

Optimal Initialization Strategies for Range-Only Trajectory Estimation

Abhishek Goudar¹, Frederike Dübngn¹, Timothy D. Barfoot¹, and Angela P. Schoellig^{1,2}

Abstract—Range-only (RO) pose estimation involves determining a robot’s pose over time by measuring the distance between multiple devices on the robot, known as tags, and devices installed in the environment, known as anchors. The non-convex nature of the range measurement model results in a cost function with possible local minima. In the absence of a good initial guess, commonly used iterative solvers can get stuck in these local minima resulting in poor trajectory estimation accuracy. In this work, we propose convex relaxations to the original non-convex problem based on semidefinite programs (SDPs). Specifically, we formulate computationally tractable SDP relaxations to obtain accurate initial pose and trajectory estimates for RO trajectory estimation under static and dynamic (i.e., constant-velocity motion) conditions. Through simulation and hardware experiments, we demonstrate that our proposed approaches estimate the initial pose and initial trajectories accurately compared to iterative local solvers. Additionally, the proposed relaxations recover global minima under moderate range measurement noise levels.

Index Terms—Localization, optimization and optimal control, range-only localization, semidefinite relaxation.

I. INTRODUCTION

RANGE-only (RO) localization involves determining the position of a mobile system, such as a robot, by measuring the distance between a range sensor on the robot, referred to as a *tag*, and known landmarks in the environment, referred to as *anchors*. Common examples of RO localization include radio frequency (RF)-based positioning such as the Global Positioning System (GPS) [1] for outdoor environments, WiFi or ultrawideband (UWB)-based positioning for indoor environments [2], and acoustic positioning for underwater environments [3]. Since a single range measurement is not sufficient to estimate a robot’s position and orientation, range sensors are typically combined with other sensing modalities such as wheel odometry [4], inertial measurement units (IMUs) [5], and cameras [6] to estimate the full pose. However, a common limitation of these sensor-fusion schemes is that sufficient motion is needed before the pose becomes observable [7], [8]. An alternative is to use multiple tags to estimate the full pose [9], referred to as RO pose estimation. The advantage of such an approach is that motion is not necessary for pose estimation,

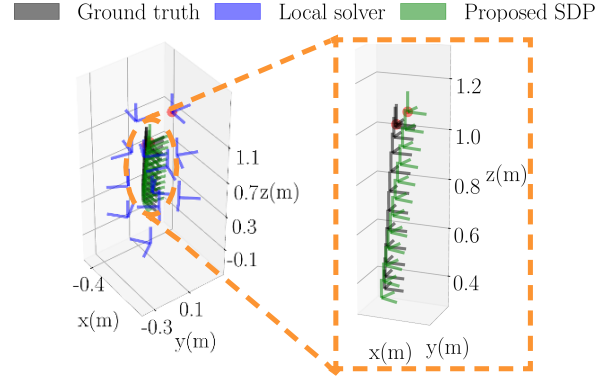


Fig. 1: Range-only trajectory estimation results from a hardware experiment. A quadrotor in motion, equipped with 2 range sensors, measures its distance to multiple anchors to estimate its 3D position, velocity, and yaw angle over a short time horizon (pitch and roll come from an IMU). We refer to this as 2.5D dynamic initialization. The trajectory is parameterized by a sequence of poses, represented here by orthogonal axes. In the absence of a good initialization, the iterative local solver gets stuck in a local minimum resulting in poor accuracy compared to the ground-truth trajectory. Our proposed approach leverages a semidefinite relaxation of the original problem to recover accurate trajectories as shown in the magnified image on the right. For each method, the robot pose at $t = 0$ is indicated by a red dot at the origin of the orthogonal axes.

but may still be beneficial. We use the term RO trajectory estimation to refer to both the estimation of the robot pose (*i*) at a single time step and (*ii*) across multiple time steps while in motion, using only range measurements from multiple tags.

From a computational perspective, a common approach to RO localization is to formulate it as a maximum a posteriori (MAP) estimation problem, which results in the optimization of a particular objective function. In RO trajectory estimation, the non-convex nature of the range measurement model, along with an additive Gaussian measurement noise assumption, results in a non-convex nonlinear least-squares objective function that is typically optimized using local solvers such as the Gauss-Newton algorithm. A well-known limitation of such local solvers is the need for a good initialization point [10], [11], without which the local solver can return suboptimal solutions, as shown in Figure 1.

The last decade has seen the development of algorithms that leverage Lagrangian duality theory [12] to obtain *convex* relaxations for non-convex problems found in many robotics applications. These convex relaxations typically take the form of semidefinite programs (SDPs) and can be used to obtain or certify globally optimal solutions. A tutorial on Lagrangian duality theory with a focus on robotics is provided in [13].

In this work, we propose strategies to obtain accurate state estimates (including pose and velocities) for RO trajectory estimation based on semidefinite programming. Given the poor scalability of SDP solvers, we focus on computationally tractable SDP relaxations to obtain optimal initial poses and trajectories over a short time horizon. These initial estimates

Manuscript received: September 15, 2023; Revised December 13, 2023; Accepted December 22, 2023.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers’ comments.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and in part by the Canada CIFAR AI Chairs Program.

¹ The authors are with the University of Toronto Institute for Aerospace Studies, Canada. They are also associated with the University of Toronto Robotics Institute and the Vector Institute for Artificial Intelligence.

² The authors are with the Technical University of Munich, Germany. They are also associated with the Munich Institute of Robotics and Machine Intelligence (MIRMI). E-mail: {abhishek.goudar@robotics.utias.utoronto.ca, {frederike.dubngn, tim.barfoot}@utoronto.ca, angela.schoellig@tum.de.

Digital Object Identifier (DOI): see top of this page.

can be used to bootstrap local solvers that, given a good initial starting point, are generally accurate and more efficient over longer estimation horizons [14].

In contrast to previous work [15], which focuses on position only and provides an optimality certificate for solutions from a local solver, our focus is to formulate tractable SDPs for *pose* estimation that can be solved quickly for online applications. Additionally, unlike previous methods [3], [16], our method does not require additional odometry sensors; it can be used to generate initial poses without any motion, which we refer to as *static initialization*. We also propose an initialization method for the case where a robot undergoes constant-velocity motion over a short time horizon; this is a challenging scenario for other methods that involve the fusion of a single range sensor and an odometry sensor as such motions lack the diversity that is required for observability [7]. We refer to initialization under such conditions as *dynamic initialization*. To summarize, the following are the contributions of our work:

- We present initialization strategies for range-only pose estimation under static conditions and range-only trajectory estimation under dynamic conditions involving constant-velocity trajectories. Our proposed initialization approaches recover optimal initial pose and trajectory estimates under moderate sensor-noise regimes.
- We validate the proposed methods in simulation and in multiple real experiments involving a ground robot and an aerial robot (see Figure 7).

Qualitative experimental results of our proposed method, including its application as a mode of initialization for a fixed-lag smoother can be found in the accompanying video¹.

II. RELATED WORK

The use of range measurements for positioning is a well-studied problem due to its widespread application in popular technologies such as GPS [1]. Other commonly used technologies for RO localization include sonar, lidar, WiFi, and UWB [2]. As mentioned previously, range sensors are typically combined with other sensing modalities such as wheel odometry [4], IMU [5], [8], or visual odometry [6]. A limitation of such sensor-fusion methods is the need for diverse motion for full-state observability [7], [8]. More recently, approaches based on Gaussian-process regression [17] have been applied to continuous-time RO position [15] and pose [9] estimation. The benefit of these approaches is that no other sensing modalities are needed as the pose is observable (with multiple tags) under static conditions and with trajectories that lack diverse motion.

For maximum-likelihood estimation, the non-convex nature of the range measurement model could result in a cost function with local minima. To recover the global minimum, a weighted least-squares (WLS) approach to planar pose estimation using range sensors aided by odometry is presented in [10]. A RO approach to planar pose estimation that converges to the maximum-likelihood estimate with minimum variance is presented in [18]. A trilateration-based approach to RO relative pose estimation using multiple range sensors is shown in [19].

Over the last decade, Lagrangian duality theory has been shown to be a powerful tool to generate optimal solutions and to certify the optimality of candidate solutions. It has been applied to various problems in robotics such as pose-graph optimization [20], simultaneous localization and mapping (SLAM) [21], synchronization over the special Euclidean group [22], extrinsic calibration between two egomotion sensors [23], and outlier-robust estimation [24]. A common feature of these methods is that they reformulate the MAP estimation problem as a quadratically constrained quadratic program (QCQP), for which SDP relaxations exist.

