6

Certifiably Optimal Solvers and Theoretical Properties of SLAM

David M. Rosen, Kasra Khosoussi, Connor Holmes, Gamini Dissanayake,
Timothy Barfoot, and Luca Carlone

Chapters 1–3 have discussed how to formalize the estimation problems arising in SLAM as optimization problems, using maximum a posteriori estimation or (more generally) *M-estimation.* Moreover, these chapters have introduced iterative local solvers (*e.g.,* Gauss-Newton, Levenberg-Marquardt, gradient descent) that look for a solution of the resulting optimization by iteratively refining a given initial guess.

In this chapter, we take a closer look at the optimization problems arising in SLAM and address two fundamental questions. First, can we design efficient algorithms that are guaranteed to compute a *globally* optimal solution of these problems, possibly without an initial guess? And second, *how accurate* is the optimal solution as an estimate of the ground truth, and what factors affect its accuracy?

The first question is about *computation* and *reliability*: the question starts from the observation that SLAM requires solving nonconvex optimization problems, which have multiple local minima. Depending upon the quality of the initial guess, iterative algorithms can get stuck in local minima, and hence produce incorrect estimates. Moreover, iterative methods do not provide tools to detect convergence to suboptimal solutions, which leads to trustworthiness and reliability concerns in practical applications. In Section 6.1, we show that despite the nonconvexity of typical SLAM problems, we can design *certifiably optimal algorithms* that solve the SLAM optimization to provable optimality, and can discern sub-optimal solutions from optimal ones. These algorithms are based on a mathematical tool, known as *semidefinite relaxation*, which we review in Section 6.1.1. The chapter presents certifiable algorithms for pose graph optimization (SE-Sync, Section 6.1.2), landmark-based SLAM with range and bearing measurements (Section 6.1.3), and then discusses extensions to other SLAM problems, including problems with range-only measurements, anisotropic noise, and outliers (Section 6.1.4).

The second question is concerned with *fundamental limits and estimation errors*: when designing a SLAM system, one often wants to ensure that the robot pose estimate and the map estimate are close to the ground truth, since large errors may induce failure in downstream tasks, including motion planning. Understanding fundamental limits is not only useful for analytical purposes but has very practical implications: as we discuss below, these limits are influenced by the structure of the factor graph, and we can actively control this structure by carefully driving the robot (*e.g.,* forcing it to revisit places or landmarks). Therefore, in Section 6.2 we discuss information-theoretic limits on the achievable accuracy of the SLAM estimate. In particular, Section 6.2.1 introduces the Cramér-Rao Lower Bound to quantify the estimation error in SLAM, and Section 6.2.2 draws connections between this bound and the graphical structure of the SLAM problem.

As we will see, investigating these questions reveals deep connections between the algebraic, geometric, and graph-theoretic structures encoded in the SLAM problem, and the computational and statistical hardness of solving it. As usual, we conclude the chapter with an outlook to recent trends and references in Section 6.3.

6.1 Certifiably Optimal Solvers for SLAM

SLAM is conventionally formulated as a *high-dimensional* and *nonconvex* problem. Finding the global solution to a general nonconvex optimization problem is inherently challenging, primarily due to the existence of numerous local minima. In particular, many special cases of SLAM are known to be *NP-hard*, including, *e.g.,* angular synchronization, rotation averaging, and pose graph optimization. It follows that there is no algorithm that is capable of efficiently solving these problems in general, unless P = NP [944].

Despite this theoretical complexity, early SLAM research consistently demonstrated surprising convergence to solutions close to the ground truth. SLAM algorithms are typically initialized with robot poses computed using odometry. In a well-calibrated mobile robot, odometry drift can be reduced to less than one percent of the distance traveled. Thus, the unexpectedly reliable convergence of SLAM was initially attributed to the availability of high-quality initial estimates.

However, subsequent research has shown that even in the presence of poor or inconsistent initializations, a range of optimization techniques, including Stochastic Gradient Descent, Levenberg–Marquardt, and preconditioned conjugate gradient methods, can often recover near-optimal solutions. These findings suggest that, despite being a nonlinear and nonconvex problem, SLAM possesses an intrinsic structure that make it particularly amenable to solution through specialized optimization strategies. At the same time, these results show that simply applying off-the-shelf *local* optimization methods (such as gradient descent or quasi-Newton methods) can produce egregiously wrong estimates, even if the underlying instance of the SLAM problem is well-posed.

This insight has motivated the development of algorithmic frameworks that explicitly leverage the graph-theoretic and geometric underpinnings of the SLAM problem to obtain improved convergence properties and guarantees. One of the most exciting recent advances in SLAM has been the development of practical optimization algorithms that — despite this problem's general intractability — are nevertheless provably capable of recovering *certifiably globally optimal* solutions of common SLAM formulations (*e.g.,* pose graph optimization) under mild conditions. These techniques, called *certifiably correct* optimization methods, are based on *convex relaxation* (rather than *local* optimization). Moreover, they are *certifiably optimal* in the sense that upon computing a solution to the optimization problem, they will be able to quantify how suboptimal that estimate is, and possibly certify its optimality. Such a statement does not contradicts the NP-hardness of the problem: in worst-case scenarios, these algorithms can still fail to produce a certificate of optimality for a solution they compute and only provide a suboptimality bound. However, these algorithms remain of practical interest for two main reasons: (i) in practice, they *do* produce an optimality certificate for virtually all SLAM problems of practical interest (*e.g.,* for reasonable amounts of measurement noise), and (ii) the failure to produce an optimality certificate is in itself informative, since it might trigger a warning to downstream tasks not to trust the SLAM estimate, or for the robot to take fail-safe measures.

In this subsection we provide a brief introduction to certifiably correct estimation methods for SLAM. We begin by considering *Shor's relaxation*, one of the fundamental tools that is used to construct the convex relaxations underpinning certifiable estimation techniques. Next, we show how to apply Shor's relaxation to produce a certifiably correct estimation method (*SE-Sync*) to solve the fundamental problem of pose graph optimization. Finally, we discuss discuss extensions of SE-Sync and Shor's relaxation to a broader range of SLAM problems.

6.1.1 Shor's Relaxation

In this subsection we introduce *Shor's relaxation*, one of the fundamental tools that we will use to construct the convex relaxations underpinning certifiable estimators.

In brief, Shor's relaxation is a procedure for constructing convex relaxations of *quadratically constrained quadratic programs* (QCQPs); that is, optimization problems in which the objective and constraint functions are *quadratics.* As we will see in the next section, many common SLAM formulations can be cast as QCQPs.

We introduce Shor's relaxation by describing its application to a generic QCQP, and then tailor it to SLAM in the next section. Consider the following QCQP:

$$\begin{aligned} p^* = \min_{\boldsymbol{x} \in \mathbb{R}^n} & \; \boldsymbol{x}^\mathsf{T} \boldsymbol{C} \boldsymbol{x} \\ \text{s.t. } & \; \boldsymbol{x}^\mathsf{T} \boldsymbol{A}_i \boldsymbol{x} = b_i \quad i = 1, \dots, m. \end{aligned} \tag{QCQP}$$

where $\boldsymbol{C}, \boldsymbol{A}_1, \dots, \boldsymbol{A}_m \in \mathbb{S}^n$ are symmetric matrices and $\boldsymbol{b} = (b_1, \dots, b_m) \in \mathbb{R}^m$ is a vector. We will show how to produce a convex relaxation of (QCQP) by applying a sequence of simple algebraic manipulations to (QCQP).

To begin, if $\boldsymbol{M} \in \mathbb{S}^n$ is any symmetric matrix, we may exploit the cyclic property of the trace to rewrite the quadratic form $\boldsymbol{x}^\mathsf{T} \boldsymbol{M} \boldsymbol{x}$ determined by $\boldsymbol{M}$ according to

$$\boldsymbol{x}^\mathsf{T} \boldsymbol{M} \boldsymbol{x} = \operatorname{tr}\left(\boldsymbol{x}^\mathsf{T} \boldsymbol{M} \boldsymbol{x}\right) = \operatorname{tr}\left(\boldsymbol{M} \boldsymbol{x} \boldsymbol{x}^\mathsf{T}\right). \tag{6.1}$$

Applying (6.1) to (QCQP), we thus obtain the equivalent form:

$$\begin{aligned} p^* = \min_{\boldsymbol{x} \in \mathbb{R}^n} & \operatorname{tr}\left(\boldsymbol{C}\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\right) \\ \text{s.t. } & \operatorname{tr}\left(\boldsymbol{A}_i\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\right) = b_i, \quad i = 1, \dots, m. \end{aligned} \tag{6.2}$$

Now observe that the decision variable $\boldsymbol{x}$ only enters problem (6.2) through outer products of the form $\boldsymbol{X} \triangleq \boldsymbol{x}\boldsymbol{x}^\mathsf{T}$; note that every such matrix $\boldsymbol{X}$ is *symmetric*, *rank-1*, and *positive-semidefinite* (by construction). Conversely, if $\boldsymbol{X} \in \mathbb{S}^n_+$ is a positive-semidefinite matrix and $\operatorname{rank}(\boldsymbol{X}) = 1$, it is easily shown (by considering a symmetric eigendecomposition) that $\boldsymbol{X}$ admits a symmetric factorization of the form $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\mathsf{T}$ for some $\boldsymbol{x} \in \mathbb{R}^n$. Putting these observations together, we thus have the equivalence

$$\boldsymbol{X} \in \mathbb{S}^n_+ \text{ and } \operatorname{rank}(\boldsymbol{X}) = 1 \qquad \Longleftrightarrow \qquad \exists \boldsymbol{x} \in \mathbb{R}^n \text{ such that } \boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\mathsf{T}. \tag{6.3}$$

In light of (6.3), problem (6.2) is equivalent to

$$\begin{aligned} p^* = \min_{\boldsymbol{X} \in \mathbb{S}^n} & \operatorname{tr}(\boldsymbol{C}\boldsymbol{X}) \\ \text{s.t. } & \operatorname{tr}(\boldsymbol{A}_i\boldsymbol{X}) = b_i, \quad i = 1, \dots, m, \\ & \boldsymbol{X} \succeq 0, \\ & \operatorname{rank}(\boldsymbol{X}) = 1. \end{aligned} \tag{6.4}$$

Thus far problems (6.2) and (6.4) are completely equivalent; however, formulation (6.4) has the advantage that it reveals a great deal of useful structure. Indeed, the objective and constraint functions in (6.4) are *linear* functions of the (matrix) decision variable $\boldsymbol{X}$, and the positive-semidefiniteness constraint $\boldsymbol{X} \succeq 0$ is *convex*. Thus, the only difficulty in solving (6.4) is due to the (nonconvex) rank constraint.

Shor's relaxation [1017] simply consists of discarding the rank constraint appearing in (6.4), thereby producing the following *convex relaxation* of (QCQP):

$$\begin{aligned} d^* = \min_{\boldsymbol{X} \in \mathbb{S}^n} & \operatorname{tr}(\boldsymbol{C}\boldsymbol{X}) \\ \text{s.t. } & \operatorname{tr}(\boldsymbol{A}_i\boldsymbol{X}) = b_i, \quad i = 1, \dots, m, \\ & \boldsymbol{X} \succeq 0. \end{aligned} \tag{SDP}$$

Note that (SDP) entails minimizing a linear function over the set of positive-semidefinite matrices, subject to a set of linear equality constraints; problems of this form are called *semidefinite programs*. Semidefinite programs, or SDPs, are convex optimization problems, and hence can be solved in polynomial time.

Now let us consider the relation between problem (QCQP) and its convex relaxation (SDP). First, note that we obtained (QCQP) from (SDP) by *expanding the former's feasible set* (*i.e.,* by dropping the rank constraint in (6.4)); indeed, it is easy to see that every feasible point $\boldsymbol{x} \in \mathbb{R}^n$ in (QCQP) *lifts* to a corresponding feasible point $\boldsymbol{X} \triangleq \boldsymbol{x}\boldsymbol{x}^\mathsf{T}$ for (SDP). It follows that the optimal values of (QCQP) and (SDP) satisfy the relation:

$$d^* \leq p^*, \tag{6.5}$$

since the latter problem minimizes the same objective over a larger feasible set.

Inequality (6.5) already provides a very useful method for *assessing the quality* of candidate solutions of (QCQP). Suppose that we have a feasible point $\hat{\boldsymbol{x}} \in \mathbb{R}^n$ of (QCQP); for example, $\hat{\boldsymbol{x}}$ might have been obtained by performing *local* optimization. Writing $f(\boldsymbol{x}) \triangleq \boldsymbol{x}^\mathsf{T}\boldsymbol{C}\boldsymbol{x}$ for the objective, inequality (6.5) implies that we may bound the *suboptimality* $f(\hat{\boldsymbol{x}}) - p^*$ of $\hat{\boldsymbol{x}}$ as a solution of (QCQP) according to

$$f(\hat{\boldsymbol{x}}) - p^* \leq f(\hat{\boldsymbol{x}}) - d^*. \tag{6.6}$$

Note that while the optimal value of p^* of (QCQP) is very hard to compute in general, the optimal value d^* *can* be computed efficiently by solving the relaxation (SDP). Inequality (6.6) thus gives us a practical way of *bounding* $\hat{\boldsymbol{x}}$'s suboptimality *without* the need to know p^* itself. In particular, if the right-hand side of (6.6) is small, we may conclude that $\boldsymbol{x}$ is a *near*-optimal solution of (QCQP).

