Motion Induced Robot-to-Robot Extrinsic Calibration

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Xun Zhou

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Stergios Roumeliotis, Advisor

May, 2012

Acknowledgements

There are many people who have earned my gratitude for their contribution to my time in graduate school. This thesis would not have been possible without their support.

First of all, I can never overstate my gratitude to my advisor, Professor Stergios Roumeliotis, for his inspiration and encouragement, for the countless hours he spent with me, and for pushing me beyond my limitations. I am also thankful for the time and invaluable advice from my committee members, Professor Giannakis, Professor Banerjee, and Professor Isler, as well as Professor Roberts, who opens the door to algebraic geometry for me. Thanks also go to my labmates, Tassos, Nikolas, Joel, Faraz, Paul, Ke, Esha, Gian-Luca, Dimitrios, Chao, Kejian, and Igor, with all of whom I have shared hours of discussion, work, laughter, food, and joy.

I am also grateful that life in graduate school consisted not only of research, but also the happy times spent with friends and playing music. I would like to thank the conductor, Yuri Ivan, and fellow musicians at the campus orchestra and the Kenwood Symphony Orchestra for the chance to play some great music after a day filled with equations.

Most importantly, I thank my parents for their unconditional love and support. Even though my studies sent me far away from home, my parents have always done everything they can for me to follow my dreams.

Abstract

Multi-robot systems, or mobile sensor networks, which have become increasingly popular due to recent advances in electronics and communications, can be used in a wide range of applications, such as space exploration, search and rescue, target tracking, and cooperative localization and mapping. In contrast to single robots, multi-robot teams are more robust against single-point failures, accomplish coverage tasks more efficiently by dispersing multiple robots into large areas, and achieve higher estimation accuracy by directly communicating and fusing their sensor measurements. Realizing these advantages of multi-robot systems, however, requires addressing certain challenges. Specifically, in order for teams of robots to cooperate, or fuse measurements from geographically dispersed sensors, they need to know their poses with respect to a common frame of reference. Initializing the robots' poses in a common frame is relatively easy when using GPS, but very challenging in the absence of external aids. Moreover, planning the motion of multiple robots to achieve optimal estimation accuracy is quite challenging. Specifically, since the estimation accuracy depends on the locations where the robots record their sensor measurements, it may take an extensive amount of time to reach a required level of accuracy, if the robots' motions are not properly designed.

This thesis offers novel solutions to the aforementioned challenges. The first part of the thesis investigates the problem of relative robot pose initialization, using robot-to-robot distance and/or bearing measurements collected over multiple time steps. In particular, it focuses on solving minimal problems and proves that in 3D there exist only 14 such problems that need to be solved. Furthermore, it provides efficient algorithms for computing the robot-to-robot transformation, which exploit recent advances in algebraic geometry.

The second part of the thesis investigates the problem of optimal motion strategies for localization in leader-follower formations using distance or bearing measurements.

Interestingly, the robot-to-robot pose is unobservable if the robots move on a straight line and maintain their formations, hence, the uncertainty of the robots' poses increases over time. If the robots, however, deviate from the desired formation, their measurements provide additional information which makes the relative pose observable. This thesis addresses the trade-off between maintaining the formation and estimation accuracy, and provides algorithms for computing the optimal positions where the robots should move to in order to collect the most informative measurements at the next time step.

By providing solutions to two important problems for multi-robot systems: motion-induced extrinsic calibration, and optimal motion strategies for relative localization, the work presented in this thesis is expected to promote the use of multi-robot teams in real-world applications.

Contents

List of Tables

List of Figures

Nomenclature

${}^{i}\mathbf{p}_{j}$ Position of frame $\{j\}$ expressed in frame $\{i\}$.

${}^{i}_{j}\mathbf{C}$ Rotation matrix that projects vectors expressed in frame $\{j\}$ to frame $\{i\}$.

$\mathbf{C}(\mathbf{u}, \alpha)$ Rotation matrix describing a rotation about the unit vector $\mathbf{u}$ by an angle α.

$\lfloor \mathbf{u} \times \rfloor$ Skew-symmetric matrix of $\mathbf{u}$ so that $\lfloor \mathbf{u} \times \rfloor \mathbf{v} = \mathbf{u} \times \mathbf{v}$.

d_{ij} Distance between the origins of frames $\{i\}$ and $\{j\}$.

$\mathbf{b}_i$ The bearing from robot R_1 to R_2 when R_1 is at pose $\{i\} = \{2n-1\}, n \in \mathbb{N}^*$, expressed in frame $\{1\}$.

$\mathbf{b}_j$ The bearing from robot R_2 to R_1 when R_2 is at pose $\{j\} = \{2n\}, n \in \mathbb{N}^*$, expressed in frame $\{2\}$.

$s\alpha$ Short for $\sin(\alpha)$.

$c\alpha$ Short for $\cos(\alpha)$.

Chapter 1

Introduction

1.1 Multi-robot Systems

Computers and sensors for navigation purposes are nowadays found not only on robots developed in research laboratories, but also in passenger vehicles often as integral parts of driver-assistance systems. Furthermore, they have become much more compact. A typical pocket size smart-phone comprises computers, cameras, microphones, and GPS which have allowed the development of numerous location-based applications. The decrease in size and cost and the increase in functionality of such systems have made the realization of multi-robot teams possible. Moreover, these developments have opened up new opportunities for deploying groups of robots in a wide range of applications. For example, four Cluster spacecraft [2] flying in formation were used to provide high precision 3D measurements of the Earth's magnetic field necessary for studying plasma physics, and the interaction of the magnetosphere with the solar wind. The Cluster has been key in understanding and modeling the magnetosphere which is impossible if using a single spacecraft. Multi-robot systems are also employed for exploring our oceans. The Autonomous Ocean Sampling Network [1] uses underwater sensors to gather data

necessary for modeling the physical properties of the ocean, such as temperature, salinity, and currents. Besides scientific missions, multi-robot systems have also been used in a large number of applications, such as search and rescue [40, 79], object manipulation [74], target tracking [59, 77, 42, 52], localization [36, 67, 64, 71, 27], and cooperative mapping [25, 37, 47, 18, 81].

The deployment of multi-robot teams has many advantages over single-robot systems. First of all, multi-robot systems can provide additional robustness since hardware failures of one robot do not necessarily prevent the team from completing its mission. Furthermore, multi-robot systems can perform tasks such as search and rescue more efficiently by dispersing multiple robots into large areas. Finally, multi-robot systems performing cooperative estimation tasks, such as tracking, localization, and mapping, can achieve higher accuracy by directly communicating and fusing spatially distributed measurements. However, optimally combining information from multiple robots and efficiently coordinating their actions poses certain challenges that need to be addressed.

1.1.1 Robot-to-robot pose initialization

One of the main challenges of multi-robot systems is the initialization of the robot-to-robot relative position and orientation (pose). Precise knowledge of the 6 degrees-of-freedom (DOF) relative transformation between any pair of robots is a critical prerequisite for multi-robot applications. This information is essential for coordinating the robots, but is also necessary for fusing measurements from geographically dispersed mobile sensors. Specifically, since the sensor measurements are registered with respect to each robot's local frame of reference, they need to be converted to a common reference frame before they can be fused. Most multi-robot estimation algorithms available today assume that the sensor-to-sensor transformation is externally provided [41, 46]. However, only few works describe how this transformation can be determined.

Estimating the robots' relative transformation can be done by *manually* measuring

their relative pose. This approach, however, has several drawbacks. Besides being tedious and time consuming, it provides low accuracy and is inefficient for large robot teams. Another straightforward approach to determining the robots' relative transformation is to use *external references* (e.g., GPS, compass, or a prior map of the environment). In this case, when all robots in a team estimate their poses with respect to a global frame of reference, their relative transformation can be computed indirectly. However, this approach is also problematic. External references, such as GPS, are not always available because of environment induced constraints (e.g., underwater, underground, or indoors). A prior map of the environment is also unavailable in unknown environments, especially during robot exploration tasks. Finally, even very small errors in each robot's global pose estimates will propagate into the derived robot-to-robot transformation. As a result, the accuracy of any subsequent multi-sensor fusion algorithm will be significantly degraded.

Alternatively, in the absence of external aids, the relative robot-to-robot configuration can be computed based on *inter-robot observations* (i.e., robot-to-robot distance and/or bearing measurements). For example, for the case of a *static* sensor network, numerous methods have been proposed for determining the sensors' locations using distance-only measurements between neighboring sensors [73, 21, 14]. However, these approaches are limited to estimating only the 2D positions of static sensors. In many robotic applications though (e.g., localization, mapping, and tracking), both the relative 3D position and orientation of the sensors are necessary. In these cases, the relative pose of robot pairs can be computed by requiring the robots to move and process (i) inter-robot measurements and (ii) estimates of their motion. This *motion-induced extrinsic calibration* process has three main advantages over employing external aids. First of all, it is more cost efficient since no additional hardware is required. Secondly, it is easier to apply in unknown environments where no external aids are available. Finally, due to the flexibility of this method, it is easy to recalibrate the robots' relative pose when necessary.

The aforementioned advantages clearly motivate the need for robust calibration methods for determining the full 6-DOF robot-to-robot relative transformation with no or minimal external aids. In contrast to static sensors, mobile robots can move and record relative measurements from different vantage points, and thus collect additional information for determining their relative poses. Hence, this work will focus on designing efficient motion-induced extrinsic calibration algorithms that do not require any special external hardware, and can be easily applied in the field with minimum assumptions about the operating environment. We believe that the results of this work will reduce the cost and effort necessary for the development and deployment of multi-robot systems.

1.1.2 Optimal motion strategies

Typically, the system dynamic models of the robots and the measurement models of their sensors are nonlinear; therefore, the locations where robots record measurements affect the calibration accuracy[1] . The second problem addressed in this thesis is that of determining the optimal robot motion for maximizing the estimation accuracy. In particular, we focus on robots moving on a straight line while maintaining a 2D formation. In such case, the robot-to-robot relative pose is unobservable given only inter-robot distance or bearing measurements. This is due to the fact that a distance (or bearing) measurement does not provide sufficient information for determining the 3-DOF relative pose between two robots when the pose remains constant over time (i.e., when the robots move in a rigid formation). However, if the robots deviate from the desired formation, their relative pose becomes observable. In this work, we will take advantage of this trade-off between maintaining the formation and making the system observable and introduce methods for maximizing the relative pose estimation accuracy when allowing the robots to slightly deviate from their desired configuration.

[1] For example, in the case of a robot equipped with a camera measuring bearing to a stationary target, it is desirable to move in a direction perpendicular to the line-of-sight to the target, so as to create larger baseline and thus determine the position of the target more accurately.

1.2 Research Objectives

This research seeks to develop efficient algorithms for estimating the robot-to-robot transformation with minimum use of external aids. The necessity for this work is evident when considering teams of robots performing cooperative estimation tasks (e.g., localization, mapping, and tracking) that require combining sensor observations expressed with respect to different frames of reference. Automating the process of determining the robots' relative transformation will not only reduce the cost of robot deployment, but will also allow recalibrating the pose of any robots in the field when their relative pose estimates' uncertainty increases beyond user-specified limits.

The main objectives of this thesis are:

- Development and analysis of motion-induced robot-to-robot extrinsic calibration algorithms. In particular, we will introduce methods for determining the initial robot-to-robot relative transformation based on different types of inter-robot measurements including robot-to-robot distance and bearing, distance-only, bearing-only, or combinations of the above measurements over multiple time steps (e.g., distance during time-step 1, bearing from robot 1 to robot 2 during time-step 2, etc).

- Development of robot-to-robot extrinsic calibration algorithms using observations of a common scene. In particular, our objective is to determine the 5-DOF transformation between two cameras from observations of three points and one direction.

- Development of optimal motion strategies that minimize the uncertainty in the robot-to-robot relative pose estimation. Our objective is to determine the sequence of locations where the robots should collect measurements that provide information along the most uncertain directions of the unknown transformation. In particular, we will focus on the problem of determining the relative pose of

robots moving in 2D formations.

1.3 Organization of the Manuscript

Chapter 2, discusses related work and the limitations of existing approaches. Chapter 3 describes the solution to the problem of estimating the initial robot-to-robot transformation in 2D using distance measurements. Chapter 4, presents algorithms for determining the initial 6-DOF robot-to-robot transformation between pairs of robots based on combinations of inter-robot range and bearing measurements. Chapter 5 addresses the problem of determining the 5-DOF relative camera pose using point-to-point correspondences and a directional constraint. Chapter 6 describes optimal motion strategies for estimating the relative pose of robots moving in 2D formations when using range or bearing measurements. Finally, Chapter 7 provides concluding remarks and directions of future research.

Chapter 2

Related Work

2.1 Calibration with External References

External references, such as a map of the environment, or identifiable features, can be used to determine the relative pose of two robots. One such example is map-based multi-robot localization. Specifically, given a map of features present in an area, robots can use feature observations to localize themselves with respect to the map's (global) frame of reference, and hence compute their relative pose indirectly. The main drawback of this approach is that a precise map is often difficult to create or even impossible in the case of unknown environments.

Alternatively, common observations of identifiable features in the environment can be used to determine the robot-to-robot relative pose. In particular, when two robots can both measure the 3D positions of the same point features, their 6-DOF relative transformation can be determined by solving a least-squares problem that minimizes the sum of the squared distances between the matched points. Horn [34, 35] presented an elegant closed-form solution to this problem. On the other hand, if one robot is able to measure the features' 3D positions while the other can measure its bearing to the same points, the robots' 6-DOF relative pose can be computed using techniques from

computer vision. Representative examples include the work of Ansar and Daniilidis [4], and Quan and Lan [62], both of which provide linear closed-form solutions based on lifting techniques that linearize an overdetermined system of polynomial equations by introducing new variables. Moreover, the robots' 5-DOF relative pose (i.e., up to scale for translation) can be determined from bearing observations of five or more points [57]. Lastly, the robots' relative pose can be estimated from common observations of lines [53] or planes [78], or by matching lines to planes [13].

2.2 Calibration with Sensor-to-Sensor Measurements

Previous work on *extrinsic calibration* of sensor networks, using sensor-to-sensor range measurements, has primarily focused on *static* sensors with the limitation that only the *positions* of the sensor nodes are determined, but not their orientations. Provided that a few anchor nodes can globally localize (e.g., via GPS measurements), the global position of the remaining nodes can be uniquely inferred if certain graph-rigidity constraints are satisfied [5, 24]. A variety of algorithms based on convex optimization [21], sum of squares (SOS) relaxation [56], and multidimensional scaling (MDS) [73] have been employed to localize the sensor nodes in 2D. Additionally, distributed approaches that reduce the communication requirements and better balance the computational load among sensors have also received significant attention [14, 72]. Moreover, flying anchor nodes have been proposed to localize sensors in 3D, e.g., an unmanned aerial vehicle aiding static sensor network localization [90, 58], or a single satellite localizing a stationary planetary rover using distance measurements [33].

For many applications (e.g., for localization, mapping, and tracking), the knowledge of both relative sensor position *and* orientation is required. For mobile sensors, the problem of relative pose determination has only been studied thoroughly in 2D. The ability to move and collect measurements from different vantage points provides additional information for localizing the sensors. This information has been shown to make

the robots' relative pose observable, given inter-robot distance and/or bearing measurements [50]. Specifically, it is known that mutual distance and bearing measurements between two robots from a single vantage point are sufficient for determining their 3-DOF transformation in closed-form [92, 37]. However, when only distance or bearing measurements are available, the robots must move and record additional measurements. Then, the relative robot pose can be found by combining the estimated robot motion (e.g., from odometry) and the mutual bearing [50] or distance [93] measurements.

In contrast to the case of motion in 2D, very little is known about motion-induced extrinsic calibration in 3D. To the best of our knowledge, with the exception of our recent work [86] no algorithms exist for solving the problem of 3D relative pose estimation using robot-to-robot (i) distance and bearing, or (ii) bearing-only measurements. Additionally, few researchers have addressed the more challenging problem of determining relative pose using range-only measurements. In [10], the relative pose is found using multiple simultaneous distance measurements by two arrays of rigidly connected sensors. In this specific problem setup, the pose can be determined from a single set of measurements without the need for intermediate sensor motion, which greatly simplifies the solution. In the most general, minimal problem case, the task of relative pose estimation using only distance measurements is actually equivalent to the forward-kinematics problem of the general Stewart-Gough mechanism [75]. This parallel manipulator consists of a base platform and a moving platform connected by six articulated legs of variable lengths. The forward kinematics problem is to determine the relative pose of the base and the end effector given the six leg lengths and the coordinates of the leg attachment points in the base and end effector frames, respectively. This problem has 40 (generally complex) solutions [88, 19], which can be found by solving a system of multivariate polynomials [38, 45, 39].

Existing approaches to the Stewart-Gough forward kinematics problem based on successively eliminating variables from the system of polynomial equations and solving a 40^{th} order univariate polynomial followed by back-substitution are quite sensitive to

numerical errors. In order to address these limitations, in [84], we presented algorithms for solving systems of multivariate polynomials based on the eigen-decomposition of an associated multiplication matrix, and applied them to the problem of finding 3D relative pose using six distance measurements. We have shown that our algorithms are very fast, use standard double precision data types, and can be very easily implemented using only linear algebra libraries. Additionally, in [85], we introduced a linear method which employs the lifting technique [4] and uses 10 distance measurements to uniquely determine the relative pose.

Finally, in [98, 99, 100], we addressed the most general case, which is also the main focus of this thesis, and determined the 6-DOF relative pose transformation between two robots for any combination of distance and bearing measurements.

2.3 Optimal Motion Strategies

Current research on localization in mobile robot formations mainly focuses on designing specific formation patterns that yield the highest localization accuracy for the team. Specifically, Zhang *et al.* [89] study the impact of the robots' spatial configuration on the localization accuracy for *static* robots that receive absolute position measurements, as well as robot-to-robot range and/or bearing measurements. They establish the necessary and sufficient conditions for the team of robots to be completely localized, and show that the covariance matrix is a function of the relative robot poses. Subsequently, a robot formation that minimizes the trace of the covariance matrix is determined by a gradient descent optimization technique. However, due to the nonconvexity of the objective function, the proposed optimization algorithm does not guarantee global optimality.

Kurazume and Hirose [44] propose a leapfrogging strategy for a team of robots comprising one master and two slave robots who alternate roles in moving toward a destination. When the slave robots are moving, the master robot estimates their positions based on its own pose estimate and range and bearing measurements to the slave

robots. Then the slave robots stop moving, while the master uses them as *portable landmarks* for localization. The authors study the effect of the master-slave relative position on localization accuracy, and find three configurations that show superior performance, by numerically minimizing a weighted least squares cost function. However, the requirement of at least one robot to remain stationary is undesirable.

In [82], all robots move continuously toward a target configuration while measuring distance to each other. In order to determine the optimal trajectories, the authors employed a gradient-based optimization algorithm to minimize the determinant of the covariance matrix at the target configuration. Even though the gradient-based algorithm can only converge to a local optimum, their numerical experiments indicate that the positioning accuracy is improved when the robots do not move in formation.

The effects of formation geometry on the localization accuracy is studied in [32], where the robots record relative position measurements, and the robots' orientations are assumed perfectly known. The optimality criterion is the steady-state position uncertainty of the robot team, and a genetic algorithm is employed for determining the optimal relative robot positions. This algorithm is shown to be a suitable tool for the problem because of the existence of multiple local minima in the objective function. Their results indicate that the optimal formation geometry comprises of adjacent equilateral triangles.

The papers mentioned above have focused on observable systems, where the robots have sufficient measurements (e.g., relative position, orientation, and absolution position) for determining their relative poses. However, due to cost considerations in practice, the robots may only have access to a limited type of measurements, such as inter-robot range or bearing. In this case, the relative poses are unobservable when the robots move in formation on a straight line. This trade-off between maintaining the formation and ensuring system observability has generally been overlooked in the literature with only one exception. Mariottini *et al.* [48] studied the problem of loss of observability in robot formations with bearing-only measurements, and proposed a

switching active-sensing strategy to maintain observability while slightly deviating from the formation. Specifically, a standard leader-follower controller drives the follower to the desired position when the leader does not move on straight paths. When the system becomes unobservable, the standard controller is switched to an active-formation controller that drives the follower to a position which is slightly different from the desired one. However, no optimality, in terms of the estimation accuracy, is claimed by the active controller.

Our work differs from that of [48] in that we seek to determine the optimal location where the follower robot should move to in order to maximize its localization accuracy given constraints on the allowable deviation from the desired formation.

Chapter 3

Distance-only Motion-induced Extrinsic Calibration in 2D

We start our discussion of motion-induced extrinsic calibration with a simple case, i.e., robots moving in 2D using robot-to-robot distance measurements, before presenting (see Chapter 4) the more challenging case of robots moving in 3D using combinations of distance and bearing measurements. Parts of this chapter have been published in a conference [96] and a journal paper [93].

3.1 Introduction

In order to solve distributed estimation problems such as cooperative localization, mapping, and tracking, robots must first determine their relative position and orientation (pose). This extrinsic calibration process is necessary for coordinating a robot team and registering measurements to the same frame of reference. Since the accuracy of the relative (robot-to-robot) transformation can significantly affect the quality of a sensor fusion task (e.g., tracking a target using observations from multiple sensors), it needs to be computed precisely. Mobile robots that move on a plane and use distance *and* bearing sensors (e.g., laser scanners or stereo cameras) can uniquely determine their

relative pose by concurrently processing one distance and two relative bearing measurements recorded in a single location [92]. However, due to power and processing constraints, robots often have to rely on exteroceptive sensors that provide only range measurements. In these cases, computing the unknown robot-to-robot transformation requires developing algorithms for processing multiple distance measurements collected at numerous locations.

Due to their low cost and power consumption, range sensors have recently become widely used for solving localization problems in sensor networks [6] and distributed groups of mobile robots [20]. Range measurements can be obtained using Radio-Frequency (RF) or ultrasound signals. Many RF devices are cheap and already integrated into communication equipment; hence no additional cost is required for range sensing. One of the most common techniques for measuring range is based on the Received Signal Strength Indicator (RSSI). The main advantage of RSSI is that it is available in practically all receivers. An example of such a distance-measuring device is the RADAR system employed for wireless network localization in [6]. Alternatively, distances can be measured precisely by estimating the time-of-flight of an acoustic signal transmitted from one sensor or beacon to another [61, 29]. This type of system comprises a transmitter and a receiver. The transmitter sends out simultaneously a RF synchronization message and a distinct sound. The receiver then measures the time difference in the reception of the two signals and computes the distance using a known model for the speed of sound in the air.

Most current research on applications of range sensing has focused on designing algorithms that process distance measurements to determine only the *position* of each node in a *static* network of sensors [5], or the position and orientation of a mobile robot when *static* beacons are deployed within an area of interest [20]. In the case of networks of sensors, a variety of algorithms based on convex optimization [21] and Multi Dimensional Scaling (MDS) [73], have been employed to localize the sensor nodes. Additionally, distributed approaches that reduce the communication requirements and better balance

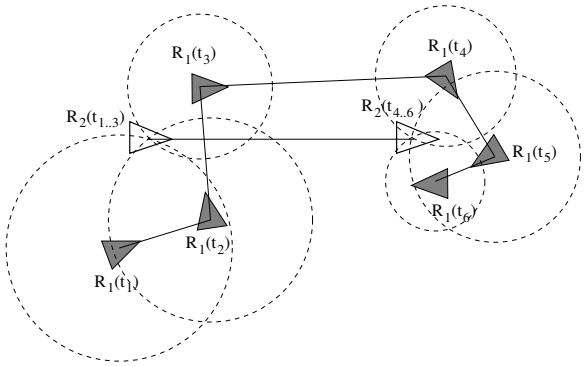


Figure 3.1: A simplistic approach to determine an initial estimate for the robot-to-robot relative transformation using 6 distance measurements. The dark (light) triangles denote the locations of robot R_1 (R_2), and t_i, $t_{i...j}$, $i, j \in \{1, \ldots, 6\}$ indicate the time step(s) that a robot remains at a certain location.

the computational load among sensors have also received significant attention in the related literature (e.g., [72, 14]). In all these cases, the objective is to determine *only* the position of the sensor nodes with respect to anchor nodes that can globally localize via GPS measurements. Similarly, in the case of mobile robots the emphasis is on using distance measurements to localize robots with respect to *static* beacons [20], but not on computing the *relative pose* of the robots. Note, however, that determining the robot-to-robot transformation is a prerequisite for efficiently coordinating the motion of teams of robots [49], localizing cooperatively [50, 69], and in general, expressing measurements in a common frame of reference [92].

