are

$$\{L_i\} = \{L_{i1}\} \cup \{L_{i2}\} \tag{4.7}$$

The distance between each pair of links $\{L_{ik}\}$ and $\{L_{jh}\}$ is computed and a

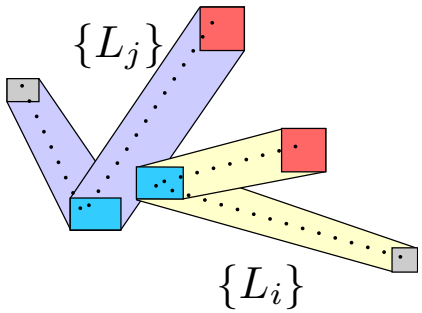


Figure 4.2: Generating the set of all line segments corresponding to the limbs for the partial collision test.

collision will not occur between limbs i and j if

$$dist(\{L_{ik}\}, \{L_{jh}\}) > \left(\frac{w_{ik}}{2} + \frac{w_{jh}}{2}\right), \text{ for } k = 1, \ldots, n_{links}, \ h = 1, \ldots, n_{links} \tag{4.8}$$

A collision between the first $n_{links} - 1$ links and the platform will not occur if

$$dist(\{L_{ik}\}, \{L_p\}) > \left(\frac{w_{ik}}{2} + w_p\right), \text{ for } k = 1, \ldots, n_{links} - 1 \tag{4.9}$$

The software library CGAL (version 4.9) is used to compute the 2D convex hulls to obtain the corresponding polytopes. It is also used to determine the distance between polytopes.

The possibility of a collision exists when Eq. 4.8 or Eq. 4.9 is not satisfied. In order to determine if a collision will always occur for every pose $\mathbf{p} \in [\mathbf{p}]$, the *Full Collision Test* must be applied when the possibility of a collision is determined.

4.4 Full Collision Test

To solve the full collision test, the use of a vertex representation of $\{L_i\}$ and $\{L_p\}$ is proposed. The vertex representations, denoted as $\{L_i\}_v$ and $\{L_p\}_v$ respectively, are finite in terms of line segments and are computed by selecting a vertex from the first joint and connecting this to each consecutive corresponding vertex (see Figure 4.3). Since the mechanism is planar, $\{L_i\}_v$ and $\{L_p\}_v$ will each have four elements, corresponding to each of the vertices.

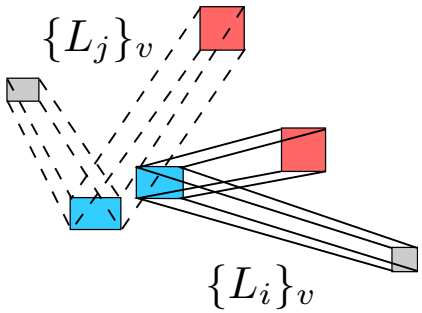


Figure 4.3: Generating the vertex representation for set of all line segments corresponding to the limbs for the full collision test.

A collision will always occur between limbs i and j when

$$\forall L_i \in \{L_i\}_v,\ \forall L_j \in \{L_j\}_v,\ dist(L_{ik}, L_{jh}) \leq \left(\frac{w_{ik}}{2} + \frac{w_{jh}}{2}\right), \quad \text{for } k = 1, \ldots, n_{links},\ h = 1, \ldots, n_{links} \tag{4.10}$$

A collision will always occur between the first $n_{links} - 1$ links and the platform

when

$$\forall L_i \in \{L_i\}_v,\ \forall L_p \in \{L_p\}_v,\ dist(L_{ik}, L_p) \leq \left(\frac{w_{ik}}{2} + w_p\right), \quad \text{for } k = 1, \ldots, n_{links} - 1 \tag{4.11}$$

The software library CGAL is used to model the line segments and to determine the distance between pairs of line segments.

If the full collision test fails, then only a partial collision can be guaranteed and the pose interval $[\mathbf{p}]$ must be bisected for additional testing.

4.5 Case Studies

The 3-$\underline{\text{R}}$RR planar parallel mechanism with appropriate design $[\mathcal{D}]$, described by the following interval design parameters, is analysed.

$$[\mathcal{D}] = \left\{ \begin{array}{l} [\mathbf{a}_1] = ([0.0997, 0.0998], [0.1077, 0.1078])^T \text{ m}; \\ [\mathbf{a}_2] = ([-0.2419, -0.2418], [0.05427, 0.05428])^T \text{ m}; \\ [\mathbf{a}_3] = ([0.07265, 0.07266], [-0.2370, -0.2369])^T \text{ m}; \\ [\mathbf{d}_1] = ([0.0498, 0.0502], [-0.0001, 0.0001])^T \text{ m}; \\ [\mathbf{d}_2] = ([-0.0251, -0.0249], [0.0432, 0.0434])^T \text{ m}; \\ [\mathbf{d}_3] = ([-0.0251, -0.0249], [-0.0434, -0.0432])^T \text{ m}; \\ [r_1] = [0.1648, 0.1650] \text{ m}; \\ [r_2] = [0.1350, 0.1352] \text{ m}; \\ [r_3] = [0.1479, 0.1481] \text{ m}; \\ [l_1] = [0.2411, 0.2413] \text{ m}; \\ [l_2] = [0.2955, 0.2957] \text{ m}; \\ [l_3] = [0.2895, 0.2897] \text{ m}; \end{array} \right\} \tag{4.12}$$

All of the links of the mechanism have a width of $w = 0.0150$ m, while the platform has $w_p = 0.0075$ m. A constant orientation is selected with a platform orientation specified as $[\psi] = [0.0]$ rad. A bisection resolution of $\beta = 0.001$ m is used for all workspaces. The constant orientation reachable workspace is provided in Figure 4.4a.

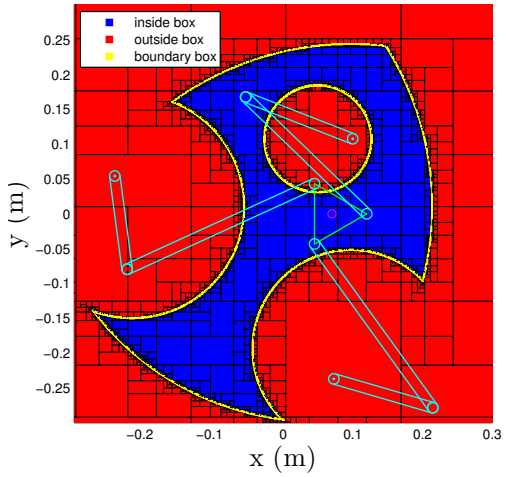
A design with all of the links on layer 1 is considered in Figure 4.4b. The joints limitations are set as $\alpha_i \in [0°, 360°]$, $\beta_i \in [0°, 165°]$, and $\gamma_i \in [-135°, 135°]$ for $i = 1, \ldots, 3$. Collisions between the proximal links, distal links and the platform are accounted for. The set of poses with the *inside* classification are guaranteed to be reachable and collision-free for the

appropriate design. Alternatively, the set of poses with the *outside* classification are guaranteed to be unreachable and/or result in a collision.

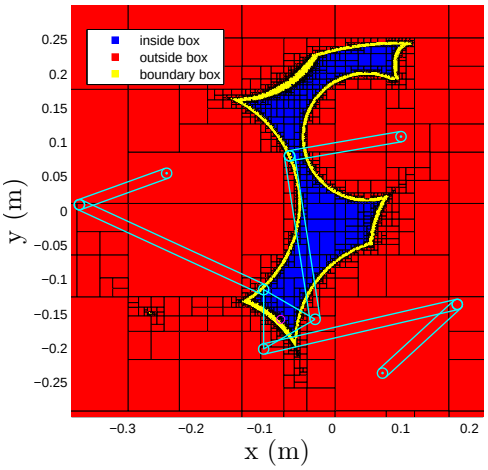
A design with the proximal links in layer 1, the distal links in layer 2, and the platform in layer 3 is considered in Figure 4.4c. Distal-distal and proximal-proximal collisions are accounted for. The joints limitations are $\alpha_i \in [0°, 360°]$, $\beta_i \in [0°, 180°]$, and $\gamma_i \in [0°, 360°]$ for $i = 1, \ldots, 3$.

A design with the proximal links and platform in layer 1 and the distal links in layer 2 is considered in Figure 4.5. Note also that the elbow configurations have been changed. Distal-distal, proximal-proximal and proximal-platform collisions are accounted for. The joints limitations are $\alpha_i \in [0°, 360°]$, $\beta_1 \in [180°, 360°]$, $\beta_2 \in [180°, 360°]$, $\beta_3 \in [0°, 180°]$ and $\gamma_i \in [0°, 360°]$ for $i = 1, \ldots, 3$. It can be noted that two disconnected components of the workspace are computed corresponding to different assemblies of the mechanism. A single assembly can be considered by adding restrictions to γ_1.

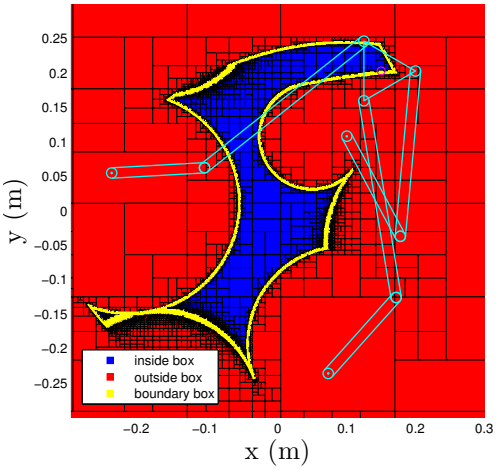
The development of a self-collision detection method applicable to parallel mechanisms with an appropriate design is an important part of the design process. In Chapter 7, self-collision detection is incorporated into an appropriate synthesis routine in order to synthesize parallel mechanisms which achieve a desired performance while also guaranteeing that the workspace is collision free.



(a) Neglecting self-collisions.



(b) Accounting for self-collisions with all links in layer 1.



(c) Accounting for self-collisions with the proximal links in layer 1, the distal links in layer 2, and the platform in layer 3.

Figure 4.4: Constant orientation and collision-free workspaces of the 3-RRR planar parallel mechanism

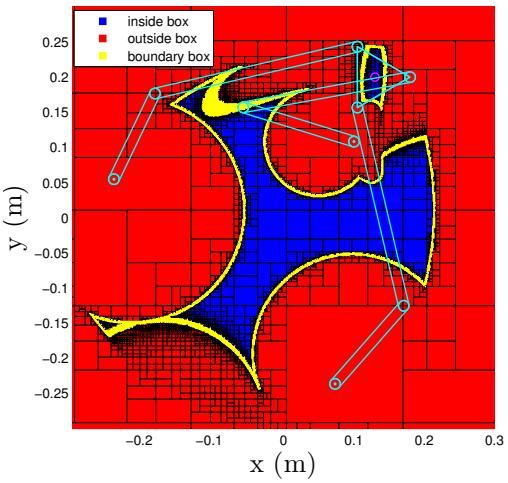


Figure 4.5: Constant orientation and collision-free workspaces of the 3-RRR planar parallel mechanism – accounting for self-collisions with the proximal links and platform in layer 1 and the distal links in layer 2.

Chapter 5

Wrench Capability Analysis

The analysis of the relation between the articular forces or torques of a mechanism and the wrench applied on the end-effector is called *static analysis*. An introduction to the statics problem is provided in Appendix A. The study of the wrench generating capabilities of a mechanism considering the actuator capabilities is called *wrench capability analysis*. When selecting a mechanism for a specific task, it is important to select an architecture with a design which is capable of achieving desired wrench capabilities and traversing desired trajectories. This is particularly important in the context of the development of human-friendly robots in which, for safety reasons, the external wrenches must be restricted. The term *Wrench Workspace*, denoted by the symbol $\mathbb{P}_{WW}$, refers to the set of poses of a mechanism which achieve desired wrench capabilities.

The determination of the wrench workspace for an exact design may be per-

formed by considering a grid of end-effector poses. Each grid point is tested for being wrench-capable (*i.e.*, the mechanism is able to generate the desired wrenches at the given pose). Discretization provides a straightforward method for determining a set of wrench-capable poses, for a desired resolution ϵ, which collectively may be used to describe an approximate wrench workspace $\mathbb{P}_{\widetilde{WW}}$. The approximate wrench workspace $\mathbb{P}_{\widetilde{WW}}$ provides a guess at the true wrench workspace $\mathbb{P}_{WW}$ by discretizing the pose space.

Computational geometry tools, such as α-shapes (*i.e.*, *"the shape formed from these points"*), can be used to obtain a linear approximation of the boundaries of the $\mathbb{P}_{\widetilde{WW}}$ from the set of wrench-capable poses (or $\mathbb{P}_{RW}$ from the set of reachable poses). Intuitively, an α-shape (in 3D) is equivalent to fitting a set of spheres, where α is the squared radius, to a set of points, such that no points are contained within any sphere. When a sphere touches any three points, a facet is created. The polyhedrons formed from the set of facets is a linear approximation of the shape formed from the points. The same technique can also be applied in 2D where the sphere becomes a circle and a facet becomes an edge. An α-shape is able to detect both exterior and interior surfaces; thus, holes in workspaces are detected, so long as $\sqrt{\alpha} > \epsilon/2$ to account for the discretization grid spacing. The C++ library `CGAL 4.5` provides several useful α-shapes algorithms.

Assuming that the grid points are uniform, the *α-shape* routine may be applied to the wrench-capable poses to obtain a polytopic approximation of wrench workspace $\mathbb{P}_{\widetilde{WW}}$. The workspace $\mathbb{P}_{\widetilde{WW}}$ provides no guarantee that

all interior poses are wrench-capable. This is because out of the infinitely many poses in the search space, only a finite quantity – those on the discretization grid – are actually tested. There may exist poses which are not wrench-capable within the $\mathbb{P}_{\widetilde{WW}}$. This is especially true since the wrench workspace may have a non-convex geometry, may contain holes, and may also be separable. Gouttefarde et al. (2011) state *"the result provided by a discretization can never be guaranteed,* i.e.*, one can never know if this result can be trusted."*. It is desirable to use techniques which can be applied to closed sets, such that every element in the set is tested. The interval analysis framework provides the necessary tools.

In this chapter, wrench capability analysis methods which are applicable to appropriate designs are developed and introduced. Methods which have been developed for the wrench capability analysis of exact designs, but are not relevant for appropriate designs, are provided in Appendix D.

5.1 Background

Papadopoulos and Gonthier (1999) studied the force capabilities of serial mechanisms and described the usable workspace of the mechanism for large-force quasi-static tasks as the *force workspace*. Riechel and Ebert-Uphoff (2004) proposed an analytical method for deriving the boundaries of the force-feasible workspace (similar to the force workspace but with strictly non-negative tension forces in the cables) for CDPMs. Nokleby et al. (2005)

developed a methodology to determine the force capabilities of both redundantly and non-redundantly actuated planar parallel mechanisms, known as the scaling factor method. The actuator limits are incorporated into an unconstrained optimization problem based on the Moore-Penrose pseudoinverse technique and singular value decomposition. The optimization-based solution generated larger maximum applicable force magnitudes for redundantly actuated mechanisms in comparison to the non-redundant mechanisms. Bosscher et al. (2006) developed a method to determine analytical expressions for the wrench-feasible workspace boundaries for planar, spatial, and point-mass CDPMs. The concept of required and available wrench sets are used to formulate the set of boundary equations. The authors state that because of the complexity of forming the upper boundaries, formulating the analytical expressions is not feasible without an efficient numerical method. Zibil et al. (2007) discuss limitations of the scaling factor method including the computationally intensive process of specifying a desired moment caused by the loss of accuracy due to many local minima in the optimization's search space. They present an explicit method for determining the force-moment capabilities of redundantly actuated planar parallel mechanisms. The method consists of properly adjusting actuator outputs to their maximum capabilities, such that the applied or sustained wrench is maximized. Four studies were presented: maximum force with prescribed moment, maximum applicable force, maximum moment with a prescribed force, and maximum applicable moment. The explicit method is extended by Garg et al. (2009a) for

redundantly actuated spatial parallel mechanisms. They derive the wrench capabilities of the 3-RRRS spatial parallel mechanism. The concept of a wrench polytope is introduced by Firmani et al. (2008a). They state that a wrench polytope is an exact representation of the force and moment capabilities in the task space and propose six study cases of determining wrench polytopes for planar parallel mechanisms. The same authors, in (Firmani et al., 2008b), then derive the wrench workspaces for their each of the study cases. Carretero et al. (2008) proposed a convex-hull method for obtaining the wrench polytope. The authors clarified the linear mapping performed by the Jacobian matrix and noted that the wrench polytope is actually a special type of centrosymmetrical polytope known as a zonotope. Coincidentally, Bouchard et al. (2008) proposed a similar convex-hull algorithm. The special properties of the zonotope are utilized to construct a new method of deriving the wrench capabilities of CDPMs, known as the hyperplane shifting method. The hyperplane shifting method uses the Jacobian matrix and the articular forces (cable tensions) to create a non-iterative method for obtaining the wrench capabilities. The authors also report on the geometric significance of a task and describe a routine for determining if a pose is wrench feasible for different task wrench requirements, namely points, convex polytopes, and ellipsoids. Gouttefarde and Krut (2010) provide a proof which justifies the hyperplane shifting method mainly based on a characterization of zonotope facets. Their proof leads directly to an improved version of the hyperplane shifting method. Carretero and Gosselin (2010) applied the

convex-hull method to CDPMs to study the wrench capabilities throughout a desired trajectory. Gouttefarde et al. (2011) proposed an interval-based method to obtain the guaranteed *wrench-feasible workspace* for a CDPM. They propose a test based on two sufficient conditions, namely, a sufficient condition for a pose interval to be fully inside the wrench-feasible workspace, and a sufficient condition for a pose interval to be fully outside of the wrench-feasible workspace. Combined within a usual branch-and-bound algorithm, the test enables guaranteed wrench-feasible workspace determinations. A means of mitigating overestimation is also presented by decomposing the Jacobian matrix and modifying the articular forces to minimize the effect of the dependency problem. Kaloorazi et al. (2013) applied the interval-based method to optimize the position of the actuators of a CDPM for which the wrench-feasible workspace covers a given circle. A method of determining the maximal workspace circle based on interval analysis techniques is also presented. A comparison of the convex-hull method and interval-based method applied to a CDPM is provided in (Pickard and Carretero, 2014).

5.2 Interval Methods for the Wrench Capabilities of Appropriate Designs

For an appropriate design $[\mathcal{D}]$, methods based on interval analysis must be developed to verify that the mechanism is capable of generating desired wrench capabilities $\mathcal{F}_{desired}$. Similar to the exact design, the wrench capabilities are

a function of the interval pose $[\mathbf{p}]$ and appropriate design $[\mathcal{D}]$: $\mathcal{F}([\mathbf{p}],[\mathcal{D}])$. As a result of the interval arithmetic, the inclusion function $[\mathcal{F}([\mathbf{p}],[\mathcal{D}])]$ overestimates $\mathcal{F}([\mathbf{p}],[\mathcal{D}])$. That is:

$$\mathcal{F}([\mathbf{p}],[\mathcal{D}]) \subseteq [\mathcal{F}([\mathbf{p}],[\mathcal{D}])] \tag{5.1}$$

In order for an interval pose to be wrench-capable, the desired wrench capabilities must be a subset of the true wrench capabilities:

$$\mathcal{F}_{desired} \subseteq \mathcal{F}([\mathbf{p}],[\mathcal{D}]) \tag{5.2}$$

Let the pose search space be given by $\mathbb{P}$. For a given pose search space $\mathbb{P}$ and desired wrench capabilities $\mathcal{F}_{desired}$, the wrench workspace $\mathbb{P}_{WW}$ is

$$\mathbb{P}_{WW} = \{[\mathbf{p}] \mid \forall\ \mathbf{p} \in [\mathbf{p}],\ \forall\ \mathcal{D} \in [\mathcal{D}],\ \mathcal{F}_{desired} \subseteq \mathcal{F}([\mathbf{p}],[\mathcal{D}]),\ [\mathbf{p}] \in \mathbb{P}\}\,. \tag{5.3}$$

Interval analysis derived tests can be used to verify the wrench capabilities of a mechanism described by an appropriate design $[\mathcal{D}]$. These tests allow for verification over a pose interval $[\mathbf{p}]$, such that each pose $\mathbf{p} \in [\mathbf{p}]$ is guaranteed to generate the desired wrench set for each design $\mathcal{D} \in [\mathcal{D}]$.

Gouttefarde et al. (2011) applied an inside-outside test for the verification of the wrench capabilities of CDPMs. Their inside test makes use of a strong feasibility theorem proposed by Rohn (2006) which relies on the assumption that the solution of an interval linear system of equations is non-negative.

This restriction makes the tests in (Gouttefarde et al., 2011) suitable for cable-driven architectures which must maintain non-negative cable tensions. The non-negative assumption for the actuator capabilities may be removed by making use of the strong solvability theorem proposed in (Rohn, 2006).

5.2.1 System of Interval Linear Equations with Bounded Solutions

Assuming that $\mathcal{F}_{desired}$ is represented in the form of an interval $[\mathcal{F}_{desired}]$, the $\mathbb{P}_{WW}$ can be represented by the forward-force solution in the form of a *system of interval linear equations*, such that at each pose:

$$[\mathbf{J}]^T \boldsymbol{\tau} = [\mathcal{F}_{desired}],\ \boldsymbol{\tau} \in [\boldsymbol{\tau}] \tag{5.4}$$

which amounts to testing infinitely many linear systems, such that

$$\forall\ \mathbf{J} \in [\mathbf{J}],\ \forall\ \mathbf{F} \in [\mathcal{F}_{desired}],\ \exists\ \boldsymbol{\tau} \in [\boldsymbol{\tau}]\ |\ \mathbf{J}^T\boldsymbol{\tau} = \mathbf{F} \tag{5.5}$$

Eq. (5.5) states that the entire set of required wrenches (*i.e.*, $[\mathcal{F}_{desired}]$) can be generated for all $\mathbf{J} \in [\mathbf{J}]$ given the actuator capabilities $[\boldsymbol{\tau}]$. In order for $[\mathbf{J}]$ to be finite, it is necessary that $[\mathbf{J_q}]$ be invertible, such that $\forall\ \mathbf{J_q} \in [\mathbf{J_q}],\ \exists\ \mathbf{J_q}^{-1}$. According to Rohn (2006), a system of linear equations ($\mathbf{Ax} = \mathbf{b}$) is called a) *solvable* if it has a solution and b) *feasible* if it has a non-negative solution, *i.e.*, feasibility implies non-negative solvability. The system of interval linear

equations $[\mathbf{A}]\mathbf{x} = [\mathbf{b}]$ is understood to represent the family of all systems of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{A} \in [\mathbf{A}],\ \mathbf{b} \in [\mathbf{b}]$. The interval system is said to be *strongly solvable* (*strongly feasible*) if each subsystem is solvable (feasible). Eq. (5.4) is a system of interval linear equations with a bounded solution, $\boldsymbol{\tau} \in [\boldsymbol{\tau}]$, such that the interval system is strongly solvable (strongly feasible) and also accounts for the actuator capabilities. It is important to use the proper theorem to ensure that each of the infinitely many linear systems are properly verified.

5.2.1.1 Vertex Matrices and Vertex Vectors

The strong solvability and strong feasibility theorems in (Rohn, 2006) require a vertex representation for the system of interval linear equations, where Y_n is the set of 2^n unique n-dimensional vectors $\mathbf{y}$ whose components y_i are either 1 or -1. For example, if $n = 2$ then $Y_n = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4\} = \{(1,1), (-1,1), (1,1), (-1,-1)\})$.

For an $n \times m$ interval matrix $[\mathbf{A}]$, whose components are intervals $[A_{ij}] = [\underline{A_{ij}}, \overline{A_{ij}}]$, the corresponding vertex matrix $\mathbf{A}_\mathbf{y}$ for each $\mathbf{y} \in Y_n$ has components

$$A_{\mathbf{y}\,ij} = \underline{A_{ij}} + (\overline{A_{ij}} - \underline{A_{ij}})(1 - y_i)/2 \tag{5.6}$$

For an n-dimensional interval vector $[\mathbf{b}]$, whose components are intervals $[b_i] = [\underline{b_i}, \overline{b_i}]$, the corresponding vertex vector $\mathbf{b}_\mathbf{y}$ for each $\mathbf{y} \in Y_n$ has compo-

nents

$$b_{\mathbf{y}i} = \underline{b_i} + (\overline{b_i} - \underline{b_i})(1 + y_i)/2 \tag{5.7}$$

5.2.1.2 Strong Feasibility

Theorem 5.2.1. (Rohn, 2006) A system $[\mathbf{A}]\mathbf{x} = [\mathbf{b}]$ is strongly feasible if and only if for each $\mathbf{y} \in Y_n$ the system

$$\mathbf{A}_\mathbf{y}\mathbf{x} = \mathbf{b}_\mathbf{y} \tag{5.8}$$

has a non-negative solution $\mathbf{x}_\mathbf{y}$. For each $\mathbf{A} \in [\mathbf{A}]$, $\mathbf{b} \in [\mathbf{b}]$, the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ has a solution in the set $\mathrm{Conv}\{\mathbf{x}_\mathbf{y} \mid \mathbf{y} \in Y_n\}$.

Gouttefarde et al. (2011) proposed a technique utilising linear programming (LP) to determine the strong feasibility of Eq. (5.4) for $\underline{\boldsymbol{\tau}} \geq 0$. The system of interval linear equations is strongly feasible if and only if the 2^n systems of linear equations $\mathbf{J}_\mathbf{y}^T\boldsymbol{\tau} = \mathbf{F}_\mathbf{y},\ \mathbf{y} \in Y_n$ are all feasible. The feasibility of a system of linear equations can be tested by means of the first phase of the simplex method applied to the LP problem

$$\begin{aligned} &\min\ \mathbf{0}^T\boldsymbol{\tau} \\ &\text{s.t.} \qquad \mathbf{J}_\mathbf{y}^T\boldsymbol{\tau} = \mathbf{F}_\mathbf{y} \\ &\qquad\quad\ \ \boldsymbol{\tau} \in [\boldsymbol{\tau}] \end{aligned} \tag{5.9}$$

where the objective function is trivial (*i.e.*, always equal to zero) since only feasibility of the system of linear equations is desired. The solution set sat-

isfies

$$\text{Conv}\{\boldsymbol{\tau}_{\mathbf{y}} \mid \mathbf{y} \in Y_n\} \subseteq [\boldsymbol{\tau}] \tag{5.10}$$

Therefore, if each system $\mathbf{J}_{\mathbf{y}}^T \boldsymbol{\tau} = \mathbf{F}_{\mathbf{y}},\ \mathbf{y} \in Y_n$ is feasible via Eq. (5.9), than the mechanism is guaranteed to be able to generate the task wrenches, $[\mathcal{F}_{desired}]$, with the actuator capabilities, $[\boldsymbol{\tau}]$, for $\underline{\boldsymbol{\tau}} \geq 0$.

5.2.1.3 Strong Solvability

Theorem 5.2.2. (Rohn, 2006) A system $[\mathbf{A}]\mathbf{x} = [\mathbf{b}]$ is strongly solvable if and only if for each $\mathbf{y} \in Y_n$ the system

$$\begin{aligned} \mathbf{A}_{\mathbf{y}}\mathbf{x}^1 - \mathbf{A}_{-\mathbf{y}}\mathbf{x}^2 &= \mathbf{b}_{\mathbf{y}} \\ \mathbf{x}^1 \geq 0,\ \mathbf{x}^2 &\geq 0 \end{aligned} \tag{5.11}$$

has a solution $\mathbf{x}_{\mathbf{y}}^1,\ \mathbf{x}_{\mathbf{y}}^2$. For each $\mathbf{A} \in [\mathbf{A}],\ \mathbf{b} \in [\mathbf{b}]$, the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ has a solution in the set $\text{Conv}\{\mathbf{x}_{\mathbf{y}}^1 - \mathbf{x}_{\mathbf{y}}^2 \mid \mathbf{y} \in Y_n\}$.

A LP test can be used to determine the strong solvability of Eq. (5.4) by testing the feasibility of each system of linear equations $\mathbf{J}_{\mathbf{y}}^T\mathbf{x}^1 - \mathbf{J}_{-\mathbf{y}}^T\mathbf{x}^2 = \mathbf{F}_{\mathbf{y}},\ \mathbf{y} \in Y_n$ via the LP problem

$$\begin{aligned} &\min\ \mathbf{0}^T(\mathbf{x}^1 - \mathbf{x}^2) \\ &\text{s.t.} \qquad\qquad \mathbf{J}_{\mathbf{y}}^T\mathbf{x}^1 - \mathbf{J}_{-\mathbf{y}}^T\mathbf{x}^2 = \mathbf{F}_{\mathbf{y}} \\ &\qquad\qquad\quad \mathbf{x}^1 \geq 0,\ \mathbf{x}^2 \geq 0 \\ &\qquad\qquad\quad \mathbf{x}^1 - \mathbf{x}^2 \in [\boldsymbol{\tau}] \end{aligned} \tag{5.12}$$

where $[\boldsymbol{\tau}]$ is not strictly positive and the solution set satisfies

$$\text{Conv}\{\mathbf{x}^1_{\mathbf{y}} - \mathbf{x}^2_{\mathbf{y}} \mid \mathbf{y} \in Y_n\} \subseteq [\boldsymbol{\tau}] \tag{5.13}$$

Therefore, if each system $\mathbf{J}^T_{\mathbf{y}}\mathbf{x}^1 - \mathbf{J}^T_{-\mathbf{y}}\mathbf{x}^2 = \mathbf{F}_{\mathbf{y}},\ \mathbf{y} \in Y_n$ is feasible via Eq. (5.12), than the mechanism is guaranteed to be able to generate the desired wrench capabilities, $[\mathcal{F}_{desired}]$, with the actuator capabilities, $[\boldsymbol{\tau}]$, for $\underline{\boldsymbol{\tau}} < 0$.

5.2.2 Wrench Workspace – Inside Box Classification

By means of the strong solvability and strong feasibility theorems, a pose interval $[\mathbf{p}]$ can be classified as an inside box via Algorithm 4. The functions *Strong_Solvability* and *Strong_Feasibility* return 1 if true, and 0 if false. The algorithm returns 1 if the box is inside and 0 if the box is not inside. A return value of 0 does not necessarily imply an outside box.

5.2.3 Wrench Workspace – Outside Box Classification

The conditions for testing if a pose interval $[\mathbf{p}]$ is fully outside of the wrench workspace are:

$$\exists\ \mathbf{F}_{\mathbf{y}} \in [\mathcal{F}_{desired}] \mid \forall\ \mathbf{J} \in [\mathbf{J}],\ \forall\ \boldsymbol{\tau} \in [\boldsymbol{\tau}],\ \mathbf{J}^T\boldsymbol{\tau} \neq \mathbf{F}_{\mathbf{y}} \tag{5.14}$$

Algorithm 4: Wrench workspace inside box determination

```
function WW_INSIDE ([J], [F_desired], [τ]);
Input  : interval Jacobian [J], interval desired wrench capabilities
         [F_desired], and actuator force workspace [τ]
Output: classification: inside(1), unclassified(0)
forall y ∈ Y_n do
    if τ̲ < 0 then
        if Strong_Solvability (y, [J], [F_desired], [τ]) == 0 then
            return 0
    else
        if Strong_Feasibility (y, [J], [F_desired], [τ]) == 0 then
            return 0
return 1 ;                                  /* Classify as inside box */
```

which implies that some wrench $\mathbf{F_y}$ cannot be generated with admissible actuator capabilities $[\boldsymbol{\tau}]$ for all $\mathbf{p} \in [\mathbf{p}]$, and therefore $[\mathbf{p}]$ must be an outside box.

The condition (5.14) can be tested by applying interval simplification to the system of interval linear equations $[\mathbf{J}]^T\boldsymbol{\tau} = \mathbf{F_y},\ \boldsymbol{\tau} \in [\boldsymbol{\tau}]$ for each vertex vector $\mathbf{F_y}$ of $[\mathcal{F}_{desired}]$ to determine a new box $[\boldsymbol{\tau}]'$, such that $[\boldsymbol{\tau}]' \subseteq [\boldsymbol{\tau}]$. If the simplification returns $[\tau_i]' = \emptyset$ for any of the systems of interval linear equations, the system is inconsistent and (5.14) is true.

A pose interval $[\mathbf{p}]$ can be classified as an outside box via Algorithm 5. For each $\mathbf{F_y} \in [\mathcal{F}_{desired}]$, the unknowns $[\mathbf{u}]$ are populated by $[\boldsymbol{\tau}]$ and the knowns $[\mathbf{k}]$ are populated by $[\mathbf{J}]$ and $\mathbf{F_y}$. A simplification method is applied to the unknowns and knowns, where a return value of -1 indicates that no solution exists for the unknowns, *i.e.*, $[\tau_i]' = \emptyset$. The algorithm returns 1 if the box

is outside and 0 if the box is not outside. A return value of 0 does not necessarily imply an inside box.

Algorithm 5: Wrench workspace outside box determination

```
function WW_OUTSIDE ([J], [F_desired], [τ]);
Input  : interval Jacobian [J], interval desired wrench capabilities
         [F_desired], and actuator force workspace [τ]
Output: classification: outside(1), unclassified(0)
forall F_y ∈ [F_desired] do
    [u] ← [τ];
    [k] ← [J], F_y;
    if SIMPLIFICATION([u], [k]) == −1 then
        return 1 ;                    /* Classify as outside box */
return 0 ;                            /* Cannot be classified */
```

5.2.4 Preconditioning

It is important to obtain a tight representation of $[\mathbf{J}]$ which minimizes the excess width in order to avoid large layers of boundary boxes in the $\mathbb{P}_{WW}$. Transforming an expression into an equivalent expression which leads to better interval evaluation may reduce excess width. Preconditioning can also be applied to (5.4) to transform the system into a preconditioned system which contains the solution set of the original system. The method was derived by Hansen and Smith (1967). Let a preconditioning matrix $\mathbf{P}$ be the inverse of the midpoint of $[\mathbf{J}]^T$. Preconditioning for square systems can be performed by pre-multiplying the preconditioning matrix $\mathbf{P}$ with the original system

such that the preconditioned system is given by

$$\mathbf{P}[\mathbf{J}]^T\boldsymbol{\tau} = \mathbf{P}[\mathcal{F}_{desired}],\ \boldsymbol{\tau} \in [\boldsymbol{\tau}] \tag{5.15}$$

The resulting problem usually has a diagonally dominant matrix, and typically gives tighter bounds to the solutions of the original system (Moore et al., 2009). The improvement is due to the fact that the diagonally dominant matrix is "closer" to the identity matrix and therefore is more appropriate for interval analysis. To further reduce overestimation caused by repeated variables, a symbolic reduction may be applied to the preconditioned system prior to evaluation (Kotlarski et al., 2009).

5.3 Wrench Workspaces of the 3-RPR Parallel Mechanism

The 3-RPR planar parallel mechanism architecture is depicted in Figure 5.1. The actuator capabilities are symmetric (*i.e.*, $\boldsymbol{\tau} = [-\overline{\boldsymbol{\tau}}, \overline{\boldsymbol{\tau}}]$). The prismatic actuators may also model cables to describe CDPMs, such that the actuator capabilities are strictly positive (*i.e.*, $\boldsymbol{\tau} = (0, \overline{\boldsymbol{\tau}}]$).

The points of the fixed base are denoted A_i, whereas B_i denotes the points of the moving platform at which cable i is attached. Point P denotes the position of the end-effector frame and the base frame is defined as $\{O\}$. Vectors $\overrightarrow{OA_i}$, $\overrightarrow{PB_i}$, and $\overrightarrow{OP}$ are denoted as $\mathbf{a}_i$, $\mathbf{b}_i$, and $\mathbf{p}$, respectively. The

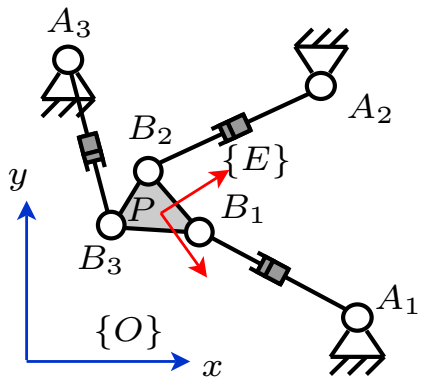


Figure 5.1: Kinematic diagram of the 3-RPR planar parallel mechanism with fixed actuator locations A_1, A_2, A_3.

vector directed along each cable i from B_i to A_i is $\mathbf{l}_i = \mathbf{a}_i - \mathbf{b}_i - \mathbf{p}$. A unit vector corresponding to each cable is represented as $\widehat{\mathbf{d}}_i = \mathbf{l}_i/\rho_i$, where ρ_i denotes the length of cable i. If a CDPM contains only three limbs, then it is necessary for the moving platform to be modelled as a point (*i.e.*, $\mathbf{b}_i = \mathbf{0}$, for $i = 1, \ldots, 3$).

The formulation of the Jacobians for the 3-RPR is given in Appendix E.

5.3.1 Methods for Exact Designs

The moving platform is considered as a point-mass to model a CDPM. That is, all cables are attached to a common point on the moving platform, such that $\mathbf{b}_i = \mathbf{0}$ for each limb. The following exact design $\mathcal{D}$ is considered, such that the fixed actuator locations form an equilateral triangle with edge

lengths of 1 m:

$$\mathcal{D} = \left\{ \begin{array}{l} \mathbf{a}_1 = (\sqrt{3}/3, 0)^T \text{ m}; \\ \mathbf{a}_2 = (\sqrt{3}/3\cos(2\pi/3), \sqrt{3}/3\sin(2\pi/3))^T \text{ m}; \\ \mathbf{a}_3 = (\sqrt{3}/3\cos(4\pi/3), \sqrt{3}/3\sin(4\pi/3))^T \text{ m}; \\ \mathbf{b}_i = (0.0, 0.0)^T \text{ m for } i = 1, \ldots, 3; \\ \rho_i = [0, 2] \text{ m for } i = 1, \ldots, 3; \\ \tau_i = [10, 90] \text{ Nm for } i = 1, \ldots, 3. \end{array} \right\} \tag{5.16}$$

The desired wrench requirement $\mathcal{F}_{desired}$ is selected as a) a box $[f_x] = [f_y] = [-20, 20]$ N ($\mathcal{F}_{box}$) and b) a ball (hyper-ellipse) with semi-principle-axes lengths $r_{f_x} = r_{f_y} = 20$ N ($\mathcal{F}_{ball}$). The allowable cable tensions are selected as $\tau_i \in [10, 90]$ N. These parameters are chosen for a comparison of similar work by Kaloorazi et al. (2013). Using a radial discretization of 1000 evenly spaced rays about the origin, the approximation of the boundaries of the associated wrench workspaces for $\mathcal{F}_{box}$ and $\mathcal{F}_{ball}$ are shown in Figure 5.2. The plotted pose is inside the wrench workspaces. It can be noted that since $\mathcal{F}_{ball}$ is a subset of $\mathcal{F}_{box}$, that the wrench workspace $\mathbb{P}_{WW\,box}$ is a subset of $\mathbb{P}_{WW\,ball}$.

5.3.2 Interval Methods

The inverse kinematics constraint equations may be implemented for each limb i as follows

$$\begin{aligned} \rho_i &= ||\mathbf{a}_i - \mathbf{b}_i - \mathbf{p}|| \\ \mathbf{d}_i &= (\mathbf{a}_i - \mathbf{b}_i - \mathbf{p})/\rho_i \end{aligned} \tag{5.17}$$

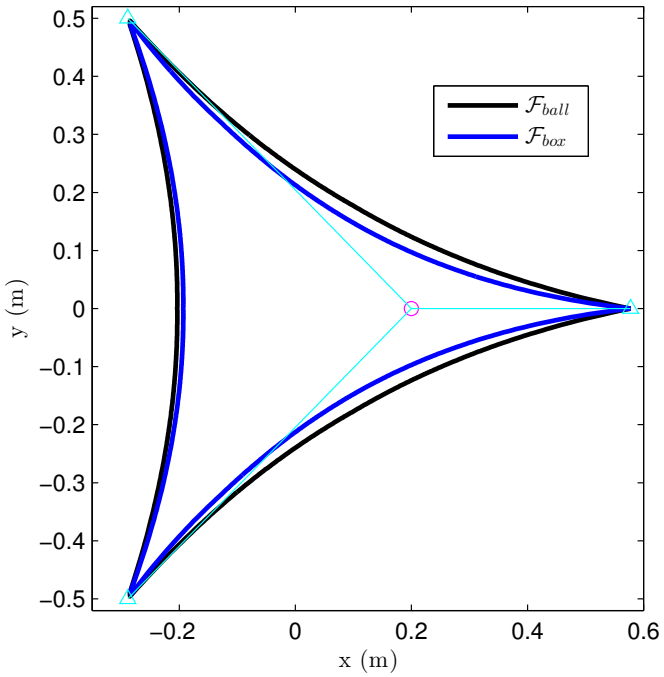


Figure 5.2: Wrench workspace of the CDPM with 2-DOF and 3 cables.

