Semidefinite Relaxations for Geometric Problems in Robotics

by

Matthew Giamou

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Institute for Aerospace Studies
University of Toronto

Abstract

Semidefinite Relaxations for
Geometric Problems in Robotics

Matthew Giamou
Doctor of Philosophy
Graduate Department of Institute for Aerospace Studies
University of Toronto
2022

Mobile robots perceive and move through the three-dimensional space of the approximately Euclidean world we share with them. In order to safely and accurately accomplish their goals, they must be able to reason about the nonlinear and nonconvex geometry of the manifold of rotations. Without this ability, tracking the poses of objects from noisy measurements and avoiding obstacles in their environment becomes impossible. Traditional approaches use *local* information and structure to estimate and optimize rotations of interest, making them susceptible to suboptimal performance. In this dissertation, we apply recent advancements in *global* convex optimization to two fundamental geometric problems in robotics: extrinsic sensor calibration and inverse kinematics in cluttered workspaces. We begin with a summary and extension of the semidefinite relaxation machinery that we apply to both problems. This machinery is used to develop fast and accurate extrinsic calibration algorithms with novel performance guarantees provided by certificates of global optimality. We proceed to develop a novel perspective of inverse kinematics inspired by noisy state estimation problems, leading to fast and accurate algorithms appropriate for a variety of challenging scenarios. We provide free and open source implementations of our algorithms, and demonstrate their superiority over conventional approaches on a variety of simulated and real-world data.

Epigraph

Upward, not Northward.

EDWIN A. ABBOTT, *Flatland*

A point is that which has no part.

EUCLID, *Elements*

Do you have the patience to wait until your mud settles and the water is clear?

LAO TZU, *Tao Te Ching*

To my wife Cassandra.

Acknowledgements

I am thankful to my sister for teaching me how to read, my father for hanging the periodic table in our dining room, and my mother for ensuring I read real literature. I am in awe of my grandparents for undertaking the journeys that have led me to this dissertation: thank you for your sacrifices and all the love that went into building our family. My aunts, uncles, cousins, and in-laws deserve credit for always believing in me and cheering me on.

I am unbelievably fortunate to have a second chosen family made up of dear friends I've known for decades. Since we generally try not to bore one another with too much shop talk, you should know that our rich social life provided the perfect shelter from the toils of grad school.

What can I possibly say about my colleagues in STARS Laboratory that I haven't already loudly proclaimed? You're all the best, and we're the most fun and supportive lab imaginable. I am still furious that COVID cut our legendary intramural soccer run short after a single wonderful season. Not being able to regularly see you all in person for years and collaborate on a whiteboard is the worst thing to happen to me on my doctoral journey. Thank you all for your humour, good natures, and inexhaustible fount of creative research ideas. My greatest debt is to Filip: our years of collaboration not only kept me productive, but were also incredibly fun! I could not have asked for a more supportive "research spouse", and this dissertation has largely been shaped by our partnership.

My unbelievable luck also extends to my dream committee! Gabe and Prasanth: I've admired both your knowledge and charisma since I was a fledgling undergraduate, and you are both role models for the type of professor I'd like to become. I wish there was enough time to address all the questions and interesting research ideas you proposed during my DEC meetings, and I'll always be thankful for your patience and genuine interest in my success.

Jonathan, it goes without saying that this thesis would not exist without you. You convinced me to stick with academia, and your immense faith in me has been a source of confidence and calm for nearly five years. Your commitment to high quality in every aspect of research has been a profound influence, and I don't think I would have felt bold enough to dive into "simple" problems like inverse kinematics without you there to cheer me on. I am also incredibly thankful that you encouraged me to take on mentorship and teaching opportunities and gave me the freedom to experiment in all aspects of my work. You are everything I could have asked for in a supervisor!

Finally and foremost, I am thankful for my dear wife Cassandra's undying love and support. You have put up with me for eleven years, and I could not have completed this dissertation without you. You fill my life outside of work with incredible joy, and you have believed in me and my research from day one.

Contents

List of Tables

List of Figures

Notation

- a : Lower-case Latin and Greek letters in this font are real scalars.
- $\mathbf{a}$: Lower-case Latin and Greek letters in this font are real column vectors.
- $\boldsymbol{a}$: Symbols in this font are real column vectors in homogeneous coordinates.
- $\mathbf{A}$: Capital Latin and Greek letters in this font are real matrices.
- $\mathbf{A}^{(i)}$: The ith column of matrix $\mathbf{A}$.
- $\mathbf{A}_{i,j}$: The element of $\mathbf{A}$ in column i and row j.
- $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$: Normally distributed with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$.
- $\hat{\cdot}$: An estimate of a scalar, vector, or matrix.
- $\tilde{\cdot}$: A scalar, vector, or matrix corrupted by noise.
- $\mathbb{E}[\cdot]$: The expectation operator.
- $\mathbf{1}$: The vector of ones.
- $\mathbf{I}$: The identity matrix.
- $\mathbf{0}$: The zero matrix.
- $\underrightarrow{\mathcal{F}}_a$: A reference frame in three dimensions.
- $\mathbf{p}_a^{cb}$: A vector from point b to point c (denoted by the superscript) and expressed in $\underrightarrow{\mathcal{F}}_a$ (denoted by the subscript).
- $\mathbf{R}_{ab}$: The 3×3 rotation matrix that transforms vectors from $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$: $\mathbf{p}_a^{cb} = \mathbf{R}_{ab}\mathbf{p}_b^{cb}$.
- $\mathbf{T}_{ba}$: The 4×4 transformation matrix that transforms homogeneous points from $\underrightarrow{\mathcal{F}}_a$ to $\underrightarrow{\mathcal{F}}_b$: $\boldsymbol{p}_b^{cb} = \mathbf{T}_{ba}\boldsymbol{p}_a^{ca}$.
- $(\cdot)^\wedge$: An operator associated with the Lie algebra for rotations and poses. It produces a matrix from a column vector.
- $(\cdot)^\vee$: The inverse operation of $(\cdot)^\wedge$.
- $\otimes$: The matrix Kronecker product.
- $\mathrm{vec}(\cdot)$: The column-wise vectorization operator.
- $[n]$: The set of integer indices $\{1, \ldots, n\}$.
- $\mathbb{S}^n$: The set of $n \times n$ real symmetric matrices.
- $\mathbb{S}^n_+$: The set of $n \times n$ real symmetric positive semidefinite matrices.
- $\mathbb{S}^n_{++}$: The set of $n \times n$ real symmetric positive definite matrices.
- ∇f : The gradient (Jacobian) of a scalar-valued (vector-valued) function f.

List of Abbreviations

ACQ Abadie Constraint Qualification

BnB Branch-and-Bound

BSOS Bounded Degree Sum-of-Squares

CIDGIK Convex Iteration for Distance Geometric Inverse Kinematics

CLIK Closed-Loop Inverse Kinematics

COSMO Conic Operator Splitting Method

CPU Central Processing Unit

DGP Distance Geometry Problem

DOF Degrees of Freedom

GPS Global Positioning System

IK Inverse Kinematics

IMU Inertial Measurement Unit

IPOPT Interior Point Optimizer

KKT Karush-Kuhn-Tucker

LICQ Linear Independence Constraint Qualification

MILP Mixed Integer Linear Program

MLE Maximum Likelihood Estimation

QCQP Quadratically Constrained Quadratic Program

PDF Probability Density Function

PD Positive Definite

PGO Pose Graph Optimization

PSD Positive Semidefinite

RAM Random-Access Memory

RIP Running Intersection Property

SCS Splitting Conic Solver

SDP Semidefinite Program/Programming

SQP Sequential Quadratic Programming

SLAM Simultaneous Localization and Mapping

SLSQP Sequential Least-Squares Quadratic Programming

SNL Sensor Network Localization

SOS Sum-of-Squares

ZDG Zero-Duality-Gap

Chapter 1

Introduction

> The whole is greater than the part.
>
> EUCLID, *Elements*

With few exceptions, autonomous robots need to navigate through three-dimensional space. To achieve this, robots must be equipped with the ability to represent and track the motion of their own bodies along with elements of their environment. This capability is intrinsic to many estimation and planning components in an autonomous guidance, navigation, and control pipeline. While the *positions* of objects of interest can be described in the vector space $\mathbb{R}^3$, their *orientations* are elements of a nonlinear manifold describing 3D rotations: the Lie group SO(3). In the context of optimization, ensuring that a variable resides on this manifold (i.e., it is in fact a valid rotation) introduces difficult nonlinear and nonconvex constraints. Dealing with this nonlinearity is not unique to robotics: it is a central challenge in aerospace engineering, biomedical imaging, and many other application domains.

The heart of this dissertation is the application of convex analysis and optimization to the nonlinearities that result from a robot's need to estimate and control rotations. The approach we develop is enabled by the recent improvement of convex optimization software, as well as theoretical progress that provides insights into the applicability of convex relaxations for challenging nonconvex problems in the applied sciences. A key feature of this approach is the *global* nature of convex optimization techniques, which is relevant to robotics for two important reasons:

1. truly autonomous systems cannot rely on *local* optimization that is sensitive to initial conditions which are often provided or tuned by a human operator; and

2. unlike stable aircraft in cruise or systems mostly limited to two-dimensional motion (e.g, many industrial machines or warehousing systems), autonomous mobile robots may find themselves or objects of interest in orientations that span the entirety of SO(3).[1]

[1] This is a characteristic shared with many astronautical applications.

We will demonstrate that applying a convex relaxation-based perspective to fundamental problems in robotics leads to algorithms which are geometrically elegant, perform well, and provide theoretical guarantees that are absent from traditional approaches. In the remainder of this introduction, we provide a high-level review of related literature before summarizing the structure and contributions of this dissertation.

1.1 Related Work: A Bird's-Eye View

Our global approach is complementary to the Lie-theoretic machinery developed in Barfoot (2017), which exploits the local structure of the SO(3) and SE(3) manifolds to construct and optimally solve precise probabilistic formulations of noisy state estimation problems in robotics. Olsson (2009) is a doctoral thesis that similarly applies global optimization methods to problems in computer vision: since its publication, convex techniques have become increasingly popular in robotics. Global optimality is particularly important to roboticists because of the safety requirements unique to our field. Accordingly, the publication of a *certifiably* globally optimal solution to simultaneous localization and mapping (SLAM) in Rosen et al. (2019) ignited a research trend (see, for example, Rosen et al. (2021) for a comprehensive review and discussion).

Applications of interest for which convex semidefinite programming (SDP) relaxations have proven fruitful include point cloud registration (Briales et al., 2017; Yang et al., 2020), variants of SLAM (Tian et al., 2021; Briales and Gonzalez-Jimenez, 2017; Fan et al., 2020; Iglesias et al., 2020; Holmes and Barfoot, 2022), rotation averaging (Eriksson et al., 2018), and relative pose estimation (Briales et al., 2018; Zhao, 2020; Garcia-Salguero et al., 2021). The first part of this dissertation is based closely on Cifuentes et al. (2022), which presents a broadly applicable theory for proving the "stability" of SDP relaxations to the aforementioned problems. We extend this theory to problems involving inequality constraints, and we apply the certifiable global optimization paradigm to extrinsic calibration, which is the problem of finding the relative pose of two sensors mounted on the same rigid body.

We would be remiss to ignore a powerful algebraic alternative to SDP relaxations that has emerged in recent years. One example of this approach described is in Wu et al. (2022), which uses a Gröbner-basis method to find globally optimal solutions to quadratic pose estimation problems (QPEPs). While extremely effective for optimization problems involving a single rotation or pose variable, algebraic approaches suffer from poor scaling as the number of variables increase, limiting their promise for more complex geometric problems in robotics (Wu et al., 2020).

1.2 Thesis Structure and Original Contributions

This dissertation presents material from five published contributions: three journal papers, and two peer-reviewed conference papers. In addition to these works, a few results and details that are in preparation for submission to academic journals at the time of writing have been included. In Chapter 2, we introduce the basic notation and mathematical concepts needed to understand the main body of work. What follows is a concise summary of each remaining chapter and its associated published contributions.

1. **Local Stability of SDP Relaxations over Semialgebraic Sets**
   Chapter 3 begins with some fundamental tools in the form of convexity, quadratically constrained quadratic programs (QCQPs), Lagrangian duality, constraint qualification, and semidefinite programming. We proceed with a summary of the relevant theory for Cifuentes et al. (2022) and present a novel extension of local stability results to QCQPs involving inequality constraints. This extension is not yet published, but simplifies and extends some of the contributions in three publications: Giamou et al. (2019), Marić et al. (2020), and Wise et al. (2020).

2. **Globally Optimal Extrinsic Calibration**
   In Chapter 4, we describe a certifiably globally optimal approach to extrinsic sensor calibration for mobile robots. This chapter is based on two publications: Giamou et al. (2019) and Wise et al. (2020). In addition to summarizing these publications, we also present a novel maximum likelihood interpretation of our approach, and use the novel refinements of the theory in Chapter 3 to discuss and characterize extensions to our approach.

3. **Global Optimization for Planar and Spherical Inverse Kinematics**
   Chapter 5 presents SOS-IK (Sum-of-Squares Inverse Kinematics), an algorithm developed in Marić et al. (2020). SOS-IK is notable in that it solves inverse kinematics, a planning problem, with ideas from the literature on noisy state estimation, highlighting the ever-present duality between estimation and "control." Once again, we include extensions and simplifications to the theory in Marić et al. (2020) based on the methods developed in Chapter 3.

4. **Semidefinite Programming for Revolute Inverse Kinematics**
   Chapter 6 adapts the ideas in Chapter 5 to more general revolute manipulators and introduces obstacles in the robot's workspace. It is based on two published contributions: Marić et al. (2021) and Giamou et al. (2022).

Finally, Chapter 7 recapitulates the main contributions and themes of this dissertation, while also briefly sketching a path for future research. Robotics and related fields are full of problems to which the methods of this work are directly applicable, and there is also a parallel need for further field testing of these ideas on challenging robotics applications outside of academic labs.

Chapter 2

Mathematical Preliminaries

> Numbers and rows and series—the nightmare and malediction harrowing pure thought and pure time—seemed bent on mechanizing his mind.
>
> VLADIMIR NABOKOV, *Ada, or Ardor*

This chapter covers the mathematics, primarily geometry and probability, required to understand subsequent chapters. The notation and conventions used draw heavily from Barfoot (2017).

2.1 Rotations and Poses

We describe the position of a point in d-dimensional physical space with a vector $\mathbf{p}_a \in \mathbb{R}^d$ containing the coordinates of the point with respect to frame of reference $\underrightarrow{\mathcal{F}}_a$. Since we are modelling classical mechanics in realistic environments for physical robots, $d \in \{2, 3\}$ throughout this work. When the vector describes the relative translation from point b to c, we write $\mathbf{p}_a^{cb}$—this notation is especially useful when b and c represent the origins of coordinate frames $\underrightarrow{\mathcal{F}}_b$ and $\underrightarrow{\mathcal{F}}_c$.

2.1.1 Rotation Matrices

The $d \times d$ rotation matrix $\mathbf{R}_{ab}$ translates a vector of coordinates in $\underrightarrow{\mathcal{F}}_b$ to its equivalent expression in a frame $\underrightarrow{\mathcal{F}}_a$ which shares its origin with $\underrightarrow{\mathcal{F}}_b$:

$$\mathbf{p}_a = \mathbf{R}_{ab}\mathbf{p}_b. \tag{2.1}$$

The set of valid rotation matrices is the *special orthogonal* matrix Lie group $\mathrm{SO}(d)$, which contains all orthogonal matrices with unit determinant:

$$\begin{aligned} \mathrm{SO}(d) : \mathbf{R} &\in \mathbb{R}^{d\times d} \\ \text{s.t.} \quad \mathbf{R}^\top \mathbf{R} &= \mathbf{R}\,\mathbf{R}^\top = \mathbf{I} \\ \det(\mathbf{R}) &= 1. \end{aligned} \tag{2.2}$$

For $d = 3$, we will find it useful to replace the determinant constraint with an equivalent expression using the following cross-product identities in $\mathbb{R}^3$:

$$\mathbf{R}^{(i)} \times \mathbf{R}^{(j)} = \mathbf{R}^{(k)},\ i, j, k \in \text{cyclic}(1, 2, 3), \tag{2.3}$$

where $\mathbf{R}^{(i)}$ is the ith 3×1 column of $\mathbf{R}$, and cyclic$(1, 2, 3)$ indicates the cyclic permutations (including identity) of the set $\{1, 2, 3\}$. These "right-handedness" constraints distinguish $\mathrm{SO}(3)$ from its complement in the orthogonal matrix Lie group $\mathrm{O}(3)$, which contains reflections. Since $\mathrm{O}(3)$ is a disjoint union of these two sets, orthogonality constraints alone are often sufficient for estimation problems (Rosen et al., 2019). However, we will demonstrate in Chapter 4 that the handedness constraints are useful for many optimization problems. Equation (2.3) replaces the cubic determinant constraint in Equation (2.2) with three quadratic constraints, giving us a purely quadratic polynomial description of $\mathrm{SO}(3)$. This description of rotation matrices is therefore a quadratic algebraic variety and in Chapter 3 will be used to connect estimation over $\mathrm{SO}(3)$ to the rich literature of optimization methods using concepts from algebraic geometry (Cifuentes et al., 2022).

2.1.2 Spatial Transformations

In order to completely describe the pose of rigid bodies in Euclidean space, we introduce homogeneous transformation matrices and the special Euclidean group $\mathrm{SE}(d)$:

$$\begin{aligned} \mathrm{SE}(d) : \mathbf{T}_{bc} &\in \mathbb{R}^{(d+1)\times(d+1)} \\ \text{s.t. } \mathbf{T}_{bc} &= \begin{bmatrix} \mathbf{R}_{bc} & \mathbf{p}_b^{cb} \\ \mathbf{0}_{1\times d} & 1 \end{bmatrix} \\ \mathbf{R}_{bc} &\in \mathrm{SO}(d),\ \mathbf{p}_b^{cb} \in \mathbb{R}^d. \end{aligned} \tag{2.4}$$

To make use of transformation matrices, we introduce homogeneous coordinates:

$$\boldsymbol{p} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}. \tag{2.5}$$

This allows us to convert coordinates for a point between frames $\underrightarrow{\mathcal{F}}_a$ and $\underrightarrow{\mathcal{F}}_b$ which do not necessarily share an origin as follows:

$$\begin{bmatrix} \mathbf{p}_b^{ab} \\ 1 \end{bmatrix} = \boldsymbol{p}_b^{ab} = \mathbf{T}_{bc}\boldsymbol{p}_c^{ac} = \begin{bmatrix} \mathbf{R}_{bc} & \mathbf{p}_b^{cb} \\ \mathbf{0}_{1\times d} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_c^{ac} \\ 1 \end{bmatrix}. \tag{2.6}$$

Lastly, we note that since $\mathbf{R}^{-1} = \mathbf{R}^\top \; \forall \; \mathbf{R} \in \mathrm{SO}(d)$, transformation matrices can also be efficiently inverted:

$$\mathbf{T}_{bc}{}^{-1} = \begin{bmatrix} \mathbf{R}_{bc}^\top & -\mathbf{R}_{bc}^\top \mathbf{p}_b^{cb} \\ \mathbf{0}_{1\times d} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{cb} & \mathbf{p}_c^{bc} \\ \mathbf{0}_{1\times d} & 1 \end{bmatrix} = \mathbf{T}_{cb}. \tag{2.7}$$

2.2 Probability and Statistics

When modelling noisy sensor measurements of translation vectors, we will assume that noise is normally distributed and denote these random variables as follows:

$$\tilde{\mathbf{p}} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right), \tag{2.8}$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean and $\boldsymbol{\Sigma} \in \mathbb{S}_{++}^d$ is the covariance.

Noisy three-dimensional rotation measurements will be modelled with the *isotropic Langevin distribution* (Rosen et al., 2019):

$$\tilde{\mathbf{R}} \sim \mathrm{Lang}\left(\mathbf{R}_\mu, \kappa\right), \tag{2.9}$$

where $\mathbf{R}_\mu \in \mathrm{SO}(3)$ is the *mode* and $\kappa \geq 0$ is the *concentration* parameter. This distribution's probability density function (PDF) is

$$p(\mathbf{R}; \mathbf{R}_\mu, \kappa) = \frac{1}{c(\kappa)} \exp\left(\kappa \;\mathrm{tr}\left(\mathbf{R}_\mu^\top \mathbf{R}\right)\right), \tag{2.10}$$

where $c(\kappa)$ is a normalization constant. We use the Langevin distribution for two key reasons: first, it is defined directly over the entirety of SO(3) and therefore appropriate for global optimization without an accurate initialization.[1] Second, the exponential form of its PDF in Equation (2.10) allows us to construct quadratic maximum likelihood cost functions for extrinsic calibration problems in Chapter 4. This quadratic form is essential for the convex relaxations we will develop in Chapter 3.

2.3 Linear Algebra

This section contains some useful linear algebra results that will help us prove key theorems.

[1] This is in contrast to normal distributions defined over the Lie algebra $\mathfrak{so}(3)$ used for local optimization in Barfoot (2017).

2.3.1 Rank of Positive Semidefinite (PSD) Matrices

In Section 3.5, we make use of the fact that the sum of the ranks of two particular matrices is bounded. This fact is used without proof in Cifuentes et al. (2022), but we could not find a concise reference containing a proof and therefore present one here.

Lemma 1 (Rank of PSD Matrices Connected by a Trace Identity)**.** *Let* $\mathbf{A}, \mathbf{B} \in \mathbb{S}^n_+$ *such that* $\operatorname{tr}(\mathbf{AB}) = 0$*. Then,* $\operatorname{rank}(\mathbf{A}) + \operatorname{rank}(\mathbf{B}) \leq n$*.*

Proof. Let $a = \operatorname{rank}(\mathbf{A})$ and $b = \operatorname{rank}(\mathbf{B})$. Since both matrices are PSD, we can write their rank-1 decompositions

$$\begin{aligned} \mathbf{A} &= \sum_{i=1}^{a} \mathbf{a}_i \, \mathbf{a}_i^\top \\ \mathbf{B} &= \sum_{i=1}^{b} \mathbf{b}_i \, \mathbf{b}_i^\top \end{aligned} \tag{2.11}$$

such that $\{\mathbf{a}_i\}_{i=1}^a$ and $\{\mathbf{b}_i\}_{i=1}^b$ are each linearly independent. We can then use the linear and cyclic[2] properties of the trace function to write

$$\begin{aligned} \operatorname{tr}(\mathbf{AB}) &= \operatorname{tr}\left(\left(\sum_{i=1}^{a} \mathbf{a}_i \, \mathbf{a}_i^\top\right)\left(\sum_{i=1}^{b} \mathbf{b}_i \, \mathbf{b}_i^\top\right)\right) \\ &= \sum_{i=1}^{a}\sum_{j=1}^{b} \operatorname{tr}\left((\mathbf{a}_i \, \mathbf{a}_i^\top)(\mathbf{b}_j \, \mathbf{b}_j^\top)\right) \\ &= \sum_{i=1}^{a}\sum_{j=1}^{b} \operatorname{tr}\left(\mathbf{a}_i^\top \, \mathbf{b}_j \, \mathbf{b}_j^\top \, \mathbf{a}_i\right) \\ &= \sum_{i=1}^{a}\sum_{j=1}^{b} (\mathbf{a}_i^\top \, \mathbf{b}_j)^2. \end{aligned} \tag{2.12}$$

Since the trace of each summand is nonnegative and $\operatorname{tr}(\mathbf{AB}) = 0$, we conclude that each summand is equal to zero and $\mathbf{a}_i^\top \mathbf{b}_j = 0$ for all $i \in [a]$ and all $j \in [b]$. This implies that $S = \{\mathbf{a}_i\}_{i=1}^a \cup \{\mathbf{b}_i\}_{i=1}^b$ is linearly independent. However, the Fundamental Theorem of Linear Algebra (Strang, 1993) states that the dimensions of a matrix's row and column spaces are equal. Applying this fact to the matrix with columns formed by the linearly independent elements of S yields $a + b \leq n$ as desired.

□

2.4 Real Analysis

An *open cover* of a set $\mathcal{A}$ is a collection F of open sets whose union contains $\mathcal{A}$. A set $\mathcal{A}$ is *compact* if every open cover F admits a finite *subcover* (i.e., there exists a finite subset of

[2] $\operatorname{tr}(\mathbf{AB}) = \operatorname{tr}(\mathbf{BA})$.

F which is also covers $\mathcal{A}$). In this section, we present a short non-constructive proof of the existence of an arbitrarily precise *union of balls* representation for compact obstacles used in Chapter 6.[3]

Theorem 1 (Heine-Borel cover Theorem (Apostol, 1974))**.** *Let F be an open cover of a closed and bounded set $\mathcal{A} \subset \mathbb{R}^n$. Then a finite subcollection of F also covers $\mathcal{A}$.*

Theorem 1 essentially states that the compact subsets of $\mathbb{R}^n$ are in fact the closed and bounded subsets of $\mathbb{R}^n$. As Corollary 1 demonstrates, this is convenient for representing objects in $\mathbb{R}^n$ up to arbitrary precision with a finite number of elements.

Corollary 1. *Let $\mathcal{O}$ be a closed and bounded subset of $\mathbb{R}^n$. Then for any $\epsilon > 0$, there exists a finite union of open balls with radius ϵ that covers $\mathcal{O}$.*

Proof. Let

$$F \triangleq \{B(\mathbf{x}, \epsilon) \mid \mathbf{x} \in \mathcal{O}\}, \tag{2.13}$$

where $B(\mathbf{x}, \epsilon)$ is an open ball centred at $\mathbf{x}$ with radius ϵ. Since F is clearly an open cover of $\mathcal{O}$ and $\mathcal{O}$ is closed and bounded, by Theorem 1 we have a finite subcover $F' \subset F$ which satisfies our requirement. □

[3] While trivial, we were unable to find a simple proof of this fact in the robotics literature.

Chapter 3

Semidefinite Relaxations

You're simply the best,
Better than all the rest.

TINA TURNER, *Simply the Best*

This chapter begins with a review of elementary material on the theory of optimization, specifically for quadratically constrained quadratic programs (QCQPs). In addition to their broad relevance to problems outside of robotics, these tools are essential for the applications in subsequent chapters. Our focus is on convex semidefinite programming (SDP) relaxations for QCQPs, and we conclude with a novel extension of the theory of parametric stability developed in Cifuentes et al. (2022) to SDP relaxations of QCQPs with inequality constraints. We make heavy use of the treatment of convex optimization in Boyd and Vandenberghe (2004), but modify and mix its notation and conventions with those found in Cifuentes et al. (2022) to better suit our purposes. Readers interested in a rich geometric treatment of semidefinite programming are advised to consult Dattorro (2005).

3.1 Quadratically Constrained Quadratic Programs

Our main objects of study are quadratically constrained quadratic programs. QCQPs are optimization problems with quadratic cost and constraints:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & \mathbf{x}^\top \mathbf{C}\mathbf{x} + \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}^\top \mathbf{A}_i \mathbf{x} + \mathbf{a}^\top \mathbf{x} \leq a_i,\ i = 1, \dots, l \\ & \mathbf{x}^\top \mathbf{B}_i \mathbf{x} + \mathbf{b}^\top \mathbf{x} = b_i,\ i = 1, \dots, m, \end{aligned} \tag{3.1}$$

where $\mathbf{C}, \mathbf{B}_i, \mathbf{A}_i \in \mathbb{S}^n$. We denote the optimal value of the objective function by p^* and the optimal decision variable by $\mathbf{x}^*$.

Note that Equation (3.1) is nonconvex in general, and can model NP-hard problems. For example, integer constraints like $x_i \in \{-1, 1\}$ can be expressed quadratically as $x_i^2 = 1$, allowing

Figure 3.1: Two sets illustrating the definition of convexity in $\mathbb{R}^2$. The "bite" taken out of the set on the right renders it nonconvex, as the line joining two of its elements is no longer completely contained in the set.

us to formulate MAX CUT[1] as a QCQP (Goemans and Williamson, 1995). Thus, we should not expect any solution method to act as a "silver bullet" for QCQPs in general, and instead focus on properties of QCQPs that arise as a consequence of the specific structure of geometric problems in robotics.

3.1.1 Homogeneous QCQPs

We can simplify a great deal of our analysis by considering only *homogeneous* QCQPs (i.e., those that do not contain any linear terms in the cost or constraints):

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & \mathbf{x}^\top \mathbf{C}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}^\top \mathbf{A}_i \mathbf{x} \leq a_i,\ i = 1,\ldots,l \\ & \mathbf{x}^\top \mathbf{B}_i \mathbf{x} = b_i,\ i = 1,\ldots,m. \end{aligned} \tag{3.2}$$

Thankfully, restricting our attention to homogenous QCQPs does not limit the variety of QCQPs we are able to study: any linear terms can be made quadratic with the addition of a homogenizing variable $s^2 = 1$ to our decision variable $\mathbf{x}$. With s, linear constraints like right-handedness for SO(3) in Equation (2.3), for example, can be made purely quadratic as follows:

$$\mathbf{R}^{(i)} \times \mathbf{R}^{(j)} = s\mathbf{R}^{(k)}. \tag{3.3}$$

This procedure introduces a spurious solution $\mathbf{x}' = -\mathbf{x}^*$, but selecting the true solution is trivial in practice and does not impede our theoretical analysis.

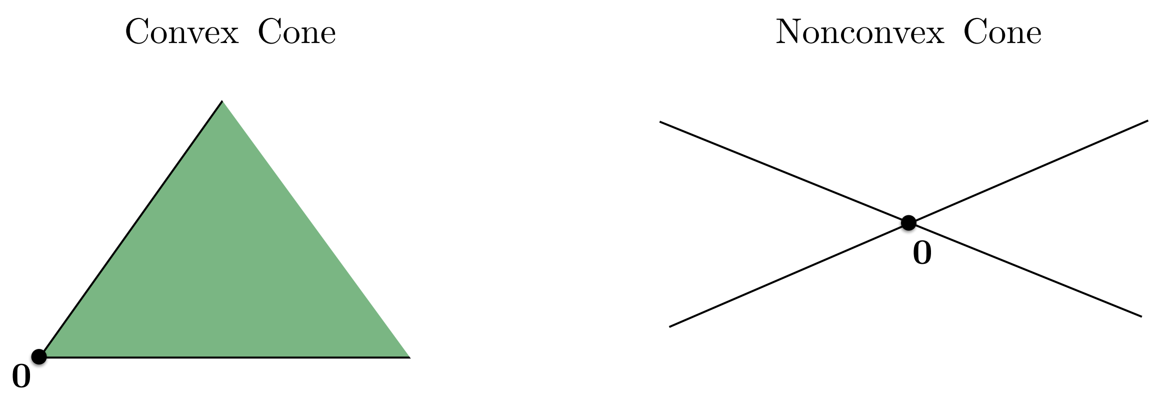


Figure 3.2: Two cones in $\mathbb{R}^2$ with black lines indicating their boundaries. The convex cone on the left has a green interior and is clearly convex by inspection. The nonconvex cone is simply the union of two non-parallel lines through the origin. Note that both cones have infinite extent.

3.2 Convex Optimization

In order to understand convex optimization problems and their unique properties, we must introduce some basic convex analysis and geometry. For a detailed theoretical treatment of convex analysis and the remarkable properties of convex sets, see Barvinok (2002).

3.2.1 Convex Sets

A set $S \subset \mathbb{R}^n$ is convex if the line segment connecting any two points in S is also contained in S, i.e.,

$$S \text{ is convex } \iff \forall\, \mathbf{x}_1, \mathbf{x}_2 \in S: \;\; \alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2 \in S \;\; \forall \alpha \in [0,1]. \tag{3.4}$$

Figure 3.1 illustrates an example of a convex and a nonconvex set in $\mathbb{R}^2$.

One important type of convex set is a *convex cone*. A set C is a *cone* if for every element, any dilation of that element is also in C:

$$C \text{ is a cone } \iff \forall\, \mathbf{x} \in C: \;\; \alpha \mathbf{x} \in C \;\; \forall \alpha \geq 0. \tag{3.5}$$

A convex cone is, thankfully, a cone which is also convex. Figure 3.2 contains some examples of convex and nonconvex cones. We are most interested in a particular convex cone: the semidefinite cone $\mathbb{S}^n_+$, which we discuss in detail in Section 3.4.

[1]Finding a "maximum cut" (or MAX CUT) of a graph is an important NP-complete problem. A *cut* is defined as a partition of a graph's vertices into two complementary sets, and its size is the number of edges between these two sets. Therefore, a maximum cut is a cut with the largest size possible.

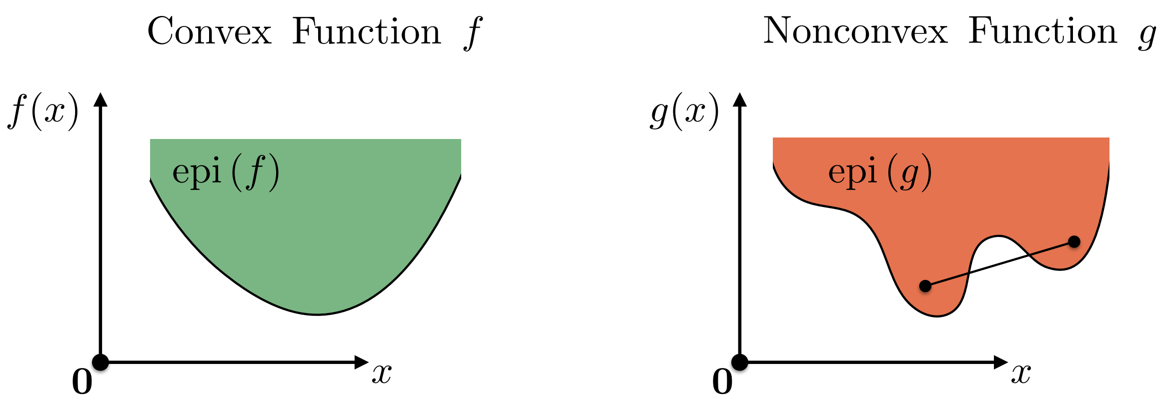


Figure 3.3: The epigraphs of a convex function f and a nonconvex function g. In contrast to the convex function f on the left, the nonconvex function g on the right has a nonconvex epigraph. Note that the epigraphs are truncated to fit on a finite page: they both extend infinitely in the y-direction.

3.2.2 Convex Functions

Building on the concept of convexity for sets, a function $f : S \to \mathbb{R}$ with domain $\mathrm{dom}\,(f) = S \subset \mathbb{R}^n$ is convex if for any $\alpha \in [0, 1]$:

$$f(\alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2) \leq \alpha f(\mathbf{x}_1) + (1-\alpha) f(\mathbf{x}_2) \ \ \forall\ \mathbf{x}_1, \mathbf{x}_2 \in S. \tag{3.6}$$

For a more geometrically satisfying definition, we introduce the *epigraph* of f:

$$\mathrm{epi}\,(f) = \left\{ \begin{bmatrix}\mathbf{x}^\top & t\end{bmatrix}^\top \,\middle|\, \mathbf{x} \in \mathrm{dom}\,(f),\ f(\mathbf{x}) \leq t \right\} \subseteq \mathbb{R}^{n+1}. \tag{3.7}$$

Intuitively, the epigraph is the hypervolume "above" the hypersurface described by f in $\mathbb{R}^{n+1}$. We can now elegantly define a convex function as one whose epigraph is a convex set: Figure 3.3 demonstrates this for simple univariate functions. Lastly, a function is *strictly* convex if the inequality in Equation (3.6) holds strictly, and a function f is (strictly) *concave* if $-f$ is (strictly) convex.[2]

3.2.3 Convex Optimization Problems

With convex sets and functions in hand, we can now define a convex optimization problem (or convex *program*) as a minimization of a convex objective function over a convex feasible set. For the QCQPs defined in Section 3.1, convexity is equivalent to all three of the following conditions being met:

1. a semidefinite cost matrix ($\mathbf{C} \succeq \mathbf{0}$);
2. affine equality constraints ($\mathbf{B} = \mathbf{0}\ \ \forall i \in [m]$); and

[2] Maximization of concave functions over convex sets is therefore reducible to convex optimization, removing the need for further elaboration on concave functions.

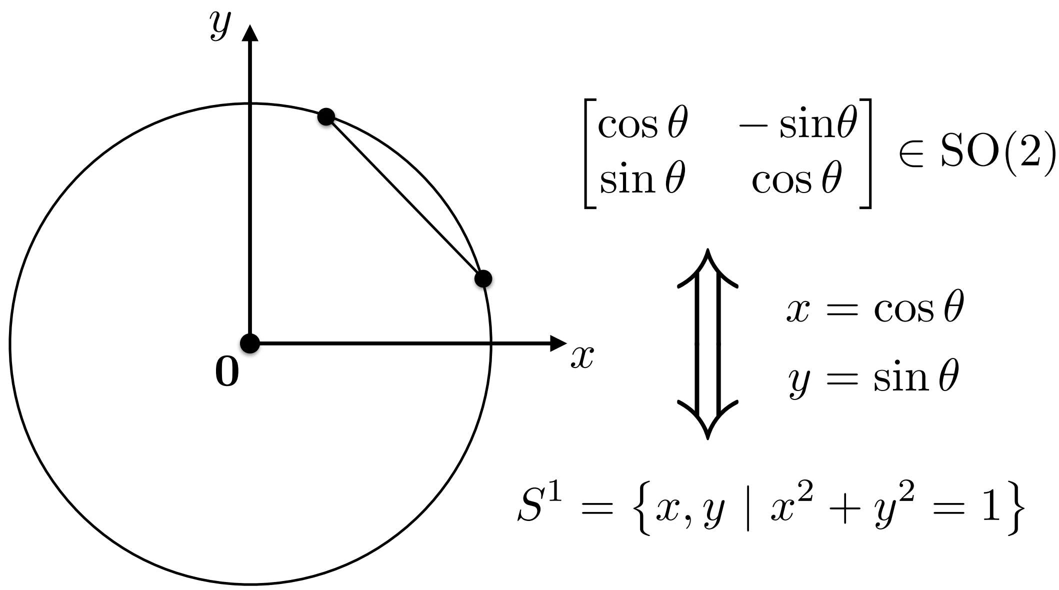


Figure 3.4: The special orthogonal Lie group $\mathrm{SO}(d)$ is defined by quadratic equality constraints and is therefore nonconvex. This is easiest to visualize for $\mathrm{SO}(2)$ because it is isomorphic to the unit circle S^1.

3. semidefinite inequality constraint matrices ($\mathbf{A}_i \succeq \mathbf{0} \ \ \forall i \in [l]$).

The QCQPs studied in this dissertation are all nonconvex because of the nonconvex constraints required to describe the feasible sets for $\mathrm{SO}(d)$ and distance geometry problems. Figure 3.4 visually demonstrates that $\mathrm{SO}(2)$ is nonconvex via its isomorphism to the unit circle S^1.

When discussing general optimization problems, we will use the following notation:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \ i = 1, \ldots, l \\ & h_i(\mathbf{x}) = 0, \ i = 1, \ldots, m. \end{aligned} \tag{3.8}$$

This problem is convex when f is convex, equality constraints h_i are affine, and inequality constraints g_i are convex. For reasons that will be made clear in Section 3.3, we will refer to Equation (3.8) as the *primal* problem. This notation is convenient for developing duality theory in Section 3.3, but we replace it with specialized forms for QCQPs and their semidefinite relaxations in Section 3.4 and subsequent chapters.

Convex optimization problems are particularly exciting because every local minimum is also a global minimum (Boyd and Vandenberghe, 2004, Sec. 4.2.2). This means that efficient *local* optimization methods, such as gradient descent and its ilk, converge to *global* minima.[3]

[3]This wonderful property led R.T. Rockafellar to sagely point out that "...the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity" in Rockafellar (1993). This will serve as a major theme of this dissertation, which reveals "hidden" convexity in seemingly difficult nonlinear and nonconvex problems.

In light of this fact, in subsequent chapters we primarily treat convex solvers as black boxes, abstracting away their details and assuming that they return a global optimum in polynomial time. Interested readers can consult Boyd and Vandenberghe (2004) for details on interior-point solvers and other methods.

3.3 Lagrangian Duality

Every optimization problem, convex or nonconvex, is intimately related to a convex problem that arises from its *Lagrangian*:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f(\mathbf{x}) + \sum_{i=1}^{l} \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^{m} \nu_i h_i(\mathbf{x}), \tag{3.9}$$

with function definitions from Equation (3.8). The variables $\boldsymbol{\lambda} \in \mathbb{R}^l$ and $\boldsymbol{\nu} \in \mathbb{R}^m$ are called *Lagrange multipliers* or *dual variables.* The Lagrangian is a powerful tool for solving nonlinear programs. One way to conceptualize dual variables is as "penalties" for breaking the constraints of our optimization problem. Minimizing the Lagrangian for a given set of multiplier values defines the *dual function* $\mathsf{d} : \mathbb{R}^l \times \mathbb{R}^m \to \mathbb{R}$:

$$\mathsf{d}(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathcal{D}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathcal{D}} \left(f(\mathbf{x}) + \sum_{i=1}^{l} \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^{m} \nu_i h_i(\mathbf{x}) \right), \tag{3.10}$$

where $\mathcal{D}$ is the intersection of the domains of the cost and constraint functions. The dual function is concave, even when the original optimization problem Equation (3.8) is not convex. This concavity allows us to define a convex optimization problem called the *dual problem*:

$$\max_{\boldsymbol{\lambda} \geq \mathbf{0}_l,\ \nu \in \mathbb{R}^m} \mathsf{d}(\boldsymbol{\lambda}, \boldsymbol{\nu}). \tag{3.11}$$

Critically, the optimal solution d^* of Equation (3.11) never overestimates the solution to the primal problem:

$$\mathsf{d}^* \leq p^*. \tag{3.12}$$

The inequality in Equation (3.12) is known as *weak duality*, and it is an important result because it reveals that the convex dual problem gives us a way to efficiently[4] compute lower bounds for the primal problem.

[4] We come to this conclusion by recalling that convex optimization problems yield global solutions with local search methods, and that weak duality holds even for nonconvex primal problems.

3.3.1 Strong Duality

We are particularly interested in cases where the lower bound provided by Equation (3.11) is *tight*, that is,

$$\mathsf{d}^* = p^*. \tag{3.13}$$

When this is the case, we say that *strong duality* holds. One important class of problems for which strong duality holds is convex optimization problems that satisfy certain *constraint qualifications* (Boyd and Vandenberghe, 2004). *Slater's condition* is one example of a simple constraint qualification that is relevant to convex optimization problems. It states that there exists an $\mathbf{x}$ which is in the relative interior[5] of the domain of Equation (3.8) such that

$$\begin{aligned} g_i(\mathbf{x}) &< 0, \ \ i = 1, \dots, l \\ h_i(\mathbf{x}) &= 0, \ \ i = 1, \dots, m. \end{aligned} \tag{3.14}$$

In other words, $\mathbf{x}$ is "strictly feasible" for Equation (3.8). We will discuss other constraint qualifications in greater detail in Section 3.5.2, as they are important tools for guaranteeing certain properties (like strong duality) hold for optimization problems.

Slater's condition can be used to demonstrate that convex optimization problems exhibit strong duality, but we are more interested in *nonconvex* problems for which strong duality holds. Essentially, strong duality allows us to extract the global optimizer $\mathbf{x}^*$ for the nonconvex primal from the global optimizer $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ of the convex, and therefore easier to solve, dual problem.

3.3.2 Optimality Conditions

In this section, we discuss the Karush-Kuhn-Tucker (KKT) necessary conditions for local minima of (possibly nonconvex) optimization problems. Let $\mathbf{x}^*$ be a local minimum with for some general nonlinear program of the form in Equation (3.8) with continuously differentiable cost and constraint functions f, g_i, and h_i. Then, if $\mathbf{x}^*$ is *regular* (i.e., some constraint qualification[6] holds), their exist Lagrange multipliers $\boldsymbol{\lambda}^* \in \mathbb{R}^l$ and $\boldsymbol{\nu}^* \in \mathbb{R}^m$ such that (Bertsekas, 1999)

$$\begin{aligned} \nabla_{\mathbf{x}} \mathcal{L}\left(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*\right) = 0 & \\ g_i(\mathbf{x}^*) \leq 0, & \ \ i = 1, \dots, l \\ h_i(\mathbf{x}^*) = 0, & \ \ i = 1, \dots, m \\ \lambda_i^* \geq 0, & \ \ i = 1, \dots, l \\ \lambda_i^* = 0, & \ \ \forall i \notin A\left(\mathbf{x}^*\right), \end{aligned} \tag{KKT}$$

[5]As opposed to the interior, the *relative interior* of a set is the interior of its affine hull. This distinction is only salient for problems where the feasible set has an affine dimension that is less than the dimension n of the decision variable $\mathbf{x} \in \mathbb{R}^n$.