The problem we are interested in is that of directly computing the 3 degrees-of-freedom (DOF) robot-to-robot transformation from *distance measurements* and *displacement estimates*, the latter expressed in the reference frame of the corresponding robot. Specifically, we consider pairs[1] of communicating robots equipped with odometric sensors for tracking their motion and range sensors for measuring the distance to each other. In this case, if no prior information about their relative pose is available,

[1] The extension to the problem of multiple robot teams is straightforward once a solution to the pair-wise problem is determined.

a human operator will need to manually measure the transformation between the two robots before they can be deployed to perform their assigned task (e.g., cooperative localization [69]). However, this tedious process limits the accuracy of the robot-to-robot transformation and increases the cost of deploying large teams of robots due to the time and effort required.

A straightforward approach to automating this *extrinsic calibration* process is for the robots to move randomly, collect distance measurements, and then compute their relative transformation by employing nonlinear least squares. However, any iterative process applied to minimize the nonlinear, in the unknown variables, cost function relies on the existence of an *accurate initial estimate* in order to converge to the global minimum. Additionally, since the necessary number of range measurements is not known a priori, a conservative strategy would require the robots to spend excessive time and energy measuring their distance numerous times at different locations. Instead it would be beneficial for the robots to follow a two-step process: (i) Employ a non-iterative algorithm to process the minimum number of distance measurements required to compute an initial estimate of their relative transformation, and (ii) Apply nonlinear least squares to iteratively refine this initial estimate using additional range measurements. This second step can be repeated until the user-specified level of accuracy is reached.

A simplistic method to compute an initial estimate for the 3-DOF transformation in closed form would require the robots to follow a sequence of *coordinated motions* and measure distances to each other at certain locations and time instants. Specifically, as shown in Fig. 3.1, if robot R_2 remains static while R_1 measures its distance to R_2 at 3 different locations (time instants t_1, t_2, and t_3), the position of R_2 with respect to R_1 can be uniquely determined. In order to also compute their relative orientation, robot R_2 will need to move to a new location and remain static again until robot R_1 records another 3 distance measurements (time instants t_4, t_5, and t_6) and triangulates the new relative position of robot R_2. Using these 2 inferred relative position measurements and knowing the direction of motion of R_2 (computed from its own odometry), the relative

orientation between the 2 robots can be uniquely determined. The main drawback of this approach is that it requires tight coordination between the robots for performing the sequence of necessary motions and recording the distance measurements at the appropriate locations. Additionally, this initial calibration phase delays the onset of the actual robot task which can be detrimental in time-critical situations involving large robot teams.

In this chapter, we address this problem by deriving non-iterative algorithms for computing an initial estimate of the 3-DOF robot-to-robot transformation without restricting their motion [96, 93]. Specifically, we prove that when 2 robots move randomly and collect 3 distance measurements at different locations, the maximum number of solutions, generically, is 6 (cf. Lemma 1, Section 3.2). When 4 range measurements are available, we show that, generically, there exist no more than 4 solutions (cf. Lemma 3, Section 3.3). Furthermore, in Section 3.4 (cf. Lemma 5) we prove that for nonsingular configurations the relative pose of the robots can be uniquely determined given 5 distance measurements (instead of 6 based on the simplistic method outlined in Fig. 3.1). Efficient algorithms for computing *all* solutions for the cases described above are presented. Additionally, we provide a novel linear algorithm for determining the unique solution (i.e., when 5 range measurements are available) that minimizes the numerical error in the computed transformation (cf. Section 3.4.2). In Section 3.5, we describe the nonlinear least-squares algorithm that uses all range measurements available to iteratively refine the initial estimate for the unknown robot-to-robot transformation. In Section 3.6, we analyze the system observability and provide necessary and sufficient conditions for observability based on the control inputs applied to the two robots. In Section 3.7, we present simulation and experimental results that verify the validity of our theoretical analysis.

3.2 Determining the Relative Pose from 3 Distance Measurements: At Most 6 Solutions

Consider two robots R_1 and R_2 whose initial poses (position and orientation) are indicated by the frames of reference $\{1\}$ and $\{2\}$ respectively (cf. Fig. 3.2). The two robots move randomly through a sequence of odd poses $\{1\}, \{3\}, \ldots, \{2n-1\}$ for R_1, and even poses $\{2\}, \{4\}, \ldots, \{2n\}$ for R_2 and measure their distance d_{ij}, $i \in \{1, \ldots, 2n-1\}$, $j \in \{2, \ldots, 2n\}$ at each of these locations. Without loss of generality, we assume that only one of the robots records range measurements at each location. If both robots measure the same distance, the two measurements can be combined to provide a more accurate estimate of their distance.

Additionally, the robots are equipped with odometric sensors for estimating their poses with respect to their initial frames of reference. That is, robot R_1 estimates the position vectors ${}^1\mathbf{p}_3, \ldots, {}^1\mathbf{p}_{2n-1}$, and the orientation angles ${}^1\phi_3, \ldots, {}^1\phi_{2n-1}$ necessary for determining the rotational matrices ${}^1_3\mathbf{C}, \ldots, {}^1_{2n-1}\mathbf{C}$. Similarly, the quantities ${}^2\mathbf{p}_4, \ldots, {}^2\mathbf{p}_{2n}$ and ${}^2\phi_4, \ldots, {}^2\phi_{2n}$ (and hence ${}^2_4\mathbf{C}, \ldots, {}^2_{2n}\mathbf{C}$) are estimated by robot R_2 from its own odometry.[2] At this point, we should note that the particular kinematic model used by each robot and the motion measurements available to it, do not affect the solution of the relative pose problem. Actually, no motion measurements need to be exchanged between the robots. Instead, only the resulting position displacement estimates (i.e., ${}^1\mathbf{p}_{2\nu-1}$ and ${}^2\mathbf{p}_{2\nu}$, $\nu = 2, \ldots, n$) must be communicated.

Our goal is to use the odometry-based estimates and the n distance measurements to determine the *maximum number of solutions* for the initial 3-DOF robot-to-robot transformation, i.e., their relative position ${}^1\mathbf{p}_2$ and orientation ${}^1\phi_2 = \phi$, or equivalently

[2] As it will become evident, only the position vectors are required for estimating the unknown robot-to-robot transformation. The orientation angles at intermediate steps are used for expressing the next position vector in the original frame of reference of each robot.

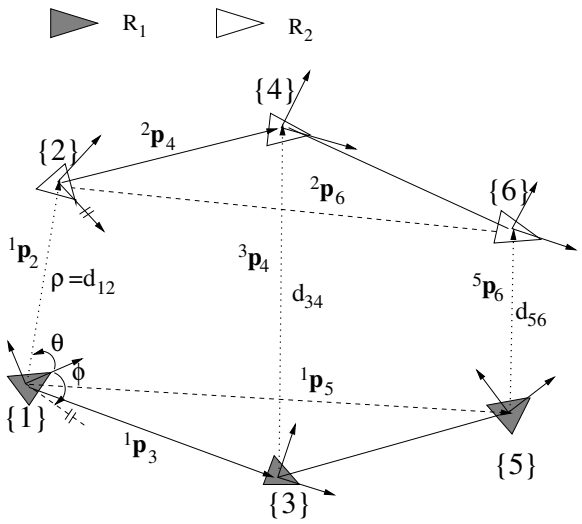


Figure 3.2: The trajectories of robots R_1 and R_2 in 2D. The odd (even) numbered frames of reference depict the consecutive poses of robot R_1 (R_2). d_{ij}, $i \in \{1, \ldots, 2n-1\}$, $j \in \{2, \ldots, 2n\}$ denotes the distance between the two robots when aligned to frames $\{i\}$ and $\{j\}$ respectively.

(in polar coordinates) θ and ϕ, with

$$ {}^1\mathbf{p}_2 = \rho \begin{bmatrix} c\theta \\ s\theta \end{bmatrix} \quad , \quad {}^1_2\mathbf{C} = \begin{bmatrix} c\phi & -s\phi \\ s\phi & c\phi \end{bmatrix} \tag{3.1} $$

Note that $\rho = d_{12}$ is measured and considered known hereafter.

We first address the case when $n = 3$ distance measurements (d_{12}, d_{34}, and d_{56}) are available. We proceed by substituting the geometric relations for the position vectors ${}^3\mathbf{p}_4$, ${}^5\mathbf{p}_6$ (cf. Fig. 3.2)

$$ {}^3\mathbf{p}_4 = {}^1_3\mathbf{C}^T({}^1\mathbf{p}_2 + {}^1_2\mathbf{C}\,{}^2\mathbf{p}_4 - {}^1\mathbf{p}_3) \tag{3.2} $$

$$ {}^5\mathbf{p}_6 = {}^1_5\mathbf{C}^T({}^1\mathbf{p}_2 + {}^1_2\mathbf{C}\,{}^2\mathbf{p}_6 - {}^1\mathbf{p}_5) \tag{3.3} $$

in the following expressions for the distance measurements d_{34} and d_{56}, respectively:

$$ d_{34}^2 = {}^3\mathbf{p}_4^T\,{}^3\mathbf{p}_4 \quad , \quad d_{56}^2 = {}^5\mathbf{p}_6^T\,{}^5\mathbf{p}_6 \tag{3.4} $$

After rearranging terms and substituting $\rho^2 = d_{12}^2$ for ${}^1\mathbf{p}_2^T {}^1\mathbf{p}_2$, these can be written as:

$$({}^1\mathbf{p}_2 - {}^1\mathbf{p}_3)^T {}^1_2\mathbf{C}\, {}^2\mathbf{p}_4 - {}^1\mathbf{p}_2^T {}^1\mathbf{p}_3 = a_0 \tag{3.5}$$

$$({}^1\mathbf{p}_2 - {}^1\mathbf{p}_5)^T {}^1_2\mathbf{C}\, {}^2\mathbf{p}_6 - {}^1\mathbf{p}_2^T {}^1\mathbf{p}_5 = b_0 \tag{3.6}$$

where

$$a_0 = 0.5(d_{34}^2 - \rho^2 - {}^2\mathbf{p}_4^T {}^2\mathbf{p}_4 - {}^1\mathbf{p}_3^T {}^1\mathbf{p}_3)$$

$$b_0 = 0.5(d_{56}^2 - \rho^2 - {}^2\mathbf{p}_6^T {}^2\mathbf{p}_6 - {}^1\mathbf{p}_5^T {}^1\mathbf{p}_5)$$

Note that a_0 and b_0 on the right-hand side of (3.5) and (3.6) are known (measured or estimated), while the unknown variables θ and ϕ, embedded in ${}^1\mathbf{p}_2$ and ${}^1_2\mathbf{C}$ (cf. (3.1)), only appear on the left-hand side expressions.

Equations (3.5) and (3.6) form a system of 2 nonlinear equations in the 2 unknowns θ and ϕ. Applying standard numerical techniques, such as Newton-Raphson [60], for solving this system has three major drawbacks: (i) may take a large number of iterations, (ii) requires an accurate initial estimate, and (iii) provides no guarantee of finding all solutions. The first drawback is that iterative processes often require a large number of steps before converging to a solution. The second drawback is that in order for the algorithm to converge to the correct answer, *initial estimates* close to the true values of the unknown variables need to be specified. In practice, however, no such information is available; the only prior knowledge we have for θ and ϕ is that each lies within the interval $[0,\ 2\pi)$. Finally, if for example only $n = 3$ distance measurements are available, the total number of solutions that needs to be determined is 6 (cf. Lemma 1). To compute all possible roots, numerous initial estimates for the unknowns θ and ϕ must be drawn from the 2-dimensional region $[0,\ 2\pi) \times [0,\ 2\pi)$. Such procedure requires a large number of initializations of the iterative process while provides no guarantees that all 6 solutions will be computed.[3]

[3] To further stress this point, consider the case where after numerous initializations of the Newton-Raphson algorithm, only 4 real roots have been determined. In this situation there is no clear way to unequivocally declare the remaining 2 roots complex and terminate the search, or continue the iterations with new initial values in search of the remaining roots.

Instead we hereafter seek to find all possible solutions algebraically. Notice that (3.5) and (3.6) can be transformed to polynomial equations by treating $c\phi$, $s\phi$, $c\theta$, and $s\theta$ as 4 unknown variables. Together with the 2 trigonometric constraints $s\phi^2 + c\phi^2 = 1$ and $s\theta^2 + c\theta^2 = 1$, these form a system of 4 polynomial equations, namely f_1^*, f_2^*, f_3^*, and f_4^*, in 4 unknowns. In general, none of these polynomials lies in the ideal generated by the remaining ones (cf. [16] for the definitions of ideals and varieties), and thus the square system considered has a finite number of solutions. However, there exist degenerate cases where the system has infinite solutions. For example, when the two robots move on parallel straight lines with equal linear velocities, the bearing angle θ can take any value within the interval $[0,\ 2\pi)$. Such a degenerate case occurs because f_4^* belongs to the ideal $< f_1^*,\ f_2^*,\ f_3^* >$, i.e., $< f_1^*,\ f_2^*,\ f_3^*,\ f_4^* >=< f_1^*,\ f_2^*,\ f_3^* >$. Based on Proposition 4 of Ch. 1, §4 in [16], if $< f_1^*,\ f_2^*,\ f_3^*,\ f_4^* >=< f_1^*,\ f_2^*,\ f_3^* >$, then the varieties of these two ideals are equal, i.e., $\mathbf{V}(f_1^*,\ f_2^*,\ f_3^*,\ f_4^*) = \mathbf{V}(f_1^*,\ f_2^*,\ f_3^*)$. Equivalently, the 4 unknowns are only constrained by 3 equations, and thus infinite solutions exist. Note though that when the two robots move randomly while satisfying the conditions of Lemma 13 (cf. Section 3.6.3), the corresponding continuous-time system is locally weakly observable, and hence within an open neighborhood around a solution of the robot-to-robot transformation, there exists no other point indistinguishable from it (cf. Definition 9 in Section 3.6.1). That is, generically, the discrete-time system will have a finite number of solutions and the variety $\mathbf{V}(f_1^*,\ f_2^*,\ f_3^*,\ f_4^*)$ has dimension zero. We hereafter concentrate on nonsingular configurations and prove the following lemma:

Lemma 1. *Given 3 distance measurements between two robots at 3 different locations, the maximum number of solutions for the 3-DOF robot-to-robot transformation, generically, is 6.*

Proof. The following derivation is based on an elimination process for removing the quantities $c\theta$, $s\phi$, $c\phi$ from the expressions in (3.5) and (3.6), which results in a 6^{th} order polynomial in the unknown variable $y := s\theta$. The key idea behind this approach is

similar to Gaussian Elimination for linear systems of equations. Due to space limitations only the main steps of this process are shown while reassignment of variables is used to preserve the clarity of presentation.

By substituting the displacement estimates (known from odometry) for the two robots:

$$ {}^1\mathbf{p}_3 = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \quad {}^2\mathbf{p}_4 = \begin{bmatrix} a_3 \\ a_4 \end{bmatrix}, \quad {}^1\mathbf{p}_5 = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad {}^2\mathbf{p}_6 = \begin{bmatrix} b_3 \\ b_4 \end{bmatrix} $$

in (3.5) and (3.6), and defining as

$$ a_5 := a_2 a_4 + a_1 a_3, \quad a_6 := a_2 a_3 - a_1 a_4 $$
$$ b_5 := b_2 b_4 + b_1 b_3, \quad b_6 := b_2 b_3 - b_1 b_4 $$

we have:

$$ \underbrace{(\rho a_3 c\theta + \rho a_4 s\theta - a_5)}_{u_1} c\phi + \underbrace{(\rho a_3 s\theta - \rho a_4 c\theta - a_6)}_{v_1} s\phi = \underbrace{a_0 + \rho(a_1 c\theta + a_2 s\theta)}_{w_1} \tag{3.7} $$

$$ \underbrace{(\rho b_3 c\theta + \rho b_4 s\theta - b_5)}_{u_2} c\phi + \underbrace{(\rho b_3 s\theta - \rho b_4 c\theta - b_6)}_{v_2} s\phi = \underbrace{b_0 + \rho(b_1 c\theta + b_2 s\theta)}_{w_2} \tag{3.8} $$

Equations (3.7) and (3.8) can be written in a matrix form as:

$$ \begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \end{bmatrix} \begin{bmatrix} c\phi \\ s\phi \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{3.9} $$

Note that (3.9), is linear in the unknowns $c\phi$ and $s\phi$. Solving for these two variables we have:

$$ \begin{bmatrix} c\phi \\ s\phi \end{bmatrix} = \frac{1}{\det} \begin{bmatrix} v_2 w_1 - v_1 w_2 \\ u_1 w_2 - u_2 w_1 \end{bmatrix} \tag{3.10} $$

where, $\det = u_1 v_2 - u_2 v_1$. Substituting the above expressions for $c\phi$ and $s\phi$ in the trigonometric constraint $s\phi^2 + c\phi^2 = 1$ results in a single equation in the variables $c\theta$

and $s\theta$

$$(u_1v_2 - u_2v_1)^2 = (v_2w_1 - v_1w_2)^2 + (u_1w_2 - u_2w_1)^2$$
$$\Rightarrow (u_1v_2 - u_2v_1)^2 = (v_2^2 + u_2^2)w_1^2 + (v_1^2 + u_1^2)w_2^2 - 2(v_1v_2 + u_1u_2)w_1w_2 \tag{3.11}$$

As detailed in [97], the terms $v_2^2+u_2^2$, $v_1^2+u_1^2$, $v_1v_2+u_1u_2$, and $u_1v_2-u_2v_1$ are all *linear* in $c\theta$ and $s\theta$, while w_1^2, w_2^2, w_1w_2 are all *quadratic* in the same quantities. Hence (3.11) is a 3^{rd} order polynomial in $x := c\theta$ and $y := s\theta$, and can be written in the following simpler form:

$$\begin{aligned} f_1 =& m_9x^3 + m_8x^2y + m_7xy^2 + m_6x^2 + m_5xy + m_4x + \\ & m_3y^3 + m_2y^2 + m_1y + m_0 = 0 \end{aligned} \tag{3.12}$$

where the constants $m_0, \ldots, m_9$ are functions of known quantities [97]. The final step in the elimination process is to invoke the trigonometric constraint

$$f_2 = x^2 + y^2 - 1 = 0 \tag{3.13}$$

and eliminate x from (3.12) by using the Sylvester Resultant [3]. Specifically, by multiplying (3.12) with x, and (3.13) with x and x^2, and rewriting the resulting equations in a matrix form, we have:

$$\begin{bmatrix} s_3 & s_2 & s_1 & s_0 & 0 \\ 0 & s_3 & s_2 & s_1 & s_0 \\ 1 & 0 & y^2-1 & 0 & 0 \\ 0 & 1 & 0 & y^2-1 & 0 \\ 0 & 0 & 1 & 0 & y^2-1 \end{bmatrix} \begin{bmatrix} x^4 \\ x^3 \\ x^2 \\ x \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.14}$$

with

$$s_0 := m_3y^3 + m_2y^2 + m_1y + m_0$$
$$s_1 := m_7y^2 + m_5y + m_4$$
$$s_2 := m_8y + m_6 \quad , \quad s_3 := m_9$$

For the polynomials in (3.12) and (3.13) to have common roots, the determinant of the 6×6 Sylvester matrix on the left-hand side of (3.14) must be equal to zero. As shown in [97], the determinant is a 6^{th} order polynomial in the single variable y:

$$g_2(y) = \xi_6 y^6 + \xi_5 y^5 + \xi_4 y^4 + \xi_3 y^3 + \xi_2 y^2 + \xi_1 y + \xi_0 \tag{3.15}$$

where the constants $\xi_0, \ldots, \xi_6$ are functions of the known quantities $m_0, \ldots, m_9$. Therefore, the maximum number of solutions for y, including complex roots, is 6.

To prove our claim that there exist at most 6 solutions for θ, we also need to show that for every solution of $y := c\theta$ (cf. (3.15)), only one solution for $x := s\theta$ can be found (cf. (3.12)). We present two possible ways for proving this:

By directly computing the Groebner basis [16] for this system of equations, it is easy to show that one base, g_2, is exactly the same as the polynomial in (3.15), while g_1 has the form:

$$g_1(x, y) = x + k_5 y^5 + k_4 y^4 + k_3 y^3 + k_2 y^2 + k_1 y + k_0$$

where the constants $k_0, \ldots, k_5$ are functions of known quantities [97]. Therefore, for every root y_i^*, $i = 1, \ldots, 6$ of $g_2(y) = 0$, there exists *only one* solution x_i^* of $g_1(x, y_i^*) = 0$ corresponding to it.

Alternatively, we can draw the same conclusion without explicitly computing the Groebner basis. Instead, we only need to show that the leading term of g_1, $\text{LT}(g_1)$, is linear in x. This can be easily seen by using the definition of a Groebner basis [16]: A set $\{g_1, \ldots, g_s\} \subset I$ is a Groebner basis of an ideal I if the leading term of *any* element of I is divisible by one of the $\text{LT}(g_i)$. Specifically, we can construct one element f_3 of the ideal $I =< f_1, f_2 >$ by setting

$$\begin{aligned} f_3 &= f_1 - (m_9 x + m_8 y + m_6) f_2 \\ &= (m_7 - m_9) x y^2 + (\text{lower order terms}) \end{aligned}$$

Since the leading term $\text{LT}(f_3) = (m_7 - m_9) x y^2$ must be divisible by $\text{LT}(g_1)$, the degree

of x in LT(g_1) has to be 1, or equivalently, g_1 is linear in x. Hence, the total number of distinct solutions for $(x,\ y)$ remains 6.

Finally, ϕ is uniquely determined by back-substitution of $x = c\theta$ and $y = s\theta$ in (3.10). □

Although in general there exist 6, possibly complex, solutions for (3.15), the total number of *real roots* will depend on the robots' trajectories and cannot be determined a priori. A scenario with 6 real solutions is shown in Fig. 3.8.

Corollary 2. *Given 3 distance measurements between two robots at 3 different locations, generically, there exist 2, 4, or 6 solutions for the 3-DOF robot-to-robot transformation.*

This is evident if one considers that $g_2(y)$ (cf. (3.15)) is a 6^{th} order polynomial with real coefficients, and complex roots appear in conjugate pairs.

3.2.1 Computing all possible solutions

There exist many methods to compute the roots of a single-variable polynomial. Our approach relies on the eigen-decomposition of the 6×6 *companion matrix* [22] for (3.15):

$$\begin{bmatrix} 0 & & & & & -\xi_0/\xi_6 \\ 1 & 0 & & & & -\xi_1/\xi_6 \\ & 1 & 0 & & & -\xi_2/\xi_6 \\ & & \ddots & & & \vdots \\ & & & & 1 & -\xi_5/\xi_6 \end{bmatrix}$$

While this method will determine all 6 roots of the polynomial (eigenvalues of the companion matrix), only the real ones are of practical interest since they have a geometric interpretation. Once y is known, x is determined by computing the null space of the matrix in (3.14).