The application of duality theory to generate a certificate of optimality for RO localization was recently described in [15]. Subsequently, an approach to certifiably correct range-aided SLAM with pose variables was demonstrated in [16]. Unlike previous works, which focus on certifying batch solutions [15] or requiring additional sensing modalities [3], [10], [16], we focus on computationally tractable SDP relaxations to obtain optimal initial pose and trajectories over a short time horizon using multiple tags on the robot.

Other works have studied the SDP relaxation of RO pose estimation [25]–[27]. An important limitation of these approaches is that the SDP relaxations are not typically tight; the solutions are not rank 1 and require a subsequent refinement procedure. In this work, we empirically show that the lack of tightness stems from insufficient *redundant* constraints. We leverage the approach of [28] to find the necessary redundant constraints to generate rank-1 solutions to RO trajectory estimation under static and dynamic conditions.

III. PROBLEM STATEMENT

The objective of our work is to estimate an accurate initial pose and trajectory, for static and dynamic conditions, respectively, using range measurements only. We assume that the robot is equipped with multiple tags (≥ 2 for 2D and ≥ 3 for 3D) and that the position of the tags is known with respect to the robot body frame, \mathcal{F}_b .

We make the following additional assumptions for dynamic initialization. We assume that the robot moves with a constant velocity for a short time duration t_v and that the range measurements arrive periodically every Δt_r seconds during this period. The value of t_v is chosen to ensure sufficient measurements given a particular range measurement rate. In all cases, we assume that a sufficient number of noncollinear anchors (≥ 4) are available and that their positions with respect to the world frame, \mathcal{F}_W , are known. We assume that the range measurements are corrupted with moderate noise levels commonly encountered in practice.

IV. METHODOLOGY

In this section, we develop SDP-based relaxations to the (non-convex) problems of RO pose and trajectory estimation. Solving these non-convex problems locally using iterative solvers can result in sub-optimal solutions when local solvers get stuck in local minima. In contrast, the proposed relaxations are convex and can be solved to global optimality in polynomial time with off-the-shelf solvers. We find empirically that the relaxations are also often approximately tight – meaning

¹http://tiny.cc/opt_ro_init_video

that a viable estimate to the original, non-convex optimization problem can be extracted from the SDP solution.

We introduce the notation that will be used in the rest of the paper. The variable $d \in \{2, 3\}$ is used to represent the pose dimension. Rotations are represented using elements of the special orthogonal group $SO(d) = \{\mathbf{R} \in \mathbb{R}^{d \times d}, \mathbf{R}^T \mathbf{R} = \mathbf{I}_d, \det(\mathbf{R}) = 1\}$, where \mathbf{I}_d is the identity matrix of dimension d , and $\det(\cdot)$ is the determinant operation. We represent the robot pose $\mathbf{T}(t)$ at time t with elements of the special Euclidean group:

$$\mathbf{T}(t) = \begin{bmatrix} \mathbf{R}(t) & \mathbf{p}(t) \\ \mathbf{0} & 1 \end{bmatrix} \in SE(d), \quad (1)$$

where the pose is parameterized by its position $\mathbf{p}(t) \in \mathbb{R}^d$ and rotation $\mathbf{R}(t) \in SO(d)$. Elements of the corresponding Lie algebra are denoted by $\varpi \in \mathfrak{se}(d)$ [29]. The *homogenized* version of a vector $\mathbf{p} \in \mathbb{R}^d$ is represented by $\bar{\mathbf{p}} = [\mathbf{p}^T \ 1]^T \in \mathbb{R}^{d+1}$. The trace of a matrix \mathbf{A} is denoted by $\text{tr}(\mathbf{A})$. The Kronecker and the Hadamard products are denoted by \otimes and \odot , respectively, and $\mathbf{X} \succcurlyeq 0$ implies \mathbf{X} is a positive-semidefinite matrix.

A. Range measurement model

We assume $N_l > 1$ tags on the robot and seek to find its pose from measurements to N_a anchors. The range measurement at any time t is given by

$$r_{jl}(t) = \|\mathbf{p}_{a_j} - \mathbf{K}\mathbf{T}(t)\bar{\mathbf{p}}_{u_l}\|_2,$$

where $\|\cdot\|_2$ is the L_2 norm, $\mathbf{T}(t)$ is the robot pose at time t , $\mathbf{p}_{a_j} \in \mathbb{R}^d$ is the position of anchor j with respect to the world frame \mathcal{F}_W , \mathbf{p}_{u_l} is the position of tag l with respect to the robot body frame \mathcal{F}_b , and $\bar{\mathbf{p}}_{u_l}$ is its homogeneous form. The matrix \mathbf{K} is such that $\mathbf{p} = \mathbf{K}\bar{\mathbf{p}}$. Similar to [10], [15], we consider a squared-distance measurement model:

$$\tilde{r}_{jl}(t)^2 = \|\mathbf{p}_{a_j} - \mathbf{K}\mathbf{T}(t)\bar{\mathbf{p}}_{u_l}\|_2^2 + \eta_r(t), \quad (2)$$

where $\eta_r(t) \sim \mathcal{N}(0, \sigma_r^2)$ is additive white Gaussian noise of variance σ_r^2 . Next, we derive the MAP formulation and the corresponding SDP relaxation for static initialization.

B. Static initialization

1) *Maximum a posteriori inference*: The state for static initialization is the robot pose at a single time step. The MAP estimate is the robot pose that minimizes the following objective function

$$\mathbf{T}(t)_{\text{MAP}}^* = \arg \min_{\mathbf{T}(t) \in SE(d)} \frac{1}{N_r} \sum_{(j,l) \in \mathcal{E}} \frac{e_{jl}(t)^2}{\sigma_r^2}, \quad (3)$$

where $\mathcal{E} \subseteq \{(j,l) \mid j = 1, \dots, N_a, l = 1, \dots, N_l\}$ is the index set for all measured anchor-tag pairs, $N_r = |\mathcal{E}|$ is the total number of range measurements, and

$$e_{jl}(t) = \tilde{r}_{jl}(t)^2 - \|\mathbf{p}_{a_j} - \mathbf{K}\mathbf{T}(t)\bar{\mathbf{p}}_{u_l}\|_2^2. \quad (4)$$

The MAP problem (3) can be solved using a local solver such as the Gauss-Newton or Levenberg-Marquardt algorithm [29, Section 9.2.5]. Static initialization across multiple time steps

can be done in a similar manner by running multiple instances of (3), one for each time step. As mentioned earlier, in the absence of a good initialization point, the local solver might return a suboptimal robot pose.

We now derive an SDP relaxation to our original problem (3) following the approach presented in [15], but extending it to accommodate pose variables instead of only position variables. First, we reformulate our original problem (3) as a QCQP for which SDP relaxations exist. We begin by making the substitution $\tilde{\mathbf{p}}_{u_l}(t) = \mathbf{K}\mathbf{T}\bar{\mathbf{p}}_{u_l}$, which we refer to as the *lever-arm* substitution, and expand the error term (4):

$$\begin{aligned} e_{jl}(t) &= \tilde{r}_{jl}(t)^2 - \|\mathbf{p}_{a_j} - \tilde{\mathbf{p}}_{u_l}\|_2^2, \\ &= \underbrace{\tilde{r}_{jl}(t)^2 - \|\mathbf{p}_{a_j}\|_2^2}_{d_{jl}(t)} - \underbrace{\|\tilde{\mathbf{p}}_{u_l}(t)\|_2^2}_{z_{u_l}} + 2\mathbf{p}_{a_j}^T \tilde{\mathbf{p}}_{u_l}(t), \\ &= d_{jl}(t) + \begin{bmatrix} 2\mathbf{p}_{a_j}^T & -1 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{p}}_{u_l}(t) \\ z_{u_l} \end{bmatrix}, \end{aligned}$$

where we have introduced a second substitution, $z_{u_l} = \|\tilde{\mathbf{p}}_{u_l}(t)\|_2^2$. With the new substitutions, the error term (4) is linear and the cost function (3) quadratic in the unknown vector $[\tilde{\mathbf{p}}_{u_l}(t)^T \ z_{u_l}]^T$. We define the vector of unknowns $\mathbf{x}_l = [\tilde{\mathbf{p}}_{u_1}(t)^T \ z_{u_1} \dots \tilde{\mathbf{p}}_{u_{N_l}}(t)^T \ z_{u_{N_l}}]^T \in \mathbb{R}^{(d+1)N_l}$. Stacking error terms corresponding to measurements from all anchor-tag pairs, we obtain the following optimization problem equivalent to (3):