[6]Depending on the particular constraint qualification used, the statement of the KKT conditions can be strengthened to the existence of *unique* Lagrange multipliers. For example, the strong linear independence constraint qualification (LICQ) of Section 3.5.2 provides this uniqueness guarantee.

where $A(\mathbf{x}) \subseteq [l]$ is the index set of the *active* inequality constraints at $\mathbf{x}$, i.e.,

$$A(\mathbf{x}) \triangleq \{i \in [l] \mid g_i(\mathbf{x}) = 0\}. \tag{3.15}$$

The first line of Equation (KKT) is often called *stationarity*, and it states that any primal/dual pair of optimal values $\mathbf{x}^*, (\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ is a critical point of the Lagrangian. The second and third lines are simply the primal feasibility of $\mathbf{x}^*$, and the fourth is the nonnegativity of the Lagrange multipliers $\boldsymbol{\lambda}^*$ which penalize inequality constraints in the Lagrangian. The final line of Equation (KKT) is called *complementary slackness*, and it can be alternatively stated, alongside the nonnegativity of each λ_i^*, as

$$\lambda_i^* g(\mathbf{x}^*) = 0, \quad i = 1, \ldots, l. \tag{3.16}$$

The KKT conditions are an essential tool for connecting primal and dual solutions, and some of our results rely on the existence of Lagrange multipliers. The example illustrated in Figure 3.5 demonstrates a case where Lagrange multipliers do not exist for the global optimum $\mathbf{x}^*$.

3.4 Semidefinite Programming Relaxations

In this section, we will introduce two semidefinite programs (SDPs) which are convex relaxations of the homogeneous QCQP in Equation (3.2). These relaxations lead to optimizations over feasible sets that involve semidefinite matrix constraints. In subsequent chapters, we include some visualizations of toy SDPs which provide geometric intuition about problems of interest and the SDP cone in general. We strongly recommend Dattorro (2005) to readers interested in geometric interpretations of semidefinite optimization.

3.4.1 The Positive Semidefinite Cone

We begin by demonstrating that the set of $n \times n$ symmetric positive semidefinite (PSD) matrices $\mathbb{S}_+^n$ is a convex cone. First, we prove that $\mathbb{S}_+^n$ is a cone:

$$\begin{aligned} \mathbf{A} \in \mathbb{S}_+^n \implies & \mathbf{x}^\top \mathbf{A}\mathbf{x} \geq 0 \;\; \forall \mathbf{x} \in \mathbb{R}^n \\ \implies & \alpha(\mathbf{x}^\top \mathbf{A}\mathbf{x}) \geq 0 \;\; \forall \mathbf{x} \in \mathbb{R}^n, \;\; \forall \alpha \geq 0 \\ \implies & \alpha \mathbf{A} \in \mathbb{S}_+^n \;\; \forall \alpha \geq 0. \end{aligned} \tag{3.17}$$

Next, we prove that it is convex. Recalling the definition of a convex set in Equation (3.4), let $\mathbf{C} = \alpha\mathbf{A} + (1-\alpha)\mathbf{B}$ for any $\alpha \in [0, 1]$ and arbitrary $\mathbf{A}, \mathbf{B} \in \mathbb{S}_+^n$. Using the definition of $\mathbb{S}_+^n$, for any $\mathbf{x} \in \mathbb{R}^n$ we have:

$$\mathbf{x}^\top \mathbf{C}\mathbf{x} = \alpha\, \mathbf{x}^\top \mathbf{A}\mathbf{x} + (1-\alpha)\, \mathbf{x}^\top \mathbf{B}\mathbf{x} \geq 0, \tag{3.18}$$

i.e., $\mathbf{C} \in \mathbb{S}_+^n$ and $\mathbb{S}_+^n$ is convex. In the next section, we show that the semidefinite cone arises naturally as a constraint on the feasible set of the Lagrangian dual for QCQPs.

3.4.2 Semidefinite Programs

A *semidefinite program* is an optimization problem with a linear cost, linear constraints, and one or more constraints involving the semidefinite cone. We mainly consider SDPs in the following standard form:

$$\begin{aligned} \min_{\mathbf{Z}\in\mathbb{S}^n_+} \quad & \operatorname{tr}(\mathbf{C}\mathbf{Z}) \\ \text{s.t.} \quad & \operatorname{tr}(\mathbf{A}_i\mathbf{Z}) \leq a_i,\ i=1,\ldots,l \\ & \operatorname{tr}(\mathbf{B}_i\mathbf{Z}) = b_i,\ i=1,\ldots,m, \end{aligned} \tag{3.19}$$

where the cost matrix $\mathbf{C}$ and constraint matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ are all elements of $\mathbb{S}^n$. Since $\mathbb{S}^n_+$ is a convex cone and the trace function is linear, it is easy to see that Equation (3.19) is a convex optimization problem. For notational convenience, we will often write Equation (3.19)'s linear constraints as

$$\begin{aligned} \mathcal{A}(\mathbf{Z}) &\leq \mathbf{a} \\ \mathcal{B}(\mathbf{Z}) &= \mathbf{b}, \end{aligned} \tag{3.20}$$

where $\mathcal{A}:\mathbb{S}^n\to\mathbb{R}^l$ and $\mathcal{B}:\mathbb{S}^n\to\mathbb{R}^m$ are linear operators.

3.4.3 The Lagrangian Dual of a QCQP

Let us apply the procedure in Section 3.3 to derive the Lagrangian dual problem for the homogeneous QCQP in Equation (3.2). We begin by forming the Lagrangian:

$$\mathcal{L}(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\nu}) = \mathbf{x}^\top\mathbf{C}\mathbf{x} + \sum_{i=1}^{l}\lambda_i(\mathbf{x}^\top\mathbf{A}_i\mathbf{x} - a_i) + \sum_{i=1}^{m}\nu_i(\mathbf{x}^\top\mathbf{B}_i\mathbf{x} - b_i). \tag{3.21}$$

Using Equation (3.11), we can compute the dual function:

$$\begin{aligned} \mathsf{d}(\boldsymbol{\lambda},\boldsymbol{\nu}) &= \inf_{\mathbf{x}\in\mathbb{R}^n}\mathcal{L}(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\nu}) \\ &= \inf_{\mathbf{x}\in\mathbb{R}^n}\left(\mathbf{x}^\top\mathbf{C}\mathbf{x} + \sum_{i=1}^{l}\lambda_i(\mathbf{x}^\top\mathbf{A}_i\mathbf{x} - a_i) + \sum_{i=1}^{m}\nu_i(\mathbf{x}^\top\mathbf{B}_i\mathbf{x} - b_i)\right) \\ &= \inf_{\mathbf{x}\in\mathbb{R}^n}\left(\mathbf{x}^\top(\mathbf{C} + \sum_{i=1}^{l}\lambda_i\mathbf{A}_i + \sum_{i=1}^{m}\nu_i\mathbf{B}_i)\mathbf{x} - \mathbf{a}^\top\boldsymbol{\lambda} - \mathbf{b}^\top\boldsymbol{\nu}\right) \\ &= \begin{cases} -\mathbf{a}^\top\boldsymbol{\lambda} - \mathbf{b}^\top\boldsymbol{\nu}, & \mathbf{C} + \sum_{i=1}^{l}\lambda_i\mathbf{A}_i + \sum_{i=1}^{m}\nu_i\mathbf{B}_i \succeq \mathbf{0} \\ -\infty, & \text{otherwise.} \end{cases} \end{aligned} \tag{3.22}$$

The final equality follows from the definition of $\mathbb{S}_+^n$: if the quadratic form is not PSD, then Equation (3.22) is unbounded below.[7] We can now state the dual problem

$$\begin{aligned} \max_{\boldsymbol{\lambda}\geq\mathbf{0},\boldsymbol{\nu}\in\mathbb{R}^m} \quad & -\mathbf{a}^\top\boldsymbol{\lambda}-\mathbf{b}^\top\boldsymbol{\nu} \\ \text{s.t.} \quad & \mathbf{C}+\sum_{i=1}^{l}\lambda_i\mathbf{A}_i+\sum_{i=1}^{m}\nu_i\mathbf{B}_i \succeq \mathbf{0}, \end{aligned} \tag{3.23}$$

which we recognize as an SDP, but not in the standard form of Equation (3.19).

3.4.4 The Primal SDP Relaxation

In this section, we explore an alternative SDP relaxation of QCQPs that ends up being closely related to the dual SDP relaxation in Equation (3.23). We begin by reformulating Equation (3.2) with the addition of auxiliary variable $\mathbf{Z}$:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & \operatorname{tr}(\mathbf{C}\mathbf{Z}) \\ \text{s.t.} \quad & \operatorname{tr}(\mathbf{A}_i\mathbf{Z})\leq a_i,\ i=1,\dots,l \\ & \operatorname{tr}(\mathbf{B}_i\mathbf{Z})=b_i,\ i=1,\dots,m \\ & \mathbf{Z}=\mathbf{x}\,\mathbf{x}^\top, \end{aligned} \tag{3.24}$$

where all the nonlinearity (and therefore nonconvexity) has been "packed" into the quadratic definition of $\mathbf{Z}$. Equation (3.24) can be simplified further to remove $\mathbf{x}$ completely:

$$\begin{aligned} \min_{\mathbf{Z}\in\mathbb{S}_+^n} \quad & \operatorname{tr}(\mathbf{C}\mathbf{Z}) \\ \text{s.t.} \quad & \operatorname{tr}(\mathbf{A}_i\mathbf{Z})\leq a_i,\ i=1,\dots,l \\ & \operatorname{tr}(\mathbf{B}_i\mathbf{Z})=b_i,\ i=1,\dots,m \\ & \operatorname{rank}(\mathbf{Z})=1. \end{aligned} \tag{3.25}$$

We now *relax* the rank-1 constraint to derive the convex *primal* SDP relaxation[8] of Equation (3.2):

$$\begin{aligned} \min_{\mathbf{Z}\in\mathbb{S}_+^n} \quad & \operatorname{tr}(\mathbf{C}\mathbf{Z}) \\ \text{s.t.} \quad & \mathcal{A}(\mathbf{Z})\leq\mathbf{a} \\ & \mathcal{B}(\mathbf{Z})\leq\mathbf{b}, \end{aligned} \tag{3.26}$$

[7] Please refer to Boyd and Vandenberghe (2004) for further details on deriving the dual of various problems, and for the useful concepts of generalized inequalities and dual cones.

[8] This is sometimes also referred to as the *Shor relaxation* (Cifuentes et al., 2022).

which is identical to the standard form of Equation (3.19) when using the compact shorthand for constraints defined in Equation (3.20). Since this SDP is a "relaxation" of constraints,[9] we know that its optimal value underestimates the optimal value p^* of the primal QCQP. We are interested in cases where this relaxation is tight: that is, the optimal value of Equation (3.26) is equal to p^*. When this is the case and $\operatorname{rank}(\mathbf{Z}^*) = 1$, we can extract $\mathbf{x}^*$ from $\mathbf{Z}^*$.

Finally, we note that the Lagrangian dual of Equation (3.23) is in fact Equation (3.26). Therefore, since SDPs are convex, when Slater's condition (or some other constraint qualification) holds, we have strong duality between Equation (3.23) and Equation (3.26). Furthermore, when one of these SDPs is tight to the primal QCQP, we can conclude that the remaining SDP is also a tight relaxation. In subsequent chapters, we make use of the rich theory of Lagrangian duality and study Equation (3.23) closely, but we will mostly *solve* the primal QCQP relaxation in Equation (3.26).[10]

3.5 Parametric Stability

In this section, we summarize the key results from Cifuentes et al. (2022) that we will extend and apply to geometric problems in robotics. These results are reproduced in a notation consistent with the rest of this dissertation, and therefore may appear slightly different than their original form in Cifuentes et al. (2022).

The central object of study in this section is the following family of homogeneous QCQPs with identical constraints and cost functions that vary with respect to some parameter:

Problem 1 (Parameteric QCQP)**.** *Let $\Xi \subseteq \mathbb{R}^d$ be a space parameterizing our problem. Find the solution $\mathbf{x}^*$ of the following homogeneous QCQP whose cost function varies continuously with $\boldsymbol{\xi} \in \Xi$:*

$$\begin{aligned}
\min_{\mathbf{x}\in\mathbb{R}^n} \quad & c_{\boldsymbol{\xi}}(\mathbf{x}) \triangleq \mathbf{x}^\top \mathbf{C}(\boldsymbol{\xi})\mathbf{x} \\
\text{s.t.} \quad & g_i(\mathbf{x}) \triangleq \mathbf{x}^\top \mathbf{A}_i\mathbf{x} - a_i \leq 0,\ i = 1,\dots,l \\
& h_i(\mathbf{x}) \triangleq \mathbf{x}^\top \mathbf{B}_i\mathbf{x} - b_i = 0,\ i = 1,\dots,m.
\end{aligned} \tag{Q$_{\boldsymbol{\xi}}$}$$

In order to avoid the trivial solution $\mathbf{x} = 0$, we assume that $\exists\, i \in [m]$ such that $b_i \neq 0$.

Recalling Section 3.4, the QCQP with parametric cost in Problem 1 has the following primal/dual pair of SDP relaxations:

Problem 2 (Parameteric SDP Relaxations)**.** *The primal and dual SDP relaxations of Problem 1 are:*

[9] This can be understood by noting that each element $\mathbf{x}$ of the feasible set of Equation (3.2) corresponds to an element $\mathbf{Z} = \mathbf{x}\,\mathbf{x}^\top$ of the feasible set of Equation (3.26). In other words, the SDP relaxation's feasible set is a superset of the primal problem's feasible set when "lifted" to $\mathbb{S}^n$. Thus, it can do no worse and underestimates the primal's optimal value.

[10] The performance of numerical solvers varies greatly with problem details and parameter settings, but we found that the primal SDP is faster to solve for the specific problems addressed in this dissertation.

$$\begin{aligned} \min_{\mathbf{Z}\in\mathbb{S}^n_+} \quad & \operatorname{tr}\left(\mathbf{C}(\boldsymbol{\xi})\mathbf{Z}\right) \\ \text{s.t.} \quad & \mathcal{A}\left(\mathbf{Z}\right) \leq \mathbf{a} \\ & \mathcal{B}\left(\mathbf{Z}\right) = \mathbf{b}, \end{aligned} \tag{$\mathrm{P}_{\boldsymbol{\xi}}$}$$

$$\begin{aligned} \max_{\boldsymbol{\lambda}\geq\mathbf{0},\boldsymbol{\nu}\in\mathbb{R}^m} \quad & -\mathbf{a}^\top\boldsymbol{\lambda} - \mathbf{b}^\top\boldsymbol{\nu} \\ \text{s.t.} \quad & \boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda},\boldsymbol{\nu}) \succeq \mathbf{0}, \end{aligned} \tag{$\mathrm{D}_{\boldsymbol{\xi}}$}$$

where $\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda},\boldsymbol{\nu}) \triangleq \mathbf{C}(\boldsymbol{\xi}) + \sum_{i=1}^{l}\lambda_i\mathbf{A}_i + \sum_{i=1}^{m}\nu_i\mathbf{B}_i$ *is half the Hessian of the Lagrangian of* $\mathrm{Q}_{\boldsymbol{\xi}}$.

In general, the optimal values of the three programs in Problem 1 and Problem 2 are related via weak duality (Cifuentes et al., 2022):

$$\operatorname{val}\left(\mathrm{D}_{\boldsymbol{\xi}}\right) \leq \operatorname{val}\left(\mathrm{P}_{\boldsymbol{\xi}}\right) \leq \operatorname{val}\left(\mathrm{Q}_{\boldsymbol{\xi}}\right), \tag{3.27}$$

where $\operatorname{val}(\cdot)$ denotes the value of its argument's cost function at a global optimum. When strong duality holds for $\mathrm{Q}_{\boldsymbol{\xi}}$ and its dual $\mathrm{D}_{\boldsymbol{\xi}}$, the entire inequality chain in Equation (3.27) becomes equalities (i.e., both convex relaxations are *tight*).

3.5.1 Tight or Zero-Duality-Gap Parameters

Consider some nominal parameter value $\bar{\boldsymbol{\xi}} \in \Xi$ for which $\mathrm{P}_{\boldsymbol{\xi}}$ and $\mathrm{D}_{\boldsymbol{\xi}}$ are tight relaxations of $\mathrm{Q}_{\boldsymbol{\xi}}$ (i.e., equality is attained in Equation (3.27)). We refer to any parameter $\boldsymbol{\xi}$ which encodes a QCQP exhibiting strong duality as zero-duality-gap (ZDG). We are interested in determining whether this tightness is *stable* to perturbations of $\bar{\boldsymbol{\xi}}$. To this end, we present a combination of two lemmas found in Cifuentes et al. (2022) and Zheng et al. (2012) respectively, which identifies a useful class of ZDG parameters $\bar{\boldsymbol{\xi}}$.

Lemma 2 (Sufficient Condition for Tightness)**.** *Consider the QCQP* $\mathrm{Q}_{\boldsymbol{\xi}}$ *for any fixed* $\boldsymbol{\xi} \in \Xi$. *Let* $\mathbf{x} \in \mathbb{R}^n$, $\boldsymbol{\lambda} \in \mathbb{R}^l_+$, *and* $\boldsymbol{\nu} \in \mathbb{R}^m$ *be such that*

i) $g_i(\mathbf{x}) \leq 0 \;\forall\, i \in [l]$, $h_i(\mathbf{x}) = 0 \;\forall\, i \in [m]$ *(i.e.,* $\mathbf{x}$ *is feasible for the primal problem* $\mathrm{Q}_{\boldsymbol{\xi}}$*);*

ii) $\forall\, i \in [l],\; \lambda_i > 0 \implies g_i(\mathbf{x}) = 0$ *(i.e., complementary slackness from KKT);*

iii) $\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda},\boldsymbol{\nu}) \succeq \mathbf{0}$, $\boldsymbol{\lambda} \geq \mathbf{0}$ *(i.e., feasibility for the dual problem* $\mathrm{D}_{\boldsymbol{\xi}}$*),*

iv) $\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda},\boldsymbol{\nu})\mathbf{x} = \mathbf{0}$ *(i.e., stationarity from KKT).*

Then, $\mathbf{x}$ *is optimal for* $\mathrm{Q}_{\boldsymbol{\xi}}$, $\boldsymbol{\lambda}$ *and* $\boldsymbol{\nu}$ *are optimal for* $\mathrm{D}_{\boldsymbol{\xi}}$, *and* $\operatorname{val}\left(\mathrm{Q}_{\boldsymbol{\xi}}\right) = \operatorname{val}\left(\mathrm{D}_{\boldsymbol{\xi}}\right)$. *In addition, if* $\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda},\boldsymbol{\nu})$ *is corank-1 (equivalently,* $\operatorname{rank}\left(\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda},\boldsymbol{\nu})\right) = n-1$*), then* $\mathbf{x}\,\mathbf{x}^\top$ *is the unique optimum of* $\mathrm{P}_{\boldsymbol{\xi}}$ *and* $\mathbf{x}$ *is the* unique *optimum of* $\mathrm{Q}_{\boldsymbol{\xi}}$, *up to sign.*[11]

Proof. From stationarity we have that

$$0 = \mathbf{x}^\top\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda},\boldsymbol{\nu})\mathbf{x} = \mathbf{x}^\top\mathbf{C}(\boldsymbol{\xi})\mathbf{x} + \sum_{i=1}^{l}\lambda_i\,\mathbf{x}^\top\mathbf{A}\mathbf{x} + \sum_{i=1}^{m}\nu_i\,\mathbf{x}^\top\mathbf{B}\mathbf{x}, \tag{3.28}$$

[11]This sign ambiguity is due to the assumption from 3.1.1 that we have homogenized any linear constraints with the auxiliary variable $s^2 = 1$.

and primal feasibility of the equality constraints gives us

$$0 = \mathbf{x}^\top \mathbf{C}(\boldsymbol{\xi})\mathbf{x} + \sum_{i=1}^{l} \lambda_i \, \mathbf{x}^\top \mathbf{A}\mathbf{x} + \sum_{i=1}^{m} \nu_i b_i. \tag{3.29}$$

Complementary slackness tells us that if $\lambda_i \neq 0$, then $g_i(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}_i\mathbf{x} - a_i = 0$ and $\lambda_i \, \mathbf{x}^\top \mathbf{A}_i\mathbf{x} = \lambda_i a_i$. Otherwise $\lambda_i = 0$ and $\lambda_i \, \mathbf{x}^\top \mathbf{A}_i\mathbf{x} = 0 = \lambda_i a_i$ trivially. Combining this with Equation (3.29) we have

$$0 = \mathbf{x}^\top \mathbf{C}\mathbf{x} + \sum_{i=1}^{l} \lambda_i a_i + \sum_{i=1}^{m} \nu_i b_i = \mathbf{x}^\top \mathbf{C}\mathbf{x} - (-\,\mathbf{a}^\top \boldsymbol{\lambda} - \mathbf{b}^\top \boldsymbol{\nu}) \geq \operatorname{val}(\mathrm{Q}_{\boldsymbol{\xi}}) - \operatorname{val}(\mathrm{D}_{\boldsymbol{\xi}}), \tag{3.30}$$

where the final inequality follows from the fact that $\mathrm{Q}_{\boldsymbol{\xi}}$ is a minimization and $\mathrm{D}_{\boldsymbol{\xi}}$ is a maximization. Combining Equation (3.30) with Equation (3.12) proves strong duality and that $\mathbf{x}$ and $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ are primal/dual optimal.

Next, suppose $\mathbf{Z}$ is an optimal solution of $\mathrm{P}_{\boldsymbol{\xi}}$. Since $\exists\, i \in [m]$ such that $b_i \neq 0$ by assumption, $\mathbf{Z} \neq \mathbf{0}$. Since strong duality holds between $\mathrm{Q}_{\boldsymbol{\xi}}$ and $\mathrm{D}_{\boldsymbol{\xi}}$, Equation (3.27) tells us that it also holds between $\mathrm{P}_{\boldsymbol{\xi}}$ and $\mathrm{D}_{\boldsymbol{\xi}}$. By complementary slackness for SDPs,[12] $\operatorname{tr}(\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}, \boldsymbol{\nu})\mathbf{Z}) = 0$. Since both $\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}, \boldsymbol{\nu})$ and $\mathbf{Z}$ are PSD, Lemma 1 tells us that $\operatorname{rank}(\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}, \boldsymbol{\nu})) + \operatorname{rank}(\mathbf{Z}) \leq n$. Therefore, if $\operatorname{rank}(\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}, \boldsymbol{\nu})) = n - 1$ (i.e., it is *corank*-1), then

$$\begin{aligned} &0 < \operatorname{rank}(\mathbf{Z}) \leq n - \operatorname{rank}(\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}, \boldsymbol{\nu})) = 1 \\ \Longrightarrow\ &\operatorname{rank}(\mathbf{Z}) = 1. \end{aligned} \tag{3.31}$$

We conclude that $\mathbf{Z}$ is the unique optimal solution of $\mathrm{P}_{\boldsymbol{\xi}}$ since a second distinct optimal solution $\mathbf{Z}'$ would imply the existence of a rank-two optimal solution in the convex hull of $\mathbf{Z}$ and $\mathbf{Z}'$. Finally, since every rank-1 optimal solution of $\mathrm{P}_{\boldsymbol{\xi}}$ corresponds to an optimal solution of $\mathrm{Q}_{\boldsymbol{\xi}}$, we conclude that $\mathbf{x}$ is the *unique* optimal solution of $\mathrm{Q}_{\boldsymbol{\xi}}$ up to sign and $\mathbf{Z} = \mathbf{x}\,\mathbf{x}^\top$. □

Lemma 2 is a statement that alongside dual feasibility, the necessary conditions in KKT are also *sufficient* for strong duality in the specific class of problems $\mathrm{Q}_{\boldsymbol{\xi}}$. We will use it as a tool to prove that problems of the form $\mathrm{Q}_{\boldsymbol{\xi}}$ have ZDG parameters $\bar{\boldsymbol{\xi}}$.

3.5.2 Constraint Qualification

In Section 3.3.1, we noted that Slater's condition is an important constraint qualification for convex optimization problems. In order to prove that interesting and practical *nonconvex* problems of the form in $\mathrm{Q}_{\boldsymbol{\xi}}$ exhibit tightness (equivalently, have ZDG parameters), we need additional theoretical tools in the form of constraint qualifications. Before introducing these,

[12] See Section 5.9.2 of Boyd and Vandenberghe (2004) for details on duality and optimality conditions for cones like $\mathbb{S}_+^n$.

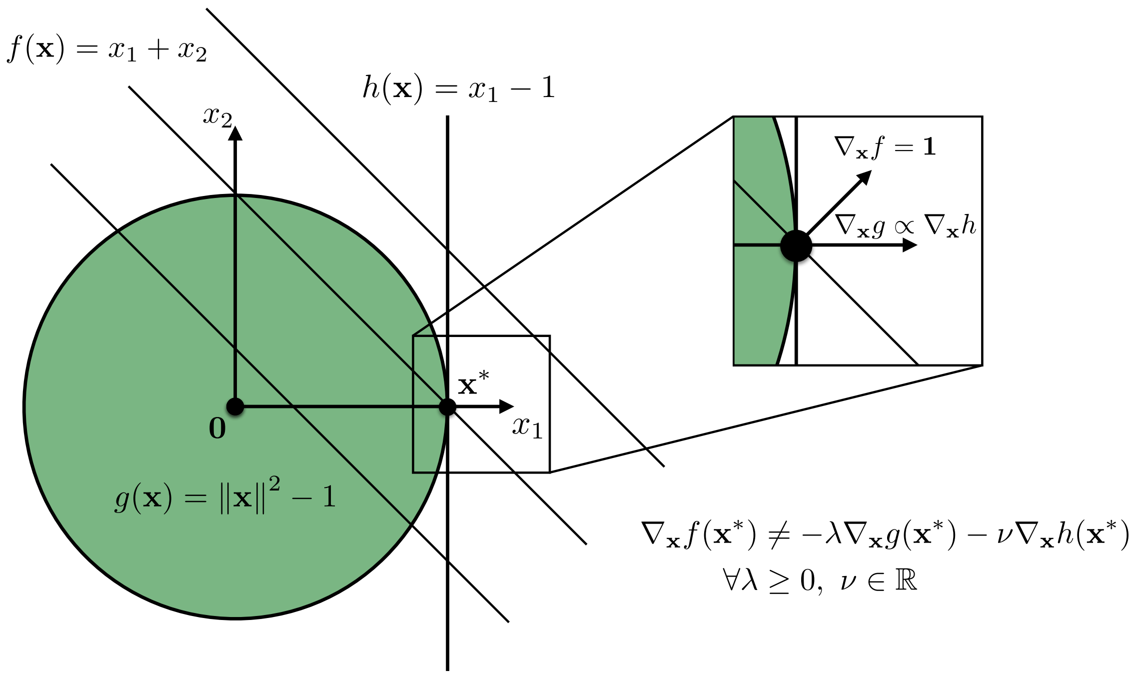


Figure 3.5: An illustration of Problem 4, for which Slater's condition and LICQ both fail to hold. The linear cost function's contours are drawn as diagonal lines. The feasible set is a singleton containing $\mathbf{x}^*$, and the span of the constraint gradients does not contain the cost function gradient. Therefore, Lagrange multipliers satisfying the KKT conditions do not exist.

we investigate a pair of toy problems for which Slater's condition (Equation (3.14)) does not hold.

Problem 3 (Constraint Qualification Example: Nonzero Duality Gap)**.** *Consider the following convex optimization problem (Nedich, 2008):*

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^2} \quad & -x_2 \\ \text{s.t.} \quad & x_2 \geq 0 \\ & \|\mathbf{x}\| \leq x_1. \end{aligned} \tag{CQ-DG}$$

The feasible set of CQ-DG is $\left\{\mathbf{x}\in\mathbb{R}^2 \middle| x_1 \geq 0, x_2 = 0\right\}$*, which does not strictly satisfy the inequality constraints. Therefore, Slater's condition does not hold and strong duality is not guaranteed.*

Problem 3 is trivial to solve by inspection: the entire feasible set has a constant cost function value of 0. In order to compute the dual problem and check if strong duality holds, we begin by writing the Lagrangian of Equation (CQ-DG):

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = -x_2 + \lambda_1(\|\mathbf{x}\| - x_1) - \lambda_2 x_2. \tag{3.32}$$

The dual function is therefore

$$\mathsf{d}(\boldsymbol{\lambda}) = \inf_{\mathbf{x}\in\mathbb{R}^2} \left\{-\lambda_1 x_1 - (1+\lambda_2)x_2 + \lambda_1 \|\mathbf{x}\|\right\}, \tag{3.33}$$

which is unbounded below for all $\lambda_2 \geq 0$. Therefore, the dual problem's value is $\mathsf{d}^* = -\infty \neq 0$ and strong duality is not attained. Let's examine another example of a problem which does not satisfy the requirements of Slater's condition.

Problem 4 (Constraint Qualification Example: No KKT Point)**.** *Consider the convex optimization problem illustrated in Figure 3.5:*

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^2} \quad & x_1 + x_2 \\ \text{s.t.} \quad & x_1 = 1 \\ & \|\mathbf{x}\|^2 \leq 1. \end{aligned} \tag{CQ-KKT}$$

The feasible set of CQ-KKT only contains the optimum $\mathbf{x}^* = [1\ 0]^\top$*, which does not strictly satisfy the inequality constraint. Therefore, Slater's condition does not hold and strong duality is once again not guaranteed.*

While trivial, Problem 4's low-dimensional geometry allows us to understand the importance of Slater's condition and other constraint qualifications. We begin by constructing the

Lagrangian for Problem 4:

$$\mathcal{L}(\mathbf{x}, \lambda, \nu) = x_1 + x_2 + \lambda(\|\mathbf{x}\|^2 - 1) + \nu(x_1 - 1). \tag{3.34}$$

Next, the stationarity condition is

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \lambda, \nu) = 0 = \mathbf{1}_2 + 2\lambda \mathbf{x}^* + [\nu \;\; 0]^\top, \tag{3.35}$$

where $\mathbf{1}_n$ is the $n \times 1$ column vector of ones. Since λ and ν only appear in the vector equation's first dimension, Equation (3.35) has no solution for the optimum $\mathbf{x}^* = [1 \; 0]^\top$. Geometrically, this means that the space spanned by the gradients of the active constraints at $\mathbf{x}^*$ does not contain the cost function's gradient $\mathbf{1}_2$. Therefore, there are no dual variables which form a "KKT point" with $\mathbf{x}^*$ and results like Lemma 2 cannot be applied to this problem.

Interestingly, in spite of the fact that the stationarity condition from Equation (KKT) does not apply to Problem 4, we can still prove that strong duality holds. To see this, we begin by solving the stationarity condition in Equation (3.35) for $\mathbf{x}$ in terms of our dual variables:

$$\mathbf{x} = \frac{-1}{2\lambda} \begin{bmatrix} 1 + \nu \\ 1 \end{bmatrix}. \tag{3.36}$$

Since Equation (3.34) is a quadratic function and its Hessian is positive definite for $\lambda > 0$, strict convexity tells us that the critical point in Equation (3.36) is the global minimum and we can use it to write the dual function (Equation (3.10)) for CQ-KKT:

$$\begin{aligned} \mathsf{d}(\lambda, \nu) &= \inf_{\mathbf{x} \in \mathbb{R}^2} \left\{ [1 + \nu \;\; 1]\mathbf{x} + \lambda \|\mathbf{x}\|^2 - \lambda - \nu \right\} \\ &= -\frac{(1+\nu)^2 + 1}{4\lambda} - \lambda - \nu \\ &= \begin{cases} -\frac{(1+\nu)^2+1}{4\lambda} - \lambda - \nu, & \lambda > 0 \\ -\infty, & \lambda = 0. \end{cases} \end{aligned} \tag{3.37}$$

The dual problem (Equation (3.11)) can now be stated in terms of the first case ($\lambda > 0$) alone as

$$\max_{\lambda > 0,\ \nu \in \mathbb{R}} -\frac{(1+\nu)^2 + 1}{4\lambda} - \lambda - \nu, \tag{3.38}$$

and we can reveal the relationship between λ and ν at a critical point by taking a partial derivative:

$$\begin{aligned} \frac{\partial \mathsf{d}(\lambda, \nu)}{\partial \nu} &= -\frac{\nu + 1}{2\lambda} - 1 = 0 \\ \implies \nu &= -2\lambda - 1. \end{aligned} \tag{3.39}$$

Substituting this expression for ν into Equation (3.38) gives us

$$-\frac{(1-2\lambda-1)^2+1}{4\lambda}-\lambda+2\lambda+1=-\frac{4\lambda^2+1}{4\lambda}+\lambda+1=\frac{4\lambda-1}{4\lambda}, \tag{3.40}$$

which implies $\lambda^* \to \infty$ and $\mathsf{d}^* \to 1$. Therefore, the duality gap in the limit is in fact zero, and this example illustrates a subtler lesson than Problem 3: a problem that does not satisfy constraint qualification may exhibit strong duality while failing to provide a dual solution $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ that forms a KKT point with $\mathbf{x}^*$.

In conclusion, even for simple convex problems, Slater's condition is critical for ensuring desirable properties like strong duality and the existence of Lagrange multipliers. Similarly, other constraint qualifications play an important role in understanding and solving the challenging QCQPs we examine in this dissertation.

Linear Independence Constraint Qualification

While useful for convex primal problems, Slater's condition will not be sufficient for proving nice properties about the nonconvex QCQPs that arise in robotics. The strongest (and therefore easiest to prove) constraint qualification we will consider is the *linear independence constraint qualification* (LICQ). We will say that LICQ holds at some feasible $\bar{\mathbf{x}}$ if the gradients

$$\begin{aligned}\nabla h_i(\bar{\mathbf{x}}),\ i&=1,\dots,m\\ \nabla g_i(\bar{\mathbf{x}}),\ i&\in A(\bar{\mathbf{x}}),\end{aligned} \tag{LICQ}$$

are linearly independent, where $A(\bar{\mathbf{x}}) \subseteq [l]$ is the index set of the *active* inequality constraints at $\bar{\mathbf{x}}$:

$$A(\bar{\mathbf{x}}) \triangleq \{i \in [l] \mid g_i(\bar{\mathbf{x}}) = 0\}. \tag{3.41}$$

We will also write "LICQ $(\bar{\mathbf{x}})$" as shorthand for "LICQ holds at $\bar{\mathbf{x}}$".

Abadie Constraint Qualification

Let $h : \mathbb{R}^n \to \mathbb{R}^m$ describe the constraints for an optimization problem constrained only by equalities (i.e., no inequalites). Denote the feasible set of this program as

$$\mathcal{X} \triangleq \{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) = 0\}. \tag{3.42}$$

The *Abadie constraint qualification* (ACQ) holds at $\mathbf{x} \in \mathcal{X}$, denoted $\mathrm{ACQ}_{\mathcal{X}}(\mathbf{x})$, if $\mathcal{X}$ is a smooth manifold nearby $\mathbf{x}$ and $\operatorname{rank}(\nabla h(\mathbf{x})) = \operatorname{codim}_{\mathbf{x}}(\mathcal{X})$.[13] Note that ACQ is *weaker* than LICQ, i.e., LICQ $(\bar{\mathbf{x}}) \implies \mathrm{ACQ}_{\mathcal{X}}(\bar{\mathbf{x}})$, but they can both be used to prove the existence of Lagrange multipliers. While stronger and therefore applicable to fewer problems, LICQ is in general much easier to prove than ACQ and other constraint qualifications. In Section 3.6, we will require

[13] The *codimension* $\operatorname{codim}_{\mathbf{x}}(\mathcal{X}) \triangleq n - \dim_{\mathbf{x}}(\mathcal{X})$ where $\dim_{\mathbf{x}}(\mathcal{X})$ is the local dimension of the manifold $\mathcal{X}$ at $\mathbf{x}$.

that LICQ holds for problem instances in order to prove that their strong duality is stable to perturbations of the cost function's parameters.

3.5.3 Stability for Equality Constraints

In this section, we present a theorem from Cifuentes et al. (2022) that gives sufficient conditions for the SDP-stability of $\mathrm{Q}_{\boldsymbol{\xi}}$ to perturbations of $\boldsymbol{\xi}$. We denote the optimizer to $\mathrm{Q}_{\bar{\boldsymbol{\xi}}}$, which is simply $\mathrm{Q}_{\boldsymbol{\xi}}$ with nominal ZDG parameter $\bar{\boldsymbol{\xi}}$, as $\bar{\mathbf{x}}$. Similarly, $\bar{\boldsymbol{\lambda}}$ and $\bar{\boldsymbol{\nu}}$ are the corresponding optimal dual variables of $\mathrm{D}_{\bar{\boldsymbol{\xi}}}$. We say that $\mathrm{Q}_{\boldsymbol{\xi}}$ is *SDP-stable* near a ZDG parameter $\bar{\boldsymbol{\xi}}$ if there exists some $\epsilon > 0$ such that $\boldsymbol{\xi}$ is ZDG if $\left\|\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}\right\| < \epsilon$.

Theorem 2 (Theorem 3.2 from Cifuentes et al. (2022)). *Consider the family $\mathrm{Q}_{\boldsymbol{\xi}}$ where $\mathbf{C}(\boldsymbol{\xi})$ is a continuous function of $\boldsymbol{\xi}$ and there are no inequality constraints. Let $\bar{\boldsymbol{\xi}}$ be such that $\mathbf{C}(\bar{\boldsymbol{\xi}}) \succeq \mathbf{0}$, $\boldsymbol{\mathcal{Q}}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\nu}})$ has corank-1, and* $\operatorname{val}\left(Q_{\bar{\boldsymbol{\xi}}}\right) = 0$. *If* $\mathrm{ACQ}_{\mathcal{X}}\left(\bar{\mathbf{x}}\right)$ *holds, then $\mathrm{Q}_{\boldsymbol{\xi}}$ is SDP-stable near $\bar{\boldsymbol{\xi}}$, and its primal SDP relaxation $\mathrm{P}_{\boldsymbol{\xi}}$ recovers its minimizer.*

We omit the full proof of Theorem 2, which can be found in Cifuentes et al. (2022).[14] Theorem 2 trivially applies to the case of inactive inequality constraints:

Corollary 2 (Inactive Inequality Constraints). *Consider the general equality- and inequality-constrained QCQP of Problem 1. When the zero-duality-gap point $\mathbf{x}^*$ is* strictly *feasible, the equality-constrained theorem applies since $\boldsymbol{\lambda}_i^* = 0$ when inequality constraint h_i is* inactive *(i.e., strictly obeyed).*

However, we will encounter QCQPs that are cost-perturbations of zero-duality gap QCQPs with active inequality constraints, and inequality constraints are only interesting when they are active at a solution of interest. To determine whether SDP relaxations of these perturbed QCQPs are globally optimal, we must extend Theorem 2 to QCQPs which include inequality constraints.

3.6 Extending Parametric Stability to Semialgebraic Sets

We begin by stating the main result of this section, which is simply an extension of Theorem 2 to include inequality constraints.

Theorem 3 (SDP Stability of $\mathrm{Q}_{\boldsymbol{\xi}}$). *Consider the family $\mathrm{Q}_{\boldsymbol{\xi}}$ where $\mathbf{C}(\boldsymbol{\xi})$ is a continuous function of $\boldsymbol{\xi}$. Let $\bar{\boldsymbol{\xi}}$ be such that $\mathbf{C}(\bar{\boldsymbol{\xi}}) \succeq \mathbf{0}$, $\boldsymbol{\mathcal{Q}}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}})$ has corank-1, and* $\operatorname{val}\left(Q_{\bar{\boldsymbol{\xi}}}\right) = 0$. *If* $\mathrm{LICQ}\left(\bar{\mathbf{x}}\right)$ *holds, then $\mathrm{Q}_{\boldsymbol{\xi}}$ is SDP-stable near $\bar{\boldsymbol{\xi}}$, and its primal SDP relaxation $\mathrm{P}_{\boldsymbol{\xi}}$ recovers its minimizer.*

[14] The statement of Theorem 2 as presented in Cifuentes et al. (2022) also includes the requirement that $\mathbf{b} \neq \mathbf{0}$. However, in this dissertation we deal exclusively with QCQPs homogenized by $s^2 = 1$ which trivially satisfies this requirement. Therefore, we have omitted $\mathbf{b} \neq \mathbf{0}$ for brevity's sake.

Remark 1 (Key Assumptions of Theorem 3): The requirements that $\mathbf{C}(\bar{\boldsymbol{\xi}}) \succeq \mathbf{0}$ and $\operatorname{val}\left(\mathrm{Q}_{\bar{\boldsymbol{\xi}}}\right) = 0$ are equivalent to requiring that $\bar{\boldsymbol{\lambda}} = \mathbf{0}$ and $\bar{\boldsymbol{\nu}} = \mathbf{0}$ are optimal for $\mathrm{D}_{\bar{\boldsymbol{\xi}}}$. This is not particularly restrictive, as the problems we study in this dissertation all have convex cost functions (ensuring $\mathbf{C}(\bar{\boldsymbol{\xi}}) \succeq \mathbf{0}$), and our choice of $\bar{\boldsymbol{\xi}}$ will correspond to some "noise-free" or base case such that $\operatorname{val}\left(\mathrm{Q}_{\bar{\boldsymbol{\xi}}}\right) = 0$ by construction. Therefore, for both Theorem 2 and Theorem 3, the constraint qualifications and a corank-1 Hessian $\boldsymbol{\mathcal{Q}}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}}) = \mathbf{C}(\bar{\boldsymbol{\xi}})$ are the key assumptions.

The inclusion of polynomial inequality constraints makes the feasible set of Problem 1 a *semialgebraic set*. This is in contrast to an *algebraic set*, which is defined as the locus of zeros of a collection of polynomials. It is possible to convert quadratic inequalities into quadratic equalities via the introduction of auxiliary variables:

$$g_i(\mathbf{x}) \leq 0 \iff \exists\, s_i \in \mathbb{R} :\; g_i(\mathbf{x}) + s_i^2 = 0. \tag{3.43}$$

However, this procedure is both computationally inefficient and mathematically inelegant.

The main difference between the sufficient conditions of Theorem 3 and Theorem 2 is the stronger constraint qualification requirement when inequalities are present. Proving Theorem 3 will require the introduction of a few concepts and tools. We define the *Lagrange multiplier mapping*[15] $\mathcal{M} : \Xi \rightrightarrows \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m$ as

$$\begin{aligned} \boldsymbol{\xi} \mapsto &\left\{(\mathbf{x}_\xi, \boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi) : \mathbf{x}_\xi \text{ feasible for } \mathrm{Q}_{\boldsymbol{\xi}},\ (\boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi) \in \Lambda_\xi\left(\mathbf{x}_\xi\right)\right\} \\ &= \left\{(\mathbf{x}_\xi, \boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi) : g(\mathbf{x}_\xi) \leq \mathbf{0}, h(\mathbf{x}_\xi) = \mathbf{0},\ \boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi)\mathbf{x}_\xi = \mathbf{0}\right\}, \end{aligned} \tag{3.44}$$

where $\Lambda_\xi\left(\mathbf{x}\right) \subseteq \mathbb{R}^l_+ \times \mathbb{R}^m$ is the space of Lagrange multipliers[16] at $\mathbf{x}$ (i.e., $\Lambda_\xi\left(\mathbf{x}\right)$ is shorthand for the set of $(\boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi)$ such that $\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi)\mathbf{x}_\xi = \mathbf{0}$). We say that the mapping in Equation (3.44) is *weakly continuous* at $\bar{\boldsymbol{\eta}} \triangleq (\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}}) \in \mathcal{M}(\bar{\boldsymbol{\xi}})$ if there exists $\boldsymbol{\eta}_{\boldsymbol{\xi}} \in \mathcal{M}(\boldsymbol{\xi})$ such that $\boldsymbol{\eta}_{\boldsymbol{\xi}} \to \bar{\boldsymbol{\eta}}$ as $\boldsymbol{\xi} \to \bar{\boldsymbol{\xi}}$. As a first step to proving Theorem 3, we establish that weak continuity is a sufficient condition for SDP-stability.

Proposition 1 (Weak Continuity is Sufficient for Stability)**.** *Let $\bar{\boldsymbol{\xi}}$ be a ZDG parameter, and let $(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}})$ be primal/dual optimal values of $Q_{\bar{\boldsymbol{\xi}}}$. Suppose that $\boldsymbol{\mathcal{Q}}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}})$ has corank-1 and that $\mathcal{M}$ is weakly continuous at $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}})$. Then $\mathrm{Q}_{\boldsymbol{\xi}}$ is SDP-stable near $\bar{\boldsymbol{\xi}}$ and $\mathrm{P}_{\boldsymbol{\xi}}$ recovers its minimizer.*

Proof. Weak continuity ensures that there exists $\boldsymbol{\eta}_{\boldsymbol{\xi}} = (\mathbf{x}_{\boldsymbol{\xi}}, \boldsymbol{\lambda}_{\boldsymbol{\xi}}, \boldsymbol{\nu}_{\boldsymbol{\xi}})$ with $\mathbf{x}_{\boldsymbol{\xi}}$ feasible for $\mathrm{Q}_{\boldsymbol{\xi}}$ and $(\boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi) \in \Lambda_\xi\left(\mathbf{x}_\xi\right)$ such that $\boldsymbol{\lambda}_{\boldsymbol{\xi}} \to \bar{\boldsymbol{\lambda}}$, $\boldsymbol{\nu}_{\boldsymbol{\xi}} \to \bar{\boldsymbol{\nu}}$ as $\boldsymbol{\xi} \to \bar{\boldsymbol{\xi}}$. By the continuity of $\mathbf{C}(\boldsymbol{\xi})$ with respect to $\boldsymbol{\xi}$, $\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}, \boldsymbol{\nu}) \to \boldsymbol{\mathcal{Q}}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}})$ as $\boldsymbol{\xi} \to \bar{\boldsymbol{\xi}}$. Since $(\boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi) \in \Lambda_\xi\left(\mathbf{x}_\xi\right)$, we have that

[15] The double-arrow notation $\rightrightarrows$ is borrowed from Cifuentes et al. (2022) and indicates that the codomain of $\mathcal{M}$ is the set of *subsets* of $\mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m$ instead of individual elements.