3.3 Determining the Relative Pose from 4 Distance Measurements: At Most 4 Solutions

Consider now the case where the robots R_1 and R_2 continue their paths shown in Fig. 3.2 and move to the new poses $\{7\}$ and $\{8\}$, respectively, where they record an additional distance measurement d_{78}. We will prove the following:

Lemma 3. *Given 4 distance measurements between two robots at 4 different locations, the maximum number of solutions for the 3-DOF robot-to-robot transformation, generically, is 4.*

Proof. We proceed in a similar manner as for the case of 3 distance measurements. Specifically, the new position estimates for the two robots at the locations where they record their 4^{th} distance measurement

$$ {}^1\mathbf{p}_7 = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}, \quad {}^2\mathbf{p}_8 = \begin{bmatrix} e_3 \\ e_4 \end{bmatrix} $$

are related through the following geometric constraint (analogous to (3.2)):

$$ {}^7\mathbf{p}_8 = {}^1_7\mathbf{C}^T({}^1\mathbf{p}_2 + {}^1_2\mathbf{C}\,{}^2\mathbf{p}_8 - {}^1\mathbf{p}_7) \tag{3.16} $$

Substituting in the expression for the new distance measurement, $d_{78}^2 = {}^7\mathbf{p}_8^T\,{}^7\mathbf{p}_8$, results in the following equation:

$$ ({}^1\mathbf{p}_2 - {}^1\mathbf{p}_7)^T\,{}^1_2\mathbf{C}\,{}^2\mathbf{p}_8 - {}^1\mathbf{p}_2^T\,{}^1\mathbf{p}_7 = e_0 \tag{3.17} $$

where

$$ e_0 = 0.5(d_{78}^2 - \rho^2 - {}^2\mathbf{p}_8^T\,{}^2\mathbf{p}_8 - {}^1\mathbf{p}_7^T\,{}^1\mathbf{p}_7) $$

is known (i.e., it is computed based on measured and estimated quantities). Following the same algebraic process as in the previous section, (3.17) can be written as:

$$ \underbrace{(\rho e_3 c\theta + \rho e_4 s\theta - e_5)}_{u_3} c\phi + \underbrace{(\rho e_3 s\theta - \rho e_4 c\theta - e_6)}_{v_3} s\phi = \underbrace{e_0 + \rho(e_1 c\theta + e_2 s\theta)}_{w_3} \tag{3.18} $$

where e_5 and e_6 are defined as before. Rewriting (3.7), (3.8), and (3.18) in a matrix form, we have:

$$\begin{bmatrix} u_1 & v_1 & -w_1 \\ u_2 & v_2 & -w_2 \\ u_3 & v_3 & -w_3 \end{bmatrix} \begin{bmatrix} c\phi \\ s\phi \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.19}$$

where the u_i's, v_i's, and w_i's, $i = 1, 2, 3$, are functions of $s\theta$, $c\theta$, and known (measured or estimated) quantities. For the above system to have nonzero solutions, the determinant of the coefficient matrix must vanish, i.e.,

$$(u_1v_2 - u_2v_1)w_3 + (v_1u_3 - v_3u_1)w_2 + (u_2v_3 - u_3v_2)w_1 = 0$$

Note that the terms $u_1v_2 - u_2v_1$, $v_1u_3 - v_3u_1$, and $u_2v_3 - u_3v_2$ are again all *linear* in $x := c\theta$ and $y := s\theta$ and so are w_1, w_2, and w_3, which make the above polynomial *quadratic* in x and y [97]. Following the same elimination procedure as in Section 3.2, we arrive at a 4^{th} order polynomial in y

$$n_4y^4 + n_3y^3 + n_2y^2 + n_1y + n_0 = 0 \tag{3.20}$$

where $n_0, \ldots, n_4$ are known constants [97]. In this case, there exist at most 4 solutions for y all of which can be found in closed form. Once y is determined, back-substitution allows us to find x. Finally, $s\phi$ and $c\phi$ are retrieved by computing the null-space vector of the coefficient matrix in (3.19). □

In this lemma, we have shown that the maximum number of possible solutions is 4. We should note, however, that in most cases in practice there exists only one real solution for the robot-to-robot transformation. Specifically, even if all 4 roots of (3.20) are reals in the interval $[-1,\ 1]$ (and so will be the corresponding values for x), back-substitution in (3.19) will not always give a real value for ϕ. This is because the first two elements of the null-space vector of the coefficient matrix in (3.19) must also satisfy the trigonometric constraint $s\phi^2 + c\phi^2 = 1$. Actually, this is a general property of

over-determined nonlinear systems, which we hereafter describe in the context of the problem at hand.

Corollary 4. *Given 4 distance measurements between two robots at 4 different locations, with probability 1 (i.e., almost surely) there exists a unique solution for the 3-DOF robot-to-robot transformation.*

To verify this, consider the case where after processing the first 3 distance measurements, 6 solutions are found, denoted as $\mathbf{x}_i = \begin{bmatrix} {}^1\mathbf{p}_2^T & {}^1\phi_2 \end{bmatrix}^T$, $i = 1, \ldots, 6$. Without loss of generality, we first assume that $\mathbf{x}_1$ corresponds to the true robot-to-robot transformation. When the robots move to their new positions, denoted by frames $\{7\}$ and $\{8\}$ respectively, to record their 4^{th} distance measurement, d_{78}, then $\mathbf{x}_1$ should also satisfy (3.17):

$$({}^1\mathbf{p}_2 - {}^1\mathbf{p}_7)^T {}^1_2\mathbf{C}\, {}^2\mathbf{p}_8 - {}^1\mathbf{p}_2^T {}^1\mathbf{p}_7 - e_0 = 0 \Leftrightarrow h(\mathbf{x}_1, \boldsymbol{\vartheta}) = 0 \tag{3.21}$$

where $\boldsymbol{\vartheta} = \begin{bmatrix} {}^1\mathbf{p}_7^T & {}^2\mathbf{p}_8^T & d_{78} \end{bmatrix}^T$. Given $\mathbf{x}_1$, we denote as $\mathcal{V}_1$ the set of all values of $\boldsymbol{\vartheta}$ that satisfy (3.21). Note that $\mathcal{V}_1$ is a 4 dimensional variety.

If we now assume that there exists a 2^{nd} solution, for example $\mathbf{x}_2$, then it should also satisfy (3.17), i.e.,

$$h(\mathbf{x}_2, \boldsymbol{\vartheta}) = 0 \tag{3.22}$$

As before, given $\mathbf{x}_2$, we denote as $\mathcal{V}_2$ the set of all values of $\boldsymbol{\vartheta}$ that satisfy (3.22).

If both $\mathbf{x}_1$ and $\mathbf{x}_2$ are valid solutions, then $\boldsymbol{\vartheta}$ – which is the realization of the robots' displacement estimates and the distance measurement – must satisfy (3.21) *and* (3.22), and thus belongs to the set $\mathcal{V} = \mathcal{V}_1 \cap \mathcal{V}_2$. Note that since $\mathcal{V}$ is constrained by an additional equation (cf. (3.22)), $\mathcal{V} \subset \mathcal{V}_1$, and the dimension of $\mathcal{V}$ is smaller than that of $\mathcal{V}_1$ (Theorem 3, Ch. 9, §4 in [16]). Hence, the probability that the robots' trajectories are such that there exist two solutions is $|\mathcal{V}|/|\mathcal{V}_1|$, which is zero.[4] Following the same

[4] $|\mathcal{V}|$ is a measure of the size of the set $\mathcal{V}$. If $\mathcal{V}$ is continuous, then $|\mathcal{V}|$ is its total volume.

process, one can show that the probability of the event that 3 (or 4) solutions exist is also zero.

3.4 Determining the Relative Pose from 5 Distance Measurements: Unique Solution

We now treat the case where the robots R_1 and R_2 move again and arrive at the locations $\{9\}$ and $\{10\}$, respectively. At that point, they record their 5^{th} distance measurement $d_{9,10}$ and also have available the estimates for their positions ${}^1\mathbf{p}_9$ and ${}^2\mathbf{p}_{10}$, respectively. We will first prove that in this case at most one solution exists (Section 3.4.1) generically and then propose an efficient and robust algorithm for computing its value (Section 3.4.2).

3.4.1 Unique solution

Lemma 5. *Given 5 distance measurements between 2 robots at 5 different locations, generically, there exists at most one solution for the 3-DOF robot-to-robot transformation.*

Proof. Following the same procedure as in Section 3.2, we arrive at the following 4 equations (3 of these are the ones in (3.19) and the 4^{th} one is computed in a similar manner using the latest distance measurement):

$$u_i c\phi + v_i s\phi = w_i \quad , \quad i = 1, \ldots, 4 \tag{3.23}$$

where the u_i's, v_i's, and w_i's, are functions of $c\theta$, $s\theta$, and known quantities. Choosing all possible pairs of these 4 equations, we construct $\binom{4}{2} = 6$ systems of equations as the ones in (3.9). Employing the trigonometric constraint $x^2 + y^2 - 1 = 0$ and applying the same elimination process as in Section 3.2, we derive 6 polynomial equations each of order 6, in the unknown variable $y := s\theta$ (cf. (3.15)):

$$\xi_{6,j} y^6 + \ldots + \xi_{1,j} y + \xi_{0,j} = 0 \tag{3.24}$$

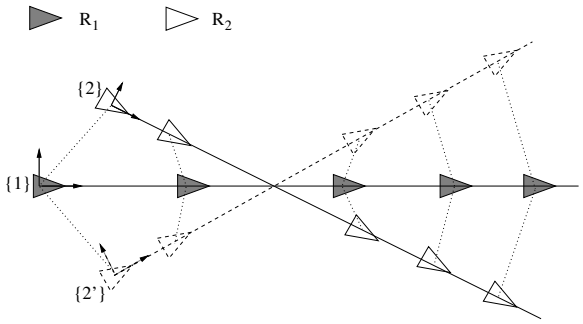


Figure 3.3: An example of a degenerate case: When both robots move on straight lines there exist two solutions for the robot-to-robot transformation (i.e., two symmetric poses for R_2 denoted by $\{2\}$ and $\{2'\}$), regardless of how many distance measurements are available. The solid line represents the true trajectory of R_2, while the dashed line denotes the symmetric solution.

where the $\xi_{i,j}$'s, $i = 0, \ldots, 6$, $j = 1, \ldots, 6$, are functions of measured and estimated quantities. Rewriting these polynomials in a matrix form, we have:

$$\begin{bmatrix} \xi_{6,1} & \cdots & \xi_{1,1} & \xi_{0,1} \\ \vdots & \ddots & \vdots & \vdots \\ \xi_{6,6} & \cdots & \xi_{1,6} & \xi_{0,6} \end{bmatrix} \begin{bmatrix} y^6 \\ \vdots \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \Longleftrightarrow \mathbf{\Xi y} = \mathbf{0} \tag{3.25}$$

The null space of matrix $\mathbf{\Xi}$ in (3.25) has, generically, dimension 1. Thus, a unique solution for the vector $\mathbf{y}$, and hence for y, can be determined. Given $y := s\theta$, the unique values of the remaining unknowns, $s\theta$, $c\phi$, and $s\phi$, are computed via back-substitution as in the previous two cases. □

Note that there exist singular cases where more than one solution exist and matrix $\mathbf{\Xi}$ will lose rank. For example, when both robots move on straight lines, two symmetric solutions exist (cf. Fig. 3.3). However, these events will not occur with probability 1 (cf. Corollary 4), and we can conclude that given 5 distance measurements, generically, there exists at most one solution for the 3-DOF robot-to-robot transformation.

3.4.2 Efficient Computation of the Unique Solution

The approach for computing the unique solution presented in the previous section, requires repetition of the elimination procedure of Section 3.2 6 times. In addition to being time consuming, this method may result in incorrect values for the robot-to-robot transformation or even fail due to the accumulation of numerical errors. In this section, we present an alternative approach based on a *linear algorithm* for efficiently computing the unique solution given 5 distance measurements.

Note that this general method for solving systems of polynomial equations has been applied in the past for computing the pose of a camera given 4 bearing measurements to known landmarks [62], [4]. However, this is the first time that the distance-based robot-to-robot transformation problem is formulated so that it can be amenable to this form of solution.

As described in Sections 3.2, 3.3, and 3.4.1, for each of the last 4 distance measurements, $d_{34}, \ldots, d_{9,10}$, we can write an equation similar to (3.7), repeated below after rearranging terms and renaming the known quantities $\alpha_{i,j}$'s:

$$\begin{aligned} &\alpha_{7,j}c\phi + \alpha_{6,j}s\phi + \alpha_{5,j}c\theta + \alpha_{4,j}s\theta + \alpha_{3,j}c(\theta-\phi) \\ &+ \alpha_{2,j}s(\theta-\phi) + \alpha_{1,j} = 0 \quad , \quad j =, 1 \ldots, 4 \end{aligned}$$

The unknowns in these 4 equations are $c\phi$, $s\phi$, $c\theta$, $s\theta$, $c(\theta-\phi)$, $s(\theta-\phi)$. Rewriting them in a matrix form, we have

$$\begin{bmatrix} \alpha_{7,1} & \ldots & \alpha_{1,1} \\ \vdots & \ddots & \vdots \\ \alpha_{7,4} & \ldots & \alpha_{1,4} \end{bmatrix} \begin{bmatrix} c\phi \\ s\phi \\ c\theta \\ s\theta \\ c(\theta-\phi) \\ s(\theta-\phi) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \Longleftrightarrow \mathbf{Ax} = \mathbf{0}$$

where $\mathbf{A}$ is the 4×7 coefficient matrix (known), and $\mathbf{x}$ is the unknown vector we want to solve for. Once we have computed the three vectors $\mathbf{r}$, $\mathbf{s}$, and $\mathbf{t}$ that span the null space of $\mathbf{A}$, $\mathbf{x}$ can be written as:

$$\mathbf{x} = \lambda_1 \mathbf{r} + \lambda_2 \mathbf{s} + \lambda_3 \mathbf{t} \tag{3.26}$$

for some scalars λ_1, λ_2, λ_3. To determine their values, we use the trigonometric identities

$$\begin{aligned} &c^2\phi + s^2\phi = 1,\ c^2\theta + s^2\theta = 1,\ c^2(\theta - \phi) + s^2(\theta - \phi) = 1 \\ &c\theta c\phi + s\theta s\phi = c(\theta - \phi),\ s\theta c\phi - c\theta s\phi = s(\theta - \phi) \end{aligned} \tag{3.27}$$

and the constraint

$$\lambda_1 r_7 + \lambda_2 s_7 + \lambda_3 t_7 = 1 \tag{3.28}$$

where r_7, s_7, and t_7 denote the 7^{th} scalar elements of vectors $\mathbf{r}$, $\mathbf{s}$, and $\mathbf{t}$, respectively. Substituting the corresponding elements of $\mathbf{x}$ from (3.26) in the constraints (3.27), and eliminating λ_3 using (3.28), we obtain the following system of equations:

$$\begin{bmatrix} \beta_{1,1} & \dots & \beta_{1,5} \\ \vdots & \ddots & \vdots \\ \beta_{5,1} & \dots & \beta_{5,5} \end{bmatrix} \begin{bmatrix} \lambda_1^2 \\ \lambda_2^2 \\ \lambda_1\lambda_2 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_5 \end{bmatrix}$$

where $\beta_{i,j}$ and γ_i, $i,\ j = 1, \dots, 5$ are functions of known quantities [97]. This system can be solved to uniquely determine the unknown vector $[\lambda_1^2\ \lambda_2^2\ \lambda_1\lambda_2\ \lambda_1\ \lambda_2]^T$. Given the values of λ_1 and λ_2, λ_3 is computed from (3.28) and subsequently $\mathbf{x}$ is uniquely determined from (3.26). Finally, the unknown robot-to-robot transformation is retrieved from the first 4 elements of $\mathbf{x}$.

3.5 Weighted Least Squares Refinement

When 5 or more distance measurements are available to the robots, their relative pose can be computed with higher accuracy based on the following two-step process: (i) Compute the initial estimate for the 3-DOF transformation from 5 distance measurements (cf. Section 3.4.2), (ii) Use the estimate from the previous step to initialize an iterative weighted least squares (WLS) algorithm that processes all distance measurements available. We hereafter describe the second step of this process.

Assume that the robots have recorded n distance measurements, which are used to form a system of $n-1$ nonlinear equations equivalent to (3.5). Rearranging terms, these can be written in a compact form as

$$\mathbf{h}(\mathbf{x}, \boldsymbol{\vartheta}) = \mathbf{0} \tag{3.29}$$

where $\mathbf{x} := [\theta, \phi]^T$ denotes the vector of the unknown variables, and $\boldsymbol{\vartheta} := [\ {}^1\mathbf{p}^T\ {}^2\mathbf{p}^T\ \mathbf{z}^T\ ]^T$ is the vector comprising the following known quantities:

$$\begin{aligned} {}^1\mathbf{p} &:= [\ {}^1\mathbf{p}_3^T\ \ldots\ {}^1\mathbf{p}_{2n-1}^T\ ]^T \\ {}^2\mathbf{p} &:= [\ {}^2\mathbf{p}_4^T\ \ldots\ {}^2\mathbf{p}_{2n}^T\ ]^T \\ \mathbf{z} &:= [\ d_{12}\ \ldots\ d_{(2n-1)(2n)}\ ]^T \end{aligned}$$

Since ${}^1\mathbf{p}$, ${}^2\mathbf{p}$, and $\mathbf{z}$ are estimated or measured independently, the covariance matrix $\mathbf{P}_{\boldsymbol{\vartheta}\boldsymbol{\vartheta}}$ of $\boldsymbol{\vartheta}$ has a block diagonal structure:

$$\mathbf{P}_{\boldsymbol{\vartheta}\boldsymbol{\vartheta}} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R} \end{bmatrix}$$

where[5] $\mathbf{P}_{11} = E[{}^1\tilde{\mathbf{p}}\,{}^1\tilde{\mathbf{p}}^T]$ and $\mathbf{P}_{22} = E[{}^2\tilde{\mathbf{p}}\,{}^2\tilde{\mathbf{p}}^T]$ are the covariance matrices for the

[5] We denote the error as $\tilde{\mathbf{p}} = \mathbf{p} - \hat{\mathbf{p}}$, where $\mathbf{p}$ and $\hat{\mathbf{p}}$ are the true and estimated vectors, respectively. Note also that the covariance matrices $\mathbf{P}_{11}$ and $\mathbf{P}_{22}$ are computed using state augmentation [68]. At every new robot position a copy of the current robot pose is added to the state vector. The covariance matrix is also appropriately augmented to include the covariance of the duplicate robot pose and its correlations with the rest of the poses in the state vector. Afterwards, only the copied state is propagated using odometric measurements. The interested reader is referred to [54] for details.

robot-position vectors ${}^1\mathbf{p}$ and ${}^2\mathbf{p}$, respectively, and $\mathbf{R} = \text{diag}(\sigma_{d_{ij}}^2)$ is the distance-measurements' noise covariance matrix, with $\sigma_{d_{ij}}$ denoting its standard deviation.

Given the estimate $\hat{\boldsymbol{\vartheta}}$ of $\boldsymbol{\vartheta}$, from the robots' odometry and the recorded distance measurements, and the initial estimate $\hat{\mathbf{x}}^{(1)}$ of $\mathbf{x}$, determined from the algebraic method of Section 3.4.2, the WLS algorithm computes the new estimate for $\mathbf{x}$ through the following iterative process:

$$\hat{\mathbf{x}}^{(\kappa+1)} = \hat{\mathbf{x}}^{(\kappa)} - \mathbf{P}_{\mathbf{xx}}\mathbf{H}_{\mathbf{x}}^T(\mathbf{H}_{\boldsymbol{\vartheta}}\mathbf{P}_{\boldsymbol{\vartheta}\boldsymbol{\vartheta}}\mathbf{H}_{\boldsymbol{\vartheta}}^T)^{-1}\mathbf{h}(\hat{\mathbf{x}}^{(\kappa)}, \hat{\boldsymbol{\vartheta}})$$

where $\mathbf{P}_{\mathbf{xx}} = [\mathbf{H}_{\mathbf{x}}^T(\mathbf{H}_{\boldsymbol{\vartheta}}\mathbf{P}_{\boldsymbol{\vartheta}\boldsymbol{\vartheta}}\mathbf{H}_{\boldsymbol{\vartheta}}^T)^{-1}\mathbf{H}_{\mathbf{x}}]^{-1}$ is the covariance of the estimates, and

$$\mathbf{H}_{\mathbf{x}} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^{(\kappa)}, \boldsymbol{\vartheta}=\hat{\boldsymbol{\vartheta}}} \quad , \quad \mathbf{H}_{\boldsymbol{\vartheta}} = \left.\frac{\partial \mathbf{h}}{\partial \boldsymbol{\vartheta}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^{(\kappa)}, \boldsymbol{\vartheta}=\hat{\boldsymbol{\vartheta}}}$$

are the Jacobians of the nonlinear function $\mathbf{h}$ [97].

Note that the iterative WLS process will converge if the robot-to-robot transformation is observable. In practice, we can detect singular cases by checking the rank of $\mathbf{H}_{\mathbf{x}}$; if it is not of rank 2, the robots will need to move to new locations and acquire additional range measurements. In order to avoid such singular configurations, in the following section we present necessary and sufficient conditions on the robots' motion for the system to be observable.

3.6 Observability Analysis

The system describing the 3-DOF robot-to-robot transformation given odometric and distance measurements is nonlinear. Therefore, tests designed for *linear time-invariant* systems (e.g., the Gramian matrix rank [51] or the Popov-Belevitch-Hautus (PBH) test [70]) cannot be used for examining its observability. Instead, we hereafter employ the observability rank criterion based on Lie derivatives [30] to determine the conditions under which our system is locally weakly observable. Recently, Mariottini *et al.* [49]

have employed this criterion to investigate the observability of 2D leader-follower formations using only *bearing* measurements. In [50], Martinelli and Siegwart have also used this test to determine *only sufficient conditions* for the observability of cooperative localization for pairs of mobile robots navigating in 2D.

In this section, we study the observability of the nonlinear system describing the time evolution of the robot-to-robot transformation given distance measurements and determine the *necessary and sufficient* observability conditions on the motion of the two robots. Specifically, after a brief review of the key concepts of observability (Section 3.6.1), the Lie derivatives and the observability rank condition (Section 3.6.2), in Section 3.6.3 we prove that the sufficient conditions for the robot-to-robot transformation to be locally weakly observable are: (i)-(ii) both robots have nonzero linear velocities, (iii) at least one of them has nonzero rotational velocity. Additionally, we prove that conditions (i)-(ii) are also necessary.

3.6.1 Observability of Nonlinear Systems

Consider the state-space representation of the following nonlinear system [12]:

$$\Sigma \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) \end{cases} \tag{3.30}$$

where $\mathbf{x} \in M$ (a C^∞ connected manifold of dimension n) is the state vector, $\mathbf{u} = [u_1 \dots u_l]^T \in \mathbb{R}^l$ is the vector of control inputs, and $\mathbf{y} = [y_1 \dots y_m]^T \in \mathbb{R}^m$ is the measurement vector, with $y_k = h_k(\mathbf{x})$, $k = 1, \dots, m$.

Definition 6. *Two initial states* $\mathbf{x}_0$ *and* $\mathbf{x}_1$ *are* indistinguishable *if given the same input* $\mathbf{u}(t)$*, the system* Σ *produces the same output* $\mathbf{y}(t)$ *for both initial states* $\mathbf{x}_0$ *and* $\mathbf{x}_1$*. The system is termed* observable *if for all* $\mathbf{x} \in M$*, the only state indistinguishable from* $\mathbf{x}$ *is* $\mathbf{x}$ *itself.*

Notice that observability is a global concept. However, it might be necessary for the state to travel a considerable distance or for a long time to distinguish two points of M.