Moreover, if we solve the relaxation (SDP) and it so happens that the resulting minimizer $\boldsymbol{X}^* = \boldsymbol{x}^*\boldsymbol{x}^{*\mathsf{T}}$ has rank 1, then it immediately follows that the vector $\boldsymbol{x}^* \in \mathbb{R}^n$ is a *global minimizer* for the original (*nonconvex*) problem (QCQP) (since $\boldsymbol{x}^*$ is feasible in (QCQP) and satisfies $f(\boldsymbol{x}^*) = d^*$ in (6.6)). As we will see, it turns out that this favorable situation actually occurs quite often for many robotic state estimation tasks, enabling us to recover *exact, globally optimal* solutions for the *nonconvex* problem (QCQP) from solutions of its relaxation (SDP).

6.1.2 SE-Sync: Certifiably Correct Pose Graph Optimization

In this subsection we show how one can apply Shor's relaxation to develop a certifiably correct algorithm (*SE-Sync*) for solving the fundamental problem of pose graph optimization (PGO). PGO is one of the simplest and most commonly used SLAM formulations, and thus provides a natural concrete example to illustrate the construction of certifiable estimation algorithms. Moreover, pose graph optimization was the first SLAM formulation that was shown to be amenable for convex relaxations [155, 157, 945], and the corresponding ideas have been shown to generalize to a wide range of SLAM problems, as we will discuss below.

Our development proceeds in three stages. First, we show how to formalize pose graph optimization as an instance of maximum likelihood estimation, and how to reduce it to a QCQP. Next, we derive Shor's SDP relaxation for PGO, and (crucially) show that this relaxation is in fact *exact* for sufficiently small measurement noise; this implies that we can recover *globally optimal* solutions to pose graph optimization by solving its (convex) SDP relaxation. Finally, we describe a specialized, structure-exploiting optimization algorithm that enables us to solve large-scale instances of this SDP relaxation in practice.

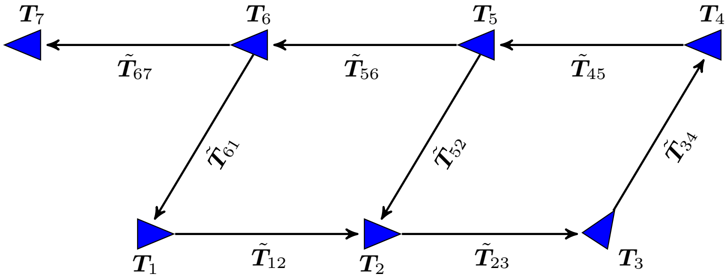

Figure 6.1 An example pose graph. Here the vertices are in one-to-one correspondence with the unknown poses $\boldsymbol{T}_i = (\boldsymbol{t}_i, \boldsymbol{R}_i) \in \mathrm{SE}(d)$ to be estimated, and the directed edges are in one-to-one correspondence with the set of noisy measurements $\tilde{\boldsymbol{T}}_{ij} \approx \boldsymbol{T}_i^{-1}\boldsymbol{T}_j$ of the relative poses between them.

6.1.2.1 Pose Graph Optimization: QCQP Formulation

Pose graph optimization estimates the values of a set of n unknown *poses* $\boldsymbol{T}_1, \ldots, \boldsymbol{T}_n \in \mathrm{SE}(d)$ in d-dimensional space (typically in SLAM $d = 2$ or 3), given noisy measurements $\tilde{\boldsymbol{T}}_{ij} \approx \boldsymbol{T}_i^{-1}\boldsymbol{T}_j$ of a set of relative pose measurements between them. In practice, the unknown poses $\boldsymbol{T}_1, \ldots, \boldsymbol{T}_n$ describe the trajectory of the robot (*i.e.,* they are sampled at discrete times along the robot trajectory), while the measurements $\tilde{\boldsymbol{T}}_{ij}$ are obtained by the SLAM front-end, *e.g.,* through LiDAR scan matching, wheel odometry, or 3D computer vision techniques. In this subsection, we show how to formalize this estimation problem via maximum likelihood estimation. As we will see, under suitable assumptions on the noise, the resulting optimization problem is a QCQP.

To begin, it is often convenient to model the data defining this estimation problem using a *pose graph* $\overrightarrow{\mathcal{G}}$,[1] constructed as follows. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple undirected graph whose nodes $i \in \mathcal{V}$ are in one-to-one correspondence with the unknown poses $\boldsymbol{T}_i$ and whose edges $\{i, j\} \in \mathcal{E}$ are in one-to-one correspondence with the set of available measurements.[2] We will assume (without loss of generality) that $\mathcal{G}$ is connected.[3] The pose graph $\overrightarrow{\mathcal{G}} = (\mathcal{V}, \overrightarrow{\mathcal{E}})$ is then obtained from $\mathcal{G}$ by assigning an orientation for each of the latter's edges (cf. Figure 6.1). By convention, the measurement $\tilde{\boldsymbol{T}}_{ij}$, which describes (a noisy version of) the pose $\boldsymbol{T}_j$ in the coordinate frame of the pose $\boldsymbol{T}_i$, is associated with a directed edge from i to j.

In order to formalize pose graph optimization as a maximum likelihood estimation, we must posit a noise model for the available measurements $\{\tilde{\boldsymbol{T}}_{ij}\}$. To do so,

[1] A pose graph is a special instance of a factor graph, where the variable nodes are poses and the factor nodes relate pairs of poses.

[2] The assumption that $\mathcal{G}$ is simple is not actually required for our methods, but will help to ease notational burden by avoiding the need to distinguish between multiple parallel edges.

[3] If $\mathcal{G}$ is not connected, then the pose graph optimization problem decomposes into a set of *independent* estimation problems over the connected components of $\mathcal{G}$; thus, the general case is always reducible to the case of a connected graph.

we will make use of the *isotropic Langevin distribution* $\mathcal{L}(\boldsymbol{M}, \kappa)$: this is an exponential family distribution over $\mathrm{SO}(d)$ whose probability density function is given by

$$p(\boldsymbol{R}; \boldsymbol{M}, \kappa) = \frac{1}{c_d(\kappa)} \exp\left(\kappa \mathrm{tr}\left(\boldsymbol{M}^\mathsf{T} \boldsymbol{R}\right)\right), \tag{6.7}$$

where $\boldsymbol{M} \in \mathrm{SO}(d)$ and $\kappa \geq 0$ are parameters, and $c_d(\kappa)$ is a normalization constant. Note that $\boldsymbol{M}$ plays the role of a *location* parameter (called the *mode*), while $\kappa \geq 0$ is a scalar *concentration* parameter. The isotropic Langevin distribution admits a particularly simple generative description in dimensions 2 and 3: to produce a sample $\tilde{\boldsymbol{R}} \sim \mathcal{L}(\boldsymbol{M}, \kappa)$, we first sample a rotation *angle* $\theta \sim \mathrm{vonMises}(0, 2\kappa)$ from the von Mises distribution on the circle, and then set $\tilde{\boldsymbol{R}} = \boldsymbol{M} \cdot \boldsymbol{R}(\theta)$ if $d = 2$, or $\tilde{\boldsymbol{R}} = \boldsymbol{M} \exp(\theta \boldsymbol{v}^\wedge)$ if $d = 3$, where $\boldsymbol{v} \sim \mathcal{U}(S^2)$ is a uniformly sampled rotation *axis* [943]. Intuitively, one can think of this distribution as an analogue of the Gaussian distribution over the (non-Euclidean) manifold of rotations.

Given a pose-graph $\overrightarrow{\mathcal{G}} = (\mathcal{V}, \overrightarrow{\mathcal{E}})$, we will assume that each measurement $\tilde{\boldsymbol{T}}_{ij} = (\tilde{\boldsymbol{t}}_{ij}, \tilde{\boldsymbol{R}}_{ij}) \in \mathrm{SE}(d)$ is obtained by sampling from the following probabilistic generative model:

$$\begin{aligned} \tilde{\boldsymbol{t}}_{ij} &= \bar{\boldsymbol{t}}_{ij} + \boldsymbol{t}^\epsilon_{ij}, & \boldsymbol{t}^\epsilon_{ij} &\sim \mathcal{N}\left(0, \tau_{ij}^{-1} I_d\right), \\ \tilde{\boldsymbol{R}}_{ij} &= \bar{\boldsymbol{R}}_{ij} \boldsymbol{R}^\epsilon_{ij}, & \boldsymbol{R}^\epsilon_{ij} &\sim \mathcal{L}\left(\mathbf{I}_d, \kappa_{ij}\right), \end{aligned} \qquad \forall (i,j) \in \overrightarrow{\mathcal{E}}, \tag{6.8}$$

where $\bar{\boldsymbol{T}}_{ij} = (\bar{\boldsymbol{t}}_{ij}, \bar{\boldsymbol{R}}_{ij}) \in \mathrm{SE}(d)$ is the true (latent) value of the relative pose between $\boldsymbol{T}_i$ and $\boldsymbol{T}_j$. Model (6.8) assumes the translation component $\tilde{\boldsymbol{t}}_{ij}$ of the ijth measurement is corrupted by *additive mean-zero isotropic Gaussian noise* with concentration parameter $\tau_{ij} > 0$, and the rotational component $\tilde{\boldsymbol{R}}_{ij}$ is corrupted by *multiplicative isotropic Langevin noise* with mode $\mathbf{I}_d$ and concentration parameter $\kappa_{ij} \geq 0$.

The primary motivation behind our use of the noise model (6.8) (as opposed to the more 'generic' exponentiated-Gaussian noise model over general Lie groups mentioned in Chapter 2) is that its associated maximum likelihood estimation takes a particularly simple algebraic form. Indeed, given a set of noisy measurements $\tilde{\boldsymbol{T}}_{ij}$ sampled from (6.8), a straightforward calculation shows that the associated maximum likelihood estimation is

Problem 1 (Pose Graph Optimization)

$$p^*_{\mathrm{MLE}} = \min_{\substack{\boldsymbol{t}_i \in \mathbb{R}^d \\ \boldsymbol{R}_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \overrightarrow{\mathcal{E}}} \kappa_{ij} \|\boldsymbol{R}_j - \boldsymbol{R}_i \tilde{\boldsymbol{R}}_{ij}\|_F^2 + \tau_{ij} \left\|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \tilde{\boldsymbol{t}}_{ij}\right\|_2^2. \tag{6.9}$$

In particular, note that the objective appearing in (6.9) is a simple (quadratic) linear least-squares loss. In the next subsection, we will exploit this fact to derive a convex relaxation of Problem 1 via Shor's relaxation.

6.1.2.2 Applying Shor's Relaxation to Pose Graph Optimization

In this subsection we show how one can apply Shor's relaxation to derive a convex relaxation of the pose graph optimization Problem 1.

Simplifying the Maximum Likelihood Estimator. Our first step will be to rewrite Problem 1 in a more compact form that only involves rotations, and reveals the correspondence between the optimization problem (6.9) and the underlying graphs $\mathcal{G}$ and $\vec{\mathcal{G}}$ from which it is constructed. To that end, we introduce several matrices constructed from these graphs, and refer the reader to the box on the next page for a primer on (algebraic) graph theory (specifically, the *incidence* and *Laplacian* matrices).

Elements of Algebraic Graph Theory. Algebraic graph theory studies how to use algebra (*e.g.,* matrices, vectors) to represent, analyze, and extract information from graphs. Here we review some basic concepts to support the explanations and connections drawn in this chapter. We also point the reader to Chapter 1, which draws connections between graphs and probabilistic graphical models.

A *directed graph* $\overrightarrow{\mathcal{G}}$ is a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of *nodes*, and $\mathcal{E}$ is a set of *edges*, where each edge contains an ordered pair of nodes. An edge $e \in \mathcal{E}$ is in the form $e = (i, j)$, meaning that edge e, incident on nodes i and j, leaves node i and is directed towards node j (i is the *tail* of the edge, and j is the *head*).

For a graph with n nodes and m edges, the incidence matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ of a directed graph $\overrightarrow{\mathcal{G}}$ is a matrix with entries in $\{-1, 0, +1\}$ that describes the structure of the graph. Each row of $\boldsymbol{A}$ corresponds to an edge, and the column corresponding to edge $e = (i, j)$ has only two non-zero elements, one on the i-th column (equal to -1) and the other on the j-th column (equal to $+1$).[4] For instance, the incidence matrix of the graph in Figure 6.1 is:

$$\boldsymbol{A} = \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & \\ -1 & +1 & & & & & & e_{12} \\ & -1 & +1 & & & & & e_{23} \\ & & -1 & +1 & & & & e_{34} \\ & & & -1 & +1 & & & e_{45} \\ & & & & -1 & +1 & & e_{56} \\ & & & & & -1 & +1 & e_{67} \\ & +1 & & & -1 & & & e_{52} \\ +1 & & & & & -1 & & e_{61} \end{array} \tag{6.10}$$

The *Laplacian* matrix $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ is defined as $\boldsymbol{L} \triangleq \boldsymbol{A}^{\mathsf{T}} \boldsymbol{A}$ and also captures the connectivity of the graph. In particular, the i-th diagonal element of $\boldsymbol{L}$ corresponds to the node degree of the i-th node in the graph (*i.e.,* the number of nodes connected to node i), while an off-diagonal element in position (i, j) is equal to -1 if there is an edge (regardless of its orientation) connecting node i and j or is zero otherwise. For instance, the Laplacian matrix of the graph in Figure 6.1 is:

$$\boldsymbol{L} = \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & \\ +2 & -1 & & & & -1 & & x_1 \\ -1 & +3 & -1 & & -1 & & & x_2 \\ & -1 & +2 & -1 & & & & x_3 \\ & & -1 & +2 & -1 & & & x_4 \\ & -1 & & -1 & +3 & -1 & & x_5 \\ -1 & & & & -1 & +3 & -1 & x_6 \\ & & & & & -1 & +1 & x_7 \end{array} \tag{6.11}$$

We remark that the Laplacian matrix no longer captures the directionality of the edges in the graph since the off-diagonal entries are -1 when there is an edge connecting the corresponding nodes regardless of its orientation.