[16] In some works, "Lagrange multipliers" refer only to the equality constrained case, and they are instead called "KKT multipliers" when inequalities are introduced. We follow the notation of Boyd and Vandenberghe (2004) and use the term Lagrange multipliers for the general case including both inequality and equality constraints.

$\mathcal{Q}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi)\mathbf{x}_\xi = \mathbf{0}$ and $\mathcal{Q}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi)$ therefore has a zero eigenvalue. Since dual feasibility of $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}})$ implies $\mathcal{Q}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}}) \succeq 0$ and $\operatorname{rank}\left(\mathcal{Q}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}})\right) = n-1$, $\mathcal{Q}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}})$ has $n-1$ positive eigenvalues. Finally, by continuity of eigenvalues, $\mathcal{Q}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi)$ also has $n-1$ positive eigenvalues when $\boldsymbol{\xi} \to \bar{\boldsymbol{\xi}}$. Therefore, $\mathcal{Q}_{\boldsymbol{\xi}}(\boldsymbol{\lambda}_\xi, \boldsymbol{\nu}_\xi) \succeq \mathbf{0}$ and has corank-1, and Lemma 2 completes the proof.

□

Proposition 1 is a modified version of Proposition 3.5 in Cifuentes et al. (2022). The main difference is our inclusion of inequality constraints, but we also ignore cases where the problem's constraints g and h are parameterized by $\boldsymbol{\xi}$.[17] With Proposition 1, we have shown that proving weak continuity of Equation (3.44) is sufficient for proving Theorem 3. Before tackling the dual variables, we will first prove that $\mathbf{x}^*_{\boldsymbol{\xi}} \to \bar{\mathbf{x}}$ as $\boldsymbol{\xi} \to \bar{\boldsymbol{\xi}}$, where $\mathbf{x}^*_{\boldsymbol{\xi}}$ denotes an optimal solution to $\mathrm{Q}_{\boldsymbol{\xi}}$. To do this, we can simply apply Lemma 3.7 from Cifuentes et al. (2022).

Lemma 3 (Continuity of the Primal Variable)**.** *For each $\boldsymbol{\xi}$, let $\mathbf{x}^*_{\boldsymbol{\xi}}$ be an optimal solution of $\mathrm{Q}_{\boldsymbol{\xi}}$. Then $\mathbf{x}^*_{\boldsymbol{\xi}}$ converges to $\bar{\mathbf{x}}$, up to sign.*

Lemma 3 gives us a useful corollary regarding the set of active inequality constraints:

Corollary 3. *For any $\bar{\boldsymbol{\xi}}$, there is a neighbourhood within which $A\left(\mathbf{x}_{\boldsymbol{\xi}}\right) \subseteq A\left(\bar{\mathbf{x}}\right)$.*

Proof. Since for each inactive constraint we have *strict* compliance $g_i(\bar{\mathbf{x}}) < 0$, the continuity of $\mathbf{x}^*_{\boldsymbol{\xi}}$ with respect to $\boldsymbol{\xi}$ ensures that there is a neighbourhood around $\bar{\boldsymbol{\xi}}$ within which all inactive constraints remain inactive. □

In Cifuentes et al. (2022), Lemma 3 is proved in the context of QCQPs without inequality constraints, but it still applies to our more general family of QCQPs. The proof relies on the assumption that $\mathbf{b} \neq \mathbf{0}$, which ensures that $\mathbf{x}^*_{\boldsymbol{\xi}}$ deviates from $\bar{\mathbf{x}}$ continuously as $\boldsymbol{\xi}$ diverges from $\bar{\boldsymbol{\xi}}$. This property is not affected by the presence of inequality constraints in $\mathrm{Q}_{\boldsymbol{\xi}}$.

Finally, recalling Remark 1, we need to prove that the Lagrange multipliers $(\boldsymbol{\lambda}_{\boldsymbol{\xi}}, \boldsymbol{\nu}_{\boldsymbol{\xi}})$ both converge to $\mathbf{0}$ as $\boldsymbol{\xi} \to \bar{\boldsymbol{\xi}}$. Our approach once again mirrors that of Cifuentes et al. (2022), with a strengthened constraint qualification that helps account for the inclusion of inequality constraints. In our proof, $f_{\boldsymbol{\xi}} : \mathbb{R}^n \to \mathbb{R}^k$ denotes the vector function containing all equality and *active* inequality constraints at $\mathbf{x}_{\boldsymbol{\xi}}$, where $k \leq l + m$ is the sum of the number of equality constraints (m) and active inequality constraints ($\left|A\left(\mathbf{x}_{\boldsymbol{\xi}}\right)\right| \leq l$). The vector $\boldsymbol{\mu}_{\boldsymbol{\xi}} \in \mathbb{R}^k$ contains the Lagrange multipliers corresponding to $\mathbf{x}_{\boldsymbol{\xi}}$ for the constraints in $f_{\boldsymbol{\xi}}$.

Lemma 4 (Continuity of Dual Variables)**.** *Let $\mathbf{x}_{\boldsymbol{\xi}}$ be a critical point of $\mathrm{Q}_{\boldsymbol{\xi}}$ for $\boldsymbol{\xi}$ in the neighbourhood of a ZDG parameter $\bar{\boldsymbol{\xi}}$ such that, by Corollary 3, $A\left(\mathbf{x}_{\boldsymbol{\xi}}\right) \subseteq A\left(\bar{\mathbf{x}}\right)$. Let $\sigma_{\boldsymbol{\xi}} > 0$ be the smallest nonzero singular value of the Jacobian $\mathbf{J}_{\boldsymbol{\xi}} \triangleq \nabla f_{\boldsymbol{\xi}}(\mathbf{x}_{\boldsymbol{\xi}}) \in \mathbb{R}^{k \times n}$.*

i) If $\mathrm{LICQ}\left(\mathbf{x}_{\boldsymbol{\xi}}\right)$ *holds, then there exists* $(\boldsymbol{\lambda}_{\boldsymbol{\xi}}, \boldsymbol{\nu}_{\boldsymbol{\xi}}) \in \Lambda_{\boldsymbol{\xi}}\left(\mathbf{x}_{\boldsymbol{\xi}}\right)$ *with* $\left\|\boldsymbol{\mu}_{\boldsymbol{\xi}}\right\| \leq \frac{1}{\sigma_{\boldsymbol{\xi}}} \left\|\nabla c_{\boldsymbol{\xi}}(\mathbf{x}_{\boldsymbol{\xi}})\right\|$.

[17] For the robotics problems we are interested in, the constraints are fixed properties of physical space (e.g., rigid transformations in Equation (2.4)) or known static parameters of a particular robot. Therefore, we do not make use of the more general theory in Cifuentes et al. (2022) for problems with parametric constraints.

ii) If LICQ $(\bar{\mathbf{x}})$ *holds and* $\mathbf{x}_{\boldsymbol{\xi}} \to \bar{\mathbf{x}}$*, then there exists* $(\boldsymbol{\lambda}_{\boldsymbol{\xi}}, \boldsymbol{\nu}_{\boldsymbol{\xi}}) \in \Lambda_{\boldsymbol{\xi}}(\mathbf{x}_{\boldsymbol{\xi}})$ *such that* $\boldsymbol{\lambda}_{\boldsymbol{\xi}} \to \mathbf{0}$ *and* $\boldsymbol{\nu}_{\boldsymbol{\xi}} \to \mathbf{0}$*.*

Proof. i) The space of Lagrange multipliers $\Lambda_{\boldsymbol{\xi}}(\mathbf{x}_{\xi})$ is the intersection of the solution space of the linear system

$$\boldsymbol{\mu_\xi}^\top \mathbf{J}_{\boldsymbol{\xi}} = -\nabla c_{\boldsymbol{\xi}}(\mathbf{x}_{\boldsymbol{\xi}}), \tag{3.45}$$

and the nonnegative orthant $\boldsymbol{\lambda} \geq \mathbf{0}$ for the active inequalities' multipliers. Since $\mathbf{x}_{\boldsymbol{\xi}}$ is a critical point, we know that $\boldsymbol{\mu}_{\boldsymbol{\xi}}$ exists. Since LICQ $(\mathbf{x}_{\boldsymbol{\xi}})$ holds, rank $(\mathbf{J}_{\boldsymbol{\xi}}) = k \leq n$ and we know that $\boldsymbol{\mu}_{\boldsymbol{\xi}}$ is in fact the *unique* solution recovered by the pseudoinverse:

$$\boldsymbol{\mu_\xi}^\top = -\nabla c_{\boldsymbol{\xi}}(\mathbf{x}_{\boldsymbol{\xi}})\mathbf{J}_{\boldsymbol{\xi}}^\dagger. \tag{3.46}$$

Noting that $\left\|\mathbf{J}_{\boldsymbol{\xi}}^\dagger\right\| = \frac{1}{\sigma_{\boldsymbol{\xi}}}$ completes the first part of the proof.

ii) Since LICQ $(\bar{\mathbf{x}})$ holds, $A(\mathbf{x}_{\boldsymbol{\xi}}) \subseteq A(\bar{\mathbf{x}})$, and full rank is an open condition,[18] we know that LICQ $(\mathbf{x}_{\boldsymbol{\xi}})$ holds in a neighbourhood of $\bar{\mathbf{x}}$ for $\boldsymbol{\xi}$ sufficiently close to $\bar{\boldsymbol{\xi}}$. Accordingly, $\sigma_{\boldsymbol{\xi}} > 0$ also holds in this neighbourhood of $\bar{\boldsymbol{\xi}}$. By assumption, $\nabla c_{\bar{\boldsymbol{\xi}}}(\bar{\mathbf{x}}) = 2\mathbf{C}(\bar{\boldsymbol{\xi}})\bar{\mathbf{x}} = \mathbf{0}$. Therefore, applying part i) of this lemma we see that

$$\left\|\boldsymbol{\mu}_{\boldsymbol{\xi}}\right\| \leq \frac{1}{\sigma_{\boldsymbol{\xi}}} \left\|\nabla c_{\boldsymbol{\xi}}(\mathbf{x}_{\boldsymbol{\xi}})\right\| \to 0 \tag{3.47}$$

as $\boldsymbol{\xi} \to \bar{\boldsymbol{\xi}}$. This means that $\boldsymbol{\nu}_{\boldsymbol{\xi}} \to \mathbf{0}$, and since $A(\mathbf{x}_{\boldsymbol{\xi}}) \subseteq A(\bar{\mathbf{x}})$, we also have that the λ_i's corresponding to inactive constraints remain zero via complementary slackness, and $\boldsymbol{\lambda}_{\boldsymbol{\xi}} \to \mathbf{0}$ as desired. □

Lemma 4 is the final tool we need to prove Theorem 3:

Proof of Theorem 3. Recalling Remark 1, we know that $\bar{\boldsymbol{\lambda}} = \mathbf{0}$, $\bar{\boldsymbol{\nu}} = \mathbf{0}$ are optimizers of $\mathrm{D}_{\bar{\boldsymbol{\xi}}}$. Additionally, $\boldsymbol{\mathcal{Q}}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}}) = \mathbf{C}(\bar{\boldsymbol{\xi}})$ has corank-1 by assumption. Lemma 3 gives us that $\mathbf{x}_{\boldsymbol{\xi}}^* \to \bar{\mathbf{x}}$ as $\boldsymbol{\xi} \to \bar{\boldsymbol{\xi}}$. Lemma 4 proves that $\boldsymbol{\nu}_{\boldsymbol{\xi}} \to \mathbf{0} = \bar{\boldsymbol{\nu}}$ and $\boldsymbol{\lambda}_{\boldsymbol{\xi}} \to \mathbf{0} = \bar{\boldsymbol{\lambda}}$ as $\boldsymbol{\xi} \to \bar{\boldsymbol{\xi}}$. Therefore, the assumptions of Proposition 1 hold, which proves Theorem 3. □

3.6.1 Twisted Cubic Example

In this section, we test Theorem 3 on a simple toy problem. Our example is based on the *twisted cubic* variety studied in Cifuentes et al. (2018):

$$Y \triangleq \left\{\mathbf{y} \in \mathbb{R}^3 \mid y_2 = y_1^2, y_3 = y_1 y_2\right\}, \tag{3.48}$$

which is depicted in Figure 3.6. In the example in Cifuentes et al. (2022), Theorem 2 predicts a zero-duality-gap region for a nearest point problem. This region is displayed in Figure 3.7.

[18] I.e., it holds in a neighbourhood (Lewis, 2009).

Problem 5 (Truncated Twisted Cubic)**.** *Consider the semialgebraic set*

$$Y' \triangleq \left\{ \mathbf{y} \in \mathbb{R}^3 \mid y_2 = y_1^2,\ y_3 = y_1 y_2,\ y_1 \geq 0 \right\}, \tag{3.49}$$

which we refer to as the truncated *twisted cubic. For some point $\boldsymbol{\xi} \in \mathbb{R}^3$, consider the problem of finding the nearest (in a Euclidean sense) point $\mathbf{y} \in Y'$ to $\boldsymbol{\xi}$, i.e.:*

$$\min_{\mathbf{y} \in Y'} \quad \|\mathbf{y} - \boldsymbol{\xi}\|^2. \tag{3.50}$$

Since Y' is a quadratic semialgebraic set, Equation (3.50) is a QCQP. For what values of $\boldsymbol{\xi} \notin Y'$ is the SDP relaxation of Equation (3.50) tight? More specifically, can we use Theorem 3 to prove that Equation (3.50) is stable *to perturbations of $\bar{\boldsymbol{\xi}} \in Y'$?*

The only nontrivial application of Theorem 3 to Problem 5 is for $\bar{\boldsymbol{\xi}} = \mathbf{0}$, because this is the only point in Y' where the inequality constraint is active. Since Equation (3.50) is a Euclidean nearest point problem, we have that $\mathbf{C}(\bar{\boldsymbol{\xi}}) \succeq \mathbf{0}$, $\boldsymbol{\mathcal{Q}}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}})$ has corank-1, and $\operatorname{val}\left(\mathrm{Q}_{\bar{\boldsymbol{\xi}}}\right) = 0$ for $\bar{\boldsymbol{\xi}} = 0$ and $\bar{\boldsymbol{\lambda}} = \mathbf{0}$.[19] Therefore, it remains to show that LICQ$\,(\bar{\mathbf{y}})$ holds in order to satisfy the requirements of Theorem 3. The equality and active inequality constraints at $\bar{\mathbf{y}} = \mathbf{0}$ are

$$f_{\bar{\boldsymbol{\xi}}}(\mathbf{y}) = \begin{bmatrix} y_2 - y_1^2 \\ y_3 - y_1 y_2 \\ y_1 \end{bmatrix} = \mathbf{0} \tag{3.51}$$

and the Jacobian is

$$\nabla f_{\bar{\boldsymbol{\xi}}}(\mathbf{y}) = \begin{bmatrix} -2y_1 & 1 & 0 \\ -y_2 & -y_1 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \tag{3.52}$$

which we evaluate at $\bar{\mathbf{y}} = \mathbf{0}$ to obtain

$$\nabla f_{\bar{\boldsymbol{\xi}}}(\mathbf{0}) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}. \tag{3.53}$$

Since Equation (3.53) has full rank, LICQ$\,(\bar{\mathbf{y}})$ holds and Theorem 3 tells us that SDP relaxations of Equation (3.50) are tight for $\boldsymbol{\xi}$ in the neighbourhood of $\mathbf{0}$. This prediction is experimentally verified in Figure 3.8, which displays a contour map of the SDP relaxation gap for values of $\boldsymbol{\xi} \in \mathbb{R}^3$.

[19] See Cifuentes et al. (2022) and Corollary 6 for details.

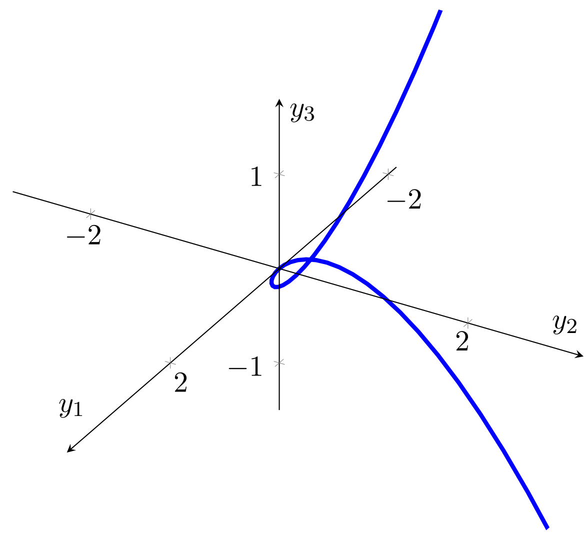


Figure 3.6: The twisted cubic described by Equation (3.48) and used in Problem 5.

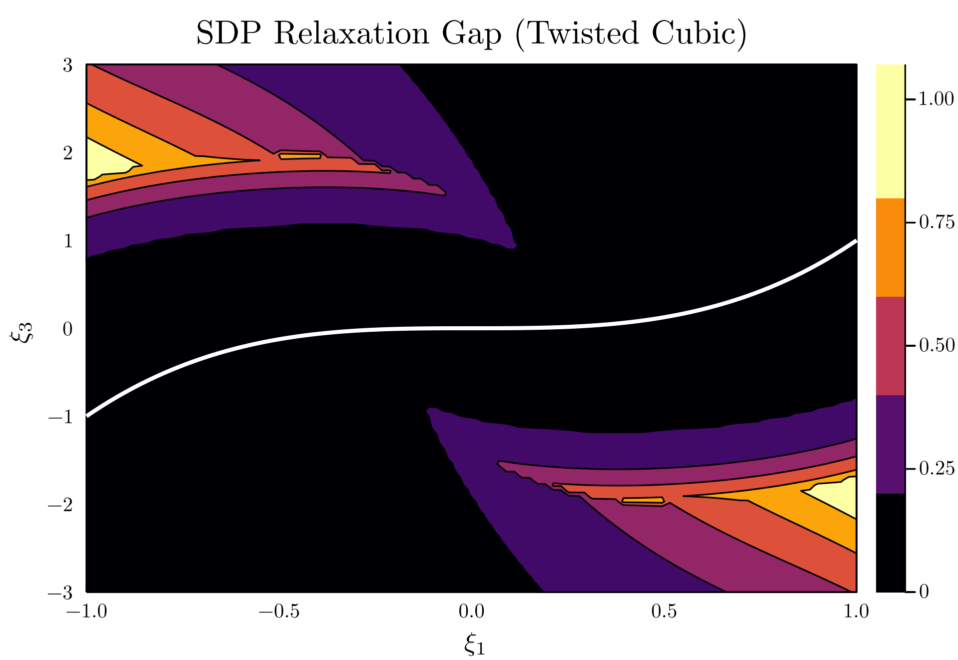


Figure 3.7: Recreation of Figure 1 from Cifuentes et al. (2022) depicting the SDP relaxation gap for the nearest point problem to Equation (3.48) for parameters of the form $\boldsymbol{\xi} \triangleq [\xi_1 \ \xi_1^2 \ \xi_3]^\top$. Note that Theorem 2 correctly predicts that there is a ZDG region in black surrounding each point of the twisted cubic, which is plotted in white.

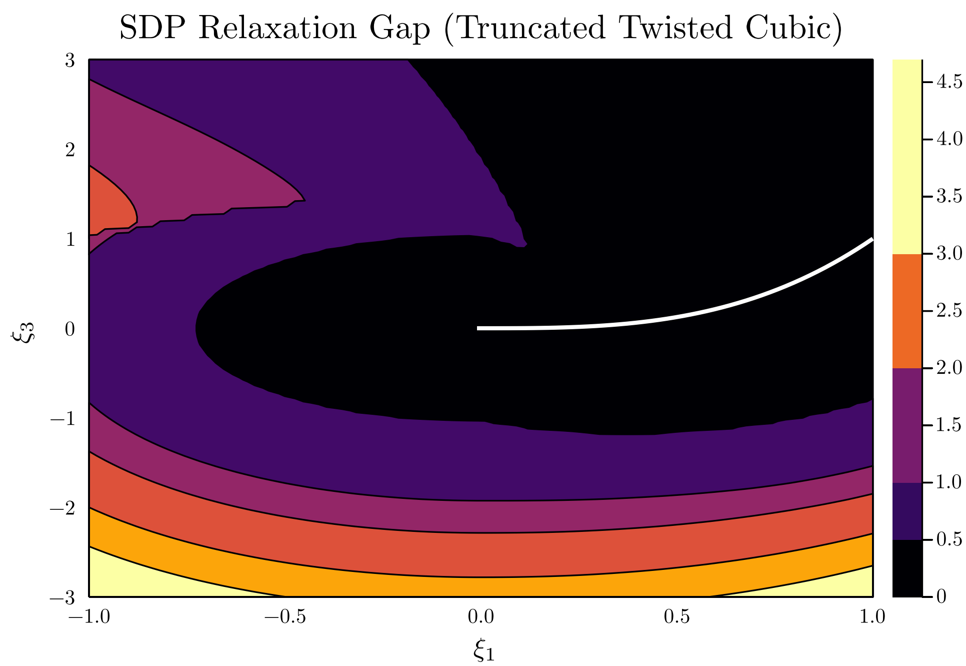


Figure 3.8: The SDP relaxation gap for Equation (3.50) for parameters of the form $\boldsymbol{\xi} \triangleq [\xi_1 \ \xi_1^2 \ \xi_3]^\top$. Note that Theorem 3 correctly predicts that there is a ZDG region in black surrounding each point of the truncated twisted cubic, which is plotted in white.

Chapter 4

Hand-Eye Calibration

> I must first know myself, as the Delphian inscription says; to be curious about that which is not my concern, while I am still in ignorance of my own self, would be ridiculous.
>
> Socrates, *Plato's Phaedrus*

Robots rely on calibrated sensors in order to safely and effectively carry out complex tasks. For mobile robots equipped with multiple instruments, an accurate estimate of the relative pose (extrinsic calibration) between each pair of sensors is crucial in enabling capabilities like reliable localization and mapping. Commercial robots may ship with a factory calibration performed by experts using precision equipment that is unavailable to the end user. During operation, intentional adjustments or unintended mechanical stresses may necessitate recalibration in the field. This need for recalibration outside of a factory or laboratory setting has led to a plethora of automatic calibration methods for a variety of sensor combinations (Pandey et al., 2015; Brookshire and Teller, 2013; Kelly and Sukhatme, 2011; Lambert et al., 2017). These methods operate with varying speed, accuracy, and assumptions about the robot's environment, and also differ in the level of sensor specificity and technician involvement required. Robots capable of true long-term autonomy must leverage accurate and fully automatic calibration procedures that work in their deployed environments. However, most existing approaches do not come with any guarantee that globally optimal[1] calibration parameters will be found. Convergence to a local minimum may result in very poor calibration quality, compromising both the reliability and safety of the robot system.

In this chapter, we develop a procedure for quickly determining the extrinsic calibration between any pair of sensors capable of producing egomotion estimates. This approach is often

[1] It is important to note that global optimality is defined in terms of the available measurements. There is no "magical" method for obtaining the ground truth, and we can expect the global optimum to degrade proportionally to sensor noise, while remaining superior to local minima.

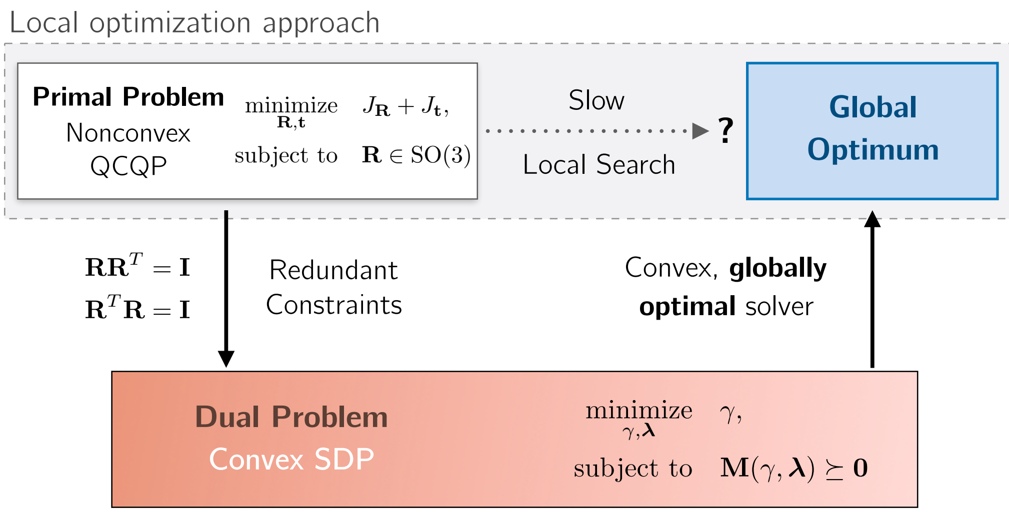


Figure 4.1: We compute the extrinsic calibration between two egomotion sensors by solving the Lagrangian dual of a nonconvex quadratically constrained quadratic program. The dual is a convex problem, meaning it can be efficiently solved to find the global optimum. We leverage recent theoretical results and prove that the primal solution can be extracted from the dual solution, even in the presence of very significant noise. This leads to a certifiably globally optimal approach that outperforms local optimization methods that have no formal guarantees.

referred to as *hand-eye calibration* from its use in determining the extrinsic transformation between a robotic manipulator and a sensor (typically a camera) mounted on or held by the manipulator. The formulation we use is also called the $\mathbf{AX} = \mathbf{XB}$ problem. The novelty of our approach stems from its use of recent advances in the optimization of quadratically constrained quadratic programs (QCQPs). This enables us to provide a *certifiably globally optimal solution* to our cost function, even when severe measurement noise is present, guaranteeing that our approach avoids local minima. The main contributions of our work are:

1. a QCQP formulation of extrinsic calibration from per-sensor egomotion measurements;
2. a fast, certifiably globally optimal convex solution method for our formulation;
3. an application of Theorem 2 which connects calibration observability and the tightness of our Lagrangian dual solution; and
4. an open source implementation and experimental analysis of our algorithm in MATLAB and Python.[2]

In our experiments we compare our work with a method that uses a maximum likelihood estimation (MLE) problem formulation over dual quaternions and a local solver (Brookshire and Teller, 2013). This comparison highlights the viability of our cost function in terms of representing the problem and getting an accurate estimate, while also demonstrating the speed and guaranteed avoidance of local minima that our formulation enables through global convex

[2] See http://github.com/utiasSTARS/certifiable-calibration for code.

optimization. Additionally, the global optimality guarantees could be useful in providing a first estimate for a more precise method using more probabilistic information or dense reconstructions from sensor data, or to present a hypothesis that helps reject outliers for a robust estimation scheme.

4.1 Related Work

We begin with a brief survey of hand-eye calibration algorithms. This well known problem has been studied since the 1980s and we direct readers to the short literature reviews in Heller et al. (2014) and Hu et al. (2019) for more information on recent approaches. We also specifically review the limited literature on extrinsic calibration approaches that use global optimization. Finally, in Section 4.1.3, we summarize the state of the art in the application of convex relaxation techniques to estimation problems in computer vision and robotics.

4.1.1 Hand-Eye Calibration

Much of the early research on hand-eye calibration explored fast, closed-form solutions appropriate for the limited computational resources available at the time. These methods are tailored to the literal hand-eye case, where a robot arm with accurate forward kinematics moves a camera, usually in front of a known calibration target (Tsai and Lenz, 1989). A dual quaternion-based formulation is explored in Daniilidis (1999) and the advantages of coupling translation and rotation estimation are reviewed. The experimental investigation in Horaud and Dornaika (1995) concludes that nonlinear optimization approaches that couple translation and rotation estimation, like the algorithm developed in this paper, provide more accurate solutions in the presence of noise than the simpler but decoupled, closed-form methods. Our problem formulation is similar to the one in Andreff et al. (2001), where observability criteria and different solution methods are presented in a systematic fashion. The unknown scale case is also studied in Wei et al. (2018), where a second-order cone programming solution without optimality guarantees is proposed.

Recent research extends the hand-eye formulation to generic robotic platforms (e.g., self-driving vehicles Walters et al. (2019)) and noisy egomotion measurements. Principled probabilistic (i.e., maximum likelihood) formulations of hand-eye calibration are the subject of Brookshire and Teller (2012) and Brookshire and Teller (2013). A similar approach is applied to a related multi-robot calibration problem in Ma et al. (2016). Our technique eschews a probabilistic cost function in order to leverage the simplicity of a classic geometric formulation, however combining the two approaches is a promising future direction.

4.1.2 Globally Optimal Calibration

The majority of automatic extrinsic calibration approaches do not guarantee that the global optimum will be found. In Levinson and Thrun (2014), the extrinsic calibration of a multi-beam lidar sensor relative to a robot's base frame is computed with a local optimization method. The authors demonstrate that their approach avoids converging to inaccurate local optima, even when initialized with highly inaccurate parameter values, but they provide no formal guarantee that this condition holds in general. Certain bespoke algorithms involving specific sensor pairs and environmental features can be solved in closed form with a minimal set of measurements (Gomez-Ojeda et al., 2015; Zhang and Pless, 2004), but these methods require a local optimization when noise is present. The approach for calibrating extrinsic sensor parameters relative to a manipulator base in Limoyo et al. (2018) relies on solving a point cloud registration problem. While there are globally optimal branch-and-bound (BnB) algorithms that can guarantee a solution to point cloud registration up to a desired accuracy Straub et al. (2017), these techniques can be extremely slow. Similarly, the global optimum of a hand-eye calibration problem is found in Heller et al. (2012) and Ruland et al. (2012) using BnB, but its runtime is orders of magnitude greater than convex methods.

The calibration methods closest to our own are found in Heller et al. (2014) and Heller and Pajdla (2014), where certifiably globally optimal solutions for hand-eye calibration are recovered using the method of convex linear matrix inequality relaxations (Lasserre, 2001). However, in this paper we focus on mobile robotic platforms, provide an MLE formulation distinct from those found in Heller et al. (2014), certify global optimality for more severe noise, theoretically prove the global optimality of our approach with new convex optimization theory (Cifuentes et al., 2022), and connect our approach with a well-known observability result.

4.1.3 Certifiably Globally Optimal Algorithms

Recently, a number of certifiably correct solutions to estimation problems in robotics have been developed. In Carlone et al. (2015), Lagrangian duality was used to verify whether a candidate solution to a pose graph optimization (PGO) is globally optimal. This approach led to fast solvers in Rosen et al. (2019) and Briales and Gonzalez-Jimenez (2017) that exploit the strong duality of PGO when measurement noise is not severe. The related problem of rotation averaging has also proven amenable to globally optimal duality-based solution methods in Fredriksson and Olsson (2012) and Eriksson et al. (2018).

Other problems involving optimization over SO(3) or SE(3) variables that can be solved via their Lagrangian dual include generalized point cloud registration with known correspondences (Briales et al., 2017; Olsson and Eriksson, 2008) and the relative camera pose problem (Briales et al., 2018). Both of these problems involve optimization over a single rotation argument, and both use the method of adding redundant orthogonality constraints found in Anstreicher and Wolkowicz (2000) and Wolkowicz (2002) for recovering a tight dual. In this work, we develop a problem structure similar to that of Briales et al. (2017) and use the same basic solution procedure. In addition to local stability properties examined in Cifuentes et al. (2022), tools from algebraic geometry are used in Brynte et al. (2022) to explain the success of SDP relaxations on nonconvex QCQPs involving SO(3) variables.

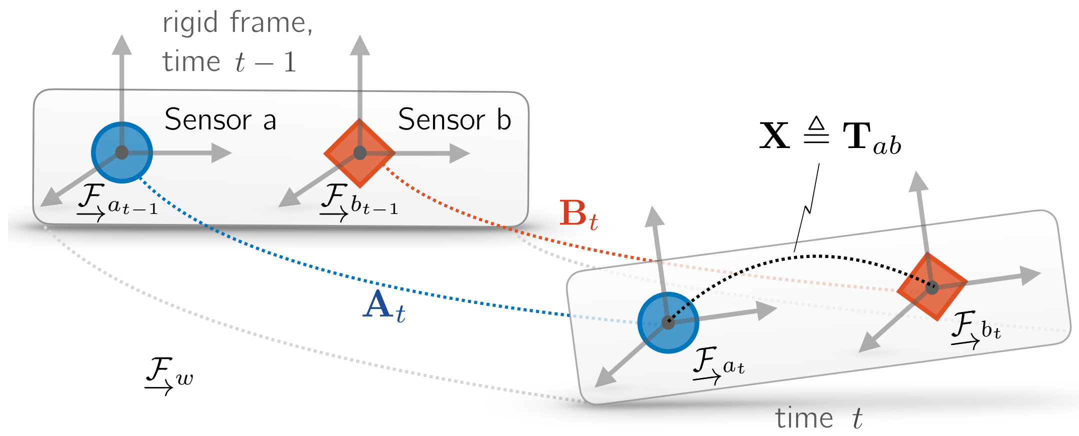


Figure 4.2: We perform extrinsic calibration by using a pair of egomotion sensors a and b fixed to a robot or other mobile rig. The constant extrinsic calibration $\mathbf{X} \in \mathrm{SE}(3)$ relates the poses the two sensors, which remain rigidly attached during measurement acquisition. At each timestep t, the sensors produce egomotion measurements $\mathbf{A}_t, \mathbf{B}_t \in \mathrm{SE}(3)$, which are used to estimate $\mathbf{X}$.

4.2 Problem Formulation

In this section, we derive the $\mathbf{AX} = \mathbf{XB}$ equation and relate it to an MLE formulation of hand-eye calibration. This MLE formulation is a QCQP to which we can apply the convex relaxation-based solution methods of Chapter 3.

4.2.1 Kinematics

Consider two sensors labelled a and b which are rigidly attached to a mobile robot as in Figure 4.2. The sensor rig can move around its environment relative to a static world frame $\underrightarrow{\mathcal{F}}_w$, and we denote the coordinate frames for a and b at time step t with $\underrightarrow{\mathcal{F}}_{a_t}$ and $\underrightarrow{\mathcal{F}}_{b_t}$. We assume that the sensors produce pairs of noisy egomotion estimates represented by homogeneous transformation matrices $\tilde{\mathbf{T}}_{a_{t-1}a_t}, \tilde{\mathbf{T}}_{b_{t-1}b_t} \in \mathrm{SE}(3)$ corresponding to the motion from time step $t-1$ to time step t. For unsynchronized sensors, this can be achieved via temporal calibration (Kelly and Sukhatme, 2014) and an interpolation procedure (Brookshire and Teller, 2012).

We represent the constant extrinsic calibration between $\underrightarrow{\mathcal{F}}_{a_t}$ and $\underrightarrow{\mathcal{F}}_{b_t}$ with the homogeneous transformation matrix

$$\mathbf{X} \triangleq \mathbf{T}_{ab} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathrm{SE}(3), \tag{4.1}$$

where the sub- and superscripts in $\mathbf{R}_{ab}$ and $\mathbf{p}_a^{ba}$ have been dropped for convenience. At any time t, the following relationship holds between $\mathbf{X}$ and the absolute pose of sensors a and b in the world frame $\underrightarrow{\mathcal{F}}_w$:

$$\mathbf{T}_{b_t} \triangleq \mathbf{T}_{wb_t} = \mathbf{T}_{a_t}\mathbf{X}, \tag{4.2}$$

where $\mathbf{T}_{a_t} \triangleq \mathbf{T}_{wa_t}$. We define the *egomotion* of sensor a from time step $t-1$ to time step t as

$$\mathbf{A}_t \triangleq \mathbf{T}_{a_{t-1}a_t} = \begin{bmatrix} \mathbf{R}_{a_t} & \mathbf{p}_{a_t} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{4.3}$$

and denote the equivalent egomotion of sensor b as $\mathbf{B}_t$.

We now use the fact that the extrinsic calibration is constant to derive a useful equation relating $\mathbf{X}$ to the egomotions $\mathbf{A}_t$ and $\mathbf{B}_t$:

$$\begin{aligned} \mathbf{T}_{a_t}\mathbf{X} &= \mathbf{T}_{b_t} \\ \mathbf{T}_{a_{t-1}}\mathbf{A}_t\mathbf{X} &= \mathbf{T}_{b_{t-1}}\mathbf{B}_t \\ \mathbf{A}_t\mathbf{X} &= \mathbf{T}_{a_{t-1}}^{-1}\mathbf{T}_{b_{t-1}}\mathbf{B}_t \\ \mathbf{A}_t\mathbf{X} &= \mathbf{X}\mathbf{B}_t. \end{aligned} \tag{4.4}$$

Equation (4.4) allows us to formulate an optimization-based estimator of $\mathbf{X}$ from the noisy relative motion measurements $\tilde{\mathbf{A}}_t$ and $\tilde{\mathbf{B}}_t$ for $t \in [T]$, where T is the number of measurements available.

4.2.2 QCQP Formulation

We are now able to formulate a QCQP that seeks an optimal $\mathbf{X}$ matrix:

$$\min_{\mathbf{R}\in \mathrm{SO}(3),\ \mathbf{p}\in\mathbb{R}^3} \quad J_{\mathbf{R}} + J_{\mathbf{p}}, \tag{4.5}$$

where

$$J_{\mathbf{R}} = \frac{1}{2}\sum_{t=1}^{T} \kappa_t \left\| \tilde{\mathbf{R}}_{a_t}\mathbf{R} - \mathbf{R}\tilde{\mathbf{R}}_{b_t} \right\|_{\mathrm{F}}^2 \tag{4.6}$$

is the rotation cost and

$$J_{\mathbf{p}} = \frac{1}{2}\sum_{t=1}^{T} \tau_t \left\| \mathbf{R}\tilde{\mathbf{p}}_{b_t} + \mathbf{p} - \tilde{\mathbf{R}}_{a_t}\mathbf{p} - \tilde{\mathbf{p}}_{a_t} \right\|^2 \tag{4.7}$$

is the translation cost. The cost function is derived by expanding the expression $\left\| \tilde{\mathbf{A}}_t\mathbf{X} - \mathbf{X}\tilde{\mathbf{B}}_t \right\|_{\mathrm{F}}^2$ and weighting each term with rotational and translational measurement parameters κ_t and τ_t. These parameters have a probabilistic interpretation which we will derive in Section 4.2.3.

Since the terms within both quadratic norms in the primal problem Equation (4.5) are either constant or linear with respect to our optimization variables $\mathbf{R}$ and $\mathbf{p}$, the resulting cost function is quadratic. The constraint $\mathbf{R} \in \mathrm{SO}(3)$ can also be enforced through quadratic equations in Equation (2.2) and Equation (2.3). These are orthogonality and "right-handedness" constraints on $\mathbf{R}$. While the handedness constraints distinguish SO(3) from its complement in O(3), O(3) is the disjoint union of these sets and therefore the orthogonality constraints alone are often sufficient for estimation problems (Rosen et al., 2019). However, our experimental results in Section 4.4.1 suggest that enforcing handedness is useful for our particular problem and solution method.

We add an additional homogenizing variable s that makes the objective and constraints purely quadratic (i.e., no linear or constant terms in the objective and no linear terms in the equality constraints). This trick lets us apply our analysis from Chapter 3, where we dealt exclusively with homogenized QCQPs of the form in Equation (3.2). The homogenized primal problem has modified translational cost

$$J'_{\mathbf{p}} = \frac{1}{2}\sum_{t=1}^{T} \tau_t \left\| \mathbf{R}\tilde{\mathbf{p}}_{b_t} + \mathbf{p} - \tilde{\mathbf{R}}_{a_t}\mathbf{p} - s\tilde{\mathbf{p}}_{a_t} \right\|^2 \tag{4.8}$$

and includes the additional constraint $s^2 = 1$. Finally, the homogenized SO(3) constraints are

$$\begin{aligned} \mathbf{R}^\top \mathbf{R} &= s^2 \mathbf{I}, \\ \mathbf{R}_i \times \mathbf{R}_j &= s\mathbf{R}_k, \ (i,j,k) \in \text{cyclic}(1,2,3). \end{aligned} \tag{4.9}$$

Note that while the orthogonality constraints matched the form of Problem 1 before homogenization in Equation (4.9), the implementation of our method closely follows Briales et al. (2017) and therefore includes this factor of s^2, which does not affect the solution or our solver's performance.

4.2.3 Maximum Likelihood Interpretation

Typically, a sensor's egomotion estimates $\tilde{\mathbf{A}}$ or $\tilde{\mathbf{B}}$ are computed from noisy measurements. In this section, we demonstrate that the minimizer of the QCQP formulation in Equation (4.5) is actually a maximum likelihood estimate (MLE) for a particular model of sensors a and b, namely, we assume that

1. sensor a produces idealized egomotion estimates (i.e., $\tilde{\mathbf{A}}_t = \mathbf{A}_t \ \forall t \in [T]$);

2. sensor b produces egomotion translation estimates corrupted by independent zero-mean isotropic Gaussian noise (i.e., $\tilde{\mathbf{p}}_{b_t} \sim \mathcal{N}\left(\mathbf{p}_{b_t}, \sigma_t^2 \mathbf{I}\right)$ and $\tau_t \triangleq \sigma_t^{-2}$); and

3. sensor b produces egomotion rotation estimates corrupted by noise drawn from independent isotropic Langevin distributions (i.e., $\tilde{\mathbf{R}}_{b_t} \sim \text{Lang}\left(\mathbf{R}_{b_t}, \kappa_t\right)$).

These assumptions are not too unrealistic in practice. For example, in the actual "hand-eye" setup involving a robot manipulator and a camera, the manipulator's joint encoders may provide precise measurements leading to egomotion estimates that are far more accurate than the camera's. In this case, we would treat the manipulator's encoders as the idealized sensor a and assume that the noisier camera is sensor b. As our experimental results in Section 4.4

reveal, ignoring a probabilistic formulation altogether and setting $\kappa_t = \tau_t = 1$ at all time steps t is often sufficient for an accurate estimate of $\mathbf{X}$.

In order to prove that Equation (4.5) yields an MLE solution under these assumptions, let us recall that the joint probability over independent random variables is simply the product

$$p(\left\{\tilde{\mathbf{B}}_t\right\}_{t=1}^T \mid \mathbf{X}) = \prod_{t=1}^T p(\tilde{\mathbf{B}}_t \mid \mathbf{X}) = \prod_{t=1}^T p(\tilde{\mathbf{p}}_{b_t} \mid \mathbf{X}) p(\tilde{\mathbf{R}}_{b_t} \mid \mathbf{X}). \tag{4.10}$$

Since both the Gaussian and Langevin distributions are in the exponential family, we can simplify our development by taking the log-likelihood of Equation (4.10):

$$\log p(\left\{\tilde{\mathbf{B}}_t\right\}_{t=1}^T \mid \mathbf{X}) = \log\left(\prod_{t=1}^T p(\tilde{\mathbf{B}}_t \mid \mathbf{X})\right) = \sum_{t=1}^T \left(\log p(\tilde{\mathbf{p}}_{b_t} \mid \mathbf{X}) + \log p(\tilde{\mathbf{R}}_{b_t} \mid \mathbf{X})\right). \tag{4.11}$$

The logarithm is monotonic, meaning the MLE problem (Barfoot, 2017) can be formulated as

$$\min_{\mathbf{X} \in \mathrm{SE}(3)} \quad -\sum_{t=1}^T \left(\log p(\tilde{\mathbf{p}}_{b_t} \mid \mathbf{X}) + \log p(\tilde{\mathbf{R}}_{b_t} \mid \mathbf{X})\right). \tag{4.12}$$

We now proceed to demonstrate that Equation (4.12) is equivalent to Equation (4.5). Using our assumptions that sensor a is perfect and $\tilde{\mathbf{p}}_{b_t} \sim \mathcal{N}\left(\mathbf{p}_{b_t}, \sigma_t^2 \mathbf{I}\right)$, the observation model for b's translation at time step t is

$$\tilde{\mathbf{p}}_{b_t} = \mathbf{R}^\top \left(\mathbf{R}_{a_t}\mathbf{p} + \mathbf{p}_{a_t} - \mathbf{p}\right) + \mathbf{p}_{\epsilon_t}, \tag{4.13}$$

where $\mathbf{p}_{\epsilon_t} \sim \mathcal{N}\left(\mathbf{0}, \sigma_t^2 \mathbf{I}\right)$. Therefore,

$$\tilde{\mathbf{p}}_{b_t} - \mathbf{R}^\top \left(\mathbf{R}_{a_t}\mathbf{p} + \mathbf{p}_{a_t} - \mathbf{p}\right) \sim \mathcal{N}\left(\mathbf{0}, \sigma_t^2 \mathbf{I}\right), \tag{4.14}$$

and since the noise is isotropic and therefore rotation invariant,

$$\mathbf{R}\tilde{\mathbf{p}}_{b_t} - \mathbf{R}_{a_t}\mathbf{p} + \mathbf{p}_{a_t} - \mathbf{p} \sim \mathcal{N}\left(\mathbf{0}, \sigma_t^2 \mathbf{I}\right), \tag{4.15}$$

which is precisely the translation residual for time step t from the QCQP cost in Equation (4.5). Therefore, each squared residual, once weighted by τ_t, is equal to $\log p(\tilde{\mathbf{p}}_{b_t} \mid \mathbf{X})$ minus a constant factor equal to the logarithm of the Gaussian's normalization constant.