The interval evaluation of ρ_i is tight; however, the evaluation of $\mathbf{d}_i$ will be overestimated. Since $\mathbf{d}_i$ should be a unit vector, $||\mathbf{d}_i|| = 1$ can be added as an additional constraint. Gouttefarde et al. (2011) proposed a modification to the Jacobian matrix to deal with the dependency problem in $\mathbf{J}$. The modification is not necessary here since the unit constraints on $\mathbf{d}_i$ are quite effective. Restrictions on the length of the prismatic actuator (cable) can be accounted for with the constraint $\rho_i \subseteq [\underline{\rho_i}, \overline{\rho_i}]$.

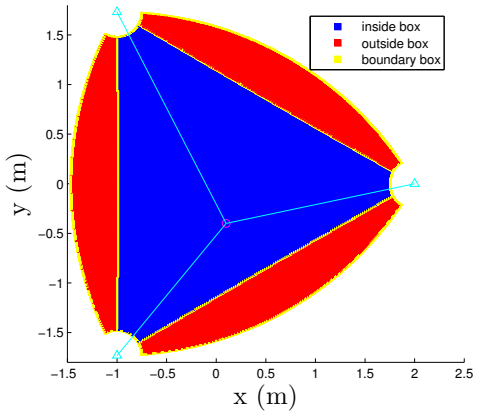
A set of exact design parameters are selected as

$$\mathcal{D} = \left\{ \begin{array}{l} \mathbf{a}_1 = (2,0)^T \text{m}; \\ \mathbf{a}_2 = (2\cos(2\pi/3), 2\sin(2\pi/3))^T \text{m}; \\ \mathbf{a}_3 = (2\cos(4\pi/3), 2\sin(4\pi/3))^T \text{m}; \\ \mathbf{b}_i = (0.0, 0.0)^T \text{m for } i = 1, \ldots, 3; \\ \rho_i = [0.25, 2\sqrt{3}] \text{m for } i = 1, \ldots, 3; \\ \tau_i = [0.1, 10] \text{Nm for } i = 1, \ldots, 3. \end{array} \right\} \tag{5.18}$$

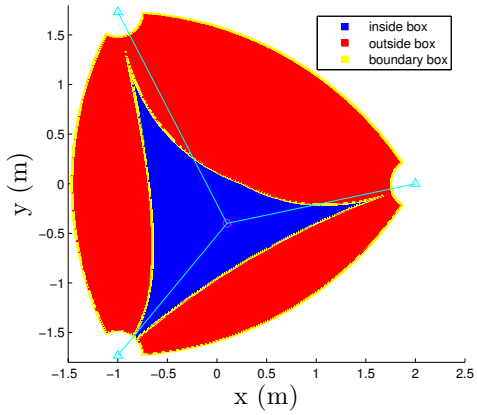
The wrench workspace associated with a zero wrench requirement ($[\mathcal{F}_{desired}] = ([f_x], [f_y])^T = ([0,0]\ \text{N}, [0,0]\ \text{N})^T$) with a bisection resolution of $\beta = 0.01$ m is provided in Figure 5.3a. The end-effector is able to reach any pose with an inside classification while ensuring that the cable tension and length restrictions are satisfied. The wrench workspace with a wrench requirement $[\mathcal{F}_{desired}] = ([f_x], [f_y])^T = ([-2,2]\ \text{N}, [-1,8]\ \text{N})^T$ and bisection resolution of $\beta = 0.01$ m is shown in Figure 5.3b. An appropriate design may also be considered without modification.

5.4 Wrench Workspaces of the 3-$\underline{\text{R}}$RR Parallel Mechanism

The initial search space, $\mathbb{P}$, is initialized to overestimate the reachable workspace of the 3-$\underline{\text{R}}$RR. A desired orientation of $\psi = [1.72908]$ is selected unless otherwise specified. The desired task is described by the box (in-



(a) Wrench workspace of the CDPM associated with a zero wrench.



(b) Wrench workspace of the CDPM associated with $[\mathcal{F}_{desired}] = ([f_x], [f_y])^T = ([-2, 2]\ \text{N}, [-1, 8]\ \text{N})^T$.

Figure 5.3: Wrench workspaces of the CDPM with 2-DOF and 3 cables using interval methods.

terval) $[\mathcal{F}_{desired}] = ([f_x], [f_y], [m_z])^T = ([-30, 30]\ \text{N}, [-30, 30]\ \text{N}, [0, 0]\ \text{Nm})^T$. Actuator capabilities of $[\tau_i] = [-10, 10]$ Nm are selected for each actuator.

5.4.1 Methods for Exact Designs

The wrench workspace may be obtained for an exact design by using a grid discretization with a resolution of $\epsilon = 0.005$ m. Radial discretization is not applicable for this mechanism since the wrench workspace may contain voids. Considering the exact design described in (3.16), the associated wrench workspace for a given orientation is provided in Figure 5.4. The blue grid points are reachable and satisfy the task wrench requirements (wrench-capable). The red grid points are reachable and dissatisfy the task wrench

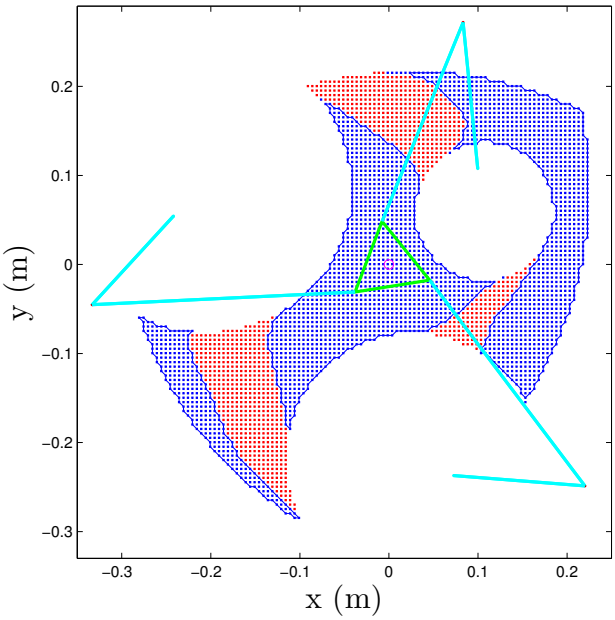


Figure 5.4: Wrench workspace of the 3-RRR parallel mechanism with exact design described in equation (3.16) and orientation $\psi = 1.72908$.

requirements.

The α-shape routine is applied with an α-radius of 0.005 m to obtain the boundaries of wrench-capable poses. Three separated regions are found; these regions are isolated in Figure 5.5. It is important to note that the wrench workspace obtained in Figure 5.5 is an approximation. There is no guarantee that the poses inside the enclosed regions are all wrench-capable.

5.4.2 Interval Methods

The wrench workspace may be obtained with interval methods by applying a branch-and-bound routine to $\mathbb{P}$ with a bisection resolution of $\beta = 0.001$ m.

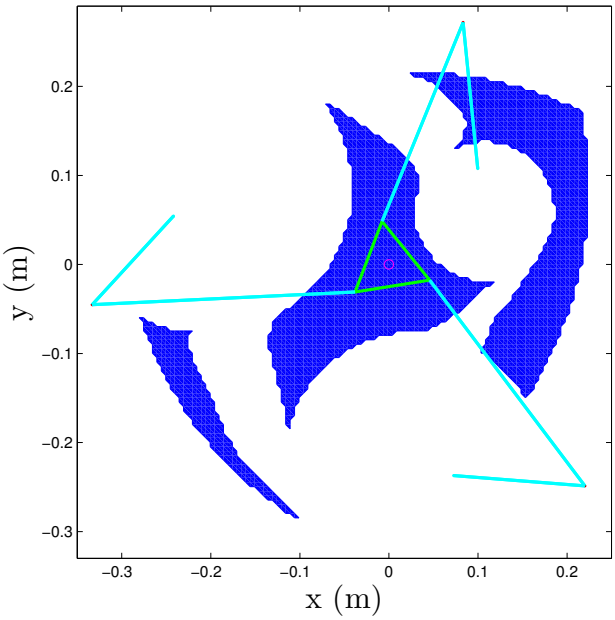


Figure 5.5: Separated regions of the wrench workspace of the 3-RRR parallel mechanism obtained with the α-shape routine.

Considering the exact design described in (3.16), the associated wrench workspace for a given orientation is provided in Figure 5.6. The blue regions are guaranteed to be reachable and satisfy the task wrench requirements. The red regions are guaranteed to be reachable and dissatisfy the task wrench requirements. The yellow regions require additional bisections in order to obtain a classification. For clarity, the edges of the boxes have been removed and the unreachable pose intervals have been neglected from the plot.

Considering the appropriate design described in (4.12), the associated wrench workspace is provided in Figure 5.7. The blue regions are guaranteed to be

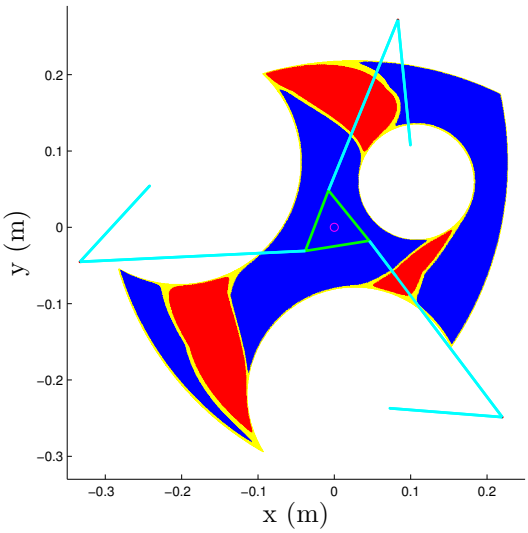


Figure 5.6: Wrench workspace, obtained with interval methods, of the 3-RRR parallel mechanism with exact design described in equation (3.16) and orientation $[\psi] = [1.72908]$.

reachable and satisfy the task wrench requirements for every design $\mathcal{D} \in [\mathcal{D}]$. The red regions are guaranteed to be reachable and dissatisfy the task wrench requirements for at least one design $\mathcal{D} \in [\mathcal{D}]$. Again, the yellow regions require additional bisections in order to obtain a classification. The orientation $[\psi] = [0.0]$ was selected because of the large number of resulting unclassified pose intervals (yellow regions).

The thickness of the boundary box layer in the $\mathbb{P}_{WW}$ of Figure 5.7 is attributed to: the overestimation present in $[\mathbf{J}]$, and the nonsatisfaction of the outside test for the appropriate design. To aid in reducing the overestimation present in $[\mathbf{J}]$, additional constraints relating to the reciprocal screws of each limb are added to the constraint system. The direct Jacobian is formulated

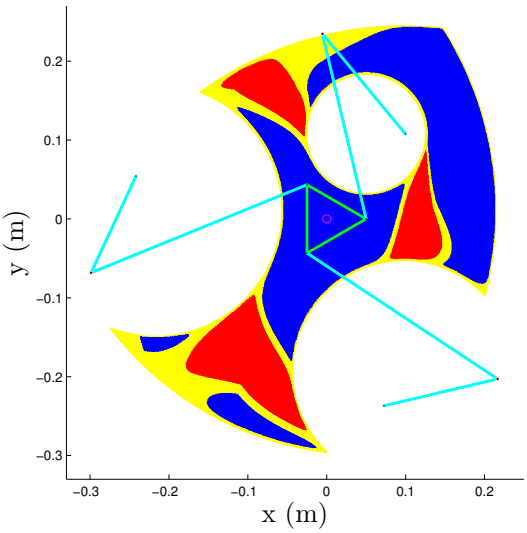


Figure 5.7: Wrench workspace of the 3-RRR parallel mechanism with appropriate design described in equation (4.12) and orientation $[\psi] = [0.0]$.

from the unit reciprocal screws for each limb (see (Tsai, 1999)). Since each reciprocal screw must have unit length, the following constraints are added to the constraint system to provide simplification to the interval solutions of the unit reciprocal screws. Let $\widehat{\mathbf{g}}_i$ denote the unit vector along the distal link of limb i. The additional constraints for each limb i are

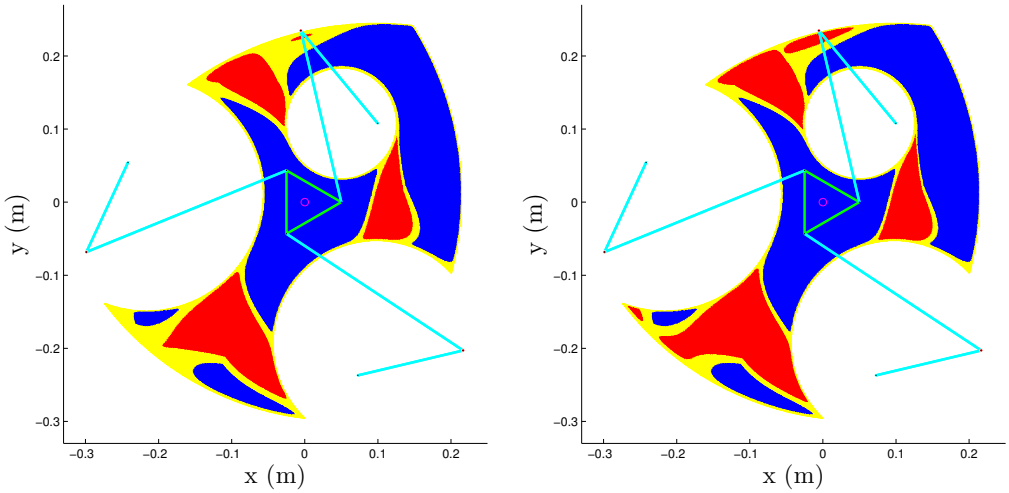
$$\begin{aligned} &0 \notin det([\mathbf{J_x}([\mathcal{D}], [\mathbf{p}])]) \text{ and} \\ &0 \notin det([\mathbf{J_q}([\mathcal{D}], [\mathbf{p}])]) \text{ and} \\ &0 \notin det([\mathbf{J}([\mathcal{D}], [\mathbf{p}])]) \end{aligned} \tag{5.19}$$

$$
\begin{aligned}
g_{ix} &= ((C_{xi} - A_{xi}) - r_i \cos(\alpha_i))/l_i; \\
g_{iy} &= ((C_{yi} - A_{yi}) - r_i \sin(\alpha_i))/l_i; \\
{g_{ix}}^2 + {g_{iy}}^2 &= 1;
\end{aligned}
\tag{5.20}
$$

The wrench workspace resulting from the improved Jacobian evaluation is provided in Figure 5.8a. Compared to Figure 5.7, the improved Jacobian evaluation provides a slight reduction in the number of boundary boxes.

The set of inside pose intervals for an appropriate design $[\mathcal{D}]$ is equal to the intersection of the set of inside pose intervals for each exact design $\mathcal{D} \in [\mathcal{D}]$. Alternatively, the set of outside pose intervals for an appropriate design $[\mathcal{D}]$ is equal to the union of the set of outside pose intervals for each exact design $\mathcal{D} \in [\mathcal{D}]$. This implies that the outside test may be improved by additionally testing sample designs $\mathcal{D}' \in [\mathcal{D}]$. If a sample design satisfies the outside test over the pose interval, then the appropriate design cannot satisfy the inside test and the pose interval may be correctly classified as *outside*. The wrench workspace resulting from the improved Jacobian evaluation and improved outside test is provided in Figure 5.8b. The improved outside test checks the samples: $\mathcal{D}'_{\min} = \underline{[\mathcal{D}]}$, $\mathcal{D}'_{mid} = mid([\mathcal{D}])$, and $\mathcal{D}'_{\max} = \overline{[\mathcal{D}]}$. Compared to Figure 5.7, the improved Jacobian evaluation combined with the design samples for the outside test provides a significant reduction in the number of boundary boxes.

The wrench workspace determination of parallel mechanisms described by an appropriate design is another useful tool for designers. Accounting for



(a) Utilising improvements to the Jacobian evaluation.

(b) Utilising improvements to the Jacobian and outside test.

Figure 5.8: Wrench workspaces for the 3-RRR parallel mechanism.

the uncertainties present in the mechanism, designers are able to guarantee the wrench capabilities of an actual mechanism. The wrench workspace algorithms developed here are applied in Chapters 6 and 7 for the synthesis of parallel mechanisms.

Chapter 6

Appropriate Synthesis: The Four-Bar Linkage

The goal of *appropriate synthesis* is to determine the complete set of design solutions for a mechanism which satisfy some desired task. An exact description of the desired task is not required. Instead, the desired task may be described by specifying the allowable errors on the various criteria.

Consider the following example. It may be desired to synthesize a four-bar linkage which is able to follow a straight-line trajectory, but rarely is the "straight" criteria strictly necessary. Therefore, an allowable error may be specified such that the goal of the linkage is to approximate the straight-line trajectory without exceeding the allowable error. Appropriate synthesis applied to the four-bar linkage will return every combination of appropriate design parameters which satisfy the desired trajectory.

Since the objective is to determine the set of design solutions which achieve some desired performance criteria, it is only necessary to determine if the performance criteria are exceeded. That is, the performance of a mechanism does not need to be evaluated tightly (*i.e.*, with minimal excess width). If the worst-case performance of a design is better than the desired performance, then the design is considered to be a solution to the problem.

The method of appropriate synthesis is introduced by formulating the necessary routines to solve the four-bar linkage problem. For a desired task, described by any number of precision points and trajectories, each with allowable errors, appropriate synthesis returns the corresponding set of appropriate design solutions for the four-bar linkage. This set of appropriate design solutions describes every possible design solution to the problem, for a given resolution. Conventional synthesis methods, on the other hand, are usually only able to sample the design space to provide a single solution or possibly a few different solutions, and these solutions are not reliable under uncertainties. Thus, appropriate synthesis provides an effective new tool to mechanism designers.

6.1 The Four-bar Linkage Description

The four-bar linkage is a planar mechanism consisting of four rigid members: the frame, the input link, the output link, and the coupler link. These members are connected by four revolute pairs forming a closed-loop kinematic

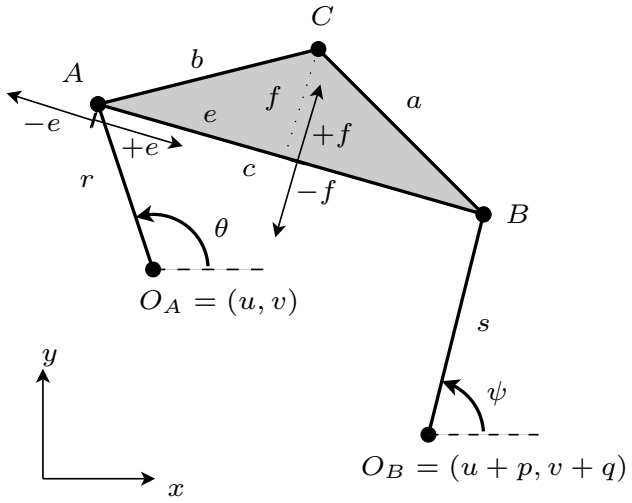


Figure 6.1: Four-bar linkage description.

chain of 1-DOF. A point on the coupler, known as the *coupler point*, traces a path as the input link is rotated. This path is known as the *coupler curve*. The four-bar linkage and its associated design parameters are provided in Figure 6.1. The fixed base location O_A is located at coordinates (u, v) with respect to the reference frame. The location O_B is located at coordinates (p, q) with respect to O_A or $(u+p, v+q)$ with respect to the reference frame. Link $O_A A$ will always be assumed to be the input link with an input angle θ (measured relative to the x axis) and length r. Link $O_B B$ is the output link, which has an output angle ψ (measured relative to the x axis) and length s. The coupler point is denoted $C = (C_x, C_y)$, where the coupler link is triangle ABC with edge lengths a, b, and c. The parameters e and f are used to describe the location of C relative to segment AB.

Useful coupler curves can be generated when the geometric parameters of

the linkages are properly selected. The process of selecting these parameters, based on some desired response, is known as dimensional synthesis. Two aspects of dimensional synthesis can be described, referred to as *exact synthesis* and *approximate synthesis*. These are subsets of the synthesis problem for exact designs, as described in Table 1.1.

Exact synthesis of four-bar linkages involves obtaining geometric parameters which exactly produce a coupler curve. Generally a curve is approximated by a set of *precision points* (desired poses for the linkage to achieve). Analytical approaches are able to solve the problem and return the exact solutions for a four-bar linkage. It is well known due to Roberts (1875) that every four-bar linkage has two four-bar cognates, so at least three design solutions are always expected. It has been proven that the four-bar linkage can guide the coupler point exactly through no more than five arbitrary crank-coordinated precision points, and no more than nine arbitrary precision points (Hunt, 1978). Wampler et al. (1992) were able to obtain the complete solution for the nine precision point problem. Bai and Angeles (2015) recently developed a new formulation of the synthesis problem to exactly determine the geometric parameters corresponding to a given algebraic coupler curve equation.

For approximate synthesis, numerical methods are used to synthesize a linkage to perform within an acceptable deviation from a desired curve. The curve may be approximated by a set of precision points, whereby numerical methods attempt to find designs which minimize the error between the coupler curve and the precision points. Global optimization methods such

as genetic algorithms (Cabrera et al., 2002) and other stochastic or heuristic methods (Bulatović and Dordević, 2009) have been used to find solutions to the synthesis problem by minimizing an objective function, such as to minimize the error between the coupler curve and the precision points. However, these methods are unable to guarantee global optimality and often suffer from premature convergence, yielding non-optimal design solutions. Recently, Goulet et al. (2016) presented a synthesis method using a non-convex optimization routine which relies on a solver which applies constraint propagation and interval arithmetic, and requires the use of precision points approximating the desired curve. A more thorough exploration of the design parameter space may be achieved with this method, although their method suffers from the approximation of the desired trajectory. Other techniques such as machine learning and neural network approaches (Unruh and Krishnaswami, 1995, Coros et al., 2013) rely on a database of linkage solutions (*e.g.*, the coupler curve atlas (Hrones and Nelson, 1951, Mullineux, 2011)). Such techniques suffer from the coverage of mechanisms in the database and sampling techniques.

The performance of a linkage cannot be guaranteed using conventional synthesis techniques, as these techniques are unable to account for the uncertainties inherent in the mechanism. The issue of uncertainties plays a significant role in the actual performance of a mechanism. Error in the crank angle results in a translational and rotational error in the coupler point (Chatterjee and Mallik, 1987). In the four-bar linkage, uncertainties make it no longer

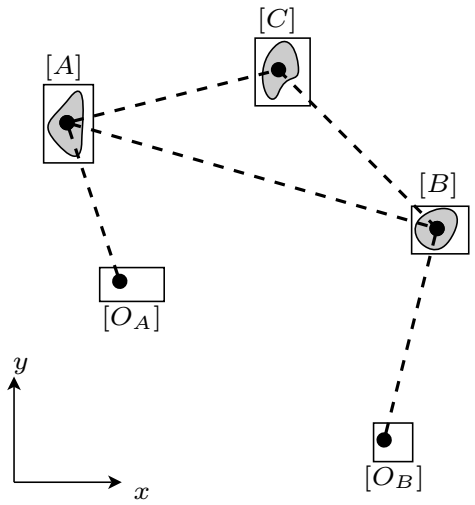


Figure 6.2: The effect of geometrical uncertainties in the four-bar linkage.

possible to obtain an exact solution for the coupler point and thus exact synthesis methods are not applicable. Instead, the coupler point can only be determined to lie within some bounds, where the bounds are functions of the uncertainties. For instance, Figure 6.2 depicts the effect of uncertainties on the linkage. The fixed locations O_A and O_B are known to lie in the corresponding boxes $[O_A]$ and $[O_B]$, respectively. For a given input and output angle, the locations A, B, and C will lie in some domain. With interval analysis, these domains are approximated by boxes $[A]$, $[B]$, and $[C]$, respectively. The coupler point will always be inside box $[C]$ for a given input and output angle.

It would be beneficial to account for the uncertainties during synthesis, such that the desired performance of a synthesized mechanism is guaranteed. Moreover, it would be useful to obtain the set of appropriate designs which

satisfy the desired criteria. The end-user would then be free to select the design which best suit their needs. Appropriate synthesis provides the set of appropriate designs.

6.2 Kinematic Equations

The equations describing the kinematics of the four-bar linkage can be formulated by solving a set of distance equations (f_1 through f_5). The vertex $A = (A_x, A_y)$ of the coupler triangle lies on the circle centred at O_A with radius r. The vertex $B = (B_x, B_y)$ of the coupler triangle lies on the circle centred at O_B with radius s. The points A and B are separated by distance c. The points A and B are functions of the design parameters and input and output angles θ and ψ (f_6 through f_9). The edge lengths a and b of the coupler link are functions of the design parameters c, e, and f (f_{10} and f_{11}). Let g be the distance between points O_A and O_B (f_{12}). A unique configuration of the platform is enforced by adding constraints f_{13} and f_{14} which take into account the signs of e and f.

$$
\begin{aligned}
f_1 &:= ||O_A A||^2 = (u - A_x)^2 + (v - A_y)^2 = r^2 \\
f_2 &:= ||O_B B||^2 = ((p + u) - B_x)^2 + ((q + v) - B_y)^2 = s^2 \\
f_3 &:= ||AB||^2 = (A_x - B_x)^2 + (A_y - B_y)^2 = c^2 \\
f_4 &:= ||AC||^2 = (A_x - C_x)^2 + (A_y - C_y)^2 = b^2 \\
f_5 &:= ||BC||^2 = (B_x - C_x)^2 + (B_y - C_y)^2 = a^2 \\
f_6 &:= A_x = u + r\cos(\theta) \\
f_7 &:= A_y = v + r\sin(\theta) \\
f_8 &:= B_x = (p + u) + s\cos(\psi) \\
f_9 &:= B_y = (q + v) + s\sin(\psi) \\
f_{10} &:= b = \sqrt{e^2 + f^2} \\
f_{11} &:= a = \sqrt{(c - e)^2 + f^2} \\
f_{12} &:= g = \sqrt{p^2 + q^2} \\
f_{13} &:= C_x = A_x + 1/c\,((B_x - A_x)e + (A_y - B_y)f) \\
f_{14} &:= C_y = A_y + 1/c\,((B_y - A_y)e + (A_x - B_x)f)
\end{aligned}
\tag{6.1}
$$

The equations f_1 to f_{11} in (6.1) may be simplified leaving a system of three equations with 11 parameters. That is,

$$
\begin{aligned}
||AB||^2 &= (r\cos(\theta) - p - s\cos(\psi)))^2 + (r\sin(\theta) - q - s\sin(\psi))^2 = c^2 \\
||AC||^2 &= (u + r\cos(\theta) - C_x)^2 + (v + r\sin(\theta) - C_y)^2 = e^2 + f^2 \\
||BC||^2 &= (p + u + s\cos(\psi) - C_x)^2 + (q + v + s\sin(\psi) - C_y)^2 = (c - e)^2 + f^2
\end{aligned}
\tag{6.2}
$$

This results in a more compact description of the system, which is typically used by numerical methods for approximate synthesis. However, the issue is that this compact form is not ideal for interval analysis since there are repetitions of variables in each equation. As such, the dependency problem with interval analysis will overestimate the solution to the system. Therefore, it is preferable to model the system as (6.1) since the equations are simpler. This is because interval analysis methods will be more effective, as interval analysis requires a trade-off between the number of equations and their complexity. The Jacobian and Hessian matrices generated from equations in (6.1) are notably simpler.

An appropriate design is described by $[\mathcal{D}]$, whereas an exact design is described by $\mathcal{D}$. The interval domains of the appropriate design parameters account for uncertainties such that:

$$[\mathcal{D}] = \{[u], [v], [p], [q], [r], [s], [c], [e], [f]\} \tag{6.3}$$

The known manufacturing uncertainties for the design parameters may be described by the vector $\Delta\mathcal{D}$ as

$$\Delta\mathcal{D} = (\Delta u, \Delta v, \Delta p, \Delta q, \Delta r, \Delta s, \Delta c, \Delta e, \Delta f)^T \tag{6.4}$$

If an exact design $\mathcal{D}$ is considered, then the appropriate design parameters

which account for manufacturing uncertainties will be given by

$$[\mathcal{D}] = \mathcal{D} \pm \Delta\mathcal{D} \tag{6.5}$$

An appropriate synthesis routine will return the complete set of appropriate designs $[\mathcal{D}]$ which satisfy the given objective.

6.3 Linkage Classifications

Four-bar linkages can be sorted into two categories, *Grashof-type* and *non-Grashof-type* linkages. Grashof linkages satisfy the the Grashof condition, which states that if the sum of the shortest and longest link is less than or equal to the sum of the remaining two links, then the shortest link can rotate fully with respect to a neighbouring link. Whereas, the links of a non-Grashof linkage cannot rotate fully. Grashof-type linkages and non-Grashof-type linkages each have four different classifications which describe the ranges of the input and output angles. The term *crank* denotes a link which is able to rotate fully, while the term *rocker* denotes a link that cannot rotate fully. A *double-crank/double-rocker* is a linkage whereby both the input and output links are cranks/rockers. For non-Grashof linkages, the input and output links will rock through 0 or π. McCarthy and Song Soh (2010) describe a routine for classifying a four-bar linkage based only on the length of the links. Their routine may be extended to accept interval

design parameters. Three parameters may be used to identify the possible classifications, $[T_i]$ for $i = 1, \ldots, 3$, and are evaluated as

$$\begin{aligned} [T_1] &= [g] - [r] + [c] - [s]; \\ [T_2] &= [g] - [r] - [c] + [s]; \\ [T_3] &= -[g] - [r] + [c] + [s]; \end{aligned} \tag{6.6}$$

The classification of the linkage is determined from the constant sign of the $[T_i]$ parameters as

Table 6.1: Classifications of four-bar linkages with an appropriate design.

sgn($[T_1]$)	sgn($[T_2]$)	sgn($[T_3]$)	Classification	Type
+	+	+	crank-rocker	Grashof
+	-	-	rocker-crank	Grashof
-	-	+	double-crank	Grashof
-	+	-	double-rocker	Grashof
-	-	-	00-double-rocker	Non-Grashof
+	+	-	0π-double-rocker	Non-Grashof
+	-	+	$\pi 0$-double-rocker	Non-Grashof
-	+	+	$\pi\pi$-double-rocker	Non-Grashof

A folding linkage occurs when any one of the parameters $[T_i]$ includes zero (non-constant sign). Such a linkage may be able to take on a configuration where points O_A, O_B, A and B lie on a line. This is nearly impossible to achieve physically without flexible links or play in the joints, as the uncertainties prevent such a scenario. For this reason, folding linkages will not be considered as appropriate design solutions.

A given appropriate design $[\mathcal{D}]$ may have multiple classifications, depending

on if any of the parameters $[T_i]$ includes zero. Let the list $\mathcal{L}_{class}$ contain the possible classifications associated with the design. In certain situations, it may be desirable to consider particular classifications of a linkage during synthesis. The list $\mathcal{L}_{allow\_class}$ will contain all allowable classifications. A routine which considers the allowed and actual classifications may be used as an additional simplification routine, to remove any appropriate designs which result in only classifications which are not allowed, and to identify if an appropriate design results in a single allowed classification. A routine CLASSIFICATION($[\mathcal{D}], \mathcal{L}_{class}, \mathcal{L}_{allow\_class}$) updates the actual classifications $\mathcal{L}_{class}$ for the appropriate design $[\mathcal{D}]$ and returns:

-1 : the allowed classifications are not contained in $\mathcal{L}_{class}$;

1 : there is a unique allowable classification in $\mathcal{L}_{class}$;

0 : there is more than one classification in $\mathcal{L}_{class}$.

6.4 Branches and Circuits

The description for circuits and branches are adopted from Chase and Mirth (1993). For a given assembly of a four-bar linkage, the coupler point will follow what is referred to as a *circuit*. To change the circuit being followed, the linkage would need to be disassembled and reassembled. The term *toggle position* describes collinearity of the coupler and output links. At a toggle position, the linkage is able to change its *branch* (a branch is defined by a transmission angle, the angle between the coupler and output links, in the range of $(0, \pi)$ or $(-\pi, 0)$). For the purpose of appropriate synthesis, a linkage

with an appropriate design $[\mathcal{D}]$ must ensure that the desired path is accomplished by a single circuit. This concept is referred to as an *assembly mode defect*. It may or may not be necessary to restrict the branch of the linkage, as some applications benefit from a change in branch, so this restriction is considered to be optional.

The number of circuits corresponding to a particular classification of linkage are described in (Schröcker et al., 2006). The ranges of the input and output angles corresponding to a particular classification are provided in (McCarthy and Song Soh, 2010). The branch and circuit conditions are extended for appropriate designs as follows. Note that the vector $O_A O_B$ is used as a reference for the angles in (McCarthy and Song Soh, 2010). Here, the angles must be shifted such that for an appropriate design the shifted angles $[\theta']$ and $[\psi']$ are

$$\begin{aligned} [\theta'] &= [\theta] - \text{atan2}([q],[p]) \\ [\psi'] &= [\psi] - \text{atan2}([q],[p]) \end{aligned} \tag{6.7}$$

where, atan2 is the quadrant-corrected inverse tangent.

The limits on the shifted angles $[\theta']$ and $[\psi']$ are determined as follows

$$\begin{aligned} [\theta'_{\text{min}}] &= \text{acos}((g^2+r^2)-(c-s)^2)/(2rg) \\ [\theta'_{\text{max}}] &= \text{acos}((g^2+r^2)-(c+s)^2)/(2rg) \\ [\psi'_{\text{min}}] &= \text{acos}((c+r)^2-(g^2+s^2))/(2sg) \\ [\psi'_{\text{max}}] &= \text{acos}((c-r)^2-(g^2+s^2))/(2sg) \end{aligned} \tag{6.8}$$

The circuits are identified as follows

- *crank-rocker* – generates 2 circuits, each with a unique branch: The input link can fully rotate. The output link has two operating ranges.

  1. $\theta' \in [0, 2\pi]$, $[\psi'_{\min}] \leq \psi' \leq [\psi'_{\max}]$
  2. $\theta' \in [0, 2\pi]$, $-[\psi'_{\max}] \leq \psi' \leq -[\psi'_{\min}]$

- *rocker-crank* – generates 2 circuits, each with 2 branches: The output link can fully rotate. The input link has two operating ranges.

  1. $\psi' \in [0, 2\pi]$, $[\theta'_{\min}] \leq \theta' \leq [\theta'_{\max}]$
  2. $\psi' \in [0, 2\pi]$, $-[\theta'_{\max}] \leq \theta' \leq -[\theta'_{\min}]$

- *double-crank* – generates 2 circuits, each with a unique branch: the input link and output link can both fully rotate.

  1. $\theta' \in [0, 2\pi]$, $\psi' \in [0, 2\pi]$, $(BA \times BO_B)_z < 0$
  2. $\theta' \in [0, 2\pi]$, $\psi' \in [0, 2\pi]$, $(BA \times BO_B)_z > 0$

- *double-rocker* – generates 2 circuits, each with 2 branches: the input link and output link both rock. The input link has two operating ranges.

  1. $[\theta'_{\min}] \leq \theta' \leq [\theta'_{\max}]$, $[\psi'_{\min}] \leq \psi' \leq [\psi'_{\max}]$
  2. $-[\theta'_{\max}] \leq \theta' \leq -[\theta'_{\min}]$, $-[\psi'_{\max}] \leq \psi' \leq -[\psi'_{\min}]$

- 00-*double-rocker* – generates 1 circuit with 2 branches: the input link will rock between $-[\theta'_{\max}] \leq \theta' \leq [\theta'_{\max}]$ and the output link will rock between $-[\psi'_{\max}] \leq \psi' \leq [\psi'_{\max}]$.

- 0π-*double-rocker* – generates 1 circuit with 2 branches: the input link will rock between $-[\theta'_{\max}] \leq \theta' \leq [\theta'_{\max}]$ and the output link will rock between $[\psi'_{\min}] \leq \psi' \leq 2\pi - [\psi'_{\min}]$.

- $\pi 0$-*double-rocker* – generates 1 circuit with 2 branches: the input link will rock between $[\theta'_{\min}] \leq \theta' \leq 2\pi - [\theta'_{\min}]$ and the output link will rock between $-[\psi'_{\max}] \leq \psi' \leq [\psi'_{\max}]$.

- $\pi\pi$-*double-rocker* – generates 1 circuit with 2 branches: the input link will rock between $[\theta'_{\min}] \leq \theta' \leq 2\pi - [\theta'_{\min}]$ and the output link will rock between $[\psi'_{\min}] \leq \psi' \leq 2\pi - [\psi'_{\min}]$.

To determine the circuit, it is also necessary to know which side of the line AB the coupler point is on. That is, $[f] \geq 0$ corresponds to one circuit, whereas $[f] < 0$ corresponds to another circuit. Therefore, depending on the classification of the linkage, there may be at most 2 circuits and each circuit may have at most 2 branches.

A given appropriate design $[\mathcal{D}]$ and coupler point $[C]$ may result in multiple assemblies (circuits and branches). It is necessary with appropriate synthesis that certain restrictions be applied to the assemblies. A desired coupler curve can only be achieved with a single circuit, and in some cases, a change of branch may or may not be allowed.

An assembly filtering routine is proposed. This routine is used as an additional simplification method in order to remove appropriate designs which cannot satisfy the assembly conditions. Let the list $\mathcal{L}_{branches}$ denote the list of associated branches and the list $\mathcal{L}_{circuits}$ denote the list of associated circuits for a given appropriate design $[\mathcal{D}]$ and coupler point $[C]$. A flag SINGLE_BRANCH identifies whether a desired coupler curve may be generated by one or two branches. A single circuit is always enforced. Allow the desired coupler points be described by the set of unknowns $\mathbb{U}$, and allow the knowns to be $[\mathbf{k}]$. Also, allow the classifications, branches, and circuits associated with each coupler point to be described by elements in the sets $\{\mathcal{L}_{class}\}$, $\{\mathcal{L}_{branches}\}$, and $\{\mathcal{L}_{circuits}\}$ respectively. The routine ASSEMBLY$(\mathbb{U}, [\mathbf{k}], \{\mathcal{L}_{class}\}, \{\mathcal{L}_{branches}\}, \{\mathcal{L}_{circuits}\})$ considers two or more coupler points and the assemblies associated with these points. The CLASSIFICATION routine is called inside this function. The ASSEMBLY routine finds the intersection of the classifications, branches, and circuits, and updates each list by the intersection. Depending on the flag SINGLE_BRANCH, branches may be ignored. The following is returned by the ASSEMBLY routine:

-1 : the intersection of all assemblies is empty;

1 : every coupler point satisfies the circuit and branch conditions (unique circuit with one of two branches);

0 : there is more than one possible assembly.

6.5 Solving the Kinematics

To solve the kinematics of the four-bar linkage, two main problems may be considered for appropriate analysis:

1. **Forward problem**: *Determine a solution for the coupler point* $([C_x], [C_y])$ *for a fixed value for* $[\theta]$ *and an allowable range for* $[\psi]$. For this problem, a fixed value for $[\theta]$ is given and $[\psi]$ is given as an allowable range. The goal is to compute the solution/solutions for $([C_x], [C_y])$ and $[\psi]$ which correspond to the fixed value of $[\theta]$.

2. **Interior problem**: *Determine a solution for* $[\theta]$ *and* $[\psi]$ *and coupler point* $([C_x], [C_y])$ *from the interior of the box* $([x], [y])$. For this problem, a box $([x], [y])$ is given and $[\theta]$ and $[\psi]$ are given as allowable ranges. The goal is to compute a solution for $[\theta]$, $[\psi]$ and $([C_x], [C_y])$ such that $([C_x], [C_y]) \subset ([x], [y])$.

The forward problem is useful for computing the coupler curve associated with an appropriate design. The interior problem is useful for determining if an appropriate design satisfies desired coupler curve characteristics, as is required for appropriate synthesis.