The smallest eigenvalue of the Laplacian matrix is always equal to zero and the corresponding eigenvector is the vector with all entries equal to one (this can be easily seen from the fact that the entries on each row sum up to zero, hence $\boldsymbol{L} \cdot \mathbf{1} = \mathbf{0}$). It turns out that the number of zero eigenvalues corresponds to the number of connected components in the graph: a connected graph (where there is path of edges between any pair of nodes, regardless of the orientation of the edges) has a single zero eigenvalue, a graph formed by two disconnected subgraphs has two zero eigenvalues, etc. Moreover, for a connected graph, the *second* smallest eigenvalue is a measure of how well-connected the graph is, and is also known as the *algebraic connectivity* or *Fiedler value* of the graph.

Let us define some key matrices that are related to the graph Laplacian matrix and that will be used in our derivation. We define the *translational weight graph* $\mathcal{W}^\tau = (\mathcal{V}, \mathcal{E}, \{\tau_{ij}\})$ to be the weighted undirected graph with node set $\mathcal{V}$, edge set $\mathcal{E}$, and edge weights τ_{ij} for $\{i,j\} \in \mathcal{E}$, and let $\boldsymbol{L}(\mathcal{W}^\tau) \in \mathbb{S}^n_+$ denote its Laplacian:

$$\boldsymbol{L}(\mathcal{W}^\tau)_{ij} = \begin{cases} \sum_{\{i,k\} \in \mathcal{E}} \tau_{ik}, & i = j, \\ -\tau_{ij}, & \{i,j\} \in \mathcal{E}, \\ 0, & \{i,j\} \notin \mathcal{E}. \end{cases} \tag{6.12}$$

This is simply a weighted version of the Laplacian matrix we defined above. Similarly, we write $\boldsymbol{L}(\tilde{G}^\rho) \in \mathbb{S}^{dn}_+$ for the *connection Laplacian* determined by the rotational measurements $\tilde{\boldsymbol{R}}_{ij}$ and precisions κ_{ij}; this is the symmetric $(d \times d)$-block-structured matrix defined by

$$\boldsymbol{L}(\tilde{G}^\rho)_{ij} \triangleq \begin{cases} \left(\sum_{\{i,k\} \in \mathcal{E}} \kappa_{ik}\right) \mathbf{I}_d, & i = j, \\ -\kappa_{ij} \tilde{\boldsymbol{R}}_{ij}, & (i,j) \in \overrightarrow{\mathcal{E}}, \\ -\kappa_{ji} \tilde{\boldsymbol{R}}_{ji}^\top, & (j,i) \in \overrightarrow{\mathcal{E}}, \\ 0_{d \times d}, & \{i,j\} \notin \mathcal{E}. \end{cases} \tag{6.13}$$

We also define a few matrices constructed from the set of translation observations $\tilde{\boldsymbol{t}}_{ij}$. We let $\tilde{\boldsymbol{V}} \in \mathbb{R}^{n \times dn}$ be the $(1 \times d)$-block-structured matrix with (i,j)-blocks determined by

$$\tilde{\boldsymbol{V}}_{ij} \triangleq \begin{cases} \sum_{\{k \in \mathcal{V} \mid (j,k) \in \overrightarrow{\mathcal{E}}\}} \tau_{jk} \tilde{\boldsymbol{t}}_{jk}^\top, & i = j, \\ -\tau_{ji} \tilde{\boldsymbol{t}}_{ji}^\top, & (j,i) \in \overrightarrow{\mathcal{E}}, \\ 0_{1 \times d}, & \text{otherwise}, \end{cases} \tag{6.14}$$

Let $\tilde{\boldsymbol{T}} \in \mathbb{R}^{m \times dn}$ be the $(1 \times d)$-block-structured matrix with rows and columns indexed by $e \in \overrightarrow{\mathcal{E}}$ and $k \in \mathcal{V}$, respectively, and whose (e,k)-block is given by

$$\tilde{\boldsymbol{T}}_{ek} \triangleq \begin{cases} -\tilde{\boldsymbol{t}}_{kj}^\top, & e = (k,j) \in \overrightarrow{\mathcal{E}}, \\ 0_{1 \times d}, & \text{otherwise}, \end{cases} \tag{6.15}$$

and $\boldsymbol{\Omega} \triangleq \operatorname{Diag}(\tau_{e_1}, \ldots, \tau_{e_m}) \in \mathbb{S}^m$ denote the diagonal matrix constructed from the translation measurement precisions. Finally, we also aggregate the rotation and translation state estimates into the block matrices $\boldsymbol{R} \triangleq \begin{pmatrix} \boldsymbol{R}_1 & \cdots & \boldsymbol{R}_n \end{pmatrix} \in \mathrm{SO}(d)^n \subset \mathbb{R}^{d \times dn}$ and $\boldsymbol{t} \triangleq \begin{pmatrix} \boldsymbol{t}_1 & \ldots & \boldsymbol{t}_n \end{pmatrix} \in \mathbb{R}^{dn}$.

With these definitions in hand, let us return to Problem 1. Observe that if we *fix* a value of the rotational states $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n$, problem (6.9) reduces to a *linear least-squares* problem in the remaining translational decision variables $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n \in \mathbb{R}^d$. Consequently, we may solve for an optimal assignment $\boldsymbol{t}^*(\boldsymbol{R})$ of the translational states as functions of the rotational states $\boldsymbol{R}$:

$$\boldsymbol{t}^*(\boldsymbol{R}) = -\operatorname{vec}\left(\boldsymbol{R}^* \tilde{\boldsymbol{V}}^\top \boldsymbol{L}(\mathcal{W}^\tau)^\dagger\right). \tag{6.16}$$

By substituting the optimal assignment (6.16) into (6.9), we may thus *analytically eliminate* the translational states from the pose-graph SLAM MLE, producing the following *simplified* (but equivalent) problem involving only the rotational states:

Problem 2 (Rotation-Only Pose Graph Optimization)

$$p^*_{\mathrm{MLE}} = \min_{\boldsymbol{R} \in \mathrm{SO}(d)^n} \operatorname{tr}\left(\tilde{\boldsymbol{Q}} \boldsymbol{R}^\mathsf{T} \boldsymbol{R}\right) \tag{6.17a}$$

$$\tilde{\boldsymbol{Q}} = \boldsymbol{L}(\tilde{G}^\rho) + \tilde{\boldsymbol{T}}^\mathsf{T} \boldsymbol{\Omega}^{\frac{1}{2}} \boldsymbol{\Pi} \boldsymbol{\Omega}^{\frac{1}{2}} \tilde{\boldsymbol{T}}, \tag{6.17b}$$

where $\boldsymbol{\Pi} \in \mathbb{R}^{m \times m}$ is the matrix of the orthogonal projection $\pi \colon \mathbb{R}^m \to \ker(\boldsymbol{A}(\overrightarrow{\mathcal{G}})\boldsymbol{\Omega}^{\frac{1}{2}})$ onto the kernel of the *weighted* incidence matrix $\boldsymbol{A}(\overrightarrow{\mathcal{G}})\boldsymbol{\Omega}^{\frac{1}{2}}$ of $\overrightarrow{\mathcal{G}}$.

We observe that (6.17) involves only n *rotation matrices* (rather than n *poses*) and the problem now resembles the standard problem of multiple rotation averaging [442] (but with a more involved expression for the data matrix $\tilde{\boldsymbol{Q}}$). On the more technical side, we note that although $\boldsymbol{\Pi}$ is generically dense, by exploiting the fact that it is derived from the graph $\overrightarrow{\mathcal{G}}$, one can show that it admits the decomposition:

$$\boldsymbol{\Pi} = \mathbf{I}_m - \boldsymbol{\Omega}^{\frac{1}{2}} \bar{\boldsymbol{A}}(\overrightarrow{\mathcal{G}})^\mathsf{T} \boldsymbol{L}^{-\mathsf{T}} \boldsymbol{L}^{-1} \bar{\boldsymbol{A}}(\overrightarrow{\mathcal{G}}) \boldsymbol{\Omega}^{\frac{1}{2}} \tag{6.18}$$

where $\bar{\boldsymbol{A}}(\overrightarrow{\mathcal{G}})\boldsymbol{\Omega}^{\frac{1}{2}} = \boldsymbol{L}\boldsymbol{Q}_1$ is a thin LQ decomposition of $\bar{\boldsymbol{A}}(\overrightarrow{\mathcal{G}})\boldsymbol{\Omega}^{\frac{1}{2}}$ and $\bar{\boldsymbol{A}}(\overrightarrow{\mathcal{G}})$ is the *reduced incidence matrix* of $\overrightarrow{\mathcal{G}}$ obtained by deleting one of $\boldsymbol{A}(\overrightarrow{\mathcal{G}})$'s rows. Note that expression (6.18) requires only the lower-triangular factor $\boldsymbol{L}$, which will be sparse whenever the underlying graph $\overrightarrow{\mathcal{G}}$ is, and can be obtained efficiently in practice. The sparse decomposition (6.17b)–(6.18) of the data matrix $\tilde{\boldsymbol{Q}}$ will play a critical role in our implementation of efficient optimization methods (cf. Sec. 6.1.2.3).

Forming the Relaxation. Now we derive the semidefinite relaxation of Problem 1 that we will solve in practice, taking advantage of the simplified form (6.17).

We begin by relaxing the condition $\boldsymbol{R} \in \mathrm{SO}(d)^n$ to $\boldsymbol{R} \in \mathrm{O}(d)^n$. The advantage of the latter condition versus the former is that since orthogonal matrices are defined by a set of (quadratic) orthonormality constraints, the orthogonal relaxation of Problem 2 is a homogeneous QCQP. Indeed, writing

$$\begin{aligned} \mathrm{BlockDiag}_{d\times d} &\colon \mathbb{R}^{dn \times dn} \to \mathbb{R}^{d \times dn} \\ \mathrm{BlockDiag}_{d\times d}(\boldsymbol{M}) &\triangleq (\boldsymbol{M}_{11}, \ldots, \boldsymbol{M}_{nn}) \end{aligned} \tag{6.19}$$

for the linear map that extracts the n diagonal blocks of a $(d \times d)$-block-structured matrix $\boldsymbol{M}$, we may expresss the orthogonal relaxation of (6.17) in an extrinsically-constrained form as

$$p^*_O = \min_{\boldsymbol{R} \in \mathbb{R}^{d \times dn}} \operatorname{tr}\left(\tilde{\boldsymbol{Q}} \boldsymbol{R}^\mathsf{T} \boldsymbol{R}\right) \quad \text{s.t.} \quad \mathrm{BlockDiag}_{d\times d}(\boldsymbol{R}^\mathsf{T}\boldsymbol{R}) = (\mathbf{I}_d, \ldots, \mathbf{I}_d). \tag{6.20}$$

In turn, since (6.20) is a homogeneous QCQP, we may apply the matrix-valued generalization of Shor's relaxation to obtain the following semidefinite relaxation of the simplified pose graph optimization problem (6.17):

Problem 3 (Semidefinite Relaxation for Pose Graph Optimization)

$$p^*_{\text{SDP}} = \min_{\boldsymbol{Z} \in \mathbb{S}_+^{dn}} \operatorname{tr}\left(\tilde{\boldsymbol{Q}}\boldsymbol{Z}\right) \quad \text{s.t.} \quad \operatorname{BlockDiag}_{d\times d}(Z) = (\mathbf{I}_d, \ldots, \mathbf{I}_d). \tag{6.21}$$

As we saw in Section 6.1.1, this construction immediately implies that $p^*_{\text{MLE}} \geq p^*_O \geq p^*_{\text{SDP}}$. Moreover, if after solving the SDP relaxation (6.21), it so happens that the recovered minimizer $\boldsymbol{Z}^* \in \mathbb{S}_+^{dn}$ admits a rank-d factorization of the form $\boldsymbol{Z}^* = \boldsymbol{R}^{*\mathsf{T}}\boldsymbol{R}^*$ with $\boldsymbol{R}^* \in \mathrm{SO}(d)^n$, then $\boldsymbol{R}^*$ will itself be a *globally optimal* solution of the pose graph optimization Problem 2. The remarkable fact that justifies our interest in the relaxation (6.21) is that this favorable situation *actually occurs* in practice. Specifically, we have the following theorem (see [943] for a proof):

Theorem 6.1 (Exact Recovery of PGO Solutions from Problem 3) *Let $\bar{\boldsymbol{Q}}$ be the matrix of the form (6.17b) constructed using the ground-truth relative transforms $\bar{\boldsymbol{T}}_{ij}$ in (6.8). There exists a constant $\beta \triangleq \beta(\bar{\boldsymbol{Q}}) > 0$ (depending upon $\bar{\boldsymbol{Q}}$) such that, if $\|\tilde{\boldsymbol{Q}} - \bar{\boldsymbol{Q}}\|_2 < \beta$, then:*

1 *The semidefinite relaxation Problem 3 has a unique solution $\boldsymbol{Z}^*$, and*
2 *$\boldsymbol{Z}^* = \boldsymbol{R}^{*\mathsf{T}}\boldsymbol{R}^*$, where $\boldsymbol{R}^* \in \mathrm{SO}(d)^n$ is a minimizer of the maximum likelihood estimation Problem 2.*

In brief, Theorem 6.1 guarantees that as long as the noise corrupting the available measurements $\tilde{\boldsymbol{T}}_{ij}$ is not too large, we can recover a *global minimizer* $\boldsymbol{R}^*$ of Problem 2 (and thus also a global minimizer $(\boldsymbol{R}^*, \boldsymbol{t}^*)$ of Problem 1 via (6.16)) by solving the SDP relaxation (6.21).