For this reason, the following local concept is introduced [12]:

Definition 7. Σ *is* locally observable *at* $\mathbf{x}_0 \in M$ *if for every open neighborhood* U *of* $\mathbf{x}_0$*, the set of points indistinguishable from* $\mathbf{x}_0$ *by trajectories in* U *only consists of* $\mathbf{x}_0$ *itself.* Σ *is* locally observable *if it is locally observable for every* $\mathbf{x} \in M$*.*

In practice, it may be sufficient to distinguish $\mathbf{x}_0$ only from its neighbors (e.g., when some prior knowledge of $\mathbf{x}_0$ is available). However, it is possible that $\mathbf{x}_0$ is indistinguishable from states that are far away. This leads to the concept of weak observability [12].

Definition 8. Σ *is* weakly observable *at* $\mathbf{x}_0$ *if there exists an open neighborhood* U *of* $\mathbf{x}_0$ *such that the only point in* U *which is indistinguishable from* $\mathbf{x}_0$ *is* $\mathbf{x}_0$ *itself. The system* Σ *is* weakly observable *if it is weakly observable at every* $\mathbf{x} \in M$*.*

Notice again that it might be necessary to travel far away from U to distinguish two points in U. For this reason, the following local concept is introduced [12].

Definition 9. Σ *is* locally weakly observable *at* $\mathbf{x}_0$ *if there exists an open neighborhood* U *of* $\mathbf{x}_0$ *such that for every open neighborhood* V *of* $\mathbf{x}_0$ *contained in* U*, the set of points indistinguishable from* $\mathbf{x}_0$ *in* U *by trajectories in* V *is* $\mathbf{x}_0$ *itself. The system* Σ *is* locally weakly observable *if it is locally weakly observable for every* $\mathbf{x} \in M$*.*

The advantage of local weak observability over other concepts is that it has a simple algebraic test which will be described in Section 3.6.2. Furthermore, and in the context of the robots' relative pose estimation problem, if the system is locally weakly observable and 3 or more distance measurements are available, then generically given an initial estimate that is close to the true solution, the iterative algorithm described in Section 3.5 (or higher-order variants of this) will converge to it. Additionally, if a unique solution exists (cf. Lemma 5, Section 3.4), the required initial estimate can be computed in closed form (Section 3.4.2), and the system is locally observable.

3.6.2 Lie Derivatives and Nonlinear Observability

We consider the special case of system (3.30) where the process function, $\mathbf{f}$, can be separated into a summation of independent functions, each one excited by a different component of the control input vector, i.e.,

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}_0(\mathbf{x}) + \mathbf{f}_1(\mathbf{x})u_1 + \cdots + \mathbf{f}_l(\mathbf{x})u_l \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) \end{cases} \tag{3.31}$$

where $\mathbf{f}_0$ is the zero-input function of the process model.

The zeroth-order Lie derivative of any (scalar) function is the function itself, i.e., $\mathfrak{L}^0 h_k(\mathbf{x}) = h_k(\mathbf{x})$. The first-order Lie derivative of function $h_k(\mathbf{x})$ with respect to $\mathbf{f}_i$ is defined as:

$$\begin{aligned} \mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x}) &= \frac{\partial h_k(\mathbf{x})}{\partial x_1} f_{i1}(\mathbf{x}) + \cdots + \frac{\partial h_k(\mathbf{x})}{\partial x_n} f_{in}(\mathbf{x}) \\ &= \nabla h_k(\mathbf{x}) \cdot \mathbf{f}_i(\mathbf{x}) \end{aligned} \tag{3.32}$$

where $\mathbf{f}_i(\mathbf{x}) = [f_{i1}(\mathbf{x}) \ , \ldots , \ f_{in}(\mathbf{x})]^T$, ∇ represents the gradient operator, and '$\cdot$' denotes the vector inner product. Considering that $\mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x})$ is a scalar function itself, the second-order Lie derivative of $h_k(\mathbf{x})$ with respect to $\mathbf{f}_i$ is:

$$\mathfrak{L}^2_{\mathbf{f}_i} h_k(\mathbf{x}) = \mathfrak{L}^1_{\mathbf{f}_i} \left(\mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x}) \right) = \nabla \mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x}) \cdot \mathbf{f}_i(\mathbf{x}) \tag{3.33}$$

Higher-order Lie derivatives are computed similarly. Additionally, it is possible to define mixed Lie derivatives, i.e., with respect to different functions of the process model. For example the second-order Lie derivative of h_k with respect to $\mathbf{f}_j$ and $\mathbf{f}_i$, given its first derivative with respect to $\mathbf{f}_i$, is:

$$\mathfrak{L}^2_{\mathbf{f}_j \mathbf{f}_i} h_k(\mathbf{x}) = \mathfrak{L}^1_{\mathbf{f}_j} \left(\mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x}) \right) = \nabla \mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x}) \cdot \mathbf{f}_j(\mathbf{x}) \tag{3.34}$$

Based on the preceding expressions for the Lie derivatives, the observability matrix is defined as the matrix with rows:

$$\mathcal{O} \triangleq \{ \nabla \mathfrak{L}^\ell_{\mathbf{f}_i \cdots \mathbf{f}_j} h_k(\mathbf{x}) | i, j = 0 \ldots l; k = 1 \ldots m; \ell \in \mathbb{N} \} \tag{3.35}$$

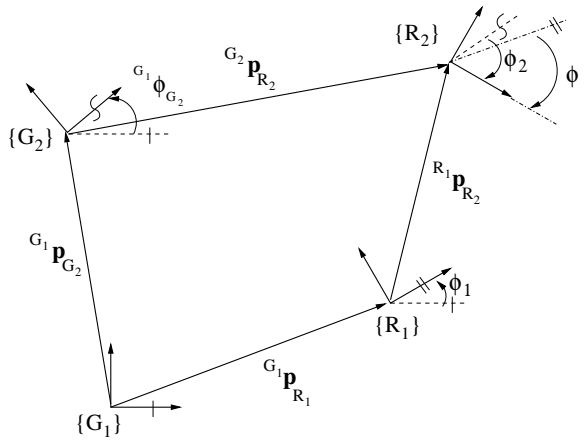


Figure 3.4: Geometric relation between the two robots. The initial (global) frames of the two robots are denoted by $\{G_1\}$ and $\{G_2\}$, while their current poses are represented by the frames $\{R_1\}$ and $\{R_2\}$, respectively.

The important role of this matrix in the observability analysis of a nonlinear system is captured by Theorems 3.1 and 3.11 in [30], repeated below:

Theorem 10 (Observability Sufficient Condition)**.** *If a system satisfies the observability rank condition then it is locally weakly observable.*

Definition 11 (Observability Rank Condition)**.** *The observability rank condition is satisfied when the observability matrix (cf.* (3.35)*) is full rank.*

Theorem 12 (Observability Necessary Condition)**.** *If a system is locally weakly observable, then the observability rank condition is satisfied generically.*

In this case, "generically" means that the observability matrix is full rank everywhere except possibly within a subset of the domain of $\mathbf{x}$ [12]. Hence, if the observability matrix is not of sufficient rank for all values of $\mathbf{x}$, the system is *not* locally weakly observable [66].

3.6.3 Observability of the 2D Robot-to-Robot Transformation

In this section, we first derive the continuous-time system model describing the relative position and orientation of the two robots and then analyze its observability, when range measurements are available.

Consider two robots that start from initial poses denoted by the (global) frames of reference $\{G_1\}$ and $\{G_2\}$, respectively. After some time, the robots move and their new poses are now depicted in Fig. 3.4 by the frames $\{R_1\}$ and $\{R_2\}$.

The 2×2 rotational matrices describing vector transformations between these frames satisfy the following relation:

$$\begin{smallmatrix} G_1 \\ G_2 \end{smallmatrix}\mathbf{C}({}^{G_1}\phi_{G_2}) = \begin{smallmatrix} G_1 \\ R_1 \end{smallmatrix}\mathbf{C}(\phi_1)\begin{smallmatrix} R_1 \\ R_2 \end{smallmatrix}\mathbf{C}(\phi)\begin{smallmatrix} R_2 \\ G_2 \end{smallmatrix}\mathbf{C}(-\phi_2)$$

or, equivalently

$${}^{G_1}\phi_{G_2} = \phi_1 + \phi - \phi_2 \tag{3.36}$$

where ϕ denotes the relative orientation of the two robots.

Additionally, their positions satisfy the following geometric constraint (cf. Fig. 3.4):

$${}^{G_1}\mathbf{p}_{G_2} = {}^{G_1}\mathbf{p}_{R_1} + \begin{smallmatrix} G_1 \\ R_1 \end{smallmatrix}\mathbf{C}(\phi_1)^{R_1}\mathbf{p}_{R_2} - \begin{smallmatrix} G_1 \\ G_2 \end{smallmatrix}\mathbf{C}({}^{G_1}\phi_{G_2})^{G_2}\mathbf{p}_{R_2} \tag{3.37}$$

where ${}^{R_1}\mathbf{p}_{R_2}$ is the robots' relative position vector.

Differentiating (3.36) and (3.37) with respect to time and noting that ${}^{G_1}\dot{\phi}_{G_2} = 0$ and ${}^{G_1}\dot{\mathbf{p}}_{G_2} = \mathbf{0}_{2\times 1}$, we have:

$$\dot{\mathbf{p}} = -\mathbf{e}_1 v_1 + \mathbf{C}(\phi)\mathbf{e}_1 v_2 - \mathbf{J}\mathbf{p}\omega_1 \tag{3.38}$$

$$\dot{\phi} = -\omega_1 + \omega_2 \tag{3.39}$$

with

$$\mathbf{p} := {}^{R_1}\mathbf{p}_{R_2} = \begin{bmatrix} p_x \\ p_y \end{bmatrix},\ \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ \mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

where v_i and $\omega_i = \dot{\phi}_i$, $i = 1, 2$, are the linear and rotational velocity, respectively, of robot R_i.

Furthermore, we rearrange the nonlinear kinematic equations (cf. (3.38) and (3.39)) in the following convenient form for computing the Lie derivatives:

$$\underbrace{\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\phi} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{f}_1} v_1 + \underbrace{\begin{bmatrix} c\phi \\ s\phi \\ 0 \end{bmatrix}}_{\mathbf{f}_2} v_2 + \underbrace{\begin{bmatrix} p_y \\ -p_x \\ -1 \end{bmatrix}}_{\mathbf{f}_3} \omega_1 + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{f}_4} \omega_2 \tag{3.40}$$

Finally in order to preserve the clarity of presentation, the measurement function is chosen to be the squared distance between the two robots divided by two, $d^2/2$, instead of the distance d, i.e.,

$$h(\mathbf{x}) = \frac{d^2}{2} = \frac{1}{2}\mathbf{p}^T\mathbf{p} \tag{3.41}$$

Note that d and $d^2/2$ are both strictly positive, there is a one-to-one correspondence between them, and provide the same information for the spatial relation of the two robots.[6]

We hereafter compute the necessary Lie derivatives of h and their gradients.

- *Zeroth-order Lie derivative* ($\mathfrak{L}^0 h$)

$$\mathfrak{L}^0 h = h = \frac{1}{2}\mathbf{p}^T\mathbf{p}$$

with gradient:

$$\nabla\mathfrak{L}^0 h = \begin{bmatrix} \mathbf{p}^T & 0 \end{bmatrix} = \begin{bmatrix} p_x & p_y & 0 \end{bmatrix}$$

- *First-order Lie derivatives* ($\mathfrak{L}^1_{\mathbf{f}_1} h$, $\mathfrak{L}^1_{\mathbf{f}_2} h$)

$$\mathfrak{L}^1_{\mathbf{f}_1} h = \nabla\mathfrak{L}^0 h \cdot \mathbf{f}_1 = -p_x$$

$$\mathfrak{L}^1_{\mathbf{f}_2} h = \nabla\mathfrak{L}^0 h \cdot \mathbf{f}_2 = p_x c\phi + p_y s\phi$$

[6] In practice, if $d^2/2$ is used as the measurement instead of d, it will change the probability density function of the noise in the observations and may affect the selection of the appropriate estimator for determining the unknown transformation. This choice, however, has no impact on the observability which is a property of the system and is independent of the observation noise.

with gradients:

$$\nabla \mathfrak{L}^1_{\mathbf{f}_1} h = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}$$
$$\nabla \mathfrak{L}^1_{\mathbf{f}_2} h = \begin{bmatrix} c\phi & s\phi & -p_x s\phi + p_y c\phi \end{bmatrix}$$

- *Second-order Lie derivatives* ($\mathfrak{L}^2_{\mathbf{f}_1\mathbf{f}_2} h$, $\mathfrak{L}^2_{\mathbf{f}_3\mathbf{f}_1} h$, $\mathfrak{L}^2_{\mathbf{f}_4\mathbf{f}_2} h$)

$$\mathfrak{L}^2_{\mathbf{f}_1\mathbf{f}_2} h = (\nabla \mathfrak{L}^1_{\mathbf{f}_2} h) \cdot \mathbf{f}_1 = -c\phi$$
$$\mathfrak{L}^2_{\mathbf{f}_3\mathbf{f}_1} h = (\nabla \mathfrak{L}^1_{\mathbf{f}_1} h) \cdot \mathbf{f}_3 = -p_y$$
$$\mathfrak{L}^2_{\mathbf{f}_4\mathbf{f}_2} h = (\nabla \mathfrak{L}^1_{\mathbf{f}_2} h) \cdot \mathbf{f}_4 = -p_x s\phi + p_y c\phi$$

with gradients:

$$\nabla \mathfrak{L}^2_{\mathbf{f}_1\mathbf{f}_2} h = \begin{bmatrix} 0 & 0 & s\phi \end{bmatrix}$$
$$\nabla \mathfrak{L}^2_{\mathbf{f}_3\mathbf{f}_1} h = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}$$
$$\nabla \mathfrak{L}^2_{\mathbf{f}_4\mathbf{f}_2} h = \begin{bmatrix} -s\phi & c\phi & -p_x c\phi - p_y s\phi \end{bmatrix}$$

- *Third-order Lie derivatives* ($\mathfrak{L}^3_{\mathbf{f}_3\mathbf{f}_1\mathbf{f}_2} h$, $\mathfrak{L}^3_{\mathbf{f}_4\mathbf{f}_1\mathbf{f}_2} h$, $\mathfrak{L}^3_{\mathbf{f}_4\mathbf{f}_4\mathbf{f}_2} h$)

$$\mathfrak{L}^3_{\mathbf{f}_3\mathbf{f}_1\mathbf{f}_2} h = (\nabla \mathfrak{L}^2_{\mathbf{f}_1\mathbf{f}_2} h) \cdot \mathbf{f}_3 = -s\phi$$
$$\mathfrak{L}^3_{\mathbf{f}_4\mathbf{f}_1\mathbf{f}_2} h = (\nabla \mathfrak{L}^2_{\mathbf{f}_1\mathbf{f}_2} h) \cdot \mathbf{f}_4 = s\phi$$
$$\mathfrak{L}^3_{\mathbf{f}_4\mathbf{f}_4\mathbf{f}_2} h = (\nabla \mathfrak{L}^2_{\mathbf{f}_4\mathbf{f}_2} h) \cdot \mathbf{f}_4 = -p_x c\phi - p_y s\phi$$

with gradients:

$$\nabla \mathfrak{L}^3_{\mathbf{f}_3\mathbf{f}_1\mathbf{f}_2} h = \begin{bmatrix} 0 & 0 & -c\phi \end{bmatrix}$$
$$\nabla \mathfrak{L}^3_{\mathbf{f}_4\mathbf{f}_1\mathbf{f}_2} h = \begin{bmatrix} 0 & 0 & c\phi \end{bmatrix}$$
$$\nabla \mathfrak{L}^3_{\mathbf{f}_4\mathbf{f}_4\mathbf{f}_2} h = \begin{bmatrix} -c\phi & -s\phi & p_x s\phi - p_y c\phi \end{bmatrix}$$

Any other higher-order Lie derivatives are repetitions of $\pm c\phi$, $\pm s\phi$, $\pm(p_x c\phi + p_y s\phi)$, and $\pm(p_x s\phi - p_y c\phi)$ [97].

At this point, we present the main results of this section regarding the observability of the system under consideration:

Lemma 13 (Sufficient Conditions)**.** *The system* (3.40)-(3.41) *is locally weakly observable, if the following conditions are satisfied:*

(i) $v_1 \neq 0$,

(ii) $v_2 \neq 0$,

(iii) $\omega_1 \neq 0$ *or* $\omega_2 \neq 0$,

Proof. When conditions (i)-(iii) are satisfied, and particularly $\omega_1 \neq 0$, the observability matrix

$$\mathcal{O} = \begin{bmatrix} \nabla \mathfrak{L}^1_{\mathbf{f}_1} h \\ \nabla \mathfrak{L}^2_{\mathbf{f}_3\mathbf{f}_1} h \\ \nabla \mathfrak{L}^2_{\mathbf{f}_1\mathbf{f}_2} h \\ \nabla \mathfrak{L}^3_{\mathbf{f}_3\mathbf{f}_1\mathbf{f}_2} h \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & s\phi \\ 0 & 0 & -c\phi \end{bmatrix}$$

is full rank, because $\mathcal{O}^T\mathcal{O} = \mathbf{I}_3$. Alternatively, if $\omega_1 = 0$ and $\omega_2 \neq 0$, we replace the rows that involve $\mathbf{f}_3$ with those corresponding to $\mathbf{f}_4$. Then the observability matrix $\mathcal{O}$ is

$$\mathcal{O} = \begin{bmatrix} \nabla \mathfrak{L}^1_{\mathbf{f}_2} h \\ \nabla \mathfrak{L}^2_{\mathbf{f}_4\mathbf{f}_2} h \\ \nabla \mathfrak{L}^2_{\mathbf{f}_1\mathbf{f}_2} h \\ \nabla \mathfrak{L}^3_{\mathbf{f}_4\mathbf{f}_1\mathbf{f}_2} h \end{bmatrix} = \begin{bmatrix} c\phi & s\phi & -p_x s\phi + p_y c\phi \\ -s\phi & c\phi & -p_x c\phi - p_y s\phi \\ 0 & 0 & s\phi \\ 0 & 0 & c\phi \end{bmatrix}$$

which is also full rank, since $\det(\mathcal{O}^T\mathcal{O}) = 1$. Therefore, in both cases, the observability rank condition is satisfied and the system is locally weakly observable (cf. Theorem 10). □

Lemma 14 (Necessary Conditions)**.** *The system* (3.40)-(3.41) *is* not *locally weakly observable, if the following conditions are* not *satisfied:*

(i) $v_1 \neq 0$,

(ii) $v_2 \neq 0$.

Proof. We hereafter show that when any of the above two conditions is violated, the observability matrix is never full rank, and thus (cf. Theorem 12) the system is not locally weakly observable.

(i) If $v_1 = 0$, we can only select Lie derivatives that do not involve $\mathbf{f}_1$. In this case, the observability matrix is:

$$\mathcal{O}_1 = \begin{bmatrix} \nabla \mathfrak{L}^0 h \\ \nabla \mathfrak{L}^1_{\mathbf{f}_2} h \\ \nabla \mathfrak{L}^2_{\mathbf{f}_4\mathbf{f}_2} h \end{bmatrix} \begin{bmatrix} p_x & p_y & 0 \\ c\phi & s\phi & -p_x s\phi + p_y c\phi \\ -s\phi & c\phi & -p_x c\phi - p_y s\phi \end{bmatrix}$$

which is singular since $\det(\mathcal{O}_1) = 0$.

(ii) If $v_2 = 0$, we can only choose Lie derivatives that do not involve $\mathbf{f}_2$. The observability matrix is now

$$\mathcal{O}_2 = \begin{bmatrix} \nabla \mathfrak{L}^0 h \\ \nabla \mathfrak{L}^1_{\mathbf{f}_1} h \\ \nabla \mathfrak{L}^2_{\mathbf{f}_3\mathbf{f}_1} h \end{bmatrix} \begin{bmatrix} p_x & p_y & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

which is also singular.

Hence, it is necessary that both linear velocities v_1 and v_2 are nonzero for the robot-to-robot relative pose to be locally weakly observable. □

Corollary 15. *Condition (iii) $\omega_1 \neq 0$ or $\omega_2 \neq 0$ is not necessary for the system* (3.40)-(3.41) *to be locally weakly observable.*

Proof. If $\omega_1 = \omega_2 = 0$, then the relative orientation will remain constant, i.e., $\phi = \phi_0$, and the system equation becomes:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\phi} \end{bmatrix} = \underbrace{\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{f}_1} v_1 + \underbrace{\begin{bmatrix} c\phi_0 \\ s\phi_0 \\ 0 \end{bmatrix}}_{\mathbf{f}_2} v_2 \tag{3.42}$$

which is a special case of (3.40). The gradients of the nonzero Lie derivatives are:

$$\nabla \mathfrak{L}^0 h = \begin{bmatrix} p_x & p_y & 0 \end{bmatrix}$$
$$\nabla \mathfrak{L}^1_{\mathbf{f}_1} h = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}$$
$$\nabla \mathfrak{L}^1_{\mathbf{f}_2} h = \begin{bmatrix} c\phi_0 & s\phi_0 & -p_x s\phi_0 + p_y c\phi_0 \end{bmatrix}$$
$$\nabla \mathfrak{L}^2_{\mathbf{f}_1\mathbf{f}_2} h = \begin{bmatrix} 0 & 0 & s\phi_0 \end{bmatrix}$$

The observability matrix is now given by

$$\mathcal{O}_3 = \begin{bmatrix} -1 & 0 & 0 \\ p_x & p_y & 0 \\ 0 & 0 & s\phi_0 \\ c\phi_0 & s\phi_0 & -p_x s\phi_0 + p_y c\phi_0 \end{bmatrix}$$

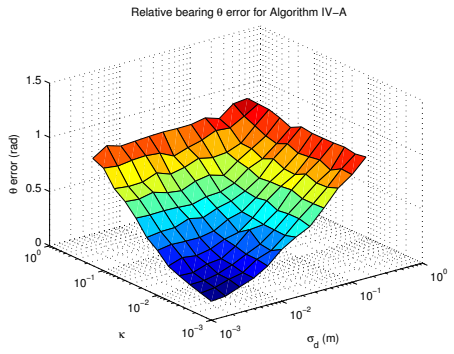
which is full rank in general. Therefore the system is still locally weakly observable (cf. Theorem 10). Hence, (iii) $\omega_1 \neq 0$ or $\omega_2 \neq 0$ is a sufficient but *not* a necessary condition for the system to be locally weakly observable. □

Finally, we note that Lemmas 13 and 14 provide sufficient and necessary conditions on the robots' motions for the 3-DOF transformation to be *locally weakly* observable. If, additionally, 5 distance measurements are available (cf. Lemma 5), then generically the system is *locally* observable and the robot-to-robot transformation can be uniquely determined.

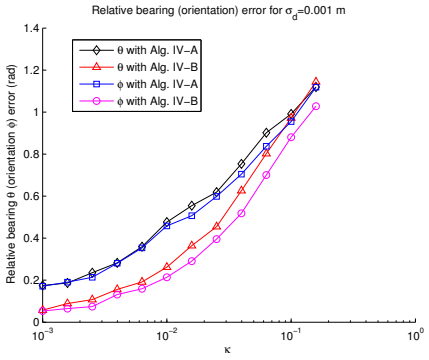
3.7 Simulation and Experimental Results

3.7.1 Simulations

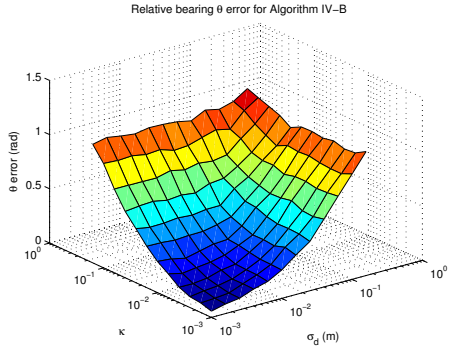
The purpose of our simulations is to verify the validity of the presented algorithms, and demonstrate the accuracy and robustness of Algorithm 3.4.2 (cf. Section 3.4.2) against that of Algorithm 3.4.1 (cf. Section 3.4.1) for computing the relative pose of two robots using 5 distance measurements.