$$\begin{aligned} \min_{\substack{\mathbf{x}_l, \mathbf{p}(t), \\ \mathbf{R}(t)}}} \quad & \frac{1}{\sigma_r^2 N_r} \|\mathbf{w} \odot (\mathbf{d} + \mathbf{P}_a \mathbf{x}_l)\|_2^2 \\ \text{s.t.} \quad & \tilde{\mathbf{p}}_{u_l}(t) = \mathbf{K}\mathbf{T}(t)\bar{\mathbf{p}}_{u_l}, \quad l = 1, \dots, N_l, \\ & \|\tilde{\mathbf{p}}_{u_l}(t)\|_2^2 = z_{u_l}, \quad l = 1, \dots, N_l, \\ & \mathbf{R}(t)^T \mathbf{R}(t) = \mathbf{I}_d, \\ & \det(\mathbf{R}(t)) = 1, \end{aligned} \quad (5)$$

where

$$\mathbf{w} = \begin{bmatrix} \delta_{11} \\ \vdots \\ \delta_{N_a N_l} \end{bmatrix}, \mathbf{d} = \begin{bmatrix} d_{11}(t) \\ \vdots \\ d_{N_a N_l}(t) \end{bmatrix}, \mathbf{P}_a = \mathbf{I}_{N_l} \otimes \begin{bmatrix} 2\mathbf{p}_{a_1}^T & -1 \\ \vdots \\ 2\mathbf{p}_{a_{N_a}}^T & -1 \end{bmatrix},$$

and

$$\delta_{jl} = \begin{cases} 1 & \text{if } (j,l) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

The orthogonality ($\mathbf{R}(t)^T \mathbf{R}(t) = \mathbf{I}_d$) and the determinant constraints ($\det(\mathbf{R}(t)) = 1$) associated with the rotation matrix are included as explicit constraints; with this formulation the domain of optimization is now a vector space.

2) *SDP relaxation*: We define our new state as $\mathbf{x} = [\mathbf{x}_l^T \ \text{vec}(\mathbf{R}(t))^T \ \mathbf{p}(t)^T \ h]^T$, where $\text{vec}(\cdot)$ converts the matrix $\mathbf{R}(t)$ into a vector by stacking its columns, and h is a *homogenization* variable. With the new state, (5) can be written as a QCQP:

$$\begin{aligned} q^* &= \min_{\mathbf{x}} \mathbf{x}^T \mathbf{Q} \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}^T \mathbf{A}_0 \mathbf{x} = 1, \\ & \mathbf{x}^T \mathbf{A}_i \mathbf{x} = 0, \quad i = 1, \dots, (d+1)N_l + 1, \\ & \mathbf{x}^T \mathbf{B}_j \mathbf{x} = 0, \quad j = 1, \dots, d(d-1). \end{aligned} \quad (6)$$

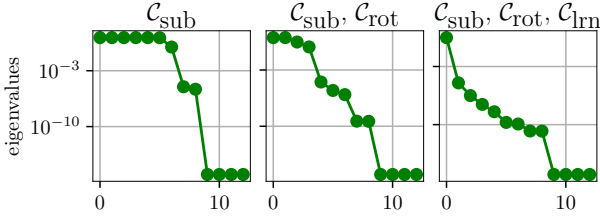


Fig. 2: Qualitative results showing the effect of redundant constraints on the eigenvalue spectrum of the SDP solution. The set \mathcal{C}_{sub} represents the orthogonality and the determinant constraints, and the set \mathcal{C}_{lrn} denotes the additional redundant constraints (8). In all cases, substitution constraints \mathcal{C}_{sub} are included. The inclusion of additional redundant constraints results in a larger ratio of the dominant eigenvalues, and can lead to a rank-1 solution, as shown in the rightmost plot.

The relation between constraints in (5) and (6) is as follows. The matrix \mathbf{A}_0 represents the homogenization constraint, $h^2 = 1$, matrices $\{\mathbf{A}_i \mid i = 1, \dots, dN_l\}$ correspond to the lever-arm constraints, $\{\mathbf{A}_i \mid i = dN_l + 1, \dots, (d+1)N_l + 1\}$ encode $z_{u_i} = \|\tilde{\mathbf{p}}_{u_i}(t)\|_2^2$, and $\{\mathbf{B}_j \mid j = 1, \dots, d(d-1)\}$ encode the orthogonality and the determinant constraints on $\mathbf{R}(t)$. Details on formulation of the orthogonality and the determinant constraints as quadratic constraints can be found in [20], [23], and in Appendix A. To obtain an SDP relaxation for (6), we make the substitution $\mathbf{X} = \mathbf{x}\mathbf{x}^T$. This substitution can be enforced with a convex positive semidefiniteness constraint $\mathbf{X} \succeq 0$ and a non-convex rank constraint $\text{rank}(\mathbf{X}) = 1$. We *relax* the rank constraint to obtain the SDP relaxation:

$$\begin{aligned} p^* = \min_{\mathbf{X}} \quad & \text{tr}(\mathbf{Q}^T \mathbf{X}) \\ \text{s.t.} \quad & \text{tr}(\mathbf{A}_0 \mathbf{X}) = 1, \\ & \text{tr}(\mathbf{A}_i \mathbf{X}) = 0, \quad i = 1, \dots, 2(d+1)N_l, \\ & \text{tr}(\mathbf{B}_j \mathbf{X}) = 0, \quad j = 1, \dots, 2(d-1), \\ & \mathbf{X} \succeq 0. \end{aligned} \quad (7)$$

The SDP relaxation (7) provides a *lower bound* to our original problem (3). If the solution \mathbf{X}^* of (7), is such that $\text{rank}(\mathbf{X}^*) = 1$, then the global minimum to our original problem (3) can be recovered using $\mathbf{X}^* = \mathbf{x}^*\mathbf{x}^{*T}$. In this case, we say that the SDP relaxation is *tight*.

In general, the relaxation (7) may not be tight and we may need to incorporate additional *redundant* constraints to obtain a rank-1 solution [31]. These constraints restrict the feasible set of our SDP relaxation (7) to favor rank-1 solutions, but do not affect the feasible set of the original problem (3). We leverage the method of [28] to automatically generate the necessary redundant constraints by sampling the state space and identifying the nullspace associated with (7). These constraints are added as additional constraints to the original SDP relaxation (7) as

$$\text{tr}(\mathbf{S}_m \mathbf{X}) = 0, \quad \forall m, \quad (8)$$

where each \mathbf{S}_m encodes one redundant constraint. Examples of some redundant constraints for our method can be found in Appendix B. Qualitative results from real 2D static initialization experiments showing the effect of the redundant constraints on the eigenvalue spectrum of the SDP solution are shown in Figure 2.

C. Dynamic initialization

The method described so far requires the robot to be stationary to obtain an initial pose. This can be restrictive in certain dynamic settings such as initializing a quadrotor mid-flight. In dynamic initialization, the goal is to estimate the robot's trajectory, while in motion, over a short time horizon.

1) *Motion model:* We use a constant-velocity motion model where the robot moves with a constant body-centric velocity for duration t_v . The motivation for such a motion model is to obtain SDPs that can be solved quickly for online applications. Specifically, under such a motion model, the robot trajectory is parameterized by the initial robot pose and the constant body-centric velocity. A similar approach has been adopted in [32]. Additionally, we assume range measurements arrive periodically every Δt_r during the time window t_v . The value of t_v depends on the measurement rate and is selected to obtain sufficiently many range measurements. For example, with measurement frequencies greater than 12 Hz, the value of t_v is less than 1s, during which the constant-velocity assumption is often a good approximation.

Let $\mathbf{T}(t_1)$ represent the unknown initial robot pose and $\varpi \in \mathfrak{se}(d) = [\mathbf{v}^T \ \boldsymbol{\omega}^T]^T$ the generalized constant velocity, where \mathbf{v} and $\boldsymbol{\omega}$ represent the body-centric linear and angular velocities, respectively. Under the constant-velocity assumption, the robot pose at any time t_k is given by

$$\mathbf{T}(t_k) = \mathbf{T}(t_1) \exp((c_k \varpi)^\wedge), \quad (9)$$

where $c_k = (k-1)\Delta t_r$, $\exp(\cdot)$ maps an element of the associated Lie algebra back to the Lie group, and $(\cdot)^\wedge$ maps a vector to the corresponding skew-symmetric matrix. Next, we modify the MAP formulation from static initialization to incorporate the motion model for dynamic initialization.