6.1.2.3 Efficiently Solving the Relaxation via the Riemannian Staircase

As a semidefinite program, Problem 3 can *in principle* be solved efficiently, *i.e.,* in polynomial time. However, in practice the high computational cost of storing and manipulating the dense matrix decision variable $\boldsymbol{Z}$ appearing in (6.21) prevents general-purpose interior-point methods from scaling effectively to problems in which the dimension of $\boldsymbol{Z}$ is greater than a few thousand. Unfortunately, the instances of (6.21) arising in robotics and computer vision applications are typically one to two orders of magnitude larger than this maximum effective problem size, placing them well beyond the reach of general-purpose techniques. Consequently, in this subsection we develop a *specialized*, *structure-exploiting* optimization procedure that is capable of solving large-scale instances of Problem 3 efficiently.

Exploiting Low-rank Structure. The main idea behind our approach is to exploit the existence of *low-rank* solutions to Problem 3. Specifically, note that while the *decision variable* $\boldsymbol{Z}$ appearing in the relaxation (6.21) is a generic high-dimensional PSD matrix, Theorem 6.1 guarantees that the *solution* $\boldsymbol{Z}^*$ we seek admits a very concise description in the factored form $\boldsymbol{Z}^* = \boldsymbol{R}^{*\mathsf{T}}\boldsymbol{R}^*$ whenever exactness obtains. Moreover, it turns out that even when exactness fails to hold, minimizers of Problem 3 typically have a rank r not much greater than d, and therefore also admit a concise symmetric rank decomposition of the form $\boldsymbol{Z}^* = \boldsymbol{Y}^{*\mathsf{T}}\boldsymbol{Y}^*$ for some $Y^* \in \mathbb{R}^{r\times dn}$.

In their seminal work, Burer, Samuel and Monteiro, Renato D C [131], Burer and Monteiro [130], proposed an elegant general approach to exploit the existence of such low-rank solutions: simply replace every instance of the decision variable $\boldsymbol{Z}$ in (6.21) with a *symmetric rank-r factorization* of the form $\boldsymbol{Y}^\mathsf{T}\boldsymbol{Y}$ (for some $\boldsymbol{Y} \in \mathbb{R}^{r\times dn}$) to produce the following *Burer-Monteiro factorization* of (6.21):

$$p^*_{\mathrm{SDPLR}}(r) = \min_{\boldsymbol{Y}\in\mathbb{R}^{r\times dn}} \operatorname{tr}\left(\tilde{\boldsymbol{Q}}\boldsymbol{Y}^\mathsf{T}\boldsymbol{Y}\right) \quad \text{s.t.} \quad \operatorname{BlockDiag}_{d\times d}(\boldsymbol{Y}^\mathsf{T}\boldsymbol{Y}) = (\mathbf{I}_d, \ldots, \mathbf{I}_d). \tag{6.22}$$

We remark that since $\boldsymbol{Y}^\mathsf{T}\boldsymbol{Y}$ is symmetric and PSD by construction, in (6.22) it is no longer necessary to explicitly enforce the positive-semidefiniteness constraint from the original SDP (6.21). Moreover, we note that this problem is strikingly similar to the problem we relaxed (6.20), with the important distinction that now the matrix $\boldsymbol{Y}$ has size $r \times dn$ instead of $d \times dn$, with $r > d$; in other words, (6.22) reformulates the problem in a higher-dimensional space compared to (6.20).

If the maximum rank parameter r in (6.22) is chosen to be "small" (*i.e., $r \ll dn$*), then $\dim(\mathbb{R}^{r\times dn}) = rnd \ll (dn+1)dn/2 = \dim(\mathbb{S}^{dn}_+)$; that is, the search space for (6.22) is *much* lower-dimensional than the search space for (6.21). Consequently, Burer, Samuel and Monteiro, Renato D C proposed to apply fast nonlinear programming algorithms to the low-dimensional NLP (6.22) in order to search for a low-rank factor $\boldsymbol{Y}^* \in \mathbb{R}^{r\times dn}$ of a minimizer $\boldsymbol{Z}^* = \boldsymbol{Y}^{*\mathsf{T}}\boldsymbol{Y}^*$ of the original SDP (6.21).

Exploiting Geometric Structure. Note that if we additionally partition $\boldsymbol{Y}$ into $r \times d$ blocks as $\boldsymbol{Y} = (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n) \in \mathbb{R}^{r\times dn}$, then the block-diagonal constraints appearing in (6.22) are equivalent to $\boldsymbol{Y}_i^\mathsf{T}\boldsymbol{Y}_i = \mathbf{I}_d$ for all $i \in [n]$; geometrically, this condition states that the columns of each block $\boldsymbol{Y}_i \in \mathbb{R}^{r\times d}$ form an *orthonormal frame.* In general, the set of all orthonormal k-frames in $\mathbb{R}^p$,

$$\operatorname{St}(k,p) \triangleq \left\{\boldsymbol{Y} \in \mathbb{R}^{p\times k} \mid \boldsymbol{Y}^\mathsf{T}\boldsymbol{Y} = \mathbf{I}_k\right\}, \tag{6.23}$$

forms a smooth compact matrix manifold, called the *Stiefel manifold.* This implies that the equality-constrained nonlinear program (6.22) is equivalent to the following *unconstrained* optimization problem defined on a product of Stiefel manifolds:

Problem 4 (Burer-Monteiro-factored SDP relaxation as manifold optimization)

$$p^*_{\mathrm{SDPLR}}(r) = \min_{\boldsymbol{Y}\in\operatorname{St}(d,r)^n} \operatorname{tr}\left(\tilde{\boldsymbol{Q}}\boldsymbol{Y}^\mathsf{T}\boldsymbol{Y}\right). \tag{6.24}$$

While formulations (6.22) and (6.24) are equivalent, the latter provides important *computational* advantages. In particular, recognizing that the feasible set is a product of Stiefel manifolds enables us to apply specialized algorithms for optimization over smooth manifolds, which are substantially *simpler*, *faster*, and *more accurate* than general-purpose equality-constrained nonlinear programming techniques [113].

Ensuring Global Optimality. While the reduction from Problem 3 to Problem 4 dramatically reduces the size of the optimization problem that needs to be solved, it comes at the expense of (re)introducing the quadratic orthonormality constraints (6.23), which are nonconvex. It may therefore not be clear whether anything has really been gained by relaxing Problem 2 to Problem 4, since it appears that we may have simply replaced one difficult nonconvex optimization problem with another. The following remarkable result (adapted from Boumal et al. [111]) justifies this approach:

Theorem 6.2 (A Sufficient Condition for Global Optimality in Problem 4) *If $\boldsymbol{Y} \in \mathrm{St}(d,r)^n$ is a (row) rank-deficient second-order critical point of Problem 4, then $\boldsymbol{Y}$ is a global minimizer of Problem 4 and $\boldsymbol{Z}^* = \boldsymbol{Y}^\mathsf{T}\boldsymbol{Y}$ is a solution of the semidefinite relaxation Problem 3.*

Theorem 6.2 immediately suggests a simple procedure, the *Riemannian Staircase*, for recovering solutions $\boldsymbol{Z}^*$ of Problem 3 by applying fast *local* optimization algorithms to a *sequence* of instances of Problem 4. In brief, starting at some (small) initial maximum rank $r \geq d$, we apply a local solver to (6.24) (more precisely, a second-order Riemannian optimization algorithm) to recover a second-order critical point $\boldsymbol{Y}^* \in \mathrm{St}(d,r)^n$. If $\boldsymbol{Y}^*$ is rank-deficient, then Theorem 6.2 proves that $\boldsymbol{Y}^*$ is a *global* minimizer of (6.24), and $\boldsymbol{Z}^* = \boldsymbol{Y}^{*\mathsf{T}}\boldsymbol{Y}^*$ is a solution of (6.21). On the other hand, if $\boldsymbol{Y}^*$ is *not* rank deficient, we can simply increase the maximum rank r and try again. Note that since *every* $\boldsymbol{Y} \in \mathbb{R}^{r\times dn}$ is (row) rank-deficient for $r \geq dn+1$, the Riemannian Staircase is *guaranteed* to terminate with an optimal solution $\boldsymbol{Y}^*$ after finitely many iterations. However, typically only one or two "stairs" suffice.

Finally, we remark that from a practical standpoint, the Riemannian Staircase functions as a lightweight meta-algorithm that "wraps around" the same class of fast (second-order) *local* optimization algorithms commonly applied to SLAM problems in practice. This approach thus enables us to preserve the speed of current state-of-the-art SLAM techniques while *additionally* guaranteeing the recovery of *globally optimal* solutions, thereby achieving the best of both worlds.

6.1.2.4 Rounding the Solution

We have just seen that the Riemannian Staircase provides an efficient algorithm for recovering a low-rank factor $\boldsymbol{Y}^* \in \mathrm{St}(d,r)^n$ of a solution $\boldsymbol{Z}^* = \boldsymbol{Y}^{*\mathsf{T}}\boldsymbol{Y}^*$ of the relaxation Problem 3. However, ideally we would like to extract an optimal solution $\boldsymbol{R}^* \in \mathrm{SO}(d)^n$ of the pose graph optimization Problem 2 from $\boldsymbol{Z}^*$ whenever the relaxation (6.21) is exact, and a *feasible approximate* solution $\hat{\boldsymbol{R}} \in \mathrm{SO}(d)^n$ otherwise. In this subsection, we describe an efficient rounding procedure that accomplishes these aims by operating directly on the low-rank factor $\boldsymbol{Y}^*$ (*i.e.,without* the need to explicitly construct the dense, high-dimensional matrix $\boldsymbol{Z}^*$).

Algorithm 3 The SE-Sync algorithm

```
Input: An initial point $\boldsymbol{Y} \in \mathrm{St}(d, r_0)^n$, $r_0 \geq d + 1$.
Output: A feasible estimate $\hat{\boldsymbol{T}} \in \mathrm{SE}(d)^n$ for the maximum likelihood estimation
    Problem 1, and the lower bound $p^*_{\mathrm{SDP}}$ for Problem 1's optimal value.
function SE-SYNC($\boldsymbol{Y}$)
    Set $\boldsymbol{Y}^* \leftarrow$ RIEMANNIANSTAIRCASE($\boldsymbol{Y}$).
    Set $p^*_{\mathrm{SDP}} \leftarrow F(\tilde{\boldsymbol{Q}} \boldsymbol{Y}^{*\mathsf{T}} \boldsymbol{Y}^*)$.
    Set $\hat{\boldsymbol{R}} \leftarrow$ ROUNDSOLUTION($\boldsymbol{Y}^*$).
    Recover the optimal translational estimates $\hat{\boldsymbol{t}}$ corresponding to $\hat{\boldsymbol{R}}$ via (6.16).
    Set $\hat{\boldsymbol{T}} \leftarrow (\hat{\boldsymbol{t}}, \hat{\boldsymbol{R}})$.
    return $\left\{\hat{\boldsymbol{T}}, p^*_{\mathrm{SDP}}\right\}$
end function
```

The main insight that underpins our approach is that, in the event that the relaxation (6.21) is exact, $\boldsymbol{R}^*$, $\boldsymbol{Z}^*$, and $\boldsymbol{Y}^*$ will satisfy the relation

$$\boldsymbol{Y}^{*\mathsf{T}} \boldsymbol{Y}^* = \boldsymbol{Z}^* = \boldsymbol{R}^{*\mathsf{T}} \boldsymbol{R}^*. \tag{6.25}$$

In this case, equation (6.25) implies that the low-rank factor $\boldsymbol{Y}^* \in \mathbb{R}^{r \times dn}$ actually has rank d, and consequently that $\boldsymbol{R}^*$ can be recovered from $\boldsymbol{Y}^*$ by computing a thin singular value decomposition of the latter. More generally, in the event that (6.21) is *not* exact, we can still recover an optimal *rank-d approximation* $\hat{\boldsymbol{R}} \in \mathbb{R}^{d \times dn}$ of $\boldsymbol{Y}^*$ using a *truncated* singular value decomposition, and then project the individual $d \times d$ blocks $\hat{\boldsymbol{R}}_i$ of $\hat{\boldsymbol{R}}$ onto $\mathrm{SO}(d)$ (again using an SVD) to produce a *feasible approximate solution* of Problem 2.

6.1.2.5 SE-Sync: The Complete Algorithm

Combining the efficient SDP optimization approach of Section 6.1.2.3 with the rounding procedure of Section 6.1.2.4 produces *SE-Sync* (Algorithm 3), our certifiably correct algorithm for pose-graph SLAM [943].