For the forward and interior problems, an existence test can be formulated from Equations f_2–f_5 to give a square system in unknowns B_x, B_y, C_x, and C_y. The Jacobian matrix $\mathbf{J}$ and Hessian matrix $\mathbf{H}$ of the corresponding

system with these unknowns are

$$\mathbf{J} = \begin{pmatrix} -2p - 2u + 2B_x & -2q - 2v + 2B_y & 0 & 0 \\ -2A_x + 2B_x & -2A_y + 2B_y & 0 & 0 \\ 0 & 0 & -2A_x + 2C_x & -2A_y + 2C_y \\ 2B_x - 2C_x & 2B_y - 2C_y & -2B_x + 2C_x & -2B_y + 2C_y \end{pmatrix} \tag{6.9}$$

$$\mathbf{H} = \left(\begin{array}{cccc|cccc|cccc|cccc} 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & -2 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & -2 & 0 & 2 \end{array}\right) \tag{6.10}$$

Each element of the Jacobian matrix (6.9), *i.e.*, $\mathbf{J}_{ij}$, does not have repeated variables, making the formulation ideal for interval analysis. Furthermore, the optimal evaluation of the Jacobian elements allows one to check the monotonicity of an equation with respect to the variables. If an equation is monotonic it allows for a better interval evaluation of the equation. As well, the Hessian matrix is constant and therefore does not need to be recomputed. To create various solving routines, a bisection phase may safely bisect the input and/or output angles $[\theta]$ and $[\psi]$. The bisection of a coordinate may lead to a loss of solutions due to the wrapping effect of interval analysis, whereas the bisection of an angle is safe.

6.6 Coupler Curve of an Appropriate Design

The coupler curve of an exact design is of sixth order in general, while some special linkage configurations may produce fourth order and second order curves (Hunt, 1978). For example, the parallelogram linkage generates circular coupler curves of order two. In general, the sixth order coupler curve may intersect a line or a circle no more than six times (Roth and Freudenstein, 1963); thus, no more than six exact precision points may lie on a given line or circle. In fact, it can be shown that the coupler curve equation is a special case of the sextic bivariate polynomial with a circularity of three and that the synthesis problem for a given coupler curve is square (Bai and Angeles, 2015). When uncertainties are introduced, representation of the coupler curve becomes a difficult problem.

Given an appropriate design $[\mathcal{D}]$ of a linkage, it is possible to represent the coupler curve as a set of boxes which tightly contain the true coupler curves associated with each $\mathcal{D} \in [\mathcal{D}]$. This provides a description of the worst-case error in the coupler curve. If the set of boxes, pertaining to some portion of the coupler curve are contained inside a desired response, then it is possible to state that the linkage design is guaranteed to satisfy the desired response. The appropriate analysis routine in Algorithm 6 solves for the set of coupler point intervals $([C_x], [C_y])$ using the forward problem formulation. Let $\Delta\theta$ represent the desired width of the input angle interval (the resolution). A loop iterates over all values of $[\theta]$ in the range $[0, 2\pi]$. Inside this loop, an

Algorithm 6: Coupler curve routine

function COUPLER_CURVE ($[\mathbf{u}], [\mathbf{k}], \Delta\theta, \epsilon, \beta$);
Input : unknowns $[\mathbf{u}]$, knowns $[\mathbf{k}]$, input angle resolution $\Delta\theta$, desired solution resolution ϵ, and desired bisection resolution β
Output: the list of solutions $\mathcal{L}_{u^*}$ for the unknowns
$[\theta] = [0, \Delta\theta]$;
while $[\theta] < 2\pi$ **do**
    $\mathcal{L}_{u^*} \leftarrow$ *GENERAL_SOLVE*$([\mathbf{u}], [\mathbf{k}], \epsilon, \beta)$; `/* Call the general solving routine */`
    $[\theta] = [\theta] + \Delta\theta$

interval general solving routine returns the list of solution coupler points $\mathcal{L}_{u^*}$ which correspond to $[\theta]$ (boundary points may also be saved). These solutions may correspond to different assemblies.

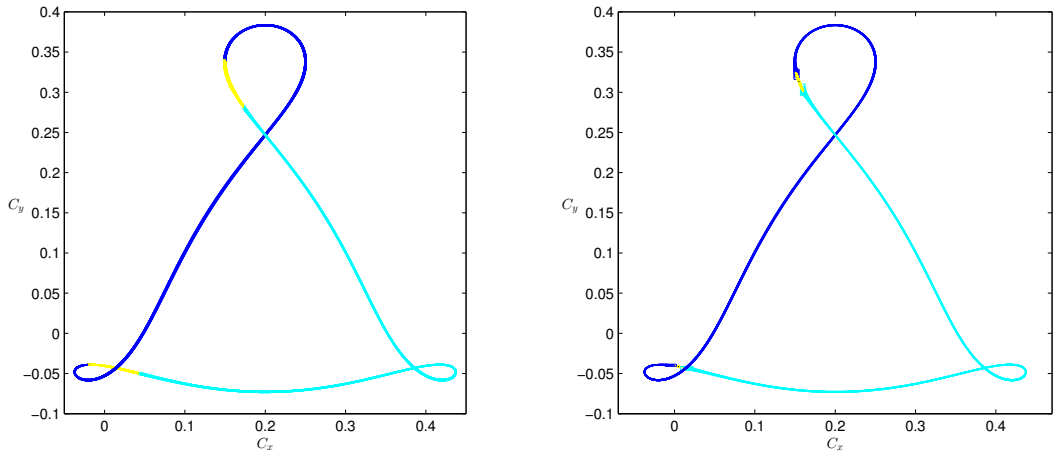
The unknowns and knowns inside the general solving routine are $[\mathbf{u}] = ([B_x], [B_y], [C_x], [C_y])$ and $[\mathbf{k}] = ([\mathcal{D}], [A_x], [A_y])$. The existence method considers the four variables in $[\mathbf{u}]$ and is formulated using the Jacobian (6.9) and Hessian (6.10). The bisection method is applied to $[\psi]$, which affects $[B_x]$ and $[B_y]$ due to f_8 to f_9 in (6.1).

As an example, the coupler curve for the appropriate design given in (6.11) is plotted in Figure 6.3a for $\Delta\theta = 0.001$ rad. The appropriate design considers an uncertainty of $\Delta\mathcal{D} = 0.0001$ added to each design parameter. The desired resolutions are selected as $\epsilon = 0.0000001$ m and $\beta = 0.0005$ rad. The Kantorovitch existence method and largest-first bisection are applied here. The resulting linkage is classified as a 0π-double-rocker non-Grashof linkage. The linkage has a single circuit containing two different branch configurations, identified by different colours. Coupler points in the neighbourhood of the

toggle position do not yield unique solutions due to the existence method not being able to guarantee solutions. These regions are considered as unknown in terms of coupler point solutions. The set of intervals provides an outer approximation for the actual coupler curve, *i.e.*, the actual coupler curve is contained inside the union of the set of intervals.

$$[\mathcal{D}] = ([u], [v], [p], [q], [r], [s], [c], [e], [f])^T$$

$$[\mathcal{D}] = ([0.0], [0.0], [0.4], [0.0], [0.24], [0.24], [0.2517], [0.12585], [0.15534])^T \pm \Delta\mathcal{D}^T \tag{6.11}$$



(a) (b)

Figure 6.3: The coupler curve corresponding to an appropriate design using: a) the Kantorovitch existence method; b) Newton's method to determine two non-intersecting solutions.

An improvement may be made to the general solving procedure in Algorithm 6 when obtaining the coupler curve. Each input angle $[\theta]$ yields two

coupler point solutions, corresponding to the two branches. The existence test may be replaced by calls to Newton's method, where the uniqueness guarantee provided by Kantorovitch is not necessary. If Newton's method is able to determine two non-intersecting solutions for the unknown parameters, then each solution is unique and corresponds to a coupler point associated with the input angle. If only a single solution is found, then Kantorovitch must be used in order to ensure that the solution is unique. If unique solutions cannot be determined, then the remaining boxes are not able to be classified. This improvement is able to increase the number of solutions found in near-singular regions. This coupler curve obtained using the improved method, with the same settings as before, is plotted in Figure 6.3b.

The effect of the uncertainties on the design parameters may be further explored through the coupler curves generated by several exact designs. Here the exact designs are selected as the lower bound, midpoint, and upper bound of the appropriate design parameters. The exact coupler curves corresponding to these exact designs are plotted in Figure 6.4. These coupler curves may also be obtained using Algorithm 6 with the initial $[\theta]$ equal to $[0]$. The solutions are a set of points[1]. The curves are approximated by connecting these points. The generated coupler curves may change position with respect to one another. In fact, do to the non-linearity of the problem, the vertices from the design space are not guaranteed to yield boundaries for the

[1]In this case a point is an interval with minimal width such that it is accurately represented in floating point format.

coupler curve. It is difficult to know which exact design, contained within the appropriate design, produces a boundary of the coupler curve of the appropriate design. Even if this design could be determined, it would likely change with the input angle. Thus, exactly determining the coupler curve of an appropriate design proves to be a difficult task.

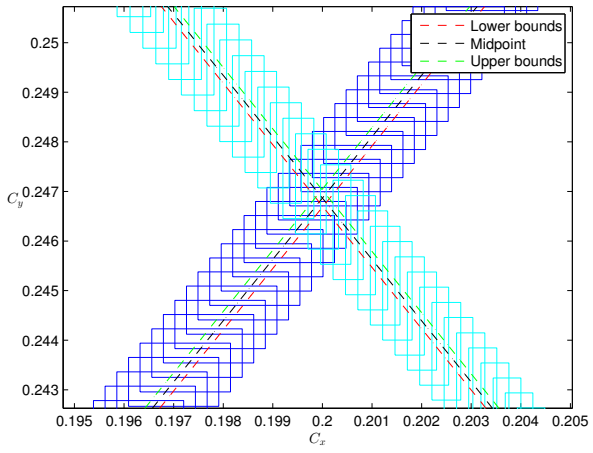


Figure 6.4: A detailed view showing the exact coupler curves corresponding to the lower bound, midpoint, and upper bound of the design parameters from the appropriate design.

To further explore the concept of coupler curves of appropriate designs, appropriate design parameters are selected which correspond to each type of linkage classification, as well as folding linkages. The resulting coupler curves are then presented. For each linkage design, an design uncertainty of $\Delta\mathcal{D} = 0.0001$ is assumed.

1. A crank-rocker linkage is obtained by selecting the appropriate design:

$$[\mathcal{D}] = ([0], [0], [0.4], [0], [0.1], [0.4], [0.2517], [0.12585], [0.15534])^T \pm \Delta\mathcal{D}^T \tag{6.12}$$

which yields the following values for classification:

$$[T_1] = [0.1513, 0.1521000125]$$
$$[T_2] = [0.4479, 0.4487000125]$$
$$[T_3] = [0.1512999875, 0.1521]$$

The corresponding coupler curves are given in Figure 6.5a.

2. A rocker-crank linkage is obtained by selecting the appropriate design:

$$[\mathcal{D}] = ([0], [0], [0.4], [0], [0.4], [0.1], [0.2517], [0.12585], [0.15534])^T \pm \Delta\mathcal{D}^T \tag{6.13}$$

which yields the following values for classification:

$$[T_1] = [0.1513, 0.1521000125]$$
$$[T_2] = [-0.1521, -0.1512999875]$$
$$[T_3] = [-0.4487000125, -0.4479]$$

The corresponding coupler curves are given in Figure 6.5b.

3. A double-crank linkage is obtained by selecting the appropriate design:

$$[\mathcal{D}] = ([0], [0], [0.1], [0], [0.4], [0.4], [0.2517], [0.12585], [0.15534])^T \pm \Delta\mathcal{D}^T \tag{6.14}$$

which yields the following values for classification:

$$[T_1] = [-0.4487, -0.44789995]$$
$$[T_2] = [-0.1521, -0.15129995]$$
$$[T_3] = [0.15129995, 0.1521]$$

The corresponding coupler curves are given in Figure 6.5c.

4. A double-rocker linkage is obtained by selecting the appropriate design:

$$[\mathcal{D}] = ([0], [0], [0.4], [0], [0.4], [0.4], [0.2517], [0.12585], [0.15534])^T \pm \Delta\mathcal{D}^T \tag{6.15}$$

which yields the following values for classification:

$$[T_1] = [-0.1487, -0.1478999875]$$
$$[T_2] = [0.1479, 0.1487000125]$$
$$[T_3] = [-0.1487000125, -0.1479]$$

The corresponding coupler curves are given in Figure 6.5d.

5. A 00-double-rocker linkage is obtained by selecting the appropriate de-

sign:

$$[\mathcal{D}] = ([0], [0], [0.3], [0], [0.44], [0.24], [0.3], [0.12585], [0.15534])^T \pm \Delta\mathcal{D}^T \tag{6.16}$$

which yields the following values for classification:

$$[T_1] = [-0.0804, -0.07959998334]$$
$$[T_2] = [-0.2004, -0.1995999833]$$
$$[T_3] = [-0.2004000167, -0.1996]$$

The corresponding coupler curve is given in Figure 6.5e.

6. A 0π-double-rocker linkage is obtained by selecting the appropriate design:

$$[\mathcal{D}] = ([0], [0], [0.4], [0], [0.24], [0.24], [0.2517], [0.12585], [0.15534])^T \pm \Delta\mathcal{D}^T \tag{6.17}$$

which yields the following values for classification:

$$[T_1] = [0.1713, 0.1721000125]$$
$$[T_2] = [0.1479, 0.1487000125]$$
$$[T_3] = [-0.1487000125, -0.1479]$$

The corresponding coupler curve is given in Figure 6.5f.

7. A $\pi 0$-double-rocker linkage is obtained by selecting the appropriate

design:

$$[\mathcal{D}] = ([0], [0], [0.4], [0], [0.24], [0.24], [0.41], [0.12585], [0.15534])^T \pm \Delta\mathcal{D}^T \tag{6.18}$$

which yields the following values for classification:

$$[T_1] = [0.3296, 0.3304000125]$$
$$[T_2] = [-0.0104, -0.009599987503]$$
$$[T_3] = [0.009599987503, 0.0104]$$

The corresponding coupler curve is given in Figure 6.5g.

8. A $\pi\pi$-double-rocker linkage is obtained by selecting the appropriate design:

$$[\mathcal{D}] = ([0], [0], [0.4], [0], [0.33], [0.44], [0.3], [0.12585], [0.15534])^T \pm \Delta\mathcal{D}^T \tag{6.19}$$

which yields the following values for classification:

$$[T_1] = [-0.0704, -0.0695999875]$$
$$[T_2] = [0.2096, 0.2104000125]$$
$$[T_3] = [0.009599987503, 0.0104]$$

The corresponding coupler curve is given in Figure 6.5h.

If the input link is a crank, the linkage is non-folding, and the uncertainties

are small, then the current existence methods are able to effectively solve for each coupler point interval which corresponds to a given input angle range. If the input link is a rocker, which includes all non-Grashof classifications, then the linkage contains a toggle position which provides a change of branch. The uncertainties present in the linkage make it difficult to determine the coupler points near a toggle position using the current existence methods. The linkage is considered to be singular near toggle positions, and therefore the coupler point becomes extremely sensitive to the input angle.

The sensitivity of the coupler points near toggle positions is witnessed in the coupler curves of Figures 6.5b, 6.5d, and 6.5e to 6.5h. As the input angle approaches a value which produces a toggle position, the resulting coupler curve interval grows in width. The existence methods are able to accommodate this growth in width to an extent, but eventually the coupler curve interval width exceeds the capabilities of the existence methods. At this point, the remaining portions of the coupler curve can only be approximated and cannot be guaranteed. The input angle may be reduced in width to attempt to obtain additional coupler curve solutions.

The coupler curve of folding linkages may also be considered. For example:

1. A folding linkage which generates a crank-rocker linkage and a 0π-double-rocker linkage is obtained by the selecting:

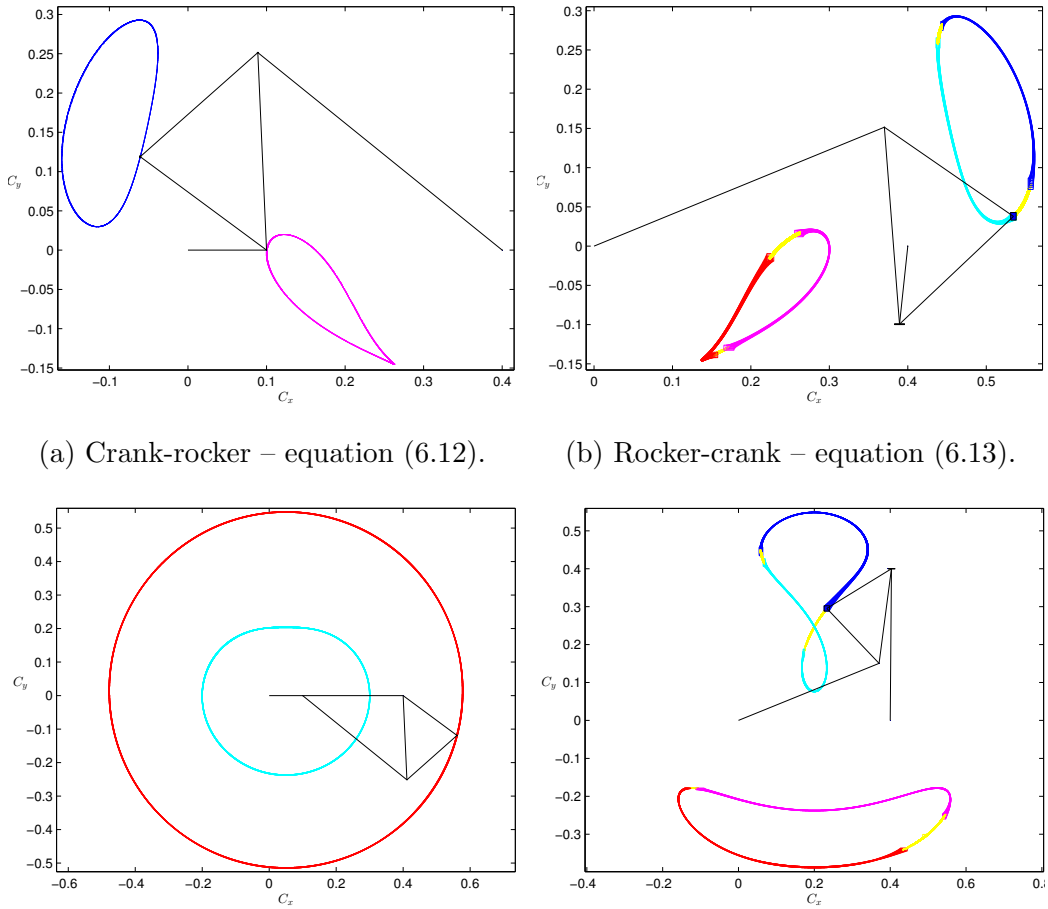
$$[\mathcal{D}] = ([0], [0], [0.4], [0], [0.34], [0.44], [0.3], [0.12585], [0.15534])^T \pm \Delta\mathcal{D}^T \tag{6.20}$$

which yields the following values for classification:

$$[T_1] = [0.1196, 0.1204000125]$$

$$[T_2] = [0.1996, 0.2004000125]$$

$$[T_3] = [-0.0004000124969, 0.0004]$$



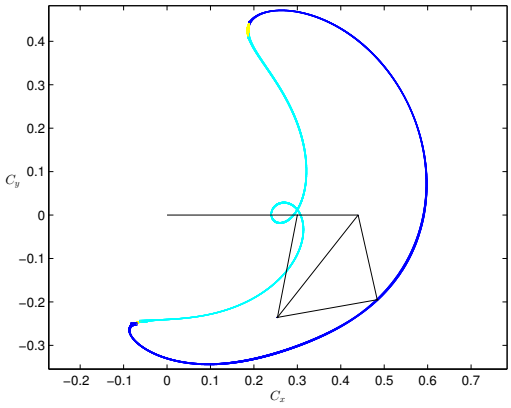
(a) Crank-rocker – equation (6.12).

(b) Rocker-crank – equation (6.13).

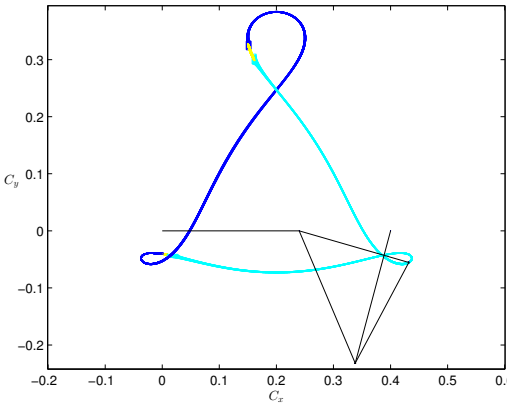
(c) Double-crank – equation (6.14).

(d) Double-rocker – equation (6.15).

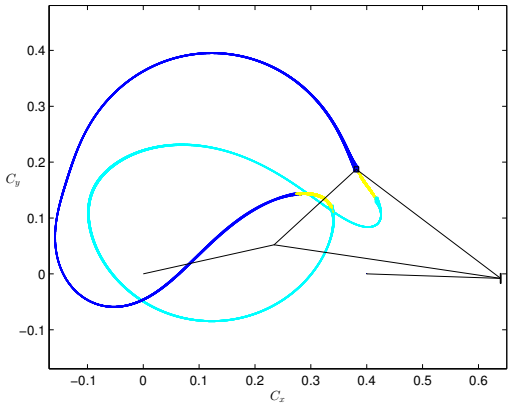
Figure 6.5: The coupler curves corresponding to appropriate designs with different classifications.



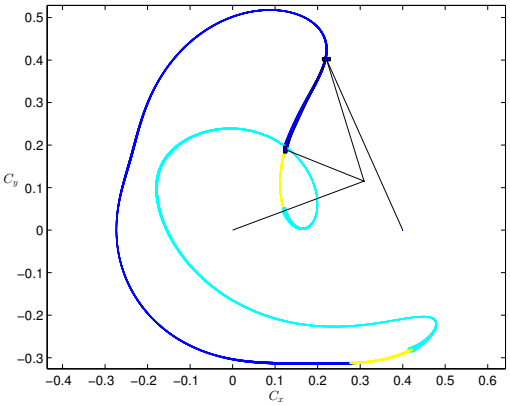
(e) 00-double-rocker – equation (6.16).



(f) 0π-double-rocker – equation (6.17).



(g) $\pi 0$-double-rocker – equation (6.18).



(h) $\pi\pi$-double-rocker – equation (6.19).

Figure 6.5: Continued. The coupler curves corresponding to appropriate designs with different classifications.

The corresponding coupler curves are given in Figure 6.6a.

2. A folding linkage which generates a crank-rocker linkage, a rocker-crank linkage, a 0π-double-rocker linkage, and a $\pi 0$-double-rocker linkage is

obtained by the selecting the appropriate design:

$$[\mathcal{D}] = ([0.0], [0.0], [0.4], [0.0], [0.24], [0.24], [0.4], [0.12585], [0.15534])^T \pm \Delta\mathcal{D}^T \tag{6.21}$$

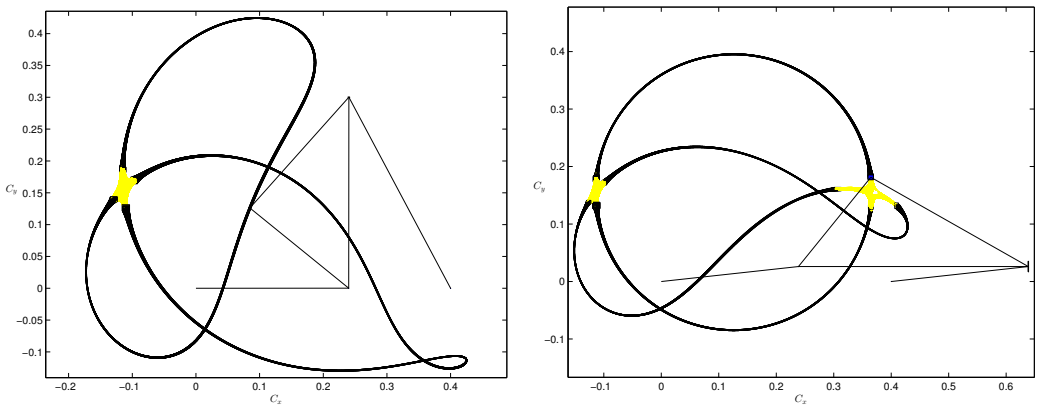
which yields the following values for classification:

$$[T_1] = [0.3196, 0.3204000125]$$

$$[T_2] = [-0.0004, 0.0004000124969]$$

$$[T_3] = [-0.0004000124969, 0.0004]$$

The corresponding coupler curves are given in Figure 6.6b.



(a) Folding linkage – equation (6.20). (b) Folding linkage – equation (6.21).

Figure 6.6: The coupler curves corresponding to appropriate designs with multiple classifications (folding linkages).

For folding linkages, each coupler point solution has two or more associated classifications. Near the toggle position, the linkage may change from one classification to another, which may be desirable or undesirable given

the usage of the linkage. In terms of manufacturing, the manufacturing uncertainties limit the ability to manufacture a linkage to exact specifications. That is not to say that a folding linkage is not possible, as the introduction of other sources of uncertainties (*e.g.*, uncertainties in the joints and flexibility in the links) can indeed generate a folding linkage. The performance of the folding linkage resulting from uncertainties cannot be determined from exact specifications. The toggle position will no longer be a point, but rather an area. This issue demonstrates the need for linkages described by appropriate designs and analysed with appropriate analysis routines.

6.7 Desired Response: Description and Verification Methods

The goal of appropriate synthesis applied to the four-bar linkage is to be able to synthesize appropriate designs of four-bar linkages given the description of a desired coupler curve. The desired coupler curve will be described by a set of precision points and/or a set of trajectories. Each element, whether a precision point or a trajectory, is described with an allowable error. The allowable error is a requirement, since an exact solution for the location of the coupler point is not possible. The descriptions of precision points and trajectories and the routines for verifying the satisfaction of the precision point and trajectory elements for a given appropriate design will now be presented.

6.7.1 Precision Points

In classical linkage synthesis, precision points are described exactly where the synthesized linkage must pass through each point. Here, an exact description of a precision point is not useful because the linkage itself has uncertainties in the design parameters leading to a non-exact solution of the coupler point. A precision point is described here as an interval vector, such that each element may have an allowable error. A precision point will be denoted by $\mathcal{P}$ as

$$\mathcal{P} = ([C_x], [C_y], [\theta], [\psi]) \tag{6.22}$$

In order for a linkage to satisfy a precision point, there must exist a solution $\mathcal{P}^*$ for the linkage, such that $[C_x^*] \subset [C_x]$, $[C_y^*] \subset [C_y]$, $[\theta^*] \subset [\theta]$ and $[\psi^*] \subset [\psi]$. That is, $\mathcal{P}^* \subset \mathcal{P}$. Multiple precision points may be considered.

6.7.2 Trajectories

A desired trajectory, having an allowable error, may be considered. The desired trajectory, denoted $f([C_x], [C_y])$, can be described by parametric equations as

$$f([C_x], [C_y]) = \{(C_x, C_y) \mid C_x = f_x(t),\ C_y = f_y(t),\ t \in [\underline{t}, \overline{t}]\} \tag{6.23}$$

An allowable error on the trajectory can be described by $\alpha\hat{\mathbf{n}}$ where $\hat{\mathbf{n}}$ is the unit normal along $f([C_x], [C_y])$ and $\alpha = [\underline{\alpha}, \overline{\alpha}]$ is the allowable error. The

allowable trajectory can then be defined as

$$f([C_x],[C_y]) = \{(C_x, C_y) \mid C_x = f_x(t) + \alpha\hat{\mathbf{n}},\ C_y = f_y(t) + \alpha\hat{\mathbf{n}}, \\ t \in [\underline{t}, \overline{t}],\ \alpha \in [\underline{\alpha}, \overline{\alpha}]\} \tag{6.24}$$

Since it is difficult to account for the limits of the trajectory, the domain of $[t]$ is inflated by some small amount δ to create start and finish end-points for the trajectory. These end-points will have a width δ and be defined by $[t_{start}] = [\underline{t} - \delta, \underline{t}]$ and $[t_{finish}] = [\overline{t}, \overline{t} + \delta]$. The task element $\mathcal{T}$ corresponding to a trajectory is represented by the desired trajectory $f([C_x],[C_y])$, input and output angles $[\theta]$ and $[\psi]$, allowable error $[\alpha]$, the parametric equation parameter $[t]$, and end-point width δ as

$$\mathcal{T} = \{f([C_x],[C_y]), [\theta], [\psi], [\alpha], [t], \delta\} \tag{6.25}$$

In order for a linkage to satisfy a trajectory, a solution must exist for each end-point. As well, the coupler curve must be continuous between the end-points while also remaining within the boundaries of the trajectory. The steps to satisfy a trajectory may be summarized as:

1. Ensure the start end-point is satisfied: There must exist a solution $\mathcal{T}^*$, such that

$$[C^*_{start}] \subset f([C_x],[C_y]),\ [\theta^*_{start}] \subset [\theta],\ [\psi^*_{start}] \subset [\psi], \\ [\alpha^*_{start}] \subset [\alpha],\ [t^*_{start}] \subset [t_{start}] \tag{6.26}$$

2. Ensure the finish end-point is satisfied: There must exist a solution $\mathcal{T}^*$, such that

$$[C^*_{finish}] \subset f([C_x],[C_y]),\ [\theta^*_{finish}] \subset [\theta],\ [\psi^*_{finish}] \subset [\psi],\\ [\alpha^*_{finish}] \subset [\alpha],\ [t^*_{finish}] \subset [t_{finish}] \tag{6.27}$$

3. Ensure the coupler curve is continuous between the start and finish end-points and remains within the boundaries of the trajectory: There must exist a solution $\mathcal{T}^*$ for each $[\theta] \in ([\theta^*_{start}]\square[\theta^*_{finish}])$, such that

$$[C^*] \subset f([C_x],[C_y]),\ [\theta^*] = [\theta],\ [\psi*] \subset [\psi],\ [\alpha^*] \subset [\alpha],\\ [t^*] \subset ([t_{start}]\square[t_{finish}]) \tag{6.28}$$

Due to the periodicity of the angle θ, the interval hull $([\theta^*_{start}]\square[\theta^*_{finish}])$ has two associated domains. These domains correspond to clock-wise and counter-clock-wise rotations from $[\theta^*_{start}]$ to $[\theta^*_{finish}]$.

Multiple trajectories may be considered.

Consider the parametric curve for $y = x^2,\ x \in [-1,1]$.

$$\begin{aligned} x &= t \\ y &= t^2 \end{aligned} \tag{6.29}$$

The normal $\mathbf{n}$ is determined as $\mathbf{n} = [2x,-1]^T = [2t,-1]^T$ and the unit normal

$\hat{\mathbf{n}}$ is $\mathbf{n}/||\mathbf{n}||$ with $||\mathbf{n}|| = \sqrt{4t^2+1}$. Thus the allowable parametric curve is

$$\begin{aligned} x &= t + \alpha\hat{\mathbf{n}}_x = t + \alpha(2t/\sqrt{4t^2+1}) \\ y &= t^2 + \alpha\hat{\mathbf{n}}_y = t^2 - \alpha(1/\sqrt{4t^2+1}) \\ t &\in [-1, 1] \end{aligned} \tag{6.30}$$

The allowable coupler curve is plotted in Figure 6.7 for $t \in [-0.99, 0.99]$ and $\alpha \in [-0.1, 0.1]$. The allowable end-points of the curve are given as $[t_{start}] \in [-1.01, -0.99]$ and $[t_{finish}] \in [0.99, 1.01]$, respectively. The trajectory must start and finish by satisfying the end-points. These end-points differ from precision points because they are not axis-aligned; thus, these end-points better approximate the desired curve.

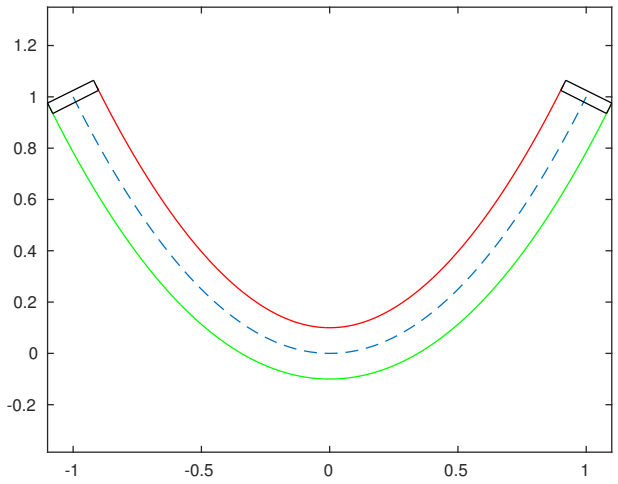


Figure 6.7: The desired trajectory corresponding to $y = x^2$.

6.7.3 Verification Test for Precision Points

Let a desired response, denoted by $\mathcal{R}$, contain m precision points $\mathcal{P}_i$.

$$\mathcal{R} = (\mathcal{P}_1, \ldots, \mathcal{P}_m) \tag{6.31}$$

The goal is to determine if an appropriate design $[\mathcal{D}]$ is able to achieve each response element in $\mathcal{R}$. A routine is used in order to verify the existence of a coupler point solution inside each of the precision points.

The known and unknown parameters will vary for the different phases: simplification, existence, and bisection. This ensures that the equations in the existence phase are as simple as possible, and that the coordinates are not bisected. The design parameters will always be considered as knowns. In the simplification phase, the unknowns are selected as $[\mathbf{u}] = ([A_x], [A_y], [B_x], [B_y], [C_x], [C_y], [\theta], [\psi])$. In the existence phase, the unknowns are selected as $[\mathbf{u}_{exist}] = ([B_x], [B_y], [C_x], [C_y])$, such that the interior problem may be used. In the bisection phase, the unknown parameters are selected as $[\mathbf{u}_{bisect}] = ([\theta], [\psi])$. Bisecting the angles, rather than the coordinates, ensures that the coordinates $[A]$ and $[B]$ always contain the true joint positions. All remaining parameters may be considered as knowns.

The routine for verifying m precision points is presented in Algorithm 7. Since there are m precision points to be verified, all m associated unknown vectors will be considered simultaneously. If any precision point is able to quickly return no solution, then the routine exits and resources are not wasted

on the other precision points. To accomplish this, the set of m unknown vectors, $\mathbb{U} = \{[\mathbf{u}_i], \ldots, [\mathbf{u}_m]\}$, are added to the list $\mathcal{L}_u$. Each phase then accepts as input the set $\mathbb{U}$ and applies the corresponding operation to each element $[\mathbf{u}_i]$ from the set. Let $\mathbb{U}_{exist}$ and $\mathbb{U}_{bisect}$ represent the set of unknowns corresponding to the existence and bisection phases respectively. The simplification phase simplifies all m unknowns $[\mathbf{u}_i]$ and returns:

$$\begin{aligned} 1 &: \text{ one or more unknowns } [\mathbf{u}_i] \text{ is simplified;} \\ -1 &: \text{ any unknown } [\mathbf{u}_i] \text{ becomes empty } (\emptyset); \\ 0 &: \text{ the } m \text{ unknowns } [\mathbf{u}_i] \text{ cannot be simplified.} \end{aligned}$$

The assembly routine verifies the assembly conditions and returns:

$$\begin{aligned} 1 &: \text{ a single valid assembly is found for all } m \text{ unknowns } [\mathbf{u}_i]; \\ -1 &: \text{ one or more unknowns have invalid assembly conditions;} \\ 0 &: \text{ otherwise.} \end{aligned}$$

The existence phase checks for the existence of a solution for all m unknowns $[\mathbf{u}_{exist_i}]$ and returns:

$$\begin{aligned} 1 &: \text{ all } m \text{ precision points are verified;} \\ -1 &: \text{ any unknown } [\mathbf{u}_{exist_i}] \text{ has no solution;} \\ 0 &: \text{ a solution cannot be found for all } m \text{ unknowns } [\mathbf{u}_{exist_i}]. \end{aligned}$$

The bisection phase bisects the unknowns $[\mathbf{u}_{bisect_i}]$ which return 0 from the existence phase.

Algorithm 7: Verify m precision points

function VERIFY_PRECISION_POINTS $(\mathcal{R}, [\mathcal{D}], \epsilon, \beta)$;

Input : desired response $\mathcal{R}$, appropriate design $[\mathcal{D}]$, desired solution resolution ϵ, and desired bisection resolution β

Output: verification - the desired response is: satisfied(1), unsatisfied(-1), partially satisfied(0)

$\mathcal{L}_u \leftarrow \mathbb{U} = \{[\mathbf{u}_i], \ldots, [\mathbf{u}_m]\}$; `/* Add set of unknowns to list */`

$\mathcal{L}_{class} \leftarrow \mathcal{L}_{allow\_class}$; `/* Add allowed classifications to list */`

$\mathcal{L}_{branches} \leftarrow \{1, 2\}$; `/* Add possible branches to list */`

$\mathcal{L}_{circuits} \leftarrow \{1, 2\}$; `/* Add possible circuits to list */`

while $\mathcal{L}_u$ *is not empty* **do**

  Pop set $\mathbb{U}$ from $\mathcal{L}_u$;

  $simp = SIMPLIFICATION(\mathbb{U}, [\mathbf{k}])$; `/* Apply simplification techniques */`

  if $simp \neq -1$ **then**

    $assembly = ASSEMBLY(\mathbb{U}, [\mathbf{k}], \{\mathcal{L}_{class}\}, \{\mathcal{L}_{branches}\}, \{\mathcal{L}_{circuits}\})$;

    `/* Verify assembly conditions */`

    if $assembly == 1$ **then**

      $exist = EXISTENCE(\mathbb{U}_{exist}, [\mathbf{k}], \epsilon)$; `/* Apply existence methods */`

      if $exist == 1$ **then**

        return 1

      else if $exist == 0$ **then**

        $\mathcal{L}_u \leftarrow BISECTION(\mathbb{U}_{bisect}, \beta)$; `/* Apply bisection method */`

      else

        return 0;

return -1;

6.7.4 Verification Test for Trajectories

Let a set of responses $\mathcal{R}$ contain n trajectories $\mathcal{T}_i$.

$$\mathcal{R} = (\mathcal{T}_1, \ldots, \mathcal{T}_n) \tag{6.32}$$

The goal is to determine if an appropriate design $[\mathcal{D}]$ is able to achieve each response element in $\mathcal{R}$. A routine is used in order to verify the existence of a coupler point solution along the entire trajectory.

Like the precision point problem, the known and unknown parameters will vary for the different phases. The design parameters will always be considered as knowns. In the simplification phase, the unknowns are selected as $[\mathbf{u}] = ([A_x], [A_y], [B_x], [B_y], [C_x], [C_y], [\theta], [\psi], [t], [\alpha])$. In the existence phase, the unknowns are selected as $[\mathbf{u}_{exist}] = ([B_x], [B_y], [C_x], [C_y], [t], [\alpha])$. Similar to the interior problem, a square system is created using equations f_2 through f_5 and the parametric equations of the trajectory curve. In the bisection phase , the unknown parameters are selected as $[\mathbf{u}_{bisect}] = ([\theta], [\psi])$. All remaining parameters may be considered as knowns.

In order to verify a trajectory, each $t \in [t]$ must have a solution which stays within the allowable error of the trajectory. Each solution must also satisfy the assembly conditions and the coupler curve must be continuous along $[t]$. The first consideration is the verification of the end-points of a trajectory. This is done in the same manner as the precision points verification, albeit with the addition of the parameters $[t]$ and $[\alpha]$. If the end-points of each

trajectory are verified, then the remainder of the trajectory can be considered. Conveniently, the verification test for an end-point is identical to the verification test for the remainder of the trajectory (internal-points). Therefore, a single routine can be used. The routine is presented in Algorithm 8. Similar to the precision point verification routine, this routine considers the end-points or internal-points for all trajectories simultaneously in order to improve performance. For the end-point verification of n trajectories, let $k = 2n$. If any point returns no solution, the entire routine exits.

A second routine is required in order to organize the testing of the end-points and internal-points. This routine is presented in Algorithm 9. It makes repeated calls to Algorithm 8 in order to verify the end-points and the internal-points. The end-points are first verified. If the verification succeeds on all end-points then the interior points of each trajectory are considered in turn, starting from the lower bound of the input angle domain associated with the trajectory $(([\theta^*_{start}]\square[\theta^*_{finish}]))$ and incrementing by $\Delta\theta$ until the upper bound of the input angle domain is reached. If any interior point is unable to yield a solution, then the routine is unable to verify all of the trajectories and must exit.