2) *Maximum a posteriori inference:* The measured distance between anchor j and the robot at time t_k is

$$\begin{aligned} \tilde{r}_{jl}(t_k)^2 &= \|\mathbf{p}_{a_j} - \mathbf{K}\mathbf{T}(t_k)\tilde{\mathbf{p}}_{u_l}\|_2^2 + \eta_r(t) \\ &= \|\mathbf{p}_{a_j} - \mathbf{K}\mathbf{T}(t_1) \exp((c_k \varpi)^\wedge) \tilde{\mathbf{p}}_{u_l}\|_2^2 + \eta_r(t), \end{aligned}$$

where the last line follows from (9). The corresponding error term for MAP inference is

$$e_{jl}(t_k) = \tilde{r}_{jl}(t_k)^2 - \|\mathbf{p}_{a_j} - \mathbf{K}\mathbf{T}(t_1) \exp((c_k \varpi)^\wedge) \tilde{\mathbf{p}}_{u_l}\|_2^2.$$

The state for dynamic initialization is $\boldsymbol{\theta} = \{\mathbf{T}(t_1), \varpi\}$ and the corresponding objective function for MAP inference is

$$\boldsymbol{\theta}_{\text{MAP}}^* = \arg \min_{\boldsymbol{\theta}} \frac{1}{N_r} \sum_{k=1}^K \sum_{(j,l) \in \mathcal{E}} \frac{e_{jl}(t_k)^2}{\sigma_r^2}. \quad (10)$$

3) *SDP relaxation:* A benefit of our constant-velocity motion model is that we can reuse the SDP relaxation we developed for static initialization, as the MAP objectives (3) and (10) are similar. To account for motion across different time steps, the lever-arm constraints need to be modified to reflect the relationship between the initial pose, $\mathbf{T}(t_1)$, the

body-centric velocity, ϖ , and the pose at time t_k , $\mathbf{T}(t_k)$. Specifically, the new lever-arm constraint for time t_k is

$$\begin{aligned}\tilde{\mathbf{p}}_{u_i}(t_k) &= \mathbf{KT}(t_k)\tilde{\mathbf{p}}_{u_i} \\ &= \mathbf{KT}(t_1)\exp((c_k\varpi)^\wedge)\tilde{\mathbf{p}}_{u_i} \\ &\approx \mathbf{KT}(t_1)(\mathbf{I} + (c_k\varpi)^\wedge)\tilde{\mathbf{p}}_{u_i} \\ &= \mathbf{KT}(t_1)\tilde{\mathbf{p}}_{u_i} + c_k\mathbf{KT}(t_1)(\varpi^\wedge)\tilde{\mathbf{p}}_{u_i},\end{aligned}\quad (11)$$

where we have used a first-order approximation to the exponential map $\exp(\varpi^\wedge) \approx \mathbf{I}_d + \varpi^\wedge$. The above equation is quadratic in the unknowns and leads to an addition of dN_r constraints to our SDP relaxation (7). Note that with the first-order approximation, we are generating a lower bound to an approximation of our original problem (10); the motivation for doing so is to keep the computational complexity low. Details of an approximation-free approach can be found in Appendix C.

4) *2.5D dynamic initialization*: The size of the state for the SDP relaxation from Section IV-C3 is larger compared to the one in Section IV-B2. Furthermore, as in Section IV-B2, we find that redundant constraints, identified using the tool described in [28], are necessary for the solutions of the SDP relaxation of (10) to be rank 1. The number of required redundant constraints grows quickly with the size of the state, which poses a computational challenge to the SDP solver.

To ensure a tractable initialization, we also consider 2.5D dynamic initialization where the robot pose is given by $\mathbf{T}(t) \in SO(2) \times \mathbb{R}^3$ and the corresponding body-centric velocity by $\varpi = [\mathbf{v}^T \ \omega_z^T]^T$, with $\omega_z = [0 \ 0 \ \omega_z]^T$. Specifically, only rotations around the body z -axis (yaw) and the corresponding angular rate are estimated. From a practical standpoint, this is adequate for most ground and aerial robots with built-in roll and pitch stabilization.

V. EXPERIMENTS

In this section, we show the efficacy of our method through simulations and real experiments. In simulation, we show that our proposed initialization methods recover the global optimum under moderate range-measurement noise regimes. We then evaluate our approach on range data collected using multiple robots.

To solve semidefinite programs, we use the CVXPY [33] package with the MOSEK [34] solver. As a baseline, we compare our method against MAP estimation with a custom implementation of the Levenberg-Marquardt solver. We compare our proposed approach (SDP) and the baseline local solver (LS) using L_2 position and rotation errors:

$$\text{Position Error} = \|\mathbf{p}_{\text{gt}} - \mathbf{p}_{\text{est}}\|_2, \quad (12)$$

$$\text{Rotation Error} = \|\mathbf{R}_{\text{gt}}^T \mathbf{R}_{\text{est}} - \mathbf{I}_d\|_F, \quad (13)$$

where \mathbf{p}_{gt} is the ground-truth position, \mathbf{p}_{est} is the estimated position, \mathbf{R}_{gt} and \mathbf{R}_{est} are the ground-truth and the estimated rotation matrices, and $\|\mathbf{C}\|_F = \|\text{vec}(\mathbf{C})\|_2$ is the Frobenius norm of matrix \mathbf{C} . As mentioned earlier, our SDP relaxation is tight if its solution has rank 1. In order to quantify the rank, we define the following ratio:

$$f_{\text{eig}}(\mathbf{X}) = \log_{10} \left(\frac{e_1}{e_2} \right), \quad (14)$$

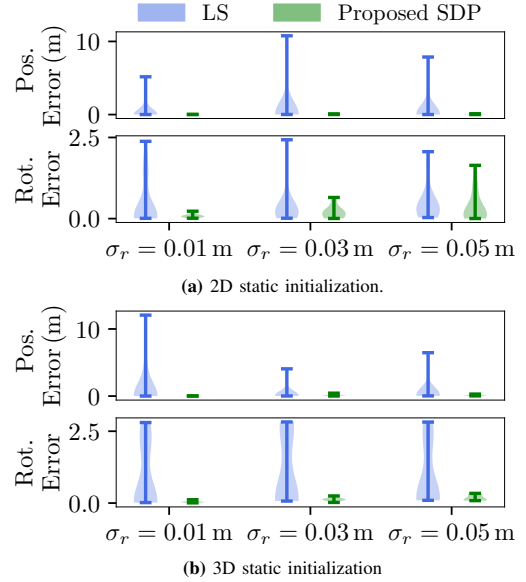


Fig. 3: Distribution of L_2 position and rotation errors from simulation for 2D static initialization (top) and 3D static initialization (bottom). The distributions are generated from 100 Monte Carlo trials across increasing range measurement noise, σ_r . The distribution of errors from the proposed method (SDP) is tighter compared to the baseline local solver (LS). The local solver accuracy is lower as it often gets stuck in local minima. The local solver typically converges to the global minimum when the robot is inside the convex hull of the anchors and performs poorly towards the boundary and outside the convex hull, whereas the proposed approach performs reliably even in such challenging scenarios.

where e_1 and e_2 are the dominant eigenvalues of the SDP solution \mathbf{X} , respectively. A large $f_{\text{eig}}(\mathbf{X})$ suggests a rank-1 solution. All experiments are run on a laptop with an Intel Core i9 9750 CPU with 32 GB RAM.

A. Simulation

The objective of our simulation experiments is to demonstrate that our proposed approach recovers the global optimum while the baseline local solver is susceptible to local minima. Simulation parameters such as the measurement frequency and lever-arm configurations are selected to reflect real sensors and robots. Unless mentioned otherwise, we consider $N_l = 2$ tags in our experiments. The positions of the tags in the body frame are $\mathbf{p}_{u_1} = [0 \ 0.095]^T$ m and $\mathbf{p}_{u_2} = [0 \ -0.095]^T$ m.