When applied to an instance of pose graph optimization, SE-Sync returns a feasible point $\hat{\boldsymbol{T}} \in \mathrm{SE}(d)^n$ of the maximum likelihood estimation Problem 1 together with a lower bound $p^*_{\mathrm{SDP}} \leq p^*_{\mathrm{MLE}}$ on its optimal value. This lower bound in turn provides an *upper* bound on the suboptimality of *any* feasible point $\boldsymbol{T} = (\boldsymbol{t}, \boldsymbol{R}) \in \mathrm{SE}(d)^n$ as a solution of Problem 1 according to

$$F(\tilde{\boldsymbol{Q}} \boldsymbol{R}^{\mathsf{T}} \boldsymbol{R}) - p^*_{\mathrm{SDP}} \geq F(\tilde{\boldsymbol{Q}} \boldsymbol{R}^{\mathsf{T}} \boldsymbol{R}) - p^*_{\mathrm{MLE}}. \tag{6.26}$$

Moreover, in the case that the relaxation in Problem 3 is exact, the estimate $\hat{\boldsymbol{T}} = (\hat{\boldsymbol{t}}, \hat{\boldsymbol{R}}) \in \mathrm{SE}(d)^n$ returned by Algorithm 3 *attains* this lower bound:

$$F(\tilde{\boldsymbol{Q}} \hat{\boldsymbol{R}}^{\mathsf{T}} \hat{\boldsymbol{R}}) = p^*_{\mathrm{SDP}}. \tag{6.27}$$

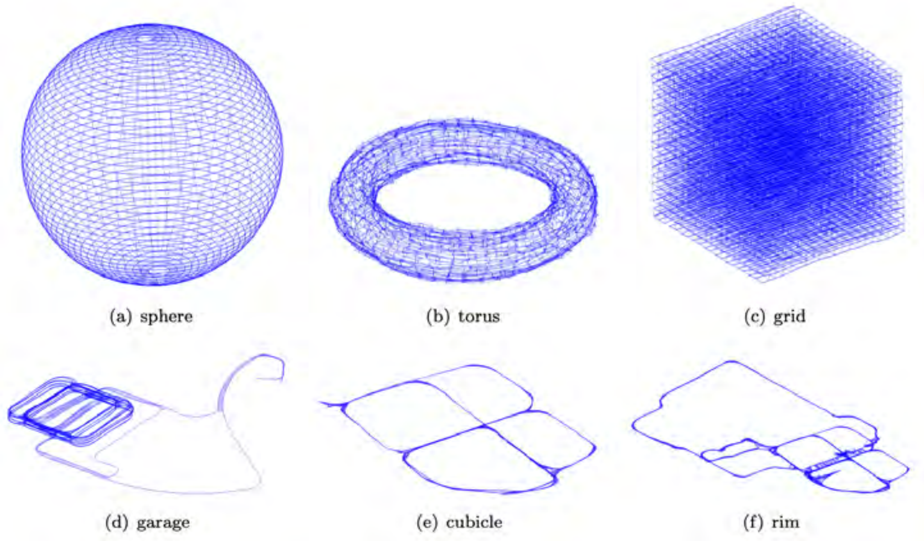

Figure 6.2 Globally optimal solutions for pose graph optimization benchmarking datasets (From [943]).

Consequently, verifying *a posteriori* that (6.27) holds provides a *computational certificate* of $\hat{\boldsymbol{T}}$'s correctness as a solution of Problem 1. SE-Sync is thus a *certifiably correct* algorithm for pose-graph SLAM, as claimed.

Sample certifiably optimal results obtained with SE-Sync are shown in Fig. 6.2 and discussed in depth in [943]. The paper [943] also reports a runtime analysis showing that the algorithm can be as fast if not faster than traditional local solvers.

6.1.3 Landmark-based SLAM

While in the previous section we have shown how to obtain a fast certifiable algorithm for pose graph optimization, in this section we show that the same derivation applies to landmark-based SLAM, specifically for the case where the robot takes bearing and range measurements to landmarks.

Let $\boldsymbol{m}_{n+1}, \ldots, \boldsymbol{m}_{n+\ell} \in \mathbb{R}^d$ be the set of ℓ landmark positions that we wish to estimate in addition to the n robot poses. It turns out that these new map variables can be easily integrated into the existing implementation of SE-Sync. To do this, we treat the map point variables as *pose variables without any rotational component.*

We assume the we have a set of N_m landmark measurements, $\{\tilde{\boldsymbol{m}}_{ik}\}$, describing the relative position of landmark $\boldsymbol{m}_k$ with respect to the i-th pose of the robot. We also assume that these measurements are corrupted by *additive mean-zero isotropic*

Gaussian noise —the same form as the translation measurements in (6.8):

$$\tilde{\boldsymbol{m}}_{ik} = \bar{\boldsymbol{m}}_{ik} + \boldsymbol{m}^{\epsilon}_{ik}, \quad \boldsymbol{m}^{\epsilon}_{ij} \sim \mathcal{N}\left(0, \mu_{ij}^{-1} I_d\right). \tag{6.28}$$

To track these new measurements, we augment the pose-graph, $\overrightarrow{\mathcal{G}}$, with a new set of vertices, $\mathcal{V}_m$, and edges, $\overrightarrow{\mathcal{E}}_m$, with a one-to-one correspondence to the map variables and landmark measurements, respectively. Moreover, we denote with $\overrightarrow{\mathcal{E}}_r$ the edges corresponding to relative pose measurements of the robot pose (*e.g.,* odometry), which are assumed to follow the same measurement model as in (6.8). The maximum likelihood estimation problem given the landmark measurements and the relative pose measurements becomes:

Problem 5 (Landmark-based SLAM)

$$\begin{aligned} \min_{\substack{\boldsymbol{R}_i \in \mathrm{SO}(d) \\ \boldsymbol{t}_i \in \mathbb{R}^d, \boldsymbol{m}_k \in \mathbb{R}^d}} \sum_{(i,j)\in \overrightarrow{\mathcal{E}}_r} & \kappa_{ij} \|\boldsymbol{R}_j - \boldsymbol{R}_i \tilde{\boldsymbol{R}}_{ij}\|_F^2 + \tau_{ij} \left\|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \tilde{\boldsymbol{t}}_{ij}\right\|_2^2 \\ & + \sum_{(i,k)\in \vec{\boldsymbol{\mathcal{E}}}_m} \mu_{ik} \left\|\boldsymbol{m}_k - \boldsymbol{t}_i - \boldsymbol{R}_i \tilde{\boldsymbol{m}}_{ik}\right\|^2. \end{aligned} \tag{6.29}$$

It turns out that the remainder of the development we presented in the previous section can be used *as is*, if we treat the map variables as if they were pose translations. The optimal rotation variables can be found by applying the SE-Sync algorithm with a cost matrix $\tilde{\boldsymbol{Q}}$ that has been modified to include the effect of the map variables. In particular, the weight matrix $\boldsymbol{\Omega}$ and measurement matrix $\tilde{\boldsymbol{T}}$ are redefined as follows:

$$\begin{aligned} \boldsymbol{\Omega} &= \mathrm{BlockDiag}(\boldsymbol{\Omega}_\tau, \boldsymbol{\Omega}_\mu), \\ \boldsymbol{\Omega}_\tau &= \mathrm{Diag}(\tau_1, \dots, \tau_{N_p}), \\ \boldsymbol{\Omega}_\mu &= \mathrm{Diag}(\mu_1, \dots, \mu_{N_m}), \end{aligned} \qquad \tilde{T}_{ek} = \begin{cases} -\tilde{\boldsymbol{t}}_{kj}^T, & e = (k,j) \in \overrightarrow{\mathcal{E}}_r, \\ -\tilde{\boldsymbol{m}}_{kj}^T, & e = (k,j) \in \overrightarrow{\mathcal{E}}_m, \\ \boldsymbol{0}_{1\times d}, & \text{otherwise} \end{cases} \tag{6.30}$$

Additionally, the projection matrix $\boldsymbol{\Pi}$ is modified to account for the form of the new measurement graph, $\overrightarrow{\mathcal{G}} = (\mathcal{V} \cup \mathcal{V}_m, \overrightarrow{\mathcal{E}}_r \cup \overrightarrow{\mathcal{E}}_m)$. Note that the connection Laplacian $\boldsymbol{L}(\tilde{G}^\rho)$ remains unaffected by the new map measurements.

Once the optimal rotations have been found, both the pose translations and the map points can be recovered in closed form similarly to (6.16):

$$\begin{bmatrix} \boldsymbol{t}^{*T} & \boldsymbol{m}^{*T} \end{bmatrix}^T = -\mathrm{vec}\left(\boldsymbol{R}^* \tilde{\boldsymbol{V}}^T \boldsymbol{L}(\mathcal{W}^\tau)^\dagger\right), \tag{6.31}$$

where we have assumed that the pose translation variables and map points have been ordered appropriately and $\tilde{\boldsymbol{V}}$ and $\boldsymbol{L}(\mathcal{W}^\tau)^\dagger$ have also been updated to include the map point measurements similarly to (6.30).

It is often the case in typical SLAM problems that there are many more map points than poses. As such, when including map variables in the SE-Sync formulation, it is important to ensure the algorithm remains efficient with respect to the number of map points in the problem. In the case where the number of map points

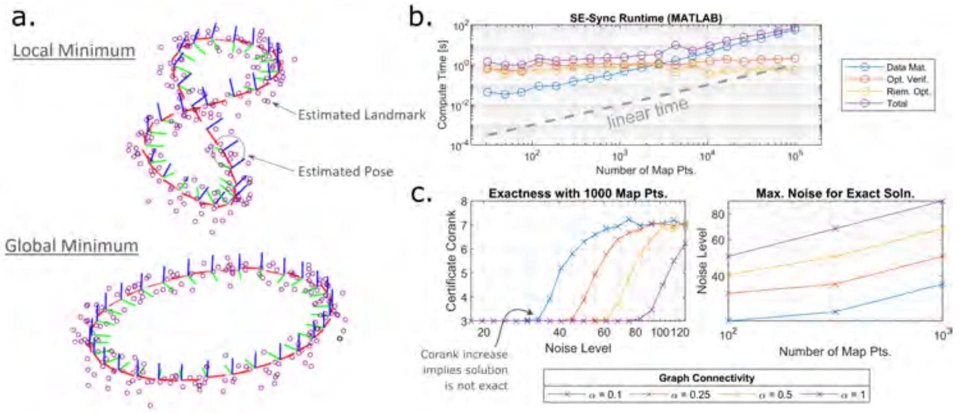

Figure 6.3 (a) An example of a local and global minimum for a simple landmark-based SLAM problem. (b) Runtime for the example shown in (a). Runtime increases linearly with respect to the size of the map and the bottleneck is the construction of the data matrix, $\tilde{\boldsymbol{Q}}$ (shown in blue). (c) A study of the *exactness* of the relaxation via corank of a *certificate matrix* (exact when this metric is three). The noise on the measurements was set to a baseline standard deviation – 0.866 meters for translation and 0.573 degrees for rotation – scaled by the "Noise Level" multiplier indicated in the plots. In general, the noise level for which the problem remains exact increases as the number of map points and the pose-to-map connectivity increase. From [473] (©2023 IEEE).

is large, the bottleneck of the SE-Sync algorithm becomes the formation of the cost matrix $\tilde{\boldsymbol{Q}}$. It has been shown that the classical *Schur-complement trick* can be used to ensure that this matrix can be constructed with time complexity that is linear in the number of landmarks [473]. This linear dependence can be seen clearly in Figure 6.3(b), where it is can also be seen that the computational cost of other components of SE-Sync algorithm does not increase with the number of landmarks.

Finally, the inclusion of the landmarks in the formulation has been shown to affect the exactness of the SDP relaxation. In particular, increasing the number of landmarks and the connectivity of the pose-to-map measurement graph have been shown to improve the exactness of the relaxation. More specifically, they increase the noise level for which a given problem has an exact relaxation. A demonstration of the effect of these parameters on a simple SLAM problem are shown in Figure 6.3(c).

6.1.4 Extensions: Range Measurements, Anisotropic Noise, and Outliers

This section shows that the machinery presented above (*i.e.,* Shor's relaxation and the Riemannian Staircase solver) can be extended to other SLAM problems (Section 6.1.4.1). Moreover, we discuss more general tools to obtain semidefinite relaxations (which can be understood as a generalization of Shor's relaxation) that further expand the set of certifiable algorithms for SLAM, but create additional challenges when solving the resulting SDP (Section 6.1.4.2).

6.1.4.1 A Fast Certifiable Algorithm for Range-Aided SLAM

In the previous section, we showed how to develop certifiable algorithms for SLAM problems where the measurements are relative positions and relative rotations. Here we show that the same approach can be applied to problems involving range measurements, following the results presented in [846]. Specifically, we assume that we can measure distances $\tilde{r}_{ij}$ between variables i and j, *e.g.,* the distance between two robot poses or between a landmark and a robot pose. The optimization problem for range-aided SLAM can be stated as follows.

Problem 6 (Range-Aided SLAM)

$$\begin{aligned} \min_{\substack{\boldsymbol{R}_i \in \mathrm{SO}(d) \\ \boldsymbol{t}_i \in \mathbb{R}^d}} \sum_{(i,j)\in \overrightarrow{\mathcal{E}}} & \kappa_{ij} \|\boldsymbol{R}_j - \boldsymbol{R}_i \tilde{\boldsymbol{R}}_{ij}\|_F^2 + \tau_{ij} \left\|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \tilde{\boldsymbol{t}}_{ij}\right\|_2^2 \\ & + \sum_{(i,j)\in \overrightarrow{\mathcal{E}}_d} \gamma_{ij} \left(\|\boldsymbol{t}_j - \boldsymbol{t}_i\| - \tilde{r}_{ij}\right)^2 \end{aligned} \tag{6.32}$$

The first line in the objective function of Problem 6 is the same we have used in the previous sections: these are relative pose measurements, *e.g.,* corresponding to the robot odometry. The difference in this problem is the second line, which contains terms corresponding to the distance measurements $\tilde{r}_{ij}$ (the set $\overrightarrow{\mathcal{E}}_d$ is the set of pairs (i, j) such that a range measurement is available), weighted by the inverse variance γ_{ij} of these measurements. The range-only measurements $\tilde{r}_{ij}$ are distinct in that they provide information about the relative distance magnitude between poses, i and j, but no information about the bearing or relative orientation. Moreover, Problem 6 can be used to model a variety of practical SLAM problems, from landmark-based SLAM with distance measurements to landmarks, to multi-robot SLAM problems where the robots take relative range measurements.[5]

The challenge in applying Shor's relaxation to Problem 6 is that (6.32) is *not* a QCQP: expanding the square $(\|\boldsymbol{t}_j - \boldsymbol{t}_i\| - \tilde{r}_{ij})^2 = \|\boldsymbol{t}_j - \boldsymbol{t}_i\|^2 + 2\tilde{r}_{ij}\|\boldsymbol{t}_j - \boldsymbol{t}_i\| + \tilde{r}_{ij}^2$ reveals that this expression is *not* quadratic in the variables $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$, due to the unsquared norm term. To address this issue, the work [846] proposes an elegant