Similarly, the observation model for b's rotation is

$$\tilde{\mathbf{R}}_{b_t} = \mathbf{R}^\top \mathbf{R}_{a_t} \mathbf{R} \mathbf{R}_{\epsilon_t}, \tag{4.16}$$

where $\mathbf{R}_{\epsilon_t} \sim \mathrm{Lang}\left(\mathbf{I}, \kappa_t\right)$. The log-likelihood function is therefore

$$\log p(\tilde{\mathbf{R}}_{b_t} \mid \mathbf{X}) = -c(\kappa_t) + \kappa_t \mathrm{tr}\left(\mathbf{R}^\top \mathbf{R}_{a_t}^\top \mathbf{R} \tilde{\mathbf{R}}_{b_t}\right), \tag{4.17}$$

to which we apply the identity $\mathrm{tr}\left(\mathbf{A}^\top \mathbf{B}\right) = d - \frac{1}{2}\left\|\mathbf{A}-\mathbf{B}\right\|_{\mathrm{F}}^2$ (Rosen et al., 2019) to obtain

$$\log p(\tilde{\mathbf{R}}_{b_t} \mid \mathbf{X}) = -c(\kappa_t) - \frac{\kappa_t}{2}\left\|\mathbf{R}_{a_t}\mathbf{R} - \mathbf{R}\tilde{\mathbf{R}}_{b_t}\right\|_{\mathrm{F}}^2 + 3, \tag{4.18}$$

demonstrating that the rotation residuals in Equation (4.5) are equivalent to the log-likelihood values in Equation (4.12).

4.2.4 Decoupling Translation and Rotation

In this section, we massage our QCQP formulation of extrinsic sensor calibration into a form that allows for a more efficient application of the SDP relaxation machinery of Chapter 3. We begin by simplifying the cost function of (4.5) by reorganizing $\mathbf{X}$ into a column vector

$$\begin{aligned}\mathbf{x} &= \left[\mathbf{p}^\top \;\; \mathbf{r}^\top \;\; s\right]^\top, \\ \mathbf{r} &= \mathrm{vec}\left(\mathbf{R}\right),\end{aligned} \tag{4.19}$$

where $\mathrm{vec}\left(\cdot\right) : \mathbb{R}^{m\times n} \to \mathbb{R}^{mn}$ is a column-wise vectorization of its matrix argument (i.e., it "vertically" stacks b columns). This allows us to use the matrix Kronecker product $\otimes$ via the identity (Fackler, 2005)

$$\mathrm{vec}\left(\mathbf{ACB}\right) = (\mathbf{B}^\top \otimes \mathbf{A})\mathrm{vec}\left(\mathbf{C}\right) \tag{4.20}$$

for each term in the cost function of Equation (4.5) and write the cost as a quadratic form,

$$J_{\mathbf{R}} + J'_{\mathbf{p}} = \mathbf{x}^\top \mathbf{C}\mathbf{x}, \tag{4.21}$$

where

$$\mathbf{C} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 10} \\ \mathbf{0}_{10\times 3} & \mathbf{C}_{\mathbf{r}} \end{bmatrix} + \mathbf{C}_{\mathbf{p}} \in \mathbb{S}^{13}, \tag{4.22}$$

and

$$\mathbf{C}_{\mathbf{r}} = \begin{bmatrix} \sum_{t=1}^{T} \kappa_t \mathbf{M}_{\mathbf{r},t}^T \mathbf{M}_{\mathbf{r},t} & \mathbf{0}_{9\times 1} \\ \mathbf{0}_{1\times 9} & 0 \end{bmatrix}, \tag{4.23}$$

with each sub-matrix $\mathbf{M}_{\mathbf{r},t}$ taking the form

$$\mathbf{M}_{\mathbf{r},t} = (\mathbf{I}\otimes\mathbf{R}_{a_t}) - (\mathbf{R}_{b_t}^\top \otimes \mathbf{I}) \in \mathbb{R}^{9\times 9}. \tag{4.24}$$

Similarly for the translation components:

$$\begin{aligned}\mathbf{C}_{\mathbf{p}} &= \sum_{t=1}^{T} \tau_i\, \mathbf{M}_{\mathbf{p},t}^\top \mathbf{M}_{\mathbf{p},t} \in \mathbb{S}^{13}, \\ \mathbf{M}_{\mathbf{p},t} &= \left[\mathbf{I} - \mathbf{R}_{a_t} \;\; (\mathbf{p}_{b_t}^\top \otimes \mathbf{I}) \;\; -\mathbf{p}_{a_i}\right].\end{aligned} \tag{4.25}$$

Next, we note that the translation $\mathbf{p}$ is unconstrained and can therefore be solved in closed form given the optimal rotation $\mathbf{R}^*$. Following the procedure in Briales et al. (2017), we express our cost as the block matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{\mathbf{p},\mathbf{p}} & \mathbf{C}_{\mathbf{p},\boldsymbol{r}} \\ \mathbf{C}_{\boldsymbol{r},\mathbf{p}} & \mathbf{C}_{\boldsymbol{r},\boldsymbol{r}} \end{bmatrix}, \tag{4.26}$$

which allows us to concisely write the optimal translation as

$$\mathbf{p}^*(\mathbf{R}^*) = -\mathbf{C}_{\mathbf{p},\mathbf{p}}^{-1}\mathbf{C}_{\mathbf{p},\boldsymbol{r}}\boldsymbol{r}^*, \tag{4.27}$$

where $\boldsymbol{r} = \begin{bmatrix}\mathbf{r}^\top & s\end{bmatrix}^\top \in \mathbb{R}^{10}$ is the homogenized rotation vector. We can now substitute this expression for $\mathbf{p}^*$ into the cost function and use the Schur complement to compactly define

$$\mathbf{C}' \triangleq \mathbf{C}/\mathbf{C}_{\mathbf{p},\mathbf{p}} = \mathbf{C}_{\boldsymbol{r},\boldsymbol{r}} - \mathbf{C}_{\boldsymbol{r},\mathbf{p}}\mathbf{C}_{\mathbf{p},\mathbf{p}}^{-1}\mathbf{C}_{\mathbf{p},\boldsymbol{r}}. \tag{4.28}$$

This approach leads to a simpler QCQP with only $\boldsymbol{r}$ as a variable:

$$\begin{aligned} \underset{\boldsymbol{r},y}{\text{minimize}} \quad & \boldsymbol{r}^T\mathbf{C}'\boldsymbol{r}, \\ \text{subject to} \quad & \mathbf{R} \in \mathrm{SO}(3), \\ & s^2 = 1. \end{aligned} \tag{4.29}$$

Equation (4.29) is an equality-constrained instance of Problem 1, and we are now ready to solve it with either the primal or dual SDP relaxation in Problem 2.

4.3 SDP Tightness and Stability

In this section, we use Theorem 2 to derive sufficient conditions for our convex relaxation-based approach to hand-eye calibration to be *tight*, ensuring that a certifiably globally optimal solution to the primal problem can be extracted from the solution to its convex relaxation. Throughout this section, we will be dealing with a relaxed and homogeneous version of the QCQP in Equation (4.5):

$$\min_{\mathbf{R}\in\mathrm{O}(3),\ \mathbf{p}\in\mathbb{R}^3,\ s^2=1} \quad c(\mathbf{x}) \triangleq J_{\mathbf{R}} + J'_{\mathbf{p}}. \tag{4.30}$$

Equation (4.30) is a relaxation of Equation (4.5) because we have expanded the feasible set of $\mathbf{R}$ to $\mathrm{O}(3)$, which includes reflections in addition to rotations. An important consequence of this fact is that any instance of Equation (4.30) which exhibits strong duality with a minimizer in SO(3) will also exhibit strong duality if its constraints are "tightened" to limit its feasible set to SO(3). Therefore, the smaller constraint set of Equation (4.30) will not only make the proof of a ZDG region simpler, it will also serve as a ZDG existence proof of any formulation that adds constraints (e.g., Equation (4.5) with handedness or redundant orthogonality constraints).

Proposition 2 (Tightness of the Noise-Free Case)**.** *Any noise-free instance of Equation* (4.30) *exhibits strong duality (i.e., the duality gap is zero and the optimal solution can be extracted from its dual SDP solution).*

Proof. We will use Lemma 2, which requires that we present a primal solution $\mathbf{x}^* \in \mathbb{R}^n$ and a dual solution $\boldsymbol{\nu}^* \in \mathbb{R}^m$ that satisfy

1. primal feasibility ($h(\mathbf{x}^*) = \mathbf{0}$);
2. dual feasibility ($\boldsymbol{\mathcal{Q}}(\boldsymbol{\nu}^*) \succeq \mathbf{0}$); and
3. stationarity ($\boldsymbol{\mathcal{Q}}_{\boldsymbol{\xi}}(\boldsymbol{\nu}^*)\mathbf{x}^* = \mathbf{0}$),

where $\boldsymbol{\mathcal{Q}}$ is half the Hessian of the Lagrangian of Equation (4.30). Selecting the ground truth rotation $\mathbf{R}^*$ and translation $\mathbf{p}^*$ clearly satisfy primal feasibility since $\mathbf{R}^*$ is a rotation matrix. If we select $\boldsymbol{\nu}^* = \mathbf{0}$, we get $\boldsymbol{\mathcal{Q}}(\boldsymbol{\nu}^*) = \nabla^2 c$ which is clearly positive semidefinite because the cost function c is a positive sum of convex norms and therefore convex. Finally, since the ground truth values $\mathbf{R}^*$ and $\mathbf{p}^*$ give a cost of zero and $c(\mathbf{x})$ is nonnegative, $\mathbf{x}^*$ is an unconstrained minimizer and

$$2\boldsymbol{\mathcal{Q}}(\boldsymbol{\nu}^*)\mathbf{x}^* = 2\mathbf{C}\mathbf{x}^* = \nabla c(\mathbf{x}^*) = \mathbf{0}. \tag{4.31}$$

□

Proposition 2 trivially extends to Equation (4.5) with any number of redundant constraints:

Corollary 4 (SO(3) and Redundant Constraints)**.** *Any formulation with* SO(3) *and redundant constraints also exhibit strong duality for all noise-free instances.*

Proof. Redundant constraints by definition do not change the primal feasibility of the ground truth. Likewise, if the ground truth value $\mathbf{x}^*$ already corresponds to an element of SO(3), then the primal feasibility is unaffected. Finally, assigning $\nu_i^* = 0$ for all new constraints h_i does not affect the dual feasibility or stationarity property, since their proof in Proposition 2 relies only on the fact that $\mathbf{x}^*$ is an unconstrained minimizer and the cost function is convex. □

Before proceeding, we present a key lemma relating sensor rig motion to observability and the rank of the cost function matrix.

Lemma 5 (Cost Function Matrix Rank)**.** *Consider the* nonhomogeneous *form of Equation* (4.5) *(i.e., without the addition of homogenizing variable s and its constraint $s^2 = 1$). Its cost function can be written as*

$$c(\mathbf{x}) = \mathbf{x}^\top \mathbf{C}\mathbf{x} + \mathbf{c}^\top \mathbf{x} + \text{const}. \tag{4.32}$$

For a cost function generated by noise-free measurement data, the Hessian $\mathbf{C}$ *of Equation* (4.32) *is full rank (and therefore positive definite) if the following conditions hold:*

1. *the sensor rig rotates about two unique axes during time steps i and j (i.e., rotations $\mathbf{R}_{a_i} \neq \mathbf{I}$ and $\mathbf{R}_{a_j} \neq \mathbf{I}$ are about distinct axes); and*

2. *the measurements at the same time steps i and j are such that*

$$\begin{bmatrix} \mathbf{R}\mathbf{p}_{b_i} \\ \mathbf{R}\mathbf{p}_{b_j} \end{bmatrix} \notin \operatorname{col} \begin{bmatrix} \mathbf{I} - \mathbf{R}_{a_i} \\ \mathbf{I} - \mathbf{R}_{a_j} \end{bmatrix}, \tag{4.33}$$

where col *is the column space or range of its matrix argument.*

Proof. We first note that $c(\mathbf{x})$ is a positively weighted sum of quadratic norms, therefore $c(\mathbf{x}) \geq 0$ and $\nabla^2 c(\mathbf{x}) = \mathbf{C} \succeq \mathbf{0}$. The first-order optimality condition Bertsekas (1999) gives us

$$\nabla c(\mathbf{x})|_{\mathbf{x}^*} = \mathbf{0} \implies 2\mathbf{C}\mathbf{x}^* = -\mathbf{c} \neq \mathbf{0}. \tag{4.34}$$

Thus, if the solution $\mathbf{x}^*$ is unique, then $\mathbf{C} \succ 0$ and is of full rank, otherwise the nullspace of $\mathbf{C}$ provides infinite solutions. Therefore, it suffices to show that the global (unconstrained) minimizer $\mathbf{x}^*$ is unique in order to prove that $\mathbf{C}$ is full rank. Since $c(\mathbf{x}) \geq 0$ is a sum of squared residuals that are all equal to zero when $\mathbf{x}^* = \begin{bmatrix} \mathbf{p}^{*\top} & \mathbf{r}^{*\top} \end{bmatrix}^\top$, where $\mathbf{p}^*$ and $\mathbf{r}^*$ are the true calibration parameters used to generate the noise-free measurements, we see that $c(\mathbf{x}^*) = 0$ is an unconstrained global minimum of $c(\mathbf{x})$. Thus, we must show that $c(\mathbf{x}) = 0$ implies that $\mathbf{x} = \mathbf{x}^*$.

We will first demonstrate that $J_{\mathbf{R}}(\mathbf{r}) = 0$ if and only if $\mathbf{r} = \alpha\mathbf{r}^\star,\ \alpha \in \mathbb{R}$. To accomplish this, we will note that each squared residual term $\mathbf{R}_{a_t}\mathbf{R} - \mathbf{R}\mathbf{R}_{b_t}$ is equal to zero if and only if

$$\mathbf{R} = \mathbf{R}_{a_t}\mathbf{R}\,\mathbf{R}_{b_t}^\top. \tag{4.35}$$

Using the Kronecker product's vectorization identity yields

$$(\mathbf{I}_9 - \mathbf{R}_{b_t} \otimes \mathbf{R}_{a_t})\mathbf{r} = \mathbf{0}_9,\ t \in \{i, j\}. \tag{4.36}$$

Since $\mathbf{R}_{b_i}$ and $\mathbf{R}_{b_j}$ are rotations about distinct axes, Lemma 1 from Andreff et al. (2001) ensures that the system in Equation (4.36) has a unique solution up to scale α.

Having established that $f(\mathbf{x}) = 0 \implies \mathbf{r} = \alpha\mathbf{r}^*$, we will now investigate the squared residual terms of $J_{\mathbf{p}}$:

$$\mathbf{R}\mathbf{p}_{b_t} + \mathbf{p} - \mathbf{R}_{a_t}\mathbf{p} - \mathbf{p}_{a_t}. \tag{4.37}$$

Substituting in $\alpha\mathbf{r}^*$ and setting the ith residual to zero gives

$$(\mathbf{I} - \mathbf{R}_{a_i})\mathbf{p} + \alpha\mathbf{R}^*\mathbf{p}_{b_i} = \mathbf{p}_{a_i}, \tag{4.38}$$

which we can rearrange and combine with the jth residual to get

$$\begin{bmatrix} \mathbf{I} - \mathbf{R}_{a_i} & \mathbf{R}^* \mathbf{p}_{b_i} \\ \mathbf{I} - \mathbf{R}_{a_j} & \mathbf{R}^* \mathbf{p}_{b_j} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \alpha \end{bmatrix} \triangleq \mathbf{M} \begin{bmatrix} \mathbf{p} \\ \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{a_i} \\ \mathbf{p}_{a_j} \end{bmatrix}, \tag{4.39}$$

where $\mathbf{M} \in \mathbb{R}^{6\times 4}$ has been defined for convenience. In order to prove that $\mathbf{p} = \mathbf{p}^*, \alpha = 1$ is a unique solution, we must demonstrate that $\mathbf{M}$ is of full rank. First, we quickly show that

$$\operatorname{rank}\left(\begin{bmatrix} \mathbf{I} - \mathbf{R}_{a_i} \\ \mathbf{I} - \mathbf{R}_{a_j} \end{bmatrix} \right) = 3. \tag{4.40}$$

Suppose $\exists\, \mathbf{w} \neq \mathbf{0}$ such that $(\mathbf{I} - \mathbf{R}_{a_i})\,\mathbf{w} = \mathbf{0}$ and $(\mathbf{I} - \mathbf{R}_{a_j})\mathbf{w} = \mathbf{0}$. This would mean that $\mathbf{w}$ is along the axis of rotation for both $\mathbf{R}_{a_i}$ and $\mathbf{R}_{a_j}$, which contradicts our assumption. Since the left three columns of $\mathbf{M}$ are rank 3, $\mathbf{M}$ is rank 4 when its fourth column is not in the span of of the first 3 columns. This is precisely our assumption in Equation (4.33). Therefore, $\mathbf{x}^*$ is the unique minimizer of $c(\mathbf{x})$ and $\mathbf{C}$ is full rank and strictly convex. □

One of Lemma 5's requirements is that the sensor platform rotates about two distinct axes in a fixed global reference frame. This is a well-known observability criterion found in other formulations of extrinsic calibration (Andreff et al., 2001; Brookshire and Teller, 2013; Kelly and Sukhatme, 2011).

With Lemma 5 in hand, we are ready to prove Theorem 4, which is the main result of this section and essentially an application of Theorem 2 to hand-eye calibration.

Theorem 4 (SDP Stability of Equation (4.30))**.** *Let*

$$\boldsymbol{\xi} \triangleq \operatorname{Vec}\left(\left\{ \tilde{\mathbf{R}}_{a_t}, \tilde{\mathbf{p}}_{a_t}, \tilde{\mathbf{R}}_{b_t}, \tilde{\mathbf{p}}_{b_t} \right\}_{t=1}^{T} \right) \in \mathbb{R}^{24T}, \tag{4.41}$$

where $\operatorname{Vec}(\cdot)$ *applies* $\operatorname{vec}(\cdot)$ *to each element of its argument and vertically concatenates its result into a column vector. All the egomotion measurements that parameterize the cost function of Equation* (4.30) *are captured in* $\boldsymbol{\xi}$*. Let* $\bar{\boldsymbol{\xi}}$ *be an idealized instance of* $\boldsymbol{\xi}$ *containing* noise-free *measurements corresponding to the motion of a sensor platform that meets the observability requirements of Lemma 5 (i.e., it contains measurements of rotations about two distinct axes and it satisfies Equation* (4.33)*). Then, there exists some* $\epsilon > 0$ *such that if* $\|\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}\| \leq \epsilon$*, then strong duality holds for the instance of Equation* (4.30) *described by* $\boldsymbol{\xi}$*, and the global optimum can be obtained via the solution of the dual problem.*

Proof. We will use Theorem 2, which requires that

1. the cost $c_{\boldsymbol{\xi}}(\mathbf{x}) = \mathbf{x}^\top \mathbf{C}(\boldsymbol{\xi})\mathbf{x}$ varies continuously as a function of $\boldsymbol{\xi}$;
2. $\bar{\boldsymbol{\xi}}$ is such that the cost matrix $\mathbf{C}(\bar{\boldsymbol{\xi}}) \succeq \mathbf{0}$;
3. the minimum value of Equation (4.30) for $\bar{\boldsymbol{\xi}}$ is 0;

4. $\mathcal{Q}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\nu}})$ has corank-1; and

5. the Abadie constraint qualification (ACQ) holds for the algebraic variety described by the constraints of Equation (4.30).

In Section 4.2.4 we demonstrated that the cost function $c_{\boldsymbol{\xi}}(\mathbf{x})$ depends quadratically (and therefore continuously) on $\boldsymbol{\xi}$, satisfying condition 1. The proof of Proposition 2 ensures that 2 and 3 hold.

In order to prove that the Hessian of the Lagrangian has corank-1, recall from Proposition 2 that $\bar{\boldsymbol{\nu}} = \mathbf{0}$ is a valid optimal dual solution for $\bar{\boldsymbol{\xi}}$. Therefore, $\mathcal{Q}_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\nu}}) = \mathbf{C}(\bar{\boldsymbol{\xi}})$. Since we are dealing with a homogenized version of the cost function in Equation (4.5), we can express the Hessian as

$$\mathbf{C}(\bar{\boldsymbol{\xi}}) = \begin{bmatrix} \mathbf{Q} & \mathbf{q} \\ \mathbf{q}^\top & q \end{bmatrix} \in \mathbb{S}^{13}, \tag{4.42}$$

where $\mathbf{Q}$ is the Hessian of the nonhomogeneous cost in Equation (4.5). Applying Lemma 5 tells us that $\operatorname{rank}(\mathbf{Q}) = 12$. Proposition 2 gives $\mathbf{C}(\bar{\boldsymbol{\xi}})\bar{\mathbf{x}} = \mathbf{0}$, so we know that $\mathbf{C}(\bar{\boldsymbol{\xi}})$ is not full rank since $\bar{\mathbf{x}} \neq \mathbf{0}$ because it is primal feasible and therefore contains a vectorized rotation matrix. Since adding rows and columns to a matrix cannot reduce its rank, we obtain

$$12 = \operatorname{rank}(\mathbf{Q}) \leq \operatorname{rank}\left(\mathbf{C}(\bar{\boldsymbol{\xi}})\right) < 13 \tag{4.43}$$

and conclude that $\operatorname{rank}\left(\mathbf{C}(\bar{\boldsymbol{\xi}})\right) = 12$ (i.e., it is corank-1) as desired.

Finally, condition 5 holds because the variety described by O(3) defines a radical ideal (see Lemma 5.1 and Example 5.4 in Cifuentes et al. (2022) for details). □

Theorem 4 demonstrates the existence of a measurement error bound ϵ within which our hand-eye calibration formulation exhibits strong duality, but we leave the computation[3] of this bound as future work. It is worth noting that the column-space requirement in Equation (4.33) for Lemma 5 is primarily a convenience: the main requirement is rotation about two distinct axes (Andreff et al., 2001; Chen, 1991). However, Equation (4.33) ensures that we can use $\bar{\boldsymbol{\nu}} = \mathbf{0}$ as our reference dual solution, and failure to satisfy Equation (4.33) is highly improbable in practice.[4] Additionally, this condition is worth including because of its importance in ensuring a unique solution to the "unknown scale" version of Equation (4.5) in which sensor b is a monocular camera (Wise et al., 2020).

We can use Theorem 3 to obtain the following corollary:

Corollary 5. *The QCQP in Equation (4.5) can be augmented with inequality constraints on the extrinsic translation* $\mathbf{p}$ *while retaining the SDP stability property in Theorem 4.*

[3] Cifuentes et al. (2022) provides a fairly straightforward method for computing one such ϵ on a per-problem instance basis. However, this method usually provides a loose and therefore overly conservative bound, motivating the development of more precise methods in future work.

[4] The range of the first three columns of $\mathbf{M}$ has measure zero in the six dimensional space of values for $\mathbf{p}_{b_i}$ and $\mathbf{p}_{b_j}$.

Proof. Since LICQ holds for all elements of O(3) and orthogonality constraints do not involve the extrinsic translation $\mathbf{p}$, any inequality constraints satisfying LICQ can be added without losing LICQ because of the block diagonal property of rank. Therefore, we can apply Theorem 3 with a proof identical to that of Theorem 4 but substituting LICQ for ACQ. $\square$

Corollary 5 allows us to incorporate useful physical constraints when formulating extrinsic calibration problems. For example, knowledge of an upper bound on the distance between sensors a and b can be encoded in the quadratic constraint

$$\|\mathbf{p}\|^2 \leq d_{\max}^2. \tag{4.44}$$

Similarly, knowledge that sensor b is "above" sensor a with respect to a's z-axis is easily encoded as the linear constraint $p_z \geq 0$. Finally, the reasoning in Corollary 5 can be applied to inequalities on auxiliary variables representing the unknown scale of a monocular camera's translation measurements as studied in Wise et al. (2020) and Wodtko et al. (2021). A practitioner can combine any number of these constraints, provided the Jacobians of all active constraints remain linearly independent at a potential solution.

As a complement to the stability method of Cifuentes et al. (2022), we note that Brynte et al. (2022) have studied SDP relaxations of QCQPs over SO(3) and other quadratic representations of rotations. Brynte et al. (2022) use sophisticated techniques from algebraic geometry to prove that while non-tight instances of problems like hand-eye calibration exist, they are in fact rare in practice.

4.4 Experiments

In this section, we evaluate our algorithm's performance on synthetic and real data. We begin with an empirical analysis of the duality gap and the effect of adding redundant, independent constraints. Synthetic data is then used to compare the accuracy and runtime of our algorithm with the local optimization method in Brookshire and Teller (2013). Finally, we apply our algorithm to real datasets that use various types of cameras and IMUs to produce egomotion estimates.

Throughout this section, we will frequently refer to the level of noise that is added to simulated measurements. We use σ_{t} and σ_{r} to denote the standard deviation of translation in meters and rotation measurements in radians, respectively. To remain consistent with the experiments in Brookshire and Teller (2013), we apply zero-mean Gaussian noise to translations and an Euler angle representation of rotations. We use isotropic covariance matrices $\mathbf{\Sigma}_{\mathrm{t}} = \sigma_{\mathrm{t}}^2\mathbf{I}$ and $\mathbf{\Sigma}_{\mathrm{r}} = \sigma_{\mathrm{r}}^2\mathbf{I}$. The performance metrics we use in this section are the L^2 norms for translation vectors (the Euclidean distance) and rotation matrices (the Frobenius norm). These metrics are used to compare the accuracy of estimates of $\mathbf{X}$ acquired by our algorithm and the local optimization approach.

4.4.1 Simulated Data

To simulate egomotion measurements, we utilize a random smooth path that contains rotations about multiple distinct axes, guaranteeing that $\mathbf{X}$ is observable (Brookshire and Teller, 2013). A sample path is displayed in Figure 4.3. The path is constructed by tracing a circular route over a landscape consisting of random sinusoidal functions in the z-axis in terms of x- and y-axis variables. The two sensor frames of reference are rigidly positioned according to their extrinsic calibration on a virtual vehicle which traverses this smooth path. The global poses of the individual sensors' paths are used to extract exact egomotion poses which are subsequently corrupted with noise.

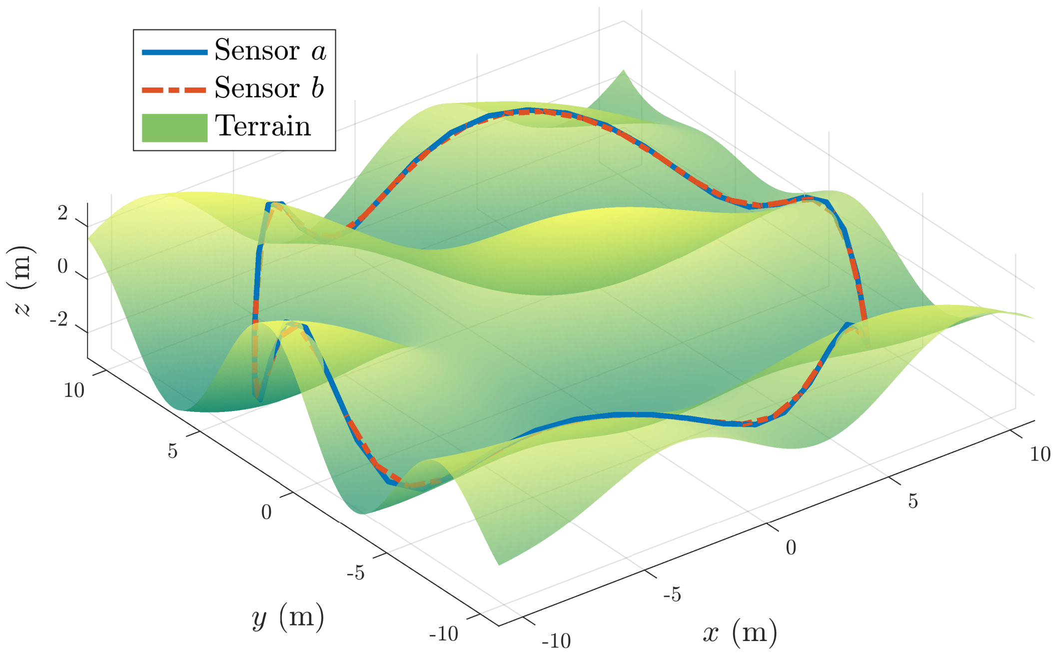


Figure 4.3: Simulated path example. A landscape generated with a mixture of random sinusoids is used to generate paths for sensors a and b, which are rigidly fixed to a virtual vehicle traversing the terrain. A circular path with varying elevation is chosen to ensure observability (in the sense of Lemma 5) of the calibration parameters.

Zero-Duality-Gap and Redundant Constraints

In Section 4.3, we demonstrated that our problem formulation is guaranteed to exhibit strong duality even in the presence of some finite measurement error (i.e., a zero-duality-gap region exists). In this section, we empirically verify this result while also demonstrating that adding handedness and redundant orthogonality constraints increases the size of the ZDG region. While the exact amount of measurement error the problem can tolerate is difficult to determine for our problem's high-dimensional measurement data, a theorem in Cifuentes et al. (2022) provides means of computing a lower bound. We leave analysis of these lower bounds as future work, and instead focus on experimental insights here.

In order to study the effect of adding constraints to the dual formulation, we formulated a simple observable problem instance: a rotation of $\frac{\pi}{2}$ radians and a translation of 1 meter about the vehicle's x-axis, followed by the same manoeuvre about the y-axis. Since our proof in Section 4.3 guarantees a ZDG region for even the minimally constrained case (i.e., only row orthogonality constraints), we expect adding constraints will improve the tolerance to measurement error in a manner consistent with Briales et al. (2017). In order to keep the error injection process simple, we only perturbed one of the rotation measurements for each trial. The top bar graph in Figure 4.4 summarizes the effect of the magnitude of this rotation perturbation (i.e., the magnitude of the rotation angle in an axis-angle description of the perturbing rotation matrix). Each individual bar represents the percentage of 100 uniformly sampled axis directions that a given set of constraints returned for a perturbation magnitude indicated by the x-axis groupings. As the legend in Figure 4.4 indicates, all constraint sets used row orthogonality (labelled R) and some combination of column orthogonality (C) and handedness (H). For all rotation noise levels tested, the handedness-augmented constraint sets (R+H and R+C+H in Figure 4.4) are able to recover an exact minimizer of the primal problem (i.e., strong duality holds), whereas the smaller constraint sets do not guarantee a zero-duality-gap problem instance. The redundant column orthogonality constraints (R+C) appear to improve the performance of the default O(3) constraints (R), but this benefit appears to be subsumed by the handedness constraints for all instances tested.

In the bottom bar graph of Figure 4.4, the rotation perturbation magnitude was fixed to $\frac{\pi}{2}$ and a similar translation perturbation scheme was introduced for one of the measurements in our minimally observable problem. Since both rotation and translation directions needed to be sampled at a sufficient resolution, each bar represents 256 trials. Once severe translation error (10 m) is introduced, the benefit of including the complete, redundant constraint set (R+C+H) becomes clear.

Robustness to Noise

Figure 4.5 compares the performance of our method with the local optimization approach in Brookshire and Teller (2013) for varying values of σ_{t} and σ_{r}. Each pair of subplots represents 100 random trials on simulated data. When σ_{t} and σ_{r} are low, the calibration results are similar for both methods. Our approach is consistently more accurate than the local optimization approach, especially as the noise becomes extreme. This regime of noise is relevant for applications with low cost, noisy sensors or environments that produce challenging conditions (e.g. an urban canopy causing intermittent GPS readings or low-texture surfaces for camera-based egomotion estimation).

We speculate that these results are due to local minima in the high-dimensional cost function of the approach in Brookshire and Teller (2013). Since that method uses relative pose measurements from one of the sensors as initial guesses for their corresponding variable in the optimization, noisy measurements mean that many of these initializations will be inaccurate.

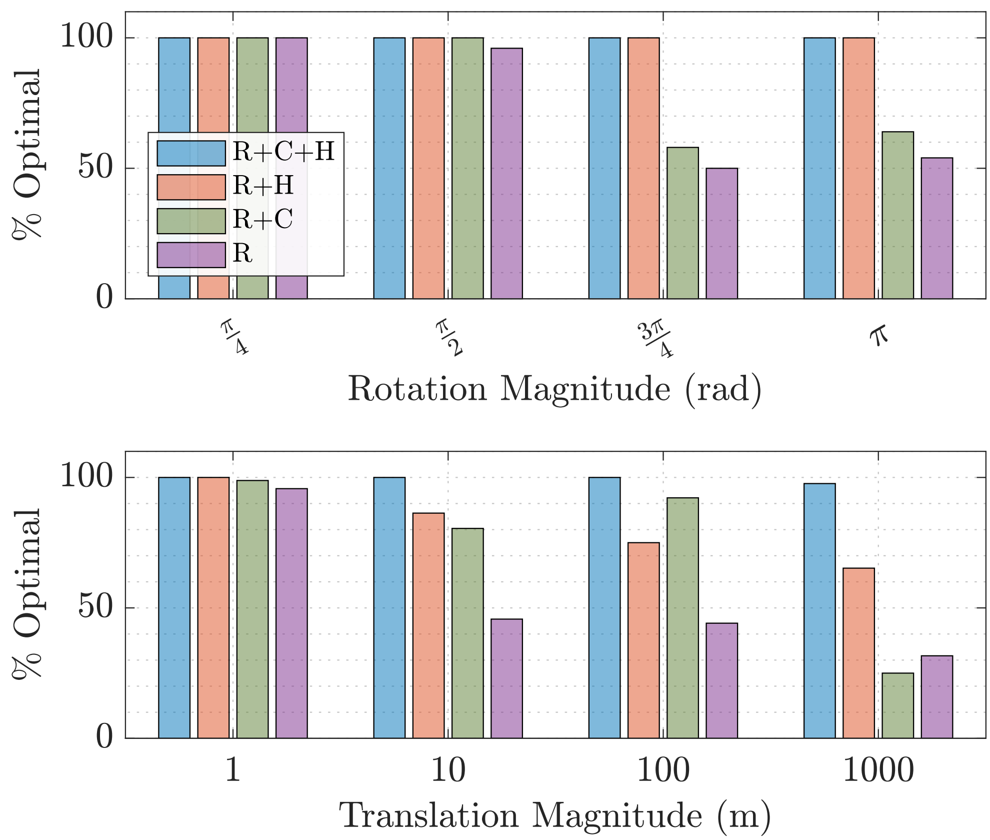


Figure 4.4: Performance of different O(3) and SO(3) constraints on a two-measurement simulated problem instance. All results used row orthogonality constraints at a minimum (R), with redundant column orthogonality (C) and right-handedness constraints (H) being added as well. In the top chart, each rotation magnitude (x-axis grouping) was used to create perturbations about 100 uniformly sampled axes of rotation. The bars indicate the percentage of certified globally optimal solutions found at a particular perturbation magnitude. The bottom chart has a fixed rotation perturbation magnitude of $\frac{\pi}{2}$ but also introduces a perturbation to one of the translation measurements over a total of 256 trials.

Our algorithm avoids this problem by treating all measurements as data instead of variables, and by solving and certifying the global optimality of a convex realization of the problem.

Initialization

For certain initial estimates of the extrinsic calibration $\mathbf{X}$, the local optimization can also converge to a local minimum, even under mild noise conditions. In Figure 4.6 we have plotted a heat map of the relative performance of the local approach and our global convex approach. Each cell corresponds to the maximum difference in the translation (left) or rotation (right) error of the two algorithms' estimates of $\mathbf{X}$ over a uniform sampling of an initial guess for $\mathbf{X}$. Higher values indicate larger error for the local approach. The x-axis varies the magnitude of the initial rotation guess angle in an axis angle form, while the y-axis varies the magnitude of the initial translation guess. As the distance from the true calibration parameters increases,

Figure 4.5: Histograms of translational and rotational L^2 error for various noise levels. The problem formulation and local solver from Brookshire and Teller (2013) is shown in red, while our convex approach is in blue. Each subplot is a histogram showing the distribution of either translation or rotation error in the estimate of $\mathbf{X}$ over 100 random trials. Rotation noise increases from left to right while translation noise increases from top to bottom. The performance of the local solver degrades much more rapidly as the noise levels increase. Note that rotation error is defined as the Frobenius norm of a rotation matrix and is therefore dimensionless.

the accuracy of the local minima degrade. In contrast, the convex formulation converges for any initial conditions.

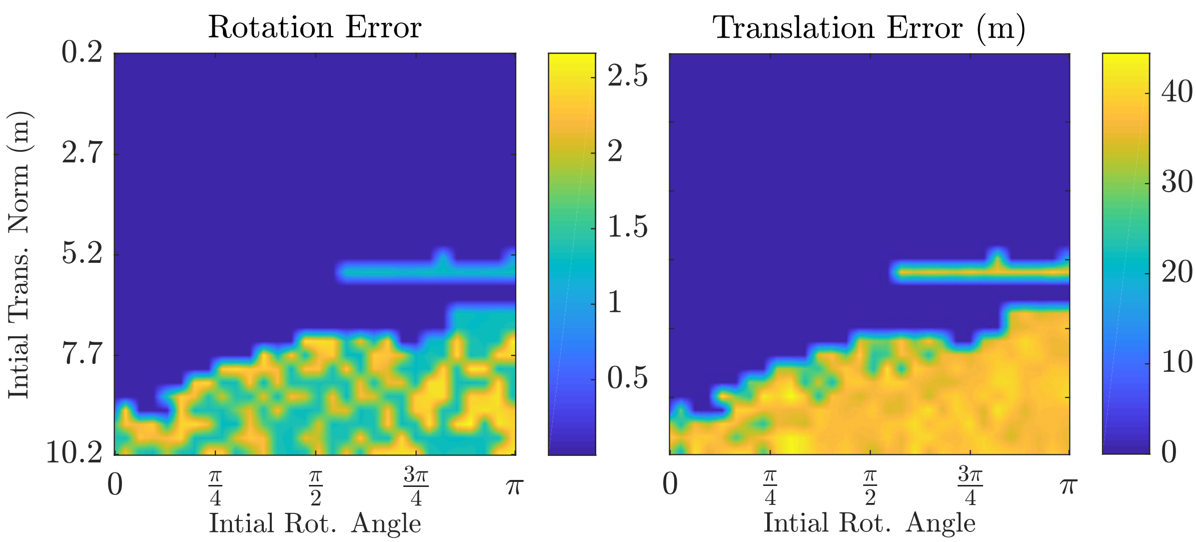


Figure 4.6: Heatmaps displaying the difference in translation and rotation calibration error between the local approach and our convex algorithm for a randomly generated path. For each cell, 64 different initial values for $\mathbf{X}$ were used to seed the local optimization. Each of these initial values had the same translation and and rotation angle magnitude, as indicated by the cell's position on the axes. The yellow colour plotted indicates higher error in the local optimization approach.

Runtime

Since our problem formulation fixes the number of optimization variables, solver runtime is independent of the amount of data available. In contrast, the MLE approach in Brookshire and Teller (2013) treats every relative motion as an optimization variable. We performed a runtime comparison of both methods on a desktop with an Intel Core I7-7820X 3.6 GHz CPU. Since both algorithms require data setup that is slow when naively implemented in MATLAB, we only compare the runtime of the solvers. Our approach uses the general-purpose SDPT3 algorithm (Toh et al., 1999) provided by the CVX package (Grant et al., 2008). The local approach in Brookshire and Teller (2013) uses a custom Levenberg-Marquardt solver.

In Figure 4.7, the mean runtime along with 1st and 3rd quartile bars from 100 random runs is plotted for both methods. The local optimization approach's runtime quickly grows as more measurements are added, while our approach is able to solve any problem instance in under half a second on average. The trend appears to continue for larger datasets, as the local approach did not converge after waiting several hours for a dataset of over 1000 poses, which our solver was able to easily handle.

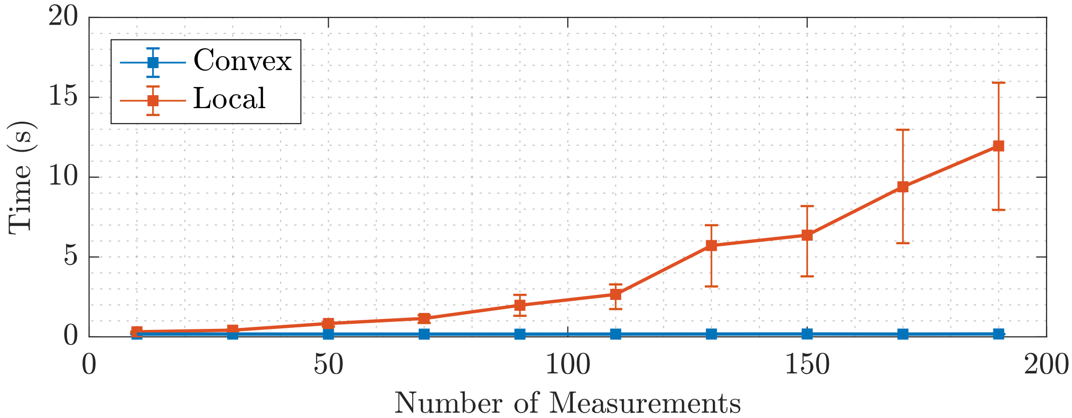


Figure 4.7: Mean solver runtime with 1st and 3rd quartile bars over 20 random runs per data point. The number of relative pose measurements in the problem data used is varied, exhibiting poor scaling for the local optimization approach and virtually constant solver time for our convex approach.

4.4.2 RGB-D Data

In this section, we use experimental data from Brookshire and Teller (2013), obtained with Xtion RGB-D sensors. For details about data collection, the extraction of relative pose measurements, and ground truth, see Brookshire and Teller (2013). In Table 4.1 we compare the error with respect to ground truth of our algorithm and the local optimization approach. Both algorithms are able to acquire accurate extrinsic calibrations. We also include the mean runtime of 100 applications of each solver on a laptop computer with an Intel Core i5-5287U 2.90 GHz CPU. For this dataset, we were able to find initial parameters that trap the local optimization method in local minima in a pattern similar to Figure 4.6, but only for initial parameters that are tens of meters away from the true value. Nevertheless, our method is much faster, and simulation results indicate that local minima arise for realistic datasets with different noise levels and types of measurements. Furthermore, an exhaustive set of experiments is impossible for the uncountably infinite number of problem instances, meaning our formal optimality guarantees are a valuable tool and safer choice.

4.4.3 Starry Night Dataset

The "Starry Night" dataset consists of stereo vision and pre-processed inertial measurement unit (IMU) readings from an environment with static landmarks (Furgale et al., 2012). The dataset also contains Vicon motion capture measurements of the sensor rig's motion and a groundtruth value of $\mathbf{X}$. We used the dataset to produce incremental egomotion estimates for the stereo camera, IMU, and a second IMU trajectory estimate using Vicon measurements of reflective markers placed on the sensor rig. See Figure 4.8 for a photograph of the sensor rig.

Table 4.1 summarizes the results of a comparison between our algorithm and the local method of Brookshire and Teller (2013). The estimate of $\mathbf{X}$ that uses Vicon and stereo camera measurements is labelled "Vicon", and both algorithms are able to obtain a fairly accurate

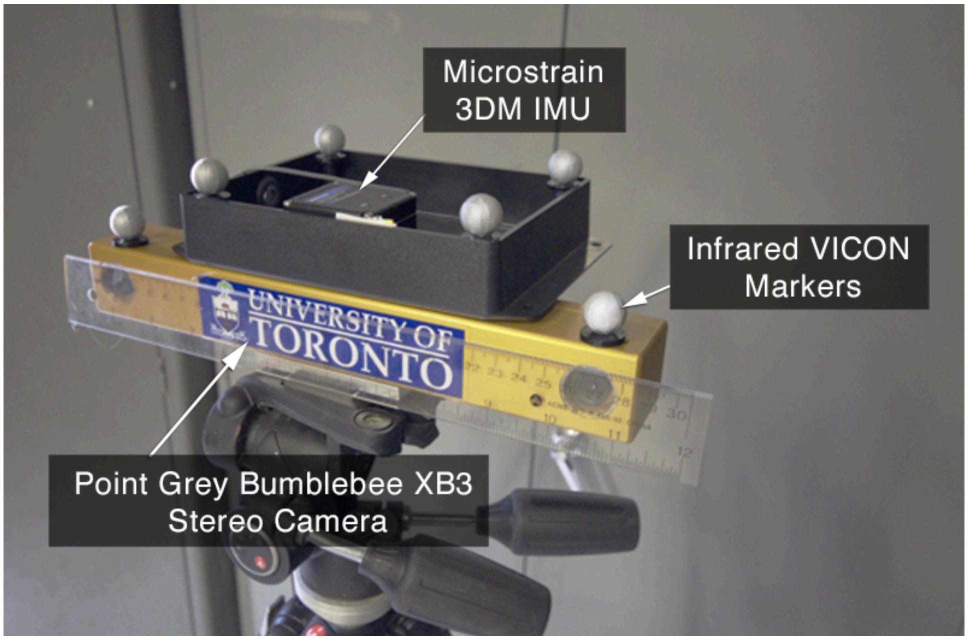


Figure 4.8: The sensor rig used in the "Starry Night" dataset. The IMU data reports translational and rotational velocities, while the stereo camera observes static point landmarks with known positions. *Image courtesy of Professor Timothy Barfoot.*

estimate from this data. However, our algorithm is once again orders of magnitude faster. The results labelled "IMU-5" compare integrated IMU measurements and camera localization on all poses where at least 5 out of 20 possible landmarks are visible to the stereo camera, ensuring accurate estimation. Even though the IMU measurements have been sanitized to remove biases, the velocity measurements are still very noisy and lead to large drift in the trajectory estimate. This is reflected in the results: both algorithms are able to recover a reasonable estimate of the extrinsic orientation, but neither manages to accurately estimate the relative position. The local approach is similarly far slower with this data. In order to study the accuracy of the algorithms on a number of poses where the local optimization approach has a competitive runtime, we downsampled the IMU and camera data to only those poses where the camera is able to see 15 landmarks. These results are labelled "IMU-15", and neither algorithm is able to obtain an accurate estimate, but the local approach's result is particularly erroneous, which indicates that it may be trapped in a local minimum. Our globally optimal approach's rotational error is not egregiously large, and its fast runtime means it could be suitable for bootstrapping another algorithm that uses more data than relative poses but needs an initialization within some basin of convergence.