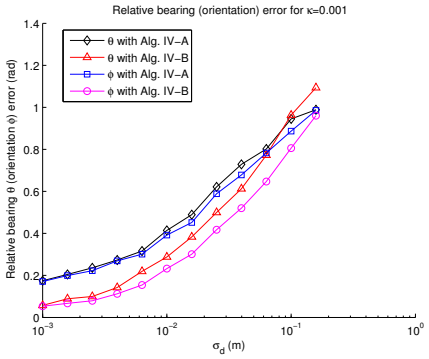
(a) Relative bearing error for IV-A



(b) Relative bearing and orientation error for $\sigma_d = 1$ mm



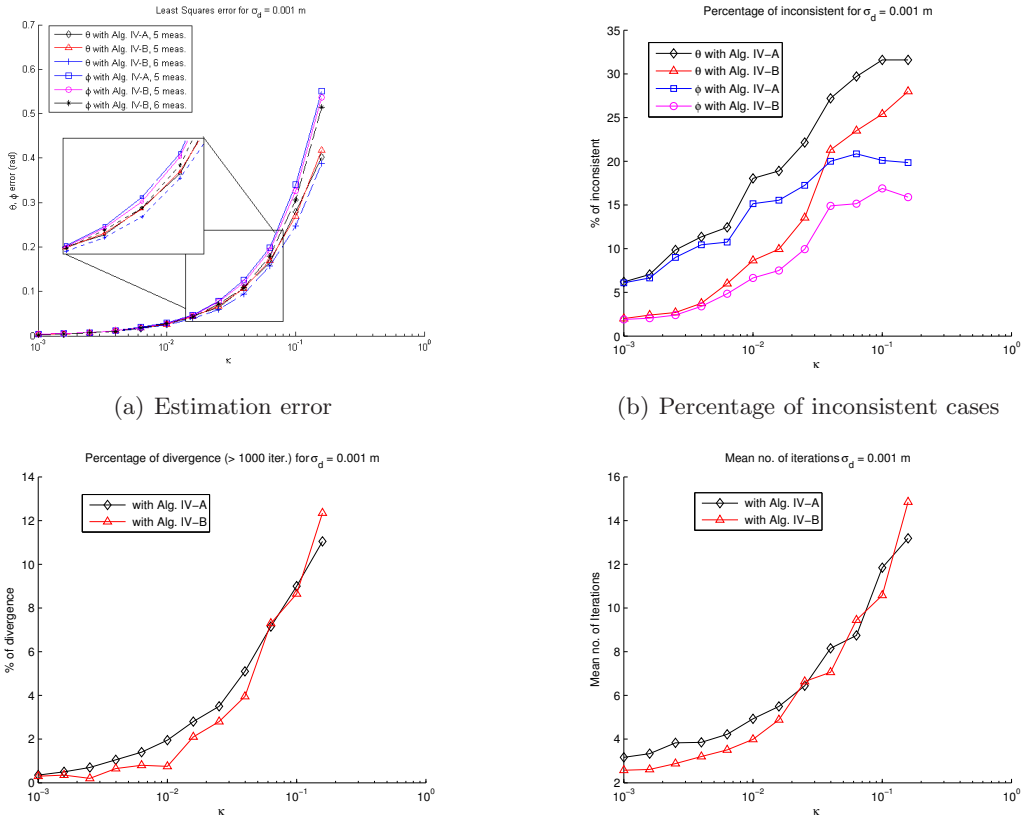
(c) Relative bearing error for IV-B



(d) Relative bearing and orientation error for $\kappa = 0.1\%$

Figure 3.5: Simulation results, averaged over 2000 trials, for Algorithms IV-A and IV-B. (a), (c): relative bearing θ estimation error vs. standard deviation of the noise in the odometry and distance measurements. (b): relative bearing θ and relative orientation ϕ estimation errors vs. standard deviation of the odometry noise (for constant distance noise standard deviation). (d): relative bearing θ and relative orientation ϕ estimation errors vs. standard deviation of the distance noise (for constant odometry noise standard deviation).

For the results shown hereafter, the trajectories and distance measurements were generated as follows: (i) the two robots start at initial positions 10 m apart from each other and record their first distance measurement; (ii) each robot moves randomly for approximately 2 m; (iii) the robots record a distance measurement at their new positions. Steps (ii) and (iii) were repeated until 6 distance measurements were collected.



(a) Estimation error

(b) Percentage of inconsistent cases

(c) Percentage of diverged cases

(d) Mean number of iterations

Figure 3.6: Performance of the Weighted Least Squares refinement step when using Algorithms IV-A and IV-B for initialization. (a) Final θ and ϕ estimation errors for the cases that converge to the true solution (consistent). (b) Percentage of cases which converge to local minima (inconsistent). (c) Percentage of cases where the maximum number of iterations (1000) is reached. (d) Average number of iterations required to convergence.

Each robot is modeled as a differential two-wheel drive vehicle equipped with encoders measuring the left, v_{ℓ_i}, and right, v_{r_i}, wheel velocities, $i = 1,\ 2$. The linear and rotational velocity of each robot is given by:

$$v_i = \frac{v_{\ell_i} + v_{r_i}}{2}, \quad \omega_i = \frac{v_{r_i} - v_{\ell_i}}{a}, \qquad i = 1,\ 2$$

where $a = 0.35$ m is the distance between the wheels.

The wheel velocities are commanded to yield constant linear velocity $v_1 = v_2 = 1$ m/s,

and rotational velocity ω_i, $i = 1,\ 2$, that varies between ± 2.8 rad/s following a uniform distribution. The measured wheel velocities are corrupted with zero-mean Gaussian noise with standard deviations $\sigma_{v_{\ell_i}} = \kappa v_{\ell_i}$, and $\sigma_{v_{r_i}} = \kappa v_{r_i}$, $i = 1,\ 2$. The distance measurement noise is also assumed to be additive zero-mean Gaussian with covariance $\mathbf{R} = \sigma_d^2 \mathbf{I}_n$. Both σ_d and κ (velocity noise standard deviation as percentage of the actually velocity) were used as parameters in our simulations for examining the robustness and accuracy of the presented algorithms.

Fig. 3.5 shows the averaged results of 2000 trials for each noise level. In particular, Fig.s 3.5(a) and 3.5(c) show the relative bearing θ error as a function of the noise standard deviation in the odometry and the distance measurements. The corresponding figures for the relative orientation ϕ error are similar to the ones for bearing and are omitted due to space limitations. To further highlight the difference in performance of the two algorithms, we fixed the standard deviation of the noise in the distance (odometry) measurements, and plotted the estimation error as a function of the odometry (distance) noise standard deviation in Fig. 3.5(b) (Fig. 3.5(d)). Evidently, the linear Algorithm 3.4.2 is consistently more accurate than Algorithm 3.4.1 over a wide range of values for κ and σ_d.

As explained in Section 3.5, in order to improve the accuracy of the relative pose estimates, a weighted least squares (WLS) refinement step is also necessary. However, every iterative method requires an initial estimate, whose accuracy greatly impacts the quality of the solution. Imprecise initial estimates can lead to local minima and even divergence. Fig. 3.6 compares the WLS performance when using the resulting estimates from Algorithms 3.4.1 and 3.4.2 for initialization. Fig. 3.6(a) shows the estimation errors when the WLS process *converged* to the true solution and the estimates are *consistent*. As evident, the estimation errors are comparable regardless of which algorithm is used for initialization. However, the impact of the initialization algorithm on the WLS's performance is better appreciated when considering additional factors such as: (i) percentage of inconsistent estimates, (ii) percentage of divergence, and (iii) required

number of iterations. Specifically, the percentage of inconsistent estimates[7] is significantly larger when Algorithm 3.4.1, instead of 3.4.2 is used for WLS initialization (cf. Fig. 3.6(b)). Furthermore, as shown in Fig.s 3.6(c) and 3.6(d), initialization with Algorithm 3.4.2 reduces the number of WLS iterations required to converge and improves the divergence rate (we declare that the WLS process has not converged if it has reached 1000 iterations). Finally, we note that the estimation accuracy is significantly improved when additional distance measurements are available (cf. Fig 3.6(a)).

3.7.2 Experiments

We hereafter experimentally validate the accuracy of the algorithms presented in Sections 3.2-3.4 for determining the robot-to-robot transformation given 3, 4, or 5 range measurements.

For our experiments we deployed two Pioneer II robots within an area of 4 m $\times$ 5 m (cf. Fig. 3.7). The robots estimated their poses with respect to their initial locations using linear and rotational velocity measurements from their wheel-encoders. An overhead camera was used to provide ground truth for evaluating the errors in the computed estimates. Additionally, using the position measurements from the camera, we were able to compute the distances between the robots and control their accuracy by adding noise in these measurements. In this experiment, the standard deviation of the noise in the distance measurements was set to $\sigma_d = 0.01$ m.

Given 3 distance measurements, the 6 possible robot-to-robot transformations are shown in Fig. 3.8. By comparing each of them to the true robot trajectory (cf. Fig. 3.7), it is evident that Solution 4 (cf. Fig. 3.8(d)) corresponds to the true configuration. The same conclusion can be reached after processing 4 or 5 distance measurements (in this case, there exists only one real solution when considering 4 distance measurements).

The computed values for the two robot's relative bearing and orientation are shown

[7] An estimate is considered inconsistent if the estimation error is larger than 3σ, where σ is obtained from the diagonal elements of the covariance matrix $\mathbf{P_{xx}}$ computed using the iterative WLS algorithm (cf. Section 3.5).

in the first three columns of Table 3.1 for 3, 4, and 5 distance measurements respectively. Note that for the case of 3 distance measurements, only the solution closest to the true value (shown in the last column and computed using the camera) is included. These data show a slight decrease in accuracy when using additional distance measurements. This is to be expected since the distance measurements are used here to compute the *initial* estimate, at time $t = 0$, for the robot-to-robot transformation, i.e., when the two robots first detected each other. In this case, as the robots move and accumulate odometry error, the uncertainty in the initial transformation will increase. However, this would not be in general the case if instead we were to compute the robot-to-robot transformation at subsequent time instants. Actually, as it was shown in Section 3.6, the system describing the *current* robot-to-robot transformation is locally observable and hence its error will remain bounded.

Finally, we tested the accuracy of the WLS process initialized with the estimate computed by Algorithm 3.4.2 given 5 distance measurements. As shown in the 4^{th} column of Table 3.1, the WLS estimates for the relative bearing and orientation are $\theta = -0.9319$ rad, and $\phi = 0.3328$ rad, both of which are of higher accuracy compared to the initial estimates.

Table 3.1: Results with 3, 4, 5 distance measurements

No. meas.	3	4	5	5 (WLS)	Cam
θ (rad)	-0.9490	-0.9469	-0.9501	-0.9319	-0.9160
ϕ (rad)	0.3312	0.3286	0.3353	0.3328	0.3280

3.8 Summary

In this chapter, we presented efficient algorithms for solving the relative pose problem for pairs of robots moving on a plane using robot-to-robot distance measurements. Non-iterative algorithms for computing the initial estimate of the 3-DOF transformation

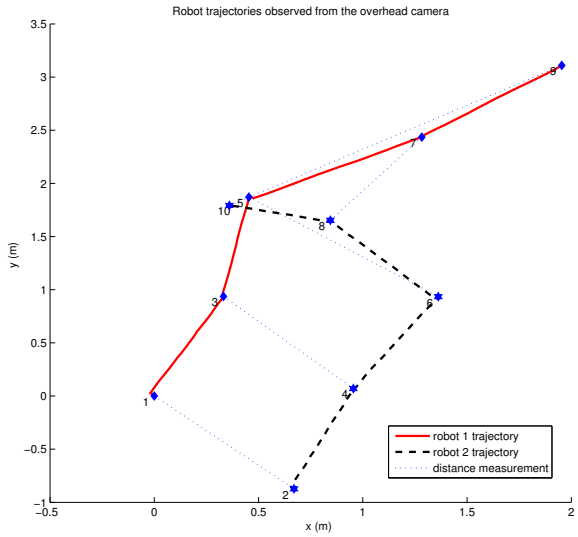
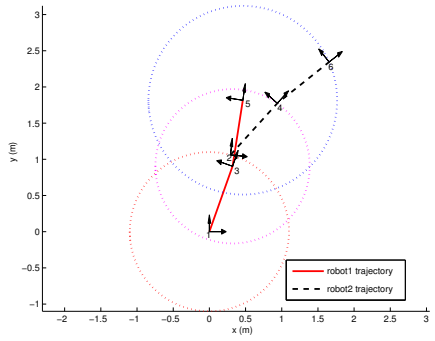
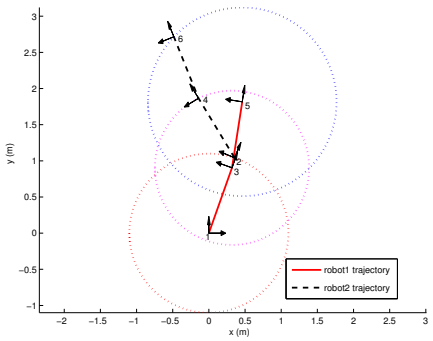


Figure 3.7: The trajectories of the two robots and the locations where distance measurements were recorded.

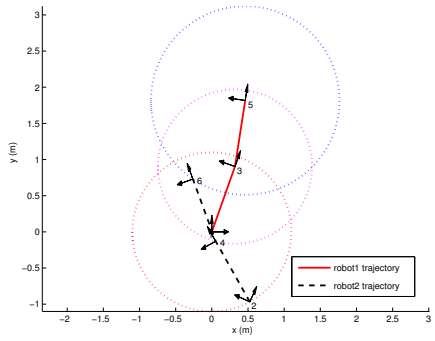
were presented for the cases when 3, 4, and 5 distance measurements were available. We have shown that for nonsingular configurations the maximum number of solutions for the above cases are 6, 4, and 1, respectively. In addition, we presented a novel linear algorithm for computing the unique solution that is robust to numerical errors. A key advantage of our approach is that it does not require any robot coordination or specific motion strategies, thus reducing the time and effort required. Furthermore, a weighted least squares process was presented that uses the result of the non-iterative algorithm as the initial relative pose estimate and iteratively improves its accuracy. Finally, we presented necessary and sufficient observability conditions for the motion of the robots based on Lie derivatives.



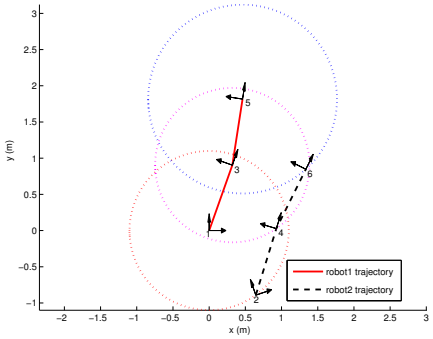
(a) Solution 1



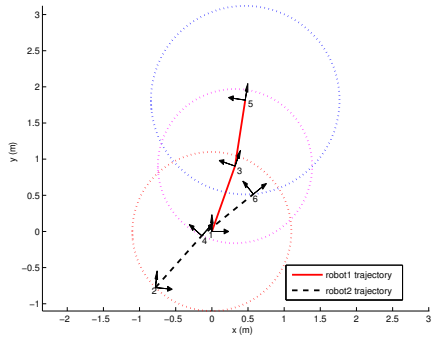
(b) Solution 2



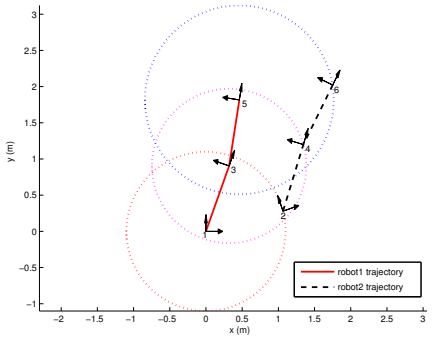
(c) Solution 3



(d) Solution 4



(e) Solution 5



(f) Solution 6

Figure 3.8: An experiment with 6 real solutions. Solution 4 is the true robot configuration. The distance measurements are depicted by circles centered at robot R_1 with radii equal to the distance to robot R_2 at each location.

Chapter 4

Motion-induced Extrinsic Calibration in 3D

In this chapter, we present methods for determining the relative robot-to-robot transformation using combinations of inter-robot range and bearing measurements. Parts of this chapter have been published in two conference papers [98, 99] and submitted as a journal paper [100].

4.1 Introduction

In order for robots to cooperate efficiently, they need to share information about their surroundings. Their measurements, however, are registered with respect to each robot's local reference frame and need to be converted to a common frame before they can be fused. This requires knowledge of the robot-to-robot transformation, i.e., their relative position and orientation. To date, only few works describe how this transformation can be determined.

Manually measuring the relative pose between robots is obviously too tedious and time consuming. Alternatively, *external references* (e.g., GPS, compass, or maps of

the environment) can be used to determine their relative pose. However, these external references are not always available (e.g., underwater, underground, outer space, or indoors).

In the absence of external references, the relative robot-to-robot transformation can be computed using *inter-robot observations*, i.e., robot-to-robot distance and/or bearing measurements. For the case of a *static* sensor network,numerous methods have been proposed for determining the locations of sensors in 2D using distance measurements between neighboring sensors (e.g., [21]). However, for mobile sensors, the problem of relative pose determination has only been studied thoroughly in 2D. The ability to move and collect measurements from different vantage points provides additional information for localizing the sensors, and hence makes their relative pose observable [50]. Specifically, it is known that mutual distance *and* bearing measurements between two robots from a single vantage point are sufficient for determining the 3-DOF relative transformation in closed-form [92, 37]. However, when only distance or bearing measurements are available, the robots must move and record additional observations. Then the relative robot pose can be found by combining the estimated robot motion (e.g., from odometry) with the mutual bearing [50] or distance [93] measurements.

In contrast to the case of motion in 2D, the more challenging problem of determining relative pose in 3D has received limited attention. Interestingly, the task of relative-pose estimation using only 6 distance measurements is actually equivalent to the forward-kinematics problem of the general Stewart-Gough platform [75]. This problem has 40 (generally complex) solutions [88], which can be found by solving a system of multivariate polynomial equations [45, 84]. Moreover, in our previous work [86] we presented methods for estimating the robots' relative pose when both robots measure relative distance and bearing, or bearing only. However, to the best of our knowledge, no algorithms exist for determining 3D relative pose using different *combinations* of robot-to-robot distance and/or bearing measurements over time (e.g., the robots can measure distance at the first time step, bearing at the second time step, etc.).

In this chapter, we focus on solving *minimal* problems where the number of equations, provided by the inter-robot measurements, equals the number of unknown parameters. In particular, we consider the case where the robots are equipped with different types of sensors, or record different types of relative measurements over time due to environment constraints. Such minimal problems are formulated as systems of polynomial equations which, in general, have multiple (complex) solutions. The solutions to these minimal problems are very useful in practice for two reasons: i) in the presence of measurement outliers, using minimal solvers as hypotheses generators minimizes the number of samples required in an outlier-rejection scheme such as Random Sample Consensus (RANSAC) [26], ii) minimal solvers can be used to initialize an iterative process [e.g., nonlinear weighted least squares (NWLS)] for improving the estimation accuracy when additional measurements are available. Note also that an additional advantage of NWLS is that it accounts for the effect of uncertainty and noise in the available motion estimates and inter-robot measurements, respectively.

The main contributions of this work are twofold:

- We identify 14 base minimal systems and prove that all other problems resulting from different combinations of inter-robot measurements can be solved using the solutions of the base systems.
- We determine the number of solutions of the 14 minimal systems and provide closed-form or efficient analytical solutions.

4.2 Problem Formulation

Consider two robots R_1 and R_2 moving randomly[1] in 3D through a sequence of poses $\{1\}, \{3\}, \ldots, \{2n-1\}$ for R_1, and $\{2\}, \{4\}, \ldots, \{2n\}$ for R_2 (see Fig. 4.1). Along their

[1] As it will become evident later on, the coefficients of the polynomials describing the geometric relation between the robots' poses depend on their trajectories and one can use this fact to enforce simplifications by appropriately restricting the robots' motions. In this work, however, we are interested in the most general and challenging case, where the robots are allowed to follow arbitrary trajectories.

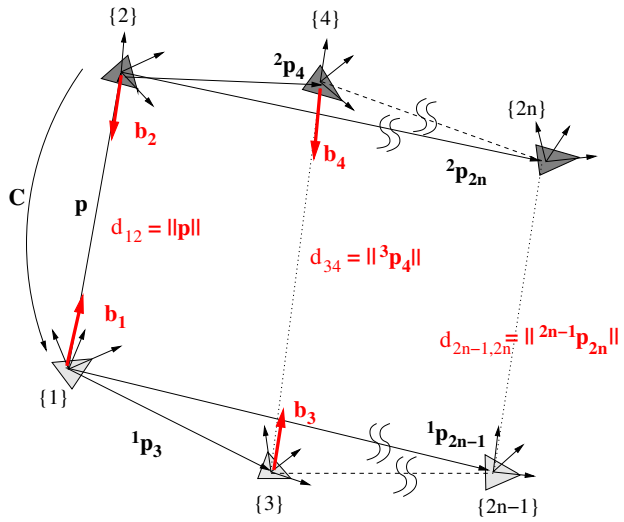


Figure 4.1: Geometry of the robot trajectories. The odd (even) numbered frames of reference depict the consecutive poses of robot R_1 (R_2). The distance between the robot poses $\{i\}$ and $\{j\}$ is denoted by d_{ij}, $i \in \{1, 3, \ldots, 2n-1\}$, $j \in \{2, 4, \ldots, 2n\}$. $\mathbf{b}_i$ ($\mathbf{b}_j$) is a unit vector pointing from $\{i\}$ to $\{j\}$ ($\{j\}$ to $\{i\}$) expressed in the initial frame $\{1\}$ ($\{2\}$). The objective is to determine the transformation between the robots' initial frames $\{1\}$ and $\{2\}$, parameterized by the translation vector $\mathbf{p}$ and the rotation matrix $\mathbf{C}$.

trajectories, the robots estimate their positions, ${}^1\mathbf{p}_i$ and ${}^2\mathbf{p}_j$, $i \in \{1, 3, ..., 2n-1\}$, $j \in \{2, 4, ..., 2n\}$, with respect to their initial frames, as well as their orientations, represented by the rotation matrices ${}^1_i\mathbf{C}$ and ${}^2_j\mathbf{C}$, respectively (e.g., by integrating linear and rotational velocity measurements over time). Additionally, at time-step t_n when robots R_1 and R_2 reach poses $\{i = 2n-1\}$ and $\{j = 2n\}$, respectively, each robot can measure the range and/or bearing towards the other robot. The range between the robots is given by $d_{ij} = ||{}^i\mathbf{p}_j||_2$, and the bearing is described by a unit vector expressed in the current local frame, ${}^i\mathbf{b}_j$ for robot R_1 and ${}^j\mathbf{b}_i$ for robot R_2. Later on, we will also need these unit vectors expressed in the robots' initial frames and, thus, we define $\mathbf{b}_i := {}^1_i\mathbf{C}\,{}^i\mathbf{b}_j$ and $\mathbf{b}_j := {}^2_j\mathbf{C}\,{}^j\mathbf{b}_i$. At each time step, the two robots can measure a *subset* of these measurements: $\{d_{ij}, \mathbf{b}_i, \mathbf{b}_j\}$.