Static initialization: We perform multiple Monte Carlo simulations to evaluate the effectiveness of our proposed approach and the baseline in estimating an arbitrary initial pose. The setup for one trial is generated as follows. First, a random anchor configuration and a ground-truth robot pose are generated by (i) sampling (robot and anchor) positions uniformly from $[-4, 4]$ m and (ii) sampling robot rotations uniformly from $[-\pi, \pi]$ rad. We then simulate range measurements between all possible anchor-tag pairs and corrupt the range measurements with Gaussian noise of increasing variance. For a given anchor configuration and ground-truth robot pose, we initialize the local solver with a random robot pose generated using the same procedure as before. Note that our proposed approach does not require an initial point.

For the 2D static initialization, we consider $N_a = 3$ anchors with $N_r = 6$ range measurements to the two tags, \mathbf{p}_{u_1} and \mathbf{p}_{u_2} . Results from 100 simulation experiments are

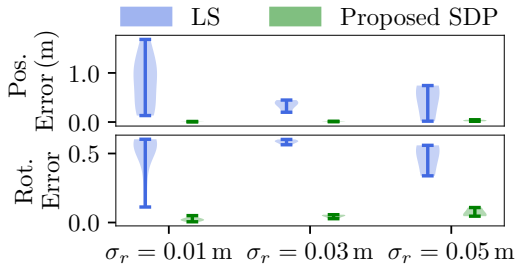


Fig. 4: Distribution of L_2 position (top row) and rotation errors (bottom row) under increasing range measurement noise (σ_r) for 2D dynamic initialization from simulation. The position and rotation errors are computed over the full trajectory, which reflects any errors associated with the estimated velocity. The distribution of errors from the proposed method (SDP) is much tighter than the baseline local solver (LS) as the local solver gets stuck in local minima whereas the proposed method does not.

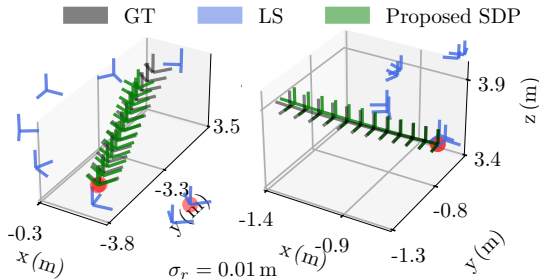


Fig. 5: Two simulation results from 2.5D dynamic initialization showing the trajectories estimated by the iterative local solver (LS) and our proposed method (SDP) along with the ground-truth trajectory (GT). Without a good initial point, the local solver estimates suboptimal trajectories, while our proposed approach is able to generate better trajectory estimates. For each method, the robot pose at $t = 0$ is indicated by a red dot at the origin of the orthogonal axes.

shown in Figure 3a. We see that the proposed method has a tighter distribution of errors compared to the local solver. The distribution of errors is larger for the local solver as it gets stuck in local minima. Additionally, solutions from our proposed method had $f_{\text{eig}}(\mathbf{X}) \geq 7$ in all of our experiments, which we consider rank 1, indicating optimal solutions. In general, the local solver converges to the global minimum when the robot is inside the convex hull of the anchors and performs poorly outside the convex hull, whereas the proposed approach performs reliably even in such challenging scenarios.

In 3D static initialization, we consider $N_a = 4$ anchors and $N_l = 3$ tags with $N_r = 12$ range measurements. The positions of the tags in the body frame are $\mathbf{p}_{u_1} = [0.01 \ 0.41 \ 0]^T$ m, $\mathbf{p}_{u_2} = [0 \ -0.43 \ 0.01]^T$ m, and $\mathbf{p}_{u_3} = [-0.57 \ 0.02 \ 0]^T$ m. Results from 100 simulations are shown in Figure 3b. We see that the proposed method has a tighter distribution of errors compared to the baseline. Additionally, the ratio of the first two dominant eigenvalues of the SDP is large, $f_{\text{eig}}(\mathbf{X}) \geq 8$, which we consider rank 1, indicating an optimal solution.

Dynamic initialization: The setup for a single trial of dynamic initialization is as follows. The anchor configuration and the initial robot pose are generated in a manner similar to the static case. In addition, a body-centric velocity is generated by uniformly sampling linear and angular velocities in the range $[-1, 1]$ m/s and $[-0.3, 0.3]$ rad/s, respectively. The initial pose and velocity are used to generate a constant-velocity trajectory for t_v seconds with a pose every Δt_r seconds. We then simulate range measurements between a tag and an anchor at each robot pose. Unlike the static case, there

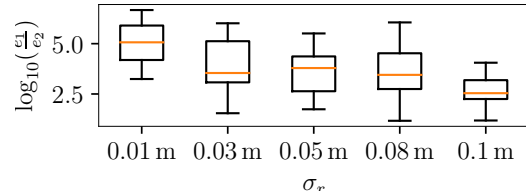


Fig. 6: Simulation results from 2D static initialization quantifying the effect of measurement noise σ_r on the optimality of the solution returned by the SDP solver. We quantify the optimality of an SDP solution as the logarithm of the ratio of its largest and second largest eigenvalues, denoted e_1 and e_2 , respectively. As the magnitude of the measurement noise increases, the ratio of eigenvalues decreases.

is a single range measurement at each time step.

The parameters for the 2D dynamic initialization are as follows. We consider $N_a = 3$ anchors, $N_l = 2$ tags, $N_r = 12$ range measurements, $t_v = 1.1$ s, and $\Delta t_r = 0.1$ s. The lever-arm configuration is the same as in the previous case. Results from 100 simulation experiments are shown in Figure 4. In this case, the position and rotation errors are calculated using all the poses in the robot trajectory, which captures any errors associated with the estimated velocity. Similarly to static initialization, the proposed method can estimate the trajectory reliably as indicated by a tighter distribution. The local solver is more susceptible to local minima in this case, potentially due to the sparse nature of the problem as there is a single range measurement at each time step. The distribution of dominant eigenvalue ratios is still relatively large, $f_{\text{eig}}(\mathbf{X}) \geq 5$, with moderate measurement noise. We still consider it adequate to claim we have rank-1 solutions.

For 2.5D dynamic initialization, we consider $N_a = 4$ anchors and $N_l = 2$ tags. The rest of the parameters are identical to the previous case. Qualitative results showing estimated trajectories with $\sigma_r = 1$ cm from two such experiments are presented in Figure 5. The local solver is particularly susceptible to poor initializations in this case, while the proposed approach recovers the initial trajectory reliably. Additional result from simulation with $\sigma_r = 5$ cm is provided in Appendix D1.

The average time required by the SDP optimizer and the local solver for different initialization methods is presented in Table I. The time taken for 2.5D dynamic initialization is on the higher end due to the additional redundant constraints. However, 2.5D dynamic initialization can still be used to perform delayed initialization of local solvers where older states are initialized followed by an application of a forward motion model until the current time step.

Effect of noise on optimality: To further quantify the effect of noise on the optimality of the solution returned by the SDP solver, we performed multiple 2D dynamic initialization experiments with varying measurement noise magnitudes. Results from 100 experiments across 5 different measurement noise levels are shown in Figure 6. As the magnitude of

TABLE I: Average computation time of the SDP optimizer and the local solver (LS) for static (stat) and dynamic (dyn) initialization in simulation.

	2D stat.	3D stat.	2D dyn.	2.5D dyn.
SDP (s)	0.02	0.05	0.83	3.15
LS (s)	0.003	0.006	1.4	2.22

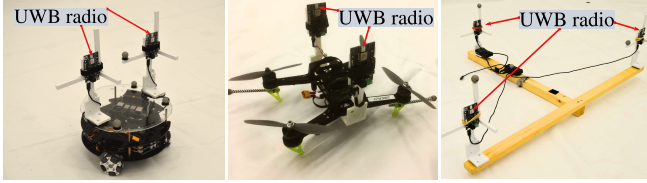


Fig. 7: Our test platform for (i) 2D static and dynamic initialization experiments is a ground robot with two ultrawideband (UWB) tags (left), (ii) 2.5D dynamic initialization experiments is a quadrotor with two tags (centre), and (iii) 3D static initialization experiments is a wand with three tags (right).