4.5 Summary and Future Work

In this chapter, we leveraged state-of-the-art certifiably globally optimal solution methods for QCQPs to create a novel, general purpose extrinsic calibration algorithm. Our method is faster

Table 4.1: Extrinsic Calibration Results for Real Data.

Dataset	**Method**	**L^2 Error**		**Runtime** (s)
		Trans. (cm)	Rot.	
RGB-D	Local	1.2	0.026	4.80
	Convex	1.3	0.026	0.31
Vicon	Local	2.6	0.021	168.2
	Convex	3.4	0.020	0.33
IMU-5	Local	16.5	0.14	133.4
	Convex	21.3	0.17	0.40
IMU-15	Local	709.7	2.30	0.90
	Convex	97.2	0.63	0.38

than previous approaches, but more important is its avoidance of convergence to local minima that other methods fall prey to, even in the presence of severe noise. Our algorithm's beneficial properties were demonstrated through a range of experiments involving simulated and real data, which we intend to compare with a greater variety of algorithms in future work. Conveniently, we proved that observability of the calibration parameters is sufficient for guaranteeing a ZDG region around noise-free observations, and demonstrated further empirical evidence in support of the theoretical analysis in Brynte et al. (2022). We also believe that the effect of redundant constraints can be more precisely characterized using the theory developed in Cifuentes et al. (2022).

The techniques used here can also be extended to the calibration of other sensor configurations. Monocular cameras are particularly interesting because they are only able to provide egomotion estimates up to an unknown scale (Wise et al., 2020). In addition, the extrinsic calibration of multiple inertial measurement units could also benefit from the globally optimal properties of a QCQP formulation. We would also like to incorporate *non-isotropic* measurement covariances into an MLE formulation that is still a QCQP, since this information, when available, can be extremely valuable to an estimator. Finally, a further avenue for future work is to perform a comparison of SO(3) representations that admit a QCQP formulation of the problem (e.g., comparing rotation matrices with unit quaternions and other representations explored in Heller et al. (2014)).

Chapter 5

Planar and Spherical Inverse Kinematics

> To find the answer, you must know the answer.
>
> Michael P. Collins,
> *CIV102: Structures and Materials*

This chapter explores the use of SDP relaxations for solving planar and spherical inverse kinematics problems with symmetric joint limits and obstacle constraints. While this may seem orthogonal to the calibration problem studied in Chapter 4, the theoretical machinery we developed in Chapter 3 reveals that these problems are not quite as unrelated as they appear at first glance. Most notably, we take inspiration from noisy estimation problems like hand-eye calibration to develop a "nearest point" formulation of inverse kinematics. This approach acts as a bridge from the techniques in applied in Chapter 4 to the more challenging revolute inverse kinematics problems in Chapter 6.

Many common robots (e.g., manipulator arms and snake-like robots) can be modelled as *kinematic chains*: rigid bodies connected by revolute joints that constrain robot motion to a specific workspace. The motion of these robots may also be constrained by joint limits or user and task-specified workspace constraints on the positions or orientations of links. Planning and controlling motion therefore requires solving the *inverse kinematics* (IK) problem: finding configurations of the kinematic chain that satisfy a set of kinematic constraints. A wide variety of techniques have been developed with the goal of solving IK for specific types of kinematic chains, such as manipulators with up to six degrees of freedom (DOFs). However, generic solvers primarily rely on nonlinear optimization techniques, which typically solve the problem locally around an initial "seed" configuration. Owing to their local nature, these solvers cannot guarantee that a feasible solution will be found, and as such may lead to the false conclusion that a problem is infeasible.

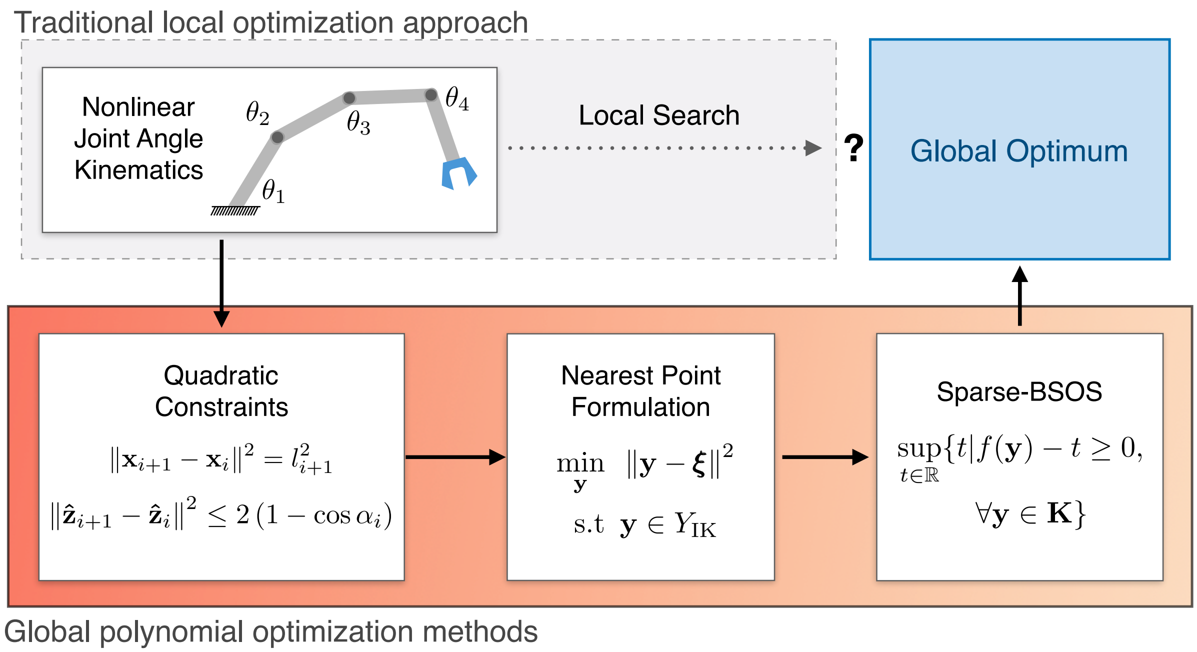


Figure 5.1: System diagram summarizing our approach. We parameterize manipulator kinematics with joint locations, leading to a QCQP problem formulation. This allows us to leverage polynomial optimization techniques to obtain certifiably globally optimal solutions or certificates of infeasibility.

Kinematic chains are often parametrized using joint angles as variables, generating an IK problem comprised of nonlinear trigonometric equations. However, alternative parameterizations exist that result in the kinematic equations taking on forms suitable for solution with a wider variety of mathematical tools. Porta et al. (2005b) show that IK can be formulated as a special case of the *Distance Geometry Problem* (DGP) (Dattorro, 2005), which consists of finding points that satisfy a given set of assigned distances. Semidefinite programming (SDP) and sum-of-squares (SOS) relaxations are convex optimization techniques that have been used to solve the DGP in the domains of sensor network localization (SNL) (So and Ye, 2007; Nie, 2009) and protein folding (Alipanahi et al., 2013). In this chapter, we demonstrate that the DGP in Porta et al. (2005b) can be represented as a quadratically constrained quadratic program (QCQP), which can be extended to include other constraints such as joint limits. The main contributions of this chapter are:

1. a polynomial formulation of IK with joint limit constraints, which admits provably tight SDP relaxations for problem instances which meet a criterion we characterize;

2. a fast solution method for our formulation that uses a sparse SOS solver; and

3. an open source implementation and experimental analysis of our algorithm in MATLAB.[1]

[1] See `https://github.com/utiasSTARS/sos-ik` for our code.

5.1 Related Work

In this section, we review the two fields at whose intersection this chapter lies: IK and global polynomial optimization.

5.1.1 Inverse Kinematics

Owing to its widespread use, IK is a subject of intense research with an abundance of relevant literature that we only briefly summarize here; see Aristidou et al. (2018) for a recent, in-depth survey. Classical theoretical results show that a general 6-DOF spatial kinematic chain has up to 16 configurations corresponding to any feasible end-effector pose (Lee and Liang, 1988). In such cases, closed form solutions can be found analytically using various parametrizations (Manocha and Canny, 1994; Husty et al., 2007; Qiao et al., 2010), and software libraries such as *IKFast* (Diankov, 2010) can be used to rapidly generate feasible configurations. One downside of these approaches is that they only account for other kinematic constraints (such as joint limits) as a post-processing step. Moreover, for spatial (planar) kinematic chains with more than six (three) DOFs, there exist an infinite number of solutions for any given end-effector pose, which means the state space for the redundant DOFs needs to be enumerated in a discrete and computationally expensive fashion.

When solutions cannot be obtained analytically, numerical methods are often used. So-called closed-loop IK (CLIK)[1] techniques use the Jacobian's (pseudo)inverse to apply differential kinematics in a closed-loop fashion, viewing IK as a feedback control problem (Sciavicco and Siciliano, 1986). Moreover, CLIK methods using variants of damped least squares (Buss and Kim, 2005) and null-space optimization (Nakamura et al., 1987) provide numerically stable redundancy resolution for multiple hierarchical criteria. Nonconvex nonlinear optimization techniques such as sequential quadratic programming (Schulman et al., 2014) iteratively generate convex subproblems, which can be efficiently solved. These numerical methods do not provide any global optimality or feasibility guarantees, however, and therefore may require a large number of initializations to retrieve a feasible solution. Dai et al. (2019) use a piecewise-convex relaxation of the SO(3) group to formulate the constrained IK problem as a mixed integer linear program (MILP). They show that, unlike local optimization, their method requires no initialization and can provide a global certificate of infeasibility when a solution cannot be found. Without approximating SO(2) or SO(3), our approach utilizes the theoretical machinery developed in Chapter 3 to derive a tight SOS relaxation of IK for planar and spatial spherical kinematic chains, while retaining the ability to certify the global infeasibility of the problem. Moreover, our SOS relaxation leverages the innate sparsity pattern of the kinematic model to efficiently scale to a high number of DOFs. Finally, in Blanchini et al. (2015) and

[1] These techniques are not to be confused with IK methods for *parallel mechanisms* whose physical structure contain "closed loops".

Blanchini et al. (2017), the authors analyze a convex formulation of inverse kinematics that is similar to our planar case but uses a linear cost function and relaxation.

5.1.2 Global Polynomial Optimization

Optimization problems with polynomial cost and constraints are amenable to a host of convex relaxations that provide globally optimal solutions or bounds on the minimum cost. These methods include SDP relaxations for QCQPs (Ma, 2010), as well as the broader class of global polynomial optimization techniques known as SOS programming (Parrilo, 2003; Lasserre, 2001). While SOS programming technically involves solving an SDP, we will reserve the term "SDP relaxation" for relaxations of the form described in Chapter 3. Convex relaxations of polynomial optimization problems have found success in applications spanning signal processing, finance, control theory, and state estimation (Boyd and Vandenberghe, 2004; Lasserre, 2010; Olsson and Eriksson, 2008).

The literature on convex relaxations for SNL is closely related to the techniques developed in this chapter. Most work focuses on the performance of SDP formulations for localization problems where noisy measurements of inter-sensor distances are provided (Biswas et al., 2006). In this chapter, we seek to solve inverse kinematics, where "measurements" are the distances between points of the kinematic chain. Thus, our approach is more closely related to the analyses of the noise free SNL problem found in So and Ye (2007) and Nie (2009), which demonstrate tightness of SDP and SOS relaxations, respectively. However, our work employs joint angle limits unique to robotics problems such as manipulation and uses a novel nearest point formulation that has theoretical guarantees for relaxation tightness (Cifuentes et al., 2022).

Convex relaxations of polynomial optimization problems have been utilized in a variety of planning and control algorithms. In Deits and Tedrake (2015a), convex obstacle-free regions of space in cluttered environments are efficiently generated via two alternating optimizations. This approach is used in Deits and Tedrake (2015b) and Landry et al. (2016) to create SOS constraints enforcing collision-free trajectories in a mixed-integer planning approach to quadrotor flight. In Paden et al. (2017), admissible heuristics for kinodynamic path planning problems are constructed with a SOS approximation method. In Jasour et al. (2015), chance-constrained formulations of optimization problems are introduced and solved via SOS programming. These methods are applied to problems in robotic motion planning and control to design trajectories with bounded collision probabilities (Jasour and Lagoa, 2016; Jasour et al., 2018). Our algorithm, while not solving the entirety of a path planning or control problem, is complementary to these works and holds promise as a means of extending various planning methods to complex, high-dimensional kinematic models. The closest work to ours is Trutman et al. (2022), where SOS programming is used for globally optimal solutions to IK for 7-DOF revolute manipulators. The approach in Trutman et al. (2022) has stronger formal guarantees than ours, but it

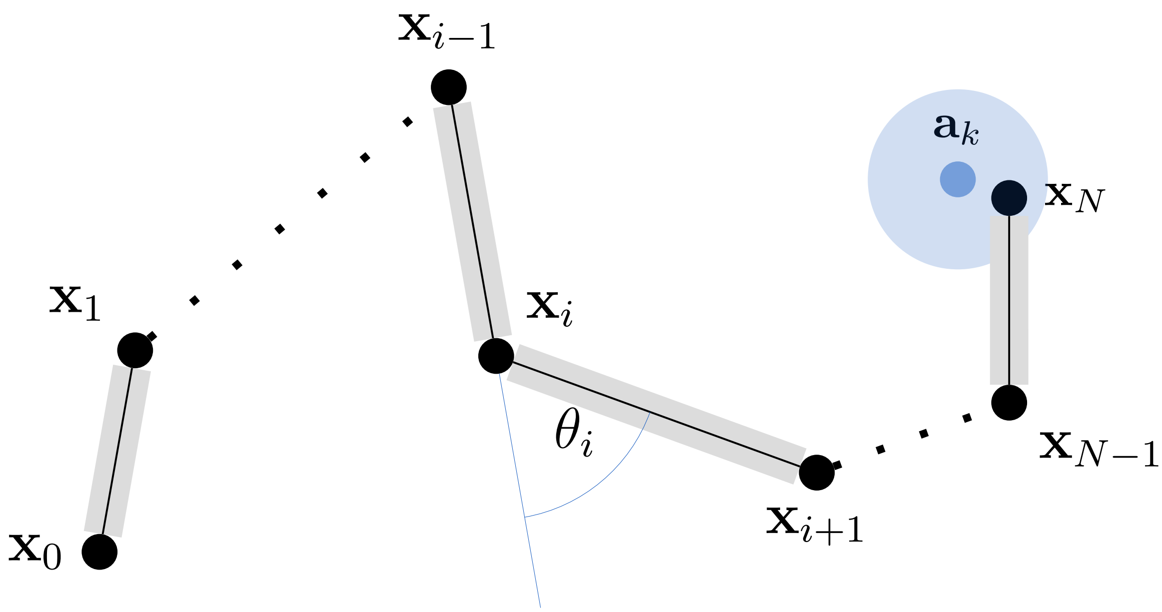


Figure 5.2: A kinematic chain comprised of $N-1$ spherical joints, and a virtual joint at the endpoint. Note that spherical joints in two dimensions correspond to revolute joints with a single rotation axis. The dotted lines represent links which are not shown. All constraints imposed on the joint $\mathbf{x}_i$ are a function of its position and the positions of nearby joints. The vertex $\mathbf{x}_N$ representing the endpoint is constrained to a ball around the point $\mathbf{a}_k$.

is tailored to a different class of manipulators and has a runtime on the order of seconds, which is prohibitive for real-time applications.

5.2 Kinematic Model

In this section, we build on Porta et al. (2005b) to devise a model of the kinematic chain for which the IK problem can be formulated as a QCQP. The resulting QCQP admits an SDP relaxation, for which we provide sufficient conditions for tightness in Section 5.3.3. We also comment on the sparsity pattern of the constraints, which allows us to derive an efficient sparse SOS relaxation (see Section 5.4.2 for details). For clarity, we restrict our analysis to planar and spatial serial kinematic chains with N spherical joints connected by N straight rigid links. However, the model presented here admits complex kinematic chains that include industrial and parallel manipulators (Porta et al., 2005b).

We begin by representing key points in the kinematic chain as vertices of a graph embedded in $\mathbb{R}^d$, where $d \in \{2, 3\}$. In Figure 5.2, we can see that each joint of the chain is represented by a vertex $\mathbf{x}_i \in \mathbb{R}^d, i = 1, 2, \dots, N$, with the vertices $\mathbf{x}_0$ and $\mathbf{x}_N$ corresponding to the base and the endpoint respectively. Note that the full set of joint angles can be geometrically recovered from this representation.

5.2.1 Distance Constraints

We can restrict the distance between two vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ to some range $D_{ij} = [D_{ij_{min}}, D_{ij_{max}}]$ by introducing the nonconvex quadratic constraint

$$D^2_{ij_{max}} \geq \|\mathbf{x}_i - \mathbf{x}_j\|^2 \geq D^2_{ij_{min}}\,, \tag{5.1}$$

which can be used to model the kinematic chain structure by constraining the distances between joints. We can model rigid links by reducing the range $D_{i,i+1}$ to a single value l_{i+1}, corresponding to the length of the link between two consecutive joints at $\mathbf{x}_{i+1}$ and $\mathbf{x}_i$. This results in an equality constraint, which restricts the distance between two joints to match the length of the link connecting them:

$$\begin{aligned} l^2_{i+1} \geq & \|\mathbf{x}_{i+1} - \mathbf{x}_i\|^2 \geq l^2_{i+1} \\ \Leftrightarrow \quad & \|\mathbf{x}_{i+1} - \mathbf{x}_i\|^2 = l^2_{i+1}. \end{aligned} \tag{5.2}$$

Without additional constraints, the vertex $\mathbf{x}_{i+1}$ is restricted to a sphere of radius l_{i+1} centred at $\mathbf{x}_i$: this corresponds to an unconstrained spherical joint in $\mathbb{R}^d$.

5.2.2 Position Constraints

The distance of any vertex from a fixed point in $\mathbb{R}^d$ (or *anchor*) $\mathbf{a}_k$ can be restricted to the range $D_{ik} = [D_{ik_{min}}, D_{ik_{max}}]$ using

$$D^2_{ik_{max}} \geq \|\mathbf{x}_i - \mathbf{a}_k\|^2 \geq D^2_{ik_{min}}. \tag{5.3}$$

Similarly to Equation (5.2), collapsing the range D_{ik} in Equation (5.3) to zero restricts the position of a vertex $\mathbf{x}_i$ to the point $\mathbf{a}_k$:

$$\begin{aligned} 0 \geq & \|\mathbf{x}_i - \mathbf{a}_k\|^2 \geq 0 \\ \Leftrightarrow \quad & \|\mathbf{x}_i - \mathbf{a}_k\|^2 = 0. \end{aligned} \tag{5.4}$$

This allows us to define the base position by constraining $\mathbf{x}_0$, as well as the exact pose (position and orientation) of the final link by constraining $\mathbf{x}_{N-1}$ and $\mathbf{x}_N$.

5.2.3 Angle Constraints

The angle θ_i of any joint $\mathbf{x}_i$ with respect to its parent joint $\mathbf{x}_{i-1}$ is often restricted by mechanism design. In Figure 5.3, the unit vector

$$\hat{\mathbf{z}}_i \triangleq \frac{1}{l_i}(\mathbf{x}_i - \mathbf{x}_{i-1}) \tag{5.5}$$

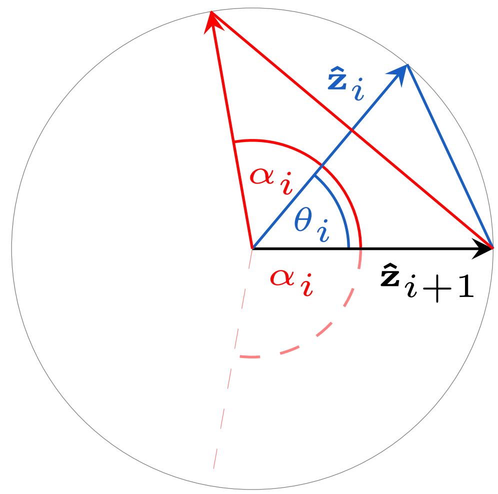


Figure 5.3: Visualization of the convex angle constraint in Equation (5.6). As both the vectors $\hat{\mathbf{z}}_i$ and $\hat{\mathbf{z}}_{i+1}$ are of unit length, the length $\|\hat{\mathbf{z}}_i - \hat{\mathbf{z}}_{i+1}\|$ depends only on the angle between them.

is related to joint angle θ_i and joint angle limit α_i. Applying the cosine law leads to the equivalence

$$\begin{aligned} & |\theta_i| \leq \alpha_i \\ \Leftrightarrow \quad & \|\hat{\mathbf{z}}_{i+1} - \hat{\mathbf{z}}_i\|^2 \leq 2\left(1 - \cos\alpha_i\right), \end{aligned} \tag{5.6}$$

which can be used to enforce joint limit constraints symmetric with respect to the previous link, as shown in Figure 5.2 and Figure 5.3. In Blackmore and Williams (2006), it is noted that quadratic constraints can also be used for nonsymmetric angle ranges smaller than 180°.

Note that the constraints described in this section form a sparsity pattern: the position of any joint $\mathbf{x}_i$ only appears in constraints with nearby joints $\mathbf{x}_{i-k}$ and $\mathbf{x}_{i+k}$ for $k \leq 2$. In Section 5.4.2 we explain how this sparsity can be exploited by an SOS solver to efficiently find IK solutions. This kinematic model can also be extended to include other quadratic constraints such as collision avoidance, which we explore with a more general model in Chapter 6.

5.3 Inverse Kinematics Formulation

In this section, we cast IK as an optimization problem seeking the feasible configuration whose joints are closest to some target positions in the workspace. This *nearest point* formulation of IK allows us to prove Theorem 5, which sheds light on the globally optimal performance of our convex relaxations.

5.3.1 Real Algebraic Variety of Feasible Configurations

The *real algebraic variety*, or, simply, *variety* Y of a set of polynomial equations $f_i(\mathbf{y}) = 0$ is the set of real-valued solutions satisfying those equations:

$$Y \triangleq \{\mathbf{y} \in \mathbb{R}^n : f_1(\mathbf{y}) = \cdots = f_m(\mathbf{y}) = 0\}. \tag{5.7}$$

In order to define the set of all kinematically feasible N-link chains that connect the origin in $\mathbb{R}^d$ to a desired end position $\mathbf{x}_N$ as a variety, we need to express the inequalities representing angle constraints from Section 5.2.3 as equalities. To this end, we introduce N auxiliary variables s_i (Park and Boyd, 2017) and note that any inequality constraints satisfy the equivalence

$$f_i(\mathbf{x}) \leq 0 \iff \exists s_i \ \text{ s.t. } \ f_i(\mathbf{x}) + s_i^2 = 0. \tag{5.8}$$

We write

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{N-1} \end{bmatrix} \in \mathbb{R}^{d(N-1)} \tag{5.9}$$

to denote the concatenation of "interior" joint variables, and $\mathbf{s} \in \mathbb{R}^N$ to denote the column vector of auxiliary s_i variables. We can now define our kinematically feasible set as the variety

$$Y_{\text{IK}} \triangleq \{\mathbf{y} \in \mathbb{R}^n : g_i(\mathbf{x}) = h_i(\mathbf{y}) = 0, \ i = 1, \ldots, N\}, \tag{5.10}$$

where $n = d(N-1) + N$ and $\mathbf{y} = \begin{bmatrix}\mathbf{x}^\top & \mathbf{s}^\top\end{bmatrix}^\top$, and

$$\begin{aligned} g_i(\mathbf{x}) &= \|\mathbf{x}_i - \mathbf{x}_{i-1}\|^2 - l_i^2, \\ h_i(\mathbf{y}) &= \|\hat{\mathbf{z}}_{i+1} - \hat{\mathbf{z}}_i\|^2 + s_i^2 - 2\left(1 - \cos\alpha_i\right) \\ &= \left\| \frac{1}{l_{i+1}}(\mathbf{x}_{i+1} - \mathbf{x}_i) - \frac{1}{l_i}(\mathbf{x}_i - \mathbf{x}_{i-1}) \right\|^2 + s_i^2 - 2\left(1 - \cos\alpha_i\right). \end{aligned} \tag{5.11}$$

To summarize, the variety Y_{IK} is the feasible set for a particular instance of IK parameterized by the number of DOFs N, the link lengths l_i, the angle limits α_i, and the target pose of the final link $\mathbf{x}_N$. Furthermore, Equation (5.11) demonstrates that Y_{IK} is in fact a quadratic variety. Lastly, our formulation assumes, without loss of generality, that $\mathbf{x}_0 = \mathbf{0}$.

5.3.2 Nearest Point Problem

For redundant manipulators, Y_{IK} contains infinitely many solutions for nearly all target positions $\mathbf{x}_N$. The set described by variety Y_{IK} is both high-dimensional and nonconvex. In order to find solutions, we will cast IK as the problem of finding the *nearest point* $\mathbf{y} \in Y_{\text{IK}}$ to some reference point $\boldsymbol{\xi} \in \mathbb{R}^n$. Since the squared Euclidean distance is used for the cost, and Y_{IK} is a quadratic variety, this allows us to cast IK as the QCQP

$$\begin{aligned} \min_{\mathbf{y}} \quad & \|\mathbf{y} - \boldsymbol{\xi}\|^2 \\ \text{s.t} \quad & \mathbf{y} \in Y_{\text{IK}}, \end{aligned} \tag{5.12}$$

where $\boldsymbol{\xi} = \begin{bmatrix} \mathbf{x}_0{}^\top & \mathbf{s}_0{}^\top \end{bmatrix}^\top \in \mathbb{R}^{d(N-1)+N}$. Note that the cost also includes the squared distance between the auxiliary variables s_i and their reference points $s_{0,i}$ in $\boldsymbol{\xi}$. Since we have massaged our problem into the familiar form of Equation (3.1), we can now once again apply the powerful machinery of Chapter 3.

5.3.3 Strong Duality of Nearest Point IK

In this section, we prove that for many instances of the problem in Equation (5.12), the convex SDP relaxation is tight, and therefore we can find a global optimum of Equation (5.12) in polynomial time with interior point solvers. Since our formulation transformed inequality constraints into equivalent equality constraints, we can use a specialized form of Theorem 2:

Corollary 6 (Nearest Point to a Quadratic Variety (Cifuentes et al., 2022))**.** *Consider the problem*

$$\min_{\mathbf{y} \in Y} \|\mathbf{y} - \boldsymbol{\xi}\|^2, \tag{5.13}$$

where $Y \triangleq \{\mathbf{y} \in \mathbb{R}^n : f_1(\mathbf{y}) = \cdots = f_m(\mathbf{y}) = 0\}$*, and* f_i *are all quadratic polynomials. Let* $\bar{\boldsymbol{\xi}} \in Y$ *be such that* $\text{ACQ}_Y\left(\bar{\boldsymbol{\xi}}\right)$*. Then there is zero-duality-gap for any* $\boldsymbol{\xi} \in \mathbb{R}^n$ *that is sufficiently close to* $\bar{\boldsymbol{\xi}}$*.*

Corollary 6 uses the properties of a nearest point cost function to simplify the requirements for SDP stability. We can now state and prove the following theorem:

Theorem 5 (Strong Duality for IK)**.** *If* $\bar{\boldsymbol{\xi}} \in Y_{IK}$ *does not represent a fully extended configuration (i.e., the joint positions are not all collinear), and does not have any joints at their angular limits for specified base and goal positions, then Equation* (5.12) *exhibits strong duality for all* $\boldsymbol{\xi}$ *sufficiently close to* $\bar{\boldsymbol{\xi}}$*.*

Proof. Recalling the definition of $\text{ACQ}_Y\left(\bar{\boldsymbol{\xi}}\right)$ from Section 3.5.2, it is sufficient to show that

$$\operatorname{rank}\left(\nabla f(\bar{\boldsymbol{\xi}})\right) = n - \dim_{\bar{\boldsymbol{\xi}}}(Y_{\text{IK}}). \tag{5.14}$$

The number of variables $n = d(N-1) + N$ scales with dimension $d \in \{2, 3\}$ and the number of links N. Theorem 1.7 in Milgram et al. (2004) gives us $\dim_{\bar{\boldsymbol{\xi}}}(Y_{\text{IK}}) = (d-1)(N-1) - 1$. Therefore, we need to show that

$$\begin{aligned} \operatorname{rank}(\nabla f(\bar{\boldsymbol{\xi}})) &= d(N-1) + N - (d-1)(N-1) + 1 \\ &= 2N. \end{aligned}$$

The structure of the Jacobian matrix $\nabla f(\bar{\boldsymbol{\xi}}) \in \mathbb{R}^{2N \times (d(N-1)+N)}$ can be understood in terms of N link length constraints representing the first N rows, and N joint limit constraints representing the final N rows:

$$\nabla f(\bar{\boldsymbol{\xi}}) = \begin{bmatrix} \mathbf{J}_{1,1} & \mathbf{0}_{N \times N} \\ \mathbf{J}_{2,1} & \mathbf{J}_{2,2} \end{bmatrix}. \tag{5.15}$$

The block lower triangular structure is due to the independence of link length constraints on $\mathbf{s}$. Since $\operatorname{rank}\left(\nabla f(\bar{\boldsymbol{\xi}})\right) \geq \operatorname{rank}(\mathbf{J}_{1,1}) + \operatorname{rank}(\mathbf{J}_{2,2})$ for block lower triangular matrices, it is sufficient to demonstrate that $\operatorname{rank}(\mathbf{J}_{1,1}) = \operatorname{rank}(\mathbf{J}_{2,2}) = N$. Since $\mathbf{J}_{2,2} = \operatorname{diag}(2\mathbf{s})$, and $s_i > 0 \; \forall \, i \in \{1, \ldots, N\}$ by our assumption that no joint is at its angular limit, $\operatorname{rank}(\mathbf{J}_{2,2}) = N$. It remains to demonstrate that

$$\mathbf{J}_{1,1}^{\top} = 2 \begin{bmatrix} \mathbf{x}_1 - \mathbf{x}_0 & \mathbf{x}_1 - \mathbf{x}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{x}_2 - \mathbf{x}_1 & \mathbf{x}_2 - \mathbf{x}_3 & \mathbf{0} & \vdots \\ \vdots & \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{x}_{N-1} - \mathbf{x}_{N-2} & \mathbf{x}_{N-1} - \mathbf{x}_N \end{bmatrix} \in \mathbb{R}^{d(N-1) \times N} \tag{5.16}$$

has (full) rank N. Suppose $\operatorname{rank}(\mathbf{J}_{1,1}) \neq N$. Then there exists $\mathbf{v} \in \mathbb{R}^N$ such that $\mathbf{v} \neq \mathbf{0}$ and $\mathbf{J}_{1,1}^{\top} \mathbf{v} = \mathbf{0}$. This implies that

$$v_i(\mathbf{x}_i - \mathbf{x}_{i-1}) = v_{i+1}(\mathbf{x}_{i+1} - \mathbf{x}_i), \quad \forall i = 1, \ldots, N-1. \tag{5.17}$$

Since there exists some i such that $v_i \neq 0$, and the link lengths $l_i = \|\mathbf{x}_i - \mathbf{x}_{i-1}\|$ are all greater than zero, Equation (5.17) tells us that $v_i \neq 0$ for all i. Therefore, $\mathbf{x}_i - \mathbf{x}_{i-1} = c_{ij}(\mathbf{x}_j - \mathbf{x}_{j-1})$ for all valid pairs of i, j, where

$$c_{ij} = \frac{v_j}{v_i} \neq 0. \tag{5.18}$$

In other words, the link orientations are all collinear, which contradicts the assumption that the arm is not fully extended. Therefore, $\operatorname{rank}(\mathbf{J}_{1,1}) = N$ and $\operatorname{rank}\left(\nabla f(\bar{\boldsymbol{\xi}})\right) = 2N$, completing the proof.

$\square$

In Chapter 4, our proof for SDP tightness was linked to the observability criteria for hand-eye calibration. The analogous requirement for IK in Theorem 5 is that the stable configuration $\bar{\boldsymbol{\xi}}$ is not "fully extended", i.e., it is not a kinematic singularity (Siciliano et al., 2009). Since having access to an admissible $\bar{\boldsymbol{\xi}} \in Y_{\mathrm{IK}}$ amounts to having solved the IK problem already, we would like to be able to use an SDP relaxation of Equation (5.12) to obtain a valid solution by using a reference point $\boldsymbol{\xi} \notin Y_{\mathrm{IK}}$. In general, most nonconvex problems do not exhibit strong duality. Equation (5.12) contains non-convex link length constraints, making the existence of tight SDP relaxations a non-trivial and useful property. In Section 5.5, we empirically demonstrate that the volume of the tight-relaxation region for instances of Equation (5.12) is substantial and randomly sampling $\boldsymbol{\xi}$ is a practical strategy.

5.3.4 Incorporating Explicit Inequality Constraints

Thus far, we have relied on the equality-constrained machinery in Cifuentes et al. (2022) to analyze our IK formulation. However, Theorem 3 lets us work with inequality constraints directly, removing the need to introduce unwieldy auxiliary variables s_i.

Corollary 7 (Strong Duality for IK With Inequalities)**.** *Let Y'_{IK} be a semialgebraic set constructed from the variety Y_{IK} by replacing constraints involving auxiliary variables s_i with inequalities. Then, if* $\mathrm{LICQ}\left(\bar{\boldsymbol{\xi}}\right)$ *holds for some $\bar{\boldsymbol{\xi}} \in Y'_{IK}$, the nearest point problem*

$$\min_{\mathbf{y} \in Y'_{IK}} \|\mathbf{y} - \boldsymbol{\xi}\|^2 \tag{5.19}$$

exhibits strong duality in the neighbourhood of $\bar{\boldsymbol{\xi}}$ (i.e., it exhibits SDP stability).

Problem 6 (SDP Stability for a Planar Manipulator)**.** *Consider the toy inverse kinematics problem depicted in Figure 5.4. It can be formulated as the following quadratically constrained feasibility program:*

$$\begin{aligned} \text{find} \quad & \mathbf{x} = \begin{bmatrix} \mathbf{x}_1^\top & \mathbf{x}_2^\top \end{bmatrix}^\top \in \mathbb{R}^4 \\ \text{s.t.} \quad & \|\mathbf{x}_1\|^2 = 1 \\ & \|\mathbf{x}_1 - \mathbf{x}_2\|^2 = 1 \\ & \|\mathbf{x}_2 - \mathbf{w}\|^2 = 1 \\ & \|\mathbf{x}_2\|^2 \leq 2, \end{aligned} \tag{5.20}$$

where $\mathbf{w} = \begin{bmatrix} 1 & 2 \end{bmatrix}^\top$ and, implicitly, $\mathbf{x}_0 = \mathbf{0}$. Defining the feasible set described by Equation (5.20) as the semialgebraic set Y'_{IK}, we can formulate the nearest point IK problem in Equation (5.19). Given some $\bar{\boldsymbol{\xi}} \in Y'_{IK}$ such that Equation (5.19) exhibits strong duality, can we show that $\boldsymbol{\xi}$ in a neighbourhood (with respect to $\mathbb{R}^{d(N-1)}$) of $\bar{\boldsymbol{\xi}}$ also exhibit strong duality?

Theorem 5 tells us that we should expect SDP stability at any $\bar{\boldsymbol{\xi}} \in Y_{\mathrm{IK}}$ such that the inequality constraint is not active and the arm is not fully extended. Indeed, dropping the inequality constraint altogether and perturbing the first two elements of $\bar{\boldsymbol{\xi}}$ yields the rank-1 SDP relaxation map in Figure 5.5.[2] As predicted, the rank-1 region, drawn in black, contains a neighbourhood about the origin, which represents $\bar{\boldsymbol{\xi}}$. In fact, the rather extensive rank-1 region suggests that a randomly selected $\boldsymbol{\xi}$ may perform well: this suggestion is borne out by our experimental results in Section 5.5.

[2] Recall that a rank-1 solution $\mathbf{Z}^*$ to an SDP relaxation of a QCQP indicates a tight SDP relaxation.

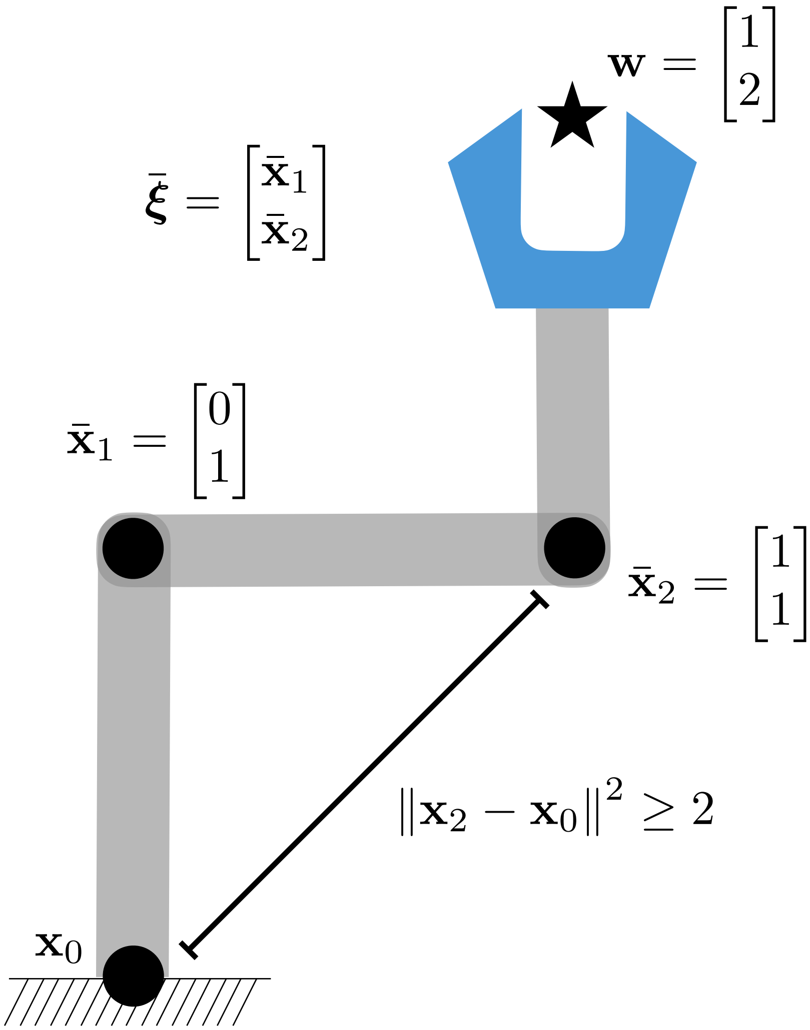


Figure 5.4: The planar 3-DOF manipulator from Problem 6 in a feasible configuration with the inequality constraint active.

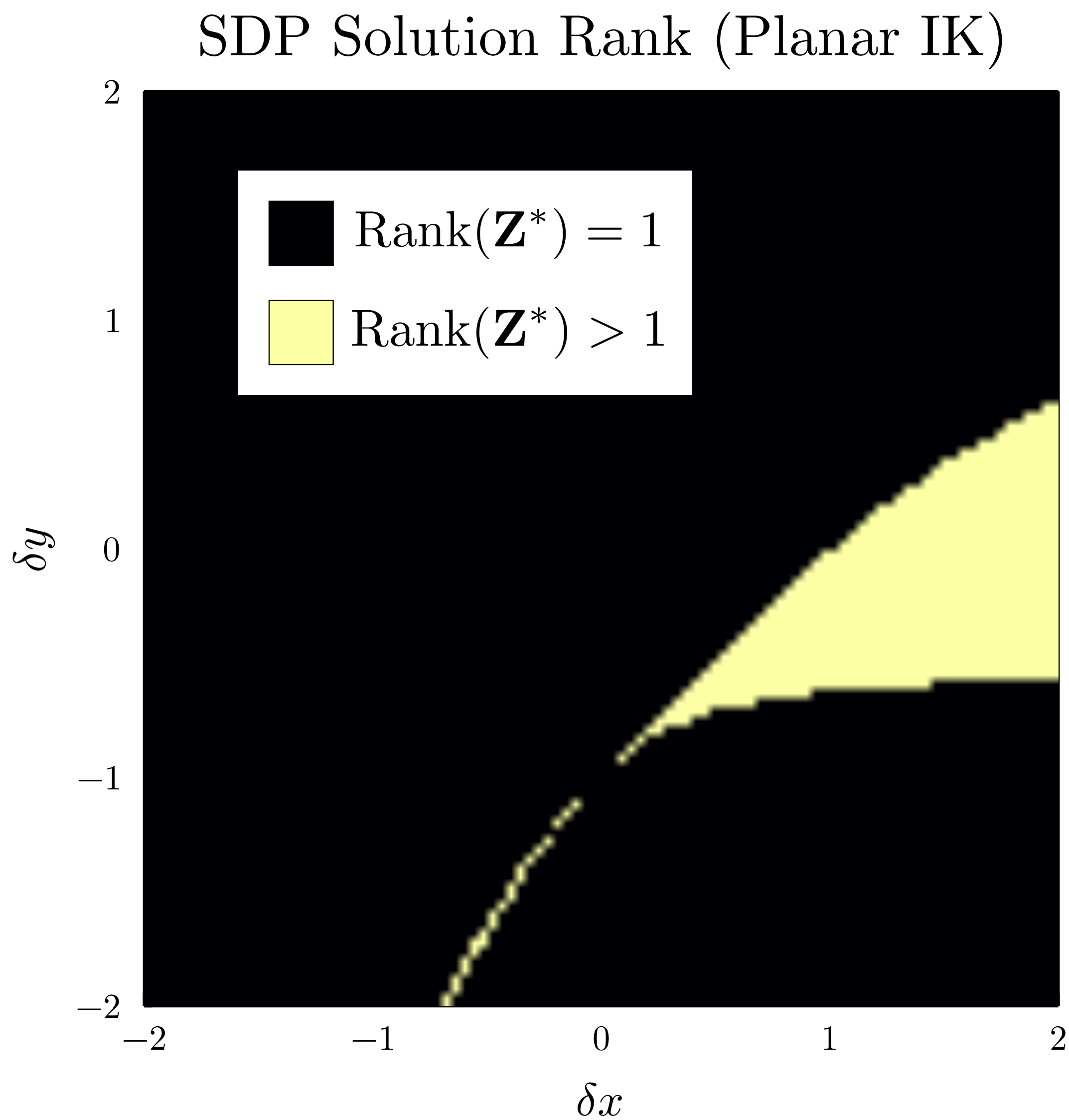


Figure 5.5: The rank of solution matrices for SDP relaxations of instances of Equation (5.12) generated by perturbing the first two elements of $\bar{\boldsymbol{\xi}}$ defined in Equation (5.21). A rank-1 solution indicates a tight SDP relaxation (i.e., zero-duality-gap). Note that Theorem 2 correctly predicts that there is a ZDG region in black surrounding the origin, which is a point in Y_{IK} at which LICQ holds.

We can now use Corollary 7 to prove that our nearest point formulation of IK is tight for $\boldsymbol{\xi}$ in the neighbourhood of

$$\bar{\boldsymbol{\xi}} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \in Y'_{\mathrm{IK}}, \tag{5.21}$$

which is the configuration depicted in Figure 5.4 for which the inequality constraint is active. As in Section 3.6.1, we need to demonstrate that LICQ $\left(\bar{\boldsymbol{\xi}}\right)$ holds in order to satisfy the requirements of Theorem 3. The equality and active inequality constraints at $\bar{\boldsymbol{\xi}}$ are

$$f_{\bar{\boldsymbol{\xi}}}(\mathbf{x}) = \begin{bmatrix} \|\mathbf{x}_1\|^2 - 1 \\ \|\mathbf{x}_1 - \mathbf{x}_2\|^2 - 1 \\ \|\mathbf{x}_2 - \mathbf{w}\|^2 - 1 \\ \|\mathbf{x}_2\|^2 - 2 \end{bmatrix} = \mathbf{0}, \tag{5.22}$$

and the Jacobian is

$$\nabla f_{\bar{\boldsymbol{\xi}}}(\mathbf{x}) = 2 \begin{bmatrix} \mathbf{x}_1{}^\top & \mathbf{0} \\ \mathbf{x}_1{}^\top - \mathbf{x}_2{}^\top & \mathbf{x}_2{}^\top - \mathbf{x}_1{}^\top \\ \mathbf{0} & \mathbf{x}_2{}^\top - \mathbf{w}^\top \\ \mathbf{0} & \mathbf{x}_2{}^\top \end{bmatrix} \in \mathbb{R}^{4\times 4}, \tag{5.23}$$

which we evaluate at $\bar{\boldsymbol{\xi}}$ to obtain

$$\nabla f_{\bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\xi}}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{5.24}$$

Since Equation (5.24) has full rank, LICQ $\left(\bar{\boldsymbol{\xi}}\right)$ holds and Theorem 3 tells us that SDP relaxations of the QCQP defined in Problem 6 are tight for $\boldsymbol{\xi}$ in the neighbourhood of $\bar{\boldsymbol{\xi}}$. This prediction is experimentally confirmed in Figure 5.6, which displays the rank-1 SDP relaxation region when perturbing the first two elements of $\bar{\boldsymbol{\xi}}$. As was the case for the twisted cubic in Section 3.6.1, the ZDG region in Figure 5.6 is a subset of the ZDG region in Figure 5.5, which ignores inequality constraints.