A complication arises in the proposed trajectory verification algorithm resulting from the coupler curve of a linkage passing through the end-points of a trajectory more than once. The end-point verification in Algorithm 8 succeeds when a solution is found for each end-point which satisfies the assembly conditions; however, the input angle domain associated with the trajectory

Algorithm 8: Verify k endpoints

function VERIFY_TRAJECTORY_POINTS $(\mathcal{R}, [\mathcal{D}], \epsilon, \beta)$;

Input : desired response $\mathcal{R}$, appropriate design $[\mathcal{D}]$, desired solution resolution ϵ, and desired bisection resolution β

Output: verification - the desired response is: satisfied(1), unsatisfied(-1), partially satisfied(0)

$\mathcal{L}_u \leftarrow \mathbb{U} = \{[\mathbf{u}_i], \ldots, [\mathbf{u}_m]\}$; `/* Add set of unknowns to list */`

$\mathcal{L}_{class} \leftarrow \mathcal{L}_{allow\_class}$; `/* Add allowed classifications to list */`

$\mathcal{L}_{branches} \leftarrow \{1, 2\}$; `/* Add possible branches to list */`

$\mathcal{L}_{circuits} \leftarrow \{1, 2\}$; `/* Add possible circuits to list */`

while $\mathcal{L}_u$ *is not empty* **do**

  Pop set $\mathbb{U}$ from $\mathcal{L}_u$;

  $simp = SIMPLIFICATION(\mathbb{U}, [\mathbf{k}])$; `/* Apply simplification techniques */`

  if $simp \neq -1$ **then**

    $assembly = ASSEMBLY(\mathbb{U}, [\mathbf{k}], \{\mathcal{L}_{class}\}, \{\mathcal{L}_{branches}\}, \{\mathcal{L}_{circuits}\})$;

    `/* Verify assembly conditions */`

    if $assembly == 1$ **then**

      $exist = EXISTENCE(\mathbb{U}_{exist}, [\mathbf{k}], \epsilon)$; `/* Apply existence methods */`

      if $exist == 1$ **then**

        return 1

      else if $exist == 0$ **then**

        $\mathcal{L}_u \leftarrow BISECTION(\mathbb{U}_{bisect}, \beta)$; `/* Apply bisection method */`

      else

        return 0;

return -1;

Algorithm 9: Verify n trajectories

function VERIFY_TRAJECTORIES $(\mathcal{R}, [\mathcal{D}], \epsilon, \beta, \Delta\theta)$;

Input : desired response $\mathcal{R}$, appropriate design $[\mathcal{D}]$, desired solution resolution ϵ, desired bisection resolution β, and angle resolution $\Delta\theta$

Output: verification - the desired response is: satisfied(1), unsatisfied(-1), partially satisfied(0)

$\mathcal{R}_E$ = Generate end-point description from $\mathcal{R}$;

$verified = VERIFY\_TRAJECTORY\_POINTS(\mathcal{R}_E, [\mathcal{D}], \epsilon, \beta)$; `/* Verify the end-points of the trajectories */`

if $verified == 1$ **then**

  forall $\mathcal{T}_i \in \mathcal{R}$ **do**

    $[\theta_i] = \overline{([\theta^*_{start}]\square[\theta^*_{finish}])} + [0, \Delta\theta]$;

    while $[\theta_i] < \overline{([\theta^*_{start}]\square[\theta^*_{finish}])}$ **do**

      $\mathcal{R}_I$ = Generate interior-point description from $\mathcal{R}_E$;

      $verified = VERIFY\_TRAJECTORY\_POINTS(\mathcal{R}_I, [\mathcal{D}], \epsilon, \beta)$;

      `/* Verify the interior-points of the trajectories */`

      if $verified \neq 1$ **then**

        return $verified$; `/* An interior-point cannot be verified */`

      $[\theta_i] = [\theta_i] + \Delta\theta$;

  return 1 ; `/* All interior-points have been verified */`

else

  return $verified$

$(([\theta^*_{start}]\square[\theta^*_{finish}]))$ cannot be assumed to have minimal width. As an example, consider applying the trajectory verification to a crank-rocker linkage which yields input angles for the end-points as $[\theta^*_{start}] = [-\pi/2] \pm 0.01$ and $[\theta^*_{finish}] = [\pi/3] \pm 0.01$. The input angle domain associated with the trajectory may be either $([\theta^*_{start}]\square[\theta^*_{finish}]) = [-\pi/2 + 0.01, \pi/3 - 0.01]$ for minimal width or $([\theta^*_{start}]\square[\theta^*_{finish}]) = [\pi/3 + 0.01, 3\pi/2 - 0.01]$. Without sampling interior points, it is unknown which input angle domain actually connects the end-points. An *input angle rejection list* is proposed to ensure that the correct end-points are selected and the correct input angle domain is then tested. The idea with the input angle rejection list is to create a list $\mathcal{L}_\theta$ which contains domains of the input angle which are known to dissatisfy the trajectory. Each time the interior-point verification test in Algorithm 9 fails, the input angle associated with the interior-point leading to failure is added to the rejection list $\mathcal{L}_\theta$. The trajectory verification routine then returns back to the end-point verification test and now uses the list $\mathcal{L}_\theta$ as an additional filtering method. Only end-point solutions which yield an input angle domain that does not contain the input angles in the rejection list are subjected to the interior-point verification test. The trajectory can only be satisfied when both the end-point and interior-point verification tests are satisfied.

Several other improvements may be made to the proposed trajectory verification algorithm. The velocity $\mathbf{v}$ of the coupler point, with respect to θ, may be used to improve the trajectory verification. If $\mathbf{j}$ is a vector tangent to the trajectory (*i.e.*, $\mathbf{j} = [\mathbf{n}_y, -\mathbf{n}_x]$, the dot product of $\mathbf{j}$ and $\mathbf{v}$ should al-

ways be positive. This would ensure that the coupler point moves only in the desired direction of the trajectory. It may also be beneficial to sample a point inside the interior region of the trajectory to better predict the input angle domain. As well, the stopping criteria of the interior-point test may be changed so that the test is successful as soon as a solution is found inside the final end-point, as it may not be necessary to test the entire input angle domain ($[\theta^*_{start}]\square[\theta^*_{finish}]$).

6.8 Appropriate Synthesis

A desired response, given by $\mathcal{R}$, contains a set of precision points and/or trajectories. Appropriate synthesis concerns finding all subsets of the initial design parameter space which generate a four-bar linkage that satisfies the desired response. This subset of designs is known as the *allowable region*. The allowable region ensures that each desired response element is satisfied, via Algorithms 7 and 9, and that the design solutions satisfy classification restrictions and assembly conditions.

In order to compute the allowable region, a list of appropriate design solutions, denoted by $\mathcal{L}_{[\mathcal{D}]}$, must be computed such that

$$\mathcal{L}_{[\mathcal{D}]} = \{[\mathcal{D}]_1, \ldots, [\mathcal{D}]_k\} \tag{6.33}$$

Since $\Delta\mathcal{D}$ is the manufacturing uncertainties on the design parameters, each

appropriate design solution $[\mathcal{D}]$ must have design parameters $[\mathcal{D}_i]$ which satisfy

$$\text{width}([\mathcal{D}_i]) \geq 2\Delta\mathcal{D}_i \tag{6.34}$$

That is, the width of each design parameter $[\mathcal{D}_i]$ must be greater than twice the corresponding manufacturing uncertainty. This is considered as a stopping criteria in the synthesis routine. An allowable design solution $[\mathcal{D}^*]$ is determined by deflating the bounds of $[\mathcal{D}]$ by $\Delta\mathcal{D}$

$$[\mathcal{D}^*] = [\underline{[\mathcal{D}]} + \Delta\mathcal{D}, \overline{[\mathcal{D}]} - \Delta\mathcal{D}] \tag{6.35}$$

This ensures that if an appropriate design solution $[\mathcal{D}]$ is found, then $[\mathcal{D}^*]$ defines a reduced box whose any interior point may be chosen as a nominal design solution, while ensuring that the desired precision points and trajectories can be performed by the real mechanism. To compute the full allowable region, adjacent appropriate design solutions in $\mathcal{L}_{[\mathcal{D}]}$ may be combined, where the boundaries of the region are deflated by $\Delta\mathcal{D}$.

In the previous section, routines for the verification of precision points and trajectories are presented. These routines may be combined into a single verification routine, described in Algorithm 10, which takes some description of $\mathcal{R}$ and determines if the appropriate design $[\mathcal{D}]$ is verified.

For appropriate synthesis, each design parameter has an initial domain, such that the design parameters are able to be bisected, until a desired bisection resolution of β, in order to search for design solutions. An initial bisection

Algorithm 10: Verify an appropriate design for a desired response

function VERIFY_APPROPRIATE_DESIGN $(\mathcal{R}, [\mathcal{D}], \epsilon, \beta, \Delta\theta)$;
Input : desired response $\mathcal{R}$, appropriate design $[\mathcal{D}]$, desired solution resolution ϵ, desired bisection resolution β, and angle resolution $\Delta\theta$
Output: verification - the desired response is: satisfied(1), unsatisfied(-1), partially satisfied(0)
$verified$ =VERIFY_PRECISION_POINTS$(\mathcal{R}, [\mathcal{D}], \epsilon, \beta)$;
if $verified \neq 1$ **then**
  return $verified$;
$verified$ =VERIFY_TRAJECTORIES$(\mathcal{R}, [\mathcal{D}], \epsilon, \beta, \Delta\theta)$;
if $verified \neq 1$ **then**
  return $verified$;
return 1;

resolution may be safely selected as $\beta = 2\Delta\mathcal{D}$, where each design parameter may have a different bisection resolution. Solutions are obtained with a desired solution resolution of ϵ, which dictates how tight of a solution is desired. The boundaries of the set of appropriate design solutions are unable to be classified as solutions, but can also not be classified as non-solutions; thus, there will always be a boundary layer between solutions and non-solutions. This provides three possible design parameter classifications:

1. Solution: every $\mathcal{D} \in [\mathcal{D}]$ contains a solution for each response element in $\mathcal{R}$;

2. Non-solution: some $\mathcal{D} \in [\mathcal{D}]$ does not contain a solution for one or more response elements in $\mathcal{R}$;

3. Unknown (boundary): it is unknown if every $\mathcal{D} \in [\mathcal{D}]$ contains a solu-

tion for each response element in $\mathcal{R}$, given the current resolution.

It should be noted that unknown classifications may occur if $\Delta\mathcal{D}$ is relatively large, as the uncertainties is the system may prevent the detection of a solution or non-solution. The volume of the set of unknown boxes may be reduced with an incremental algorithm. That is, run the appropriate synthesis again on the unknown boxes with a reduced $\Delta\mathcal{D}$ (*e.g.*, $\Delta\mathcal{D}/2$). Then, the adjacent appropriate design solutions in $\mathcal{L}_{[\mathcal{D}]}$ may be combined and the boundaries of the region may be deflated by $\Delta\mathcal{D}$ to determine the full allowable region. Such a process may be pursued until *volume*(boundaries)/*volume*(solutions) is arbitrarily low.

An appropriate synthesis routine is proposed in Algorithm 11, which returns three sets, the solutions $\mathcal{L}_{[\mathcal{D}]}$, the boundaries $\mathcal{L}_{[\mathcal{D}]_b}$, and the non-solutions $\mathcal{L}_{[\mathcal{D}]_n}$ for the appropriate designs. A list $\mathcal{L}_{test}$ contains the set of design parameters being considered. The appropriate design verification routine is applied to each $[\mathcal{D}]$ in the list $\mathcal{L}_{test}$. If the routine returns 0, then bisection is applied to the design parameters $[\mathcal{D}]$; bisection is only applied if $width([\mathcal{D}]) \geq 2\Delta\mathcal{D}$; otherwise, the design is saved to the boundary set. If the routine returns 1, then the design is verified and is saved to the solution set. If the routine returns -1, the design is not a solution and is discarded (or saved to the non-solution set).

Algorithm 11: Appropriate synthesis routine

function APPROPRIATE_SYNTHESIS $(\mathcal{R}, [\mathcal{D}], \Delta\mathcal{D}, \epsilon, \beta, \Delta\theta)$;
Input : desired response $\mathcal{R}$, appropriate design $[\mathcal{D}]$, design parameter uncertainties $\Delta\mathcal{D}$, desired solution resolution ϵ, desired bisection resolution β, and angle resolution $\Delta\theta$
Output: set of solutions $\mathcal{L}_{[\mathcal{D}]}$, non-solutions $\mathcal{L}_{[\mathcal{D}]_n}$, and boundaries $\mathcal{L}_{[\mathcal{D}]_b}$
$\mathcal{L}_{test} \leftarrow [\mathcal{D}]$;
while $\mathcal{L}_{test}$ *is not empty* **do**
  Pop $[\mathcal{D}]$ from $\mathcal{L}_{test}$;
  $appropriate = VERIFY\_APPROPRIATE\_DESIGN(\mathcal{R}, [\mathcal{D}], \epsilon, \beta, \Delta\theta)$;
  `/* Apply the verification routines */`
  if $appropriate == 1$ **then**
    $\mathcal{L}_{[\mathcal{D}]} \leftarrow [\mathcal{D}]$; `/* Save design to solution set */`
  else if $appropriate == -1$ **then**
    $\mathcal{L}_{[\mathcal{D}]_n} \leftarrow [\mathcal{D}]$; `/* Save design to non-solution set */`
  $(\mathcal{L}_{test}, \mathcal{L}_{[\mathcal{D}]_b}) \leftarrow BISECTION([\mathcal{D}], \Delta\mathcal{D})$; `/* Apply bisection */`

6.9 Case Studies

In this section, the appropriate synthesis routine is applied to solve for the design solutions (allowable regions) of four-bar linkages for different task requirements. For visualization purposes, mostly 2D solutions are presented.

6.9.1 Obtaining Allowable Regions

Consider the desired coupler curve described in (6.36) with three precision points and two trajectories. The response elements are chosen, such that they correspond to the coupler curve already obtained from the design parameters

given in (6.11). The desired response elements are plotted in Figure 6.8.

$$
\begin{aligned}
\mathcal{P}_1 &= \{[0.24, 0.26], [0.323706, 0.343706], [-\pi, \pi], [-\pi, \pi]\} \\
\mathcal{P}_2 &= \{[0.19, 0.21], [0.373706, 0.393706], [-\pi, \pi], [-\pi, \pi]\} \\
\mathcal{P}_3 &= \{[0.14, 0.16], [0.333706, 0.353706], [-\pi, \pi], [-\pi, \pi]\} \\
f([C_x], [C_y])_1 &= \{(C_x, C_y) \mid C_x = t,\ C_y = \alpha - 0.065,\ t \in [\underline{t}, \overline{t}]\} \\
\mathcal{T}_1 &= \{f([C_x], [C_y])_1, [-\pi, \pi], [-\pi, \pi], [-0.01, 0.01], [0.13, 0.17], 0.005\} \\
\mathcal{T}_2 &= \{f([C_x], [C_y])_1, [-\pi, \pi], [-\pi, \pi], [-0.01, 0.01], [0.19, 0.23], 0.005\}
\end{aligned}
\tag{6.36}
$$

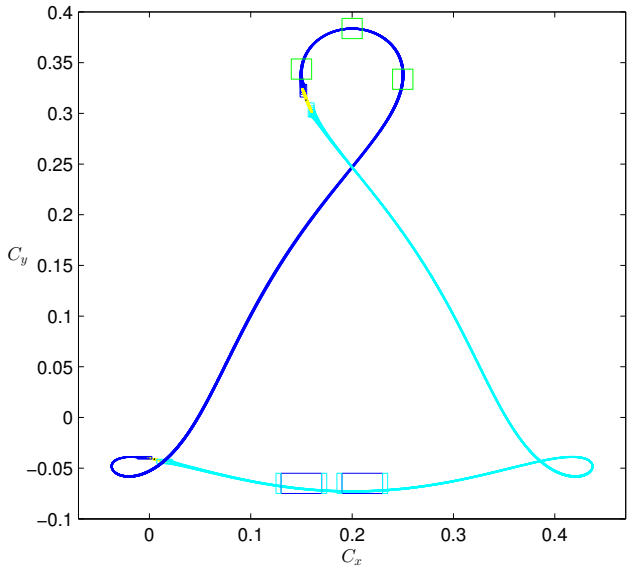


Figure 6.8: A plot of the desired task elements described in equation (6.36).

To demonstrate appropriate synthesis of the four-bar linkage, the domains of the design parameters $[p]$ and $[q]$ are initialized as $[-1.0, 1.0]$. The remaining

design parameters are the same as those in (6.11). The design parameter uncertainties are selected as $\Delta\mathcal{D} = 0.0005$, the bisection resolution is selected as $\beta = 0.0005$, and the solution resolution is selected as $\epsilon = 0.0000001$.

The allowable region considering the precision point task elements from (6.36) is shown in Figure 6.9. The design solutions do not consider restrictions on the branch configuration and any design which achieves the desired precision points, regardless of branch configuration is accepted as a solution (a single branch configuration may optionally be enforced). This generates several disconnected regions. Each allowable region is surrounded by unknown design intervals. The design solutions all have a classification of 0π-double-rocker. Figure 6.10a zooms in on one of the regions. An appropriate design solution may be chosen from inside of the allowable regions while considering the manufacturing uncertainties. Selecting the values of $[p] = [0.5699, 0.5701]$ and $[q] = [0.4299, 0.4301]$ from the allowable region of Figure 6.10a, the corresponding coupler curve and desired precision points are plotted in Figure 6.10b. The resulting linkage is classified as a 0π-double-rocker non-Grashof linkage, which has a single circuit containing two different branch configurations. The coupler curve passes through each of the precision points.

Considering only the trajectory elements from (6.36), Figure 6.11 shows a subset of the allowable region considering the first trajectory. This image shows that there exists a division in the design solutions. Some designs in this region correspond to a folding linkage, and such a linkage, while theoretically

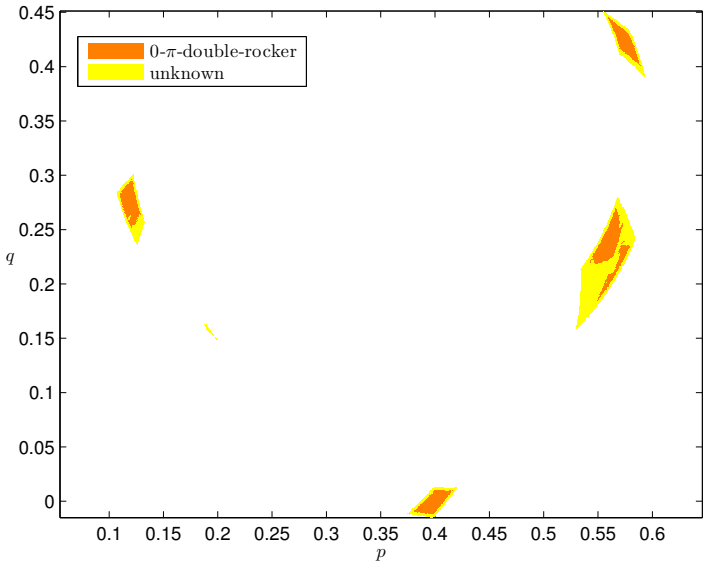
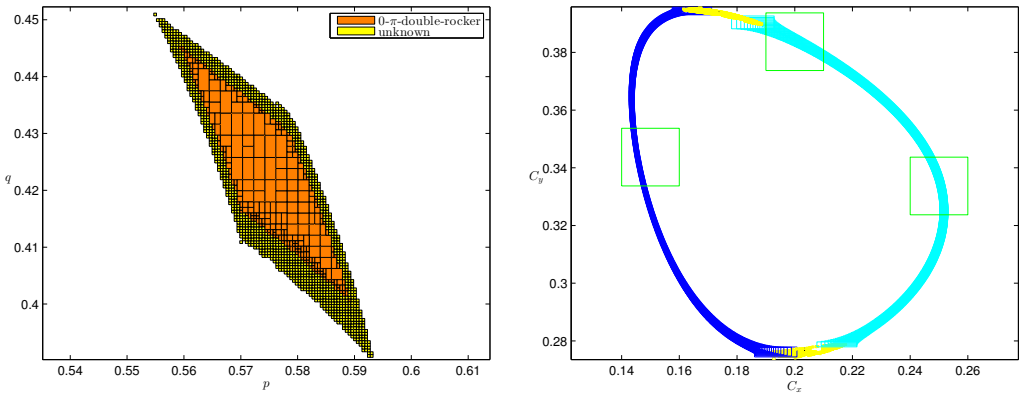


Figure 6.9: Appropriate design solutions for p and q for precision points response elements described in equation (6.36).

possible, it is impossible to manufacture as it can only be achieved using flexible links or introducing uncertainties into the joints. For this reason, folding linkages are not considered as design solutions in this work. Next, Figure 6.12 shows the allowable region considering both trajectories. Several disconnected regions are found. Selecting the values of $[p] = [0.2999, 0.3001]$ and $[q] = [0.0199, 0.0201]$ from the allowed region, the corresponding coupler curve and desired trajectories are plotted in Figure 6.13a. Selecting the values of $[p] = [0.2499, 0.2501]$ and $[q] = [-0.4401, -0.4399]$ from the allowed region, the corresponding coupler curve and desired trajectories are plotted in Figure 6.13b. These two coupler curves are quite different, yet are able to achieve the same objective.



(a) (b)

Figure 6.10: a) Zoomed-in view of one of the disconnected allowed regions from Figure 6.9; b) the coupler curve corresponding to a solution from the allowed region.

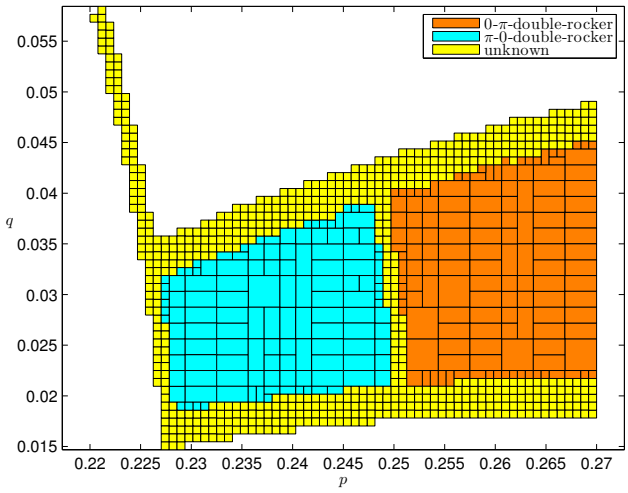


Figure 6.11: A subset of the design solutions for p and q for the first trajectory response element described in equation (6.36).

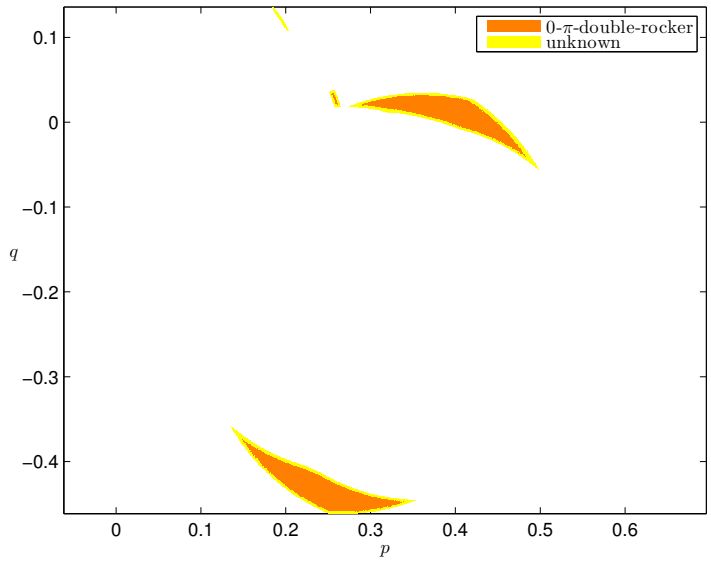
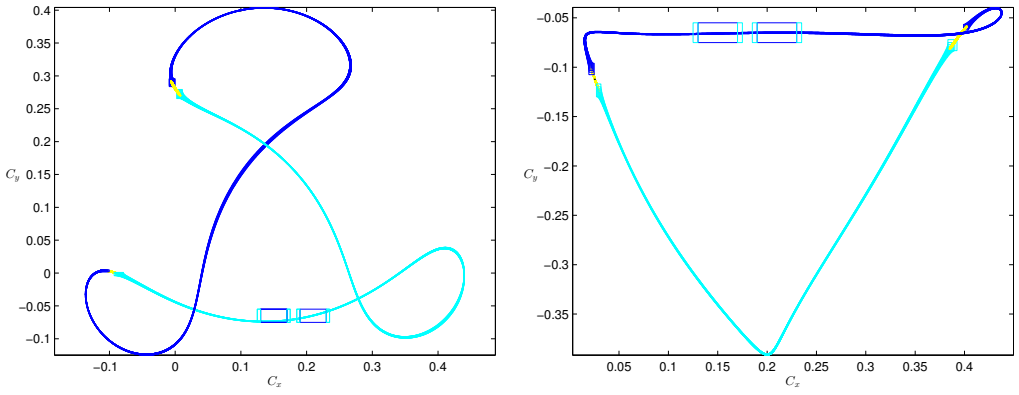


Figure 6.12: Design solutions for p and q for all trajectory response elements described in equation (6.36).



(a) (b)

Figure 6.13: a-b) Coupler curves corresponding to a design from the allowed region.

Using all task elements from (6.36), Figure 6.14 shows the complete set of appropriate design solutions to the problem. This is equal to the intersection of the allowed regions from the precision point solutions and trajectory solutions. The original design parameters from (6.11) are contained inside the allowed region. This is an important point. If a set of requirements are selected for an appropriate synthesis routine, and all of the requirements are simultaneously considered, then it may be possible that no appropriate design solutions are obtained because one (or several) of the requirements are too stringent. By considering all of the requirements simultaneously the user does not know which requirement needs to be relaxed. Another approach is to consider each requirement independently, then to find the intersection of the appropriate design solutions resulting from each requirement. This may also be approached in a consecutive manner, such that the appropriate design solutions from the first requirement are used as the starting point for the second requirement, as so on.

These examples help to demonstrate the usefulness of the appropriate design methodology. Additional constraints and response elements may always be added to further refine the allowed region. As well, the bisection resolutions may be reduced to improve the classification of unknown design intervals.

6.9.2 Finding Appropriate Design Solutions

The appropriate synthesis routine is capable of solving the full 9-dimensional problem to determine every possible linkage design that satisfies the desired

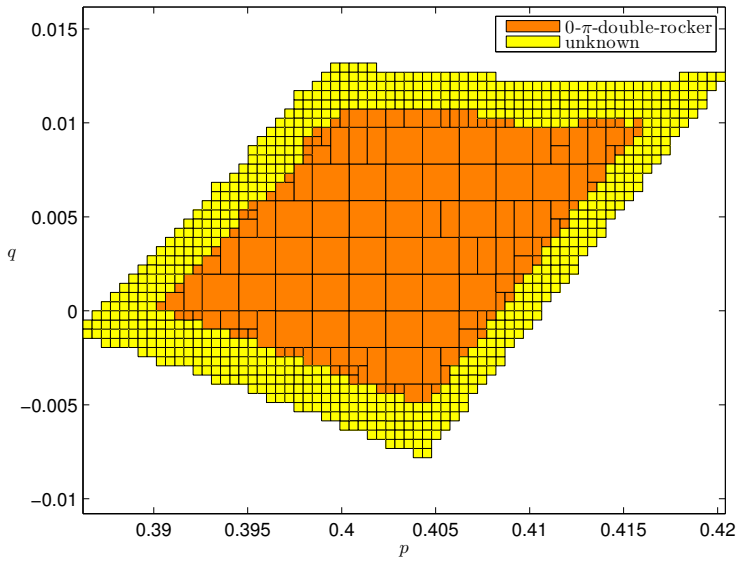


Figure 6.14: Design solutions for p and q for all response elements described in equation (6.36).

response; however, the full problem is generally very time consuming (on the order of days) and in most cases it is unnecessary to solve for every possible linkage design. The end-user may not require an infinite number of solutions. It may be desirable to determine several solutions to the problem, rather than obtaining the complete allowable region. Incorporating methods which are able to quickly identifying appropriate design solutions greatly improve the general usefulness of the appropriate synthesis routine.

Two techniques are presented to assist in more quickly identifying appropriate design solutions. One technique considers reordering the design list in such a way that boxes more likely to contain a solution are tested first. Another technique considers applying conventional global optimization to find

regions of interest, then inflating these regions and applying appropriate synthesis to determine appropriate design solutions. Both of these techniques require a value which describes the offset of the coupler curve from the desired response, termed the *constraint violation*.

6.9.3 Constraint Violation

The total constraint violation of a design for a given task is equal to the sum of the constraint violations for each task element. The constraint violation associated with a precision point task element may be described by the generalized volume of the interval hull of $([C_x], [C_y], [\theta], [\psi])_{actual}$ of the linkage and $([C_x], [C_y], [\theta], [\psi])_{desired}$ of the precision point. These domains will always intersect since a non-intersection results in a non-solution and is removed from the search. The constraint value is normalized by the generalized volume of the precision point, such that a value of 1.0 implies that a solution is found for the precision point. The constraint violation for a precision point i (V_{p_i}) is therefore described as:

$$V_{p_i} = \frac{volume(([C_x], [C_y], [\theta], [\psi])_{actual} \square ([C_x], [C_y], [\theta], [\psi])_{desired})}{volume(([C_x], [C_y], [\theta], [\psi])_{desired})} \tag{6.37}$$

and the constraint violation V_p associated with all n precision points is:

$$V_p = \sum_{i=1}^{n} V_{p_i} \tag{6.38}$$

For trajectories, the associated constraint violation may be described by the generalized volume of the interval hull of the actual $([C_x],[C_y],[t],[\alpha],[\theta],[\psi])_{actual}$ of the linkage and the desired $([C_x],[C_y],[t],[\alpha],[\theta],[\psi])_{desired}$ of the each trajectory end-point. Again, these domains will always intersect since a non-intersection results in a non-solution and is removed from the search. The constraint value is normalized by the generalized volume of the trajectory end-points. The constraint violation for an end-point j of trajectory i (V_{tij}) is therefore described as:

$$V_{tij} = \frac{volume(([C_x],[C_y],[t],[\alpha],[\theta],[\psi])_{actual} \square ([C_x],[C_y],[t],[\alpha],[\theta],[\psi])_{desired})}{volume(([C_x],[C_y],[t],[\alpha],[\theta],[\psi])_{desired})} \tag{6.39}$$

and the constraint violation V_t associated with all n trajectories is:

$$V_t = \sum_{i=1}^{n} (V_{ti1} + V_{ti2}) \tag{6.40}$$

The total constraint violation V for a given design is equal to the sum of V_p and V_t as

$$V = V_p + V_t \tag{6.41}$$

6.9.4 Reordering the Design List

The design list may be ordered by ascending values of constraint violation, such that the next design to be tested is the most likely to contain a solution. As well, global optimization routines may be used to minimize the constraint

violation in order to assist in finding appropriate design solutions.

To reorder the design list, the constraint violation is divided by a factor which is proportional to the size of the design parameter box. This ensures that the search remains exploratory unless a very promising box is discovered. Here, this factor is set as the sum of the design parameter widths, such that a larger width decreases the value of constraint violation. Following the bisection, the design list is ordered by ascending values of constraint violation. This modification is tested on the three precision point elements in Eq. (6.36) using the same design parameter domains in Eq. (6.11) with the inflated parameters $v = p = q = [-1, 1]$. The solutions are restricted to have the same branch and circuit through each precision point. Table 6.2 presents the first 5 appropriate design solutions obtained for $[v]$, $[p]$ and $[q]$. Considering the first solution, the values of of $[x]$, $[y]$, $[\psi]$ and $[\theta]$ associated with each precision point are given in Table 6.3.

6.9.5 Identifying Regions of Interest using Global Optimization

Global optimization routines can be applied to minimize the constraint violation in order to find appropriate design solutions. Here Differential Evolution (Price and Storn, 1997) is applied to minimize the constraint violation. The variant DE/rand/1/bin is selected with a population size of 50. The full search space of the design parameters is considered and is initialized as

$u = v = p = q = e = f = [-1, 1]$ and $r = s = c = [0.01, 1]$. Uncertainties on the design parameters are neglected during the optimization in order to more quickly detect solutions. When a solution is found, the design parameters, which are exact, are saved. Since appropriate design solutions are of interest, the exact design parameters are inflated. This generates the regions of interest. These regions of interest can then be used as the starting domains of the appropriate synthesis routine. This initialization procedure quickly narrows in on a region which is known to contain an exact solution, and is therefore also likely to contain appropriate design solutions. Table 6.4 presents several different solutions returned by the optimization routine. Each solution has the same branch and circuit at each precision point.

Selecting the second design solution from Table 6.4, then inflating the design parameters and running the appropriate synthesis routine yields several solutions. Table 6.5 gives three adjacent appropriate design solutions. Figure 6.15 shows the coupler curve and desired response elements corresponding to the first appropriate design solution from Table 6.5.

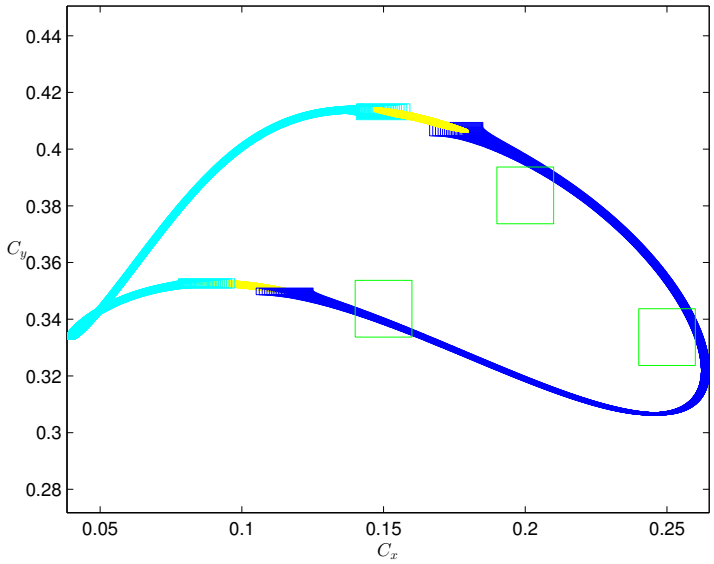


Figure 6.15: The coupler curve and desired response elements corresponding to the first appropriate design solution from Table 6.5.

Table 6.2: Appropriate solutions for parameters v, p and q of the four-bar linkage given three precision points.

	$[v]$	$[p]$	$[q]$
1	$[0.318359375, 0.3193359375]$	$[0.1875, 0.1884765625]$	$[0.1572265625, 0.158203125]$
2	$[0.3173828125, 0.318359375]$	$[0.1875, 0.1884765625]$	$[0.15625, 0.1572265625]$
3	$[0.318359375, 0.3193359375]$	$[0.189453125, 0.1904296875]$	$[0.15625, 0.1572265625]$
4	$[0.31640625, 0.3173828125]$	$[0.189453125, 0.1904296875]$	$[0.15625, 0.1572265625]$
5	$[0.3173828125, 0.318359375]$	$[0.1875, 0.1884765625]$	$[0.1572265625, 0.158203125]$

Table 6.3: The corresponding values of x, y, ψ and θ of the four-bar linkage for solution 1 of Table 6.2.

	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$
$[x]$	$[0.2461904336, 0.2596146262]$	$[0.1950530965, 0.1988727108]$	$[0.1526794322, 0.1576601016]$
$[y]$	$[0.3332982461, 0.3434695714]$	$[0.38770475, 0.3906041114]$	$[0.3372621532, 0.3402846851]$
$[\psi]$	$[-2.06897068, -2.06768783]$	$[-0.535834093, -0.5294028677]$	$[-0.8153834589, -0.8069989005]$
$[\theta]$	$[0.9093361389, 0.9104831748]$	$[-0.5734602361, -0.5668668546]$	$[-0.8520921674, -0.8434970458]$

Table 6.4: Exact design solutions for all nine design parameters of the four-bar linkage given three precision points.

	1	2	3
$[u]$	$[-0.9289111486, -0.9289111486]$	$[0.2211662638, 0.2211662638]$	$[-0.2682893589, -0.2682893589]$
$[v]$	$[-0.093204005, -0.093204005]$	$[0.9204238355, 0.9204238355]$	$[0.6630814969, 0.6630814969]$
$[p]$	$[0.2910117265, 0.2910117265]$	$[0.3546410697, 0.3546410697]$	$[0.6676501777, 0.6676501777]$
$[q]$	$[0.3198877927, 0.3198877927]$	$[-0.9982231968, -0.9982231968]$	$[-0.6886899306, -0.6886899306]$
$[r]$	$[0.09514724012, 0.09514724012]$	$[0.07060766208, 0.07060766208]$	$[0.0963348742, 0.0963348742]$
$[s]$	$[0.9363144014, 0.9363144014]$	$[0.2527432702, 0.2527432702]$	$[0.8318085416, 0.8318085416]$
$[c]$	$[0.4516096883, 0.4516096883]$	$[0.8053404545, 0.8053404545]$	$[0.7944526478, 0.7944526478]$
$[e]$	$[-0.8571114422, -0.8571114422]$	$[0.4754641102, 0.4754641102]$	$[0.3859080303, 0.3859080303]$
$[f]$	$[0.9240155799, 0.9240155799]$	$[-0.2637536855, -0.2637536855]$	$[-0.3639535806, -0.3639535806]$

Table 6.5: Appropriate design solutions for all nine design parameters given three precision points.

	1	2	3
$[u]$	$[0.2210662638, 0.2212662638]$	$[0.2210662638, 0.2212662638]$	$[0.2210662638, 0.2212662638]$
$[v]$	$[0.9203238355, 0.9205238355]$	$[0.9203238355, 0.9205238355]$	$[0.9203238355, 0.9205238355]$
$[p]$	$[0.3512660697, 0.3518910697]$	$[0.3506410697, 0.3512660697]$	$[0.3506410697, 0.3512660697]$
$[q]$	$[-1.000223197, -0.9999106968]$	$[-1.000223197, -0.9999106968]$	$[-0.9999106968, -0.9995981968]$
$[r]$	$[0.07050766208, 0.07070766208]$	$[0.07050766208, 0.07070766208]$	$[0.07050766208, 0.07070766208]$
$[s]$	$[0.2526432702, 0.2528432702]$	$[0.2526432702, 0.2528432702]$	$[0.2526432702, 0.2528432702]$
$[c]$	$[0.8052404545, 0.8054404545]$	$[0.8052404545, 0.8054404545]$	$[0.8052404545, 0.8054404545]$
$[e]$	$[0.4753641102, 0.4755641102]$	$[0.4753641102, 0.4755641102]$	$[0.4753641102, 0.4755641102]$
$[f]$	$[-0.2638536855, -0.2636536855]$	$[-0.2638536855, -0.2636536855]$	$[-0.2638536855, -0.2636536855]$

Chapter 7

Appropriate Synthesis: The 3-RRR Planar Parallel Mechanism

In this chapter, the 3-RRR parallel mechanism is considered. The goal of appropriate synthesis applied to the 3-RRR mechanism is to determine the appropriate design parameters which allow the 3-RRR mechanism to accomplished a given task, consisting of a desired task workspace $\mathbb{P}_{TW}$ and desired wrench capabilities $\mathcal{F}_{desired}$ (see Chapter 5).

The conventional method of discretizing the mechanism's workspace (see Appendix F) and computing some criteria over the discretized poses within the workspace provides a good estimate for a design solution, but it is not a reliable approach. A method to determine the complete set of appropriate design

solutions is presented in this chapter, which combines: interval kinematics routines (Chapter 3), collision detection (Chapter 4), and wrench capability analysis (Chapter 5).

7.1 A Prototype Mechanism and its Associated Constraints

A prototype 3-$\underline{\text{R}}$RR mechanism has been designed and fabricated for use with experimental research (see Figure 7.1). The concept uses optical breadboards with under-mounted actuators. Through-holes for the drive shafts are organized in a checker-board arrangement (not shown in the image). Distal-distal, proximal-proximal, and proximal-platform collisions may occur in the prototype, and therefore must be accounted for. Each actuator requires an axis-aligned square region to accommodate the actuator mounting bracket. The separation between adjacent actuators must also be considered. As well, due to the actuator drives shafts, it is also necessary for the mounting of the actuators to be separated from the trajectories of the task. Methods from the previous chapters are incorporated, such that appropriate design solutions are subject to the following constraints:

- Every pose in the desired trajectory, described by the task workspace $\mathbb{P}_{TW}$, must reachable (see Chapter 3) and free of self-collisions (see Chapter 4);

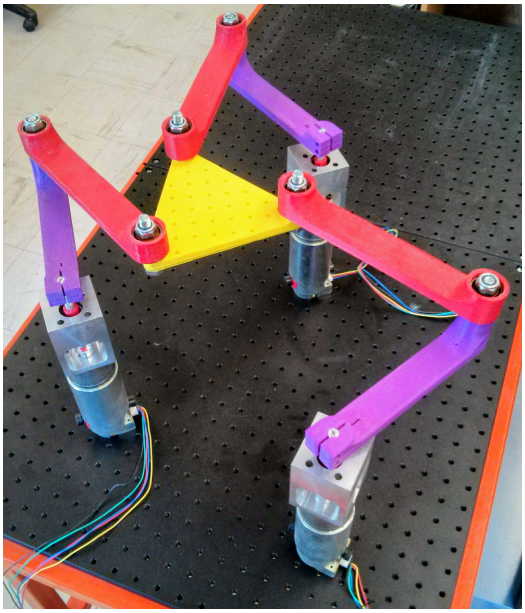


Figure 7.1: Prototype design of the 3-$\underline{\text{R}}$RR with relocatable actuators.