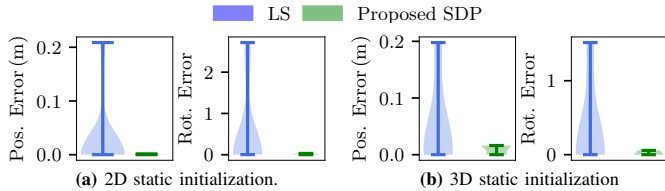


Fig. 8: Distribution of L_2 position and rotation errors for 2D static initialization (left) and 3D static initialization (right) from real experiments. Error distributions for 2D static initialization are generated from 12 experiments where a ground robot’s initial pose is estimated at different locations in the test space without any prior knowledge. For 3D static initialization, the distributions are generated from 10 experiments where a sensor wand’s initial pose is estimated. The tighter spread of errors from the proposed method (SDP) shows that it recovers the initial pose consistently. The accuracy of the local solver (LS) is lower since it gets stuck in local minima.

measurement noise increases, the solution returned by the SDP solver is no longer obviously rank 1.

B. Hardware experiments

Our test space is an indoor flight arena with 6 UWB anchors at the corners of a room of dimensions $7\text{ m} \times 8\text{ m} \times 3.5\text{ m}$. The arena is equipped with a Vicon motion capture system for ground truth. We use the following test platforms: (i) a ground robot with two tags for 2D static and dynamic initialization, (ii) a quadrotor with two tags for 2.5D dynamic initialization, and (iii) a sensor wand with three tags for 3D static initialization (see Figure 7). The positions of the tags with respect to the body frame are the same as in simulation. We remove any constant biases in the range data using ground-truth information.

Static initialization: For 2D static initialization, we performed multiple experiments by driving the ground robot to arbitrary positions in the test space. The range data and the ground-truth pose information were collected onboard for offline evaluation. The local solver is initialized with a random pose. Results from 12 experiments are shown in Figure 8a. We see that the proposed method has a tighter distribution of errors compared to the baseline. The local solver performs poorly as it gets stuck in local minima. The solutions from our proposed method had $f_{\text{eig}}(\mathbf{X}) \geq 7$, which we consider rank 1, indicating global minima. The estimated pose and the eigenvalue spectrum of the SDP solution from one such experiment are shown in Figure 9. As in simulation, the local solver converges to the global minimum when the robot is inside the convex hull of the anchors and performs poorly outside the convex hull, whereas the proposed method performs reliably even in such conditions.

In 3D static initialization, we place the sensor wand at arbitrary poses at multiple locations in the test space and

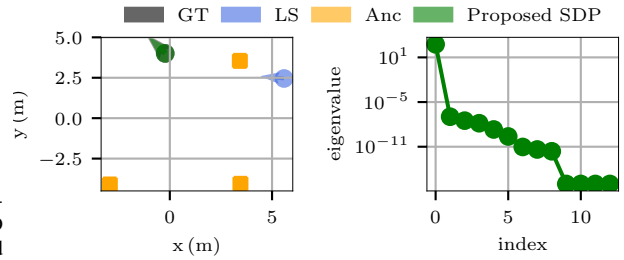


Fig. 9: (Left) Estimated pose from real experiment for 2D static initialization. The pose estimated by the local solver (LS) and the proposed method (SDP) is shown along with the ground-truth pose (GT) and the anchor (Anc) positions. The GT pose overlaps with the SDP pose. (Right) The eigenvalue spectrum suggests a rank-1 solution, indicating a global minimum.

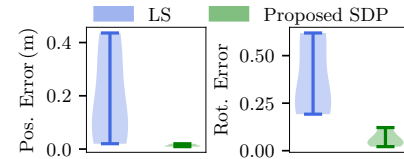


Fig. 10: Distribution of L_2 position and rotation errors for 2D dynamic initialization from 5 real experiments. The narrow spread of errors for the proposed method (SDP) shows the efficacy of the proposed SDP relaxation in estimating the initial trajectory accurately compared to the local solver (LS), which gets stuck in local minima.

collect range data and ground-truth pose information for offline evaluation. Error plots from 10 experiments are shown in Figure 8b. The proposed method has a tighter spread of errors compared to the local solver. Qualitative results along with the eigenvalue spectrum for 3D static initialization can be found in Appendix E1.

Dynamic initialization: For 2D dynamic initialization, the ground robot is commanded along constant-velocity trajectories and the sensor data are recorded onboard for offline evaluation. The parameters for 2D dynamic initialization are the same as in simulation. Results from 5 real experiments are shown in Figure 10. The proposed method estimates the robot trajectory accurately relative to the baseline. We observed that the measurement noise was higher than in simulation ($\sigma_r = 8\text{ cm}$), leading to a drop in the ratio of the first two dominant eigenvalues with $f_{\text{eig}}(\mathbf{X}) = 4$. However, even with a lower $f_{\text{eig}}(\mathbf{X})$, the extracted solution candidates are superior to LS, as seen by the errors in Figure 10. Qualitative results including the eigenvalue spectra can be found in Appendix E2.

The parameters for 2.5D dynamic initialization are $N_a = 5$ anchors, $N_l = 2$ tags, $t_v = 1.1\text{ s}$, $\Delta t_r = 0.1\text{ s}$, and $N_r = 12$ range measurements. The tag positions in quadrotor body frame are $\mathbf{p}_{u_1} = [0\ 0.09]^T\text{ m}$ and $\mathbf{p}_{u_2} = [0\ -0.09]^T\text{ m}$. In each experiment, the quadrotor is commanded along constant-velocity trajectories and the sensor data and the ground-truth information are collected on the onboard computer. Ground-truth pose information from the Vicon system is used for closed-loop control of the quadrotor. We evaluated dynamic initialization on arbitrary segments of the trajectory. The local solver is initialized with the ground-truth pose in each case. Results from one such experiment is shown in Figure 1. The average position and rotation errors from four experiments are provided in Table II. As with 2D dynamic initialization, we observed that the measurement noise was high ($\sigma_r = 8\text{ cm}$).

TABLE II: Average L_2 position and rotation error of the baseline local solver (LS) and the proposed method (SDP) from four hardware 2.5D dynamic initialization experiments.

Algorithm	L_2 position error (m)	L_2 rotation error
LS	0.47	0.72
SDP	0.02	0.07

The baseline LS performs poorly without a good initialization, while our method estimates the trajectories accurately.

VI. CONCLUSION AND FUTURE WORK

In this work, we presented approaches for estimating the initial pose and trajectory of a robot equipped with multiple range sensors. We developed semidefinite relaxations for range-only pose estimation under static conditions and range-only trajectory estimation for motion under constant-velocity trajectories. Through simulation and real experiments, we showed that the proposed relaxations achieve lower estimation error than local solvers and recover optimal initial poses and trajectories under moderate measurement noise. Because redundant constraints are required for tightness, our proposed methods are limited to low-dimensional states. Future work will look at trajectory estimation over longer horizons and attempt to use substitutions that require fewer redundant constraints. Another future direction involves exploring robust estimation approaches such as the truncated least-squares formulation [24] to handle outlier range measurements.