5.4 SOS Programming

Sum-of-squares programming is an approach for solving polynomial optimization problems with convex optimization. The standard SOS relaxation hierarchy (Parrilo, 2003; Lasserre, 2001) is equivalent to the Lagrangian dual relaxation with particular redundant constraints added (Ci-

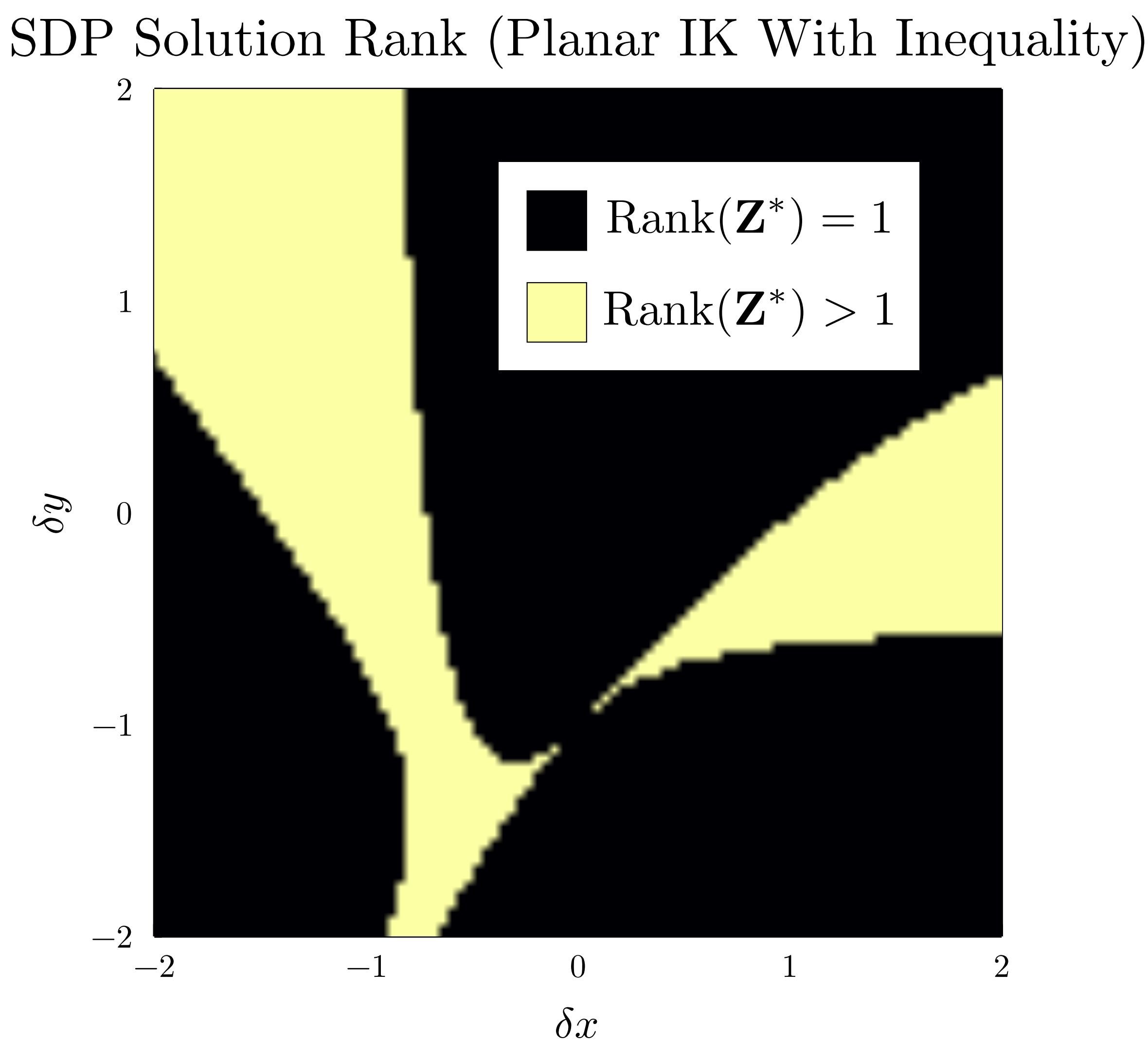


Figure 5.6: The rank of solution matrices for SDP relaxations of instances of inequality-constrained Equation (5.19) generated by perturbing the first two elements of $\bar{\boldsymbol{\xi}}$ defined in Equation (5.21). A rank-1 solution indicates a tight SDP relaxation (i.e., zero-duality-gap). Note that Theorem 3 correctly predicts that there is a ZDG region in black surrounding the origin, which is a point in Y'_{IK} for which LICQ holds.

fuentes et al., 2022). These redundant constraints can only make the relaxation tighter. Combined with the fact that the Lagrangian dual relaxation is a lower bound of the SDP relaxation of Equation (5.12), this tells us that the standard SOS hierarchy shares the stability property proved in Theorem 5. This work uses the Sparse Bounded Degree SOS (Sparse-BSOS) method of Weisser et al. (2018), a recent sparse extension of Lasserre et al. (2017), which introduced a SOS hierarchy that is less computationally costly than the standard SOS hierarchy in many cases, while remaining just as tight for QCQPs.

5.4.1 Sparse-BSOS

For a complete treatment of the Sparse-BSOS hierarchy, please refer to Weisser et al. (2018). Briefly, we are interested in solving Equation (5.12) in the equivalent form

$$t^\star = \sup_{t \in \mathbb{R}} \{t \,|\, f(\mathbf{y}) - t \geq 0,\ \forall \mathbf{y} \in \mathcal{K}\}, \tag{5.25}$$

where $\mathcal{K} = \{\mathbf{y} \in \mathbb{R}^n | 0 \leq g_j(\mathbf{y}) \leq 1,\ j = 1, \ldots, m\}$ is a semialgebraic set equivalent to Y_{IK} in Section 5.3.1, and $f(\mathbf{y}) = \|\mathbf{y} - \boldsymbol{\xi}\|^2$. The key insight of SOS optimization is that this problem (and other polynomial optimization problems) can be solved as a semidefinite program (SDP) with *Positivstellensatz*[3] results from real algebraic geometry (Lasserre, 2010; Parrilo, 2003). Many SOS relaxation hierarchies have been developed, but we use the Sparse-BSOS hierarchy of Weisser et al. (2018) because it leverages the natural sparsity of kinematic chains. The method enforces $f(\mathbf{y}) - t \geq 0$ by introducing the function

$$\begin{aligned} h_r(\mathbf{y}, \boldsymbol{\lambda}) &= \sum_{\alpha, \beta \in \mathbb{N}^m}^{|\alpha|_1 + |\beta|_1 \leq r} \lambda_{\alpha\beta} h_{r,\alpha\beta}(\mathbf{y}), \\ h_{r,\alpha\beta}(\mathbf{y}) &:= \prod_{j=1}^{m} g_j(\mathbf{y})^{\alpha_j} (1 - g_j(\mathbf{y}))^{\beta_j},\ \mathbf{y} \in \mathbb{R}^n, \end{aligned} \tag{5.26}$$

where $\boldsymbol{\lambda}$ contains the coefficients $\lambda_{\alpha\beta} \geq 0$ indexed by $\alpha, \beta \in \mathbb{N}^m$, and the maximum degree parameter r allows us to restrict the number of monomials used to construct h_r. Next, we seek to optimize

$$t^\star = \sup_{t, \boldsymbol{\lambda}} \{t \,|\, f(\mathbf{y}) - t - h_r(\mathbf{y}, \boldsymbol{\lambda}) \geq 0,\ \forall \mathbf{y}, \boldsymbol{\lambda} \geq 0\}, \tag{5.27}$$

where $h_r(\mathbf{y}, \boldsymbol{\lambda}) > 0$ when $\mathbf{y} \in \mathcal{K}$ (see Lasserre et al. (2017) for details). The problem is converted to an SDP by restricting the search to $\Sigma[\mathbf{y}]_k$, the set of SOS polynomials of degree at most $2k$, which constitute a subset of nonnegative polynomials:

$$q_r^k = \sup_{t, \boldsymbol{\lambda}} \{t \,|\, f(\mathbf{y}) - t - h_r(\mathbf{y}, \boldsymbol{\lambda}) \in \Sigma[\mathbf{y}]_k,\ \forall \mathbf{y}, \boldsymbol{\lambda} \geq 0\}. \tag{5.28}$$

[3]This is a German term that translates to "positive-locus-theorem", which is an apt descriptor for statements concerning the positivity of polynomials.

Each q_r^k describes a level of the BSOS hierarchy indexed by r and k (Lasserre et al., 2017). Since Equation (5.12) is a QCQP, $k = 1$ in our use of the solver. Finally, to produce the Sparse-BSOS hierarchy we partition Equation (5.28) into smaller blocks of variables and relevant constraints.

5.4.2 The Running Intersection Property

In order for the Sparse-BSOS hierarchy to converge to the global optimum as $r \to \infty$, the variables and functions involved must satisfy a sparsity property called the running intersection property or RIP (Weisser et al., 2018). The RIP holds if there exists $p \in \mathbb{N}$ and subsets $I_l \subseteq [n]$ and $J_l \subseteq [m]$ for all $l \in [p]$ such that:

- $f = \sum_{l=1}^{p} f^l$, for some $f^1, \ldots, f^p$ such that $f^l \in \mathbb{R}[\mathbf{x}; I_l],\ l \in [p]$;
- $g_j \in \mathbb{R}[\mathbf{x}; I_l]$ for all $j \in J_l$ and $l \in \{1, \ldots, p\}$;
- $\bigcup_{l=1}^{p} I_l = [n]$;
- $\bigcup_{l=1}^{p} J_l = [m]$;
- for all $l \in [p-1]$ there exists $s \leq l$ such that $(I_{l+1} \cap \bigcup_{k=1}^{l} I_k) \subseteq I_s$.

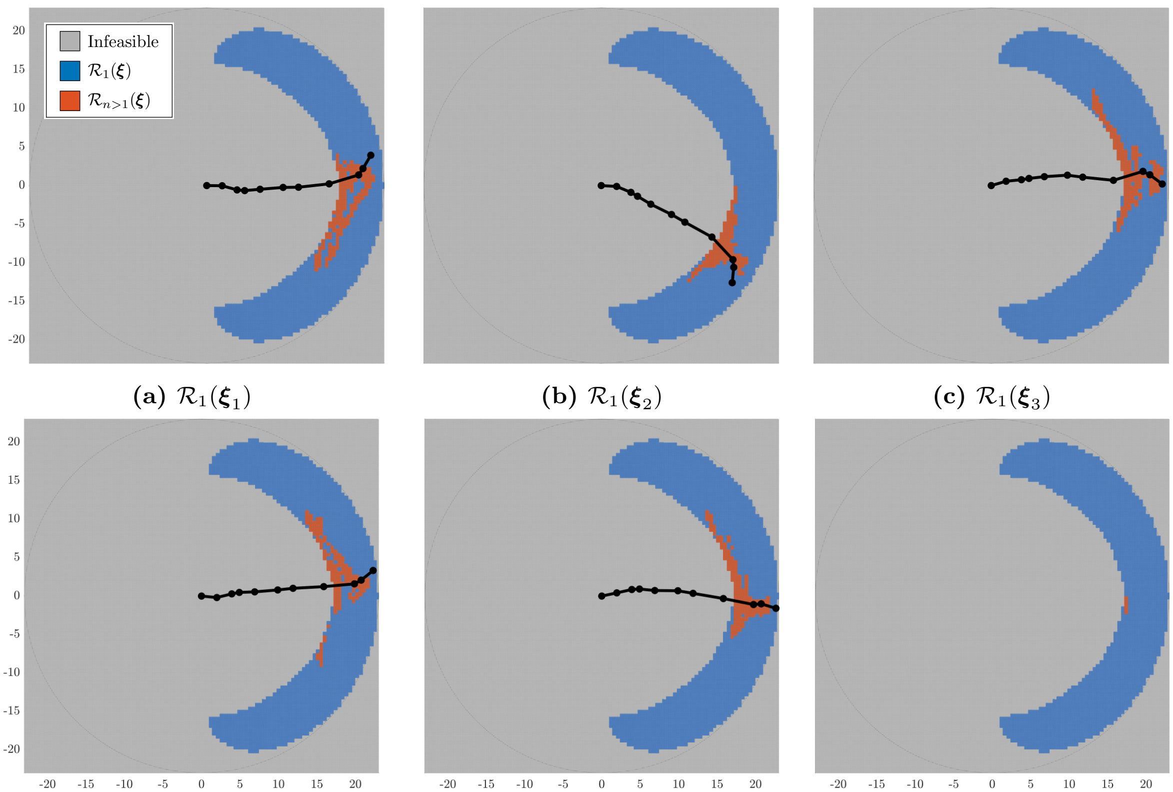
For the case of a 2D manipulator with only link length constraints, the partition consisting of overlapping pairwise joints satisfies the RIP. When joint limit constraints are introduced, overlapping triplets of joints are required. For redundant manipulators with many links, this partition amounts to an SDP with far fewer variables and constraints than a standard SDP or SOS relaxation would generate. The SDP produced by Sparse-BSOS has semidefinite constraints on variables of size $O(n^\star)$ for $k = 1$, where $n^\star = \max_l n_l$ and n_l is the number of variables in I_l (Weisser et al., 2018). For a d-dimensional manipulator using our nearest point formulation of IK with a partition that satisfying the RIP, $n^\star = 3d+1$. Using the Sparse-BSOS hierarchy therefore requires less memory and runtime as compared with its dense equivalent, whose semidefinite constraint variables would be of size $O(dN)$ (where N is the number of degrees of freedom as in prior sections). Our entire algorithm is summarized in Figure 5.1.

5.5 Experiments

In this section, we present IK solutions for simulated planar (2D) and spatial (3D) manipulators. All experiments were conducted with a MATLAB implementation of our approach on a computer with a 2.2GHz Intel Core i7-8750H CPU. Please refer to Table 5.1 for the kinematic chain link lengths and angle limits used in our experiments. The primary purpose of our experiments is to explore our global method and its tightness property presented in Theorem 5. We recognize that there may be local solvers that are competitive in some instances, but the focus of our work is on the global optimality properties of convex optimization methods.

Table 5.1: Parameters used to define the 10-DOF manipulator in the `SOS-IK` experiments.

Joint	1	2	3	4	5	6	7	8	9	10
$\lvert\theta_{\mathrm{i}}\rvert_{\max}$ (rad)	$\frac{\pi}{4}$	$\frac{\pi}{4}$	$\frac{\pi}{8}$	$\frac{\pi}{4}$	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{\pi}{4}$	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{\pi}{8}$
l_i (m)	2	2	1	2	3	2	4	4	1	2



(a) $\mathcal{R}_1(\boldsymbol{\xi}_1)$ **(b)** $\mathcal{R}_1(\boldsymbol{\xi}_2)$ **(c)** $\mathcal{R}_1(\boldsymbol{\xi}_3)$

(d) $\mathcal{R}_1(\boldsymbol{\xi}_4)$ **(e)** $\mathcal{R}_1(\boldsymbol{\xi}_5)$ **(f)** $\bigcup_{i=1}^{5} \mathcal{R}_1(\boldsymbol{\xi}_i)$

Figure 5.7: Heatmaps demonstrating rank-1 regions $\mathcal{R}_1(\boldsymbol{\xi})$ in target end-effector space for 5 different randomly sampled feasible reference configurations $\boldsymbol{\xi}_i$. In heatmaps (a)-(e), the joint positions for $\boldsymbol{\xi}_i$ for $i = 1, \ldots, 5$ are plotted in black. Each pixel's coordinates are a goal position for the end-effector, and the colour of each pixel represents the result of running `SOS-IK` with that goal position using $\boldsymbol{\xi}_i$ as the nearest point. Blue regions are end-effector goals for which `SOS-IK` gave a globally optimal rank-1 solution. Red regions indicate rank greater than 1, and grey indicates infeasible goal positions. Heatmap (f) displays the union of the rank-1 regions for all 5 reference configurations. See Table 5.1 for the manipulator's kinematic parameters.

5.5.1 Global Optimality

In Figure 5.7, we display the results of using `SOS-IK` to solve IK problems with five randomly sampled reference points $\boldsymbol{\xi}_i$ for a 10-DOF manipulator described by the parameters in Table 5.1. These points were chosen to be the joint positions of valid manipulator configurations which are plotted as black chains in subfigures (a)-(e). For each heatmap, the $x-$ and $y-$coordinates represent the position of uniformly sampled end-effector goal positions. The result of applying `SOS-IK` to each of these end-effector goals is indicated by colour. The blue cells are rank-1

Table 5.2: Performance of our method over 10,000 randomly generated feasible 2D (top) and 3D (bottom) pose goals.

DOF	5		7		10		12	
Method	`SOS-IK`	`fmincon`	`SOS-IK`	`fmincon`	`SOS-IK`	`fmincon`	`SOS-IK`	`fmincon`
2D (top)								
Position Error [m]	3.51×10^{-4}	6.78×10^{-5}	3.17×10^{-4}	4.93×10^{-5}	1.64×10^{-4}	2.69×10^{-4}	2.00×10^{-4}	2.67×10^{-4}
Solved [%]	97.46	97.15	99.52	93.33	99.47	87.65	98.11	53.19
Total Time [min]	9.08	1.67	13.73	1.93	20.05	3.52	21.52	3.6
3D (bottom)								
Position Error [m]	6.68×10^{-7}	2.18×10^{-5}	1.33×10^{-6}	3.79×10^{-5}	2.64×10^{-5}	6.93×10^{-4}	1.41×10^{-5}	3.1×10^{-3}
Solved [%]	99.29	99.67	99.68	99.84	98.51	95.79	99.72	94.55
Total Time [min]	18.88	9.44	25.36	23.23	38.22	46.10	33.92	51.69

(i.e., ZDG) solutions, which we denote $\mathcal{R}_1(\boldsymbol{\xi}_i)$ The red regions corresponds to goals that led to Sparse-BSOS solutions with rank greater than 1, whereas grey areas contain goals found to be infeasible. Since Sparse-BSOS solves a relaxation of the primal problem in Equation (5.12), infeasibility in the relaxed problem's solution provides a certificate of infeasibility for the primal. Moreover, the existence of a rank-1 solution is a certificate of optimality for the extracted solution, meaning it is the global optimum in terms of distance to $\boldsymbol{\xi}_i$. Figure 5.7 displays the union of the rank-1 regions for all 5 randomly sampled $\boldsymbol{\xi}_i$ and almost covers the entirety of the feasible goal space.

5.5.2 Inverse Kinematics for 2D and 3D Serial Chains

Theorem 5 proves that by formulating IK as Equation (5.12), globally optimal solutions can be recovered in certain workspace regions using convex relaxations such as Sparse-BSOS. The experiment in Section 5.5.1 demonstrates that such workspace regions are quite large even for complex kinematic chains like redundant spherical manipulators in two and three dimensions, which don't admit analytical solutions in the presence of joint limits. By solving 10,000 feasible IK problems, we demonstrate that our method (dubbed `SOS-IK`) usually outperforms a local numerical `fmincon`-based implementation of a joint angle-based IK solver in MATLAB in terms of percentage of recovered solutions, while also providing post-hoc numerical certificates of problem (in)feasibility. The results in Table 5.2 show the final end-effector position error, percentage of feasible IK solutions found, and total computation time over all problems for manipulators with an increasing number of DOF.

The upper half of Table 5.2 shows results for planar manipulators of increasing DOF, where `SOS-IK` outperforms the local optimization in the percentage of solved problems for every problem instance. While the solve times for `fmincon` are significantly lower in the planar case, we note that our method has well understood polynomial scaling properties which present themselves favourably for higher DOF and dimensionality. This can be seen in the case of a 12-DOF planar manipulator, where `SOS-IK` finds almost twice as many feasible solutions than `fmincon`, which falls into local minima. We show how this trend continues for spherical (3D) manipulators in the bottom of Table 5.2. As DOF increases, solve times become comparable and `SOS-IK` outperforms `fmincon` in terms of the number of problems successfully solved. We expect this difference to be even more pronounced when further kinematic constraints are introduced, as the IK problem will admit more local minima.

5.6 Summary and Future Work

In this chapter, we developed a novel and elegant formulation and solution of the IK problem for redundant manipulators. Our formulation of IK as a nearest point problem to a quadratic variety enabled us to prove the existence of problem instances admitting tight convex relaxations. Our use of convex relaxations provides certificates of global optimality alongside solutions. Our experiments demonstrated that the tight cases predicted by Theorem 5 encompass many practical situations which can be efficiently solved via standard interior point methods. Furthermore, we empirically demonstrated that convex relaxations are efficient tools for reliably determining the (in)feasibility of IK problems; this is in stark contrast to local solver-based methods that need frequent re-starts and sampling-based methods that scale inefficiently with the number of joints.

The tools presented in this chapter hold promise for a variety of robotic manipulation and planning tasks. Careful selection of the nearest point could be incorporated with task-specific goals like obstacle avoidance or low-energy motion planning. Methods for potentially extracting solutions from Sparse-BSOS solutions with rank greater than 1 warrant investigation. Additionally, theoretical tools developed in Cifuentes et al. (2022) and Cifuentes et al. (2018) could be used to precisely quantify the values of $\boldsymbol{\xi}$ and goal poses for which SDP relaxations of our problem are tight. Using our method as a sub-solver in a branch-and-bound or mixed-integer nonlinear programming approach to inverse kinematics similar to Dai et al. (2019) also deserves attention as a means of incorporating complex obstacle avoidance constraints into a fast and efficient IK solver with performance guarantees.

Finally, the Sparse-BSOS solver used here is a generic MATLAB library that does not exploit structure specific to our problem or use performance optimizations available in lower-level languages. Faster performance could be achieved by considering a custom sparse SDP relaxation (e.g., similar to the one in Nie (2009)) for sparsity patterns specific to IK. For kinematic chains with tens or hundreds of degrees of freedom, it may also be fruitful to investigate the use of Burer-Monteiro methods (Boumal et al., 2020), which can require less time and memory, instead of standard interior point solvers.

Chapter 6

Inverse Kinematics for Generic Revolute Manipulators

> Indeed, convexity has an immensely rich structure and numerous applications. On the other hand, almost every "convex" idea can be explained by a two-dimensional picture. There must be some reason for that apart from the tautological one that all our pictures are two-dimensional.
>
> Alexander Barvinok,
> *A Course in Convexity*

Solving inverse kinematics (IK) is an essential step for motion planning with articulated robots. However, an efficient algorithm with a high success rate for robots with redundant degrees of freedom in obstacle-laden workspaces remains elusive. Solving this problem would help enable fast and reliable autonomous mobility for manipulators, humanoid robots, and other articulated mechanisms. In this chapter, we move beyond the relatively simple planar and spherical manipulators of Chapter 5 and develop a QCQP model capable of describing a broad class of revolute manipulators.

Optimization-based approaches typically use the joint angles of the robot as decision variables. While low-dimensional, this parameterization leads to nonconvex cost and constraint functions involving the product of multiple trigonometric functions of the joint angles. This nonconvexity makes finding global minima challenging for numerical solvers.

Recently, a number of IK techniques have utilized alternative parameterizations based on the distances between control points fixed to a robot (Porta et al., 2005b; Le Naour et al., 2019; Blanchini et al., 2017). This *distance-geometric* view of IK increases the number of parameters needed, but elegantly describes the workspace of a robot with simple pairwise distance

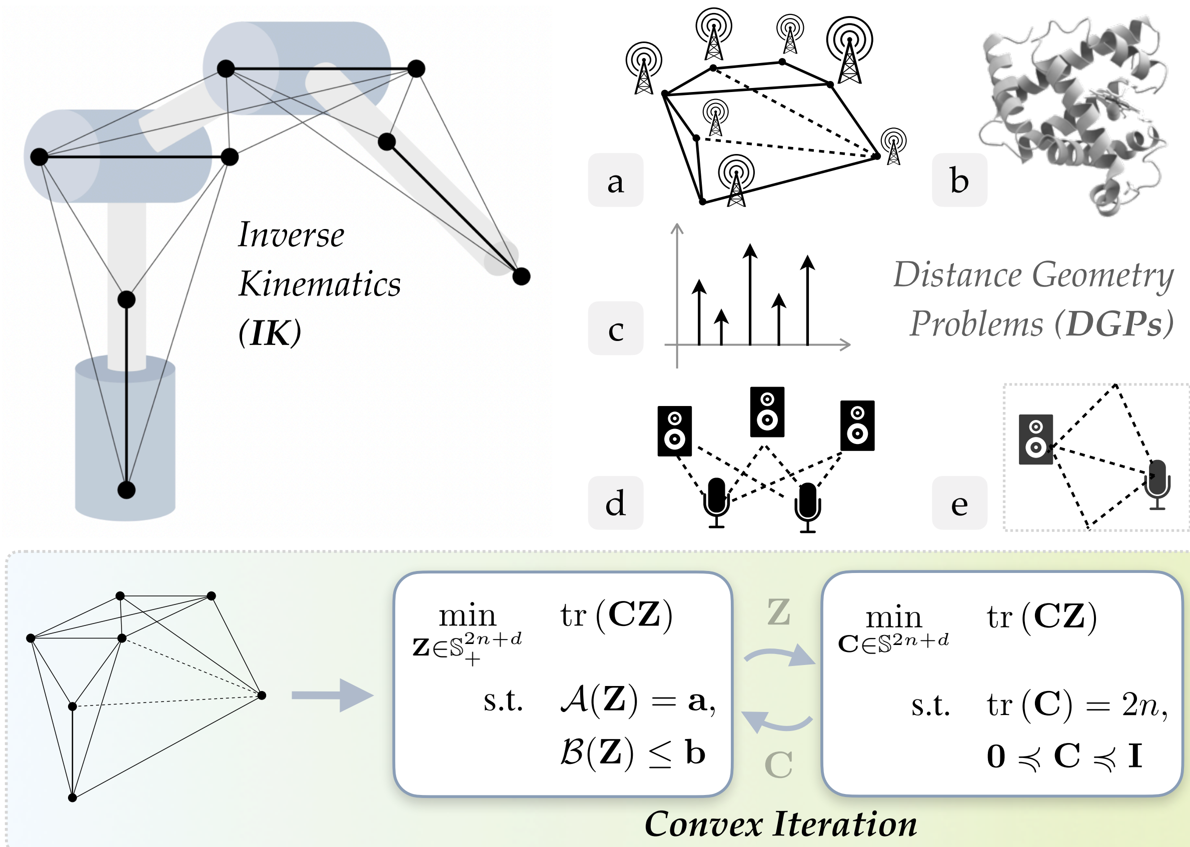


Figure 6.1: We apply a distance-geometric formulation of IK to develop a fast and accurate IK solver based on low-rank convex optimization. Our formulation connects IK to the rich literature on convex relaxations for other distance geometry problems such as: a) sensor network localization, b) molecular conformation, c) sparse phase retrieval, d) microphone calibration, and e) indoor acoustic localization.

constraints. In this chapter, we connect IK to the classical distance geometry problem (DGP), which has found application in many domains, including those in Figure 6.1 and Table 6.1. Noting similarities between the IK problem and DGPs such as sensor network localization (SNL), our main contribution is to leverage the mature literature on semidefinite programming (SDP) relaxations for DGPs to develop `CIDGIK` (Convex Iteration for Distance-Geometric Inverse Kinematics), a novel IK solver. `CIDGIK` applies a fast, minimal-rank-promoting algorithm (Dattorro, 2005) to an SDP relaxation of our formulation of IK, encompasses a wide variety of redundant robot models, and naturally incorporates spherical obstacles and planar workspace constraints. We provide a free and open source Python implementation[1] of `CIDGIK` along with experiments demonstrating its superior success rate and speed when compared with a standard nonlinear approach to solving the conventional angular formulation of IK.

[1] Code: `https://github.com/utiasSTARS/graphIK`.

6.1 Related Work

In this section we briefly review the state of the art in IK, distance geometry, and semidefinite optimization. We emphasize the intersection of these three fields, and contrast the properties of a variety of recent IK solvers.

6.1.1 Inverse Kinematics

Simple closed-form solutions can be derived for common manipulator robots that do not possess redundant degrees of freedom (Spong et al., 2005). For example, kinematic chains with six revolute degrees of freedom (DOF) possess at most 16 solutions for IK problems with pose goals (Manocha and Canny, 1994). This chapter deals with the more challenging case of IK with collision avoidance for redundant manipulators with one or more end-effectors. First-order methods for IK (Buss, 2004) are a mature and popular choice for a broad class of problems in robotics and graphics (Aristidou et al., 2018). TRAC-IK (Beeson and Ames, 2015) is a fast and freely-available software implementation of IK *without* collision avoidance for redundant manipulators that concurrently runs an inverse Jacobian algorithm and a sequential quadratic programming algorithm. A recent mixed-integer programming approach (Dai et al., 2019) provides approximate solutions with global guarantees for a general formulation of IK with joint limits, multiple end-effectors, and expressive workspace constraints (e.g., specifying free space with polyhedra). In spite of its high success rate on challenging problems, the approximate nature and long runtime (on the order of 15 seconds for an 18-DOF quadruped) prohibit the use of Dai et al. (2019) in realtime applications.

6.1.2 Distance Geometry

Euclidean distance geometry is concerned with estimating the positions of points given a subset of their pairwise distances (Liberti et al., 2014; Dattorro, 2005). Applications, some of which are listed in Table 6.1 and depicted in Figure 6.1, include problems in computational chemistry, machine learning, and signal processing (Dokmanic et al., 2015). A distance-geometric formulation of IK is presented in Porta et al. (2005b) for a family of common revolute robot manipulators. This formulation is solved with the complete but computationally expensive (and therefore inappropriate for real-time robotics applications) branch-and-prune solver introduced in Porta et al. (2005a). Other approaches to IK that incorporate distance constraints on point variables include the unconstrained quartic optimization approach to IK for graphics found in Le Naour et al. (2019) and the convex relaxation for simple robotic chains in Blanchini et al. (2015, 2017). In this work, we apply methods and analysis based on SDP relaxations for SNL (So and Ye, 2007; Dattorro, 2005) to a formulation inspired by Porta et al. (2005b). We characterize the similarities and differences between IK and other DGPs in Table 6.1 and Section 6.3.

Application	Distance Measurements *missing*	*noisy*	*free*†	Under Dtmnd.
Wireless sensor networks	✓	✓	✗	✗
Molecular conformation	✓	✓	✗	✓/✗
Sparse phase retrieval	✗	✓	✓	✗
Microphone calibration	✓	✓	✗	✗
Indoor acoustic localization	✗	✓	✓	✗
Inverse kinematics	✓	✗	✗	✓

Table 6.1: Properties of different types of DGPs (extended from Dokmanic et al. (2015)). IK is unique in that it deals with "noiseless" (i.e., exact) geometric constraints *and* is underdetermined because of the existence of multiple valid solutions. †Free (or *unlabeled*) measurements are those which lack a known association with two points.

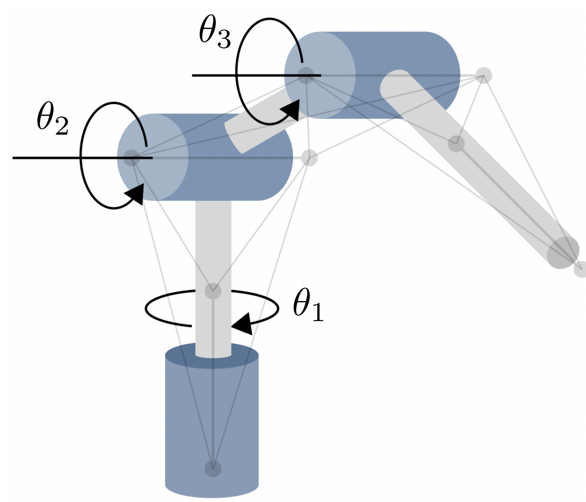
6.1.3 Semidefinite Programming

Many parameter estimation problems in statistics and engineering can be expressed as QCQPs (Cifuentes et al., 2022; So and Ye, 2007). While difficult to solve in the worst case, primal SDP relaxations of QCQPs (Boyd and Vandenberghe, 1997), which we employ in Section 6.3, can often be efficiently solved by interior-point methods (So and Ye, 2007). Remarkably, in many cases, this relaxation is provably *tight* under the assumption of low noise and one can recover the global optimum by solving the relaxed problem (Cifuentes et al., 2022). Furthermore, many of these problems exhibit structure (e.g., chordal sparsity) that can be exploited for improved performance over generic SDP solvers on large-scale problem instances (Majumdar et al., 2019). In Trutman et al. (2020), Lasserre's hierarchy of SDP relaxations is applied to IK for 7-DOF revolute manipulators, but solutions are prohibitively slow for real-time applications.

Our approach extends the distance-geometric description of the kinematics of a revolute robot in Marić et al. (2021) to a form very similar to the SNL problem. Building on our work with planar and spherical chains in Chapter 5, we analyze and solve this formulation using the rich literature on SDP relaxations for SNL (So and Ye, 2007; Nie, 2009; Ding et al., 2010), while noting the properties in Table 6.1 that make IK distinct from SNL. Our approach is similar to the work presented in Yenamandra et al. (2019), where the authors apply a different convex relaxation to a formulation of IK for arbitrary joint angle-limited tree-like revolute models. However, they do not incorporate obstacle avoidance, report runtimes in excess of 10 seconds, and their relaxation only provides coarse initializations for a local solver.

6.2 Problem Formulation

We begin with a summary of our notation, followed by a detailed description of our "distance-geometric" formulation of robot kinematics. Since we are dealing with a more general class of revolute manipulators, this formulation is significantly more complex than the one in Chapter 5.



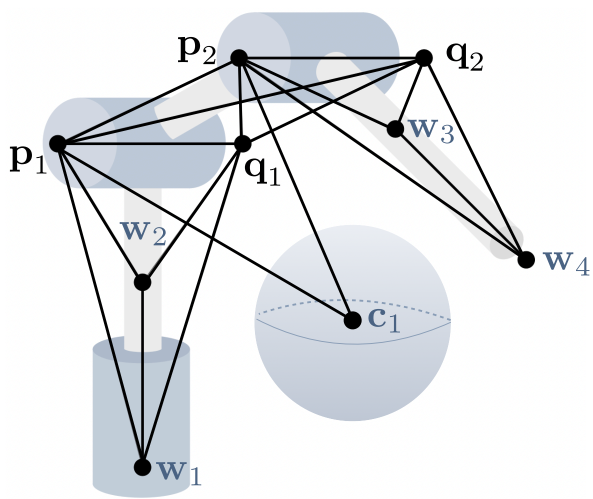


Figure 6.2: Visualization of a 3-DOF revolute manipulator: a) Actuated joints and links, overlaid with the graph of known distances. b) Depiction of our distance-geometric formulation with a spherical obstacle (cf. Section 6.2.2).

6.2.1 Notation

Boldface lower and upper case letters (e.g., $\mathbf{x}$ and $\mathbf{P}$) represent vectors and matrices respectively. The bracketed superscript in $\mathbf{A}^{(i)}$ indicates the ith column of the matrix $\mathbf{A}$. We write $\mathbf{I}_n$ (or $\mathbf{I}$ when clear from context) for the $n \times n$ identity matrix. The space of $n \times n$ symmetric and symmetric positive semidefinite matrices are denoted $\mathbb{S}^n$ and $\mathbb{S}^n_+$, respectively, and we also write $\mathbf{A} \succeq \mathbf{B}$ ($\mathbf{A} \succ \mathbf{B}$) to indicate that $\mathbf{A} - \mathbf{B}$ is PSD (PD). We denote the set of indices $\{1, \ldots, n\}$ as $[n]$ for any $n \in \mathbb{N}$. Finally, $\|\cdot\|$ always represents the Euclidean norm.

6.2.2 Kinematic Model

Once again, our proposed kinematic model eschews joint angle variables in favour of points embedded in $\mathbb{R}^d$ (Porta et al., 2005b), where $d \in \{2, 3\}$ for physically realizable revolute robots. These points are strategically fixed relative to an N-DOF robot's articulated joints such that their positions fully describe the underlying angular configuration $\boldsymbol{\theta} \in \mathcal{C} \subseteq \mathcal{T}^N$, where $\mathcal{T}^N$ is the N-dimensional torus. IK is typically expressed as finding, for a given goal $\mathbf{w}^{\mathrm{g}}$, joint angles $\boldsymbol{\theta} \in \mathcal{C}$ that satisfy the system of equations

$$F(\boldsymbol{\theta}) = \mathbf{w}^{\mathrm{g}} \in \mathcal{W}, \tag{6.1}$$

where $F : \mathcal{C} \to \mathcal{W}$ is the trigonometric *forward* kinematics function that maps joint angles in the configuration space $\mathcal{C}$ to end-effector positions or poses in the workspace $\mathcal{W}$.

When a closed form solution $\boldsymbol{\theta} = F^{-1}(\mathbf{w}^{\mathrm{g}})$ is unavailable, numerical methods are typically used to solve optimization-based formulations of IK (Erleben and Andrews, 2019):

Problem 7 (Inverse Kinematics)**.** *Given a robot's forward kinematics $F : \mathcal{C} \to \mathcal{W}$ and desired end-effector position(s) or pose(s) $\mathbf{w} \in \mathcal{W}$, find joint angles $\boldsymbol{\theta}$ that solve*

$$\min_{\boldsymbol{\theta} \in \mathcal{C}} \quad \left\| F(\boldsymbol{\theta}) - \mathbf{w}^{\mathrm{g}} \right\|^2 . \tag{6.2}$$

Our formulation, which is illustrated for a simple 3-DOF manipulator in Figure 6.2, specifies the goal poses or positions of a robot's end-effector(s) and base using $m \geq 2$ points or *anchors* $\mathbf{w}_k \in \mathbb{R}^d$ for $k \in [m]$. In contrast with the traditional angular formulation of Problem 7 displayed in Figure 6.2a, the state space of our formulation is comprised of points rigidly fixed to n "unanchored" joints.[2] For the example in Figure 6.2, $n = 2$ and the unanchored joints are those actuated by θ_2 and θ_3, while the joint actuated by θ_1 is anchored. To enforce rigid link lengths that are invariant to the robot's angular configuration $\boldsymbol{\theta}$, each unanchored joint $i \in [n]$ is assigned a pair of points $\mathbf{p}_i, \mathbf{q}_i$ positioned along its axis of rotation as in Figure 6.2b such that $\|\mathbf{p}_i - \mathbf{q}_i\| = 1 \ \forall \ \boldsymbol{\theta}$. Since the relative pose between consecutive joints i and j is fixed by a rigid link, each pair of joints is described by the six pairwise distance constraints between $\mathbf{p}_i$, $\mathbf{q}_i$, $\mathbf{p}_j$, and $\mathbf{q}_j$ shown as lines in Figure 6.2b. Conveniently, our formulation treats a robot's fixed base in a manner identical to end-effectors (e.g., $\mathbf{w}_1$ and $\mathbf{w}_2$ in Figure 6.2b).

End-effector position and pose targets are enforced via the fixed distance constraints between unanchored joints and neighbouring anchors (e.g., $\|\mathbf{p}_2 - \mathbf{w}_3\|$ and $\|\mathbf{q}_1 - \mathbf{w}_1\|$ in Figure 6.2b). A problem instance is defined by an assignment of all anchors

$$\mathbf{W} = [\mathbf{w}_1 \ \cdots \ \mathbf{w}_m] \in \mathbb{R}^{d \times m}, \tag{6.3}$$

and the unknown decision variables can be collected to form the matrix

$$\mathbf{X} = [\mathbf{p}_1 \ \mathbf{q}_1 \ \cdots \ \mathbf{p}_n \ \mathbf{q}_n] \in \mathbb{R}^{d \times 2n}. \tag{6.4}$$

For example, an end-effector's position can be constrained by specifying a single anchor position on the tip of the end-effector (e.g., $\mathbf{w}_4$ in Figure 6.2b). The "direction" (i.e., orientation without yaw), of this end-effector can be constrained by specifying the position of an additional distinct anchor (e.g., $\mathbf{w}_3$ in Figure 6.2b) along the desired direction (Marić et al., 2021).

In order to form a graph describing our kinematic model, let $\mathcal{V}_{\mathrm{j}} = [2n]$ and $\mathcal{V}_{\mathrm{w}} = [2n{+}m] \backslash [2n]$ be index sets for vertices representing variable points ($\mathbf{p}_i$ and $\mathbf{q}_i$) and anchors ($\mathbf{w}_j$) respectively, and let $\mathcal{V} = \mathcal{V}_{\mathrm{j}} \cup \mathcal{V}_{\mathrm{w}} = [2n + m]$. Similarly, the edge sets describing fixed distance constraints between variable joints and between variable joints and anchors are $\mathcal{E}_{\mathrm{l}} \subseteq \mathcal{V}_{\mathrm{j}} \times \mathcal{V}_{\mathrm{j}}$ and $\mathcal{E}_{\mathrm{w}} \subseteq \mathcal{V}_{\mathrm{j}} \times \mathcal{V}_{\mathrm{w}}$, respectively. We represent the equality constraints with a weighted directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\mathrm{eq}}, \ell)$, where $\ell : \mathcal{E}_{\mathrm{eq}} \to \mathbb{R}_+$ encodes the fixed distances:

$$\ell : (i, j) \mapsto \left\| \mathbf{x}_i - \mathbf{x}_j \right\| , \tag{6.5}$$

[2] A joint is *anchored* if its axis of rotation is invariant to changes in $\boldsymbol{\theta}$.

where $\mathbf{x}_i$ and $\mathbf{x}_j$ can refer to either variable joints or fixed anchors, and $\mathcal{E}_{\mathrm{eq}} = \mathcal{E}_{\mathrm{l}} \cup \mathcal{E}_{\mathrm{w}}$. To simplify our notation, we introduce the incidence matrix

$$\mathbf{B}(\mathcal{E}_{\mathrm{eq}})_{i,e} = \begin{cases} 1 \text{ if } e \in \delta(i)^+, \\ -1 \text{ if } e \in \delta(i)^-, \\ 0 \text{ otherwise}, \end{cases} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}_{\mathrm{eq}}|} \tag{6.6}$$

where $\delta(i)^-$ and $\delta(i)^+$ are the set of edges leaving and entering $i \in \mathcal{V}$, respectively.[3] We also introduce the matrix

$$\mathbf{P} = [\mathbf{X}\ \mathbf{W}] \in \mathbb{R}^{d \times (2n+m)}. \tag{6.7}$$

Thus, the eth column

$$\mathbf{P}\mathbf{B}(\mathcal{E}_{\mathrm{eq}})^{(e)} = \mathbf{P}^{(j)} - \mathbf{P}^{(i)} \tag{6.8}$$

is the relative position of vertices i and j. The diagonal elements of the product

$$\mathbf{B}(\mathcal{E}_{\mathrm{eq}})^{\mathsf{T}}\mathbf{P}^{\mathsf{T}}\mathbf{P}\mathbf{B}(\mathcal{E}_{\mathrm{eq}}) \in \mathbb{S}^{|\mathcal{E}_{\mathrm{eq}}|} \tag{6.9}$$

are therefore equal to

$$\ell(e) = \left\|\mathbf{P}^{(j)} - \mathbf{P}^{(i)}\right\|^2, \ \ e = (i,j) \in \mathcal{E}_{\mathrm{eq}}. \tag{6.10}$$

Recalling that $\mathbf{d}_e = \ell(e)^2$, we can now use $\mathbf{B}(\mathcal{E}_{\mathrm{eq}})$ to compactly summarize the squared distance constraints as

$$\mathrm{diag}\left(\mathbf{B}(\mathcal{E}_{\mathrm{eq}})^{\mathsf{T}}\mathbf{P}^{\mathsf{T}}\mathbf{P}\mathbf{B}(\mathcal{E}_{\mathrm{eq}})\right) = \boldsymbol{\ell}, \tag{6.11}$$

where

$$\boldsymbol{\ell}_e = \ell(e)^2 \ \ \forall e \in \mathcal{E}_{\mathrm{eq}}. \tag{6.12}$$

This formulation is equivalent to the one used in So and Ye (2007) for SNL.