- Every pose in the desired trajectory ($\mathbb{P}_{TW}$) must satisfy the desired wrench capabilities (see Chapter 5);
- Actuators must be separated from the trajectory by a minimum distance, and adjacent actuators must be separated by a minimum distance.

7.2 Appropriate Design Solutions for Actuator Locations

An appropriate synthesis routine is developed for determining the allowable region of appropriate design solutions from the initial design parameter space

$[\mathcal{D}]$. An appropriate design solution must satisfy a given task described by the desired wrench capabilities $\mathcal{F}_{desired}$ and the desired trajectory of task workspace $\mathbb{P}_{TW}$. The appropriate synthesis routine is described in Algorithm 12.

A list $\mathcal{L}_{test}$ is populated by the initial appropriate design parameters $[\mathcal{D}]$. The algorithm calls the design parameter initialization routines *TRAJECTORY_SPACING* and *ACTUATOR_SPACING* to refine the search space of the design parameters. The list $\mathcal{L}_{test}$ is updated with the refined search space. Next, a workspace verification routine called *VERIFY_REACHABLE* checks the reachability and self-collisions of the task workspace $\mathbb{P}_{TW}$ for each set of appropriate design parameters $[\mathcal{D}]$ in the list $\mathcal{L}_{test}$. Appropriate design solutions are saved to the list $\mathcal{L}_{[\mathcal{D}]}$. The list $\mathcal{L}_{test}$ is updated with the solutions from $\mathcal{L}_{[\mathcal{D}]}$ and the list $\mathcal{L}_{[\mathcal{D}]}$ is reset to empty. Next, a wrench capability verification routine called *VERIFY_WRENCH_CAPABLE* checks the wrench capabilities for each set of appropriate design parameters $[\mathcal{D}]$ in the list $\mathcal{L}_{test}$. Appropriate design solutions are saved to the list $\mathcal{L}_{[\mathcal{D}]}$. These solutions are able to guarantee that every pose in the task workspace $\mathbb{P}_{TW}$ is reachable, self-collision free, and wrench-capable. Additional considerations may also be added (*e.g.*, singularities).

Algorithm 12: Appropriate synthesis routine for the 3-RRR mechanism.

function APPROPRIATE_SYNTHESIS ($[\mathcal{D}], \mathcal{F}_{desired}, \mathbb{P}_{TW}, \Delta\mathcal{D}, \beta$);

Input : appropriate design $[\mathcal{D}]$, desired wrench capabilities $\mathcal{F}_{desired}$, desired task workspace $\mathbb{P}_{TW}$, design parameter uncertainties $\Delta\mathcal{D}$, desired bisection resolution β

Output: set of solutions $\mathcal{L}_{[\mathcal{D}]}$, non-solutions $\mathcal{L}_{[\mathcal{D}]_n}$, and boundaries $\mathcal{L}_{[\mathcal{D}]_b}$

$\mathcal{L}_{test} \leftarrow [\mathcal{D}]$;
$\mathcal{L}_{test} = TRAJECTORY\_SPACING(\mathcal{L}_{test})$;
$\mathcal{L}_{test} = ACTUATOR\_SPACING(\mathcal{L}_{test})$;
while $\mathcal{L}_{test}$ *is not empty* **do**
    Pop $[\mathcal{D}]$ from $\mathcal{L}_{test}$;
    $reachable = VERIFY\_REACHABLE([\mathcal{D}], \beta)$;
    if $reachable == 1$ **then**
        $\mathcal{L}_{[\mathcal{D}]} \leftarrow [\mathcal{D}]$; `/* Save design to solution set */`
    else if $reachable == -1$ **then**
        $\mathcal{L}_{[\mathcal{D}]_n} \leftarrow [\mathcal{D}]$; `/* Save design to non-solution set */`
    else
        $(\mathcal{L}_{test}, \mathcal{L}_{[\mathcal{D}]_b}) \leftarrow BISECTION([\mathcal{D}], \Delta\mathcal{D})$; `/* Apply bisection */`
$\mathcal{L}_{test} \leftarrow \mathcal{L}_{[\mathcal{D}]}$;
$\mathcal{L}_{[\mathcal{D}]} = \mathcal{L}_{[\mathcal{D}]_b} = \mathcal{L}_{[\mathcal{D}]_n} = empty$;
while $\mathcal{L}_{test}$ *is not empty* **do**
    Pop $[\mathcal{D}]$ from $\mathcal{L}_{test}$;
    $capable = VERIFY\_WRENCH\_CAPABLE([\mathcal{D}], \beta)$;
    if $capable == 1$ **then**
        $\mathcal{L}_{[\mathcal{D}]} \leftarrow [\mathcal{D}]$; `/* Save design to solution set */`
    else if $capable == -1$ **then**
        $\mathcal{L}_{[\mathcal{D}]_n} \leftarrow [\mathcal{D}]$; `/* Save design to non-solution set */`
    else
        $(\mathcal{L}_{test}, \mathcal{L}_{[\mathcal{D}]_b}) \leftarrow BISECTION([\mathcal{D}], \Delta\mathcal{D})$; `/* Apply bisection */`

7.3 Case Studies

Several case studies are considered for the placement of one of more actuators.

7.3.1 Allowable Regions for the Placement of a Single Actuator

Initially, the design (7.1) is considered (see Figure 3.1), such that uncertainties are neglected on all parameters except the appropriate actuator locations $[\mathbf{a}_3]$.

$$[\mathcal{D}] = \left\{ \begin{array}{l} \mathbf{a}_1 = (0.3, 0.0)^T \text{ m}; \\ \mathbf{a}_2 = (0.0, 0.3)^T \text{ m}; \\ [\mathbf{a}_3] = ([-1.0, 1.0], [-1.0, 1.0])^T \text{ m}; \\ \mathbf{d}_1 = (0.1, 0.0)^T \text{ m}; \\ \mathbf{d}_2 = (-0.05, 0.0866)^T \text{ m}; \\ \mathbf{d}_3 = (-0.05, -0.0866)^T \text{ m}; \\ r_1 = 0.2 \text{ m}, r_2 = 0.3 \text{ m}, r_3 = 0.4 \text{ m}; \\ l_1 = 0.2 \text{ m}, l_2 = 0.2 \text{ m}, l_3 = 0.1 \text{ m}; \\ w_r = 0.015 \text{ m}, w_l = 0.015 \text{ m}, w_p = 0.015 \text{ m}; \end{array} \right\} \tag{7.1}$$

A trajectory generation routine is developed to build trajectories from curved and linear segments, which are then converted to a set of tight bounding boxes which slightly overestimate the desired trajectories. If all of the boxes satisfy the previous constraints, then the desired trajectories are also satisfied. Con-

sider the desired trajectory described by four linear segments, as shown in Figure 7.2. It is assumed that the desired wrench capabilities over the trajectory is $\mathcal{F}_{desired} = (f_x = [-10.0, 10.0]\ \text{N}, f_y = [-10.0, 10.0]\ \text{N}, m_z = [0.0]\ \text{Nm})$.

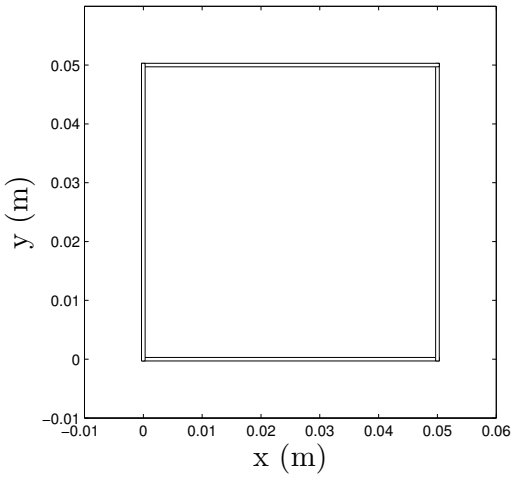


Figure 7.2: The desired trajectory selected for the 3-RRR.

The design parameter search space is initialized with two routines, described as follows:

1. A routine removes actuator locations which are within a distance of $\delta_{trajectory}$ along either axis of the trajectory boxes. See Figure 7.3.

2. A routine removes actuator locations which are within a distance of $\delta_{actuator}$ along either axis of another actuator. See Figure 7.4.

A collision-detection routine then removes actuator locations which result in any pose in the trajectory being unreachable and/or containing self-collisions.

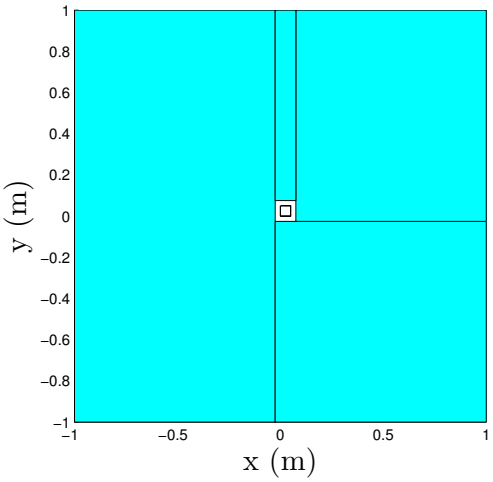


Figure 7.3: Search space for $[\mathbf{a}_3]$ considering trajectory separation for $\delta_{trajectory} = 0.025$ m.

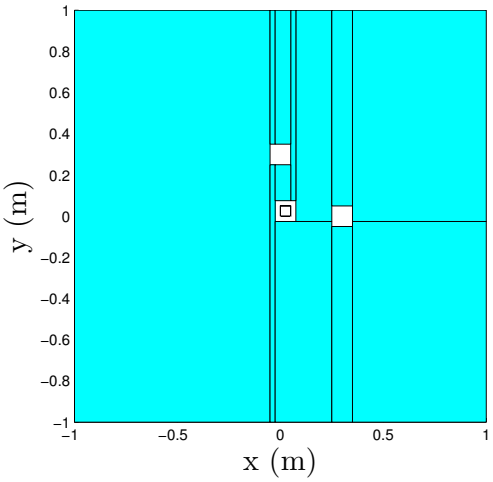


Figure 7.4: Search space for $[\mathbf{a}_3]$ considering trajectory and actuator separations for $\delta_{trajectory} = 0.025$m and $\delta_{actuator} = 0.05$m.

Figure 7.5 shows the appropriate solutions for $[\mathbf{a}_3]$ which yield a reachable and self-collision-free trajectory for a resolution of 0.002m. Three separated

solution regions are found. If the actuator is placed such that $[\mathbf{a}_3]$ is a subset of the appropriate design solutions described by the allowable regions, then the mechanism is guaranteed to able to reach every pose in the trajectory without self-collisions. As well, the previous actuator constraints are satisfied.

A wrench capability routine then refines the previous allowable regions to remove actuator locations which result in any pose in the trajectory not satisfying $\mathcal{F}_{desired}$. Figure 7.6 shows the appropriate solutions for $[\mathbf{a}_3]$ which yield a wrench-capable trajectory for a resolution of 0.002m. The three previous solution regions are refined. If the actuator is placed such that $[\mathbf{a}_3]$ is a subset of the refined design solutions, then the mechanism is able to reach and generate the desired wrench capabilities for every pose in the trajectory without self-collisions.

Several improvements may be applied to the reachable and wrench capability routines to improve performance. If a pose interval returns a boundary classification, then the remaining pose intervals in the trajectory can be sampled to attempt to find a pose interval with an outside classification. This is because it is not necessary to test every pose once a boundary pose interval has been obtained. Another improvement may be made if a design interval returns an outside classification. The pose interval which resulted in the outside classification will also likely return an outside classification for adjacent design intervals. Therefore, the outside pose interval may be added to the top of the trajectory list, such that it is the first pose to be tested on adjacent

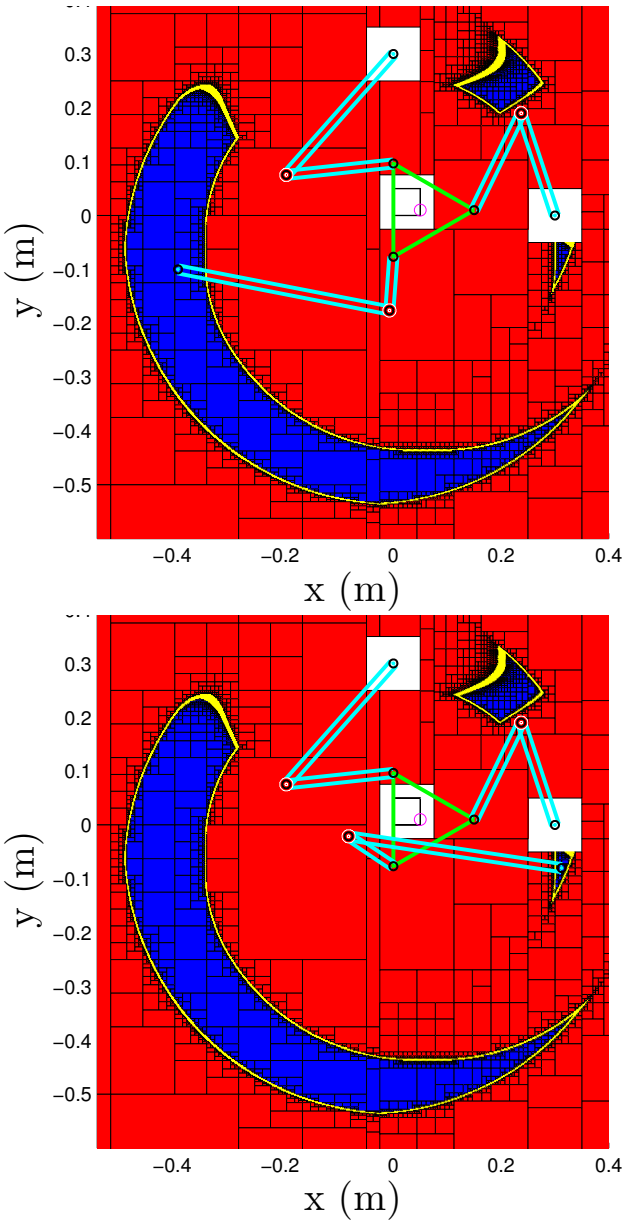


Figure 7.5: Design solutions for $[\mathbf{a}_3]$ which yield a reachable trajectory ensuring no self-collisions.

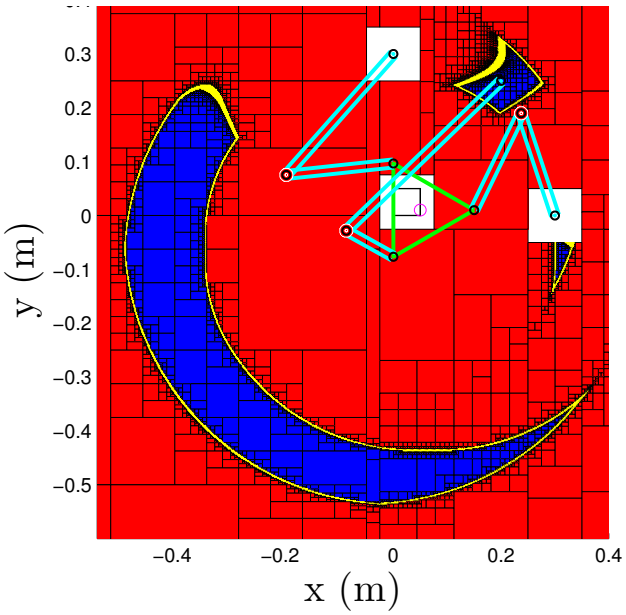


Figure 7.5: Continued. Design solutions for $[\mathbf{a}_3]$ which yield a reachable trajectory ensuring no self-collisions.

design intervals. If the desired resolution of the design parameter space is selected such that the design is minimal (*i.e.*, $width([\mathcal{D}]) = 2\Delta\mathcal{D}$), then the design sampling improvements for the outside wrench workspace test may be applied.

7.3.2 Allowable Regions for the Placement of All Actuators

Suppose now that it is desirable to know where to locate each of the three actuators. The design parameters $\mathbf{a}_1$ and $\mathbf{a}_2$ become interval vectors. If $\mathcal{L}_{[\mathcal{D}]}$ is the list of appropriate design solutions $[\mathcal{D}]$, then for each $[\mathcal{D}] \in \mathcal{L}_{[\mathcal{D}]}$ the

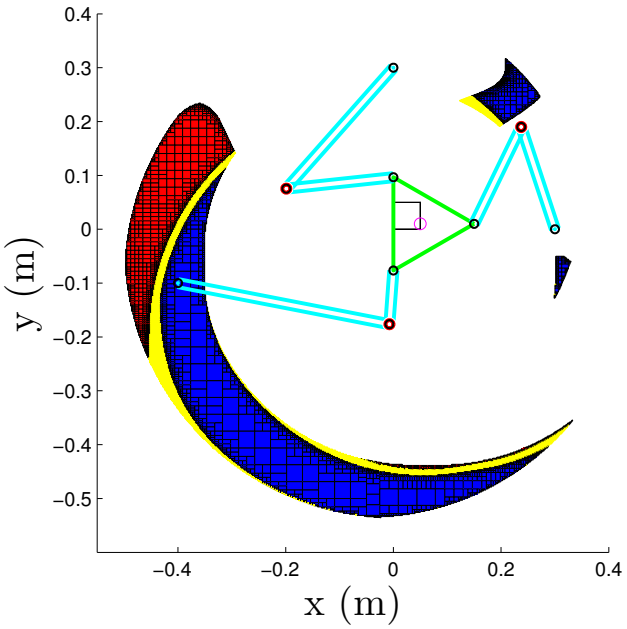


Figure 7.6: Design solutions for $[\mathbf{a}_3]$ which yield a wrench-capable trajectory.

reachability of the task is satisfied when:

$$\forall \mathbf{a}_1 \in [\mathbf{a}_1],\ \forall \mathbf{a}_2 \in [\mathbf{a}_2],\ \forall \mathbf{a}_3 \in [\mathbf{a}_3],\ \mathbb{P}_{TW} \subseteq \mathbb{P}_{RW} \tag{7.2}$$

and each $\mathbf{p} \in \mathbb{P}_{TW}$ is free of self-collisions.

The wrench-capabilities of the task are satisfied when:

$$\forall \mathbf{a}_1 \in [\mathbf{a}_1],\ \forall \mathbf{a}_2 \in [\mathbf{a}_2],\ \forall \mathbf{a}_3 \in [\mathbf{a}_3],\ \mathbb{P}_{TW} \subseteq \mathbb{P}_{WW} \tag{7.3}$$

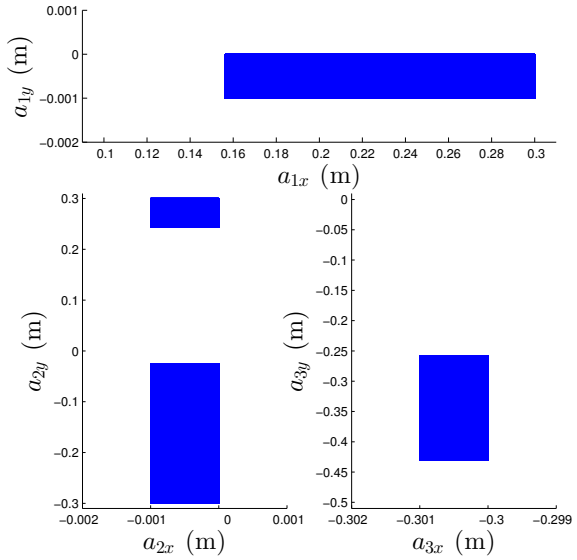
Consider the following design parameter domains for the previous task (Fig-

ure 7.2):

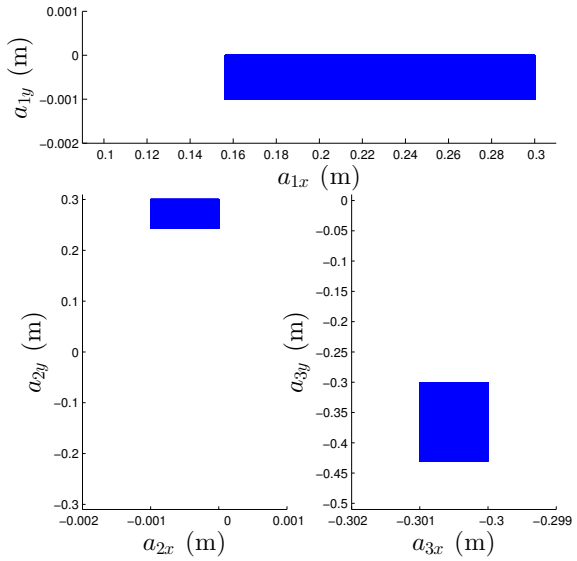
$$[\mathcal{D}] = \left\{ \begin{array}{l} [\mathbf{a}_1] = ([0.1, 0.3], [-0.001, 0.0])^T \text{m}; \\ [\mathbf{a}_2] = ([-0.001, 0.0], [-0.3, 0.3])^T \text{m}; \\ [\mathbf{a}_3] = ([-0.301, 0.3], [-0.5, 0.0])^T \text{m}; \\ \mathbf{d}_1 = (0.1, 0.0)^T \text{m}; \\ \mathbf{d}_2 = (-0.05, 0.0866)^T \text{m}; \\ \mathbf{d}_3 = (-0.05, -0.0866)^T \text{m}; \\ r_1 = 0.2\text{m}, r_2 = 0.3\text{m}, r_3 = 0.4\text{m}; \\ l_1 = 0.2\text{m}, l_2 = 0.2\text{m}, l_3 = 0.1\text{m}; \\ w_r = 0.015\text{m}, w_l = 0.015\text{m}, w_p = 0.015\text{m}; \end{array} \right\} \tag{7.4}$$

Limb 1 and 2 have elbow-right configurations, while limb 3 has elbow-left configuration.

The solutions may be presented by plotting projections of the inside boxes for $[\mathbf{a}_1]$, $[\mathbf{a}_2]$, and $[\mathbf{a}_3]$. A resolution of 0.005 m is used to generate the solutions. Projections of the solutions considering a reachable task workspace are given in Figure 7.7a. Next, projections of the solutions considering a collision-free task workspace are given in Figure 7.7b. The updated actuator solutions for $[\mathbf{a}_2]$ and $[\mathbf{a}_3]$ are significantly reduced when collisions are accounted for. The projected solutions of $[\mathbf{a}_1]$ are not affected by collisions. Lastly, projections of the solutions considering a collision-free and wrench-capable task workspace are given in Figure 7.7c.

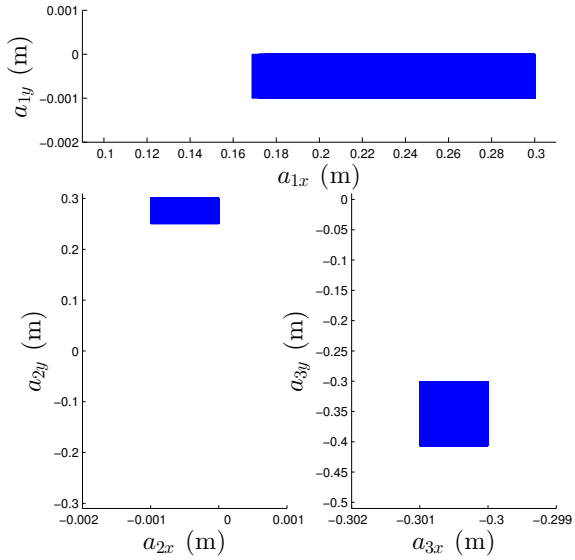


(a) Considering a reachable task workspace.



(b) Considering a collision-free task workspace.

Figure 7.7: Projected actuator location solutions.



(c) Collision-free and wrench-capable task workspace.

Figure 7.7: Continued. Projected actuator location solutions.

7.4 Identifying Design Solutions from High-Dimensional Design Spaces

When the number of design parameters is high, it becomes difficult to identify the design solutions, as is the case for the placement of all three actuators. A procedure which makes use of projections of the design parameters to lower-dimensional spaces allows to more easily identify all possible design solutions.

The procedure for viewing the solutions of high-dimensional design spaces is as follows. Select a point from one of the lower-dimensional projections of the

design parameters (*e.g.*, a given point $\mathbf{a}_3^*$ may be selected from Figure 7.7c). The selected point has one or more associated parent intervals which contain the selected point. That is, the associated parent intervals are the set of appropriate designs $\{[\mathcal{D}_i] \mid \mathbf{a}_3^* \in [\mathbf{a}_{3i}]\}$. Let $[\mathbf{a}_3^*]$ be the intersection of $[\mathbf{a}_{3i}]$ for the associated parent intervals. The lower-dimensional projections can be updated to show the refined solutions $[\mathbf{a}_{1i}]$, $[\mathbf{a}_{2i}]$, and $[\mathbf{a}_3^*]$ for each solution remaining solution $[\mathcal{D}_i]$. This process is repeated until all design parameters are considered.

For example, selecting $\mathbf{a}_3^* = (-0.3005, -0.31)$ from Figure 7.7c, the plots of the associated solutions are given in Figure 7.8. Next, from Figure 7.8, a point $\mathbf{a}_2^* = (0.0005, 0.29)$ may be selected. Let $[\mathbf{a}_2^*]$ be the intersection of $[\mathbf{a}_{2i}]$ for the associated parent intervals. The plots of the associated solutions are given in Figure 7.9. The design solutions presented in Figure 7.9 provide a detailed view of a particular region of the design solution space. A design solution for $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ may be selected as:

$$\begin{aligned} \mathbf{a}_1 &\in \bigcup_i [\mathbf{a}_{1i}], \ \forall \ [\mathcal{D}_i] \in \mathcal{L}_{[\mathcal{D}]} \\ \mathbf{a}_2 &\in [\mathbf{a}_2^*] \\ \mathbf{a}_3 &\in [\mathbf{a}_3^*] \end{aligned} \tag{7.5}$$

That is to say that any combination of actuator locations from Figure 7.9 may be used to satisfy the task requirements. Different regions of the design solution space may be considered by selecting different points during the

selection procedure.

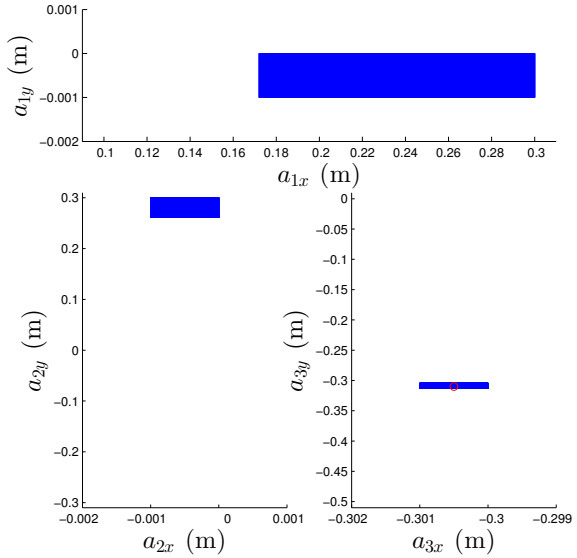


Figure 7.8: Projected actuator location solutions after selecting a desired location $\mathbf{a}_3 = (-0.402, 0.2)$.

As a second example, the selection procedure selects the points $\mathbf{a}_3 = (-0.3005, -0.4)$, then $\mathbf{a}_2 = (-0.0005, 0.26)$. The resulting associated solutions are given in Figure 7.10.

7.5 Sampling the Appropriate Design Solutions

Once the set of appropriate design solutions are obtained for a problem, it may be desirable to sample the design solutions in order to identify regions

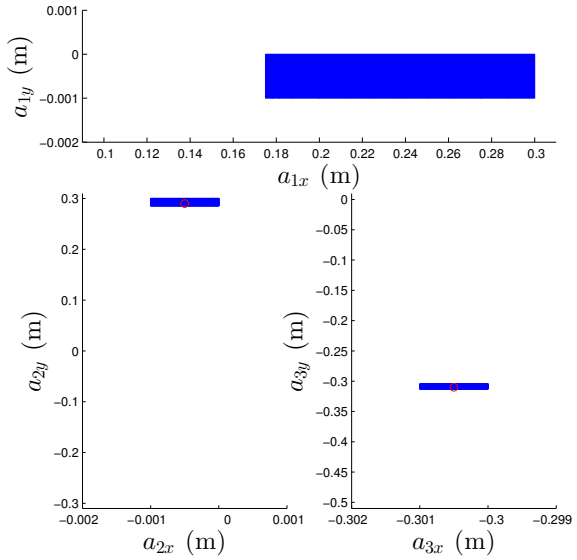


Figure 7.9: Projected actuator location solutions after selecting a desired locations $\mathbf{a}_3 = (-0.402, 0.2)$ and $\mathbf{a}_2 = (-0.002, 0.35)$.

of designs which may have benefits over other regions. With this approach, secondary objectives can be incorporated to describe the design solutions. Consider the previous example of determining the locations of all three actuators. Sample design solutions may be selected from Figures 7.9 and 7.10 as follows:

$$\begin{aligned} \mathbf{a}_1 &= (0.299, -0.0005)^T \text{m}; \\ \mathbf{a}_2 &= (-0.0005, 0.29)^T \text{m}; \\ \mathbf{a}_3 &= (-0.3005, -0.31)^T \text{m}; \end{aligned} \tag{7.6}$$

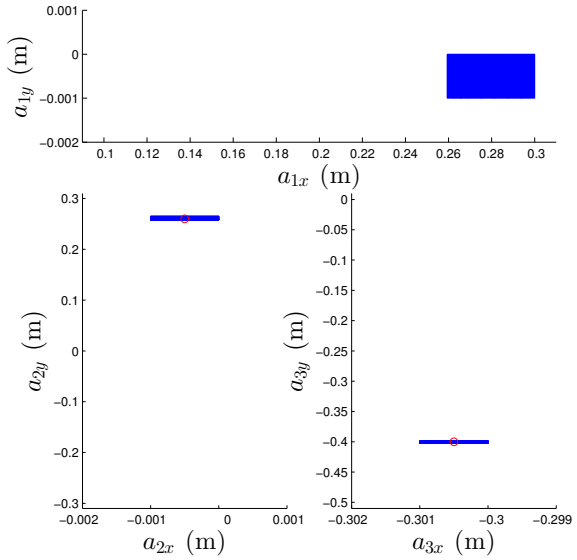


Figure 7.10: The projected actuator location solutions after selecting a desired locations $\mathbf{a}_3 = (-0.3005, -0.4)$, then $\mathbf{a}_2 = (-0.0005, 0.26)$.

$$\begin{aligned} \mathbf{a}_1 &= (0.26, -0.0005)^T \text{m}; \\ \mathbf{a}_2 &= (-0.0005, 0.26)^T \text{m}; \\ \mathbf{a}_3 &= (-0.3005, -0.4)^T \text{m}; \end{aligned} \tag{7.7}$$

The collision-free wrench workspace corresponding to each sample design solution is generated. Figure 7.11 corresponds to the design (7.6) and Figure 7.12 corresponds to the design (7.7).

For design (7.6), the collision-free workspace is more than 46.4% wrench capable, while only 18.4% is determined to not be wrench capable. The remainder has a boundary classification. Compared to design (7.7), design (7.6) is more compact using the criteria from Appendix F, Section F.4.

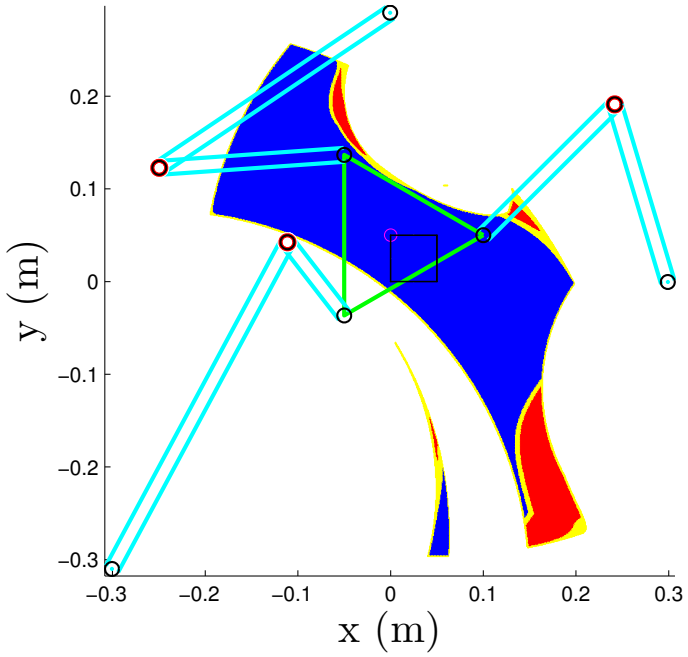


Figure 7.11: Collision-free wrench workspace corresponding to the design described in equation (7.6).

For design (7.7), the collision-free workspace is more than 49.5% wrench capable, while only 18.5% is determined to not be wrench capable. The remainder has a boundary classification. The inside poses surrounding the task trajectory form a larger region for design (7.7) compared to design (7.6), providing the opportunity of enlarging the trajectories of the desired task.

A mechanism designer may safely select any design from within the appropriate design solutions and be able to guarantee task completion. By sampling the design solution space, regions of designs with secondary benefits may be identified which allow the designer to determine the subset of design solu-

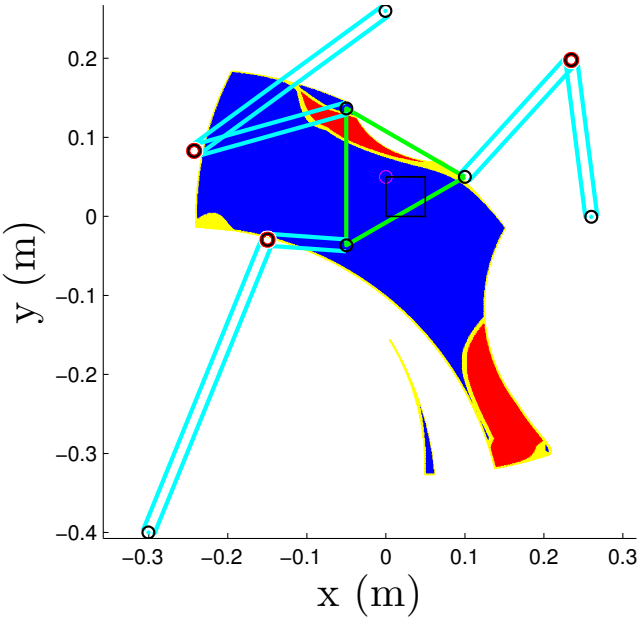


Figure 7.12: The collision-free wrench workspace corresponding to the design described in equation (7.7).

tions which are more desirable. For example, if a design is selected near to design (7.6), then a more compact design is obtained, whereas if a design near to design (7.7) is selected, a larger usable surrounding area is obtained. It is left up to the task as hand and the designer to decide which design is more suitable. The availability of parts and manufacturing capabilities may also play a role in the selecting of designs.

Chapter 8

Conclusions

This thesis has explored the effects of uncertainties in parallel mechanisms. Conventional mechanism design, which was termed an exact design, is unable to account for uncertainties which are present in mechanisms. These uncertainties arise from the fabrication and operation of a mechanism. With the assistance of interval analysis, the design description of a mechanism was improved, such that uncertainties present in a mechanism are modelled directly in the design parameters. The design description, accounting for uncertainties, was termed an appropriate design.

Analysis and synthesis methods applicable to parallel mechanisms with both exact designs and appropriate designs were developed. Analysis and synthesis methods which are able to be applied to appropriate designs were termed appropriate analysis and appropriate synthesis respectively. Appropriate analysis methods were developed and applied to several planar parallel

mechanisms for evaluating the forward and inverse kinematics, and determining singularity-free, collision-free, and wrench-capable poses. These methods are capable of generating reliable descriptions for the reachable workspaces, singularity-free workspaces, collision-free workspaces, and wrench workspaces of parallel mechanisms with appropriate designs.

Appropriate synthesis considers the allowable performance criteria for a desired task, without consideration for an objective function. Any design which satisfies the allowable performance criteria is saved as a design solution. This provides mechanism designers with a large range of potential designs, where each design is guaranteed to accomplish the desired task. Methods of more quickly exploiting the design parameter search space and sampling the appropriate design solutions were proposed. Appropriate synthesis was applied to the classical four-bar linkage, and the popular 3-$\underline{\text{R}}$RR planar parallel mechanism to demonstrate the capabilities of the method.

8.1 Appropriate Design

The concept of appropriate design introduces a powerful tool to mechanism designers. Rather than analyze a parallel mechanism with an exact design for kinematics, singularities, collisions, and wrench capabilities, only to later consider uncertainties, appropriate design describes a framework for accounting for uncertainties automatically within a routine. The conventional issues relating to a discretization of a workspace are overcome with an appropriate

design, where continuous spaces are treated the same as discretized spaces. Generating a good description for the appropriate design of a parallel mechanism is not without challenge. Interval analysis requires significant experience to use effectively, and extending problems for interval analysis typically requires reformulation and is rarely straight-forward. Once properly extended though, many mechanism problems can benefit from an appropriate design description.

The methods developed here for appropriate designs provide reliable results at the expense of computational time. A significant amount of work is still required to make the concept of appropriate design comparable in performance to state-of-the-art methods for exact designs. As well, to truly predict the performance of real mechanisms, spatial uncertainties must be incorporated into the appropriate design description of mechanisms. The inherent benefits of an appropriate design over an exact design should help to justify the continuation of this work.

8.2 Collision Detection for Appropriate Designs

Collision detection techniques for appropriate designs of parallel mechanisms are necessary to ensure that a pose or set of poses are always or never collision-free. The collision detection methods in the literature were unable to handle uncertainties in the design parameters of parallel mechanisms. The method

developed here is able to efficiently detect self-collisions in planar parallel mechanisms. The physical structure of the mechanism is able to be used to specify the possibility of collisions between links, in order to account for the layered designs of mechanisms.

The detection of collisions for appropriate designs of parallel mechanisms provides another useful tool. The complete set of poses resulting in self-collisions can be determined and saved to a list, where each element in the list is an interval vector. Then, rather than testing every pose along a trajectory for collisions, it is only necessary to determine if every pose is a subset of the collision-free poses in the list. Trajectory planning then becomes a much simpler problem, as a trajectory must simply remain in the collision-free workspace.

8.3 Wrench Capabilities

Wrench capabilities have been shown in the literature to be an important aspect of parallel mechanism design. Wrench capabilities may be effectively used to determine mechanism designs which satisfy a desired task. The determination of a mechanism design which satisfies the static requirements of a desired task is a first step of designing a mechanism. The dynamic capabilities of a mechanism may be considered following the study of the wrench capabilities.

Efficient techniques for evaluating the wrench capabilities for exact designs

have been developed in the literature. The issue with these techniques is that they are not reliable when uncertainties in the mechanism are considered. Methods for evaluating the wrench capabilities of parallel mechanisms with appropriate designs were developed. These methods are able to reliably determine the wrench workspace of a mechanism relating to a desired task.

8.4 Appropriate Synthesis

The determination of the complete set of design solutions which are guaranteed to accomplish a given problem is a significant accomplishment. Appropriate synthesis provides the mechanism designer with the freedom of selecting any design from a infinite set of possibilities. Uncertainties in the fabrication and operation of a mechanism are accounted for by properly formulating an appropriate design description. Discretization of the design and/or task space are not necessary, since appropriate synthesis is able to operate over continuous spaces. Thus, a solution obtained with appropriate synthesis is considered reliable. As long as all relevant sources of uncertainties are accounted for, the real mechanism produced from an appropriate synthesis solution is guaranteed to accomplish the desired task.

8.5 Future Work

- Additional work relating to appropriate design is to incorporate spatial uncertainties into the appropriate design description of mechanisms.

- Additional work relating to collision detection of appropriate designs is to build trajectory planning methods which utilize precomputed collision-free poses.

- Additional work relating to the wrench capabilities of parallel mechanisms is to extend the concept of wrench capabilities in order to account for the dynamics of the mechanism.

- Additional work relating to appropriate synthesis is to incorporate additional performance criteria. As well, techniques to reduce the computational time on high-dimensional problems must be explored. Appropriate synthesis is an ideal candidate for parallel computing techniques and significant speed-up is possible.

Bibliography

G. Alefeld and J. Herzberger. *Computer Science And Applied Mathematics: A Series of Monographs and Textbooks*. Academic Press, San Diego, 1983.

M. Arsenault and R. Boudreau. The synthesis of three-degree-of-freedom planar parallel mechanisms with revolute joints (3-Rrr) for an optimal singularity-free workspace. *Journal of Robotic Systems*, 21(5):259–274, 2004.