[5] Ultra-wideband (UWB) radios are popular sensors to obtain distance-only measurements.

reformulation of (6.32), which introduces auxiliary unit vectors $\boldsymbol{b}_{ij}$ (with $\|\boldsymbol{b}_{ij}\| = 1$):

$$\begin{aligned} \min_{\substack{\boldsymbol{R}_i \in \mathrm{SO}(d) \\ \boldsymbol{t}_i \in \mathbb{R}^d \\ \boldsymbol{b}_{ij} \in S^{d-1}}} \sum_{(i,j) \in \overrightarrow{\mathcal{E}}} & \kappa_{ij} \|\boldsymbol{R}_j - \boldsymbol{R}_i \tilde{\boldsymbol{R}}_{ij}\|_F^2 + \tau_{ij} \left\|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \tilde{\boldsymbol{t}}_{ij}\right\|_2^2 \\ & + \sum_{(i,j) \in \overrightarrow{\mathcal{E}}_d} \gamma_{ij} \left\|\boldsymbol{t}_j - \boldsymbol{t}_i - \tilde{r}_{ij} \boldsymbol{b}_{ij}\right\|^2, \end{aligned} \tag{6.33}$$

Intuitively, formulation (6.33) also infers the *bearing* $\boldsymbol{b}_{ij}$ between variables i and j for which a range measurement $\tilde{r}_{ij}$ is available, and then recasts the corresponding term in the objective function, namely $\left\|\boldsymbol{t}_j - \boldsymbol{t}_i - \tilde{r}_{ij} \boldsymbol{b}_{ij}\right\|^2$, as a range-and-bearing measurement (similar to the ones we have seen in the previous sections). The advantage in doing so is that (6.33) is now a QCQP, since the last term in the objective is now quadratic in the unknowns, and the additional unit-norm constraints $\|\boldsymbol{b}_{ij}\|^2 = 1$ on the $\boldsymbol{b}_{ij}$ are quadratic as well. The new formulation can then be relaxed to an SDP using Shor's relaxation and solved efficiently using the Riemannian Staircase, leading to CORA [846], a fast certifiable algorithm for range-aided SLAM.

The work [846] also evaluates the exactness of the relaxation, showing that now the relaxation becomes more brittle compared to the cases discussed in the previous sections. For instance, when used to solve multi-robot range-aided SLAM problems, the relaxation is typically not exact unless there are also relative pose measurements between the robots (*i.e.,* range-only measurements between the robots do not suffice to obtain exact relaxations). However, the relaxation is still exact in a variety of practical SLAM problems, and allows computing certifiably optimal solutions with runtime comparable to local solvers. More generally, [846] shows that the connectivity of the graph underlying the SLAM problem largely impacts the exactness of the SDP relaxation, a phenomenon also observed in [943, 473]. As we will see in Section 6.2, the graph connectivity also affects the accuracy of the SLAM estimate, creating very interesting graph-theoretic insights into the SLAM problem.

6.1.4.2 Certifiable Algorithms Beyond Shor's Relaxation: Anisotropic Noise and Outliers

So far we have reviewed certifiable algorithms (and fast solvers) for SLAM problems that can be reformulated as QCQP. Unfortunately, not all SLAM instances fall into this category. Below, we consider a broader class of SLAM problems, namely, problems with anisotropic measurement noise and problems with outliers. The interesting observation behind these problems is that, while they can no longer be written as QCQP, they can often be formulated as Polynomial Optimization Problems (POPs), a generalization of QCQPs where both the objective and constraints are polynomial functions (instead of quadratic functions). This observation is important since there exists a generalization of Shor's relaxation, namely the *Moment (or Lasserre's) Relaxation*, that allows deriving SDP relaxations for POPs, thus enabling the design of certifiable algorithms for a broader set of problems. Below, we review examples of SLAM problems that can cannot be written as QCQPs, and then provide an overview of the Moment Relaxation and practical considerations.

Example: Landmark-based SLAM with Anisotropic Noise. In previous sections, we assumed the measurement noise to be isotropic. However, measurements produced by common sensing modalities (*e.g.,* stereo cameras, LiDAR, and RADAR) typically exhibit anisotropic noise. For instance, the measurement of the position of a landmark as observed by a stereo camera is typically more uncertain along the viewing direction of the cameras, due to the uncertainty induced by the stereo matching and triangulation process. More formally, the measurement model for the landmark measurements becomes $\tilde{\boldsymbol{m}}_{ik} = \boldsymbol{R}_i^\mathsf{T}(\boldsymbol{m}_k - \boldsymbol{t}_i) + \boldsymbol{\epsilon}_{ik}$, where $\boldsymbol{\epsilon}_{ik} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_{ik})$ and $\mathbf{W}_{ik}$ is an anisotropic covariance (*i.e.,* $\mathbf{W}_{ik}$ cannot be written as a scalar multiple of the identity matrix). In the presence of anisotropic noise, we need to generalize the landmark-based SLAM Problem 5 as follows [473]:

Problem 7 (Landmark-based SLAM with Anisotropic Noise)

$$\min_{\substack{\boldsymbol{R}_i \in \mathrm{SO}(d) \\ \boldsymbol{t}_i \in \mathbb{R}^d,\ \boldsymbol{m}_k \in \mathbb{R}^d}} \sum_{(i,j)\in\vec{\mathcal{E}}} \kappa_{ij} \|\boldsymbol{R}_j - \boldsymbol{R}_i \tilde{\boldsymbol{R}}_{ij}\|_F^2 + \tau_{ij} \left\|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \tilde{\boldsymbol{t}}_{ij}\right\|_2^2 + \sum_{(i,k)\in\vec{\boldsymbol{\mathcal{E}}}_m} \|\boldsymbol{R}_i^\mathsf{T}(\boldsymbol{m}_k - \boldsymbol{t}_i) - \tilde{\boldsymbol{m}}_{ik}\|_{\mathbf{W}_{ik}}^2 \tag{6.34}$$

In (6.34), $\vec{\boldsymbol{\mathcal{E}}}_m$ denotes the set of edges corresponding to landmark measurements, and, for generic vectors $\boldsymbol{a}$ and matrix $\boldsymbol{W}$ of suitable dimensions, the notation $\|\boldsymbol{a}\|_{\boldsymbol{W}}^2 = \boldsymbol{a}^\mathsf{T}\boldsymbol{W}\boldsymbol{a}$ denotes the standard Mahalanobis squared norm. This seemingly innocuous change with respect to Problem 5 is quite consequential in practice. Indeed, if $\mathbf{W}_{ik}$ is isotropic (*e.g.,* $\mathbf{W}_{ik} = \mu_{ik}^{-1}\mathbf{I}_3$) we can manipulate the expression in (6.34) to the QCQP in Problem 5, whereas if $\mathbf{W}_{ik}$ is anisotropic the problem is *quartic* (*i.e.,* it involves degree 4 polynomials in the variables).[6]

Example: SLAM with Outliers. So far, we assumed all measurements to be affected by zero-mean (but possibly anisotropic) Gaussian noise. Unfortunately, as we discussed in Chapter 3, in real SLAM problem some measurements might be outliers. This is typically the case for loop closure or landmark measurements, where incorrect place recognition or data association might cause adding incorrect measurements to the SLAM back-end. As we discussed in Chapter 3, an effective approach to mitigate the impact of outliers is to use robust loss functions. For instance, in the presence of outliers, the landmark-based SLAM Problem 5 becomes:

[6] Intuitively, when $\mathbf{W}_{ik} = \mu_{ik}^{-1}\mathbf{I}_3$, we can simplify the quartic function $\|\boldsymbol{R}_i^\mathsf{T}(\boldsymbol{m}_k - \boldsymbol{t}_i) - \tilde{\boldsymbol{m}}_{ik}\|_{\mathbf{W}_{ik}}^2$ to a quadratic one: $\|\boldsymbol{R}_i^\mathsf{T}(\boldsymbol{m}_k - \boldsymbol{t}_i) - \tilde{\boldsymbol{m}}_{ik}\|_{\mathbf{W}_{ik}}^2 = \mu_{ik}\|\boldsymbol{R}_i^\mathsf{T}(\boldsymbol{m}_k - \boldsymbol{t}_i) - \tilde{\boldsymbol{m}}_{ik}\|^2 = \mu_{ik}\|\boldsymbol{m}_k - \boldsymbol{t}_i - \boldsymbol{R}_i\tilde{\boldsymbol{m}}_{ik}\|^2$, where in the last equality we used the rotation invariance of the ℓ_2 norm.

Problem 8 (Pose Graph Optimization with Outliers)

$$\begin{aligned}\min_{\substack{\boldsymbol{R}_i\in\mathrm{SO}(d)\\ \boldsymbol{t}_i\in\mathbb{R}^d}} \sum_{(i,j)\in\vec{\mathcal{E}}_o} & \kappa_{ij}\|\boldsymbol{R}_j-\boldsymbol{R}_i\tilde{\boldsymbol{R}}_{ij}\|_F^2+\tau_{ij}\left\|\boldsymbol{t}_j-\boldsymbol{t}_i-\boldsymbol{R}_i\tilde{\boldsymbol{t}}_{ij}\right\|_2^2\\ &+\sum_{(i,k)\in\vec{\boldsymbol{\mathcal{E}}}_m}\rho\left(\sqrt{\mu_{ik}}\,\|\boldsymbol{m}_k-\boldsymbol{t}_i-\boldsymbol{R}_i\tilde{\boldsymbol{m}}_{ik}\|\right).\end{aligned} \tag{6.35}$$

In (6.35), $\rho(\cdot)$ is a robust loss function (*cf.* Chapter 3) which is designed to reduce the influence of potential outliers in the landmark measurements. From Chapter 3, we know that we can use the Black-Rangarajan duality to reformulate (6.35) as a least-squares problem with auxiliary variables w_{ik}, one for each robust loss term. For instance, when choosing $\rho(\cdot)$ to be the truncated quadratic loss, then (6.35) can be rewritten as

$$\begin{aligned}\min_{\substack{\boldsymbol{R}_i\in\mathrm{SO}(d)\\ \boldsymbol{t}_i\in\mathbb{R}^d\\ w_{ik}\in[0,1]}} \sum_{(i,j)\in\vec{\mathcal{E}}_o} & \kappa_{ij}\|\boldsymbol{R}_j-\boldsymbol{R}_i\tilde{\boldsymbol{R}}_{ij}\|_F^2+\tau_{ij}\left\|\boldsymbol{t}_j-\boldsymbol{t}_i-\boldsymbol{R}_i\tilde{\boldsymbol{t}}_{ij}\right\|_2^2\\ &+\sum_{(i,k)\in\vec{\boldsymbol{\mathcal{E}}}_m} w_{ik}\mu_{ik}\,\|\boldsymbol{m}_k-\boldsymbol{t}_i-\boldsymbol{R}_i\tilde{\boldsymbol{m}}_{ik}\|^2+(1-w_{ik})\beta^2.\end{aligned} \tag{6.36}$$

where β is the maximum inlier error, as specified by the truncated quadratic loss. We note that (6.36) now includes auxiliary variables w_{ik}, which indicate whether a measurement is classified as an inlier or outlier. Interestingly, the objective function in (6.36) includes polynomials of degree up to 3, while the constraints are still at most quadratic functions. The work [1217] shows that the same conclusion holds for several choices of robust losses and for other variations of the problem, including pose graph optimization and multiple rotation averaging. Below, we discussion how to obtain semidefinite relaxations for optimization problems involving polynomials.

Polynomial Optimization Problems and Moment Relaxation. The previous examples show how a broad range of SLAM problems can be reformulated as optimization problems involving polynomials. More formally, they can be written as Polynomial Optimization Problems:

$$\begin{aligned}\min_{\boldsymbol{x}} \quad & p(\boldsymbol{x})\\ \text{subject to } & h_i(\boldsymbol{x})=0,\;\; i=1,\ldots,n_h\\ & g_i(\boldsymbol{x})\le 0,\;\; i=1,\ldots,n_g,\end{aligned} \tag{POP}$$

where the functions p, h_i, g_i are real polynomials in the variable $\boldsymbol{x}$, and n_h and n_g are the number of equality and inequality constraints.

Rewriting the SLAM problem as (POP) does not immediately imply any computational advantage: POPs are a very general class of optimization problems (which also includes QCQPs) and are generally intractable to solve. Our interest towards (POP) stems from the fact that there exists a standard procedure, known as the *Moment (or Lasserre's) relaxation*, to obtain a semidefinite relaxation of (POP). Even more interestingly, the procedure provides tools to obtain a *hierarchy of* relaxations and also guarantees that —under mild assumptions— certain relaxations in this hierarchy are exact. While we refer the reader to [153] and the seminal works [634, 633] for a more extensive introduction to the Moment relaxation, below we provide a simple example to convey the underlying ideas.