REFERENCES

- [1] Kaplan. *Understanding GPS: Principles and Applications*. Artech, 2018.
- [2] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys and Tutorials*, 21(3):2568–2599, 2019.
- [3] Alan Papalia, Joseph Morales, Kevin J. Doherty, David M. Rosen, and John J. Leonard. SCORE: A Second-Order Conic Initialization for Range-Aided SLAM. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 10637–10644, 2023.
- [4] Jose-Luis Blanco, Javier Gonzalez, and Juan-Antonio Fernandez-Madriral. A pure probabilistic approach to range-only SLAM. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1436–1441, 2008.
- [5] Jeroen D. Hol, Fred Dijkstra, Henk Luinge, and Thomas B. Schon. Tightly coupled UWB/IMU pose estimation. In *Proc. of the IEEE International Conference on Ultra-Wideband (ICUWB)*, pages 688–692, 2009.
- [6] Thien Hoang Nguyen, Thien-Minh Nguyen, and Lihua Xie. Range-Focused Fusion of Camera-IMU-UWB for Accurate and Drift-Reduced Localization. *IEEE Robotics and Automation Letters*, 6(2):1678–1685, 2021.
- [7] Nikolas Trawny, Xun S. Zhou, Ke Zhou, and Stergios I. Roumeliotis. Interrobot transformations in 3-D. *IEEE Transactions on Robotics*, 26(2):226–243, 2010.
- [8] Abhishek Goudar and Angela P. Schoellig. Online Spatio-temporal Calibration of Tightly-coupled Ultrawideband-aided Inertial Localization. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1161–1168, 2021.
- [9] Abhishek Goudar, Timothy D. Barfoot, and Angela P. Schoellig. Continuous-Time Range-Only Pose Estimation. In *2023 20th Conference on Robots and Vision (CRV)*, pages 29–36, Montreal, QC, Canada, June 2023.
- [10] Nikolas Trawny and Stergios I. Roumeliotis. On the global optimum of planar, range-based robot-to-robot relative pose estimation. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3200–3206, 2010.
- [11] Junlin Yan, Christian C. J. M. Tiberius, Gerard J. M. Janssen, Peter J. G. Teunissen, and Giovanni Bellusci. Review of range-based positioning algorithms. *IEEE Aerospace and Electronic Systems Magazine*, 28(8):2–27, 2013.
- [12] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [13] Timothy D Barfoot, Connor Holmes, and Frederike Dümbgen. A Fine Line: Total Least-Squares Line Fitting as QCQP Optimization. arXiv:2206.05082(2), 2022.
- [14] David M. Rosen, Charles DuHadway, and John J. Leonard. A convex relaxation for approximate global optimization in simultaneous localization and mapping. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5822–5829, 2015.
- [15] Frederike Dümbgen, Connor Holmes, and Timothy D. Barfoot. Safe and Smooth: Certified Continuous-Time Range-Only Localization. *IEEE Robotics and Automation Letters*, 8(2):1117–1124, 2022.
- [16] Alan Papalia, Andrew Fishberg, Brendan W. O’Neill, Jonathan P. How, David M. Rosen, and John J. Leonard. Certifiably Correct Range-Aided SLAM. arXiv:2302.11614, 2023.
- [17] Timothy D. Barfoot, Chi Hay Tong, and Simo Särkkä. Batch Continuous-Time Trajectory Estimation as Exactly Sparse Gaussian Process Regression. *Robotics: Science and Systems*.
- [18] Haodong Jiang, Wentao Wang, Yuan Shen, Xinghan Li, Xiaoqi Ren, Biqiang Mu, and Junfeng Wu. Efficient Planar Pose Estimation via UWB Measurements. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1954–1960, 2023.
- [19] Andrew Fishberg and Jonathan P How. Multi-agent relative pose estimation with UWB and constrained communications. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 778–785, 2022.
- [20] Luca Carlone, Giuseppe C. Calafiore, Carlo Tommolillo, and Frank Dellaert. Planar Pose Graph Optimization: Duality, Optimal Solutions, and Verification. *IEEE Transactions on Robotics*, 32(3):545–565, 2016.
- [21] Luca Carlone, David M Rosen, Giuseppe Calafiore, John J Leonard, and Frank Dellaert. Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 125–132, 2015.
- [22] David M. Rosen, Luca Carlone, Afonso S. Bandeira, and John J. Leonard. A Certifiably Correct Algorithm for Synchronization over the Special Euclidean Group. *Springer Proceedings in Advanced Robotics*, 13:64–79, 2020.
- [23] Matthew Giamou, Ziye Ma, Valentin Peretroukhin, and Jonathan Kelly. Certifiably Globally Optimal Extrinsic Calibration From Per-Sensor Egomotion. *IEEE Robotics and Automation Letters*, 4(2):367–374, 2019.
- [24] Heng Yang and Luca Carlone. Certifiably Optimal Outlier-Robust Geometric Perception: Semidefinite Relaxations and Scalable Global Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2816–2834, 2023.
- [25] Jian Jiang, Gang Wang, and K. C. Ho. Accurate Rigid Body Localization via Semidefinite Relaxation Using Range Measurements. *IEEE Signal Processing Letters*, 25(3):378–382, 2018.
- [26] Jian Jiang, Gang Wang, and K. C. Ho. Sensor Network-Based Rigid Body Localization via Semi-Definite Relaxation Using Arrival Time and Doppler Measurements. *IEEE Transactions on Wireless Communications*, 18(2):1011–1025, 2019.
- [27] Xiaoping Wu, Qinman Lin, and Hengnian Qi. Cooperative Multiple Rigid Body Localization via Semidefinite Relaxation Using Range Measurements. *IEEE Transactions on Signal Processing*, 70:4788–4803, 2022.
- [28] Frederike Dümbgen, Connor Holmes, Ben Agro, and Timothy D. Barfoot. Toward Globally Optimal State Estimation Using Automatically Tightened Semidefinite Relaxations. arXiv:2308.05783, 2023.
- [29] Timothy D. Barfoot. *State estimation for robotics*. Cambridge University Press, Second edition, 2023.
- [30] Abhishek Goudar, Frederike Dümbgen, Timothy D. Barfoot, and Angela P. Schoellig. Optimal Initialization Strategies for Range-Only Trajectory Estimation. arXiv:2309.09011, 2023.
- [31] Heng Yang, Jingnan Shi, and Luca Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2021.
- [32] Sean Anderson and Timothy D. Barfoot. RANSAC for motion-distorted 3D visual sensors. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2093–2099, 2013.
- [33] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 2016. To appear.
- [34] MOSEK ApS. *The MOSEK optimization toolbox for Python manual. Version 10.0.*, 2023.

APPENDIX

This appendix is published along with the *arXiv* version of this paper and provides additional details and results.

A. Quadratic formulation of constraints of $SO(d)$ for $d = 2$ and $d = 3$

In this section, we provide details on formulating the constraints associated with the special orthogonal group $SO(d)$, $d \in \{2, 3\}$ as quadratic constraints. Next, we show that nonconvex determinant constraint for $SO(3)$ is equivalent to the columns of the matrix satisfying the right-hand rule, which can be expressed as convex quadratic constraints.

1) *Equivalence of the determinant and the right-handedness constraint for $SO(3)$* : Consider a matrix \mathbf{R} in the orthogonal group:

$$O(3) = \{\mathbf{R} \in \mathbb{R}^3 \mid \mathbf{R}^T \mathbf{R} = \mathbf{I}_3, \mathbf{R} \mathbf{R}^T = \mathbf{I}_3\}.$$

We represent the matrix by its columns as $\mathbf{R} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3]$, where $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\} \in \mathbb{R}^3$. Then we have

$$\mathbf{R}^T (\mathbf{r}_1 \times \mathbf{r}_2) = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} (\mathbf{r}_1 \times \mathbf{r}_2) = \begin{bmatrix} 0 \\ 0 \\ \det(\mathbf{R}) \end{bmatrix},$$

where \times denotes the cross product and $\det(\cdot)$ denotes the determinant operator. The last equality follows from properties of scalar triple product. Multiplying by \mathbf{R} on both sides, we get

$$\mathbf{R} \mathbf{R}^T (\mathbf{r}_1 \times \mathbf{r}_2) = \det(\mathbf{R}) \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{r}_1 \times \mathbf{r}_2 = \det(\mathbf{R}) \mathbf{r}_3.$$

In a similar manner, we have

$$\mathbf{r}_3 \times \mathbf{r}_1 = \det(\mathbf{R}) \mathbf{r}_2,$$

$$\mathbf{r}_2 \times \mathbf{r}_3 = \det(\mathbf{R}) \mathbf{r}_1.$$

From the above equations, the following equivalence holds:

$$\det(\mathbf{R}) = 1 \iff \begin{cases} \mathbf{r}_1 \times \mathbf{r}_2 = \mathbf{r}_3 \\ \mathbf{r}_3 \times \mathbf{r}_1 = \mathbf{r}_2 \\ \mathbf{r}_2 \times \mathbf{r}_3 = \mathbf{r}_1 \end{cases} \quad (15)$$

2) *Quadratic formulation of orthogonality and right-handedness constraints*: The determinant of any 2D matrix \mathbf{R} is

$$\det(\mathbf{R}) = r_1 r_4 - r_2 r_3,$$

where

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix}$$

Thus, the determinant constraint ($\det(\mathbf{R}) = 1$) for a 2D rotation matrix can be expressed as a homogeneous quadratic equation:

$$r_1 r_4 - r_2 r_3 - h^2 = 0, \quad (16)$$

where h is the homogenization variable with the constraint $h^2 = 1$. In this case, $\text{vec}(\mathbf{R}) = [r_1 \ r_3 \ r_2 \ r_4]^T$. For the state

defined in static initialization, $\mathbf{x} = [\mathbf{x}_l^T \ \text{vec}(\mathbf{R}(t))^T \ \mathbf{p}(t)^T \ h]^T$, the above equation can be represented as a quadratic constraint:

$$\mathbf{x}^T \mathbf{B}_j \mathbf{x} = 0, \quad (17)$$

where the entries of the matrix \mathbf{B}_j are the coefficients of the corresponding terms in (16) with the rest of entries set to zero.