S. Bai and J. Angeles. Coupler-curve synthesis of four-bar linkages via a novel formulation. *Mechanism and Machine Theory*, 94:177–187, 2015.

C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4): 469–483, 1996.

M. Baumann. *A regularity criterion for interval matrices*, pages 45–50. Freiburg University, 1984.

F. Benhamou and L. Granvilliers. *Chapter 16–Continuous and Interval Constraints*, volume 2, pages 571–603. Elsevier, 2006.

F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget. Revising hull and box consistency. In *Proceedings of the 1999 International Conference on Logic Programming*, pages 230–244, Cambridge, MA, USA, 1999. Massachusetts Institute of Technology.

L. Blanchet and J.-P. Merlet. Interference detection for cable-driven parallel robots. In *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1413–1418, July 2014.

P. Bosscher, A. T. Riechel, and I. Ebert-Uphoff. Wrench-feasible workspace generation for cable-driven robots. *IEEE Transactions on Robotics*, 22(5): 890–902, October 2006.

S. Bouchard, C. M. Gosselin, and B. Moore. On the ability of a cable-driven robot to generate a prescribed set of wrenches. In *Proceedings of the 2008 ASME Design Engineering Technical Conferences*, page 12, New York, NY, USA, August 3-6 2008.

R. Boudreau and C. Gosselin. The synthesis of planar parallel manipulators with a genetic algorithm. *ASME. J. Mech. Des.*, 121(4):533–537, 1999.

G. Brown, 1984. URL `http://www.garrettcam.com/camsSkyCam.php`. [Accessed January 20, 2014].

R. R. Bulatović and S. R. Dordević. On the optimum synthesis of a four-bar linkage using differential evolution and method of variable controlled deviations. *Mechanism and Machine Theory*, 44(1):235–246, 2009.

J.A. Cabrera, A. Simon, and M. Prado. Optimal synthesis of mechanisms with genetic algorithms. *Mechanism and Machine Theory*, 37(10):1165–1177, 2002.

L. Campos, F. Bourbonnais, I. A. Bonev, and P. Bigras. Development of a five-bar parallel robot with large workspace. In *Proceedings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 917–922, Montreal, Quebec, Canada, August 15-18 2010.

S. Caro, F. Bennis, and P. Wenger. Tolerance Synthesis of Mechanisms: a Robust Design Approach. *Journal of Mechanical Design*, 127:86–94, January 2005.

J. A. Carretero and C. M. Gosselin. Wrench capabilities of cable-driven parallel mechanisms using wrench polytopes. In *Proceeding of the 2010 IFToMM Symposium on Mechanism Design for Robotics*, Universidad Panamericana, Mexico City, Mexico, September 28–30 2010.

J. A. Carretero, S. B. Nokleby, and F. Firmani. Force polyhedra for spatial redundant parallel manipulators. In N. Andreff, O. Company, M. Gouttefarde, and F. Krut, S.and Pierrot, editors, *Proceedings of the Second*

International Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators, page 12, Montpellier, France, September 21-22 2008.

D. Chablat and P. Wenger. Moveability and collision analysis for fully-parallel manipulators. In *Proceedings of the 1998 RoManSy*, pages 1–8, July 1998.

T. R. Chase and J. A. Mirth. Circuits and branches of single-degree-of-freedom planar linkages. *Journal of Mechanical Design–Transactions of the ASME*, 115(2):223–230, 1993.

G. B. Chatterjee and A. K. Mallik. Mechanical error of a four-bar linkage coupler curve. *Mechanism and Machine Theory*, 22(1):85–88, 1987.

G. Chen, H. Wang, and Z. Lin. A unified approach to the accuracy analysis of planar parallel manipulators both with input uncertainties and joint clearance. *Mechanism and Machine Theory*, 64:1–17, 2013.

R. Clavel. Device for the movement and positioning of an element in space, 11 1990. US Patent No. 4,976,582.

C. A. C. Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, January 2002. doi: 10.1016/S0045-7825(01)00323-1.

H. Collavizza, F. Delobel, and M. Rueher. Comparing partial consistencies. *Reliable Computing*, 5(3):213–228, 1999.

S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, and B. Bickel. Computational design of mechanical characters. *ACM Trans. Graph.*, 32(4):83:1–83:12, July 2013.

J. Cortés and T. Siméon. Probabilistic motion planning for parallel mechanisms. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 3, pages 4354–4359, Sept 2003.

S. Das, S. S. Mullick, and P. N. Suganthan. Recent advances in differential evolution – an updated survey. *Swarm and Evolutionary Computation*, 27: 1–30, 2016.

L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.

K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2):311–338, 2000.

I. Ebrahimi, J. A. Carretero, and R. Boudreau. Kinematic analysis and path planning of a new kinematically redundant planar parallel manipulator. *Robotica*, 26(3):405–413, May 2008.

Y. Fang and L.-W. Tsai. Structure synthesis of a class of 3-dof rotational parallel manipulators. *IEEE Transactions on Robotics and Automation*, 20(1):117–121, Feb 2004.

M. Fiedler, J. Nedoma, Ramik, J. Rohn, and K. Zimmermann. *Linear Optimization Problems with Inexact Data*. Springer US, 2006.

F. Firmani and R. P. Podhorodeski. Force-unconstrained poses for a redundantly-actuated planar parallel manipulator. *Mechanism and Machine Theory*, 39(5):459–476., 2004.

F. Firmani, A. Zibil, S. B. Nokleby, and R. P. Podhorodeski. Force-moment capabilities of revolute-jointed planar parallel manipulators with additional actuated branches. *Transactions of the Canadian Society for Mechanical Engineering*, 31(4):469–481, 2007.

F. Firmani, A. Zibil, R. P. Podhorodeski, and S. B. Nokleby. Wrench capabilities of planar parallel manipulators. Part I: Wrench polytopes and performance indices. *Robotica*, 26(6):791–802, November 2008a.

F. Firmani, A. Zibil, R. P. Podhorodeski, and S. B. Nokleby. Wrench capabilities of planar parallel manipulators. Part II: Redundancy and wrench workspace analysis. *Robotica*, 26(6):803–815, November 2008b.

F. Freudenstein and L. Dobrjanskyj. *On a theory for the type synthesis of mechanisms*, pages 420–428. Springer, Berlin, Heidelberg, 1966.

M. Gallant and R. Boudreau. The synthesis of planar parallel manipulators with prismatic joints for an optimal, singularity-free workspace. *Journal of Robotic Systems*, 19(1):13–24, 2002.

V. Garg. Force-moment capability analysis of redundantly actuated spatial parallel manipulators. Master's thesis, University of New Brunswick, Fredericton, New Brunswick, Canada, 2007.

V. Garg, J. A. Carretero, and S. B. Nokleby. A new method to calculate the force and moment workspaces of actuation redundant spatial parallel manipulators. *Journal of Mechanisms and Robotics*, 1(3):1–8, August 2009a.

V. Garg, S. B. Nokleby, and J. A. Carretero. Wrench capability analysis of redundantly-actuated spatial parallel manipulators. *Mechanism and Machine Theory*, 44(5):1070–1081, May 2009b.

G. Gogu. *Structural Synthesis of Parallel Robots: Part 1: Methodology*. Solid Mechanics and Its Applications. Springer Netherlands, 2009.

C. M. Gosselin and J. Angeles. Singularity analysis of closed-loop kinematic chains. *IEEE Transactions on Robotics and Automation*, 6(3):281–290, June 1990.

C. M. Gosselin and M. M. Guillot. The synthesis of manipulators with prescribed workspace. *ASME. J. Mech. Des.*, 113(4):451–455, 1991.

C. M. Gosselin and J. F. Hamel. The agile eye: a high-performance three-degree-of-freedom camera-orienting device. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, volume 1, pages 781–786, May 1994.

V. E. Gough. Contribution to discussion of papers on research in automobile stability, control and tyre performance. In *Proceedings of the Automotive Division of the Institution of Mechanical Engineers*, pages 392–395, 1956.

V. E. Gough and S. G. Whitehall. Universal tyre test machine. In *9th. International Congress of F.I.S.I.T.A. (Fédération Internationale des Sociétés d'Ingénieurs des Techniques de l'Automobile)*, volume 117, pages 117–135, May 1962.

V. Goulet, W. Li, H. Cheong, F. Iorio, and C.-G. Quimper. *Four-Bar Linkage Synthesis Using Non-convex Optimization*, pages 618–635. Springer International Publishing, 2016.

M. Gouttefarde and S. Krut. Characterization of parallel manipulator available wrench set facets. In J. Lenarčič and M. M. Stanišić, editors, *Advances in Robot Kinematics: Motion in Man and Machine*, Piran-Portorož, Slovenia, June 27-July 1 2010. Springer.

M. Gouttefarde, D. Daney, and J.-P. Merlet. Interval-analysis-based determination of the wrench-feasible workspace of parallel cable-driven robots. *IEEE Transactions on Robotics*, 27(1), 2011.

C. Han, J. Kim, J. Kim, and F.C. Park. Kinematic sensitivity analysis of the 3-upu parallel mechanism. *Mechanism and Machine Theory*, 37(8): 787–798, 2002.

E. Hansen. Global optimization using interval analysis: The one-dimensional case. *Journal of Optimization Theory and Applications*, 29(3):331–344, 1979.

E. Hansen. Global optimization using interval analysis — the multi-dimensional case. *Numerische Mathematik*, 34(3):247–270, 1980.

E. Hansen and R. Smith. Interval arithmetic in matrix computations, part ii. *SIAM Journal on Numerical Analysis*, 4(1):1–9, 1967.

E. Hansen and G. W. Walster. *Global Optimization Using Interval Analysis: Revised And Expanded*. Monographs and textbooks in pure and applied mathematics. CRC Press, 2003.

F. Hao and J.-P. Merlet. Multi-criteria optimal design of parallel manipulators based on interval analysis. *Mechanism and Machine Theory*, 40(2): 157–171, 2005.

B. Hayes. A lucid interval. *American Scientist*, 91(6):484–488, 2003.

G. Heindl, V. Kreinovich, and A.V. Lakeyev. Solving linear interval systems is np-hard even if we exclude overflow and underflow. *Reliable Computing*, 4:383–388, 1995.

J. A. Hrones and G. L. Nelson. *Analysis of the Four-Bar Linkage: Its Application to the Synthesis of Mechanisms*. Number 2. Technology Press, 1951.

Yang S Sun T Chetwynd DG Xie F. Huang T, Wang M. Force/motion transmissibility analysis of six degree of freedom parallel mechanisms. *ASME. J. Mechanisms Robotics*, 6(3):031010–1–031010–5, 2014.

K. H. Hunt. *Kinematic Geometry of Mechanisms*. Oxford engineering science series. Clarendon Press, 1978.

T. P. Jones and G. R. Dunlop. Analysis of rigid-body dynamics for closed loop mechanismsits application to a novel satellite tracking device. 217(4): 285–298, 2003.

S. A. Joshi and L.-W. Tsai. Jacobian analysis of limited-dof parallel manipulators. *Journal of Mechanical Design*, 124(2):254–258, June 2002. doi: 10.1115/1.1469549.

M. H. F. Kaloorazi, S. Esfahani, and M. T. Masouleh. Dimensional synthesis of planar parallel cable-driven parallel robots via interval analysis. Proceedings of the 2013 CCToMM Mechanisms, Machines, and Mechatronics Symposium, May 30–31 2013.

M. H. F. Kaloorazi, , M. T. Masouleh, and S. Caro. Collision-free workspace of parallel mechanisms based on an interval analysis approach. *Robotica*, 35(8):1–14, 2016a.

M. H. F. Kaloorazi, , M. T. Masouleh, and S. Caro. Determining the maximal singularity-free circle or sphere of parallel mechanisms using interval analysis. *Robotica*, 34(1):135–149, 2016b.

L. V. Kantorovich and G. E. Forsythe. *Functional analysis and applied mathematics*. NBS report. National Bureau of Standards, 1952.

R. Kearfott and N. I. Manuel. INTBIS, a portable interval newton/bisection package. *ACM Trans. on Mathematical Software*, 16(2):152–157, June 1990.

J. S. Ketchel and P. M. Larochelle. Self-collision detection in spatial closed chains. *Journal of Mechanical Design*, 130(092305):9, September 2008.

X. Kong and C. M. Gosselin. *Type Synthesis of Parallel Mechanisms*, volume 33 of *Springer Tracts in Advanced Robotics*. Springer, 2007.

J. Kotlarski, H. Abdellatif, T. Ortmaier, and B. Heimann. Enlarging the useable workspace of planar parallel robots using mechanisms of variable geometry. In *2009 ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots*, pages 63–72, June 2009.

R. Krawczyk. Newton-algorithmen zur bestimmung von nullstellen mit fehlerschranken. *Computing*, 4(3):187–201, 1969.

M. A. Laribi, L. Romdhane, and S. Zeghloul. Analysis and dimensional synthesis of the delta robot for a prescribed workspace. *Mechanism and Machine Theory*, 42(7):859–870, 2007.

O. Lhomme. Consistency techniques for numeric CSPs. In *Proceedings of the 13th International Joint Conference on Artifical Intelligence–Volume 1*, IJ-

CAI'93, pages 232–238, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

E. Loh and G. W. Walster. Rump's example revisited. *Reliable Computing*, 8(3):245–248, 2002.

J. McCarthy and G. Song Soh. *Geometric Design of Linkages*. Interdisciplinary Applied Mathematics. Springer, 2010.

J-P. Merlet. Trajectory verification in the workspace for parallel manipulators. *The International Journal of Robotics Research*, 13(4):326–333, 1994.

J.-P. Merlet. Redundant parallel manipulators. *Laboratory Robotics and Automation*, 8(1):17–24, 1996.

J.-P. Merlet. Solving the forward kinematics of a gough-type parallel manipulator with interval analysis. *International Journal of Robotics Research*, 23(3):221–235, 2004.

J.-P. Merlet. *Parallel Robots*. Solid Mechanics and Its Applications. Springer Netherlands, 2005.

J.-P. Merlet. *A C++ Algorithms Library of Interval Analysis for equation Systems Version 2.4*. The COPRIN project, 2007a.

J.-P. Merlet. A formal-numerical approach for robust in-workspace singularity detection. *IEEE Transactions on Robotics*, 23(3):393–402, June 2007b.

J.-P Merlet. Interval analysis for certified numerical solution of problems in robotics. *International Journal of Applied Mathematics and Computer Science*, 19(3):399–412, 2009.

J.-P. Merlet. *MARIONET, A Family of Modular Wire-Driven Parallel Robots*, pages 53–61. Springer Netherlands, Dordrecht, 2010.

J.-P. Merlet. *ALIAS-Maple: The Maple interface for ALIAS – Version 2.7*. The COPRIN project, 2012.

J.-P. Merlet and D. Daney. Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 942–947, Barcelona, Spain, April 2005.

J.-P. Merlet and D. Daney. Legs interference checking of parallel robots over a given workspace or trajectory. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 757–762, Orlando, FL, USA, May 2006.

J.-P. Merlet and D. Daney. *Smart Devices and Machines for Advanced Manufacturing*, chapter Appropriate Design of Parallel Manipulators, pages 1–25. Springer London, London, 2008.

J.-P. Merlet, C. M. Gosselin, and T. Huang. *Parallel Mechanisms*, pages 443–462. Springer International Publishing, 2016.

R. E. Moore. *Interval Arithmetic and Automatic Error Analysis in Digital Computing*. PhD thesis, Stanford, CA, USA, 1963.

R. E. Moore. *Interval analysis*. Prentice-Hall series in automatic computation. Prentice-Hall, 1966.

R. E. Moore. The dawning. *Reliable Computing*, 4(5):423–424, 1999.

R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, 3600 Market Street, 6th Floor, Philadelphia, PA, 19104-2688 USA, 2009.

A. Müeller. *Redundant Actuation of Parallel Manipulators*, pages 87–108. InTech, 2008.

G. Mullineux. Atlas of spherical four-bar mechanisms. *Mechanism and Machine Theory*, 46(11):1811–1823, 2011.

A. Neumaier. Linear interval equations. In *Proceedings of the International Symposium on Interval Mathematics on Interval Mathematics 1985*, pages 109–120, London, UK, 1985. Springer-Verlag.

B. Neveu and G. Trombettoni. Adaptive constructive interval disjunction. In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, pages 900–906, Herndon, VA, USA, Nov 2013.

Y. Ni, C. Shao, B. Zhang, and W. Guo. Error modeling and tolerance design

of a parallel manipulator with full-circle rotation. *Advances in Mechanical Engineering*, 8(5):1–16, 2016.

S. B. Nokleby, R. Fisher, R. P. Podhorodeski, and F. Firmani. Force capabilities of redundantly-actuated parallel manipulators. *Mechanism and Machine Theory*, 40(5):578 – 599, 2005.

D. Oetomo, D. Daney, B. Shirinsadeh, and J.-P. Merlet. An interval-based method for workspace analysis of planar flexure-jointed mechanism. 131, 2008.

J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 2nd edition, 1998.

E. Papadopoulos and Y. Gonthier. A framework for large-force task planning of mobile and redundant manipulators. *Journal of Robotic Systems*, 16(3): 151–162, 1999.

Y. Patel and P. George. Parallel manipulators application – a survey. 2(3): 57–64, 2012.

J. K. Pickard. Geometry-based method for determining the wrench capabilities of redundant parallel manipulators. In *Mechanical Engineering Graduate Students Conference (MEGSC8)*, Fredericton, Canada, November 8 2012a.

J. K. Pickard. Geometry-based method for determining the wrench capabilities of redundant parallel manipulators. In *Mechanical Engineering*

Graduate Students Conference Poster Presentation (MEGSC8), Fredericton, Canada, November 8 2012b.

J. K. Pickard and J. A. Carretero. Towards the appropriate synthesis of the four-bar linkage. *Transactions of the Canadian Society for Mechanical Engineering*, Accepted November, 2017 .

J. K. Pickard and J. A. Carretero. Wrench workspaces of redundantly actuated spatial parallel manipulators using minimum wrench capabilities. In *Proceeding of the 2013 CCToMM Mechanisms, Machines, and Mechatronics Symposium*, Montreal, Canada, May 30–31 2013a.

J. K. Pickard and J. A. Carretero. Applications of interval analysis techniques for solving constraint satisfaction problems. In *Mechanical Engineering Graduate Students Conference Poster Presentation (MEGSC9)*, Fredericton, Canada, October 10 2013b.

J. K. Pickard and J. A. Carretero. A comparison of wrench-feasible workspaces using interval analysis and convex linear mapping techniques. In *Proceedings of the Canadian Society for Mechanical Engineering International Congress 2014*, Toronto, ON, Canada, June 1–4 2014.

J. K. Pickard and J. A. Carretero. An interval analysis method for wrench workspace determination of parallel manipulator architectures. In *Proceeding of the 2015 CCToMM Mechanisms, Machines, and Mechatronics Symposium*, Ottawa, Canada, May 28–29 2015a.

J. K. Pickard and J. A. Carretero. Design optimisation of the 3-RRR planar parallel manipulator via wrench capability analysis. In *Proceedings of the 14th World Congress in Mechanism and Machine Science*, Taipei, Taiwan, October 25–30 2015b.

J. K. Pickard and J. A. Carretero. An interval analysis method for wrench workspace determination of parallel manipulator architectures. *Transactions of the Canadian Society for Mechanical Engineering*, 40(2):139–154, November 2015c.

J. K. Pickard, J. A. Carretero, and V. C. Bhavsar. Task-optimized cable-actuated planar parallel manipulator architecture and its concurrent implementation. In *Proceedings of the International Conference on High Performance Computing and Simulation (HPCS 2014)*, Bologna, Italy, July 21–25 2014.

J. K. Pickard, J. A. Carretero, and V. C. Bhavsar. On the convergence and origin bias of the teaching-learning-based-optimization algorithm. *Applied Soft Computing*, 46(1):115–127, September 2016a.

J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Accounting for tolerances in the design parameters of the 3-RRR. In *Proceeding of Advances in Robot Kinematics*, Grasse, France, June 27–30 2016b.

J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Towards the appropriate synthesis of the four-bar linkage. In *Proceedings of the 2017 CCToMM*

Mechanisms, Machines, and Mechatronics Symposium, Montreal, Canada, May 25–26 2017a.

J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Analysis methods for the 3-RRR with uncertainties in the design parameters. In Jadran Lenarčič and Jean-Pierre Merlet, editors, *Advances in Robot Kinematics 2016*, pages 109–118. Springer International Publishing, 2017b.

J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Appropriate analysis of the four-bar linkage. *Mechanism and Machine Theory*, Submitted 2018a.

J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Appropriate synthesis of the four-bar linkage. *Mechanism and Machine Theory*, Submitted 2018b.

W. L. G. Pollard. Spray painting machine, August 1940. US Patent No. 2,213,108.

K. Price and R. Storn. Differential evolution. *Dr. Dobb's Journal of Software Tools for Professional Programmer*, 22(4):18–24, April 1997.

L. B. Rall. *Computational solution of nonlinear operator equations*. Wiley, 1969.

R. S. Rao. Inverse kinematic solution of robot manipulators using interval analysis. *Journal of Mechanical Design*, 120(1):147–150, 1998.

S. Ratschan. Efficient solving of quantified inequality constraints over the real numbers. *CoRR*, cs.LO/0211016, 2002.

A. T. Riechel and I. Ebert-Uphoff. Force-feasible workspace analysis for underconstrained, point-mass cable robots. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 5, pages 4956–4962, April 2004.

S. Roberts. On three-bar motion in plane space. *Proceedings of the London Mathematical Society*, s1-7(1):14–23, 1875.

J. Rohn. Strong solvability of interval linear programming problems. *Computing*, 26(1):79–82, 1981.

J. Rohn. Systems of linear interval equations. *Linear Algebra and its Applications*, 126(1):39–78, 1989.

J. Rohn. Solvability of systems of interval linear equations and inequalities. pages 35–77, 2006.

B. Roth and F. Freudenstein. Synthesis of path-generating mechanisms by numerical methods. *ASME. J. Eng. Ind.*, 3(85):298–304, 1963.

S. M. Rump. Algebraic computation, numerical computation and verified inclusions. In Rainer Janßen, editor, *Trends in Computer Algebra*, pages 177–197. Springer Berlin Heidelberg, 1988.

H. Schröcker, M. L. Husty, and J. McCarthy. Kinematic mapping based assembly mode evaluation of planar four-bar mechanisms. *Journal of Mechanical Design*, 129(9):924–929, 2006.

S .P. Shary. On optimal solution of interval linear equations. *SIAM Journal on Numerical Analysis*, 32(2):610–630, 1995.

S. Skelboe. Computation of rational interval functions. *BIT Numerical Mathematics*, 14(1):87–95, 1974.

D. Stevenson. Floating-point working group, microprocessor standards subcommittee. IEEE standard for binary floating point arithmetic (IEEE/ANSI 754-1985). Technical report, IEEE, 1985.

D. Stewart. A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, 180(1):371–386, 1965.

R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, December 1997.

R. A. Tapia. The kantorovich theorem for newton's method. *The American Mathematical Monthly*, 78(4):389–392, 1971.

K.-L. Ting, J. Zhu, and D. Watkins. The effects of joint clearance on position and orientation deviation of linkages and manipulators. *Mechanism and Machine Theory*, 35(3):391–401, 2000.

K.-L. Ting, K.-L. Hsu, Z. Yu, and J. Wang. Clearance-induced output position uncertainty of planar linkages with revolute and prismatic joints. *Mechanism and Machine Theory*, 111(1):66–75, 2017.

G. Trombettoni and G. Chabert. *Constructive Interval Disjunction*, pages 635–650. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

L.-W. Tsai. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. Wiley-Interscience, John Wiley and Sons, Inc., New York, 1999.

V. Unruh and P. Krishnaswami. A computer-aided design technique for semi-automated infinite point coupler curve synthesis of four-bar linkages. *Journal of Mechanical Design*, 117(1):143–149, 1995.

C. W. Wampler, A. P. Morgan, and A. J. Sommese. Complete solution of the nine-point path synthesis problem for four-bar linkages. *Journal of Mechanical Design–Transactions of the ASME*, 114(1):153–159, 1992.

J. Wang and O. Masory. On the accuracy of a stewart platform - part i: The effect of manufacturing tolerances. In *Proceedings IEEE International Conference on Robotics and Automation*, volume 1, pages 114–120, May 1993.

F. Xi, A. Ross, and S. Lang. Exploring a re-configurable parallel robot for space applications. In *Proceeding of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space*, pages 1–8, St-Hubert, Quebec, Canada, 18-22 June 2001.

A. Zibil, F. Firmani, S. B. Nokleby, and R. P. Podhorodeski. An explicit method for determining the force-moment capabilities of redundantly ac-

tuated planar parallel manipulators. *Journal of Mechanical Design*, 129 (10):1046–1055, October 2007.

K. Zielinski and R. Laur. *Stopping Criteria for Differential Evolution in Constrained Single-Objective Optimization*, volume 143 of *Studies in Computational Intelligence*, pages 111–138. Springer Berlin Heidelberg, 2008.

Appendix A

Screw Theory

Plücker coordinates can be used to define a line in 3-dimensional Euclidean space. Given the direction of the line, a particle of unit mass moving along the line would create a moment about the origin. By definition, the moment vector is perpendicular to every displacement along the line; therefore, the dot product of direction and moment vectors yields zero. Together, the direction and moment vectors uniquely define a line.

A unit screw, denoted $\hat{\$}$, is a line in space having an associated pitch. It is represented by a six-dimensional vector of Plücker coordinates, also called the screw coordinates, as:

$$\hat{\$} = \begin{Bmatrix} \hat{\mathbf{l}} \\ \mathbf{l}_m + \lambda\hat{\mathbf{l}} \end{Bmatrix} \tag{A.1}$$

where $\hat{\mathbf{l}}$ is a unit vector describing the direction of the screw axis, $\mathbf{l}_m$ is the

moment of the screw axis about the origin of a reference frame ($\mathbf{l}_m = \mathbf{l}_0 \times \hat{\mathbf{l}}$, where $\mathbf{l}_0$ defines any point on the screw axis), and λ is the pitch of the screw. A *twist* is a unit screw having a certain intensity which describes a finite or infinitesimal displacement of a rigid body while a *wrench* is a unit screw having a certain intensity which describes a resultant force and moment acting at a point on a rigid body.

A.1 Twists

The pitch of a twist is equal to the ratio of translation to rotation. For a revolute joint, there is no translation and thus $\lambda = 0$. The unit screw for a revolute joint reduces to a zero pitch screw:

$$\hat{\$}_{revolute} = \begin{Bmatrix} \hat{\mathbf{l}} \\ \mathbf{l}_0 \times \hat{\mathbf{l}} \end{Bmatrix} \tag{A.2}$$

For a prismatic joint, there is no rotation and thus $\lambda = \inf$. The unit screw for a prismatic joint is defined as:

$$\hat{\$}_{prismatic} = \begin{Bmatrix} \mathbf{0} \\ \hat{\mathbf{l}} \end{Bmatrix} \tag{A.3}$$

The angular velocity $\boldsymbol{\omega}$ and linear velocity $\mathbf{v}$ of a point on a moving body can be assembled into a screw quantity called a twist. The twist, denoted

$\mathbf{V}$, is obtained by multiplying the unit screw by the twist intensity, t:

$$\mathbf{V} = \begin{Bmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{Bmatrix} = t\hat{\$} \tag{A.4}$$

where $t = \dot{\theta}$ for a revolute joint and $t = \dot{d}$ for a prismatic joint ($\dot{\theta}$ and $\dot{d}$ represent the angular and linear velocities of the revolute and prismatic joints respectively).

A.2 Wrenches

The pitch of a wrench is represented as the ratio of moment to force. The unit screw for a pure force where $\lambda = 0$ is:

$$\hat{\$}_{pure-force} = \begin{Bmatrix} \hat{\mathbf{l}} \\ \mathbf{l}_0 \times \hat{\mathbf{l}} \end{Bmatrix} \tag{A.5}$$

The unit screw for a pure moment where $\lambda = \inf$ is:

$$\hat{\$}_{pure-moment} = \begin{Bmatrix} \mathbf{0} \\ \hat{\mathbf{l}} \end{Bmatrix} \tag{A.6}$$

The resultant force $\mathbf{f}$ and moment $\mathbf{m}$ acting at a point on the body can be assembled into a screw quantity called a wrench. The wrench, denoted $\mathbf{F}$, is

obtained by multiplying the unit screw by the wrench intensity, w:

$$\mathbf{F} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{m} \end{Bmatrix} = w\hat{\$} \tag{A.7}$$

A.3 Reciprocity of Screws

If a wrench acts on a body undergoing an infinitesimal twist in such a way that the virtual work produced is equal to zero, the two screws are said to be reciprocal (Firmani and Podhorodeski, 2004), and the reciprocal product, denoted $\otimes$, of $\mathbf{V}$ and $\mathbf{F}$ will be zero:

$$\mathbf{V} \otimes \mathbf{F} = \boldsymbol{\omega} \cdot \mathbf{m} + \mathbf{v} \cdot \mathbf{f} = 0 \tag{A.8}$$

Tsai (1999) provides a proof for the reciprocal condition using the principle of virtual work. Since a unit screw requires five independent parameters to specify its location and pitch, there is a quadruple infinitude of screws reciprocal to a given screw; thus, all screws that are reciprocal to a single screw form a five-system in three-dimensional space. Any five screws selected from the five-system can be used as "wrenches" to constrain the rigid body to a single-degree-of-freedom motion. Reciprocal screws can be used to project the motion of the passive joints onto the actuated joints; thus, reciprocal screws aid in the force analysis of a parallel mechanism (Garg, 2007).

A.4 Screw-based Jacobian

The reciprocity of screws provides an effective method for obtaining the Jacobian of parallel mechanisms (Joshi and Tsai, 2002). All of the joints may be replaced with an equivalent set of simplified 1-DOF joints. Let $\hat{\$}_j$ represent the unit screws describing the motion of each active and passive joint in a particular limb, where $j = 1, \ldots, m_{a+p}$ (m_{a+p} represents the number of screws required to describe the motion of all active and passive joints in the limb). Also, let $\hat{\$}_{rk}$ represent a unit reciprocal screw for the kth actuated joint of the limb, where $k = 1, \ldots, m$ (m represents the number of screws required to describe the motion of the active joints in limb). Screw $\hat{\$}_{rk}$ is reciprocal to all other joint screws except $\hat{\$}_k$. Due to the properties of a reciprocal screw, the virtual work done by $\hat{\$}_{rk}$, projected to the other joints in the limb, is zero. The reciprocal product for the various joint combinations is given by:

$$\hat{\$}_{rk} \otimes \$_j = 0 \text{ for } j = 1, \ldots, m_{a+p},\ k = 1, \ldots, m,\ j \neq k \tag{A.9}$$

Considering each limb as a serial chain, the overall instantaneous twist expressed in terms of the reference frame $\{0\}$ for each limb is:

$$^0\mathbf{V}_p = \sum_{j=1}^{m} (t_j\ {}^0\hat{\$}_j) \tag{A.10}$$

Taking the reciprocal product of both sides of the equation with the reciprocal

screws $\hat{\$}_{rk}$, the following simplification is obtained:

$$ {}^0\hat{\$}_{rk} \otimes^0 \mathbf{V}_p = {}^0\ \hat{\$}_{rk} \otimes^0 \hat{\$}_k t_k \text{ for } k = 1, \ldots, m \tag{A.11} $$

These equations can be written in matrix form for all m active joints and n-DOF as:

$$ \mathbf{J_x}\ \dot{\mathbf{x}} = \mathbf{J_q}\dot{\mathbf{q}} \tag{A.12} $$

where $\mathbf{J_x}$ is the $m \times n$ direct Jacobian matrix; $\mathbf{J_q}$ is an $m \times m$ diagonal matrix; $\dot{\mathbf{x}}$ is the vector of end-effector linear and angular velocities with respect to the reference frame; $\dot{\mathbf{q}}$ is the joint velocities. Since $\mathbf{J} = \mathbf{J_q}^{-1}\mathbf{J_x}$, $\mathbf{J_x}$ and $\mathbf{J_q}^{-1}$ are defined as:

$$ \mathbf{J_x} = \begin{bmatrix} {}^0\hat{\$}^T_{r_1} \\ {}^0\hat{\$}^T_{r_2} \\ \vdots \\ {}^0\hat{\$}^T_{r_m} \end{bmatrix} \tag{A.13} $$

$$ \mathbf{J_q}^{-1} = diag\left(\frac{1}{{}^0\hat{\$}^T_{r_1}\ {}^0\hat{\$}_1}, \frac{1}{{}^0\hat{\$}^T_{r_2}\ {}^0\hat{\$}_2}, \cdots, \frac{1}{{}^0\hat{\$}^T_{r_m}\ {}^0\hat{\$}_m} \right) \tag{A.14} $$

where the transpose of a screw is defined as $\hat{\$}^T = [S_4, S_5, S_6, S_1, S_2, S_3]$ for convenience.

A.5 Statics

Using the formulation of (Tsai, 1999), the static force problem may be derived[1]. Allow $\delta\mathbf{q}$ to represent the vector of virtual displacements associated with the actuated joints, and let $\delta\mathbf{x}$ represent the virtual displacements associated with the end effector. The application of virtual work yields:

$$\boldsymbol{\tau}^T\delta\mathbf{q} - \mathbf{F}^T\delta\mathbf{x} = 0 \tag{A.15}$$

where, $\boldsymbol{\tau}$ is a vector of the joint torques or forces.
The virtual displacements are related by the Jacobian matrix

$$\delta\mathbf{q} = \mathbf{J}\delta\mathbf{x} \tag{A.16}$$

substituting (A.16) into (A.15) yields:

$$(\boldsymbol{\tau}^T\mathbf{J} - \mathbf{F}^T)\delta\mathbf{x} = 0 \tag{A.17}$$

which holds for any virtual displacement $\delta\mathbf{x}$, therefore we obtain:

$$\mathbf{F} = \mathbf{J}^T\boldsymbol{\tau} \tag{A.18}$$

Equation (A.18) is an important relationship which describes a linear trans-

[1]Note that this formulation differs from that in (Merlet, 2005), although the two are equivalent.

formation between the joint torque and the end-effector wrench.

Appendix B

Interval Analysis: Additional Routines

B.1 Extended Interval Arithmetic

Extended interval arithmetic extends ordinary interval arithmetic to allow for divisions by intervals containing 0. Interval division is described as follows

$$1/[c,d] = \{1/y : y \in [c,d]\} \tag{B.1}$$

where c and d are any real numbers and $c < d$.

If $0 \in [c,d]$, ordinary interval arithmetic cannot be used. Instead, extended interval arithmetic introduces three cases (Moore et al., 2009):

1. If $c = 0 < d$, then $1/[c,d] = [1/d, +\infty)$;

2. If $c < 0 < d$, then $1/[c, d] = (-\infty, 1/c] \cup [1/d, +\infty)$.

3. If $c < d = 0$, then $1/[c, d] = (-\infty, 1/c]$.

Problems which may otherwise not be solved with ordinary interval arithmetic may be solved with extended interval arithmetic.

B.2 Regularity of Interval Matrices

A square interval matrix $[\mathbf{X}]$ is said to be regular if each $\mathbf{X} \in [\mathbf{X}]$ is non-singular. It is called singular otherwise. The regularity test is NP-hard (Fiedler et al., 2006). The following theorem by Baumann (1984) may be used to determine if an interval matrix is regular:

Theorem B.2.1 (Baumann regularity theorem)**.** A square $n \times n$ interval matrix $[\mathbf{X}]$ is regular if and only if the determinants of all sub-matrices $\mathbf{X}_{yz},\ y, z \in \mathbb{Y}_n$ are non-zero and of the same sign. The set $\mathbb{Y}_n$ contains all 2^n combinations of n-dimensional vectors with elements 1 or -1. That is, if $n = 2$, then $\mathbb{Y}_n$ is the set

$$\mathbb{Y}_n = \left\{ \begin{pmatrix} -1 \\ -1 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\} \tag{B.2}$$

For each $y, z \in \mathbb{Y}_n$ the matrix $\mathbf{X}_{yz}$ is

$$(\mathbf{X}_{yz})_{ij} = \begin{cases} \overline{[\mathbf{X}_{ij}]} & \text{if } y_i z_i = -1, \\ \underline{[\mathbf{X}_{ij}]} & \text{if } y_i z_i = 1 \end{cases} \quad i, j = 1, \ldots, n \tag{B.3}$$

with cardinality of at most 2^{2n-1} since $(\mathbf{X}_{-y,-z}) = (\mathbf{X}_{yz})$.

B.3 Simplification Methods

B.3.1 2B Filtering

The 2B filtering method (see (Lhomme, 1993)) rearranges an equation to isolate each occurrence of an interval variable on the left-hand side. The isolated interval variable, denoted $[U_i]$, has an initial domain. The right-hand side of the equation is intersected with the initial domain of the variable in order to simplify the variable.

Consider the equation $f([u]) = [u]^2 - 2[u] + 1 = 0$ with exact coefficients (example from (Merlet, 2012)). The 2B procedure introduces a new variable $[U_1]$, such that $[U_1] = 2[u]-1$ and computes the interval evaluation. Since $[U_1]$ is equal to $[u]^2$, $[u]$ is updated by the intersection of $[u]$ and $[-\sqrt{\overline{[U_1]}}, \sqrt{\overline{[U_1]}}]$. Next, the procedure introduces another new variable $[U_2]$, such that $[U_2] = ([u]^2 + 1)/2$ and computes the interval evaluation. Since $[U_2]$ is equal to $[u]$, $[u]$ is updated by the intersection of $[u]$ and $[U_2]$. If $[u]$ becomes empty, then there is no solution inside the domain of $[u]$. The 2B procedure for this

example is visualized in Figure B.1 for $[u] = [-10, 10]$.

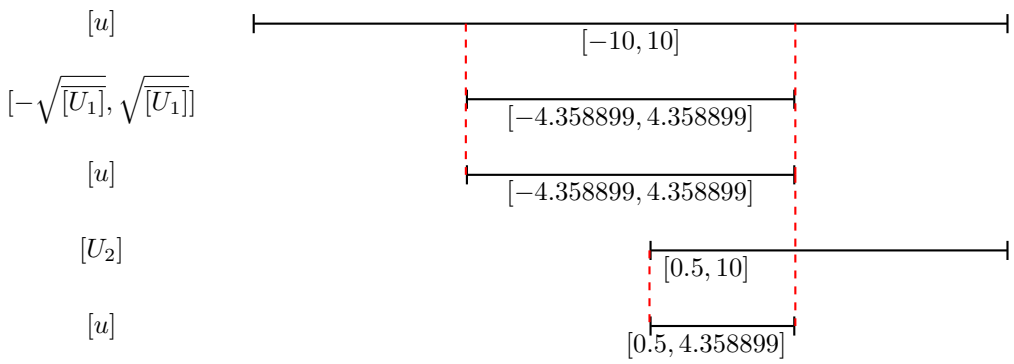


Figure B.1: Visualization of 2B filtering applied to $f([u]) = [u]^2 - 2[u] + 1 = 0$ with $[u] = [-10, 10]$.

B.3.2 3B Filtering

The 3B filtering method (see (Collavizza et al., 1999)) attempts to reduce the width of an interval variable $[u_i]$ by removing inconsistent portions of the upper or lower bounds of the interval. In practice, 3B filtering selects end-point sub-intervals of width δ_{3B} (selected by the user), also called slices. The unknowns $[\mathbf{u}]$ are set as the end-point sub-interval. A slice is inconsistent if the inclusion function of $f([\mathbf{u}], [\mathbf{k}]) = 0$ does not contain 0. Inconsistent slices are removed from $[u_i]$, reducing the width of the interval. This is repeated until no additional slices are removed. Simplification is applied to each unknown variable. The 3B procedure for the previous example is visualized in Figure B.2 for $[u] = [0.5, 4.358899]$, where $[u_l]$ and $[u_u]$ denote the lower and upper sub-intervals.

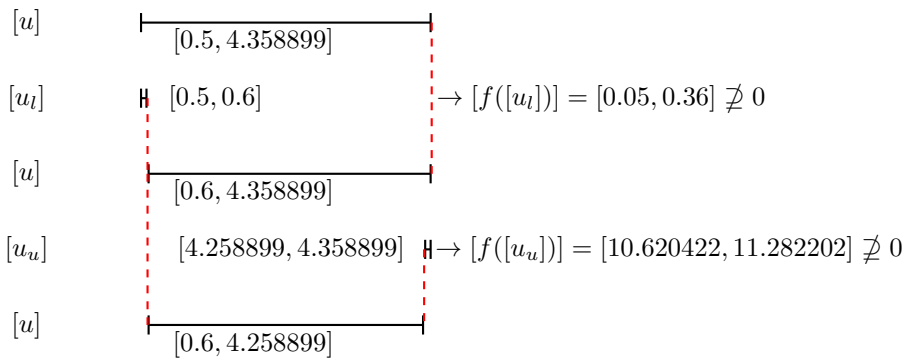


Figure B.2: Visualization of 3B filtering applied to $f([u]) = [u]^2 - 2[u] + 1 = 0$ with $[u] = [0.5, 4.358899]$ with 3B width of $\delta_{3B} = 0.1$.