6.2.3 Workspace Constraints

Consider a workspace $\mathcal{W}$ with obstacles or other regions that our robot is forbidden from occupying. We model these constraints with a finite set of spheres $\mathcal{O}$ whose union is chosen to cover the restricted regions:

$$\|\mathbf{x}_i - \mathbf{c}_j\|^2 \geq l_j^2 \ \ \forall i \in \mathcal{V}_{\mathrm{j}}, \ \ \forall j \in \mathcal{O}, \tag{6.13}$$

where $\mathbf{c}_j \in \mathbb{R}^d$ is the centre and $l_j > 0$ is the radius of sphere $j \in \mathcal{O}$. This "union of balls" environment representation has been used in previous work on robot motion planning (Varava et al., 2020). Furthermore, Corollary 1 tells us that we can approximate compact (i.e., closed

[3] Throughout this section we slightly abuse our notation by using e to refer both to a directed edge $e = (i,j) \in \mathcal{E}_{\mathrm{eq}}$ as well as an integer ($e \in [|\mathcal{E}_{\mathrm{eq}}|]$) corresponding to a fixed index of this same edge.

and bounded) obstacles of arbitrary complexity up to any desired precision with a large number of spheres. For specifying a spherical region of free space *within* which some subset of the joints must lie (e.g., the obstacle-free regions computed in Deits and Tedrake (2015a)), the inequality in Equation (6.13) is simply reversed.

Constraining a point $\mathbf{x} \in \mathbb{R}^d$ on our robot to lie in (on one side of) a plane can be simply encoded as an affine equality (inequality) constraint with the plane's normal $\mathbf{n}$ and its minimum Euclidean distance c from the origin:

$$\mathbf{x}^\mathsf{T}\mathbf{n} \leqq c. \tag{6.14}$$

One useful application of planar constraints is in restricting a legged robot's feet to lie on (or above) the floor.

In addition, self-collisions and workspace constraints on points lying *between* joints (i.e., on long links) can be introduced by adding auxiliary variables $\mathbf{y} \in \mathbb{R}^d$ that are fixed between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ for some $(i,j) \in \mathcal{E}_{\mathrm{eq}}$. We can parameterize the interior of the line segment connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ with $\alpha \in (0,1)$ and constrain $\mathbf{y}$ to lie at some point of our choosing on this line segment:

$$\mathbf{y} = (1-\alpha)\mathbf{x}_i + \alpha\mathbf{x}_j. \tag{6.15}$$

This auxiliary point can be used in collision avoidance constraints between $\mathbf{y}$ and obstacles:

$$\|\mathbf{y} - \mathbf{c}_k\|^2 \geq l_k^2 \ \ \forall k \in \mathcal{O}. \tag{6.16}$$

Finally, `CIDGIK` can easily incorporate self-collision constraints with distance inequalities between variables:

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 \geq \epsilon_{i,j} \ \ \forall (i,j) \in \mathcal{E}_{\mathrm{eq}}, \tag{6.17}$$

where $\epsilon_{i,j}$ is any user-defined threshold, ideally based on robot geometry. Equations of this form can also replace $\mathbf{x}_i$ or $\mathbf{x}_j$ with auxiliary variables defined in Equation (6.15). Since the constraints defined in this section are either linear or distance-geometric, they are supported by the "rank-d" semidefinite relaxation of Equation (6.19) that we introduce in Section 6.3.

6.2.4 QCQP Formulation

We can use the quadratic expressions of Equations (6.11) and (6.13) along with the linear constraints of Equation (6.14) to describe the solutions to our IK problem as the following quadratic feasibility program:

Problem 8 (Feasibility QCQP)**.**

$$
\begin{aligned}
\text{find} \quad & \mathbf{X} \in \mathbb{R}^{d \times 2n} \\
\text{s.t.} \quad & \mathbf{P} = [\mathbf{X} \ \mathbf{W}], \\
& \operatorname{diag}(\mathbf{B}(\mathcal{E}_{eq})^\top \mathbf{P}^\top \mathbf{P} \mathbf{B}(\mathcal{E}_{eq}))) = \boldsymbol{\ell}, \\
& \|\mathbf{x}_i - \mathbf{c}_j\|^2 \geq l_j^2 \ \ \forall i \in \mathcal{V}_\mathrm{j}, \ \ \forall j \in \mathcal{O}, \\
& \mathbf{x}_i^\top \mathbf{n}_i = c_i \ \ \forall i \in \mathcal{V}_\mathrm{p} \subset \mathcal{V}_\mathrm{j},
\end{aligned}
\tag{6.18}
$$

where $\mathcal{E}_{eq}$, $\boldsymbol{\ell}$, and $\mathbf{W}$ are problem parameters defined in Section 6.2.2, $\mathcal{O}$ is the set of spherical obstacle constraints of the form in Equation (6.13)*, and $\mathcal{V}_\mathrm{p}$ indexes the nodes confined to lie in some plane described by $\mathbf{n}_i$ and c_i in Equation* (6.14)*.*

Remark 2 (Hardness of Problem 8): When solutions exist, the feasible set of Problem 8 is nonconvex and therefore challenging to characterize. In fact, the analogous formulation of Problem 8 for SNL with exact distance measurements, no obstacles, and known dimension d is strongly NP-hard (Krislock and Wolkowicz, 2010; Aspnes et al., 2004).

While quite general, our formulation does have a few limitations: it is only capable of describing the kinematics of an articulated robot with neighbouring joint axes that are coplanar (i.e., parallel or intersecting),[4] it cannot enforce arbitrary joint angle limits, and it only supports spherical and planar workspace constraints (Marić et al., 2021).

6.3 Semidefinite Relaxations

With a QCQP formulation in hand, we once again turn to the SDP relaxation machinery in Chapter 3 as a means of efficiently computing solutions to Problem 8. Introducing the lifted matrix variable

$$
\mathbf{Z}(\mathbf{X}) \triangleq [\mathbf{X} \ \ \mathbf{I}_d]^\top [\mathbf{X} \ \ \mathbf{I}_d] = \begin{bmatrix} \mathbf{X}^\top \mathbf{X} & \mathbf{X}^\top \\ \mathbf{X} & \mathbf{I}_d \end{bmatrix} \in \mathbb{S}_+^{2n+d}
\tag{6.19}
$$

permits us to rewrite the quadratic constraints of Problem 8 as *linear* functions of $\mathbf{Z}(\mathbf{X})$. Since $\mathbf{Z}(\mathbf{X})$ is an outer product of matrices with rank of at most d (the dimension of the space in which the robot operates), we know that $\operatorname{rank}(\mathbf{Z}) \leq d$. This differs from the primal SDP relaxation in Problem 2, which was implicitly "rank-1" due to $\mathbf{Z}$'s construction as a lifting of a vector, which can be thought of as a matrix of rank at most 1.

Equation (6.19) allows us to write a number of degree-2 or lower polynomial expressions in $\mathbf{X}$ as linear expressions of $\mathbf{Z}$. Since $\mathbf{Z}_{i,j} = \mathbf{x}_i^\top \mathbf{x}_j \ \forall i, j \leq 2n$, each equation in 6.11 that is

[4] Fortunately, many commercial manipulators, including those in Figure 6.5, satisfy this requirement. Additionally, we discuss potential extensions to arbitrary revolute manipulators in Section 6.5.4.

between two variables $\mathbf{x}_k, \mathbf{x}_l$, $k \neq l$ can be written $\operatorname{tr}(\mathbf{AZ}) = \ell((k,l))$, where

$$\mathbf{A}_{j,i} = \mathbf{A}_{i,j} = \begin{cases} 1 \text{ if } i = j \in \{k, l\}, \\ -1 \text{ if } i = k,\ j = l, \\ 0 \text{ otherwise.} \end{cases} \tag{6.20}$$

Similarly, the expression for the squared distance between a variable $\mathbf{x}_k$ and some anchor $\mathbf{w}$ (or obstacle centre $\mathbf{c}$) can be encoded with matrix $\mathbf{M} \in \mathbb{S}^{2n+d}$ where

$$\begin{aligned} \mathbf{M}_{k,k} &= 1, \\ \mathbf{M}_{2n:2n+d,k} &= -\mathbf{w}, \\ \mathbf{M}_{k,2n:2n+d} &= -\mathbf{w}^\top, \\ \mathbf{M}_{2n+d,2n+d} &= \|\mathbf{w}\|^2. \end{aligned} \tag{6.21}$$

Likewise, linear constraints can be enforced with matrices manipulating the elements in $\mathbf{Z}_{2n:2n+d,1:2n}$ and its symmetric counterpart. These constraints can be collected in the linear maps

$$\mathcal{A}(\mathbf{Z}) = \mathbf{a},\ \ \mathcal{A} : \mathbb{S}^{2n+d} \to \mathbb{R}^{m+d^2+|\mathcal{V}_\mathrm{p}|}, \tag{6.22}$$

$$\mathcal{B}(\mathbf{Z}) \leq \mathbf{b},\ \ \mathcal{B} : \mathbb{S}^{2n+d} \to \mathbb{R}^{|\mathcal{O}|}, \tag{6.23}$$

which appear in our SDP problem formulation. The final constraint worth noting, which contributes d^2 to the dimension of the codomain of $\mathcal{A}$, simply arises from the requirement that the bottom-right $d \times d$ diagonal of $\mathbf{Z}$ is equal to $\mathbf{I}_d$. As in Section 3.4.4, we can now replace $\mathbf{Z}(\mathbf{X})$ with a *generic* PSD matrix $\mathbf{Z} \succeq \mathbf{0}$ to produce the following semidefinite relaxation:

Problem 9 (SDP Relaxation of Problem 8)**.**

$$\begin{aligned} \text{find} \quad & \mathbf{Z} \in \mathbb{S}_+^{2n+d} \\ \text{s.t.} \quad & \mathcal{A}(\mathbf{Z}) = \mathbf{a}, \\ & \mathcal{B}(\mathbf{Z}) \leq \mathbf{b}, \end{aligned} \tag{6.24}$$

where $\mathcal{A} : \mathbb{S}^{2n+d} \to \mathbb{R}^{m+d^2+|\mathcal{V}_\mathrm{p}|}$ *and* $\mathbf{a} \in \mathbb{R}^{m+d^2+|\mathcal{V}_\mathrm{p}|}$ *encode the linear equations that enforce the constraints in Equation* (6.11) *and Equation* (6.14) *after applying the substitution in Equation* (6.19)*, and the linear map* $\mathcal{B} : \mathbb{S}^{2n+d} \to \mathbb{R}^{|\mathcal{O}|}$ *and vector* $\mathbf{b} \in \mathbb{R}^{|\mathcal{O}|}$ *enforce the inequalities in Equation* (6.13)*.*

Problem 9 is a convex feasibility problem, which can be efficiently solved by numerous interior-point methods (Boyd and Vandenberghe, 2004). Unfortunately, solutions to Problem 9 are not limited to the rank-d solutions originally sought in Problem 8. In fact, Nie (2009) and So and Ye (2007) point out that when there are multiple possible solutions, interior-point SDP solvers return a max-rank solution. For the case of SNL problems with exact measurements and no inequalities, So and Ye (2007) use rigidity theory to prove that the existence of a unique solution in dimension d to an instance of Problem 8 is a sufficient condition for its corresponding SDP relaxation (Problem 9) to yield a rank-d or lower solution. Unfortunately, even though the lengths in our kinematic model are in fact exact "measurements", we are particularly interested in *redundant* kinematic models, which, by definition, admit multiple solutions. Thus, we turn our attention to methods for finding low-rank solutions to Problem 9.

6.3.1 Nearest Point Cost Functions

With Problem 8, we find ourselves in a situation similar to the one in Chapter 5, where the quadratic variety Equation (5.10) was used to describe feasible solutions to planar and spherical inverse kinematics problems. Therefore, we may be tempted to mimic the "nearest-point" strategy used in Equation (5.12) to define a cost function which gives us a unique solution and a tight SDP relaxation. However, this strategy does not work for the model in Problem 8. To understand why, recall that Theorem 3 requires LICQ to hold at some ZDG parameter $\bar{\boldsymbol{\xi}}$. For each pair (i, j) of neighbouring joints, Problem 8 includes the following six pairwise distance constraints:

$$\begin{aligned}
\|\mathbf{p}_i - \mathbf{q}_i\|^2 &= d_1^2 \\
\|\mathbf{p}_i - \mathbf{p}_j\|^2 &= d_2^2 \\
\|\mathbf{p}_i - \mathbf{q}_j\|^2 &= d_3^2 \\
\|\mathbf{q}_i - \mathbf{p}_j\|^2 &= d_4^2 \\
\|\mathbf{q}_i - \mathbf{q}_j\|^2 &= d_5^2 \\
\|\mathbf{p}_j - \mathbf{q}_j\|^2 &= d_6^2.
\end{aligned} \tag{6.25}$$

By collecting Equation (6.25) into the vector-valued function $f_{ij}(\mathbf{x}_{ij})$, where

$$\mathbf{x}_{ij} = \begin{bmatrix} \mathbf{p}_i \\ \mathbf{q}_i \\ \mathbf{p}_j \\ \mathbf{q}_j \end{bmatrix}, \tag{6.26}$$

we see that the Jacobian is

$$\nabla f_{ij} = \begin{bmatrix}
(\mathbf{p}_i - \mathbf{q}_i)^\top & (\mathbf{q}_i - \mathbf{p}_i)^\top & \mathbf{0} & \mathbf{0} \\
(\mathbf{p}_i - \mathbf{p}_j)^\top & \mathbf{0} & (\mathbf{p}_j - \mathbf{p}_i)^\top & \mathbf{0} \\
(\mathbf{p}_i - \mathbf{q}_j)^\top & \mathbf{0} & \mathbf{0} & (\mathbf{q}_j - \mathbf{p}_i)^\top \\
\mathbf{0} & (\mathbf{q}_i - \mathbf{p}_j)^\top & (\mathbf{p}_j - \mathbf{q}_i)^\top & \mathbf{0} \\
\mathbf{0} & (\mathbf{q}_i - \mathbf{q}_j)^\top & \mathbf{0} & (\mathbf{q}_j - \mathbf{q}_i)^\top \\
\mathbf{0} & \mathbf{0} & (\mathbf{p}_j - \mathbf{q}_j)^\top & (\mathbf{q}_j - \mathbf{p}_j)^\top
\end{bmatrix} \in \mathbb{R}^{6 \times 12}. \tag{6.27}$$

Recall that for LICQ to hold, Equation (6.27) must have full rank.[5] The matrix in Equation (6.27) is also known as the *rigidity matrix*, and it plays a central role in the mathematical field of rigidity theory (Singer and Cucuringu, 2010). In general, for a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with incidence matrix $\mathbf{B}(\mathcal{E})$ and a "realization" or assignment of points $\mathbf{X} \in \mathbb{R}^{d \times |\mathcal{V}|}$, the rigidity matrix is (Sun et al., 2015):

$$\mathcal{R}(\mathbf{z}) \triangleq \operatorname{diag}\left(\mathbf{z}_1, \ldots, \mathbf{z}_{|\mathcal{E}|}\right)^\top (\mathbf{B}(\mathcal{E}) \otimes \mathbf{I}_d), \tag{6.28}$$

where

$$\mathbf{z}_k = \mathbf{x}_i - \mathbf{x}_j \tag{6.29}$$

is the relative distance for the kth edge $(i, j) = e_k \in \mathcal{E}$ for all $k \in [|\mathcal{E}|]$. Unfortunately, the requirement from Section 6.2.4 that successive joints possess coplanar axes presents an obstacle for the nearest-point approach.

Proposition 3. *For $\mathbf{x}_{ij}$ such that $\mathbf{p}_i$, $\mathbf{q}_i$, $\mathbf{p}_j$, and $\mathbf{q}_j$ are coplanar, the Jacobian in Equation (6.27) (equivalently, the rigidity matrix) is rank deficient.*

Proof. By the rank-nullity theorem (Strang, 1993), it is sufficient to demonstrate that the matrix in Equation (6.27) has a nullspace of dimension ≥ 6. Every 3D configuration's rigidity matrix has at least six nullspace basis vectors corresponding to the degrees of freedom of a Euclidean transformation applied to all points in the configuration (Singer and Cucuringu, 2010). Since our configuration is coplanar by assumption, there exists some nonzero vector $\mathbf{z}^\perp \in \mathbb{R}^3$ such that

$$\mathbf{z}_k{}^\top \mathbf{z}^\perp = 0 \ \forall k \in [6]. \tag{6.30}$$

Consider perturbations of the form

$$\delta \mathbf{x}_{ij} = \begin{bmatrix} \alpha \mathbf{z}^\perp \\ \mathbf{0}_9 \end{bmatrix} \in \mathbb{R}^{12}, \tag{6.31}$$

where $\alpha \neq 0$. Since, by its fundamental property in Equation (6.30), $\mathcal{R}(\mathbf{z}) \delta \mathbf{x}_{ij} = \mathbf{0}$, our perturbation $\delta \mathbf{x}_{ij}$ is in the nullspace of the rigidity matrix $\mathcal{R}(\mathbf{z})$. Finally, since only one point is perturbed, $\delta \mathbf{x}_{ij}$ does not correspond to a linearized Euclidean movement (i.e., a uniform rotation or translation of the entire configuration). Therefore, we have found a seventh nullspace basis vector and completed our proof. □

In rigidity-theoretic language, Proposition 3 says that Equation (6.27) is *infinitesimally flexible* for a coplanar configuration. This is a well understood and fundamental phenomenon

[5] Since the constraints in Equation (6.25) only depend on the variables in $\mathbf{x}_{ij}$, this is true no matter how many DOF the robot of interest possesses (i.e., the inclusion of the entire robot model cannot change the fact that the rows of this partial Jacobian need to be linearly independent). Therefore, without loss of generality, we will only examine the constraints corresponding to a single pair of joints in isolation.

in the study of rigidity, and it prevents our application of Theorem 3 and its associated SDP solution to revolute 3D manipulators.

6.3.2 Rank Minimization

With the nearest-point formulation that worked for planar and spherical chains in Chapter 5 off the table, we need an alternative approach. Ideally, we could augment Problem 9 with $\operatorname{rank}(\mathbf{Z})$ as its cost function to find the lowest-rank solution possible. However, the matrix rank function is nonconvex and therefore difficult to globally minimize, even over the convex feasible set of Problem 9. Thus, we minimize convex (linear) heuristic cost functions that encourage low rank solutions:

Problem 10 (Problem 9 with a Linear Cost)**.** *Find the symmetric PSD matrix* $\mathbf{Z}$ *that solves*

$$\begin{aligned} \min_{\mathbf{Z} \in \mathbb{S}_+^{2n+d}} \quad & \operatorname{tr}(\mathbf{C}\mathbf{Z}) \\ \text{s.t.} \quad & \mathcal{A}(\mathbf{Z}) = \mathbf{a}, \\ & \mathcal{B}(\mathbf{Z}) \leq \mathbf{b}, \end{aligned} \tag{6.32}$$

where $\mathbf{C} \in \mathbb{S}^{2n+d}$.

When $\mathbf{C} = \mathbf{I}$, the cost function is the nuclear norm of $\mathbf{Z}$, which is the *convex envelope* of $\operatorname{rank}(\mathbf{Z})$ over the set of matrices whose largest singular value is less than or equal to one (Dattorro, 2005). The nuclear norm heuristic has been successfully applied to a variety of linear inverse problems with matrix variables, and is even guaranteed to produce the minimum rank solution when certain conditions are met (Recht et al., 2010; Chandrasekaran et al., 2012). However, we conducted a simple experiment demonstrating that the nuclear norm heuristic is unable to yield rank-d solutions to Problem 9 for the 6-DOF UR10 manipulator. Figure 6.3 shows the distribution of $\log_{10}(h(\mathbf{Z}))$ over 7,000 experiments with feasible UR10 goal poses in an obstacle-free environment, where

$$h(\mathbf{Z}) = \sum_{i=d+1}^{2n+d} \lambda_i(\mathbf{Z}) \tag{6.33}$$

is our "excess rank" heuristic. The convergence threshold for $h(\mathbf{Z})$ we use in Section 6.4 is 10^{-6}, with some high quality solutions recovered for $h(\mathbf{Z}) \approx 10^{-3}$. As you can see in Figure 6.3, the nuclear norm heuristic led to very few instances with $h(\mathbf{Z}) < 10^{-2}$ and is therefore only appropriate as an initialization for `CIDGIK`, which we describe in Section 6.3.3 and typically requires fewer than 10 iterations to converge below our threshold of 10^{-6}.

Consider the following surrogate for the "excess rank" (i.e., $\operatorname{rank}(\mathbf{Z}) - d$) of $\mathbf{Z} \in \mathbb{S}_+^{2n+d}$:

$$h(\mathbf{Z}) = \sum_{i=d+1}^{2n+d} \lambda_i(\mathbf{Z}), \tag{6.34}$$

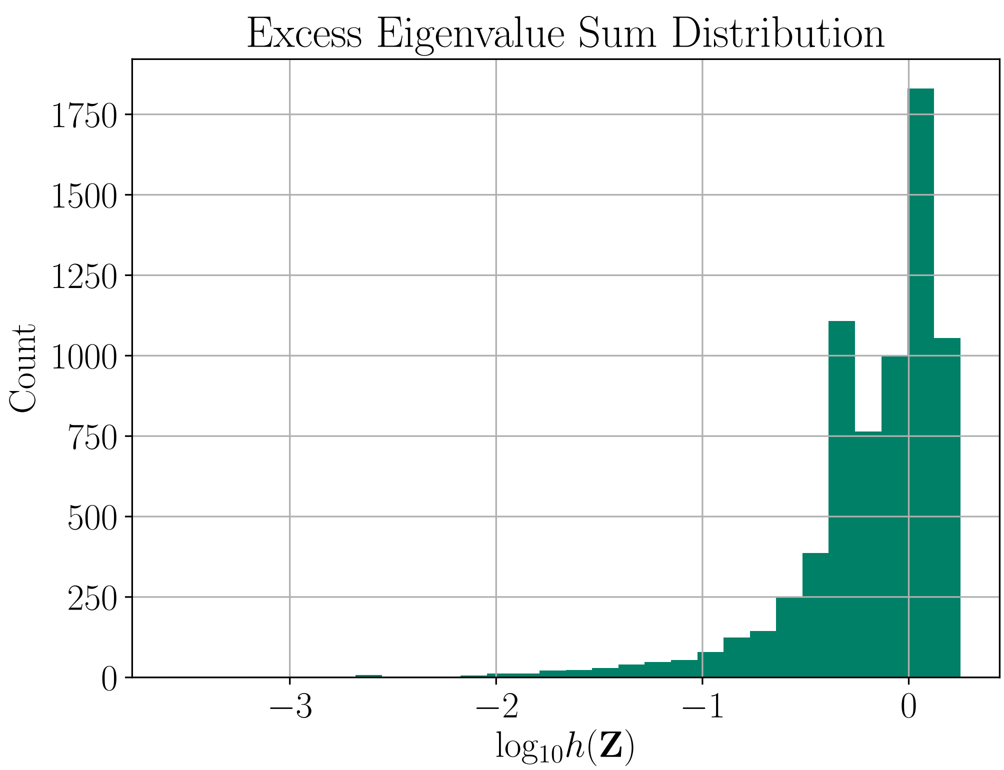


Figure 6.3: Distribution of $\log_{10}(h(\mathbf{Z})$ over 7,000 runs of our SDP formulation with the nuclear norm heuristic ($\mathbf{C} = \mathbf{I}$).

where $\lambda_i(\mathbf{Z})$ is the ith largest eigenvalue of $\mathbf{Z}$. Since $\mathbf{Z}$ has nonnegative eigenvalues, $h(\mathbf{Z}) = 0$ implies that rank $(\mathbf{Z}) \leq d$. Computing $h(\mathbf{Z})$ is equivalent to solving a particular SDP (Dattorro, 2005):

Problem 11 (Sum of $2n$ Smallest Eigenvalues (Dattorro, 2005))**.** *Find the symmetric PSD matrix* $\mathbf{C}$ *that solves*

$$
\begin{aligned}
\sum_{i=d+1}^{2n+d} \lambda_i(\mathbf{Z}) = \min_{\mathbf{C} \in \mathbb{S}^{2n+d}} \quad & \operatorname{tr}(\mathbf{CZ}) \\
\text{s.t.} \quad & \operatorname{tr}(\mathbf{C}) = 2n, \\
& \mathbf{0} \preccurlyeq \mathbf{C} \preccurlyeq \mathbf{I}.
\end{aligned} \tag{6.35}
$$

A closed-form solution to Problem 11 is given by Dattorro (2005):

$$
\begin{aligned}
\mathbf{C}^\star &= \mathbf{U}\mathbf{U}^\mathsf{T}, \\
\mathbf{U} &= \mathbf{Q}(:,\ d+1 : 2n+d),
\end{aligned} \tag{6.36}
$$

where $\mathbf{Q} \in \mathrm{O}(2n+d)$ is from the eigendecomposition $\mathbf{Z} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\mathsf{T}$.

6.3.3 Convex Iteration

In Dattorro (2005), the method of *convex iteration* between Problem 10 and Problem 11 is proposed. We summarize the approach in Algorithm 1. Each iteration of Problem 11 computes $\mathbf{C}^{\{k\}}$ corresponding to $h(\mathbf{Z})$ at the current iteration's value of $\mathbf{Z} = \mathbf{Z}^{\{k\}}$. Since this $\mathbf{C}^{\{k\}}$ is only exact at $\mathbf{Z}^{\{k\}}$, each iteration of Problem 10 can therefore be treated as minimizing an approximation of $h(\mathbf{Z})$ in the neighbourhood of $\mathbf{Z}^{\{k\}}$. Since the closed-form solution in Equation (6.36) is used to quickly solve Problem 11 in this procedure, most of `CIDGIK`'s computational cost comes from solving Problem 10. This approach has been successfully applied to noisy SNL (Dattorro, 2005) and optimal power flow problems (Wang and Yu, 2018).

Algorithm 1: Convex Iteration for Distance Geometric IK (`CIDGIK`)

Input: Problem 10 specification (i.e., $\mathcal{A}(\cdot), \mathcal{B}(\cdot), \mathbf{a}, \mathbf{b}$)
Result: PSD matrix $\mathbf{Z}^\star$ that solves Problem 10
Initialize $\mathbf{C}^{\{k\}} = \mathbf{I}_{2n+d}$
while ***not converged*** **do**
  Solve $\mathbf{Z}^{\{k\}} = \operatorname{argmin}_{\mathbf{Z}}$ Problem 10 with $\mathbf{C} = \mathbf{C}^{\{k\}}$
  Solve $\mathbf{C}^{\{k\}} = \operatorname{argmin}_{\mathbf{C}}$ Problem 11 with $\mathbf{Z} = \mathbf{Z}^{\{k\}}$ using Equation (6.36)
end
Return $\mathbf{Z}^\star = \mathbf{Z}^{\{k\}}$

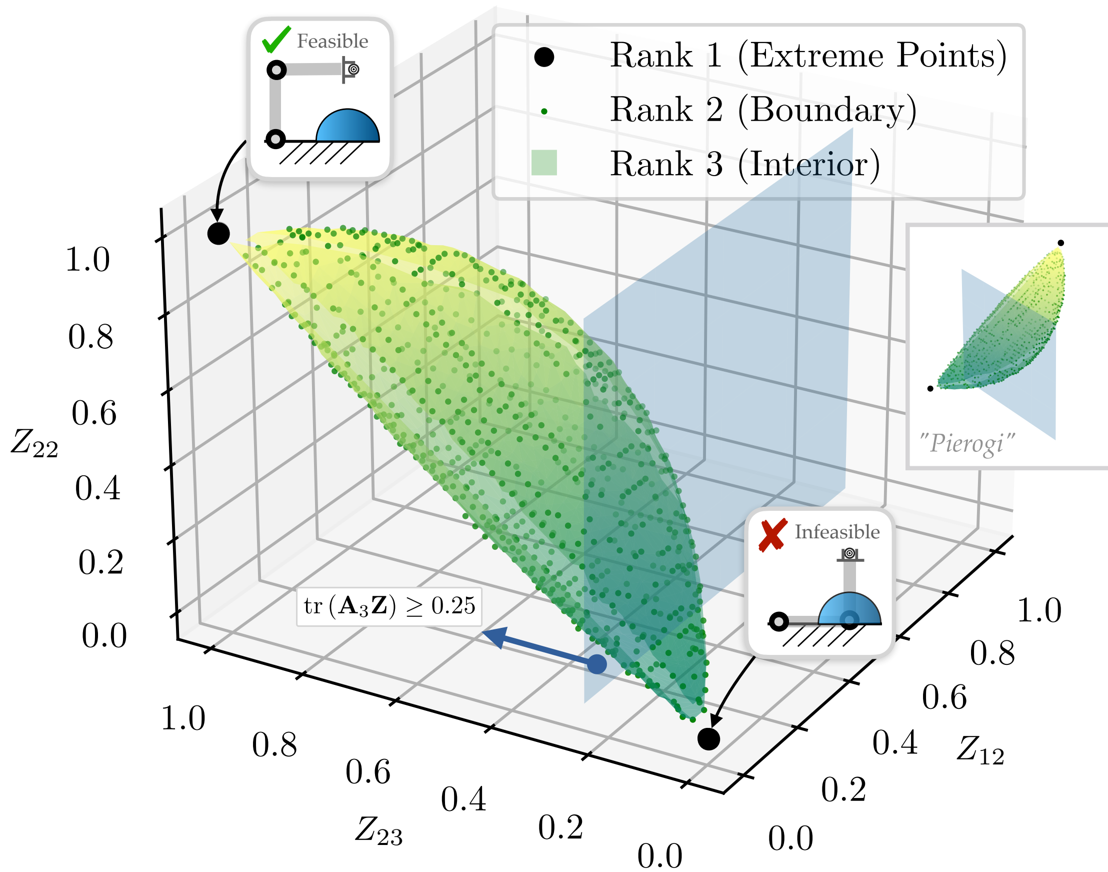


Figure 6.4: Spectrahedron (in the shape of a *pierogi*) describing the feasible set of Problem 9 for the simple 2-DOF planar manipulator IK problem in Section 6.3.4. The spectrahedron's interior (green gradient) represents the set of full-rank 3×3 PSD matrices $\mathbf{Z}$ that satisfy the constraints (linear in $\mathbf{Z}$) imposed by manipulator geometry and end-effector pose. The spectrahedron boundary (green dots) contains all degenerate $\mathbf{Z}$, and the extreme points (black dots) represent the desired rank-1 solutions. The constraint imposed by the circular obstacle is enforced in the lifted PSD representation by a half-space constraint induced by the translucent blue plane.

6.3.4 Geometric Interpretation

Here, we motivate the convex iteration algorithm described in Section 6.3.2 and explain why we expect some $\mathbf{C} \in \mathbb{S}_+^{2n+d}$ to yield a low-rank solution. Aside from the interpretation of $\mathrm{tr}(\mathbf{CZ})$ as a local approximation of the excess rank heuristic $h(\mathbf{Z})$, it is fruitful to view $\mathbf{C}$ as a *direction* in the space $\mathbb{S}^{2n+d} \supset \mathbb{S}_+^{2n+d}$. More precisely: since $\partial \mathrm{tr}(\mathbf{CZ}) / \partial \mathbf{Z} = \mathbf{C}$, we are in effect *designing* the objective in Problem 10 so that its steepest descent direction at Z points towards *low-rank* minimizers on the boundary of the feasible set. In practice, this heuristic chooses rank-d matrices with high probability for typical IK problems.

Consider the toy problem of a 2-DOF planar manipulator rooted at the origin of the plane and with links of unit length. Using the formulation in Section 6.2, we can write the IK problem of reaching a point $\mathbf{w} \in \mathbb{R}^2$ with the manipulator's end-effector as the following quadratic feasibility problem:

$$
\begin{aligned}
\text{find} \quad & \mathbf{x} \in \mathbb{R}^2 \\
s.t. \quad & \|\mathbf{x}\|^2 = 1, \\
& \|\mathbf{x} - \mathbf{w}\|^2 = 1, \\
& \|\mathbf{x} - \mathbf{o}\|^2 \geq 0.25,
\end{aligned}
\tag{6.37}
$$

where $\mathbf{x}$ is the position of the "elbow" joint, and $\mathbf{o} = [1,\ 0]^\top$ is the position of a unit-diameter circular obstacle. Consider the case of $\mathbf{w} = [1,\ 1]^\top$: the insets of Figure 6.4 show that of the two candidate solutions to this problem, the "elbow down" configuration in the bottom right collides with the obstacle at $\mathbf{o}$ (partially depicted as a blue semicircle). Homogenizing (6.37) with $s^2 = 1$ and lifting to the rank-1 matrix variable

$$
\mathbf{Z}(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ s \end{bmatrix} \begin{bmatrix} \mathbf{x}^\top & s \end{bmatrix}
\tag{6.38}
$$

lets us apply the SDP relaxation $\mathbf{Z} \succeq \mathbf{0}$ to yield:

$$
\begin{aligned}
\text{find} \quad & \mathbf{Z} \in \mathbb{S}^3_+ \\
s.t. \quad & \mathcal{A}(\mathbf{Z}) = \mathbf{a}, \\
& \operatorname{tr}(\mathbf{B}\mathbf{Z}) \leq b.
\end{aligned}
\tag{6.39}
$$

In Equation (6.37), the homogenization equation and the constraints of (6.37) have been replaced by their lifted SDP equivalents in $\mathcal{A}$. Specifically, the matrices

$$
\mathbf{A}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix}
\tag{6.40}
$$

describe the unit link length constraints and matrices

$$
\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{A}_3 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}
\tag{6.41}
$$

describe the homogenization equation ($s^2 = 1$) and the obstacle avoidance constraint, respectively.

The intersection of the three affine equality constraints with $\mathbf{Z} \succeq \mathbf{0}$ produce the *spectrahedron* in Figure 6.4. Valid solutions to the unrelaxed QCQP problem are rank-1 elements of this spectrahedron. Since the interior of this set ($\mathbf{Z} \succ \mathbf{0}$) contains full rank solutions, we know that rank-1 solutions will lie on the boundary. Indeed, the two extreme points, denoted in black in Figure 6.4, represent the two valid solutions to our toy IK problem in the absence of obstacles. The half-space constraint induced by the blue plane illustrates the effect of the obstacle constraint in our lifted problem domain, eliminating the infeasible "elbow down" configuration from the feasible set of (6.39). The method employed in this chapter seeks to find, via convex iteration, a "direction" $\mathbf{C}$ such that the linear cost function $\operatorname{tr}(\mathbf{CZ})$ is minimized at a rank-d solution. While the solution to this toy example can be obtained analytically, its low-dimensional structure allows us to illustrate the geometric ideas motivating our approach.

6.4 Experiments

We evaluate our proposed approach on IK problems for the three commercial robots shown in Figure 6.5 in a variety of environments. In all cases, we generate *feasible* IK problems by taking uniform random sample configurations $\boldsymbol{\theta} \in \mathcal{C}$, rejecting configurations that violate collision avoidance constraints,[6] and using the resulting end-effector pose $\mathbf{T}^{\mathrm{g}} = F(\boldsymbol{\theta})$ as a goal. Each problem instance is solved with Algorithm 1 (`CIDGIK`), with the MOSEK interior point solver (Andersen and Andersen, 2000) used for each iteration of Problem 10. To determine whether Algorithm 1's solution to a particular IK problem is successful, we first use the procedure in Marić et al. (2021) to reconstruct the joint configuration $\boldsymbol{\theta}$ from points $\mathbf{X}$ extracted from the rank-d matrix $\mathbf{Z}^{\star}$ returned after a maximum of 10 iterations of the algorithm in Section 6.3.3. This joint configuration is treated as `CIDGIK`'s solution and fed into the forward kinematics in Equation (6.1) to obtain the end-effector pose and any workspace constraint violations. A solution is considered correct when it satisfies obstacle constraints to within a 0.01 m tolerance, has an end-effector position error lower than 0.01 m, and has an orientation error lower than 0.01 rad (0.6°).

In order to evaluate the advantages of our approach over formulations based on joint angles, we also implement an IK solver that uses nonlinear optimization. Namely, we use the square of the end-effector pose error, $\mathbf{e}$, as the objective function of a nonlinear program, with collision avoidance constraints equivalent to those used by `CIDGIK` represented as nonlinear inequality constraints:

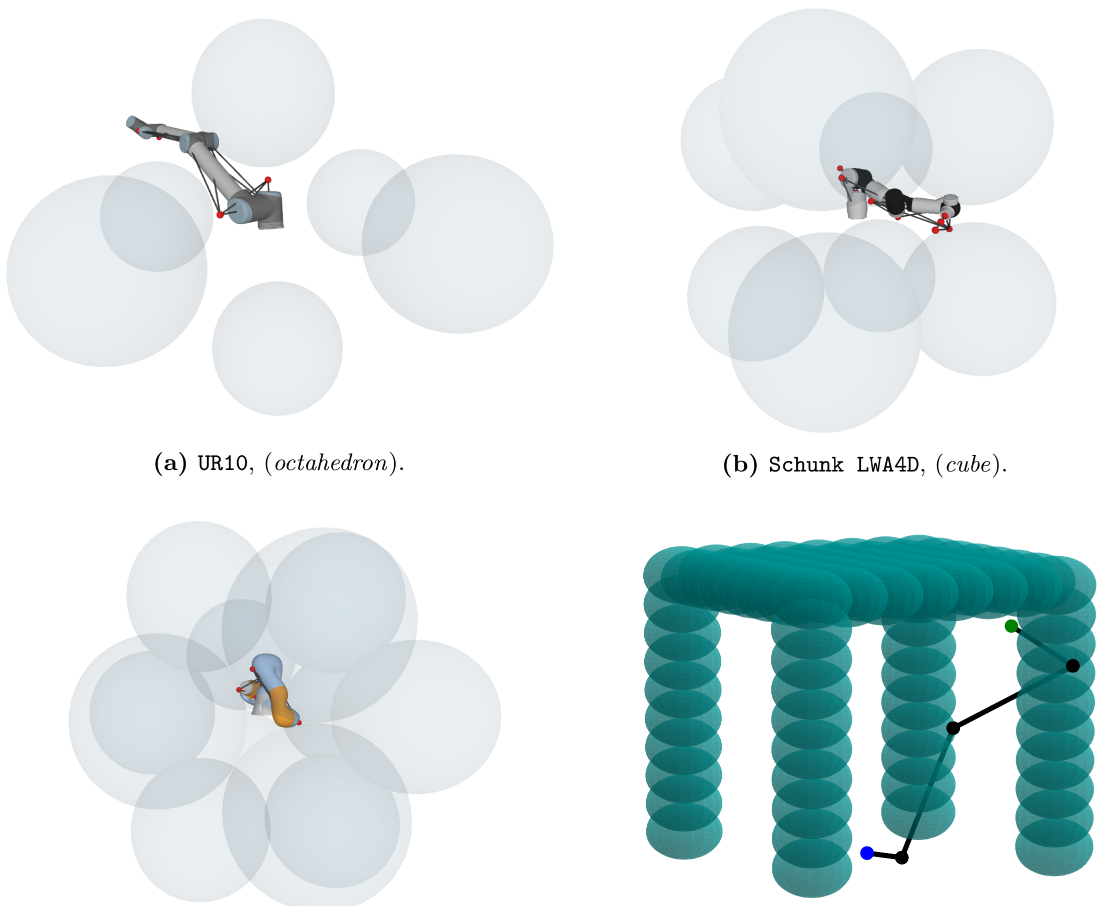
$$\begin{aligned} \min_{\boldsymbol{\theta} \in \mathcal{C}} \quad & \left\| \mathbf{e}\left(\boldsymbol{\theta}, \mathbf{T}^{\mathrm{g}}\right) \right\|^2 \\ s.t. \quad & \left\| \mathbf{x}_i(\boldsymbol{\theta}) - \mathbf{c}_j \right\|^2 \geq l_j^2 \;\; \forall i \in \mathcal{V}_{\mathrm{j}}, \;\; \forall j \in \mathcal{O}. \end{aligned} \tag{6.42}$$

Our choice of error function $\mathbf{e}$ between the current ($F(\boldsymbol{\theta})$) and goal ($\mathbf{T}^{\mathrm{g}}$) poses is the matrix logarithm

$$\mathbf{e}\left(\boldsymbol{\theta}, \mathbf{T}^{\mathrm{g}}\right) = \ln(F(\boldsymbol{\theta})^{-1}\mathbf{T}^{\mathrm{g}}) \in \mathbb{R}^6, \tag{6.43}$$

[6] Note that for redundant manipulators this procedure can reject some feasible goals, since (infinitely) many other configurations may exist that reach the desired end-effector pose without any collisions.

which is a proper function over SE(3) when defined with a convention that selects for a rotation component with a magnitude smaller than π (Barfoot, 2017, Section 7.1.3). The spherical obstacles in $\mathcal{O}$ are parameterized as in Equation (6.13). This formulation can be solved using



(a) `UR10`, (*octahedron*).

(b) `Schunk LWA4D`, (*cube*).

(c) `KUKA IIWA`, (*icosahedron*).

(d) `9-DOF`, (*table*).

Figure 6.5: Four robot manipulators (and obstacle configurations) used in our experiments. For (a)-(c), we also visualize the edges (dark grey lines) and nodes (red) of the associated acyclic graph used to form our distance-geometric model (cf. Figure 6.2b). These three environments, labeled *octahedron*, *cube*, and *icosahedron*, correspond to obstacle configurations defined by the vertices of the respective Platonic solid centred at the base of the robot.

Sequential Quadratic Programming (SQP) and has previously been used within the TRAC-IK algorithm (Beeson and Ames, 2015). Since TRAC-IK does not support obstacles, we implement our solver using the `SLSQP` routine in the `scipy` Python package (Virtanen et al., 2020). As an additional comparison, we solve Equation (6.42) using `IPOPT`, a highly efficient implementation of an interior-point filter line-search algorithm (Wächter and Biegler, 2006). All experiments are performed on a laptop with an Intel i7-8750H CPU running at 2.20 GHz and with 16 GB of RAM.

Env.	Free			Octahedron			Cube			Icosahedron		
	SLSQP	IPOPT	CIDGIK	SLSQP	IPOPT	CIDGIK	SLSQP	IPOPT	CIDGIK	SLSQP	IPOPT	CIDGIK
UR10	89.9 ± 1.1	**100 ± 0.0**	90.8 ± 1.0	56.3 ± 1.8	47.8 ± 1.8	**76.2 ± 1.5**	57.9 ± 1.8	46.7 ± 1.8	**67.1 ± 1.7**	15.2 ± 1.3	15.4 ± 1.3	**51.6 ± 1.8**
KUKA	98.1 ± 0.5	**99.9 ± 0.1**	99.7 ± 0.2	97.6 ± 0.5	66.6 ± 1.7	**99.8 ± 0.2**	95.5 ± 0.7	61.4 ± 1.7	**99.2 ± 0.3**	93.1 ± 0.9	64.4 ± 1.7	**99.6 ± 0.2**
LWA4D	97.0 ± 0.6	**100 ± 0.0**	99.5 ± 0.3	95.6 ± 0.7	52.2 ± 1.8	**99.8 ± 0.2**	95.6 ± 0.7	53.4 ± 1.8	**99.7 ± 0.2**	93.6 ± 0.9	53.3 ± 1.8	**99.7 ± 0.2**

Table 6.2: Percentages of successfully solved problems as 95% Jeffreys confidence intervals (Tony Cai, 2005).

Env.	Free			Octahedron			Cube			Icosahedron		
	SLSQP	IPOPT	CIDGIK	SLSQP	IPOPT	CIDGIK	SLSQP	IPOPT	CIDGIK	SLSQP	IPOPT	CIDGIK
UR10	0.07 (0.02)	**0.01 (0.00)**	0.24 (0.20)	0.36 (0.14)	2.3 (2.1)	**0.16 (0.11)**	0.43 (0.17)	2.6 (2.1)	**0.18 (0.12)**	0.40 (0.19)	1.6 (1.8)	**0.20 (0.12)**
KUKA	**0.09 (0.04)**	0.74 (0.72)	0.16 (0.07)	0.58 (0.24)	4.0 (2.4)	**0.14 (0.07)**	0.64 (0.26)	4.7 (2.5)	**0.14 (0.08)**	0.87 (0.40)	4.6 (2.5)	**0.14 (0.07)**
LWA4D	**0.10 (0.04)**	0.88 (0.78)	0.15 (0.07)	0.58 (0.22)	5.1 (2.1)	**0.13 (0.06)**	0.64 (0.25)	5.2 (2.1)	**0.13 (0.06)**	0.91 (0.46)	12.8 (5.4)	**0.13 (0.06)**

Table 6.3: Mean, with standard deviation in brackets, of solution times in seconds. For each algorithm, "solution time" does not include problem setup time. For `CIDGIK`, the sum of solver times returned by each iteration of both Problem 10 and Problem 11 is used.