For a 3D rotation matrix, using the column representation from Appendix A1 and the equivalence from (15), we can express the determinant constraint with the following homogeneous quadratic equations:

$$[\mathbf{r}_i]_{\times} \mathbf{r}_j - \mathbf{r}_k h - \mathbf{1} h^2 = 0,$$

where $(i, j, k) \in \{(1, 2, 3), (3, 1, 2), (2, 3, 1)\}$, $[\cdot]_{\times}$ maps a vector to the corresponding skew-symmetric matrix [29] and $\mathbf{1}$ is a vector of ones of appropriate dimensions.

The orthogonality constraint ($\mathbf{R}^T \mathbf{R} = \mathbf{I}_d$) for both 2D and 3D rotation matrices can be written as the following homogeneous quadratic equations:

$$\mathbf{r}_i^T \mathbf{r}_i - h^2 = 0, \quad i \in \{1, \dots, d\}, \quad (18)$$

$$\mathbf{r}_i^T \mathbf{r}_j = 0, \quad i, j \in \{1, \dots, d\}, i \neq j, \quad (19)$$

where $d \in \{2, 3\}$.

B. Redundant constraints

We provide some examples of redundant constraints from our 2D static initialization setup. The lever-arm constraint can also be written as

$$\tilde{\mathbf{p}}_{u_l}(t) = \mathbf{R}(t) \mathbf{p}_{u_l} + \mathbf{p}(t), \quad (20)$$

We rewrite the above equation in expanded form, using the lever-arm configuration from our setup, as

$$\underbrace{\begin{bmatrix} \tilde{x}_{ul} \\ \tilde{y}_{ul} \end{bmatrix}}_{\tilde{\mathbf{p}}_{u_l}(t)} = \underbrace{\begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix}}_{\mathbf{R}(t)} \underbrace{\begin{bmatrix} 0 \\ y_{ul} \end{bmatrix}}_{\mathbf{p}_{u_l}} + \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\mathbf{p}(t)}.$$

Examples of redundant constraints corresponding to (20) are

$$(\tilde{x}_{ul} - x)^2 = (\tilde{x}_{ul} - x)(r_2 y_{ul}), \quad l = 1, \dots, N_l,$$

$$(\tilde{y}_{ul} - y)^2 = (\tilde{y}_{ul} - y)(r_4 y_{ul}), \quad l = 1, \dots, N_l.$$

Some redundant constraints involving the rotation matrix are

$$r_1 - r_4 = 0,$$

$$r_2 + r_3 = 0,$$

$$z_{ul}(t)(r_1 - r_4) = 0.$$

Using the homogenization variable, h , the above constraints and all others determined using the method of [28] can be expressed as quadratic constraints for inclusion in our SDP formulation.

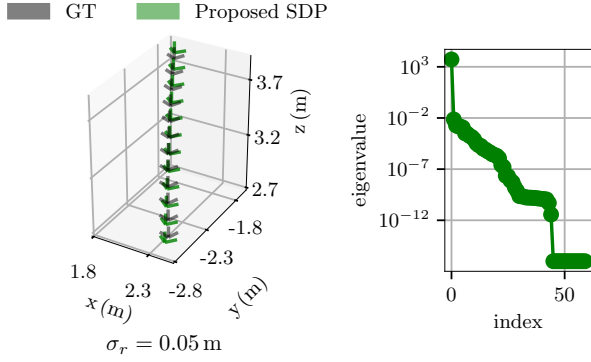


Fig. 11: (Left) Simulation results from 2.5D dynamic initialization with range measurement noise $\sigma_r = 5$ cm. The trajectory estimated by the proposed method (SDP) and the ground-truth trajectory (GT) are shown. The trajectory estimated by the local solver (LS) is not in the vicinity of the ground truth (GT) and hence is not visible. (Right) The eigenvalue spectrum of the SDP solution.

C. Approximation-free dynamic initialization

An exact SDP formulation for (10), without a first-order approximation to the exponential map ($\exp(\varpi^\wedge) \approx \mathbf{I}_d + \varpi^\wedge$) can be obtained as follows. For each lever-arm substitution, $\tilde{\mathbf{p}}_u = \mathbf{K}\mathbf{T}(t_k)\tilde{\mathbf{p}}_{u_i}$, we add the following additional constraints:

$$\mathbf{T}(t_k) = \mathbf{T}(t_{k-1})\delta\mathbf{T}, \quad k = 2 \dots K \quad (21)$$

where $\delta\mathbf{T} := \exp((\varpi\Delta t_r)^\wedge) \in SE(d)$. In addition, the orthogonality and the right-handedness constraints for each $\mathbf{R}(t_k)$ are included in our SDP formulation. The state is then extended to include $\{\mathbf{T}_k | k = 1, \dots, N_r\}$ and $\delta\mathbf{T}$:

$$\mathbf{x} = [\mathbf{x}_i^T \text{vec}(\mathbf{R}(t_1))^T \mathbf{p}(t_1)^T \dots \text{vec}(\mathbf{R}(t_K))^T \mathbf{p}(t_K)^T \text{vec}(\delta\mathbf{R})^T \delta\mathbf{p}^T h]^T$$

This formulation incurs higher computational cost due to the additional variables in the state. Specifically, we found that in the case of 2D dynamic initialization, the SDP optimizer took on average 12 s for the approximation-free approach, even without the inclusion of additional redundant constraints.

D. Simulation

1) *2.5D dynamic initialization:* Qualitative results of the estimated trajectory and the eigenvalue spectrum with range measurement noise $\sigma_r = 5$ cm are shown in Figure 11. The trajectory estimated by the SDP relaxation aligns with the ground-truth trajectory. The trajectory estimated by the local solver is not in the vicinity of the ground truth and hence is not visible.

E. Real experiments

1) *3D static initialization:* Qualitative results from 3D static initialization showing the pose estimated by the local solver and the proposed method are shown in Figure 12. The estimated pose by the proposed method overlaps with the ground-truth pose. The eigenvalue spectrum of the SDP solution shown in Figure 12 suggests a rank-1 solution.

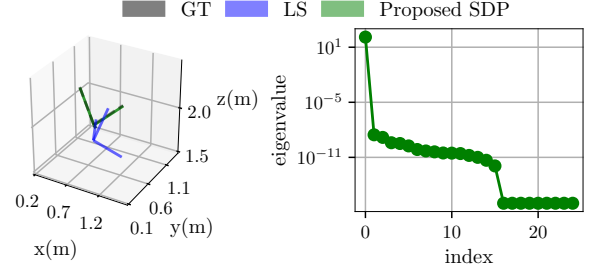


Fig. 12: (Left) Qualitative results of the estimated pose from real 3D static initialization experiments. The pose estimated by the local solver (LS) and the proposed method (SDP) are shown along with the ground-truth pose (GT). The GT pose overlaps with the SDP pose. (Right) Log plot of the eigenvalue magnitudes of the corresponding SDP solution. The eigenvalue spectrum suggests a rank-1 solution, indicating that the proposed method recovers the global minimum.

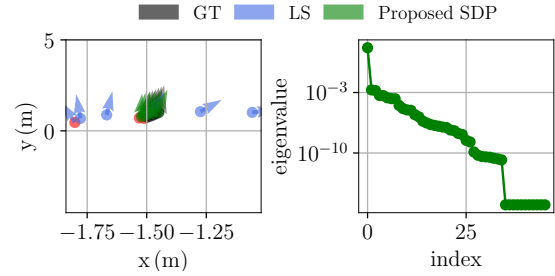


Fig. 13: Results from a real 2D dynamic initialization experiment. (Left) Estimated trajectories from the local solver (LS) and the proposed method (SDP) along with the ground-truth trajectory (GT). The GT trajectory is not discernible as it overlaps with the SDP trajectory. For each method, the robot pose at $t = 0$ is indicated by a red dot at the robot position. (Right) Log plot of the eigenvalue spectrum of the corresponding SDP solution.

2) *2D dynamic initialization:* Results from 2D dynamic initialization with the trajectory estimated by the proposed method and the local solver are shown in Figure 13. The eigenvalue spectrum of the SDP solution suggests a rank-1 solution.