B.4 Existence Methods

B.4.1 Interval Newton Method

The interval extension of Newton's method (see (Moore, 1966)) may be used for simplification and it also has the ability to find unique solutions. The benefit of using Interval Newton for simplification is that it works as a global filtering technique applied to all of the unknowns, whereas the 2B and 3B filtering methods are local filtering techniques (*i.e.*. they operate on a single variable consecutively).

The Interval Newton iterate is given by

$$[\mathbf{u}^{k+1}] = [\mathbf{u}^k] \cap N([\mathbf{u}^k]) \tag{B.4}$$

with

$$N([\mathbf{u}]) = mid([\mathbf{u}]) - \frac{\mathbf{f}(mid([\mathbf{u}]), [\mathbf{k}])}{\mathbf{f}'([\mathbf{u}], [\mathbf{k}])} \tag{B.5}$$

If the result of Interval Newton $N([\mathbf{u}])$ is included in $[\mathbf{u}]$, then there exists a zero in $[\mathbf{u}]$. That is,

$$N([\mathbf{u}]) \subseteq [\mathbf{u}] \rightarrow \exists \mathbf{y} \in [\mathbf{u}] : \mathbf{f}(\mathbf{y}, [\mathbf{k}]) = \mathbf{0} \tag{B.6}$$

If the intersection $[\mathbf{u}] \cap N([\mathbf{u}])$ is empty, there is no zero in $[\mathbf{u}]$. That is,

$$N([\mathbf{u}]) \cap [\mathbf{u}] = \emptyset \rightarrow \nexists \mathbf{y} \in [\mathbf{u}] : \mathbf{f}(\mathbf{y}, [\mathbf{k}]) = \mathbf{0} \tag{B.7}$$

Otherwise, the Interval Newton iterates may be used to simplify the unknowns.

Interval Newton repeats until a desired resolution ϵ is achieved on either the solution or the simplification. The method is described in Algorithm 13.

The procedure is applied to the previous example and is visualized in Figure B.3 for $[u] = [0.6, 4.258899]$. Extended interval analysis is used to simplify the domain of $[u]$. Indeed, the solution $[u^*] = 1$ is contained inside the simplified domain. While not demonstrated in the example, repeated calls to the interval Newton method will tightly converge to the actual solution.

B.4.2 Kantorovitch Method

The Kantorovitch method (Kantorovich and Forsythe, 1952) determines if a unique solution can be found for the system $\mathbf{f}([\mathbf{u}], [\mathbf{k}]) = \mathbf{0}$. Let a vector norm be given by $||\mathbf{a}|| = \max_i |a_i|$ and a matrix norm be given by $||\mathbf{A}|| =$

Algorithm 13: Interval Newton method

function INTERVAL_NEWTON $([\mathbf{u}], [\mathbf{k}], \epsilon)$;
Input : unknowns $[\mathbf{u}]$, knowns $[\mathbf{k}]$, and desired resolution ϵ
Output: classification: converged(1), no solution(-1), unknown(0)
if $N([\mathbf{u}]) \subseteq [\mathbf{u}]$ **then**
  if $width([\mathbf{u}]) - width(N([\mathbf{u}])) < \epsilon$ **then**
    $[\mathbf{u}] = N([\mathbf{u}])$ and **return** 1 ; `// Convergence achieved`
  else
    $[\mathbf{u}] = N([\mathbf{u}])$ and repeat
else if $N([\mathbf{u}]) \cap [\mathbf{u}] == \emptyset$ **then**
  return -1 ; `// There is no solution`
else
  if $width([\mathbf{u}]) - width([\mathbf{u}] \cap N([\mathbf{u}])) < \epsilon$ **then**
    $[\mathbf{u}] = [\mathbf{u}] \cap N([\mathbf{u}])$ and **return** 0 ; `// [u] cannot be simplified further`
  else
    $[\mathbf{u}] = [\mathbf{u}] \cap N([\mathbf{u}])$ and repeat

$\max_i \sum_j |a_{ij}|$.

Theorem B.4.1 (Kantorovitch (Tapia, 1971))**.** For a square $(n \times n)$ nonlinear system $\mathbf{f}(\mathbf{u}, \mathbf{k}) = \mathbf{0}$, consider a convex set $\mathbb{U}$ and assume for some point $\mathbf{u}_0 \in \mathbb{U}$ that $\mathbf{f}(\mathbf{u}_0, \mathbf{k})^{-1}$ exists and that

1. $||\mathbf{f}'(\mathbf{u}_0, \mathbf{k})^{-1}|| \leq A_0$;
2. $||\mathbf{f}'(\mathbf{u}_0, \mathbf{k})^{-1}\mathbf{f}(\mathbf{u}_0, \mathbf{k})|| \leq B_0$;
3. $||\mathbf{f}'(\mathbf{x}, \mathbf{k}) - \mathbf{f}'(\mathbf{y}, \mathbf{k})|| \leq K||(\mathbf{x} - \mathbf{y})||$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{U}$;
4. the constants A_0, B_0, K satisfy $A_0 B_0 K \leq 1/2$.

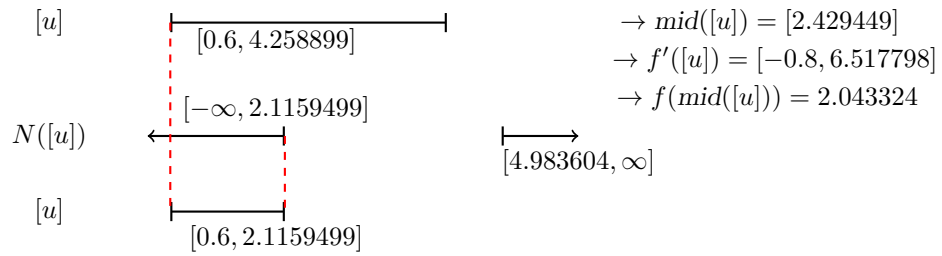


Figure B.3: Visualization of one iteration of Interval Newton applied to $f([u]) = [u]^2 - 2[u] + 1 = 0$ with $[u] = [0.6, 4.258899]$. Extended interval analysis is applied.

Let $\Omega_* = \{\mathbf{u} \mid ||\mathbf{u} - \mathbf{u}_0|| \leq 2B_0\}$ describe a ball. If $\Omega_* \subset \mathbb{U}$, then the Newton iterates $\mathbf{u}_{k+1} = \mathbf{u}_k - \mathbf{f}'(\mathbf{u}_k, \mathbf{k})^{-1}\mathbf{f}(\mathbf{u}_k, \mathbf{k})$ are well defined, remain in Ω_* and converge to the unique solution $\mathbf{u}^* \in \Omega_*$ such that $\mathbf{f}(\mathbf{u}^*, \mathbf{k}) = \mathbf{0}$.

This theorem is used in (Rall, 1969) to explicitly compute the Lipschitz constant K using an interval arithmetic evaluation to bound the Hessian of the system over the interval containing Ω_*, $\mathbf{f}''([\Omega_*], \mathbf{k}))$, such that

$$||\mathbf{f}''([\Omega_*], \mathbf{k})|| \leq K \tag{B.8}$$

where $f''_{ijk} = \frac{\delta^2 f_i}{\delta u_j \delta u_k}$ and

$$K = \max_i \sum_{j=1}^{n} \sum_{k=1}^{n} |f''_{ijk}([\Omega_*], \mathbf{k})| \tag{B.9}$$

Merlet (2005) provides a slightly different formulation which is implemented in the ALIAS library (Merlet, 2007a).

Given the thick system with interval coefficients $[\mathbf{k}]$ and unknown domain $[\mathbf{u}]$,

Algorithm 14: Kantorovitch existence method

```
function KANTOROVITCH ([u], [k], ε);
Input  : unknowns [u], knowns [k], and desired resolution ε
Output: classification: unique solution(1), unknown(0)
if 2A_o B_o K ≤ 1 then a unique solution is detected
    [u_ball] = [mid([u]) − 2B_0, mid([u]) + 2B_0];
    NEWTON([u_ball], [k], ε) ;          // Apply Newton's method
    [u] = [u_ball];
    return 1 ;    // There is a unique solution
else
    return 0 ;    // It is unknown if there is a solution
```

$\mathbf{f}([\mathbf{u}], [\mathbf{k}]) = \mathbf{0}$, the Kantorovitch method is used to determine if an *existence ball* $[\mathbf{u}_{\text{ball}}] = [\mathbf{u}_0 - 2B_0, \mathbf{u}_0 + 2B_0]$, centred at $\mathbf{u}_0 = mid([\mathbf{u}])$, can be found. The solution is then obtained by iterating Newton's method started at $\mathbf{u}_0$ to a resolution ϵ. The point $\mathbf{u}_0$ is considered as an approximate solution at which the Jacobian matrix $\mathbf{f}'(\mathbf{u}_0, [\mathbf{k}])$ is locally Lipschitz and invertible. The pseudo-code corresponding to Theorem B.4.1 is given in Algorithm 14. The routine *NEWTON* implements the classical Newton's method. The set difference between $[\mathbf{u}_{\text{ball}}]$ and $[\mathbf{u}]$ may be incorporated to detect the existence of more than one solution in the domain $[\mathbf{u}]$ (see (Merlet, 2007a)).

B.4.3 Krawczyk Method

The Krawczyk method (also known as the Moore-Krawczyk method) is based on the Brouwer fixed-point theorem. Here $K([\mathbf{u}])$ represents the Krawczyk evaluation of the unknowns $[\mathbf{u}]$. For convenience, let $\mathbf{f}_{mid} = \mathbf{f}(mid([\mathbf{u}]), [\mathbf{k}])$,

$\mathbf{J}_{mid} = \mathbf{f}'(mid([\mathbf{u}]), [\mathbf{k}])$, and $\mathbf{J} = \mathbf{f}'([\mathbf{u}], [\mathbf{k}])$. The Krawczyk evaluation is then

$$K([\mathbf{u}]) = mid([\mathbf{u}]) - \mathbf{J}_{mid}^{-1}\mathbf{f}_{mid} + (\mathbf{I} - \mathbf{J}_{mid}^{-1}\mathbf{J})([\mathbf{u}] - mid([\mathbf{u}]))) \tag{B.10}$$

If $K([\mathbf{u}]) \subseteq [\mathbf{u}]$, then there is a fixed point (a unique solution) in $[\mathbf{u}]$ and $K([\mathbf{u}])$ converges to this solution. If $K([\mathbf{u}]) \cap [\mathbf{u}]$ is empty, then there is no solution in $[\mathbf{u}]$. A benefit of Krawczyk is that it also works as a global filtering technique and simplifies $[\mathbf{u}]$. If $K([\mathbf{u}]) \cap [\mathbf{u}]$ is non-empty, then the result of the intersection may be taken as a simplification of $[\mathbf{u}]$. The Krawczyk method repeats until a desired resolution ϵ is achieved on the solution or simplification. The method is described in Algorithm 15.

Algorithm 15: Krawczyk existence method

function KRAWCYZK ($[\mathbf{u}], [\mathbf{k}], \epsilon$);
Input : unknowns $[\mathbf{u}]$, knowns $[\mathbf{k}]$, and desired resolution ϵ
Output: classification: unique solution(1), no solution(-1), unknown(0)

```
if K([u]) ⊆ [u] then
    if width([u]) − width(K([u])) < ε then
        [u] = K([u]) and return 1 ;      // A unique solution is
                                            found within tolerance
    else
        [u] = K([u]) and repeat;
else if K([u]) ∩ [u] == ∅ then
    return −1 ;                          // There is no solution
else
    if width([u]) − width([u] ∩ K([u])) < ε then
        [u] = [u] ∩ K([u]) and return 0 ;  // [u] cannot be
                                              improved further
    else
        [u] = [u] ∩ K([u]) and repeat
```

Appendix C

Inverse Kinematics and Jacobians of the 3-RRR

C.1 Inverse Kinematics Formulation

Using vector notation $\overrightarrow{A_iC_i}$ may be written for each limb i, in terms of the reference frame $\{O_i\}$, as follows

$$\overrightarrow{A_iC_i} = \overrightarrow{A_iB_i} + \overrightarrow{B_iC_i} \tag{C.1}$$

Which may be written for each component as

$$C_{xi} = r_i \cos(\alpha_i) + l_i \cos(\alpha_i + \beta_i) \tag{C.2}$$

$$C_{yi} = r_i \sin(\alpha_i) + l_i \sin(\alpha_i + \beta_i) \tag{C.3}$$

Squaring and adding (C.2) and (C.3) gives

$$C_{xi}^2 + C_{yi}^2 = r_i^2 + l_i^2 + 2r_i l_i \cos(\beta_i) \tag{C.4}$$

Isolating for the cosine gives

$$\cos(\beta_i) = \frac{C_{xi}^2 + C_{yi}^2 - r_i^2 - l_i^2}{2r_i l_i} \tag{C.5}$$

The sine term may be written as

$$\sin(\beta_i) = \pm\sqrt{1 - \cos(\beta_i)^2} \tag{C.6}$$

The trigonometric identity $\cos(\alpha_i + \beta_i) = \cos(\alpha_i)\cos(\beta_i) - \sin(\alpha_i)\sin(\beta_i)$ may be substituted into (C.2) then simplified to:

$$C_{xi} = (r_i + r_i \cos(\beta_i))\cos(\alpha_i) - l_i \sin(\beta_i)\sin(\alpha_i) \tag{C.7}$$

Likewise, the trigonometric identity $\sin(\alpha_i + \beta_i) = \cos(\alpha_i)\sin(\beta_i) + \sin(\alpha_i)\cos(\beta_i)$ may be substituted into (C.3) then simplified to:

$$C_{yi} = l_i \sin(\beta_i)\cos(\alpha_i) + (r_i + l_i \cos(\beta_i))\sin(\alpha_i) \tag{C.8}$$

Solving (C.7) and (C.8) for $\cos(\alpha_i)$ and $\sin(\alpha_i)$ yields:

$$\cos(\alpha_i) = (C_{xi}(r_i + l_i \cos(\beta_i)) + C_{y_i} l_i \sin(\beta_i))/(C_{x_i}^{\ 2} + C_{y_i}^{\ 2}) \tag{C.9}$$

$$\sin(\alpha_i) = (C_{y_i}(r_i + l_i\cos(\beta_i)) - C_{xi}l_i\sin(\beta_i))/(C_{xi}^{\ 2} + C_{y_i}^{\ 2}) \tag{C.10}$$

C.2 Jacobian Formulation

The direct Jacobian $\mathbf{J_x}$, inverse Jacobian $\mathbf{J_q}$, and combined Jacobian $\mathbf{J}$ may be obtained with screw theory (see Section A.4). Allow a reference frame $\{O'\}$ to be coincident with the end-effector frame $\{E\}$ and have the same orientation as the base frame $\{O\}$.

Let $\mathbf{p}_i$ be the vector $\overleftarrow{PA_i}$. The unit joint screws of the actuated joints for each limb i are given by

$$\hat{\$}_i = \left(\widehat{z}, \mathbf{p}_i \times \widehat{z}\right)^T \tag{C.11}$$

where $\widehat{z}$ is the unit vector along z.

Let $\widehat{\mathbf{g}}_i$ be a unit vector along $\overleftarrow{B_iC_i}$. The unit reciprocal screws for each limb i are given by

$$\hat{\$}_{ri} = \left(\widehat{\mathbf{g}}_i, \mathbf{d}_i \times \widehat{\mathbf{g}}_i\right)^T \tag{C.12}$$

The unit joint screws and unit reciprocal screws are evaluated as

$$\hat{\$}_i = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -((C_{yi} - A_{yi}) - (C_{yi} - P_y)) \\ ((C_{xi} - A_{xi}) - (C_{xi} - P_x) \\ 0 \end{pmatrix} \tag{C.13}$$

$$\hat{\$}_{ri} = \begin{pmatrix} (C_{xi} - A_{xi}) - r_i \cos(\alpha_i))/l_i \\ (C_{yi} - A_{yi}) - r_i \sin(\alpha_i))/l_i \\ 0 \\ 0 \\ 0 \\ (-(C_{yi} - P_y)((C_{xi} - A_{xi}) - r_i \cos(\alpha_i)) + (C_{yi} - P_x)((C_{yi} - A_{yi}) - \\ r_i \sin(\alpha_i)))/l_i \end{pmatrix} \tag{C.14}$$

The direct Jacobian is the matrix of the reciprocal screws $\hat{\$}_{ri}$ for $i = 1, \ldots, 3$. The inverse Jacobian is the diagonal matrix whose elements are the reciprocal product of the unit reciprocal screws and unit joint screws. Each diagonal elements of $\mathbf{J_q}$ simplifies to

$$\mathbf{J_q}(i,i) = r_i/l_i \left((C_{yi} - A_{yi}) \cos(\alpha_i) - (C_{xi} - A_{xi}) \sin(\alpha_i) \right) \tag{C.15}$$

Appendix D

Wrench Capabilities of Exact Designs

Each actuator is capable of supplying a force/torque τ_i such that $\tau_i \in [\tau_i]$ where $[\tau_i] = \{\tau_i \mid \underline{\tau_i} \leq \tau_i \leq \overline{\tau_i}\}$. If $\boldsymbol{\tau}$ is a vector containing the articular forces/torques of all active joints, where $\boldsymbol{\tau} \in \mathbb{R}^m$, then a linear mapping from the joint space to the task space is accomplished using the forward force solution:

$$\mathbf{F} = \mathbf{J}^T \boldsymbol{\tau} \tag{D.1}$$

where, $\mathbf{J}$ is the $m \times n$ screw-based Jacobian matrix, as defined in (Tsai, 1999). Carretero and Gosselin (2010) denote the set of available articular forces or torques $[\boldsymbol{\tau}]$ as the *Actuator Force Workspace*. In non-redundant mechanisms, $n = m$, thus $\mathbf{J}$ is square and (D.1) represents a unique mapping from $\boldsymbol{\tau}$ to $\mathbf{F}$. This is not the case for actuation redundant mechanisms, where $n < m$, as

the solution of Eq. (D.1) is no longer unique. Instead, there can be multiple combinations of $\boldsymbol{\tau}$ which can produce exactly the same $\mathbf{F}$.

The wrench capabilities $\mathcal{F}$ are the set of wrenches $\mathbf{F}$ obtained as the image of (D.1) over the Actuator Force Workspace $[\boldsymbol{\tau}]$. That is:

$$\mathcal{F} = \left\{ \mathbf{F} \mid \mathbf{F} = \mathbf{J}^T \boldsymbol{\tau}, \boldsymbol{\tau} \in [\boldsymbol{\tau}] \right\} \tag{D.2}$$

Each vertex of $[\boldsymbol{\tau}]$ represents all m actuators working at their respective maximum or minimum limits. The interval vector describes an axis-aligned box, which is classified as a convex polytope. As discussed in (Carretero et al., 2008), $[\boldsymbol{\tau}]$ may be further classified as a m-dimensional zonotope. Due to the property of linear transformations in which a convex space maps to a convex space and symmetry is preserved, the mapping (D.2) yields a n-dimensional zonotope for $\mathcal{F}$.

A convex object can be represented in either a vertex or hyperplane representation ($\mathcal{V}$ and $\mathcal{H}$ respectively). The vertices of $[\boldsymbol{\tau}]$, denoted by the finite set $\mathcal{T}_\mathcal{V}$, can be mapped in order to obtain the corresponding finite set of vertices of the wrench capabilities, denoted $\mathcal{F}_\mathcal{V}$. The complete description of $\mathcal{F}$ is obtained by taking the convex hull of $\mathcal{F}_\mathcal{V}$. That is, the wrench capabilities $\mathcal{F}$ can be obtained by

$$\begin{aligned} \mathcal{F}_\mathcal{V} &= \left\{ \mathbf{F} \mid \mathbf{F} = \mathbf{J}^T \boldsymbol{\tau}, \boldsymbol{\tau} \in \mathcal{T}_\mathcal{V} \right\} \\ \mathcal{F} &= conv\left(\mathcal{F}_\mathcal{V}\right) \end{aligned} \tag{D.3}$$

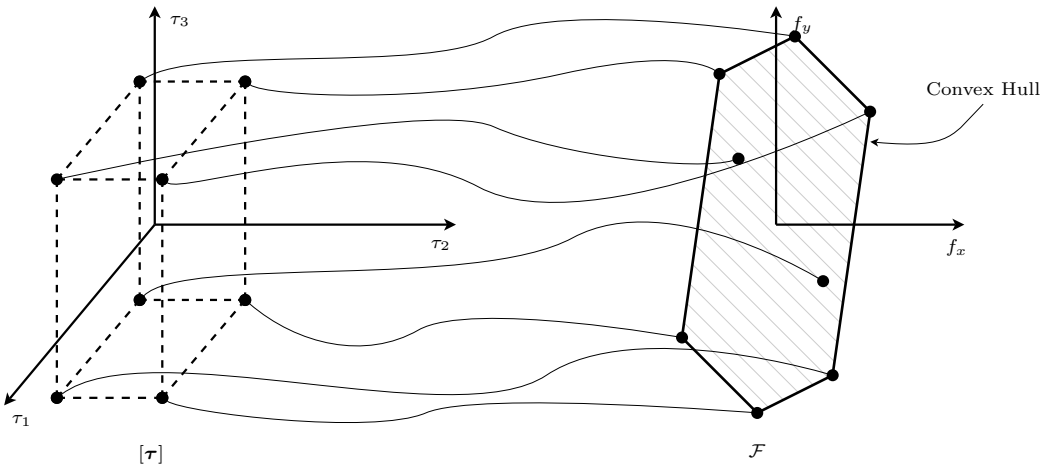


Figure D.1: Actuator force workspace $[\boldsymbol{\tau}]$ (left) formed from the torque limits of three actuators (τ_1, τ_2, τ_3) mapped to form the wrench capabilities $\mathcal{F}$ (right) using equation (D.3).

This mapping is uni-directional for actuation redundant mechanisms, as not all vertices of $[\boldsymbol{\tau}]$ map to vertices of $\mathcal{F}$; thus, the vertices of the $[\boldsymbol{\tau}]$ cannot be obtained from $\mathcal{F}$. This is due to the projection from $\mathbb{R}^m$ to $\mathbb{R}^n$, where $n < m$, caused by the redundant actuators. The Jacobian matrix in (D.1) applied to the m-dimensional joint space results in a n-dimensional projection in the task space. Figure D.1 shows an example of the Jacobian mapping for a mechanism with $n = 2$ and $m = 3$.

A numerical procedure such as the *quickhull* algorithm (Barber et al., 1996) is a widely used tool to find the convex hull of a finite set of points in an arbitrary number of dimensions. Even if *quickhull* is usually fast, a method that is non-iterative is desirable. The hyperplane shifting method (Bouchard et al., 2008, Gouttefarde and Krut, 2010), uses the Jacobian matrix and the articular forces/torques to obtain the wrench capabilities.

D.1 Describing the Desired Wrench Capabilities

The desired wrench capabilities, denoted $\mathcal{F}_{desired}$, have taken many forms in the literature. They have been described geometrically as:

- A point (a single wrench) (Bouchard et al., 2008, Firmani et al., 2008b, 2007, Papadopoulos and Gonthier, 1999).

  Point representation is important when a task requires the generation of a single wrench throughout a trajectory, *e.g.*, the static equilibrium workspace of a CDPM[1].

- A hyper-ellipse (Bouchard et al., 2008, Carretero and Gosselin, 2010, Garg et al., 2009a).

  The hyper-ellipse representation allows the set of wrenches to be modelled as ball-shaped, where the different wrench axes can have different physical units so the ball deforms to a hyper-ellipse. An axis-aligned hyper-ellipse description for the wrench capabilities is given for the spatial case by:

$$\mathcal{F} = \left\{ \mathbf{F} \;\middle|\; \frac{f_x^2}{r_{f_x}^2} + \frac{f_y^2}{r_{f_y}^2} + \frac{f_z^2}{r_{f_z}^2} + \frac{m_x^2}{r_{m_x}^2} + \frac{m_y^2}{r_{m_y}^2} + \frac{m_z^2}{r_{m_z}^2} = 1 \right\} \tag{D.4}$$

[1]The static equilibrium workspace of a CDPM is the set of end-effector poses which are kinematically permissible and maintain static equilibrium considering gravity with tension forces in the cables.

where, r_* ($*$ represents all applicable subscripts) represent the length of the axis-aligned semi-principal axes of the hyper-ellipse for forces and moments denoted by subscripts f and m respectively. The independent lengths for the force and moment semi-principal axes account for the mixed units. This method, and projections of this method for planar cases, with $r_{f_x} = r_{f_y} = r_{f_z}$ and $r_{mx} = r_{my} = r_{mz}$, has been used in many papers (Carretero and Gosselin, 2010, Zibil et al., 2007, Bouchard et al., 2008), although some do not report on the geometrical significance. The popularity of this technique is because it provides knowledge of the maximum magnitude of forces and moments that the manipulator can sustain irrespective of the direction, making it useful for design purposes.

- A box (Gouttefarde et al., 2011).

  The box representation independently defines the lower and upper bounds of the wrench elements and is suitable for interval analysis techniques. An axis-aligned box description for the wrench capabilities is given for the spatial case by:

$$\mathcal{F} = \Big([f_x], [f_y], [f_z], [m_x], [m_y], [m_z] \Big)^T \tag{D.5}$$

- A convex polytope (Bouchard et al., 2008, Firmani et al., 2008b).

  The convex polytope representation (a general case of the box repre-

sentation) allows for more complicated descriptions of desired wrenches described by the intersection of a set of half-spaces.

Considering an exact design, the wrench capabilities are a function of the pose $\mathbf{p}$ and design $\mathcal{D}$: $\mathcal{F}(\mathbf{p}, \mathcal{D})$. In order for the pose of a mechanism to satisfy the desired wrench capabilities, the desired wrench capabilities must be a subset of the wrench capabilities at that pose. That is:

$$\mathcal{F}_{desired} \subseteq \mathcal{F}(\mathbf{p}, \mathcal{D}) \tag{D.6}$$

Selecting a convex polytope representation for $\mathcal{F}_{desired}$, validation of (D.6) can be simplified using the $\mathcal{H}$-representation for $\mathcal{F}(\mathbf{p}, \mathcal{D})$ and the $\mathcal{V}$-representation for the $\mathcal{F}_{desired}$. This validation can be performed by verifying the following equation (Bouchard et al., 2008):

$$\mathbf{NV} \leq \mathbf{D} \tag{D.7}$$

where, $\mathbf{N}$ is a matrix whose rows are the unit normal vectors of the supporting hyperplanes of $\mathcal{F}(\mathbf{p}, \mathcal{D})$, $\mathbf{V}$ is a matrix where each column is a vertex of the $\mathcal{F}_{desired}$, and $\mathbf{D}$ is a matrix whose rows are the projected offsets of the corresponding hyperplanes of $\mathcal{F}(\mathbf{p}, \mathcal{D})$ along their respective normal vector. The resulting set of poses which satisfy (D.7) are said to be wrench-feasible and collectively form the wrench workspace $\mathbb{P}_{WW}$.

D.2 Discretization

Two discretization methods for exact design are introduced:

1. *Grid search*: The grid search discretization provides a straightforward sampling of the continuous pose search space $\mathbb{P}$. Given a desired resolution ϵ, the search space $\mathbb{P}$ is divided along each dimension into a set of slices. The intersections of the slices are the grid points which will be sampled. The set of pose grid points associated with the desired region can be obtained.

2. *Radial search*: The radial search discretization searches radially about a point. Given a desired resolution ϵ, the search space $\mathbb{P}$ is divided into a set of rays. A line search is performed along each ray to detect the edge of the desired region.

Appendix E

Jacobians of the 3-RPR Parallel Mechanism

The units joint screw for each active joint is an infinite pitch screw and can be written for each limb i as:

$$\hat{\$}_i = \begin{pmatrix} \mathbf{0} \\ \widehat{\mathbf{d}}_i \end{pmatrix} \tag{E.1}$$

Since the axes of the passive joints in each limb intersect the line passing through points A_i and B_i, a unique screw that is reciprocal to the passive joint screws is defined as

$$\hat{\$}_{ri} = \begin{pmatrix} \widehat{\mathbf{d}}_i \\ \mathbf{b}_i \times \widehat{\mathbf{d}}_i \end{pmatrix} \tag{E.2}$$

Since, $\widehat{\mathbf{d}}_i$ is a unit vector, we obtain $\widehat{\mathbf{d}}_i^{\,T} \cdot \widehat{\mathbf{d}}_i = 1$, and the reciprocal product

of $\hat{\$}_{ri}$ and $\hat{\$}_i$ is 1. That is, the inverse Jacobian $\mathbf{J_q}$ is an identity matrix. The direct Jacobian $\mathbf{J_x}$ is:

$$\mathbf{J_x} = \begin{bmatrix} \widehat{\mathbf{d}_1}^T & (\mathbf{b}_1 \times \widehat{\mathbf{d}_1})^T \\ \widehat{\mathbf{d}_2}^T & (\mathbf{b}_2 \times \widehat{\mathbf{d}_2})^T \\ \widehat{\mathbf{d}_3}^T & (\mathbf{b}_3 \times \widehat{\mathbf{d}_3})^T \end{bmatrix} \tag{E.3}$$

Substituting $\mathbf{J} = \mathbf{J_q}^{-1}\mathbf{J_x}$ into the forward force solution, $\mathbf{F} = \mathbf{J}^T\boldsymbol{\tau}$, yields:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{d}_1} & \widehat{\mathbf{d}_2} & \widehat{\mathbf{d}_3} \\ (\mathbf{b}_1 \times \widehat{\mathbf{d}_1}) & (\mathbf{b}_2 \times \widehat{\mathbf{d}_2}) & (\mathbf{b}_3 \times \widehat{\mathbf{d}_3}) \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

Each column of $\mathbf{J}^T$ is equivalent to the reciprocal screws defined for each limb, thus some authors denote the Jacobian as the *Wrench matrix*.

Appendix F

Task-Optimized Parallel Mechanisms

Given a desired task for a mechanism to complete, the performance of the mechanism may be improved by optimising the design parameters. The result of optimising the design parameters of a mechanism for a desired task is a *task-optimized mechanism*. The methods proposed here are developed for mechanisms with exact designs.

Assuming that a desired task is described in terms of a task workspace $\mathbb{P}_{TW}$, and desired wrench capabilities $\mathcal{F}_{desired}$, a mechanism will have an associated reachable workspace $\mathbb{P}_{RW}$ and wrench workspace $\mathbb{P}_{WW}$. A task-optimized mechanism must satisfy each pose in $\mathbb{P}_{TW}$, while also minimising the difference between $\mathbb{P}_{RW}$ and $\mathbb{P}_{WW}$. In addition, it is usually desirable to minimize the cost of producing the mechanism, which primarily relates to the size and

selection of actuators.

Approximations of the workspaces are considered and the continuous design space is only partially searched. Determination of a task-optimized mechanism relies on using global optimization methods to solve constrained optimization problems. A task-optimized mechanism (*i.e.*, a solution) is a considered as a set of design parameters which locally minimize the design space and satisfy all constraints. Locating the global optimum with global optimization methods is possible, but not guaranteed.

Task-optimized mechanisms obtained with the techniques presented here are not considered to be reliable, although interval-based routines may be applied to a task-optimized mechanism in order to verify its performance. Therefore, the combination of methods applicable to exact design with methods applicable to appropriate designs provides a useful tool for synthesising and verifying the performance of mechanisms.

F.1 Global Optimization

Given the non-linearity of the kinematics and statics of parallel mechanisms, design optimization is a highly non-linear problem which contains many local optimal solutions. Conventional local optimization techniques converge to local solutions which are not necessarily optimal and the majority of the search spaces remains unexplored. Global optimization is a separate branch in the field of optimization which is concerned with attempting to find global

solutions to a problem, *i.e.*, the global optimum. These methods cannot guarantee global optimality, so a solution can only be regarded as a local optimum.

Global optimization algorithms can be classified into different, yet non-exclusive groups, depending on the criteria being considered, such as population based, iterative based, stochastic, or deterministic (to name a few). Population based algorithms have two important groups: Evolutionary Algorithms (EAs) and swarm intelligence based algorithm. Some of the more recognized EAs are *Genetic Algorithms* (Davis, 1991) and *Differential Evolution* (Storn and Price, 1997). These algorithms operate on an initial population (set of solutions) and employ heuristic strategies to explore the search space in an intelligent way, such that the initial set of solutions "evolve" into more preferable solutions. Differential Evolution is applied here. The canonical algorithm for Differential Evolution is given in Appendix G. The Teaching-Learning-Based Optimization (TLBO) algorithm was also considered for use due to its reportedly superior performance. However, during testing, it was discovered that TLBO contains an inherent origin bias which significantly impacts the algorithms performance. The TLBO origin bias work is found in (Pickard et al., 2016a).

Constraint handling techniques, such as penalty functions and repair algorithms, are used to deal with constraints in EAs. A survey of the state of the art constraint handling techniques is found in (Coello, 2002). The technique applied here is Deb's constraint handling technique (Deb, 2000). Deb's con-

straint handling technique guides the global optimization algorithm towards feasible regions of the search space by preferring:

- feasible individuals over infeasible individuals;
- infeasible individuals with less constraint violation;
- feasible individuals with a better objective function.

The incorporation of Deb's constraint handling technique into the Differential Evolution algorithm is provided in (Zielinski and Laur, 2008).

F.2 Objectives of Task-Optimized Mechanisms

Consider the three workspaces: the reachable workspace $\mathbb{P}_{RW}$, the wrench workspace $\mathbb{P}_{WW}$, and the task workspace $\mathbb{P}_{TW}$. The task workspace is used to describe the desired wrench workspace of the mechanism. To achieve a task-optimized mechanism, a first requirement is that each pose in the task workspace is reachable, *i.e.*, each pose is inside the reachable workspace. A second requirement is that each pose in the task workspace can generates the desired wrench capabilities, *i.e.*, each pose is also inside the wrench workspace. Since the wrench workspace is a subset of the reachable workspace, the requirements simplify to verifying that the task workspace is a subset of the wrench workspace.

Two objectives are considered for determining task-optimized mechanisms. The goal of the first objective is to minimize the wasted volume within the reachable workspace that cannot be used by the task. This equates to determining an area representation of the set difference of $\mathbb{P}_{RW}$ and $\mathbb{P}_{WW}$ (*i.e.*, $\mathbb{P}_{RW} \backslash \mathbb{P}_{WW}$), where the set difference of two sets A and B is $A \backslash B = \{x \mid x \in A \text{ and } x \notin B\}$. The goal of the second objective is to minimize the wasted volume within the wrench workspace which is outside of the task workspace. This is computed by determining an area representation of the set difference of $\mathbb{P}_{WW}$ and $\mathbb{P}_{TW}$ (*i.e.*, $\mathbb{P}_{WW} \backslash \mathbb{P}_{TW}$). Both of the objectives are considered equally important.

Considering that both objectives functions compute a result in terms of an area which is strictly positive, the multi-objective function problem can be reduced to a single objective function represented by the sum of the two objectives. The coefficients K_1 and K_2 of the two objectives are set to 1. A single constraint is associated with this problem. The goal of the optimization is to determine the optimal design of a mechanisms which can accomplish a desired task; therefore, a constraint which must be enforced is that the task workspace is a subset of the wrench workspace.

Let the allowable domains of the design parameters be described by $[\mathcal{D}]$. The objective function, for optimising the exact design $\mathcal{D}$, is formulated as

follows:

$$\begin{aligned} &\text{minimize :} \quad K_1 volume(\mathbb{P}_{RW} \backslash \mathbb{P}_{WW}) + K_2 volume(\mathbb{P}_{WW} \backslash \mathbb{P}_{TW}) \\ &\text{subject to :} \\ &\qquad \mathbb{P}_{TW} \subseteq \mathbb{P}_{WW} \\ &\qquad \mathcal{D} \in [\mathcal{D}] \end{aligned} \tag{F.1}$$

F.2.1 Determining if the Task Workspace is a Subset of the Wrench Workspace

A primary and secondary test are proposed to determine if the task workspace is a subset of the wrench workspace. The primary test provides the necessary condition and the secondary test provides the sufficient condition for the task workspace to be a subset of the wrench workspace.

The primary test concerns testing the set of vertices of the task workspace and verifying that they are inside the wrench workspace. O'Rourke (1998) describes a *ray crossing test* which can be used to verify if a point is inside a polygon or polyhedron. If one draws a ray pointing in any direction from a point inside a polygon or polyhedron and counts the number of times it crosses a facet, the point will always be inside if the number of crosses is odd. If the primary test returns that the set of vertices are inside the wrench workspace, then a secondary test must be applied to verify that a concave region of the wrench workspace does not intersect the task workspace. The secondary test is accomplished by verifying that each facet of the wrench

workspace does not intersect a facet of the task workspace. If two facets intersect, then the task workspace is not a subset of the wrench workspace.

F.2.2 Approximating the Task Workspace

It will be assumed for now that the task workspace is an axis-aligned box. Determining the area of the set difference of the wrench workspace and the task workspace is a difficult problem if the task workspace is not a subset of the wrench workspace. This test is complicated by the fact that the wrench workspace is not necessarily convex. For example, consider the task workspace (yellow) and wrench workspace (blue) in Figure F.1. The set difference of the wrench workspace and the task workspace is not easy to compute.

Rather than spend a lot of resources trying to compute the set difference of the wrench workspace and the task workspace, the task workspace may be replaced with an axis-aligned box approximation $\mathbb{P}_{\widetilde{TW}}$. The approximated task workspace is guaranteed to be inside the wrench workspace. If $\mathbb{P}_{TW} \subset \mathbb{P}_{WW}$, then $\mathbb{P}_{\widetilde{TW}} := \mathbb{P}_{TW}$. Otherwise, an approximation is obtained by fixing the geometric centre of the task workspace and scaling the task workspace by a scaling factor $\lambda \in [0, 1]$. A line search optimization is implemented to obtain the value of λ which yields the largest approximated task workspace (within a specified tolerance) that is completely inside the wrench workspace. The optimization problem may be written in terms of the approximated task

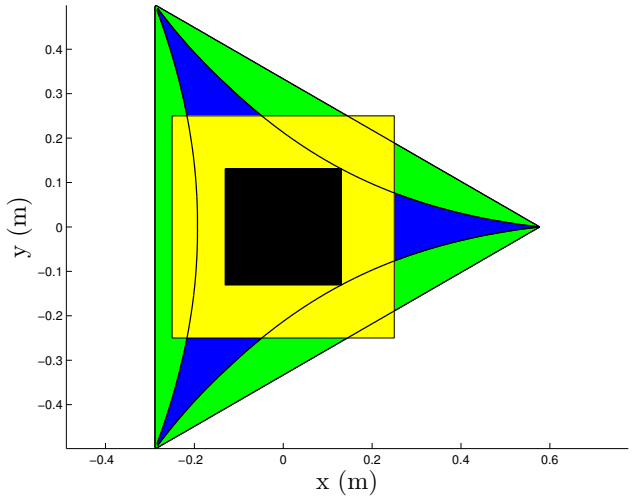


Figure F.1: An axis-aligned approximation of the task workspace (approximated task workspace) which is guaranteed to be a subset of the wrench workspace.

workspace $\mathbb{P}_{\widetilde{TW}}$ as

$$\begin{aligned} &\text{minimize :} \quad K_1 volume(\mathbb{P}_{RW} \backslash \mathbb{P}_{WW}) + K_2 volume(\mathbb{P}_{WW} \backslash \mathbb{P}_{\widetilde{TW}}) \\ &\text{subject to :} \\ &\qquad \mathbb{P}_{TW} \subseteq \mathbb{P}_{WW} \\ &\qquad \mathcal{D} \in [\mathcal{D}] \end{aligned} \tag{F.2}$$

F.2.3 Evaluating the Differences of Workspaces

An example of the four workspaces associated with a mechanism are depicted in Figure F.1 for the planar cable-driven parallel mechanism (CDPM). The green workspace depicts $\mathbb{P}_{RW}$, the blue workspace depicts $\mathbb{P}_{WW}$, the yellow workspace depicts $\mathbb{P}_{TW}$, and the black workspace depicts $\mathbb{P}_{\widetilde{TW}}$. The gener-

alized volume of the workspaces in Figure F.1 are computed as:

$$\begin{aligned} volume(\mathbb{P}_{RW}) &= 0.431652\text{m}^2; \\ volume(\mathbb{P}_{WW}) &= 0.222554\text{m}^2; \\ volume(\mathbb{P}_{TW}) &= 0.25\text{m}^2; \\ volume(\mathbb{P}_{\widetilde{TW}}) &= 0.068266\text{m}^2. \end{aligned} \tag{F.3}$$

The generalized volume of the set differences are computed as:

$$\begin{aligned} volume(\mathbb{P}_{RW} \backslash \mathbb{P}_{WW}) &= volume(\mathbb{P}_{RW}) - volume(\mathbb{P}_{WW}) = 0.209098; \\ volume(\mathbb{P}_{WW} \backslash \mathbb{P}_{\widetilde{TW}}) &= volume(\mathbb{P}_{WW}) - volume(\mathbb{P}_{\widetilde{TW}}) = 0.154288. \end{aligned} \tag{F.4}$$

The mechanism cannot accomplish the desired task because $\mathbb{P}_{\widetilde{TW}} \neq \mathbb{P}_{TW}$; therefore, optimization of the design parameters using (F.2) is required to obtain a task-optimized design for the mechanism.