Our goal is to use the ego-motion estimates and the relative pose measurements to

	t_1	t_2	t_3	t_4	t_5	t_6	r
1	$d_{12}, \mathbf{b}_1, \mathbf{b}_2$	d_{34}					2
2	$\mathbf{b}_1, \mathbf{b}_2$	$\mathbf{b}_3$					2
3	$d_{12}, \mathbf{b}_1$	$d_{34}, \mathbf{b}_3$					∞
4	$d_{12}, \mathbf{b}_1$	$d_{34}, \mathbf{b}_4$					∞
5	$\mathbf{b}_1, \mathbf{b}_2$	d_{34}	d_{56}				4
6	$d_{12}, \mathbf{b}_1$	$\mathbf{b}_3$	d_{56}				4
7	$d_{12}, \mathbf{b}_1$	$\mathbf{b}_4$	d_{56}				4
8	$\mathbf{b}_1$	$\mathbf{b}_3$	$\mathbf{b}_5$				8
9	$\mathbf{b}_1$	$\mathbf{b}_3$	$\mathbf{b}_6$				8
10	$d_{12}, \mathbf{b}_1$	d_{34}	d_{56}	d_{78}			8
11	$\mathbf{b}_1$	$\mathbf{b}_3$	d_{56}	d_{78}			16
12	$\mathbf{b}_1$	$\mathbf{b}_4$	d_{56}	d_{78}			16
13	$\mathbf{b}_1$	d_{34}	d_{56}	d_{78}	$d_{9,10}$		28
14	d_{12}	d_{34}	d_{56}	d_{78}	$d_{9,10}$	$d_{11,12}$	40

Table 4.1: 14 minimal problems. Under column t_i are measurements recorded at time step i. The last column (under r) shows the number of solutions to each system. System 14 is not covered in this chapter, because it is addressed in [88, 45, 84].

determine the 6-DOF initial transformation between the two robots, i.e., their relative position $\mathbf{p} := {}^1\mathbf{p}_2$ and orientation $\mathbf{C} := {}^1_2\mathbf{C}$. In this chapter, we only focus on solving the minimal problems where the number of measurement constraints equals the number of unknowns. In what follows, we will show that only the 14 systems listed in Fig. 4.1 need to be considered, while all other combinations of inter-robot measurements result into problems equivalent to these 14 systems.

4.3 All Possible Minimal Problems

We start by noting that there are 7 possible combinations of inter-robot measurements at each time step: $\{d_{ij}, \mathbf{b}_i, \mathbf{b}_j\}$, $\{\mathbf{b}_i, \mathbf{b}_j\}$, $\{d_{ij}, \mathbf{b}_i\}$, $\{d_{ij}, \mathbf{b}_j\}$, $\{\mathbf{b}_i\}$, $\{\mathbf{b}_j\}$, $\{d_{ij}\}$, and at most 6 time steps need to be considered if, e.g., when only one distance measurement is recorded at each time step. This naive analysis will give us 7^6 possible cases to consider. Fortunately, we can reduce this number significantly by considering only

minimal problems and using problem equivalence based on the following lemma.

Lemma 16. *One instance of the relative pose problem can be transformed to an equivalent problem by the following two operations:*

1. *Changing the order of the robots.*

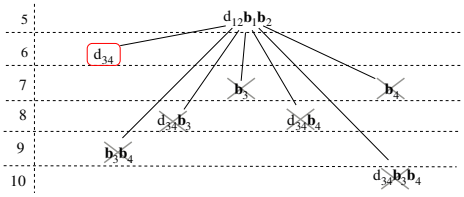
2. *Changing the order of the measurements taken.*

Proof. In order to establish problem equivalence, we here demonstrate how to use the solution of the transformed problem (i.e., when the order of the robots or measurements has changed) to solve the original problem (i.e., determine ${}^1_2\mathbf{C}$, ${}^1\mathbf{p}_2$).

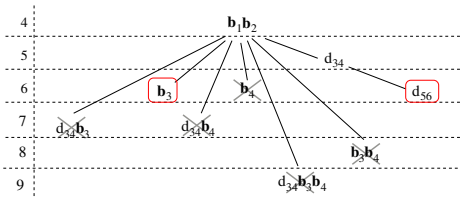
First, if we exchange the order of the robots, i.e., rename robot R_2 as R_1 and vice versa, the solution of the transformed problem is $({}^2_1\mathbf{C}, {}^2\mathbf{p}_1)$. Therefore, the solution of the original system is computed from the inverse transformation: ${}^1_2\mathbf{C} = {}^2_1\mathbf{C}^T$, ${}^1\mathbf{p}_2 = -{}^1_2\mathbf{C}^2\mathbf{p}_1$.

Exchanging the order of inter-robot measurements will only make a difference to the problem formulation when the swapping involves measurements recorded at the first time step, since the unknown variables are the 6-DOF initial robot-to-robot transformation. Without loss of generality, assume that measurements taken at the first and second time steps are swapped. Then the solution of the transformed system is actually the transformation $({}^3_4\mathbf{C}, {}^3\mathbf{p}_4)$ between the frames of reference $\{3\}$ and $\{4\}$ of the original system. The solution of the original system can then be computed using: ${}^1_2\mathbf{C} = {}^1_3\mathbf{C}^3_4\mathbf{C}^2_4\mathbf{C}^T$, and ${}^1\mathbf{p}_2 = {}^1\mathbf{p}_3 + {}^1_3\mathbf{C}^3\mathbf{p}_4 - {}^1_2\mathbf{C}^2\mathbf{p}_4$. □

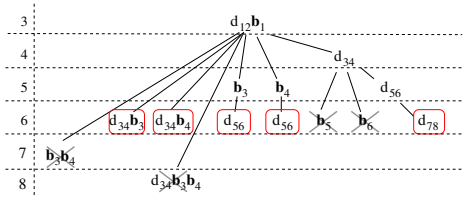
Now we will describe why we only need to consider 14 minimal systems. First of all, we are looking for combinations of measurements that provide 6 equations to determine the 6-DOF transformation, since we are only interested in minimal systems where the number of equations equals the number of unknowns. A distance measurement provides one equation, and a bearing measurement provides two. So we will collect measurements



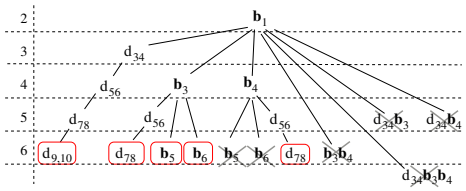
(a) Expansions providing more than 6 constraints are removed.



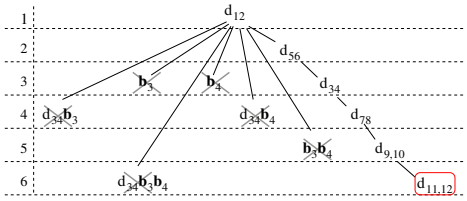
(b) Expansion $\mathbf{b}_4$ is removed since it is equivalent to $\mathbf{b}_3$ by exchanging the order of robots.



(c) The two combinations $\{d_{12}, \mathbf{b}_1; d_{34}; \mathbf{b}_5\}$ and $\{d_{12}, \mathbf{b}_1; d_{34}; \mathbf{b}_6\}$ are removed since they are equivalent to $\{d_{12}, \mathbf{b}_1; \mathbf{b}_3; d_{56}\}$ and $\{d_{12}, \mathbf{b}_1; \mathbf{b}_4; d_{56}\}$ by changing the order of measurements.



(d) $\{\mathbf{b}_1; \mathbf{b}_4; \mathbf{b}_5\}$ and $\{\mathbf{b}_1; \mathbf{b}_4; \mathbf{b}_6\}$ are equivalent to $\{\mathbf{b}_1; \mathbf{b}_3; \mathbf{b}_6\}$ by changing the order of measurements, or by changing the order of robots and order of measurements. Other nodes: $\{d_{34}, \mathbf{b}_3, \mathbf{b}_4\}$, $\{\mathbf{b}_3, \mathbf{b}_4\}$, $\{d_{34}, \mathbf{b}_3\}$ and $\{d_{34}, \mathbf{b}_4\}$, are removed since they are considered in (a), (b), and (c) as root nodes.



(e) The only case we need is the distance-only case. All other cases have been considered in previous expansions.

Figure 4.2: Measurement expansion trees. The numbers on the left of each graph denote the number of constraints provided by the measurements. Since we are only interested in minimal systems, all nodes having more than 6 constraints are removed. The leaf nodes marked by "X" are the ones removed. The nodes marked with red boxes are the 14 base systems. Note that inside { }, the measurements recorded at the same (consecutive) time step are separated by a comma (semicolon).

until we accumulate 6 constraints. To keep track of these combinations, we use an expansion tree (see Fig. 4.2) and prune its branches using Lemma 16.

At the first time step, we can exclude $\{\mathbf{b}_2\}$ and $\{d_{12}, \mathbf{b}_2\}$ from the 7 combinations by changing the order of the robots. Hence, we only need to expand 5 sets of measurements: $\{d_{12}, \mathbf{b}_1, \mathbf{b}_2\}$, $\{\mathbf{b}_1, \mathbf{b}_2\}$, $\{d_{12}, \mathbf{b}_1\}$, $\{\mathbf{b}_1\}$, $\{d_{12}\}$. We will discuss each one of them in the following:

(a) Starting from $\{d_{12}, \mathbf{b}_1, \mathbf{b}_2\}$ we only need to include $\{d_{34}\}$, since all other choices are overdetermined systems [see Fig. 4.2(a)].

(b) From $\{\mathbf{b}_1, \mathbf{b}_2\}$ we need to consider two cases: $\{\mathbf{b}_3\}$, and $\{d_{34}\}$. Besides removing overdetermined systems, we can also remove $\{\mathbf{b}_4\}$ by exchanging the order of the robots [see Fig. 4.2(b)]. Moreover, we only need to keep $\{d_{56}\}$ from the possible expansions of $\{d_{34}\}$, since all other problems are overdetermined.

(c) From $\{d_{12}, \mathbf{b}_1\}$ we can exclude two second level expansions from $\{d_{34}\}$ [see Fig. 4.2(c)], since $\{d_{12}, \mathbf{b}_1; d_{34}; \mathbf{b}_5\}$ and $\{d_{12}, \mathbf{b}_1; d_{34}; \mathbf{b}_6\}$ are equivalent to $\{d_{12}, \mathbf{b}_1; \mathbf{b}_3; d_{56}\}$ and $\{d_{12}, \mathbf{b}_1; \mathbf{b}_4; d_{56}\}$, respectively, by changing the order of the measurements.

(d) Similarly, from $\{\mathbf{b}_1\}$ we can exclude two second level expansions from $\{\mathbf{b}_4\}$, since $\{\mathbf{b}_1; \mathbf{b}_4; \mathbf{b}_5\}$ is equivalent to $\{\mathbf{b}_1; \mathbf{b}_3; \mathbf{b}_6\}$ by changing the order of measurements, and $\{\mathbf{b}_1; \mathbf{b}_4; \mathbf{b}_6\}$ is also equivalent to $\{\mathbf{b}_1; \mathbf{b}_3; \mathbf{b}_6\}$ by first exchanging the order of the robots and then changing the order of measurements. The other cases $\{d_{34}, \mathbf{b}_3, \mathbf{b}_4\}$, $\{\mathbf{b}_3, \mathbf{b}_4\}$, and $\{d_{34}, \mathbf{b}_3\}$ have already been considered in the first three expansions (a)–(c) as root nodes. $\{d_{34}, \mathbf{b}_4\}$ is also considered in (c) after changing the order of the robots.

(e) Finally, for branches expanding from $\{d_{12}\}$, we only need to consider the distance-only case [see Fig. 4.2(e)], since all other cases have also been considered before.

Adding all the cases together, we have a total of 14 minimal systems listed in Table 4.1. Next, we will present closed-form or analytical solutions to these problems.

4.4 Closed-form solutions to the minimal problems of Systems 1 and 2

For System 1, we measure $\{d_{12}, \mathbf{b}_1, \mathbf{b}_2; d_{34}\}$, and for System 2, we measure $\{\mathbf{b}_1, \mathbf{b}_2; \mathbf{b}_3\}$. Since the mutual bearing measurements $\mathbf{b}_1$ and $\mathbf{b}_2$ appear in both systems, their equations have similar structure and can be solved using the same approach. In this section, we will first derive the systems of equations for both problems, and then provide their solutions.

4.4.1 System 1: Measurements $\{d_{12}, \mathbf{b}_1, \mathbf{b}_2; d_{34}\}$

For this problem, the relative position is directly measured as $\mathbf{p} = d_{12}\mathbf{b}_1$. Therefore, we only need to compute the relative orientation, parameterized by $\mathbf{C}$.

From the mutual bearing measurements $\mathbf{b}_1$ and $\mathbf{b}_2$, we have the following constraint:

$$\mathbf{b}_1 + \mathbf{C}\mathbf{b}_2 = \mathbf{0} \tag{4.1}$$

Additionally, by expanding the constraint from the distance measurement d_{34}, we have

$$\begin{aligned} {}^3\mathbf{p}_4^T {}^3\mathbf{p}_4 &= (\mathbf{p} + \mathbf{C}^2\mathbf{p}_4 - {}^1\mathbf{p}_3)^T(\mathbf{p} + \mathbf{C}^2\mathbf{p}_4 - {}^1\mathbf{p}_3) = d_{34}^2 \\ \Rightarrow &\mathbf{v}^T\mathbf{C}^2\mathbf{p}_4 + a = 0 \end{aligned} \tag{4.2}$$

where $\mathbf{v} = 2(\mathbf{p} - {}^1\mathbf{p}_3)$ and $a = \mathbf{p}^T\mathbf{p} + {}^2\mathbf{p}_4^T{}^2\mathbf{p}_4 + {}^1\mathbf{p}_3^T{}^1\mathbf{p}_3 - 2\mathbf{p}^T{}^1\mathbf{p}_3 - d_{34}^2$ are known quantities.

The last step of the solution process is to find $\mathbf{C}$ from equations (4.1) and (4.2), which is described in Section 4.4.3.

4.4.2 System 2: Measurements $\{\mathbf{b}_1, \mathbf{b}_2; \mathbf{b}_3\}$

For this system, besides the mutual bearing constraint (4.1), we have the following equation using $\mathbf{b}_1$ and $\mathbf{b}_3$, which is the sum of vectors from {1}, through {2}, {4}, {3} and back to {1} (see Fig. 4.1):

$$\mathbf{p} + \mathbf{C}^2\mathbf{p}_4 - {}^1_3\mathbf{C}^3\mathbf{p}_4 - {}^1\mathbf{p}_3 = \mathbf{0}$$

$$\Rightarrow d_{12}\mathbf{b}_1 + \mathbf{C}^2\mathbf{p}_4 - d_{34}\mathbf{b}_3 - {}^1\mathbf{p}_3 = \mathbf{0}. \tag{4.3}$$

If the rotation $\mathbf{C}$ is known, the relative position can be found by first determining the distance d_{12}. To do this, we eliminate d_{34} from equation (4.3) by forming the cross product with $\mathbf{b}_3$, i.e.,

$$d_{12}\lfloor \mathbf{b}_3 \times \rfloor \mathbf{b}_1 + \lfloor \mathbf{b}_3 \times \rfloor \mathbf{C}^2\mathbf{p}_4 - \lfloor \mathbf{b}_3 \times \rfloor {}^1\mathbf{p}_3 = \mathbf{0} \tag{4.4}$$

where $\lfloor \mathbf{b}_3 \times \rfloor$ is a 3×3 skew-symmetric matrix corresponding to the cross product. Then d_{12} can be computed from (4.4) by forming the dot product with $\lfloor \mathbf{b}_3 \times \rfloor \mathbf{b}_1$, i.e.,

$$d_{12} = \frac{(\lfloor \mathbf{b}_3 \times \rfloor \mathbf{b}_1)^T \lfloor \mathbf{b}_3 \times \rfloor ({}^1\mathbf{p}_3 - \mathbf{C}^2\mathbf{p}_4)}{(\lfloor \mathbf{b}_3 \times \rfloor \mathbf{b}_1)^T (\lfloor \mathbf{b}_3 \times \rfloor \mathbf{b}_1)} \tag{4.5}$$

The relative position is then readily available as $\mathbf{p} = d_{12}\mathbf{b}_1$. Next, we will describe how to compute $\mathbf{C}$.

The unknown distances d_{12} and d_{34} can be eliminated from equation (4.3) by projecting it on $\mathbf{v} = \mathbf{b}_1 \times \mathbf{b}_3$, i.e., the cross product of $\mathbf{b}_1$ and $\mathbf{b}_3$:

$$\mathbf{v}^T\mathbf{C}^2\mathbf{p}_4 - \mathbf{v}^T{}^1\mathbf{p}_3 = 0 \tag{4.6}$$

If we define a scalar $a := -\mathbf{v}^T{}^1\mathbf{p}_3$, then it is easy to see that equations (4.6) and (4.2) have identical structure.

4.4.3 Rotation Matrix Determination

We have shown that for both Systems 1 and 2, in order to determine the rotation matrix $\mathbf{C}$, we need to solve the following system of equations:

$$\mathbf{b}_1 + \mathbf{C}\mathbf{b}_2 = \mathbf{0} \tag{4.7}$$

$$\mathbf{v}^T\mathbf{C}^2\mathbf{p}_4 + a = 0 \tag{4.8}$$

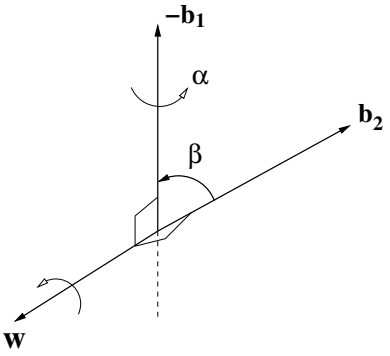


Figure 4.3: A rotation that satisfies the constraint $\mathbf{b}_1 + \mathbf{C}\mathbf{b}_2 = \mathbf{0}$, where $\mathbf{w} = \frac{\mathbf{b}_1 \times \mathbf{b}_2}{||\mathbf{b}_1 \times \mathbf{b}_2||}$.

The key idea behind our approach is to first exploit the geometric properties of (4.7) which will allow us to determine two degrees of freedom in rotation. The remaining unknown degree of freedom can subsequently be computed using (4.8).

We start by first showing the following lemma (see Fig. 4.3).

Lemma 17. *A particular solution to* (4.7) *is* $\mathbf{C}^* = \mathbf{C}(\mathbf{w}, \beta)$*, where* $\mathbf{w} = \frac{\mathbf{b}_1 \times \mathbf{b}_2}{||\mathbf{b}_1 \times \mathbf{b}_2||}$*, and* $\beta = Atan2(||\mathbf{b}_1 \times \mathbf{b}_2||, -\mathbf{b}_1^T \mathbf{b}_2)$.

Proof. Using the Rodrigues rotation formula, we have

$$\mathbf{C}(\mathbf{w}, \beta) = c\beta\mathbf{I} + s\beta\lfloor \mathbf{w} \times \rfloor + (1 - c\beta)\mathbf{w}\mathbf{w}^T \tag{4.9}$$

where $\lfloor \mathbf{w} \times \rfloor$ is the skew-symmetric matrix of $\mathbf{w}$, so that $\lfloor \mathbf{w} \times \rfloor \mathbf{b}_2 = \mathbf{w} \times \mathbf{b}_2$. Substituting $\mathbf{C}(\mathbf{w}, \beta)$ in (4.7), we have

$$-\mathbf{b}_1 = (c\beta\mathbf{I} + s\beta\lfloor \mathbf{w} \times \rfloor + (1 - c\beta)\mathbf{w}\mathbf{w}^T)\mathbf{b}_2 \tag{4.10}$$

$$\Rightarrow -\mathbf{b}_1 = c\beta\mathbf{b}_2 + s\beta\lfloor \mathbf{w} \times \rfloor \mathbf{b}_2 \tag{4.11}$$

Projecting (4.11) on $\mathbf{b}_2$, yields:

$$c\beta = -\mathbf{b}_1^T \mathbf{b}_2. \tag{4.12}$$

Premultiplying both sides of (4.11) with $\lfloor \mathbf{b}_2 \times \rfloor$ yields,

$$\begin{aligned} -\lfloor \mathbf{b}_2 \times \rfloor \mathbf{b}_1 &= s\beta \lfloor \mathbf{b}_2 \times \rfloor \lfloor \mathbf{w} \times \rfloor \mathbf{b}_2 \\ \Rightarrow \mathbf{w}||\mathbf{b}_1 \times \mathbf{b}_2|| &= -s\beta \lfloor \mathbf{b}_2 \times \rfloor \lfloor \mathbf{b}_2 \times \rfloor \mathbf{w} \\ \Rightarrow \mathbf{w}||\mathbf{b}_1 \times \mathbf{b}_2|| &= -s\beta(-\mathbf{I} + \mathbf{b}_2\mathbf{b}_2^T)\mathbf{w} \\ \Rightarrow \mathbf{w}||\mathbf{b}_1 \times \mathbf{b}_2|| &= s\beta\mathbf{w} \\ \Rightarrow s\beta &= ||\mathbf{b}_1 \times \mathbf{b}_2|| \end{aligned} \tag{4.13}$$

□

We next show the general form of solutions of (4.7) in Lemma 18.

Lemma 18. *Any solution of* (4.7) *assumes the form*

$$\mathbf{C} = \mathbf{C}(\mathbf{w}, \beta)\mathbf{C}(\mathbf{b}_2, \alpha) = \mathbf{C}(-\mathbf{b}_1, \alpha)\mathbf{C}(\mathbf{w}, \beta)$$

where α is an unknown angle to be determined.

Proof. Given that $\mathbf{C}(\mathbf{w}, \beta)$ is a particular solution of (4.7), i.e.,

$$-\mathbf{b}_1 = \mathbf{C}(\mathbf{w}, \beta)\mathbf{b}_2 \tag{4.14}$$

we seek to find all matrices $\mathbf{C}$ that satisfy (4.7). From (4.7) and (4.14), we have

$$\begin{aligned} \mathbf{C}\mathbf{b}_2 = -\mathbf{b}_1 &= \mathbf{C}(\mathbf{w}, \beta)\mathbf{b}_2 \\ \Rightarrow \mathbf{C}(\mathbf{w}, -\beta)\mathbf{C}\mathbf{b}_2 &= \mathbf{b}_2 \end{aligned} \tag{4.15}$$

Since $\mathbf{b}_2 = \mathbf{C}(\mathbf{b}_2, \alpha)\mathbf{b}_2$ for any α, we have

$$\begin{aligned} \mathbf{C}(\mathbf{w}, -\beta)\mathbf{C} &= \mathbf{C}(\mathbf{b}_2, \alpha) \\ \Rightarrow \mathbf{C} &= \mathbf{C}(\mathbf{w}, \beta)\mathbf{C}(\mathbf{b}_2, \alpha). \end{aligned} \tag{4.16}$$

We now prove the second part of the lemma, i.e., that $\mathbf{C} = \mathbf{C}(-\mathbf{b}_1, \alpha)\mathbf{C}(\mathbf{w}, \beta)$. Substituting the Rodrigues formula in (4.16) to expand $\mathbf{C}(\mathbf{b}_2, \alpha)$, we have

$$\begin{aligned}
\mathbf{C} &= \mathbf{C}(\mathbf{w}, \beta)(c\alpha\mathbf{I} + s\alpha\lfloor \mathbf{b}_2 \times \rfloor + (1 - c\alpha)\mathbf{b}_2\mathbf{b}_2^T) && (4.17)\\
&= c\alpha\mathbf{C}(\mathbf{w}, \beta) + s\alpha\mathbf{C}(\mathbf{w}, \beta)\lfloor \mathbf{b}_2 \times \rfloor + (1 - c\alpha)\mathbf{C}(\mathbf{w}, \beta)\mathbf{b}_2\mathbf{b}_2^T && (4.18)\\
&= c\alpha\mathbf{C}(\mathbf{w}, \beta) + s\alpha\lfloor \mathbf{C}(\mathbf{w}, \beta)\mathbf{b}_2 \times \rfloor \mathbf{C}(\mathbf{w}, \beta) \\
&\quad + (1 - c\alpha)(\mathbf{C}(\mathbf{w}, \beta)\mathbf{b}_2)(\mathbf{C}(\mathbf{w}, \beta)\mathbf{b}_2)^T\mathbf{C}(\mathbf{w}, \beta) && (4.19)\\
&= [c\alpha\mathbf{I} + s\alpha\lfloor \mathbf{C}(\mathbf{w}, \beta)\mathbf{b}_2 \times \rfloor + (1 - c\alpha) \cdot (\mathbf{C}(\mathbf{w}, \beta)\mathbf{b}_2)(\mathbf{C}(\mathbf{w}, \beta)\mathbf{b}_2)^T]\mathbf{C}(\mathbf{w}, \beta) && (4.20)\\
&= \mathbf{C}(\mathbf{C}(\mathbf{w}, \beta)\mathbf{b}_2, \alpha)\mathbf{C}(\mathbf{w}, \beta) && (4.21)\\
&= \mathbf{C}(-\mathbf{b}_1, \alpha)\mathbf{C}(\mathbf{w}, \beta) && (4.22)
\end{aligned}$$

where for the last equality, we used (4.14), while from (4.18) to (4.19) we employed the equality $\mathbf{C}(\mathbf{w}, \beta)\lfloor \mathbf{b}_2 \times \rfloor = \lfloor \mathbf{C}(\mathbf{w}, \beta)\mathbf{b}_2 \times \rfloor \mathbf{C}(\mathbf{w}, \beta)$.