To illustrate how to obtain a relaxation of a POP, consider the following problem:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & p(\boldsymbol{x}) \\ \text{subject to } & h_i(\boldsymbol{x}) = 0, \;\; i = 1, \ldots, n_h \end{aligned} \tag{6.37}$$

where, for simplicity, we assume $\boldsymbol{x} = [x_1; x_2]$, and that $p(\boldsymbol{x})$ and $h(\boldsymbol{x})$ are polynomials of degree at most 4; in this simplified example we only have equality constraints. Then, to derive the moment relaxation of (POP), we define the vector of monomials of degree up to $r = 2$, where r is called the *order* of the relaxation:

$$[\boldsymbol{x}]_2 = [1; x_1; x_2; x_1^2; x_1 x_2; x_2^2] \tag{6.38}$$

Then, we can form the moment matrix as the following outer product:

$$\boldsymbol{X}_4 \triangleq [\boldsymbol{x}]_2 [\boldsymbol{x}]_2^\mathsf{T} = \begin{bmatrix} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \\ x_1 & x_1^2 & x_1x_2 & x_1^3 & x_1^2x_2 & x_1x_2^2 \\ x_2 & x_1x_2 & x_2^2 & x_1^2x_2 & x_1x_2^2 & x_2^3 \\ x_1^2 & x_1^3 & x_1^2x_2 & x_1^4 & x_1^3x_2 & x_1^2x_2^2 \\ x_1x_2 & x_1^2x_2 & x_1x_2^2 & x_1^3x_2 & x_1^2x_2^2 & x_1x_2^3 \\ x_2^2 & x_1x_2^2 & x_2^3 & x_1^2x_2^2 & x_1x_2^3 & x_2^4 \end{bmatrix}. \tag{6.39}$$

Now the key observation is that we can write any polynomial of degree up to 4 as a linear combination of the entries of $\boldsymbol{X}_4$. Therefore, we can rewrite (6.37) as:

$$\begin{aligned} \min_{\boldsymbol{X}, \boldsymbol{x}} \quad & \operatorname{tr}(\boldsymbol{C}\boldsymbol{X}) \\ \text{subject to } & \operatorname{tr}(\boldsymbol{H}_i \boldsymbol{X}) = 0, \;\; i = 1, \ldots, n_h \\ & \boldsymbol{X} = [\boldsymbol{x}]_2 [\boldsymbol{x}]_2^\mathsf{T} \end{aligned} \tag{6.40}$$

In full analogy with Shor's relaxation, we can now replace the constraint $\boldsymbol{X} = [\boldsymbol{x}]_2[\boldsymbol{x}]_2^\mathsf{T}$ with $\boldsymbol{X} \succeq 0$ and $\operatorname{rank}(\boldsymbol{X}) = 1$ and then relax the rank constraint to obtain a semidefinite relaxation. Importantly, the moment relaxation also adds *redundant constraints*[7] to improve its quality. These constraints capture the fact that the moment matrix contains repeated entries (*e.g.,* the term $x_1 x_2$ appears 3 times in (6.40)) as well as the fact that if $h_i(\boldsymbol{x}) = 0$, then also $x_1 \cdot h_i(\boldsymbol{x}) = 0$ and $x_2 \cdot h_i(\boldsymbol{x}) = 0$ must hold. The moment relaxation provides a systematic way to identify all these redundant constraints, and also extends to the case of inequality constraints. Moreover, while above we derived an order $r = 2$ relaxation, we can repeat the procedure for any integer $r \geq 2$, obtaining larger but better SDP

[7] These are constraints that are inconsequential for problem (6.40), but improve the quality of the relaxation.

relaxations (hence a *hierarchy* of relaxations). Indeed, the seminal work [812] establishes that the moment relaxation will produce exact relaxations (*i.e.,* recovers a certifiably optimal solution to the original POP) at a finite order r, under mild assumptions. Interestingly, related work [1217, 473] has observed that moment relaxations of problems with anisotropic noise and outliers are already exact at a low relaxation order. Moreover, they remain tight even when using a subset of the variables in the monomial basis (6.38) (*i.e.,* a *sparse* monomial basis), which further reduces the size of the resulting SDP.

The 'Catch': Solving the SDP Moment Relaxation. One important difference between the relaxations we have seen in the previous sections and the ones discussed in this section is that the latter involve a large number of redundant constraints in the SDP relaxation. While this difference might seem minor, the consequences for the SDP solver are profound. In particular, SDPs with redundant constraints (as the ones typically obtained from the moment relaxation) are *degenerate* [34], which makes the application of the Riemannian Staircase solver problematic for two reasons. First, constraint qualification conditions typically fail for these problems, and these conditions are required to ensure convergence of the Riemannian Staircase [847]. Second, degenerate SDPs have an infinite number of dual solutions, which creates computational obstacles in the implementation of the Riemannian Staircase. In other words, the Riemannian Staircase is no longer a viable solver for the degenerate SDPs typically produced by the moment relaxation. While the recent literature includes specialized solvers for moment relaxations of POPs [1216], these solvers are still relatively slow compared to local solvers.

6.2 How Accurate is the Optimal Solution of a SLAM Problem?

In the previous section we discussed how to obtain certifiably optimal solutions to certain SLAM problems. However, another fundamental question remains: *how accurate is the optimal solution compared to the ground truth?* Understanding this fundamental limit and identifying the key factors that influence estimation accuracy provides critical guidance for both system designers and end users. For example, at design time such insights can guide the choice of sensors the robot is equipped with; on the other hand, at deployment time, these insights can be used to guide the motion of the robot (and the corresponding acquisition of measurements) to ensure robust SLAM performance. Before offering a concrete answer, we first need to formalize the question. Modern SLAM pipelines are complex, with many interacting subsystems affecting overall performance. Consequently, we approach this question from an estimation-theoretic perspective, focusing on the SLAM back-end.

Some Terminology and Facts. The goal of the SLAM back-end is to estimate unknown quantities such as robot poses and landmarks' positions from noisy measurements. These measurements are random variables because they are corrupted by random sensor noise. An *estimator* is a function that maps collected noisy measurements to an estimate of the unknown parameters. Since the estimator depends on random measurements, it is itself a random variable. The Mean Squared Error (MSE) is a commonly used metric for evaluating an estimator's performance. As the name suggests, the MSE represents the squared estimation error averaged over all possible measurements. In the univariate case, the MSE can be expressed as the sum of the estimator's squared bias and its variance. The bias refers to the difference between the ground truth and the estimator's output, averaged over all possible measurements. The variance captures the variability of the estimator's output around its mean. The relationship between bias and variance, and the definition of the MSE extends naturally to the multivariate setting, where the variance is replaced by a covariance matrix. The MSE is still the sum of the squared norm of the bias and the trace of the covariance matrix. Clearly, for *unbiased* estimators (*i.e.,* estimators with zero bias), the only quantity influencing the MSE is the covariance matrix. In the following, we show how to lower bound the covariance of the SLAM estimate, thus getting a fundamental limit on the accuracy achievable by a SLAM system.

6.2.1 Cramér-Rao Lower Bound and the Fisher Information Matrix

The *Cramér-Rao Lower Bound* (CRLB) provides a theoretical limit on the best estimator covariance achievable by any unbiased estimator. Formally,

$$\mathrm{Cov}(\hat{\mathbf{x}}) \succeq \mathcal{I}(\mathbf{x}_{\mathrm{true}})^{-1}, \tag{6.41}$$

where $\hat{\mathbf{x}}$ is any unbiased estimator of $\mathbf{x}_{\mathrm{true}} \in \mathbb{R}^m$, and $\boldsymbol{A} \succeq \boldsymbol{B}$ indicates that $\boldsymbol{A} - \boldsymbol{B}$ is positive semidefinite. The matrix $\mathcal{I}(\mathbf{x}_{\mathrm{true}})$ appearing on the right-hand side of (6.41) is the *Fisher information matrix* (FIM), which is defined as[8]

$$[\mathcal{I}(\mathbf{x}_{\mathrm{true}})]_{i,j} \triangleq \mathbb{E}_{\mathbf{z}} \left[\frac{\partial}{\partial x_i} \log\, p(\mathbf{z};\mathbf{x})\ \frac{\partial}{\partial x_j} \log\, p(\mathbf{z};\mathbf{x}) \right]. \tag{6.42}$$

Here the expectation is taken over the possible realizations of measurements $\mathbf{z}$ drawn from the probability density function $p(\mathbf{z};\mathbf{x}_{\mathrm{true}})$, and the partial derivatives of the log-likelihood function are *evaluated at the true parameter value* $\mathbf{x}_{\mathrm{true}}$. Under certain regularity conditions, the FIM can also be expressed as the expected value of the Hessian of the log-likelihood:

$$[\mathcal{I}(\mathbf{x}_{\mathrm{true}})]_{i,j} = \mathbb{E}_{\mathbf{z}} \left[\frac{\partial^2}{\partial x_i \partial x_j} \log\, p(\mathbf{z};\mathbf{x}) \right]. \tag{6.43}$$

[8] The classical CRLB assumes that the variables live in a Euclidean space. In SLAM, we often need to estimate poses and rotation matrices. The CRLB has been extended to Riemannian submanifolds and matrix Lie group parameters [112, 101]. In such cases, CRLB has an additional (additive) term that depends on the curvature of the parameter space. The curvature term is often negligible when the signal-to-noise ratio is large.

This connection to the Hessian provides an intuitive understanding of the CRLB: the CRLB establishes a lower bound on the covariance of any unbiased estimator, expressed in terms of the local sensitivity (curvature) of the (expected) log-likelihood function with respect to the parameters around the true parameter value $\mathbf{x}_{\text{true}}$. If the log-likelihood is relatively flat around the true parameter value across different realizations of measurements (*i.e.,* in expectation), any unbiased estimator will struggle to accurately localize the true parameter based on the observed data. In such cases, unbiased estimators will exhibit higher variance, leading to a higher MSE. In summary, the FIM captures the amount of information one can learn about the true parameter's value from measurements using any unbiased estimator.

Under certain regularity conditions, it has been shown that the maximum likelihood estimator $\hat{\mathbf{x}}_{\text{mle}}$ asymptotically (*i.e.,* when the number of measurements tends to ∞) converges (in distribution) to $\mathcal{N}\left(\mathbf{x}_{\text{true}}, \mathcal{I}(\mathbf{x}_{\text{true}})^{-1}\right)$. Therefore, the maximum likelihood estimator is *asymptotically* unbiased and achieves the CRLB (*i.e.,* minimum variance among all unbiased estimators). Since the true value of parameters $\mathbf{x}_{\text{true}}$ is unknown, we often approximate the FIM by $\mathcal{I}(\hat{\mathbf{x}}_{\text{mle}})$. Furthermore, it is common to approximate the covariance of the maximum likelihood estimator $\hat{\mathbf{x}}_{\text{mle}}$ with $\mathcal{I}(\hat{\mathbf{x}}_{\text{mle}})^{-1}$, the so-called *Laplace approximation.*

As an example, consider the common scenario where measurements are generated by corrupting a smooth (potentially nonlinear) function with additive Gaussian noise. In this case, the measurement model can be written as

$$\mathbf{z} = \mathbf{h}(\mathbf{x}_{\text{true}}) + \boldsymbol{\epsilon}, \tag{6.44}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is the noise. The likelihood function *evaluated at* $\mathbf{x}$ is thus given by $p(\mathbf{z}; \mathbf{x}) = \mathcal{N}(\mathbf{h}(\mathbf{x}), \boldsymbol{\Sigma})$. Plugging this likelihood in (6.42) yields the FIM (evaluated at $\mathbf{x}$):

$$\mathcal{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})^{\top} \boldsymbol{\Sigma}^{-1} \mathbf{J}(\mathbf{x}), \tag{6.45}$$

in which $\mathbf{J}(\mathbf{x})$ denotes the Jacobian of the measurement model $\mathbf{h}$ evaluated at $\mathbf{x}$.[9]

A Scalar Measure of Uncertainty: Optimal Experimental Design Criteria. In many applications (*e.g.,* system design, active SLAM), one needs to map the FIM, a PSD matrix, to a real number that captures a meaningful and 'optimizable' aspect of the estimation error uncertainty. Standard choices have been investigated in the field of Optimal Experimental Design. These include the determinant of the FIM (D-optimality), the trace of its inverse (A-optimality), and its smallest eigenvalue (E-optimality). These criteria are *spectral* functions of the FIM (*i.e.,* can be computed from the eigenvalues of the FIM). Each criterion reflects a different aspect of estimation error: the D-optimality criterion quantifies the uncertainty hyper-volume, the A-optimality criterion measures the average variance, and the E-optimality criterion represents the worst-case estimation variance.

[9] The matrix (6.45) is often called the *information matrix* in the SLAM literature.

6.2.2 Fisher Information Matrix and Graph Laplacian

In this section, we study how the graphical structure of SLAM problems affects the FIM. Inspecting the FIM (6.45) reveals the intuitive fact that the measurement noise covariance matrix $\mathbf{\Sigma}$ affects the FIM: unsurprisingly, higher measurement noise covariance increases the lower bound on the smallest achievable MSE among all unbiased estimators in the CRLB. Additionally, the Jacobians of the measurements appear in (6.45), but it is difficult to intuitively understand from this equation *how* the measurements' Jacobian affects the CRLB.[10] In the following, we gain insights on the structure of the Jacobians and the resulting FIM by relating them to properties of the graph underlying the SLAM problem.

All variants of SLAM naturally admit a graphical representation as we have seen in Chapter 1 and earlier in this chapter. The graph essentially encodes "who is observing what" and provides a concise overview of the SLAM problem. For instance, in pose graph optimization and landmark-based SLAM, each variable (e.g., robot pose or landmark position) is represented by a vertex, while pairwise measurements (pose-pose or pose-landmark) correspond to edges between the respective vertices.

Next, we study how properties of the graph underlying the SLAM problem impact the accuracy of the resulting estimate. In particular, the degree of *connectivity* within the graph reflects the *redundancy* in measurements. Intuitively, a "better" connected SLAM graph is expected to be more robust to noise, yielding accurate estimates even under higher noise levels due to redundant measurements. While it is straightforward to show from (6.45) that introducing additional measurements (*i.e.,* edges) always reduces the lower bound in the CRLB (in the Loewner order),[11] the impact of different measurements varies depending on which variables are involved in the additional measurements (*i.e.,* the resulting graph connectivity). This is particularly evident in the context of loop closure in SLAM: closing a "larger" loop has a more significant effect on improving the accuracy of the SLAM solution. This intuition has been formalized in a series of works, establishing connections between the graphical structure of SLAM and desirable properties in both estimation and optimization. Below we provide a brief overview of these findings.