F.3 Optimising the Planar Cable-Driven Parallel Mechanism

The design variables for the planar CDPMs are described in Figure F.2. Up to twelve design variables are considered for optimization. Three design variables represent the radial displacement of the actuators (d_i); three represent the angular position of the actuators (θ_i); three represent the minimum cable tension of each cable ($\underline{\tau}_i$); and three represent the maximum cable tension

of each cable ($\overline{\tau}_i$). It is necessary that $\underline{\tau}_i < \overline{\tau}_i$, which imposes an additional constraint in the optimization.

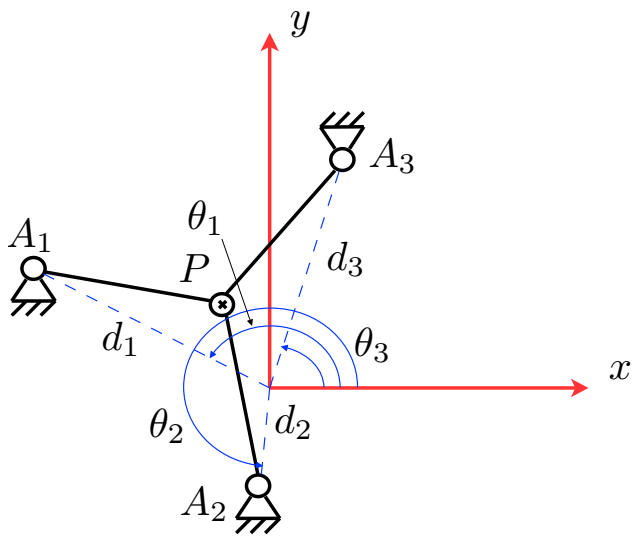


Figure F.2: Design variables of the planar CDPM.

The search space is multi-modal and may be separable depending on the selected design variables. Figure F.3 shows an example where θ_1 and θ_2 are design variables and the objective function is evaluated over a grid with a resolution of $\pi/180$ rad. In Figure F.3 the objective function is separable and there are six local optima, thus a local optimization routine may not converge to the best solution. Global optimization must be used to find a solution to the optimization problem. Differential Evolution (DE) is used with Deb's constraint handling technique.

Multiple optimization tests can be performed by fixing certain design variables. Four tests are considered, for which the design variables are:

1. $d_1,\ d_2,\ d_3$.

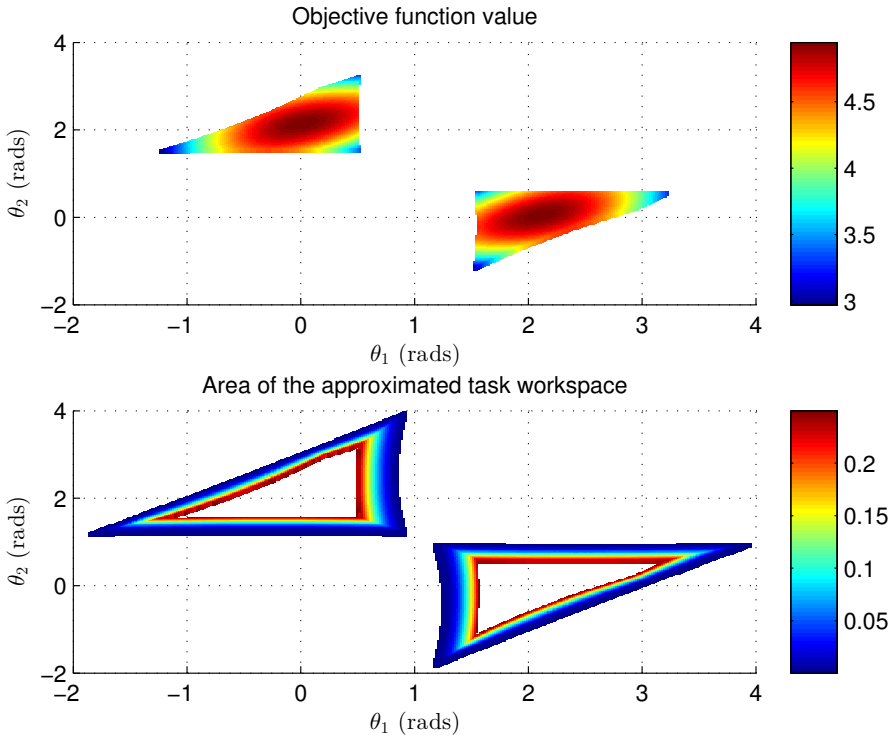


Figure F.3: The objective function value for each grid coordinate in the feasible search-space (top) and the area of the approximated task workspace for each grid coordinate in the infeasible search-space (bottom).

2. $\theta_1,\ \theta_2,\ \theta_3$

3. $d_1,\ d_2,\ d_3,\ \theta_1,\ \theta_2,\ \theta_3$

4. $d_1,\ d_2,\ d_3,\ \theta_1,\ \theta_2,\ \theta_3,\ \underline{\tau}_1,\ \underline{\tau}_2,\ \underline{\tau}_3,\ \overline{\tau}_1,\ \overline{\tau}_2,\ \overline{\tau}_3$

The initial value of each variable is:

$$\mathcal{D} = \left\{ \begin{array}{c} d_i = \sqrt{3}/3 \text{ m for } i = 1, 2, 3 \\ \theta_1 = 0 \text{ rad} \\ \theta_2 = 2\pi/3 \text{ rad} \\ \theta_3 = 4\pi/3 \text{ rad} \\ \underline{\tau}_i = 10 \text{ N for } i = 1, 2, 3 \\ \overline{\tau}_i = 100 \text{ N for } i = 1, 2, 3 \end{array} \right\} \tag{F.5}$$

The variable ranges are defined as:

$$\begin{aligned} d_i &\in [0, 2] \text{ m for } i = 1, 2, 3 \\ \theta_i &\in [0, 2\pi] \text{ rad for } i = 1, 2, 3 \\ \underline{\tau}_i &\in [10, 100] \text{ N for } i = 1, 2, 3 \\ \overline{\tau}_i &\in [10, 100] \text{ N for } i = 1, 2, 3 \end{aligned} \tag{F.6}$$

The mutation factor of DE is selected as 0.7 and the crossover rate is selected as 0.1. These selections are based on values found in the literature and experimental results. A population size of 50 was selected for each test, and the optimization is evaluated over 500 generations. Additional stopping criteria are not used for these tests. Each test is conducted for a task requiring an isotropic force of 20 N (*i.e.*, a force with a magnitude of 20 N in all directions). The task workspace is defined as the square axis-aligned box centred at the origin with side lengths of 0.5 m. The computation of the

reachable workspace and wrench workspace are accomplished by performing a line search along 300 equally-spaced rays.

Figure F.4a shows an optimized result of the workspaces for test 1. The optimal values for d_1, d_2, and d_3 are determined to be 1.244 m, 0.721 m, and 0.721 m, respectively. The solution is feasible since the approximated task workspace is equal to the task workspace, and the objective function value is 0.752 m^2. The legend supplied for Figure F.4a is applicable also for Figures F.4b, F.4c, and F.4d.

Figure F.4b shows the workspaces for test 2. The optimal values of θ_1, θ_2, and θ_3 are determined to be 1.119 rad, 3.138 rad, and 5.154 rad, respectively. The area of the approximated task workspace is 0.115 m^2 which yields an objective function value of 0.315 m^2. It is important to note that this solution is infeasible (*i.e.*, the approximated task workspace is smaller than the task workspace), and therefore the task cannot be completed with this manipulator. The task workspace is shown in the plot as a representation of the constraint violation.

Figure F.4c shows the workspaces for test 3. The optimal values of d_1, d_2, d_3, θ_1, θ_2, and θ_3 are determined to be 0.780 m, 0.924 m, 0.817 m, 1.105 rad, 3.135 rad, and 5.172 rad, respectively. The solution is feasible and the objective function value is 0.679 m^2.

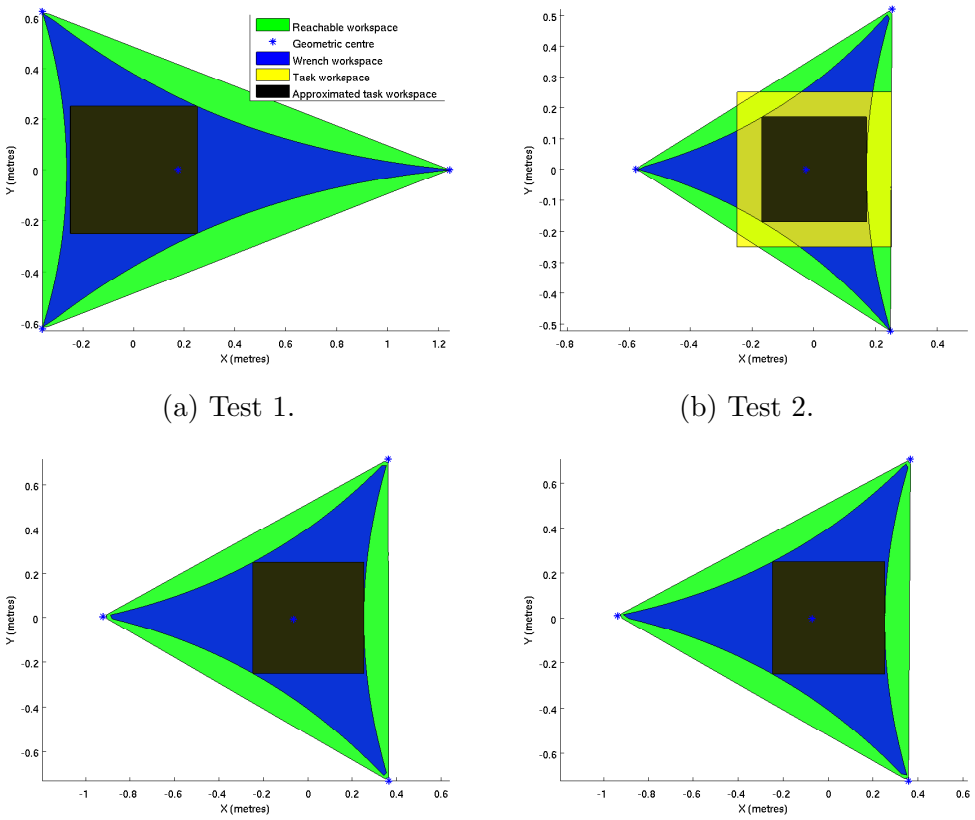
Figure F.4d shows the workspaces for test 4. The optimal values of d_1, d_2, d_3, θ_1, θ_2, θ_3, $\underline{\tau}_1$, $\underline{\tau}_2$, $\underline{\tau}_3$, $\overline{\tau}_1$, $\overline{\tau}_2$, and $\overline{\tau}_3$ are determined to be 0.795 m, 0.940 m, 0.806 m, 1.095 rad, 3.131 rad, 5.169 rad, 10 N, 10 N, 10 N, 99.89 N, 100 N,

and 100 N, respectively. The solution is feasible and the objective function value is 0.679 m^2.

The results of test 3 and test 4 are very similar in design variable values and also objective function value. During the optimization, the minimum cable tensions for each cable are optimized to the lower boundary. The maximum cable tensions are also optimized towards the upper boundary. The convergence behaviour of test 4 is provided in Figure F.5. Two plots are used to depict the convergence of the population. The first plot is the objective function value, which shows the change of the value of the objective function as the population evolves. The second plot shows information relating to the constraint violation. The vertical axis labelled "constraint amount" is used to represent the area of the approximated task workspace. When the area of the approximated task workspace is equal to the area of the task workspace, the design is feasible. During the initial stage where the population contains infeasible solutions, the constraint amount is maximized. This promotes feasibility throughout the population. Once the entire population is feasible, infeasible solutions cannot replace the feasible solutions and the objective function is then minimized. In Figure F.5, population feasibility occurs at generation 40.

F.3.1 Concurrent Implementation

The objective function evaluation is computationally expensive due to the computation of the wrench workspace. The mean computational time re-



(a) Test 1.

(b) Test 2.

(c) Test 3.

(d) Test 4.

Figure F.4: Task-optimized workspaces of the planar CDPM.

quired to evaluate one objective function value sequentially using the default design variable values and the previously described testing parameters is 0.0301 seconds (taken over 1000 runs). All of the tests in this section were completed on the Atlantic Computational Excellence Network (ACEnet) Fundy compute resource utilising Parallel Sun x4600 and x2200 AMD Opteron (dual-core) clusters with up to 16 processing cores available. A maximum of 10 processing cores were used.

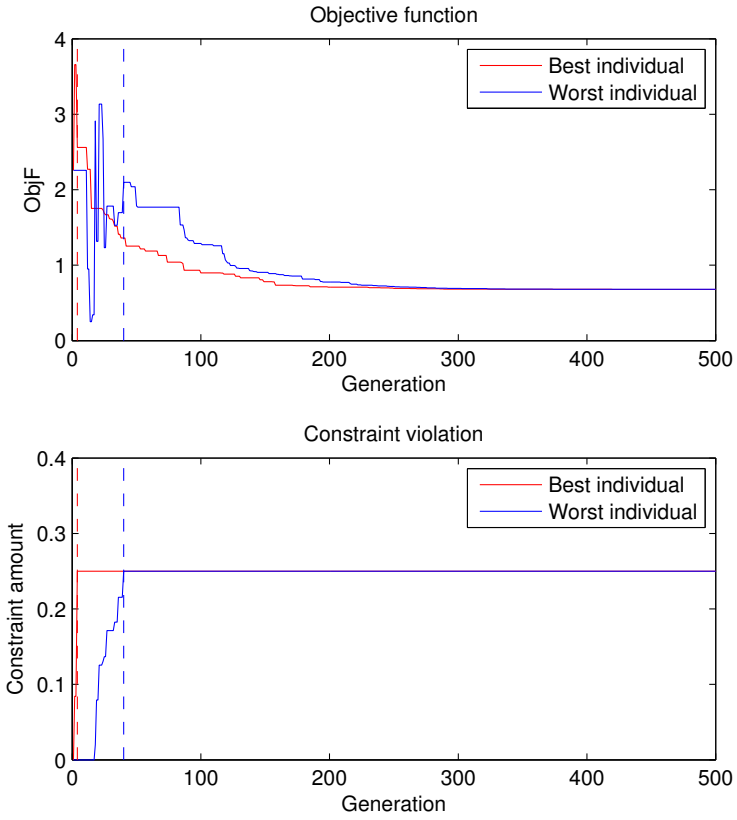


Figure F.5: Convergence results for test 4 of the best and worst individuals in the population. The vertical dashed lines signify a change from infeasible to feasible.

The selection of 300 equally-spaced rays provides a coarse approximation of the reachable and wrench workspaces, thus a realistic test would require testing many more rays. It is important to note that the 3-RPR CDPM with point-mass end-effector is a simplified type of PM. The 3-RPR CDPM is used in this work for illustration purposes. The methods presented in this work are applicable to many other PM architectures, such as the more

complicated 3-RRRS spatial PM. Wrench workspaces of the 3-RRRS spatial PM have been thoroughly studied in (Garg et al., 2009a,b). The computation of a wrench workspace for this spatial PM requires many more ray tests to yield a reasonable approximation of the reachable and wrench workspaces. In addition to the increased number of rays, each wrench test using the convex-hull method applied to the 3-RRRS spatial PM requires 64 times the number of computations required by the 3-RPR CDPM. The computations are proportional to the number of active actuators (the 3-RRRS spatial PM has three active joints in each of it's three limbs). The number of design variables may also increase accordingly.

The application of the optimization technique presented in this work to more complicated manipulators shows the requirement of using parallel computing techniques to achieve reasonable computational times. Several portions of the algorithm can be performed concurrently. Each ray test requires evaluation of the convex-hull method over a series of coordinates until the workspace boundary is accurately detected. Coordinates are selected along the ray according to the line search algorithm; therefore, there is a relationship between the coordinates along the ray which requires some communication. A set of rays can be initialized at the start of the program. The reachable and wrench workspace algorithms return a set of boundary coordinates, where each coordinate is obtained from a line search along a single ray. The majority of the computational time is consumed during the evaluation of the wrench workspace, followed by the evaluation of the reachable workspace. Therefore,

an effective parallelization strategy involves the concurrent evaluation of the set of rays for each reachable and wrench workspace test. The resolution of each workspace test is directly related to the number of rays. A higher resolution requires more rays. Parallel evaluation of the rays provides a fine-grained parallelism compared to the parallel evaluation of individuals for instance as the number of individuals is typically limited and is proportional to the number of design variables.

The following pseudo-code algorithm describes the evaluation of the workspaces and the application of DE. There are two parallel sections which are labelled "parallel" in the pseudo-code. The first parallel section evaluates each ray ($\mathbf{r}_i$) of the set of rays ($\mathbf{r}_n$) to obtain an approximation of the reachable workspace. Likewise, in the second parallel section, the set of rays are computed in parallel to obtain an approximation of the wrench workspace. OpenMP is used to evaluate each ray independently. The reachable and wrench workspace arrays have shared memory access. A synchronization point directly follows the parallel sections. The objective function value is stored in F, the constraint amount is stored in C, and the constraint satisfaction (yes or no) is stored in CS.

Additional considerations for concurrent computations include evaluating each individual in the population in parallel to provide another layer of parallelism. The determination of the objective function and constraint violations for each individual is an independent problem, thus each individual can be computed concurrently. The primary and secondary verification tests (in-

Algorithm 16: Architecture optimization pseudo-code

```
Initialize a random population;
for gen = 1 : ngen do
    /* Evaluate the population                    */;
    for i = 1 : npop do
        for r_i = 1 : r_n do /* Compute P_RW (parallel)        */
            P_RW(r_i) = boundary of P_RW along r_i
        for r_i = 1 : r_n do /* Compute P_WW (parallel)        */
            P_WW(r_i) = boundary of P_WW along r_i
        /* Synchronization                          */;
        Compute P_TW~;
        /* Evaluate objective function and constraints  */;
        F(i) = volume(P_RW \ P_WW) + volume(P_WW \ P_TW~);
        C(i) = P_TW~;
        CS(i) = Is the constraint satisfied ? yes(1) : no(0);
    Mutation and recombination;
    Selection using Deb's constraint handling technique;
```

cluded in the computation of $\mathbb{P}_{\widetilde{TW}}$) can also be computed in parallel. Each point of the approximated task workspace can be tested concurrently in the primary test. In the secondary test, each edge of the wrench workspace can be tested for intersection with the approximated task workspace concurrently. These considerations provide additional parallelism which can aid in reducing computation time.

For the parallel computing framework, the concurrent evaluation of the rays for the reachable and wrench workspaces are considered. OpenMP is used to decompose the workspace evaluations into a set of coarsely-grained concurrent evaluations. This allows for an effective parallelization strategy, by means of OpenMP directives, with minimal input from the programmer's

perspective. The major limitation of OpenMP in this scenario is that it is only applicable to non-distributed computing systems with shared memory resources.

An important consideration when applying parallel computing techniques to a problem is the speedup of the parallel implementation relative to the sequential implementation. OpenMP makes this comparison easy by setting the environment variable "OMP_NUM_THREADS". An overview of the speedup test is provided in Figure F.6. One thread is assigned per core. The data is obtained by averaging 3 runs using the same random seed for tests with 100, 300, and 500 rays. The rays are distributed among the processing cores using the *runtime* schedule clause provided by OpenMP. A comparison between the parallel implementation (running on 10 cores) and the sequential implementation (running on a single core) shows a 7.4× speed-up with the parallel implementation for 300 rays. This parallel implementation completed with an average time of 67 seconds. Amdahl's law predicts that the algorithm is approximately 96% parallelizable. Increasing the number of rays in the test improves the speed-up relative to the 100 ray test by up to 2.16% and 3.55% for the 300 ray and 500 ray tests respectively. This shows that the parallel efficiency increases with increasing numbers of rays.

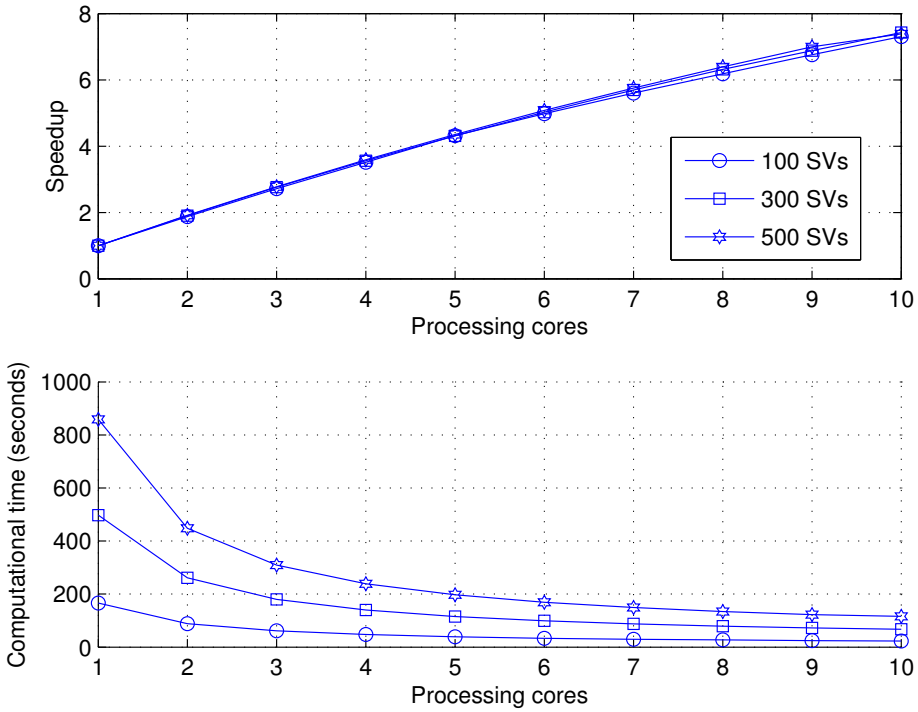


Figure F.6: Computational time and speed-up for test 4 for various numbers of processing cores.

F.4 Task-Optimized Design of the 3-RRR Planar Parallel Mechanism

Consider a grid discretization with a resolution ϵ, such that the set of discretized poses is given by $\mathcal{P} = grid(\mathbb{P}, \epsilon)$. The set of reachable discretized poses is denoted $\mathcal{P}_{RW}$. The set of wrench-capable discretized poses is denoted $\mathcal{P}_{WW}$.

While α-shapes are particularly useful for approximating the boundaries of

and detecting holes in workspaces, it is possible, and much more efficient, to compare workspaces based on the number of discretized poses contained in the workspace. Rather than applying the α-shape routine, then computing a value for the generalized volume of a workspace, the size of the workspace can be modelled by the number of discretized poses. The set difference between two workspaces becomes the difference in the pose count of the two workspaces.

The task workspace $\mathbb{P}_{TW}$ is continuous an must be discretized for easy comparison to $\mathcal{P}_{RW}$ and $\mathcal{P}_{WW}$. The set of discretized poses corresponding to the task workspace is denoted $\mathcal{P}_{TW}$, such that each grid point in $\mathbb{P}_{TW}$ is contained in $\mathcal{P}_{TW}$. $\mathcal{P}_{TW}$ is then inflated to contain the all adjacent discretized poses, such that $\alpha\text{-}shape(\mathcal{P}_{TW}) \supseteq \mathbb{P}_{TW}$. It is important to note that $\mathcal{P}_{TW}$ is a liberal estimate of $\mathbb{P}_{TW}$. In certain cases $\mathbb{P}_{TW} \subseteq \alpha\text{-}shape(\mathcal{P}_{WW})$ whereas $\alpha\text{-}shape(\mathcal{P}_{TW}) \nsubseteq \alpha\text{-}shape(\mathcal{P}_{WW})$.

F.4.1 Optimization Problem for a Grid Discretization

A task-optimized exact design $\mathcal{D}$ of the 3-RRR can be obtained for a given task using the following primary and secondary objectives:

1. Primary objective:

    - *Ensure task completion.* Verifying task completion can be performed by the constraint $\mathcal{P}_{TW} \subseteq \mathcal{P}_{WW}$. which ensures that each grid point in $\mathcal{P}_{TW}$ is also in $\mathcal{P}_{WW}$.

2. Secondary objectives:

   - *Minimize the size of the mechanism.* Assuming that $\mathbb{P}_{TW}$ is geometrically centred at the origin, O, the initial search space, denoted by $\mathcal{P}$, is proportional to the size of the mechanism where $\mathcal{P}_x = \mathcal{P}_y = [-w, w]$ ($w = \max(||\mathbf{a}_i|| + ||\mathbf{d}_i|| + r_i + l_i)$) and $\mathcal{P}_\psi = 0$. Minimising the size of the mechanism is equivalent to minimising the number of grid points in $\mathcal{P}$.

   - *Minimize the poses not used by the task.* The set $\mathcal{P}_{RW} \backslash \mathcal{P}_{WW}$ represents the unusable poses in the reachable workspace, considering the task wrench requirements. Likewise, the set $\mathcal{P}_{WW} \backslash \mathcal{P}_{TW}$ represents the unneeded poses in the wrench workspace, considering the task workspace requirements. The two sets can be minimized in order to aid in reducing the size of the mechanism.

     Without this objective, the optimization would optimize the design with only minor consideration for wrench-capable poses. That is, any wrench-capable pose outside of $\mathcal{P}_{TW}$ would have no impact on the optimization of. Thus, this objective helps guide the optimization to portions of the search space which contain a greater quantity of wrench-capable poses. These portions of the search space are more likely to contain a feasible solution.

The optimization problem (F.1) may be modified to account for the primary and secondary objectives. All objectives will be considered equally

important, such that coefficients K_1, K_2, and K_3 are set to 1. For a grid discretization, the simplified optimization problem is:

$$\begin{aligned} &\text{minimize :} \quad K_1 count(\mathcal{P}_{RW} \backslash \mathcal{P}_{WW}) + K_2 count(\mathcal{P}_{WW} \backslash \mathcal{P}_{TW}) + K_3 count(\mathcal{P}) \\ &\text{subject to :} \\ &\qquad \mathcal{P}_{TW} \subseteq \mathcal{P}_{WW} \\ &\qquad \mathcal{D} \in [\mathcal{D}] \end{aligned} \tag{F.7}$$

where the function *count*() returns the number of discretized poses in the set.

F.4.2 Task-Optimized Design Solutions

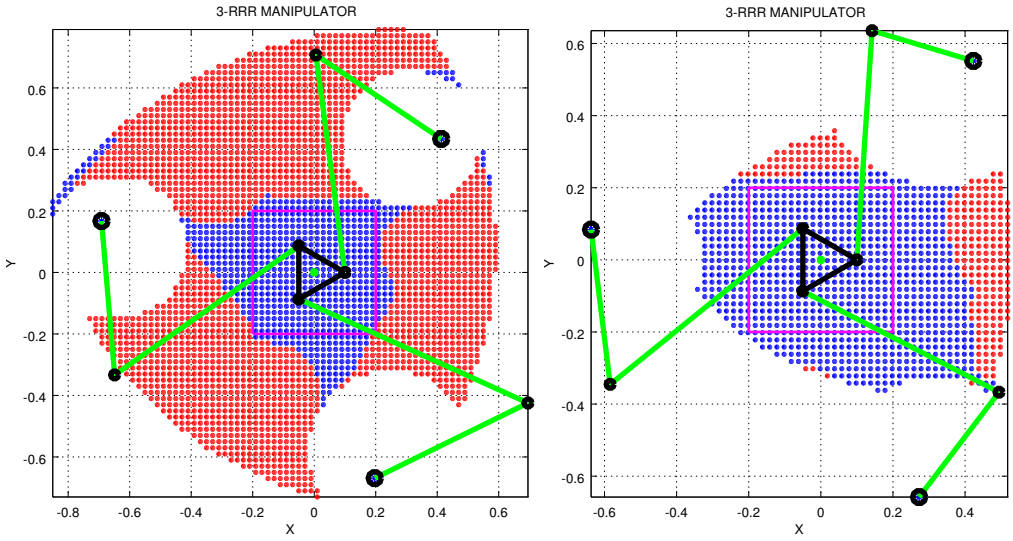
The moving platform geometry and orientation are fixed, such that the moving platform is an equilateral triangle with 0.173 m edge lengths and $\psi = 0$ rad. The actuator capabilities are fixed as $\tau_i = [-10, 10]$ Nm. Elbow-right configurations are selected for each limb. The variable design parameters consist of: proximal and distal link lengths, l_i and r_i, and base actuator locations, $\mathbf{a}_i$, where the set of variable constraints, $[\mathcal{D}]$, is

$$[\mathcal{D}] = \left\{ \begin{array}{l} r_i = [0.05, 1] \text{ m for } i = 1, 2, 3 \\ l_i = [0.05, 1] \text{ m for } i = 1, 2, 3 \\ \mathbf{a}_i = \begin{pmatrix} [0.2, 2]\cos([0, 2\pi])) \\ [0.2, 2]\sin([0, 2\pi])) \\ 0 \end{pmatrix} \text{ m for } i = 1, 2, 3 \end{array} \right\} \tag{F.8}$$

Applying the Differential Evolution global optimization algorithm utilising Deb's constraint handling technique to the optimization problem (F.7) for the design variables in Equation (F.8), a task-optimized design of the 3-$\underline{\text{R}}$RR is obtained. A population size of 20 is used for DE. The $\mathcal{F}_{desired}$ are selected as $f_x = 0$ N, $f_y = 0$ N and $m_z = [-5, 5]$ Nm and the $\mathbb{P}_{TW}$ is a square centred at the origin with 0.4 m edge lengths. A grid resolution of $\epsilon = 0.01$ m is selected. The algorithm converges to a solution in 500 generations, where the best design, and corresponding $\mathcal{P}_{RW}$ and $\mathcal{P}_{WW}$ are shown for generations 125, 250, and 500 in Figure F.7. The optimal solution, $\mathcal{D}^*$, returned by the algorithm (Figure F.7 (c)) satisfies all of the constraints and is therefore able to fully complete the task. The solution is determined to be:

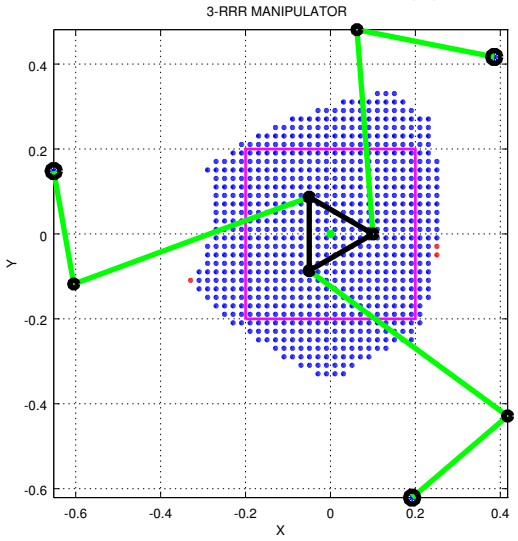
$$\mathcal{D}^* = \left\{ \begin{array}{l} r_1 = 0.3297 \text{ m},\ r_2 = 0.4824 \text{ m},\ r_3 = 0.2700 \text{ m} \\ l_1 = 0.5911 \text{ m},\ l_2 = 0.2959 \text{ m},\ l_3 = 0.5792 \text{ m} \\ \mathbf{a}_{1x} = 0.3858 \text{ m},\ \mathbf{a}_{1y} = 0.4167 \text{ m} \\ \mathbf{a}_{2x} = -0.6516 \text{ m},\ \mathbf{a}_{2y} = 0.1480 \text{ m} \\ \mathbf{a}_{3x} = 0.1923 \text{ m},\ \mathbf{a}_{3y} = -0.6211 \text{ m} \end{array} \right\} \tag{F.9}$$

It also optimizes the secondary objectives and has a very compact design; as well, the majority of the reachable workspace is part of the wrench workspace which is a result of the minimization of the unusable poses. It can be noted that the selection of limb configurations (elbow-left or elbow-right) will result in a different wrench workspace, therefore incorporating limb configurations into the optimization may result in more optimal designs.



(a) Generation 125.

(b) Generation 250.



(c) Generation 500.

Figure F.7: Task-optimized 3-RRR results.

Appendix G

The Canonical Differential Evolution Algorithm

Differential Evolution (DE) attempts to search for a global optimum point in a d-dimensional real parameter space. An initial population consisting of n_{pop} individuals is randomly generated as a constrained d-dimensional vector, such that each individual i is represented as:

$$\mathbf{x}_i = (x_{1i}, x_{2i}, \ldots, x_{di})^T \tag{G.1}$$

where the elements may be constrained by the restricted domains:

$$x_1 \in [x_1],\ x_2 \in [x_2],\ \ldots,\ x_d \in [x_d] \tag{G.2}$$

Each individual forms a candidate solution to the optimization problem.

The role of DE is to perturb the population, such that the candidate solutions better solve the optimization problem. Each individual j in the initial population may be initialized as:

$$\mathbf{x}_j = \left\{ x_i = \underline{[x_i]} + rand[0,1](\overline{[x_i]} - \underline{[x_i]}),\ \text{for } i = 1, \ldots, d \right\} \tag{G.3}$$

where $rand[0,1]$ generates a uniformly distributed random number in the range $[0,1]$.

The mutation operation creates a mutant vector $\mathbf{v}_j$ for each individual $\mathbf{x}_j$. Using the "DE/rand/1" strategy, the mutant vector is created as:

$$\mathbf{v}_j = \mathbf{x}_{R1} = F(\mathbf{x}_{R2} - \mathbf{x}_{R3}) \tag{G.4}$$

where $R1$, $R2$, and $R3$ are random and mutually exclusive indices from the population in the range $[1, n_{pop}]$, and F is the *scaling factor*.

The crossover operation in DE mixes the elements of the mutant vector $\mathbf{v}_j$ and individual $\mathbf{x}_j$ to form a trial vector $\mathbf{u}_j$. Using the *binomial crossover* method, the trial vector is created as:

$$\mathbf{u}_j = \begin{cases} v_{ij} & \text{if } j = k \text{ or } rand[0,1] < Cr \\ x_{ij} & \text{otherwise} \end{cases} \tag{G.5}$$

where k is a random index from the population in the range $[1, n_{pop}]$, and Cr is the *crossover rate*.

The selection operation determines whether the trial vector $\mathbf{u}_j$ replaces the current individual $\mathbf{x}_j$ in the population. The selection operation for a minimization problem is given as

$$\mathbf{x}_j = \begin{cases} \mathbf{u}_j & \text{if } f(\mathbf{u}_j) \leq f(\mathbf{x}_j) \\ \mathbf{x}_j & \text{otherwise} \end{cases} \tag{G.6}$$

where f evaluates the objective function.

Each generation from 1 to n_{gen} iterates mutation, crossover, and selection for each individual $j = 1, \ldots, n_{pop}$. The best individual in the population is returned as the best solution found for the optimization problem. Additional stopping criteria may be incorporated to limit the number of objective function values. A recent survey of the advances in DE is given in (Das et al., 2016).

Vita

Surname: Pickard **Given Names**: Joshua K.

Date of birth: April 23, 1990

Place of birth: Woodstock, N.B., Canada

Educational Institutions Attended:

University of New Brunswick (UNB)	2008 to 2018
Inria Sophia-Antipolis, France	Dec. 2015 to June 2016

Degrees Awarded:

BScE (Mechanical with a Mechatronics option) UNB	2008 to 2012

Honours and Awards:

CCToMM Student Delegates Award	2017
IFToMM Young Delegates Program Grant	2016
Mitacs Globalink Research Award - Inria	2015 to 2016
CCToMM Student Delegates Award	2015
IFToMM Young Delegates Program Grant	2014
10th MEGS Conference - 1st Place Poster Presentation	2014
The President's Doctoral Tuition Award	2014 to 2017
Alexander Graham Bell Canada Graduate Scholarship: PGSD	2014 to 2017
Alexander Graham Bell Canada Graduate Scholarship: CGSM	2013
Board of Governors Merit Awards for Graduate Studies	2013 to 2017
Mechanical Engineering Faculty/Staff Graduate Prize	2012
UNB Third Century Fund Scholarship	2011

UNBF Student Union Human Development Scholarship	2010
Edwin Jacob Special University Scholarship	2008
Richard B. Logie Memorial Scholarship	2008

Publications:

1. J. K. Pickard. Geometry-based method for determining the wrench capabilities of redundant parallel manipulators. In *Mechanical Engineering Graduate Students Conference (MEGSC8)*, Fredericton, Canada, November 8 2012a;

2. J. K. Pickard. Geometry-based method for determining the wrench capabilities of redundant parallel manipulators. In *Mechanical Engineering Graduate Students Conference Poster Presentation (MEGSC8)*, Fredericton, Canada, November 8 2012b;

3. J. K. Pickard and J. A. Carretero. Applications of interval analysis techniques for solving constraint satisfaction problems. In *Mechanical Engineering Graduate Students Conference Poster Presentation (MEGSC9)*, Fredericton, Canada, October 10 2013b;

4. J. K. Pickard and J. A. Carretero. Wrench workspaces of redundantly actuated spatial parallel manipulators using minimum wrench capabilities. In *Proceeding of the 2013 CCToMM Mechanisms, Machines, and Mechatronics Symposium*, Montreal, Canada, May 30–31 2013a;

5. J. K. Pickard and J. A. Carretero. A comparison of wrench-feasible workspaces using interval analysis and convex linear mapping techniques. In *Proceedings of the Canadian Society for Mechanical Engineering International Congress 2014*, Toronto, ON, Canada, June 1–4 2014;

6. J. K. Pickard, J. A. Carretero, and V. C. Bhavsar. Task-optimized cable-actuated planar parallel manipulator architecture and its concurrent implementation. In *Proceedings of the International Conference on High Performance Computing and Simulation (HPCS 2014)*, Bologna, Italy, July 21–25 2014;

7. J. K. Pickard and J. A. Carretero. An interval analysis method for wrench workspace determination of parallel manipulator architectures. In *Proceeding of the 2015 CCToMM Mechanisms, Machines, and Mechatronics Symposium*, Ottawa, Canada, May 28–29 2015a;

8. J. K. Pickard and J. A. Carretero. An interval analysis method for wrench workspace determination of parallel manipulator architectures. *Transactions of the Canadian Society for Mechanical Engineering*, 40 (2):139–154, November 2015c;

9. J. K. Pickard and J. A. Carretero. Design optimisation of the 3-RRR planar parallel manipulator via wrench capability analysis. In *Proceedings of the 14th World Congress in Mechanism and Machine Science*, Taipei, Taiwan, October 25–30 2015b;

10. J. K. Pickard, J. A. Carretero, and V. C. Bhavsar. On the convergence and origin bias of the teaching-learning-based-optimization algorithm. *Applied Soft Computing*, 46(1):115–127, September 2016a;

11. J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Accounting for tolerances in the design parameters of the 3-RRR. In *Proceeding of Advances in Robot Kinematics*, Grasse, France, June 27–30 2016b;

12. J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Towards the appropriate synthesis of the four-bar linkage. In *Proceedings of the 2017 CCToMM Mechanisms, Machines, and Mechatronics Symposium*, Montreal, Canada, May 25–26 2017a;

13. J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Analysis methods for the 3-RRR with uncertainties in the design parameters. In Jadran Lenarčič and Jean-Pierre Merlet, editors, *Advances in Robot Kinematics 2016*, pages 109–118. Springer International Publishing, 2017b;

14. J. K. Pickard and J. A. Carretero. Towards the appropriate synthesis of the four-bar linkage. *Transactions of the Canadian Society for Mechanical Engineering*, Accepted November, 2017 ;

15. J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Appropriate analysis of the four-bar linkage. *Mechanism and Machine Theory*, Submitted 2018a;

16. J. K. Pickard, J. A. Carretero, and J.-P. Merlet. Appropriate synthesis of the four-bar linkage. *Mechanism and Machine Theory*, Submitted 2018b.