□

The last step of this process is to substitute $\mathbf{C} = \mathbf{C}(-\mathbf{b}_1, \alpha)\mathbf{C}(\mathbf{w}, \beta)$ into (4.8) to determine α, i.e.,

$$0 = \mathbf{v}^T\mathbf{C}(-\mathbf{b}_1, \alpha)^2\mathbf{p}_4' + a \tag{4.23}$$

where ${}^2\mathbf{p}_4' = \mathbf{C}(\mathbf{w}, \beta)^2\mathbf{p}_4$.

Substituting the Rodrigues formula for $\mathbf{C}(-\mathbf{b}_1, \alpha)$ in (4.23), yields,

$$\begin{aligned}
&\mathbf{v}^T(c\alpha\mathbf{I} + s\alpha\lfloor -\mathbf{b}_1 \times \rfloor + (1 - c\alpha)\mathbf{b}_1\mathbf{b}_1^T)^2\mathbf{p}_4' + a && (4.24)\\
&= \underbrace{(\mathbf{v}^T{}^2\mathbf{p}_4' - \mathbf{v}^T\mathbf{b}_1\mathbf{b}_1^T{}^2\mathbf{p}_4')}_{l_1} c\alpha - \underbrace{(\mathbf{v}^T\lfloor \mathbf{b}_1 \times \rfloor^2\mathbf{p}_4')}_{l_2} s\alpha + \underbrace{\mathbf{v}^T\mathbf{b}_1\mathbf{b}_1^T{}^2\mathbf{p}_4' + a}_{l_3} = 0 \\
&\Rightarrow c\alpha = \frac{l_2 s\alpha - l_3}{l_1} && (4.25)
\end{aligned}$$

Finally, substituting (4.25) into the trigonometric constraint $c\alpha^2 + s\alpha^2 = 1$, we arrive at a quadratic polynomial in $s\alpha$.

$$m_0 s\alpha^2 + m_1 s\alpha + m_2 = 0$$

where $m_0 = l_1^2 + l_2^2$, $m_1 = -2l_2l_3$, and $m_2 = l_3^2 - l_1^2$. Back substituting the two solutions for $s\alpha$ into equation (4.25), we get two solutions for $c\alpha$.

$$s\alpha_1 = \frac{-m_1 + \triangle}{2m_0}, \qquad s\alpha_2 = \frac{-m_1 - \triangle}{2m_0} \tag{4.26}$$

$$c\alpha_1 = \frac{-l_2(m_1 - \triangle)}{2l_1m_0} - \frac{l_3}{l_1}, \; c\alpha_2 = \frac{-l_2(m_1 + \triangle)}{2l_1m_0} - \frac{l_3}{l_1} \tag{4.27}$$

where $\triangle = \sqrt{m_1^2 - 4m_0m_2}$.

Therefore, there exist up to two distinct[2] solutions for the transformation between frames $\{1\}$ and $\{2\}$.

4.5 Unidentifiability of Systems 3 and 4

For these two systems, we will show that there exist infinite solutions for the rotation $\mathbf{C}$.

4.5.1 System 3: Measurements $\{d_{12}, \mathbf{b}_1; d_{34}, \mathbf{b}_3\}$

Let us first examine System 3. Substituting the measurements d_{12}, $\mathbf{b}_1$, d_{34}, and $\mathbf{b}_3$ in equation (4.3), we have only one geometric constraint for $\mathbf{C}$

$$\begin{aligned} \mathbf{p} + \mathbf{C}^2\mathbf{p}_4 - \tfrac{1}{3}\mathbf{C}^3\mathbf{p}_4 - {}^1\mathbf{p}_3 = 0 \\ \Rightarrow d_{12}\mathbf{b}_1 + \mathbf{C}^2\mathbf{p}_4 - d_{34}\mathbf{b}_3 - {}^1\mathbf{p}_3 = 0 \\ \Rightarrow (d_{12}\mathbf{b}_1 - d_{34}\mathbf{b}_3 - {}^1\mathbf{p}_3) + \mathbf{C}^2\mathbf{p}_4 = 0 \end{aligned} \tag{4.28}$$

where, as evident, the rotation around the unit vector in the direction of $d_{12}\mathbf{b}_1 - d_{34}\mathbf{b}_3 - {}^1\mathbf{p}_3$ is undetermined.

[2] In case $\triangle = 0$, these two solutions collapse to one.

4.5.2 System 4: Measurements $\{d_{12}, \mathbf{b}_1; d_{34}, \mathbf{b}_4\}$

Similarly, System 4 also has one degree of freedom in rotation undetermined. The difference is that the vector ${}^4\mathbf{p}_3$ is measured instead of vector ${}^3\mathbf{p}_4$.

$$\mathbf{p} + \mathbf{C}^2\mathbf{p}_4 + \mathbf{C}\,{}^2_4\mathbf{C}^4\mathbf{p}_3 - {}^1\mathbf{p}_3 = 0 \tag{4.29}$$

$$\Rightarrow \mathbf{p} - {}^1\mathbf{p}_3 + \mathbf{C}({}^2\mathbf{p}_4 + d_{34}\mathbf{b}_4) = 0 \tag{4.30}$$

In this case, the rotation around $\mathbf{p} - {}^1\mathbf{p}_3$ is undetermined. Therefore, these two systems have infinite number of solutions.

4.6 Closed-form Solutions to the minimal problem of System 5

For System 5, the available measurements are $\{\mathbf{b}_1,\ \mathbf{b}_2;\ d_{34};\ d_{56}\}$. Using the mutual bearing measurements $\mathbf{b}_1$ and $\mathbf{b}_2$, we can again express the rotation matrix as $\mathbf{C} = \mathbf{C}(-\mathbf{b}_1, \alpha)\mathbf{C}(\mathbf{w}, \beta)$ (see Lemma 18), where $\mathbf{b}_1$, $\mathbf{w}$, and β are known while α as well as d_{12} can be computed from the two distance constraints:

$$(d_{12}\mathbf{b}_1 + \mathbf{C}^2\mathbf{p}_4 - {}^1\mathbf{p}_3)^T(d_{12}\mathbf{b}_1 + \mathbf{C}^2\mathbf{p}_4 - {}^1\mathbf{p}_3) = d_{34}^2 \tag{4.31}$$

$$(d_{12}\mathbf{b}_1 + \mathbf{C}^2\mathbf{p}_6 - {}^1\mathbf{p}_5)^T(d_{12}\mathbf{b}_1 + \mathbf{C}^2\mathbf{p}_6 - {}^1\mathbf{p}_5) = d_{56}^2 \tag{4.32}$$

After expanding the above two equations, we have

$$d_{12}^2 + 2d_{12}(\mathbf{b}_1^T\mathbf{C}^2\mathbf{p}_4 - \mathbf{b}_1^T{}^1\mathbf{p}_3) - 2{}^1\mathbf{p}_3^T\mathbf{C}^2\mathbf{p}_4 + \epsilon_1 = 0 \tag{4.33}$$

$$d_{12}^2 + 2d_{12}(\mathbf{b}_1^T\mathbf{C}^2\mathbf{p}_6 - \mathbf{b}_1^T{}^1\mathbf{p}_5) - 2{}^1\mathbf{p}_5^T\mathbf{C}^2\mathbf{p}_6 + \epsilon_2 = 0 \tag{4.34}$$

where $\epsilon_1 = {}^2\mathbf{p}_4^T{}^2\mathbf{p}_4 + {}^1\mathbf{p}_3^T{}^1\mathbf{p}_3 - d_{34}$, and $\epsilon_2 = {}^2\mathbf{p}_6^T{}^2\mathbf{p}_6 + {}^1\mathbf{p}_5^T{}^1\mathbf{p}_5 - d_{56}$ are known. Substituting $\mathbf{C} = \mathbf{C}(-\mathbf{b}_1, \alpha)\mathbf{C}(\mathbf{w}, \beta)$ in (4.33) and (4.34) yields,

$$d_{12}^2 + 2d_{12}\mathbf{b}_1^T({}^2\mathbf{p}'_4 - {}^1\mathbf{p}_3) - 2{}^1\mathbf{p}_3^T\mathbf{C}(-\mathbf{b}_1, \alpha)^2\mathbf{p}'_4 + \epsilon_1 = 0 \tag{4.35}$$

$$d_{12}^2 + 2d_{12}\mathbf{b}_1^T({}^2\mathbf{p}'_6 - {}^1\mathbf{p}_5) - 2{}^1\mathbf{p}_5^T\mathbf{C}(-\mathbf{b}_1, \alpha)^2\mathbf{p}'_6 + \epsilon_2 = 0 \tag{4.36}$$

where we have used the property $\mathbf{b}_1^T\mathbf{C}(-\mathbf{b}_1,\alpha)=\mathbf{b}_1^T$ and set ${}^2\mathbf{p}'_4=\mathbf{C}(\mathbf{w},\beta)^2\mathbf{p}_4$, ${}^2\mathbf{p}'_6=\mathbf{C}(\mathbf{w},\beta)^2\mathbf{p}_6$. Employing the Rodrigues formula for $\mathbf{C}(-\mathbf{b}_1,\alpha)$ in (4.35)–(4.36) yields:

$$\mathbf{L}\begin{bmatrix}c\alpha\\ s\alpha\end{bmatrix}=\boldsymbol{\xi}\Rightarrow\begin{bmatrix}c\alpha\\ s\alpha\end{bmatrix}=\mathbf{L}^{-1}\boldsymbol{\xi} \tag{4.37}$$

where

$$\mathbf{L}=2\begin{bmatrix}{}^1\mathbf{p}_3^T(\mathbf{I}-\mathbf{b}_1\mathbf{b}_1^T)^2\mathbf{p}'_4 & -{}^1\mathbf{p}_3^T\lfloor\mathbf{b}_1\times\rfloor^2\mathbf{p}'_4\\ {}^1\mathbf{p}_5^T(\mathbf{I}-\mathbf{b}_1\mathbf{b}_1^T)^2\mathbf{p}'_6 & -{}^1\mathbf{p}_5^T\lfloor\mathbf{b}_1\times\rfloor^2\mathbf{p}'_6\end{bmatrix} \tag{4.38}$$

is a known 2×2 matrix, while each of the two components of $\boldsymbol{\xi}$ is quadratic in the unknown d_{12}, i.e.,

$$\boldsymbol{\xi}=\begin{bmatrix}d_{12}^2+2d_{12}\mathbf{b}_1^T({}^2\mathbf{p}'_4-{}^1\mathbf{p}_3)-2({}^1\mathbf{p}_3^T\mathbf{b}_1)({}^2\mathbf{p}_4'^T\mathbf{b}_1)+\epsilon_1\\ d_{12}^2+2d_{12}\mathbf{b}_1^T({}^2\mathbf{p}'_6-{}^1\mathbf{p}_5)-2({}^1\mathbf{p}_5^T\mathbf{b}_1)({}^2\mathbf{p}_6'^T\mathbf{b}_1)+\epsilon_2\end{bmatrix}.$$

Taking the norm of both sides of (4.37) and employing the trigonometric constraint $c\alpha^2+s\alpha^2=1$, gives

$$\boldsymbol{\xi}^T\mathbf{L}^{-T}\mathbf{L}^{-1}\boldsymbol{\xi}=1 \tag{4.39}$$

which is a 4^{th} order univariate polynomial in d_{12} which can be solved in closed form [76] to yield up to four real solutions for d_{12}. Back-substituting each positive root of (4.39) in (4.37) provides a unique solution for the rotation angle α, and hence the rotation matrix $\mathbf{C}$. Therefore, there exist up to four solutions for the relative robot-to-robot transformation of System 5.

4.7 Closed-form Solutions to the Minimal Problems of Systems 6 and 7

Due to their similarities, Systems 6 and 7 can be formulated as an identically structured system of equations and solved using the same methodology. Note that in both systems,

the relative position $\mathbf{p} = d_{12}\mathbf{b}_1$ is directly measured and the unknown quantities are in the rotation matrix $\mathbf{C}$. In the following sections, we first derive the system of equations for both problems, and then present the closed-form solution.

4.7.1 System 6: Measurements $\{d_{12},\ \mathbf{b}_1;\ \mathbf{b}_3;\ d_{56}\ \}$

Given the relative position $\mathbf{p} = d_{12}\mathbf{b}_1$ and the second bearing measurement $\mathbf{b}_3$, we have the following constraint, which is the sum of vectors from {1} through {2}, {4}, {3}, and back to {1} (see Fig. 4.1):

$$\begin{aligned} \mathbf{p} + \mathbf{C}^2\mathbf{p}_4 - {}^1_3\mathbf{C}^3\mathbf{p}_4 - {}^1\mathbf{p}_3 = 0 \\ \Rightarrow \mathbf{p} + \mathbf{C}^2\mathbf{p}_4 - d_{34}\mathbf{b}_3 - {}^1\mathbf{p}_3 = 0. \qquad (4.40) \\ \Rightarrow \lfloor \mathbf{b}_3 \times \rfloor ({}^1\mathbf{p}_3 - \mathbf{p}) = \lfloor \mathbf{b}_3 \times \rfloor \mathbf{C}^2\mathbf{p}_4 \qquad (4.41) \end{aligned}$$

where we have multiplied both sides of (4.40) with the cross-product matrix $\lfloor \mathbf{b}_3 \times \rfloor$ to eliminate d_{34}.

Finally, by expanding the constraint from the distance measurement d_{56}, we have the third equation necessary for solving the 3-DOF relative orientation $\mathbf{C}$.

$$\begin{aligned} {}^5\mathbf{p}_6^{T\,5}\mathbf{p}_6 = (\mathbf{p} + \mathbf{C}^2\mathbf{p}_6 - {}^1\mathbf{p}_5)^T(\mathbf{p} + \mathbf{C}^2\mathbf{p}_6 - {}^1\mathbf{p}_5) = d_{56}^2 \\ \Rightarrow 2(\mathbf{p} - {}^1\mathbf{p}_5)^T\mathbf{C}^2\mathbf{p}_6 + \epsilon = 0 \qquad (4.42) \end{aligned}$$

where $\epsilon = \mathbf{p}^T\mathbf{p} + {}^2\mathbf{p}_6^{T\,2}\mathbf{p}_6 + {}^1\mathbf{p}_5^{T\,1}\mathbf{p}_5 - 2\mathbf{p}^{T\,1}\mathbf{p}_5 - d_{56}^2$.

4.7.2 System 7: Measurements $\{d_{12},\ \mathbf{b}_1;\ \mathbf{b}_4;\ d_{56}\}$

Similar to System 6, we have the following constraint from the first three measurements.

$$\begin{aligned} \mathbf{p} + \mathbf{C}^2\mathbf{p}_4 + \mathbf{C}_4^2\mathbf{C}^4\mathbf{p}_3 - {}^1\mathbf{p}_3 = 0 \qquad (4.43) \\ \Rightarrow \mathbf{C}^T(\mathbf{p} - {}^1\mathbf{p}_3) + {}^2\mathbf{p}_4 + d_{34}\mathbf{b}_4 = 0 \qquad (4.44) \\ \Rightarrow \lfloor \mathbf{b}_4 \times \rfloor^2\mathbf{p}_4 = \lfloor \mathbf{b}_4 \times \rfloor \mathbf{C}^T({}^1\mathbf{p}_3 - \mathbf{p}) \qquad (4.45) \end{aligned}$$

where we have multiplied both sides of (4.44) with the cross-product matrix $\lfloor \mathbf{b}_4 \times \rfloor$ to eliminate d_{34}. Together with the distance constraint (4.42), rewritten as

$$2^2 \mathbf{p}_6^T \mathbf{C}^T (\mathbf{p} - {}^1\mathbf{p}_5) + \epsilon = 0 \tag{4.46}$$

we have three equations for solving $\mathbf{C}^T$.

4.7.3 Closed-form Solution for the Rotation Matrix

For System 6, we need to determine the rotation matrix $\mathbf{C}$ (or $\mathbf{C}^T$ for System 7) from equations of the form [see (4.41), (4.42) and (4.45), (4.46)]:

$$\lfloor \mathbf{b} \times \rfloor \mathbf{u} = \lfloor \mathbf{b} \times \rfloor \mathbf{C}\mathbf{v} \tag{4.47}$$

$$\mathbf{u}'^T \mathbf{C}\mathbf{v}' + a = 0 \tag{4.48}$$

where $\mathbf{u}$, $\mathbf{v}$, $\mathbf{u}'$, $\mathbf{v}'$ and a are known, and $\mathbf{b}$ is a unit vector. In our solution method, we will exploit the results of Section 4.4.3 [see (4.7), (4.8)]. To do so, we first show the following:

Lemma 19. *The rotational matrices satisfying* (4.47)*, also satisfy the following equations:*

$$\mathbf{w}_0^+ = \mathbf{C}\mathbf{v}_0 \tag{4.49}$$

$$\textit{or} \quad \mathbf{w}_0^- = \mathbf{C}\mathbf{v}_0 \tag{4.50}$$

where $\mathbf{v}_0 = \mathbf{v}/||\mathbf{v}||$, $\mathbf{w}_0^+ = \frac{1}{||\mathbf{v}||}[\mathbf{u} + (\gamma_0 - \mathbf{b}^T\mathbf{u})\mathbf{b}]$, $\mathbf{w}_0^- = \frac{1}{||\mathbf{v}||}[\mathbf{u} + (-\gamma_0 - \mathbf{b}^T\mathbf{u})\mathbf{b}]$, *and* $\gamma_0 = \sqrt{\mathbf{v}^T\mathbf{v} + (\mathbf{b}^T\mathbf{u})^2 - \mathbf{u}^T\mathbf{u}}$.

Proof. We start by rewriting $\mathbf{u}$ and $\mathbf{v}$ as

$$\mathbf{u} = \mathbf{u} - (\mathbf{b}^T\mathbf{u})\mathbf{b} + (\mathbf{b}^T\mathbf{u})\mathbf{b} = \mathbf{u}^\perp + (\mathbf{b}^T\mathbf{u})\mathbf{b} \tag{4.51}$$

$$\begin{aligned} \mathbf{C}\mathbf{v} &= \mathbf{C}\mathbf{v} - (\mathbf{b}^T\mathbf{C}\mathbf{v})\mathbf{b} + (\mathbf{b}^T\mathbf{C}\mathbf{v})\mathbf{b} \\ &= \mathbf{v}^\perp + (\mathbf{b}^T\mathbf{C}\mathbf{v})\mathbf{b} \end{aligned} \tag{4.52}$$

where $\mathbf{u}^\perp$ and $\mathbf{v}^\perp$ are the components of $\mathbf{u}$ and $\mathbf{Cv}$ respectively that are perpendicular to $\mathbf{b}$ (i.e., $\mathbf{b}^T\mathbf{u}^\perp = 0$ and $\mathbf{b}^T\mathbf{v}^\perp = 0$).

Substituting (4.51) and (4.52) in (4.47) yields:

$$\begin{aligned} \lfloor \mathbf{b} \times \rfloor(\mathbf{u}^\perp + (\mathbf{b}^T\mathbf{u})\mathbf{b}) &= \lfloor \mathbf{b} \times \rfloor(\mathbf{v}^\perp + (\mathbf{b}^T\mathbf{Cv})\mathbf{b}) \\ \Rightarrow \lfloor \mathbf{b} \times \rfloor\mathbf{u}^\perp &= \lfloor \mathbf{b} \times \rfloor\mathbf{v}^\perp \\ \mathbf{u}^\perp &= \mathbf{v}^\perp \end{aligned} \tag{4.53}$$

where in this last equality, we used the fact that both $\mathbf{u}^\perp$ and $\mathbf{v}^\perp$ are perpendicular to $\mathbf{b}$ and their cross products are equal.

Employing the expressions for $\mathbf{u}^\perp$ and $\mathbf{v}^\perp$ from (4.51)–(4.52) in (4.53) we have

$$\mathbf{u} - (\mathbf{b}^T\mathbf{u})\mathbf{b} = \mathbf{Cv} - (\mathbf{b}^T\mathbf{Cv})\mathbf{b} \tag{4.54}$$

$$\Rightarrow \mathbf{u} + (\gamma - \mathbf{b}^T\mathbf{u})\mathbf{b} = \mathbf{Cv} \tag{4.55}$$

where $\gamma = \mathbf{b}^T\mathbf{Cv}$ in this last equation can be determined by computing the norm of both sides of (4.55), i.e.,:

$$\begin{aligned} \mathbf{v}^T\mathbf{v} &= [\mathbf{u} + (\gamma - \mathbf{b}^T\mathbf{u})\mathbf{b}]^T[\mathbf{u} + (\gamma - \mathbf{b}^T\mathbf{u})\mathbf{b}] \\ \Rightarrow \gamma^2 &= \mathbf{v}^T\mathbf{v} + (\mathbf{b}^T\mathbf{u})^2 - \mathbf{u}^T\mathbf{u} \\ \Rightarrow \gamma &= \gamma_0 \quad \text{or} \quad \gamma = -\gamma_0 \end{aligned} \tag{4.56}$$

where $\gamma_0 = \sqrt{\mathbf{v}^T\mathbf{v} + (\mathbf{b}^T\mathbf{u})^2 - \mathbf{u}^T\mathbf{u}}$. Substituing each of them back in (4.55) and dividing both sides with the norm of the corresponding vector yields (4.49)-(4.50). □

Combining each of the (4.49)-(4.50) with (4.48) provides the following two systems of equations

$$\left.\begin{array}{c} \mathbf{w}_0^+ = \mathbf{Cv}_0 \\ \mathbf{u}'^T\mathbf{Cv}' + a = 0 \end{array}\right\} \Sigma_1 \text{ or } \left.\begin{array}{c} \mathbf{w}_0^- = \mathbf{Cv}_0 \\ \mathbf{u}'^T\mathbf{Cv}' + a = 0 \end{array}\right\} \Sigma_2 \tag{4.57}$$

Note that both (Σ_1) and (Σ_2) have identical structures as (4.7)–(4.8). Thus, by employing the approach of Section 4.4.3, we can compute up to two solutions for each of them for a total of up to four solutions for Systems 6 and 7.