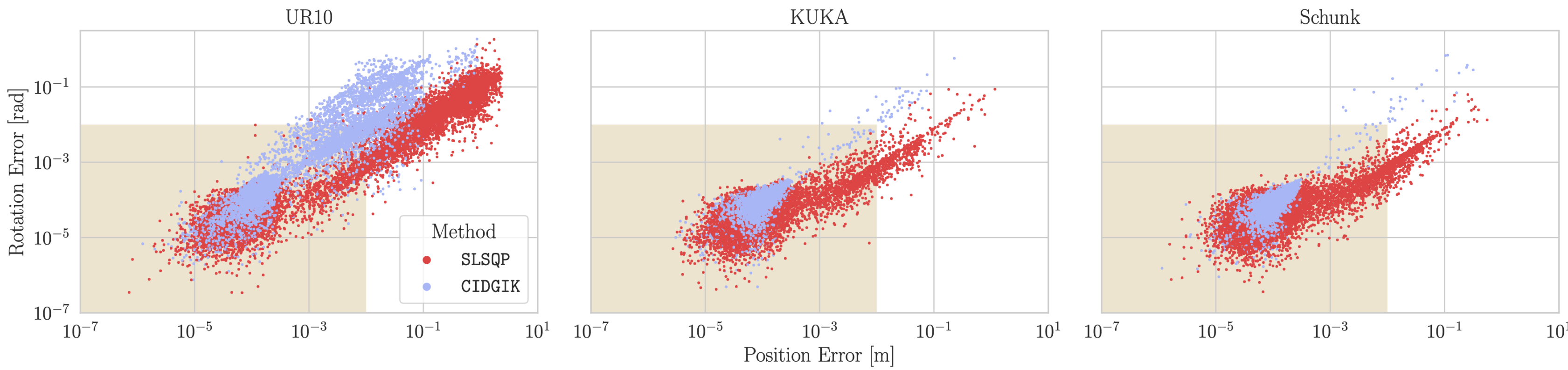


Figure 6.6: A comparison of the end-effector position and orientation errors. Each sub-plot contains problem instances from all four obstacle environments. The shaded rectangle indicates the region of error tolerance (i.e., contains all successful runs).

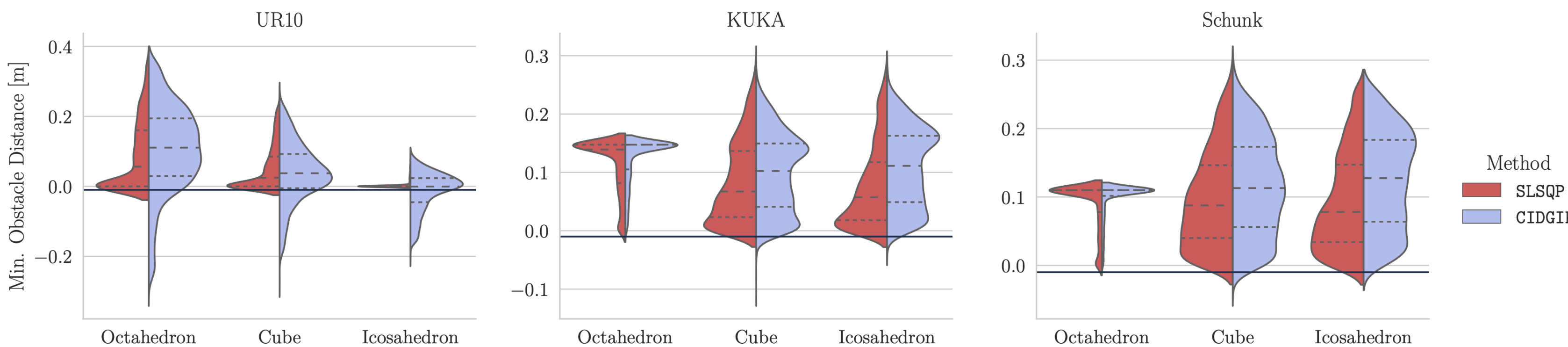


Figure 6.7: Violin plots describing the distribution of the minimum distance to an obstacle for different algorithms, robots, and environments. A negative value indicates that one or more of the robot's control points fell within an obstacle. The dashed lines are the quartile boundaries of each distribution, and the solid dark line is the error tolerance (0.01 m). Note that the performance of `SLSQP` on the UR10 in the *icosahedron* environment has extremely low variance around zero.

To provide a quantitative evaluation of our approach, we generate and solve 3,000 problem instances for each of the three robots depicted in Figures 6.5a-6.5c in each of the distinct environments depicted in those same figures, as well as a fourth obstacle-free environment. For consistency, all three algorithms only consider collisions between joint locations and obstacles (i.e., auxiliary points are not added along the links in `CIDGIK`). Tables 6.2 and 6.3 summarize respectively the success rate and runtime of each algorithm for each robot-environment pair.

Comparing success rates in Table 6.2 reveals that our method solves a larger percentage of problems overall than both `SLSQP` and `IPOPT` in all environments featuring obstacles, achieving a success rate of $> 99\%$ in many instances. Likewise, `CIDGIK`'s runtime in Table 6.3 is consistently lower than its competitors' across all three obstacle-filled environments. In the obstacle-free case, `SLSQP` runs faster than `CIDGIK` with similar accuracy, and `IPOPT` achieves a perfect success rate and extremely low mean runtime on the UR10, which the other two algorithms struggle with. Curiously, while its competitors slow down significantly with the addition of obstacles, the solution times for `CIDGIK` barely increase (or even slightly *decrease*) as the number of obstacles grows. In future work, we plan to investigate whether this trend holds for other more complex robots, and for environments with many hundreds or thousands of obstacles.

Figure 6.6 displays the distribution of position and rotation errors for `CIDGIK` and `SLSQP` on each robot, aggregated across experiments from all four environments. Figure 6.7 depicts the distribution of the distance to the nearest obstacle across experiments. Together, these two figures reveal that with the exception of `CIDGIK` on the UR10 robot, most failures were in fact due to end-effector error as opposed to obstacle constraint violations.

Finally, Table 6.4 summarizes 1,000 experiments comparing `CIDGIK` and `SLSQP` on the hyper-redundant 9-DOF manipulator from Xu et al. (2014) in the 100-obstacle *table* environment displayed in Figure 6.5d. These results demonstrate `CIDGIK`'s statistically significant superiority over `SLSQP`, both in terms of accuracy and runtime, as both the number of obstacles and robot DOF increase.

Method	Pos. Err. [cm]	Rot. Err. [mrad]	Sol. Time [s]	Succ. [%]
`SLSQP`	1.6 (12)	1 (14)	1.5 (0.7)	97 ± 1.1
`CIDGIK`	0.02 (0.21)	0.13 (2)	0.16 (0.1)	99 ± 0.4

Table 6.4: Results for 1,000 9-DOF manipulator experiments in the *table* environment (Figure 6.5d). Errors and and solve times are reported as mean with standard deviation in brackets, while the success rates are given as 95% Jeffreys confidence intervals (Tony Cai, 2005).

6.4.1 SDP Solver Performance

This section provides a comparison of two SDP solvers: SCS, a conic operator splitting method (O'Donoghue et al., 2016), and MOSEK, an interior point optimizer (Andersen and Andersen, 2000). These solvers were selected because they use different approaches, and because of their interfaces with the `cvxpy` modelling language (Diamond and Boyd, 2016) used to implement

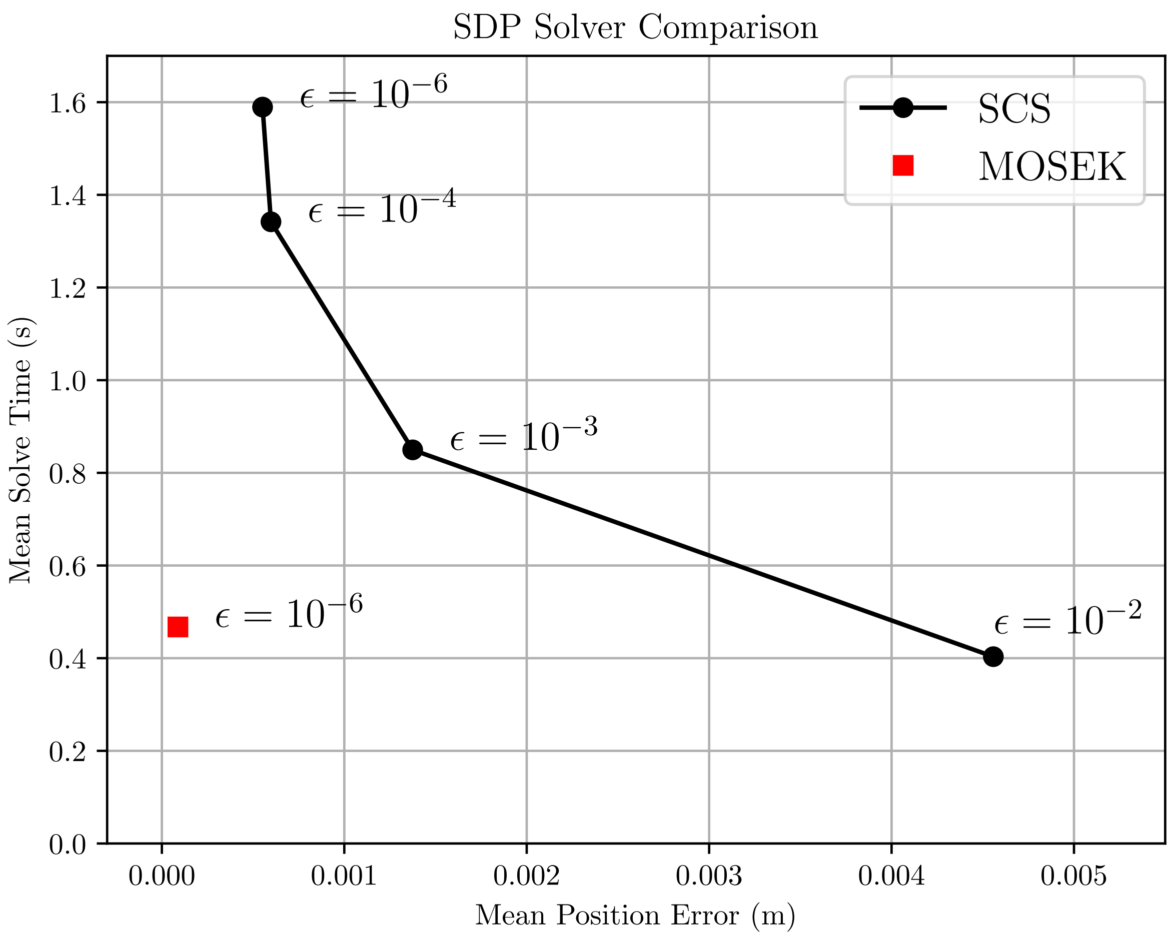


Figure 6.8: A comparison of MOSEK (used in this paper's experiments) with SCS for multiple values of ϵ, the convergence tolerance. The reported mean solve times and position errors are over 100 randomly generated IK problems. Each algorithm was allowed up to 1,000 iterations.

`CIDGIK`. Figure 6.8 demonstrates that MOSEK is the far superior option on a dataset of 100 randomly generated IK problems using the 9-DOF manipulator in the *table* environment. In spite of its use of warm starting, the performance of SCS is dominated by MOSEK. SCS is only able to run faster than MOSEK when it tolerates end-effector position errors an order of magnitude greater than MOSEK's. These results motivate our decision to use MOSEK as `CIDGIK`'s backend for all experiments.

Finally, in order to highlight the importance of choosing a fast SDP solver, we measured the total runtime of the closed-form eigendecomposition solution when solving the 100 IK problem instances with MOSEK. The mean runtime of this procedure is 0.56 ms, with a standard deviation of 0.26 ms. This is dwarfed by the nearly half-second runtime of MOSEK on the main SDP.

6.4.2 Infeasibility Certification

Convex relaxations can often be used certify the infeasibility of a problem. To determine whether `CIDGIK` is capable of infeasibility certification, we conducted a simple experiment. Using the *Cube* environment and a closed-form analytic solver for the UR10, we gave `CIDGIK` 10,000 goal poses with the following properties:

1. the analytic UR10 solver deemed the goal pose *infeasible* by checking for collisions over all 8 possible solutions,

2. the end-effector was not in collision with an obstacle, and

3. the point attached to the end-effector's parent joint was not in collision with an obstacle.

Goal poses that are infeasible because of conditions 2) and 3) are easy for `CIDGIK` to certify as infeasible because the Euclidean distance constraints they violate are unavoidable, even in higher-rank solutions representing configurations embedded in $\mathbb{R}^{d'}$ where $d' > d$. Unfortunately, the infeasible cases that remain are much more challenging, with `CIDGIK` certifying a mere 152 out 10,000 (1.52%) as infeasible.

6.5 Summary and Future Work

We have presented a novel distance-geometric approach to solving inverse kinematics problems involving redundant manipulators with arbitrary spherical and planar workspace constraints. Preliminary experiments demonstrate that our algorithm significantly outperforms benchmark algorithms in obstacle-laden environments. Crucially, our problem formulation connects IK to the rich literature on SDP relaxations for distance geometry problems, providing us with the novel and elegant geometric interpretation of IK discussed in Section 6.3.4.

6.5.1 Rank-1 Relaxation and Ellipsoids

Like SNL, our IK problem leads to a rank-d SDP relaxation because of the d-dimensional nature of our spatial problem. More concretely, we saw in the previous section that since we are only constraining the squared *distances* between points in $\mathbb{R}^d$, quadratic terms in these constraints appear as inner products between variables (e.g., $\mathbf{x}_i^\mathsf{T}\mathbf{x}_j$). In general, QCQPs do not contain this structure, and the lifted $\mathbf{Z}$ variable would be a "rank-1" lifting. For our problem, a rank-1 SDP relaxation, like the one used for the *pierogi* example would take the form

$$\mathbf{Z} = [\mathrm{vec}\,(\mathbf{X})^\mathsf{T} \;\; 1]^\mathsf{T}[\mathrm{vec}\,(\mathbf{X})^\mathsf{T} \;\; 1] \in \mathbb{S}_+^{2dn+1}, \tag{6.44}$$

. As you can see, this $\mathbf{Z}$ variable is much larger, leading to greater computation and memory requirements in SDP solvers, and our rank-d relaxation, by contrast, exploits the spatial structure of our problem. However, since the generic rank-1 relaxation allows us to work with any quadratic constraint, in future work we are interested in exploring its ability to model constraints like arbitrary ellipsoidal obstacles:

$$(\mathbf{x}_i - \mathbf{c})^\mathsf{T}\mathbf{A}(\mathbf{x}_i - \mathbf{c}) \geq l^2. \tag{6.45}$$

6.5.2 Chordal Sparsity

A graph is *chordal* if every cycle of length four or greater has a chord (an edge that is not part of the cycle but connects two vertices in the cycle). In our problem, the edges of $\mathcal{G}$ describe points whose distances are constrained. Since the SDP variable $\mathbf{Z}$ contains the dot product of point variables (i.e., $Z_{ij} = \mathbf{x}_i{}^\top \mathbf{x}_j$), $\mathcal{G}$ also describes the elements of $\mathbf{Z}$ that appear together in constraints. In this section, we demonstrate that $\mathcal{G}$ is chordal due to a sparsity pattern induced by the fact that distance constraints only affect points fixed to neighbouring joints. This "chordal sparsity" can be exploited to speed up the solution of large SDPs with the methods reviewed in Majumdar et al. (2019). In future work, we intend to use these methods within `CIDGIK` to quickly solve IK problems for tree-like robots with many redundant degrees of freedom.

Assume that $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\mathrm{eq}}, \ell)$ is the distance constraint graph for a tree-like robot (i.e., no loops in its joints). Since the robot's joints and links describe a tree, the only cycles in $\mathcal{G}$ occur in the cliques described by $\mathbf{p}_i$, $\mathbf{q}_i$, $\mathbf{p}_j$, and $\mathbf{q}_j$ for neighbouring joints i and j. Since the cycles occur within cliques, they are chordal (because cliques are complete by definition). In order to apply the methods in Majumdar et al. (2019), we would additionally need to ensure that the workspace constraints and cost function matrix $\mathbf{C}$ also exhibit this sparsity pattern. Spherical and planar obstacle constraints clearly do not ruin the chordal sparsity, since each constraint only involves a single point $\mathbf{p}_i$. Finally, the work on optimal power flow in Wang and Yu (2018) describes a heuristic method for constructing $\mathbf{C}$ in a manner that preserves problem sparsity.

The COSMO algorithm (Garstka et al., 2019) has a freely available implementation that is able to automatically exploit chordal sparsity in conic optimization problems.

6.5.3 Joint Angle Limits

Quadratic joint angle limits for revolute manipulators can be defined in a manner similar to the method used in Chapter 5 on planar and spherical chains. Unfortunately, without the addition of auxiliary variables, joint angle ranges which are not symmetric about the "straightened" configuration are impossible to enforce. Crucially, this precludes the use of joints that behave like fingers or knees (i.e., only able to bend in one direction away from the straightened configuration) in IK problems we wish to solve with `CIDGIK`. This need for "directionality" in angular constraints can be addressed with the addition of constraints involving the cross product of differences between adjacent points, which we leave for future work.

6.5.4 "Tetrahedral" Robot Link Geometry

The work in this chapter assumes that consecutive joints have axes of rotation that are coplanar (i.e., either parallel or intersecting). This requirement ensures that for consecutive joints i and j, the points $\mathbf{p}_i, \mathbf{q}_\mathbf{i}, \mathbf{p}_j$, and $\mathbf{q}_j$ form a quadrilateral or triangle. When the axes are *not* coplanar, these points form a tetrahedron that we are specifying from distances alone. Generic tetrahedra are chiral in that their reflections cannot be reproduced by rigid transformations. Therefore, our method of enforcing a tetrahedral joint pair's structure with distances alone allows `CIDGIK` to search a feasible set that contains physically unrealizable configurations. Much like asymmetrical joint limits, this problem can be addressed with the addition of auxiliary variables and constraints involving cross-products, but the effect of these additions on the accuracy and performance of `CIDGIK` remains to be seen.

These modifications can be incorporated into our framework, but their effect on accuracy and runtime remains to be seen. Finally, while our method uses global optimization to solve the *subproblem* in each iteration, it remains to be shown whether global convergence guarantees exist for `CIDGIK`. These guarantees may depend on robot structure or hyperparameter settings (e.g., a particular choice of $\mathbf{C}^{\{0\}} \neq \mathbf{I}$), and modifications might affect which of the possibly infinite feasible solutions `CIDGIK` returns. Understanding this behaviour is key to making `CIDGIK` a fast and reliable subroutine for a variety of challenging motion planning applications.

Chapter 7

Conclusion

> But then, as she knew too well, the more fondly we imagine something will last forever, the more ephemeral it often proves to be.
>
> IAIN M. BANKS, *Excession*

This dissertation has applied convex semidefinite programming relaxations to variants of two classic problems in robotics, extrinsic sensor calibration and inverse kinematics. To do so, we exploited quadratic polynomial representations of two fundamental structures in Euclidean space: orientations and sets of points with fixed distances. In fact, it is useful to think of rotation matrices described by Equation (2.2) as sets of points with fixed distances imbued with additional orthogonality ($\mathbf{p}^\top \mathbf{q} = 0$) and handedness (Equation (2.3)) constraints. In addition to presenting concrete experimental results, we hope that this work has strengthened the elegant theory of global polynomial optimization for geometric problems in robotics by identifying a thread connecting extrinsic calibration and inverse kinematics. To conclude, this section contains a list of novel contributions and a brief discussion of two potential high-level directions for future work.

7.1 Summary of Contributions

In summary, this dissertation's main novel contributions are:

1. an extension of the theory in Cifuentes et al. (2022) to semialgebraic sets (Section 3.6);
2. a maximum likelihood QCQP formulation of hand-eye calibration (Section 4.2);
3. a proof of the local stability of SDP relaxations of our QCQP formulation of hand-eye calibration (Section 4.3);

4. an open source extrinsic calibration package in Python and MATLAB with experimental results (Section 4.4);

5. a QCQP formulation of inverse kinematics for serial chains (Section 5.3);

6. a proof of the local stability of SDP relaxations of our QCQP formulation of inverse kinematics (Section 5.3.3);

7. an open source MATLAB package implementing SOS-IK, a sparse sum-of-squares solution to IK for serial chains and its associated experiments (Section 5.5);

8. an extension of our QCQP formulation of IK to revolute manipulators with pairwise-coplanar axes of rotation in environments with workspace collision constraints (Section 6.2);

9. an application and analysis of low-rank semidefinite programming relaxations to our QCQP formulation of IK resulting in `CIDGIK`, a fast and accurate solution method (Section 6.3); and

10. an open source Python package implementing `CIDGIK` and its associated experiments (Section 6.4).

7.2 Future Research Directions

Our focus on SDP relaxations enabled the use of optimization techniques with global optimality guarantees. While this spurred the development of fast and accurate methods that do not require an initialization, our theoretical analysis of both calibration and IK had a decidedly *local* flavour stemming from use of the machinery in Cifuentes et al. (2022). This limitation is perhaps unsurprising, as we should not expect the use of SDP relaxations to be a "silver bullet" for problems which are closely related to NP-hard problems (Antonante et al., 2021; Aspnes et al., 2004; Canny, 1988). In other words, there is "no free lunch" (Wolpert and Macready, 1997) in robotic state estimation and planning! However, better characterizing the performance and applicability of this approach, through, for example, the computation of ZDG-region bounds (Cifuentes et al., 2022) or detailed geometric descriptions (Cifuentes et al., 2018) remains an exciting future research direction.

One promising use for SDP relaxations in robotics is as a complementary technology to data-driven approaches. This direction is enabled by fast and *differentiable* solvers for convex programs which can be treated as layers in deep neural networks (Agrawal et al., 2019). Indeed, this inspired our recent work in Peretroukhin et al. (2020), which indirectly predicts rotations by learning quadratic *cost* functions over unit quaternions from perceptual data. A similar strategy is used in Yang et al. (2021) for learning visual feature descriptors for robust geometric perception.

Bibliography

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., and Kolter, J. Z. (2019). Differentiable Convex Optimization Layers. *Advances in Neural Information Processing Systems*, 32.

Alipanahi, B., Krislock, N., Ghodsi, A., Wolkowicz, H., Donaldson, L., and Li, M. (2013). Determining Protein Structures from NOESY Distance Constraints by Semidefinite Programming. *Journal of Computational Biology*, 20(4):296–310.

Andersen, E. D. and Andersen, K. D. (2000). The MOSEK Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm. In *High Performance Optimization*, volume 33, pages 197–232. Boston, MA.

Andreff, N., Horaud, R., and Espiau, B. (2001). Robot Hand-Eye Calibration Using Structure-from-Motion. *The International Journal of Robotics Research*, 20(3):228–248.

Anstreicher, K. and Wolkowicz, H. (2000). On Lagrangian Relaxation of Quadratic Matrix Constraints. *SIAM Journal on Matrix Analysis and Applications*, 22(1):41–55.

Antonante, P., Tzoumas, V., Yang, H., and Carlone, L. (2021). Outlier-Robust Estimation: Hardness, Minimally Tuned Algorithms, and Applications. *IEEE Transactions on Robotics*, 38(1):281–301.

Apostol, T. M. (1974). *Mathematical Analysis*. Addison Wesley, second edition.

Aristidou, A., Lasenby, J., Chrysanthou, Y., and Shamir, A. (2018). Inverse Kinematics Techniques in Computer Graphics: A Survey. *Computer Graphics Forum*, 37(6):35–58.

Aspnes, J., Goldenberg, D., and Yang, Y. R. (2004). On the Computational Complexity of Sensor Network Localization. In *Algorithmic Aspects of Wireless Sensor Networks*, pages 32–44.

Barfoot, T. D. (2017). *State Estimation for Robotics*. Cambridge University Press.

Barvinok, A. (2002). *A Course in Convexity*, volume 54 of *Graduate Studies in Mathematics*. American Mathematical Society.

Beeson, P. and Ames, B. (2015). TRAC-IK: An Open-Source Library for Improved Solving of Generic Inverse Kinematics. In *15th Int. Conf. on Humanoid Robots (Humanoids)*, pages 928–935.

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific, second edition.

Biswas, P., Lian, T.-C., Wang, T.-C., and Ye, Y. (2006). Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks (TOSN)*, 2(2):188–220.

Blackmore, L. and Williams, B. (2006). Optimal Manipulator Path Planning with Obstacles using Disjunctive Programming. In *American Control Conference*. IEEE.

Blanchini, F., Fenu, G., Giordano, G., and Pellegrino, F. A. (2015). Inverse kinematics by means of convex programming: Some developments. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 515–520, Gothenburg, Sweden.

Blanchini, F., Fenu, G., Giordano, G., and Pellegrino, F. A. (2017). A convex programming approach to the inverse kinematics problem for manipulators under constraints. *European Journal of Control*, 33:11–23.

Boumal, N., Voroninski, V., and Bandeira, A. S. (2020). Deterministic Guarantees for Burer-Monteiro Factorizations of Smooth Semidefinite Programs. *Communications on Pure and Applied Mathematics*, 73(3):581–608.

Boyd, S. and Vandenberghe, L. (1997). Semidefinite Programming Relaxations of Non-Convex Problems in Control and Combinatorial Optimization. In *Communications, Computation, Control, and Signal Processing*, pages 279–287.

Boyd, S. P. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Briales, J. and Gonzalez-Jimenez, J. (2017). Cartan-Sync: Fast and Global SE(d)-Synchronization. *IEEE Robotics and Automation Letters*, 2(4):2127–2134.

Briales, J., Gonzalez-Jimenez, J., et al. (2017). Convex Global 3D Registration with Lagrangian Duality. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Briales, J., Kneip, L., and Gonzalez-Jimenez, J. (2018). A certifiably globally optimal solution to the non-minimal relative pose problem. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 145–154.

Brookshire, J. and Teller, S. (2012). Automatic Calibration of Multiple Coplanar Sensors. *Robotics: Science and Systems VII*, 33.

Brookshire, J. and Teller, S. (2013). Extrinsic Calibration from Per-Sensor Egomotion. *Robotics: Science and Systems VIII*, pages 504–512.

Brynte, L., Larsson, V., Iglesias, J. P., Olsson, C., and Kahl, F. (2022). On the Tightness of Semidefinite Relaxations for Rotation Estimation. *Journal of Mathematical Imaging and Vision*, 64(1):57–67.

Buss, S. R. (2004). Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares Methods. *IEEE Journal of Robotics and Automation*, 17(16):1–19.

Buss, S. R. and Kim, J.-S. (2005). Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools*, 10(3):37–49.

Canny, J. (1988). *The Complexity of Robot Motion Planning.* MIT Press.

Carlone, L., Rosen, D. M., Calafiore, G., Leonard, J. J., and Dellaert, F. (2015). Lagrangian Duality in 3D SLAM: Verification Techniques and Optimal Solutions. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 125–132. IEEE.

Chandrasekaran, V., Recht, B., Parrilo, P. A., and Willsky, A. S. (2012). The Convex Geometry of Linear Inverse Problems. *Found. Comput. Math*, pages 805–849.

Chen, H. (1991). A Screw Motion Approach to Uniqueness Analysis of Head-Eye Geometry. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 145–151.

Cifuentes, D., Agarwal, S., Parrilo, P. A., and Thomas, R. R. (2022). On the Local Stability of Semidefinite Relaxations. *Springer Mathematical Programming*, 193(2):629–663.

Cifuentes, D., Harris, C., and Sturmfels, B. (2018). The geometry of SDP-exactness in quadratic optimization. *Springer Mathematical Programming*, 182(1):399–428.

Dai, H., Izatt, G., and Tedrake, R. (2019). Global Inverse Kinematics via Mixed-Integer Convex Optimization. *The International Journal of Robotics Research*, 38(12-13):1420–1441.

Daniilidis, K. (1999). Hand-Eye Calibration Using Dual Quaternions. *The International Journal of Robotics Research*, 18(3):286–298.

Dattorro, J. (2005). *Convex Optimization & Euclidean Distance Geometry.* MeBoo Publishing, Palo Alto, California.

Deits, R. and Tedrake, R. (2015a). Computing Large Convex Regions of Obstacle-Free Space Through Semidefinite Programming. In *Algorithmic Foundations of Robotics XI*, volume 107, pages 109–124.

Deits, R. and Tedrake, R. (2015b). Efficient Mixed-Integer Planning for UAVs in Cluttered Environments. In *2015 IEEE international Conference on Robotics and Automation (ICRA)*, pages 42–49. IEEE.

Diamond, S. and Boyd, S. (2016). CVXPY: A Python-embedded Modeling Language for Convex Optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913.

Diankov, R. (2010). *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Ding, Y., Krislock, N., Qian, J., and Wolkowicz, H. (2010). Sensor Network Localization, Euclidean Distance Matrix Completions, and Graph Realization. *Optim. Eng.*, 11(1):45–66.

Dokmanic, I., Parhizkar, R., Ranieri, J., and Vetterli, M. (2015). Euclidean Distance Matrices: Essential Theory, Algorithms and Applications. *IEEE Signal Processing Magazine*, 32(6):12–30.

Eriksson, A., Olsson, C., Kahl, F., and Chin, T.-J. (2018). Rotation Averaging and Strong Duality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 127–135.

Erleben, K. and Andrews, S. (2019). Solving Inverse Kinematics Using Exact Hessian Matrices. *Computers & Graphics*, 78:1–11.

Fackler, P. L. (2005). Notes on Matrix Calculus. *North Carolina State University*.

Fan, T., Wang, H., Rubenstein, M., and Murphey, T. (2020). CPL-SLAM: Efficient and Certifiably Correct Planar Graph-based SLAM Using the Complex Number Representation. *IEEE Transactions on Robotics*, 36(6):1719–1737.

Fredriksson, J. and Olsson, C. (2012). Simultaneous Multiple Rotation Averaging Using Lagrangian Duality. In *Asian Conference on Computer Vision*, pages 245–258. Springer.

Furgale, P., Barfoot, T. D., and Sibley, G. (2012). Continuous-time Batch Estimation using Temporal Basis Functions. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2088–2095. IEEE.

Garcia-Salguero, M., Briales, J., and Gonzalez-Jimenez, J. (2021). Certifiable Relative Pose Estimation. *Image and Vision Computing*, 109:104142.

Garstka, M., Cannon, M., and Goulart, P. (2019). COSMO: A conic operator splitting method for large convex problems. In *2019 18th European Control Conference (ECC)*, pages 1951–1956, Naples, Italy.

Giamou, M., Ma, Z., Peretroukhin, V., and Kelly, J. (2019). Certifiably Globally Optimal Extrinsic Calibration From Per-Sensor Egomotion. *IEEE Robotics and Automation Letters*, 4(2):367–374.

Giamou, M., Marić, F., Rosen, D. M., Peretroukhin, V., Roy, N., Petrović, I., and Kelly, J. (2022). Convex Iteration for Distance-Geometric Inverse Kinematics. *IEEE Robotics and Automation Letters*, 7(2):1952–1957.

Goemans, M. X. and Williamson, D. P. (1995). Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *Journal of the ACM*, 42(6):1115–1145.

Gomez-Ojeda, R., Briales, J., Fernandez-Moral, E., and Gonzalez-Jimenez, J. (2015). Extrinsic Calibration of a 2D Laser-rangefinder and a Camera Based on Scene Corners. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3611–3616. IEEE.

Grant, M., Boyd, S., and Ye, Y. (2008). CVX: MATLAB software for disciplined convex programming.

Heller, J., Havlena, M., and Pajdla, T. (2012). A branch-and-bound Algorithm for Globally Optimal Hand-eye Calibration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1608–1615. IEEE.

Heller, J., Henrion, D., and Pajdla, T. (2014). Hand-eye and Robot-world Calibration by Global Polynomial Optimization. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3157–3164. IEEE.

Heller, J. and Pajdla, T. (2014). World-base Calibration by Global Polynomial Optimization. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 593–600. IEEE.

Holmes, C. and Barfoot, T. D. (2022). An Efficient Global Optimality Certificate for Landmark-Based SLAM. *arXiv preprint arXiv:2206.12961*.

Horaud, R. and Dornaika, F. (1995). Hand-eye Calibration. *The international journal of robotics research*, 14(3):195–210.

Hu, X., Olesen, D., and Per, K. (2019). A Novel Robust Approach for Correspondence-Free Extrinsic Calibration. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–6. IEEE.

Husty, M. L., Pfurner, M., and Schröcker, H.-P. (2007). A New and Efficient Algorithm for the Inverse Kinematics of a General Serial 6R Manipulator. *Mechanism and Machine Theory*.

Iglesias, J. P., Olsson, C., and Kahl, F. (2020). Global Optimality for Point Set Registration Using Semidefinite Programming. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8287–8295.

Jasour, A., Aybat, N. S., and Lagoa, C. M. (2015). Semidefinite Programming for Chance Constrained Optimization over Semialgebraic Sets. *SIAM Journal on Optimization*, 25(3):1411–1440.

Jasour, A. and Lagoa, C. (2016). Convex Chance Constrained Model Predictive Control. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6204–6209. IEEE.

Jasour, A. M., Hofmann, A., and Williams, B. C. (2018). Moment-Sum-Of-Squares Approach For Fast Risk Estimation In Uncertain Environments. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2445–2451. IEEE.

Kelly, J. and Sukhatme, G. S. (2011). Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-Calibration. *The International Journal of Robotics Research*, 30(1):56–79.

Kelly, J. and Sukhatme, G. S. (2014). A General Framework for Temporal Calibration of Multiple Proprioceptive and Exteroceptive Sensors. In *Experimental Robotics: The 12th International Symposium on Experimental Robotics*, pages 195–209. Springer Berlin Heidelberg.

Krislock, N. and Wolkowicz, H. (2010). Explicit Sensor Network Localization using Semidefinite Representations and Facial Reductions. *SIAM J. on Optimization*, 20(5):2679–2708.

Lambert, J., Clement, L., Giamou, M., and Kelly, J. (2017). Entropy-Based $Sim(3)$ Calibration of 2D Lidars to Egomotion Sensors. *arXiv preprint arXiv:1707.08680*.

Landry, B., Deits, R., Florence, P. R., and Tedrake, R. (2016). Aggressive quadrotor flight through cluttered environments using mixed integer programming. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1469–1475. IEEE.

Lasserre, J. B. (2001). Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817.

Lasserre, J.-B. (2010). *Moments, Positive Polynomials and their Applications*, volume 1. World Scientific.

Lasserre, J. B., Toh, K.-C., and Yang, S. (2017). A Bounded Degree SOS Hierarchy for Polynomial Optimization. *EURO Journal on Computational Optimization*.

Le Naour, T., Courty, N., and Gibet, S. (2019). Kinematics in the Metric Space. *Computers & Graphics*, 84:13–23.

Lee, H.-Y. and Liang, C.-G. (1988). A new vector theory for the analysis of spatial mechanisms. *Mechanism and Machine Theory*, 23(3):209–217.

Levinson, J. and Thrun, S. (2014). Unsupervised Calibration for Multi-beam Lasers. In *Experimental Robotics*, pages 179–193. Springer.

Lewis, A. D. (2009). Semicontinuity of Rank and Nullity and Some Consequences. *Unpublished Online Notes*.

Liberti, L., Lavor, C., Maculan, N., and Mucherino, A. (2014). Euclidean Distance Geometry and Applications. *SIAM Rev.*, 56(1):3–69.

Limoyo, O., Ablett, T., Marić, F., Volpatti, L., and Kelly, J. (2018). Self-Calibration of Mobile Manipulator Kinematic and Sensor Extrinsic Parameters Through Contact-Based Interaction. *arXiv preprint arXiv:1803.06406*.

Ma, Q., Goh, Z., and S. Chirikjian, G. (2016). Probabilistic Approaches to the AXB = YCZ Calibration Problem in Multi-Robot Systems. In *Robotics: Science and Systems XII*. Robotics: Science and Systems Foundation.

Ma, W.-K. K. (2010). Semidefinite relaxation of quadratic optimization problems and applications. *IEEE Signal Processing Magazine*, 1053(5888/10):20–34.

Majumdar, A., Hall, G., and Ahmadi, A. A. (2019). A Survey of Recent Scalability Improvements for Semidefinite Programming with Applications in Machine Learning, Control, and Robotics. *arXiv:1908.05209 [cs, eess, math]*.

Manocha, D. and Canny, J. (1994). Efficient Inverse Kinematics for General 6R Manipulators. *IEEE Trans. on Robotics and Automation*, 10(5):648–657.

Marić, F., Giamou, M., Hall, A. W., Khoubyarian, S., Petrović, I., and Kelly, J. (2021). Riemannian Optimization for Distance-Geometric Inverse Kinematics. *IEEE Transactions on Robotics*, pages 1–20.

Marić, F., Giamou, M., Khoubyarian, S., Petrović, I., and Kelly, J. (2020). Inverse Kinematics for Serial Kinematic Chains via Sum of Squares Optimization. In *International Conference on Robotics and Automation (ICRA)*, pages 7101–7107.

Milgram, R. J., Trinkle, J. C., et al. (2004). The geometry of configuration spaces for closed chains in two and three dimensions. *Homology, Homotopy and Applications*, 6(1):237–267.

Nakamura, Y., Hanafusa, H., and Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3–15.

Nedich, A. (2008). Lecture 8: Strong Duality Results. In *Convex Optimization (IE 598 AN)*.

Nie, J. (2009). Sum of Squares Method for Sensor Network Localization. *Computational Optimization and Applications*, 43(2):151–179.

O'Donoghue, B., Chu, E., Parikh, N., and Boyd, S. (2016). Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068.

Olsson, C. (2009). *Global Optimization in Computer Vision: Convexity, Cuts and Approximation Algorithms*. Number 2009:2. Faculty of Engineering, Centre for Mathematical Sciences, Mathematics.

Olsson, C. and Eriksson, A. (2008). Solving Quadratically Constrained Geometrical Problems Using Lagrangian Duality. In *Pattern Recognition, 2008. 19th International Conference on*, pages 1–5. IEEE.

Paden, B., Varricchio, V., and Frazzoli, E. (2017). Verification and Synthesis of Admissible Heuristics for Kinodynamic Motion Planning. *IEEE Robotics and Automation Letters*, 2(2):648–655.

Pandey, G., McBride, J. R., Savarese, S., and Eustice, R. M. (2015). Automatic Extrinsic Calibration of Vision and Lidar by Maximizing Mutual Information. *Journal of Field Robotics*, 32(5):696–722.

Park, J. and Boyd, S. (2017). General Heuristics for Nonconvex Quadratically Constrained Quadratic Programming. *arXiv preprint arXiv:1703.07870*.

Parrilo, P. A. (2003). Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320.

Peretroukhin, V., Giamou, M., Rosen, D. M., Greene, W. N., Roy, N., and Kelly, J. (2020). A Smooth Representation of Belief Over SO(3) for Deep Rotation Learning with Uncertainty. *Robotics: Science and Systems XVI*.

Porta, J., Ros, L., Thomas, F., and Torras, C. (2005a). A Branch-and-Prune Solver for Distance Constraints. 21(2):176–187.

Porta, J. M., Ros, L., and Thomas, F. (2005b). Inverse Kinematics by Distance Matrix Completion. In *Proc. of 12th Int. Workshop on Computational Kinematics*. Elsevier.

Qiao, S., Liao, Q., Wei, S., and Su, H.-J. (2010). Inverse kinematic Analysis of the General 6R Serial Manipulators based on Double Quaternions. *Mechanism and Machine Theory*.

Recht, B., Fazel, M., and Parrilo, P. A. (2010). Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization. *SIAM Review*, 52(3):471–501.

Rockafellar, R. T. (1993). Lagrange Multipliers and Optimality. *SIAM Review*, 35(2):183–238.

Rosen, D. M., Carlone, L., Bandeira, A. S., and Leonard, J. J. (2019). SE-Sync: A Certifiably Correct Algorithm for Synchronization Over the Special Euclidean Group. *The International Journal of Robotics Research*, 38(2-3):95–125.

Rosen, D. M., Doherty, K. J., Terán Espinoza, A., and Leonard, J. J. (2021). Advances in Inference and Representation for Simultaneous Localization and Mapping. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):215–242.

Ruland, T., Pajdla, T., and Krüger, L. (2012). Globally optimal hand-eye calibration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1035–1042. IEEE.

Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., Pan, J., Patil, S., Goldberg, K., and Abbeel, P. (2014). Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*.

Sciavicco, L. and Siciliano, B. (1986). Coordinate transformation: A solution algorithm for one class of robots. *IEEE transactions on Systems, Man, and Cybernetics*.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). Differential kinematics and statics. *Robotics: Modelling, Planning and Control*, pages 105–160.

Singer, A. and Cucuringu, M. (2010). Uniqueness of low-rank matrix completion by rigidity theory. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1621–1641.

So, A. M.-C. and Ye, Y. (2007). Theory of Semidefinite Programming for Sensor Network Localization. *Mathematical Programming*, 109(2-3):367–384.

Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2005). *Robot Modeling and Control*. John Wiley & Sons.

Strang, G. (1993). The Fundamental Theorem of Linear Algebra. *The American Mathematical Monthly*, 100(9):848–855.

Straub, J., Campbell, T., How, J. P., and Fisher III, J. W. (2017). Efficient Global Point Cloud Alignment Using Bayesian Nonparametric Mixtures. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2.

Sun, Z., Helmke, U., and Anderson, B. D. O. (2015). Rigid formation shape control in general dimensions: An invariance principle and open problems. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 6095–6100. IEEE.

Tian, Y., Khosoussi, K., Rosen, D. M., and How, J. P. (2021). Distributed Certifiably Correct Pose-Graph Optimization. *IEEE Transactions on Robotics*, 37(6):2137–2156.

Toh, K.-C., Todd, M. J., and Tütüncü, R. H. (1999). Sdpt3–a matlab software package for semidefinite programming, version 1.3. *Optimization methods and software*, 11(1-4):545–581.

Tony Cai, T. (2005). One-Sided Confidence Intervals in Discrete Distributions. *J. of Statistical Planning and Inference*, 131(1):63–88.

Trutman, P., Din, M. S. E., Henrion, D., and Pajdla, T. (2022). Globally Optimal Solution to Inverse Kinematics of 7DOF Serial Manipulator. *IEEE Robotics and Automation Letters*, 7(3):6012–6019.

Trutman, P., Mohab, S. E. D., Henrion, D., and Pajdla, T. (2020). Globally Optimal Solution to Inverse Kinematics of 7DOF Serial Manipulator. *arXiv:2007.12550 [cs, math]*.

Tsai, R. and Lenz, R. (1989). A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358.

Varava, A., Carvalho, J. F., Pokorny, F. T., and Kragic, D. (2020). Caging and Path Non-Existence: A Deterministic Sampling-Based Verification Algorithm. In *Robotics Research*, volume 10, pages 589–604.

Virtanen, P. et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3):261–272.

Wächter, A. and Biegler, L. T. (2006). On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical programming*, 106(1):25–57.

Walters, C., Mendez, O., Hadfield, S., and Bowden, R. (2019). A Robust Extrinsic Calibration Framework for Vehicles with Unscaled Sensors. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 36–42. IEEE.

Wang, W. and Yu, N. (2018). Chordal Conversion Based Convex Iteration Algorithm for Three-Phase Optimal Power Flow Problems. *IEEE Trans. on Power Systems*, 33(2):1603–1613.

Wei, L., Naiguang, L., Mingli, D., and Xiaoping, L. (2018). Calibration-free robot-sensor calibration approach based on second-order cone programming. In *MATEC Web of Conferences*, volume 173, page 02005. EDP Sciences.

Weisser, T., Lasserre, J. B., and Toh, K.-C. (2018). Sparse-BSOS: a bounded degree SOS hierarchy for large scale polynomial optimization with sparsity. *Mathematical Programming Computation*, 10(1):1–32.

Wise, E., Giamou, M., Khoubyarian, S., Grover, A., and Kelly, J. (2020). Certifiably Optimal Monocular Hand-Eye Calibration. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 271–278. IEEE.

Wodtko, T., Horn, M., Buchholz, M., and Dietmayer, K. (2021). Globally Optimal Multi-Scale Monocular Hand-Eye Calibration Using Dual Quaternions. In *2021 International Conference on 3D Vision (3DV)*, pages 249–257. IEEE.

Wolkowicz, H. (2002). A note on lack of strong duality for quadratic problems with orthogonal constraints. *European Journal of Operational Research*, 143(2):356–364.

Wolpert, D. H. and Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Wu, J., Liu, M., Zhu, Y., Zou, Z., Dai, M.-Z., Zhang, C., Jiang, Y., and Li, C. (2020). Globally Optimal Symbolic Hand-Eye Calibration. *IEEE/ASME Transactions on Mechatronics*, pages 1369–1379.

Wu, J., Zheng, Y., Gao, Z., Jiang, Y., Hu, X., Zhu, Y., Jiao, J., and Liu, M. (2022). Quadratic Pose Estimation Problems: Globally Optimal Solutions, Solvability/Observability Analysis, and Uncertainty Description. *IEEE Transactions on Robotics*, pages 3314–3335.

Xu, X., Ananthanarayanan, H., and Ordonez, R. (2014). Design and Construction of 9-DOF Hyper-Redundant Robotic Arm. In *IEEE National Aerospace and Electronics Conference*, pages 321–326.

Yang, H., Dong, W., Carlone, L., and Koltun, V. (2021). Self-Supervised Geometric Perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14350–14361.

Yang, H., Shi, J., and Carlone, L. (2020). Teaser: Fast and Certifiable Point Cloud Registration. *IEEE Transactions on Robotics*, 37(2):314–333.

Yenamandra, T., Bernard, F., Wang, J., Mueller, F., and Theobalt, C. (2019). Convex Optimisation for Inverse Kinematics. *2019 Intl. Conf. on 3D Vision (3DV)*, pages 318–327.

Zhang, Q. and Pless, R. (2004). Extrinsic Calibration of a Camera and Laser Range Finder (Improves Camera Calibration). In *IROS*, volume 3, pages 2301–2306.

Zhao, J. (2020). An Efficient Solution to Non-Minimal Case Essential Matrix Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1777–1792.

Zheng, X. J., Sun, X. L., Li, D., and Xu, Y. F. (2012). On Zero Duality Gap in Nonconvex Quadratic Programming Problems. *Journal of Global Optimization*, 52(2):229–242.