Connections between FIM and Graph Laplacian for a Simplified PGO Problem. The FIM in landmark-based SLAM and pose graph optimization is closely linked connected to graph Laplacian [569, 568, 884, 197]. This relationship is intuitive given that the measurements in these frameworks consist of pairwise relative observations between vertices. To illustrate the concept, we derive the FIM for the simpler problem of estimating robot positions in a 3D pose graph optimization

[10] Note that even without understanding the structure of the Jacobians, we can still compute the FIM (approximated at the current estimate) and use it to assess the estimation accuracy of a given SLAM problem. However, a deeper understanding of the FIM will allow us to *predict* how modifications of the SLAM problem (*e.g.,* driven by active data collection by the robot) will impact the future uncertainty in the SLAM estimates, thus informing active perception and design tasks.

[11] Each new measurements adds a positive semi-definite matrix to the FIM. This is known as the "information never hurts" principle.

problem when the robot orientations are known. Let $\mathbf{z}_k$ be denote the kth relative measurement in which pose i_k observes pose j_k in its local frame:

$$\mathbf{z}_k = \mathbf{R}_{i_k}^\top(\mathbf{t}_{j_k} - \mathbf{t}_{i_k}) + \boldsymbol{\epsilon}_k, \tag{6.46}$$

where $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, w_k^{-1}\mathbf{I}_3)$ in which $\mathbf{I}_3$ is the 3×3 identity matrix. Let $\mathbf{z}$, $\mathbf{t}$, and $\boldsymbol{\epsilon}$ be the stacked vectors of measurements, positions, and noise variables. Additionally, let $\mathbf{R}$ be the block-diagonal matrix of rotation matrices such that the kth block is the rotation matrix involved in the kth measurement:

$$\mathbf{R} \triangleq \mathrm{BlockDiag}(\mathbf{R}_{i_1}, \mathbf{R}_{i_2}, \ldots, \mathbf{R}_{i_m}). \tag{6.47}$$

The stacked measurement model can be expressed as:

$$\mathbf{z} = \mathbf{R}^\top(\mathbf{A} \otimes \mathbf{I}_3)^\top \mathbf{t} + \boldsymbol{\epsilon}, \tag{6.48}$$

where $\mathbf{A}$ denotes the reduced incidence matrix of the pose graph,[12], and $\otimes$ denotes the Kronecker product. Therefore, the Jacobian matrix of the stacked measurement model is given by

$$\mathbf{J} = \mathbf{R}^\top(\mathbf{A} \otimes \mathbf{I}_3)^\top. \tag{6.49}$$

The information matrix of the stacked noise vector is given by

$$\boldsymbol{\Sigma}^{-1} = \mathrm{BlockDiag}(w_1\mathbf{I}_3, w_2\mathbf{I}_3, \ldots, w_m\mathbf{I}_3). \tag{6.50}$$

Let $\mathbf{W}$ denote the diagonal matrix of edge weights:

$$\mathbf{W} \triangleq \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_m \end{bmatrix}. \tag{6.51}$$

Using (6.45), the FIM can be computed as follows:

$$\begin{aligned} \mathcal{I} &= \mathbf{J}^\top \boldsymbol{\Sigma}^{-1} \mathbf{J} && (6.52\text{a}) \\ &= (\mathbf{A} \otimes \mathbf{I}_3)\mathbf{R}\boldsymbol{\Sigma}^{-1}\mathbf{R}^\top(\mathbf{A} \otimes \mathbf{I}_3)^\top && (6.52\text{b}) \\ &= (\mathbf{A} \otimes \mathbf{I}_3)(\mathbf{W} \otimes \mathbf{I}_3)(\mathbf{A} \otimes \mathbf{I}_3)^\top && (6.52\text{c}) \\ &= (\mathbf{A}\mathbf{W}\mathbf{A}^\top) \otimes \mathbf{I}_3 && (6.52\text{d}) \\ &= \mathbf{L}_w \otimes \mathbf{I}_3, && (6.52\text{e}) \end{aligned}$$

where we observed that the block diagonal entries of $\boldsymbol{\Sigma}^{-1}$ are invariant to rotation, hence $\mathbf{R}\boldsymbol{\Sigma}^{-1}\mathbf{R}^\top = \boldsymbol{\Sigma}^{-1}$, and we used the fact that $\boldsymbol{\Sigma}^{-1} = \mathbf{W} \otimes \mathbf{I}_3$. In (6.52a), $\mathbf{L}_w = \mathbf{A}\mathbf{W}\mathbf{A}^\top$ is the reduced weighted Laplacian matrix of the graph where edge

[12] The *reduced* incidence matrix is obtained by removing the row(s) corresponding to anchored poses from the incidence matrix (e.g., anchoring the first position to the origin of the world frame) [154]. This is done to remove gauge symmetry due to the symmetric nature of SLAM with relative measurements.

weights are given by $w_1, w_2, \ldots, w_m$. This clearly shows that in this case (simplified pose graph optimization where robot orientations are assumed known and noise is assumed isotropic), the FIM is fully characterized by the reduced weighted Laplacian of the underlying graph.

FIM for Other SLAM Problems. The result above can be generalized to pose graph optimization and landmark-based SLAM problems [569, 884, 197]. In these problems, the FIM involves both the reduced weighted Laplacian matrix of graph and additional terms that depend on the robot trajectory. In particular, the FIM for 3D pose-graph optimization can be written as [884, Eq. 29]:

$$\mathcal{I}(\mathbf{x}) = \sum_{k=1}^{m} \mathbf{L}_k \otimes \left(\mathrm{Ad}(\mathbf{T}_{i_k}^{-1})^\top \mathbf{\Sigma}_k^{-1} \mathrm{Ad}(\mathbf{T}_{i_k}^{-1})\right), \tag{6.53}$$

where $\mathbf{T}_{i_k}$ is the pose of the robot making the kth measurement, $\mathbf{\Sigma}_k$ is the covariance matrix of the noise corrupting the kth measurement, and $\mathbf{L}_k$ is the reduced *elementary* Laplacian matrix of the kth edge defined as $\mathbf{L}_k \triangleq \mathbf{a}_k\mathbf{a}_k^\top$ where $\mathbf{a}_k$ is the kth column of the reduced incidence matrix of the graph. Empirical results and theoretical analyses demonstrate that, under certain conditions, the (approximate) optimal design criteria derived from the reduced Laplacian closely match those obtained from the FIM [569, 884, 197].

Practical Considerations. The connection between the FIM and the graph Laplacian allows approximating optimal design criteria — which are spectral functions of the FIM — using the spectrum of the reduced weighted Laplacian. For instance, the D-optimality criterion can be approximated by a function of the determinant of the weighted reduced Laplacian. According to Kirchhoff's matrix-tree theorem, this determinant equals the weighted number of spanning trees of the graph [569]. Since the weighted number of spanning trees serves as a measure of connectivity in edge-weighted graphs, this result formalizes the intuition that a graph's connectivity directly influences estimation accuracy. Similarly, the E-optimality criterion is related to the graph's algebraic connectivity [283, 569, 884]. Overall, measuring uncertainty using the graph Laplacian instead of the FIM offers two main advantages: it leads to more computationally efficient techniques (*e.g.,* the Laplacian has a dimension six times smaller than that of the FIM in 3D pose-graph optimization), and it eliminates the need to solve the SLAM problem or acquire actual measurements, as the calculations are based solely on the graph structure. These graphical approximations of design criteria have been successfully applied in active SLAM [197, 590, 884] — where a robot plans its trajectory to maximize the anticipated SLAM accuracy — and in measurement selection and pruning [283, 569, 1098], where the goal is to select and retain only the *most informative* measurements in lifelong SLAM problems.

6.3 New Trends

Certifiable Algorithms. The first certifiable algorithm for 2D SLAM traces back to [155], and builds on earlier work done in computer vision and related areas, including [346, 1020]. Extensions to 3D SLAM and variations quickly followed [157, 1112, 943, 945, 120, 158], with SE-Sync providing the first blueprint for building *fast* certifiable algorithms [943]. Since then, certifiable algorithms have been designed for a variety of problems related to SLAM,[13] including rotation averaging [128, 977], pose graph optimization [943, 155], landmark-based SLAM [473], multi-robot SLAM [1097], range-aided SLAM [846], 3D registration [121, 1219], multi-set registration [509], 2-view geometry [122, 1280, 375, 1101, 546], perspective-n-point problems [1058], calibration [390], single-frame pose and shape estimation [1012], multi-frame pose and shape estimation [999], and structure from motion with learned depth [1251]. Recent work has also extended certifiable algorithms to cope with anisotropic noise [474] and outliers [1217], a topic that we discuss in more detail below.

While there has been incredible progress in this area over the last decade, three exciting open questions remain. First of all, there are still SLAM problems that cannot be attacked with certifiable algorithms. For instance, when doing visual SLAM (Chapter 7), the perspective projection arising in the objective function is a rational function rather than a polynomial; while the problem can still reformulated as a POP by introducing extra variables, it is typically impractical to add one variable for each keypoint measurement in an image. Similarly, the modeling of IMU measurements (that we discuss in detail in Chapter 11) has not been conducive to the design of certifiable algorithms. Second, except for certain problems, *e.g.,* [943, 546, 1251, 846], where the Riemannian Staircase method can be applied as a fast solver, SDPs might need to be solved by interior-point methods or other ad hoc solvers. While interior-point methods are very effective and fast in solving low-dimensional optimization problems arising in the SLAM front-end (*e.g.,* [121, 1219, 122, 1280, 375, 1101, 546]), they become impractically slow when applied to large-scale problems in the SLAM back-end (*e.g.,* [473]), and ad hoc solvers — while being more scalable — are still relatively slow compared to local solvers [1216]. Recent work to address this problem has not only investigated faster SDP solvers, but has also focused on how to reduce the size of the SDP (*e.g.,* by sparsifying the monomial basis underlying the moment relaxation) [1216], or how to reduce the number of constraints, in the attempt to make the SDP non-degenerate or faster to solve [298]. Finally, while current works compute per-instance certificates of optimality (*i.e.,* they compute an estimate and possibly provide a certificate of optimality for it), the literature lacks a fundamental understanding of *when* the

[13] Some of these optimization problems are solved as part of the SLAM front-end, *e.g.,* to get relative poses from sensor measurements.

relaxation is expected to be exact, with only few papers providing *conditions for exactness* in specific problems [1214, 866, 943, 318, 153].

Problems with Outliers. Real-world SLAM problems are typically plagued with outliers, a problem that we discussed at length in Chapter 3. While several works wrap outlier-free certifiable algorithms in a Graduated Non-Convexity outer loop [1219, 1012, 999] to gain empirical robustness to outliers, recent work directly attempts the develop certifiable algorithms for robust estimation problems involving outliers. Efforts in this direction tackled pose graph optimization [626, 159], rotation estimation [1214], 3D registration [1219], multiple rotation averaging, absolute pose estimation, and pose and shape estimation [1217]. A good summary of these results is provided in [1217], while connections with parallel work in robust statistics are discussed in [153]. These algorithms rely on the moment relaxation described earlier in this chapter and the resulting SDPs are still relatively slow to solve. An alternative approach has been to obtain solutions via local solvers, and use the same insights behind the moment relaxation to derive methods to only *check* optimality [1215].

Many interesting open questions also remain in this case.[14] In the presence of outliers, we still see the same challenges and opportunities discussed in the previous paragraph, including how to extend certifiable algorithms to other problems with outliers, how to design faster solvers, and how to derive conditions for exactness (*e.g.,* as a function of the amount of noise and outliers). At the same time, there are additional challenges related to the presence of outliers. First of all, the SLAM approaches discussed above assume an outlier-free odometry backbone: in other words, only the loop-closure measurements are wrapped in a robust loss function. The assumption of having a reliable odometry source is acceptable in many SLAM problems, but can be restrictive in certain cases. For instance, in visual SLAM problems, the odometry can become unreliable if the feature tracking fails, while in multi-robot SLAM problems, there is no odometry backbone connecting the different robots and all the inter-robot measurements can be outliers. While the formulation of current certifiable algorithms would extend to the case where also the odometry is wrapped in a robust loss, the resulting relaxations are known to be loose, *e.g.,* [846], and it is unclear how to improve them. Second, even if we compute an optimal estimate when solving a robust estimation problem, if the majority of the measurements are outliers, the estimate can still be grossly incorrect. An interesting and relatively unexplored area is to design certifiable algorithms that can recover *multiple* hypotheses while guaranteeing that at least one hypothesis is correct, a setting called *list decodable regression* in statistics [153].

Uncertainty Quantification and Downstream Applications. While Section 6.2 provided computational tools to bound the uncertainty in our SLAM estimate and potentially predict its evolution as a function of the structure of the

[14] The reader also also referred to [153] to a more in-depth discussion of some of these challenges.

underlying graph, it still leaves many questions open. First of all, the computation of the covariance of the SLAM estimate relies upon the knowledge of the measurement covariances: if the measurement covariance matrices are inaccurate, the resulting uncertainty bounds become unreliable. Recent work uses learning to estimate the measurement covariances or even the entire measurement model [1026, 896, 1239] (see also Chapter 4 for a broader discussion on differentiable optimization). Second, the traditional approach to compute a covariance for the estimate does not account for the potential presence of outliers. In the presence of outliers, the covariance estimates can be incorrect and the distribution of potential estimates (*e.g.,* of the robot trajectory) can become highly multimodal, hence limiting the use of the covariance estimates described earlier in this chapter. Finally, there is a growing literature using uncertainty quantification to guide active perception and active SLAM (see [883] for a survey) and to determine how to subselect measurements during life-long SLAM or in the presence of resource constraints [283, 569, 1098, 156].