4.8 Analytical Solutions to the Minimal Problems of Systems 8, 9, and 10

In the following, we first derive the system of equations for these three problems, and then present our analytic solutions. In our approach, we first solve for the rotation matrix $\mathbf{C}$ and then compute the translation vector $\mathbf{p}$.

4.8.1 System 8: Measurements $\{\mathbf{b}_1; \mathbf{b}_3; \mathbf{b}_5\}$

From Fig. 4.1, using pairwise the measurements $\mathbf{b}_1$ and $\mathbf{b}_3$, and $\mathbf{b}_1$ and $\mathbf{b}_5$, we have the following two constraints

$$d_{12}\mathbf{b}_1 + \mathbf{C}^2\mathbf{p}_4 - d_{34}\mathbf{b}_3 - {}^1\mathbf{p}_3 = 0 \tag{4.58}$$

$$d_{12}\mathbf{b}_1 + \mathbf{C}^2\mathbf{p}_6 - d_{56}\mathbf{b}_5 - {}^1\mathbf{p}_5 = 0 \tag{4.59}$$

whose difference gives:

$$d_{34}\mathbf{b}_3 + \mathbf{C}({}^2\mathbf{p}_6 - {}^2\mathbf{p}_4) - d_{56}\mathbf{b}_5 - ({}^1\mathbf{p}_5 - {}^1\mathbf{p}_3) = 0. \tag{4.60}$$

If the rotation matrix $\mathbf{C}$ was known, we can solve for d_{12} by first eliminating the unknown d_{34} from (4.58) by forming the cross product with $\mathbf{b}_3$ [see (4.5)].

Next, we describe how to eliminate all the unknown distances, d_{12}, d_{34}, and d_{56}, from (4.58), (4.59), and (4.60), so that we can solve for $\mathbf{C}$. Specifically, we define three vectors

$$\mathbf{v}_1 = \mathbf{b}_1 \times \mathbf{b}_3, \quad \mathbf{v}_2 = \mathbf{b}_1 \times \mathbf{b}_5, \quad \mathbf{v}_3 = \mathbf{b}_3 \times \mathbf{b}_5 \tag{4.61}$$

and form their dot products with (4.58), (4.59), and (4.60), respectively:

$$\mathbf{v}_1^T\mathbf{C}^2\mathbf{p}_4 - \mathbf{v}_1^T{}^1\mathbf{p}_3 = 0 \tag{4.62}$$

$$\mathbf{v}_2^T\mathbf{C}^2\mathbf{p}_6 - \mathbf{v}_2^T{}^1\mathbf{p}_5 = 0 \tag{4.63}$$

$$\mathbf{v}_3^T\mathbf{C}({}^2\mathbf{p}_6 - {}^2\mathbf{p}_4) - \mathbf{v}_3^T({}^1\mathbf{p}_5 - {}^1\mathbf{p}_3) = 0. \tag{4.64}$$

From these three equations, we determine $\mathbf{C}$ (see Section 4.8.4).

4.8.2 System 9: Measurements $\{\mathbf{b}_1;\ \mathbf{b}_3;\ \mathbf{b}_6\ \}$

Systems 9 and 8 both contain measurements $\mathbf{b}_1$ and $\mathbf{b}_3$. Therefore, System 9 also contains equation (4.62). The differences lie in the next two constraints using $\mathbf{b}_1$ and $\mathbf{b}_6$, and $\mathbf{b}_3$ and $\mathbf{b}_6$:

$$d_{12}\mathbf{b}_1 + \mathbf{C}\,{}^2\mathbf{p}_6 + d_{56}\mathbf{C}\mathbf{b}_6 - {}^1\mathbf{p}_5 = 0 \tag{4.65}$$

$$d_{34}\mathbf{b}_3 + \mathbf{C}({}^2\mathbf{p}_6 - {}^2\mathbf{p}_4) + d_{56}\mathbf{C}\mathbf{b}_6 - ({}^1\mathbf{p}_5 - {}^1\mathbf{p}_3) = 0. \tag{4.66}$$

Similarly to System 8, we define

$$\mathbf{v}_1 = \mathbf{b}_1 \times \mathbf{b}_3, \quad \mathbf{v}_2 = \mathbf{b}_1 \times \mathbf{C}\mathbf{b}_6, \quad \mathbf{v}_3 = \mathbf{b}_3 \times \mathbf{C}\mathbf{b}_6 \tag{4.67}$$

and project equation (4.65) to $\mathbf{v}_2$ to obtain

$$(\mathbf{b}_1 \times \mathbf{C}\mathbf{b}_6)^T \mathbf{C}\,{}^2\mathbf{p}_6 - {}^1\mathbf{p}_5^T(\mathbf{b}_1 \times \mathbf{C}\mathbf{b}_6) = 0 \tag{4.68}$$

$$\Rightarrow \mathbf{b}_1^T(\mathbf{C}\mathbf{b}_6 \times \mathbf{C}\,{}^2\mathbf{p}_6) - ({}^1\mathbf{p}_5 \times \mathbf{b}_1)^T\mathbf{C}\mathbf{b}_6 = 0 \tag{4.69}$$

$$\Rightarrow \mathbf{b}_1^T\mathbf{C}(\mathbf{b}_6 \times {}^2\mathbf{p}_6) - ({}^1\mathbf{p}_5 \times \mathbf{b}_1)^T\mathbf{C}\mathbf{b}_6 = 0 \tag{4.70}$$

where from (4.68) to (4.69), we have applied the identity $(\mathbf{a} \times \mathbf{b})^T\mathbf{c} = \mathbf{a}^T(\mathbf{b} \times \mathbf{c})$.

Similarly projecting (4.66) to $\mathbf{v}_3$, yields

$$\mathbf{b}_3^T\mathbf{C}\mathbf{u}_3 - \mathbf{u}_2^T\mathbf{C}\mathbf{b}_6 = 0 \tag{4.71}$$

where $\mathbf{u}_3 = \mathbf{b}_6 \times ({}^2\mathbf{p}_6 - {}^2\mathbf{p}_4)$, and $\mathbf{u}_2 = ({}^1\mathbf{p}_5 - {}^1\mathbf{p}_3) \times \mathbf{b}_3$. Finally, from (4.62), (4.70), and (4.71), we solve for $\mathbf{C}$ (see Section 4.8.4). Note also that, once $\mathbf{C}$ is determined, d_{12} is computed from (4.58) after forming the cross product with $\mathbf{b}_3$ [see (4.5)].

4.8.3 System 10: Measurements $\{d_{12},\ \mathbf{b}_1;\ d_{34};\ d_{56};\ d_{78}\ \}$

In this problem, the relative position is known from $\mathbf{p} = d_{12}\mathbf{b}_1$. The remaining quantity to be determined is the rotation matrix $\mathbf{C}$. From the three distance measurements $d_{2i-1,2i}$, $i = 2,\ 3,\ 4$, we have [see (4.42)]:

$$2(\mathbf{p} - {}^1\mathbf{p}_{2i-1})^T\mathbf{C}\,{}^2\mathbf{p}_{2i} + \epsilon_i = 0,\ i = 2,\ 3,\ 4 \tag{4.72}$$

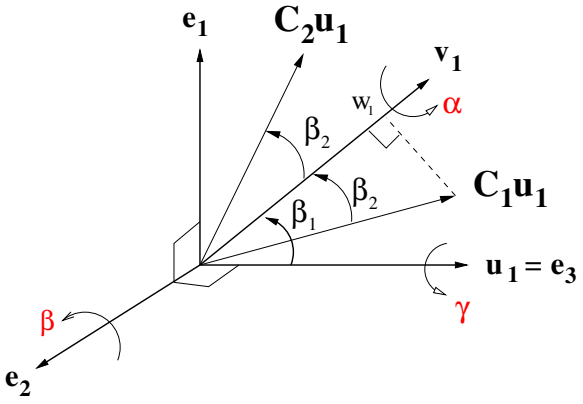


Figure 4.4: Sequence of rotations that satisfies the constraint: $\mathbf{v}_1^T\mathbf{C}\mathbf{u}_1 = w_1$. Let $\mathbf{e}_3 = \mathbf{u}_1$, $\mathbf{e}_2 = \frac{\mathbf{u}_1\times\mathbf{v}_1}{||\mathbf{u}_1\times\mathbf{v}_1||}$, and $\mathbf{e}_1 = \mathbf{e}_2 \times \mathbf{e}_3$. Rotating around $\mathbf{e}_2$ by $\beta = \beta_1 - \beta_2$ or $\beta = \beta_1 + \beta_2$ satisfies $\mathbf{v}_1^T\mathbf{C}\mathbf{u}_1 = w_1$. The constraint is satisfied for arbitrary rotations around the other two axes $\mathbf{u}_1$ and $\mathbf{v}_1$.

where $\epsilon_i = \mathbf{p}^T\mathbf{p} + {}^2\mathbf{p}_{2i}^T{}^2\mathbf{p}_{2i} + {}^1\mathbf{p}_{2i-1}^T{}^1\mathbf{p}_{2i-1} - 2\mathbf{p}^T{}^1\mathbf{p}_{2i-1} - d_{2i-1,2i}^2$.

4.8.4 Analytical Solution for the Rotation Matrix

Systems 8, 9, and 10 share the same form of equations for solving $\mathbf{C}$, i.e.,

$$f_1 = \mathbf{v}_1^T\mathbf{C}\mathbf{u}_1 - w_1 = 0 \tag{4.73}$$

$$f_i = \sum \mathbf{v}_i^T\mathbf{C}\mathbf{u}_i - w_i = 0, \quad i = 2,\ 3. \tag{4.74}$$

where we have normalized (4.73) to make the vectors $\mathbf{v}_1$ and $\mathbf{u}_1$ unit vectors.

Instead of using rotations around perpendicular axes, in this case we parameterize the rotation matrix as a product of three consecutive rotations around axes spanning 3-DOF:

$$\mathbf{C} = \mathbf{C}(\mathbf{v}_1, \alpha)\mathbf{C}(\mathbf{e}_2, \beta)\mathbf{C}(\mathbf{u}_1, \gamma) \tag{4.75}$$

where $\mathbf{e}_2 := \frac{\mathbf{u}_1\times\mathbf{v}_1}{||\mathbf{u}_1\times\mathbf{v}_1||}$. Substituting (4.75) in (4.73), we have

$$\mathbf{v}_1^T\mathbf{C}(\mathbf{e}_2, \beta)\mathbf{u}_1 - w_1 = 0 \tag{4.76}$$

From the geometry[3] of Fig. 4.4, we can find two particular solutions for the rotational angle β

$$\beta = \beta_1 - \beta_2 = \arccos(\mathbf{u}_1^T \mathbf{v}_1) - \arccos(w_1), \text{ or} \tag{4.77}$$

$$\beta = \beta_1 + \beta_2 = \arccos(\mathbf{u}_1^T \mathbf{v}_1) + \arccos(w_1). \tag{4.78}$$

However, later on we will prove that choosing any one of the two solutions leads to exactly the same set of 8 solutions for the relative rotation matrix. For now, we select the first solution for β and continue solving for the other two rotation angles.

Next, we will find the rotation angles α and γ from the next two equations.

$$\sum \mathbf{v}_2^T \mathbf{C}(\mathbf{v}_1, \alpha)\mathbf{C}(\mathbf{e}_2, \beta)\mathbf{C}(\mathbf{u}_1, \gamma)\mathbf{u}_2 = w_2 \tag{4.79}$$

$$\sum \mathbf{v}_3^T \mathbf{C}(\mathbf{v}_1, \alpha)\mathbf{C}(\mathbf{e}_2, \beta)\mathbf{C}(\mathbf{u}_1, \gamma)\mathbf{u}_3 = w_3 \tag{4.80}$$

Since β is known, the above equations are bilinear in $c\alpha$, $s\alpha$ and $c\gamma$, $s\gamma$. We rewrite them as linear functions in $c\gamma$ and $s\gamma$.

$$\mathbf{A} \begin{bmatrix} c\gamma \\ s\gamma \end{bmatrix} = \begin{bmatrix} w_2 - \ell_1 \\ w_3 - \ell_2 \end{bmatrix} \tag{4.81}$$

$$\Rightarrow \begin{bmatrix} c\gamma \\ s\gamma \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} w_2 - \ell_1 \\ w_3 - \ell_2 \end{bmatrix} \tag{4.82}$$

where $\mathbf{A}$ and ℓ_1, ℓ_2 are linear in $c\alpha$ and $s\alpha$ (see our technical report [94] for details).

Note that $c\gamma$ and $s\gamma$ in (4.82) are rational functions whose numerator and denominator are quadratic functions in $c\alpha$ and $s\alpha$. Substituting them in the trigonometric constraint $c\gamma^2 + s\gamma^2 = 1$, and after some algebraic manipulation we obtain a 4^{th} order polynomial $f(c\alpha, s\alpha)$.

$$\begin{aligned} f = {} & h_1 c\alpha^4 + (h_2 s\alpha + h_3) c\alpha^3 + (h_4 s\alpha^2 + h_5 s\alpha + h_6) c\alpha^2 \\ & + (h_7 s\alpha^3 + h_8 s\alpha^2 + h_9 s\alpha + h_{10}) c\alpha \\ & + h_{11} s\alpha^4 + h_{12} s\alpha^3 + h_{13} s\alpha^2 + h_{14} s\alpha + h_{15} \end{aligned} \tag{4.83}$$

[3] Alternatively, the two solutions can be found algebraically by substituting the Rodrigues formula for $\mathbf{C}(\mathbf{e}_2, \beta)$ in (4.76).

where the coefficients h_l, $l = 1 \ldots 15$, are functions of the measured quantities $\mathbf{v}_i$, $\mathbf{u}_i$ and w_i, $i = 2, 3$.

Using the trigonometric constraint $c\alpha^2 + s\alpha^2 = 1$ and the Sylvester resultant [15], we eliminate the variable $s\alpha$ from (4.83), and arrive at an 8^{th} order univariate polynomial [94]:

$$g(s\alpha) = \sum_{i=0}^{8} k_i s\alpha^i. \tag{4.84}$$

which we solve analytically using the companion matrix [15]. Back-substituting each of the 8 solutions for $s\alpha$ in equation (4.83), we get one solution for $c\alpha$, because (4.83) is linear in $c\alpha$ after replacing all even order terms of $c\alpha^{2k}$ by $(1 - s\alpha^2)^k$ and $c\alpha^3$ by $(1 - s\alpha^2)c\alpha$. Finally, each pair of solutions for $c\alpha$ and $s\alpha$ corresponds to one solution for $c\gamma$ and $s\gamma$ using equation (4.82).

We now prove that the two particular solutions of β lead to the same set of 8 solutions for the rotation matrix. Let the two particular rotation matrices be $\mathbf{C}_1 = \mathbf{C}(\mathbf{e}_2, \beta_1 - \beta_2)$ and $\mathbf{C}_2 = \mathbf{C}(\mathbf{e}_2, \beta_1 + \beta_2)$. From the geometry of Fig. 4.4, we have:

$$\mathbf{C}_1\mathbf{u}_1 = \mathbf{C}(\mathbf{v}_1, 180^\circ)\mathbf{C}_2\mathbf{u}_1 \tag{4.85}$$

$$\Rightarrow \mathbf{C}_1^T\mathbf{C}(\mathbf{v}_1, 180^\circ)\mathbf{C}_2\mathbf{u}_1 = \mathbf{u}_1 \tag{4.86}$$

Therefore,

$$\mathbf{C}_1^T\mathbf{C}(\mathbf{v}_1, 180^\circ)\mathbf{C}_2 = \mathbf{C}(\mathbf{u}_1, \theta) \tag{4.87}$$

where θ as shown in the Appendix, equals to 180°, i.e.,

$$\mathbf{C}_2 = \mathbf{C}(\mathbf{v}_1, -180^\circ)\mathbf{C}_1\mathbf{C}(\mathbf{u}_1, 180^\circ). \tag{4.88}$$

Given 8 solutions for the rotational matrix C corresponding to the particular solution $\mathbf{C}_1$:

$$\mathbf{C}_{1,i} = \mathbf{C}(\mathbf{v}_1, \alpha_i)\mathbf{C}_1\mathbf{C}(\mathbf{u}_1, \gamma_i), \quad i = 1 \ldots 8 \tag{4.89}$$

and another 8 solutions corresponding to the particular solution $\mathbf{C}_2$

$$\mathbf{C}_{2,j} = \mathbf{C}(\mathbf{v}_1, \alpha_j)\mathbf{C}_2\mathbf{C}(\mathbf{u}_1, \gamma_j), \quad j = 1 \dots 8, \tag{4.90}$$

we assume that at least one of the $\mathbf{C}_{2,j}$'s is different from all the $\mathbf{C}_{1,i}$'s, i.e., there exists at least one j, e.g., $j = g$ such that $\mathbf{C}_{2,g} \neq \mathbf{C}_{1,i}$, $\forall i$. We will show that this will lead to a contradiction. Specifically, substituting (4.88) in (4.90) for $j = g$, we have:

$$\begin{aligned} \mathbf{C}_{2,g} &= \mathbf{C}(\mathbf{v}_1, \alpha_g)\mathbf{C}(\mathbf{v}_1, -180^\circ)\mathbf{C}_1\mathbf{C}(\mathbf{u}_1, 180^\circ)\mathbf{C}(\mathbf{u}_1, \gamma_g) \\ &= \mathbf{C}(\mathbf{v}_1, \alpha_g - 180^\circ)\mathbf{C}_1\mathbf{C}(\mathbf{u}_1, \gamma_g + 180^\circ). \end{aligned} \tag{4.91}$$

However, this would mean that we have a 9^{th} solution stemming from $\mathbf{C}_1$, which is impossible.

4.9 Analytical Solutions for Systems 11, 12, and 13

Up to now, we have derived closed-form or analytical (resulting in univariate polynomials) solutions for Systems 1–2 and 5–10. However, for Systems 11–13, it is very challenging to follow a similar process. Fortunately, recent progress in algebraic geometry has provided a symbolic-numerical method for constructing the multiplication matrix from whose eigenvectors, we can read off the solutions. In the following sections, we derive the polynomial equations for Systems 11, 12, and 13, and then present the symbolic-numerical method [63] we employed for solving these systems.

4.9.1 System 11: Measurements $\{\mathbf{b}_1; \mathbf{b}_3; d_{56}; d_{78}\}$

For this problem, we first write the unknown d_{12} as a function of the rotation matrix $\mathbf{C}$. Then by substituting it into the geometric constraints for d_{56} and d_{78}, we form a system of polynomials only in the rotation parameters. Even though the total degree of the resulting system is higher than before eliminating d_{12}, it reduces the problem size and makes the computations faster, as we will see later on.

Specifically, from the geometric constraint involving the bearing measurements $\mathbf{b}_1$ and $\mathbf{b}_3$ [see (4.58)], we eliminate the unknown d_{34} by premultiplying both sides with $\lfloor \mathbf{b}_3 \times \rfloor$ and solve for d_{12} [see (4.5)]:

$$d_{12} = \mathbf{v}_1^T \mathbf{C}^2 \mathbf{p}_4 + a \tag{4.92}$$

where

$$\mathbf{v}_1 = -\frac{(\lfloor \mathbf{b}_3 \times \rfloor \mathbf{b}_1)^T \lfloor \mathbf{b}_3 \times \rfloor}{(\lfloor \mathbf{b}_3 \times \rfloor \mathbf{b}_1)^T (\lfloor \mathbf{b}_3 \times \rfloor \mathbf{b}_1)}, \quad a = -\mathbf{v}_1^T {}^1\mathbf{p}_3. \tag{4.93}$$

We then substitute d_{12} [see (4.92)] into the following two distance constraints:

$$(\mathbf{p} + \mathbf{C}^2 \mathbf{p}_6 - {}^1\mathbf{p}_5)^T (\mathbf{p} + \mathbf{C}^2 \mathbf{p}_6 - {}^1\mathbf{p}_5) = d_{56}^2 \tag{4.94}$$

$$(\mathbf{p} + \mathbf{C}^2 \mathbf{p}_8 - {}^1\mathbf{p}_7)^T (\mathbf{p} + \mathbf{C}^2 \mathbf{p}_8 - {}^1\mathbf{p}_7) = d_{78}^2. \tag{4.95}$$

The resulting equations are functions of only $\mathbf{C}$

$$(\mathbf{v}_1^T \mathbf{C}^2 \mathbf{p}_4)^2 + 2(\mathbf{v}_1^T \mathbf{C}^2 \mathbf{p}_4)(\mathbf{b}_1^T \mathbf{C}^2 \mathbf{p}_6) + 2\mathbf{b}_1^T \mathbf{u}_1 (\mathbf{v}_1^T \mathbf{C}^2 \mathbf{p}_4) + 2\mathbf{u}_1^T \mathbf{C}^2 \mathbf{p}_6 + \epsilon_1 = 0 \tag{4.96}$$

$$(\mathbf{v}_1^T \mathbf{C}^2 \mathbf{p}_4)^2 + 2(\mathbf{v}_1^T \mathbf{C}^2 \mathbf{p}_4)(\mathbf{b}_1^T \mathbf{C}^2 \mathbf{p}_8) + 2\mathbf{b}_1^T \mathbf{u}_2 (\mathbf{v}_1^T \mathbf{C}^2 \mathbf{p}_4) + 2\mathbf{u}_2^T \mathbf{C}^2 \mathbf{p}_8 + \epsilon_2 = 0 \tag{4.97}$$

where

$$\mathbf{u}_1 = a\mathbf{b}_1 - {}^1\mathbf{p}_5, \qquad \epsilon_1 = {}^2\mathbf{p}_6^T {}^2\mathbf{p}_6 + \mathbf{u}_1^T \mathbf{u}_1 - d_{56}^2$$

$$\mathbf{u}_2 = a\mathbf{b}_1 - {}^1\mathbf{p}_7, \qquad \epsilon_2 = {}^2\mathbf{p}_8^T {}^2\mathbf{p}_8 + \mathbf{u}_2^T \mathbf{u}_2 - d_{78}^2.$$

Finally, let $\mathbf{v}_2 = \mathbf{b}_1 \times \mathbf{b}_3$. Projecting (4.58) to $\mathbf{v}_2$ results in a third equation in $\mathbf{C}$

$$\mathbf{v}_2^T \mathbf{C}^2 \mathbf{p}_4 - \mathbf{v}_2^T {}^1\mathbf{p}_3 = 0. \tag{4.98}$$

Now, we have three equations (4.96)–(4.98) to solve for the 3-DOF rotation matrix $\mathbf{C}$.

We choose the (non-Hamiltonian) quaternion to parameterize orientation defined as $\bar{q} = [q_1 \; q_2 \; q_3 \; q_4]^T = [\mathbf{q}^T \; q_4]^T$, and related to the rotational matrix as[4]

$$\mathbf{C}(\bar{q}) = \mathbf{I}_3 - 2q_4 \lfloor \mathbf{q} \times \rfloor + 2\lfloor \mathbf{q} \times \rfloor^2. \tag{4.99}$$

[4] Note that since the quaternions $\bar{q}$ and $-\bar{q}$ both represent the same rotation, the number of solutions is double. One can easily eliminate the redundant solutions by discarding those with $q_4 < 0$.

Substituting (4.99) in (4.96) and (4.97) yields two 4^{th} order polynomials, while (4.98) is quadratic. Together with the unit-norm constraint

$$\bar{q}^T \bar{q} - 1 = 0 \tag{4.100}$$

forms a square system, which we can solve for $\bar{q}$. Once $\bar{q}$, and hence $\mathbf{C}$, is known (see Section 4.9.4), we find d_{12} by back-substituting in (4.92).

4.9.2 System 12: Measurements $\{\mathbf{b}_1; \mathbf{b}_4; d_{56}; d_{78}\}$

In this case, we cannot easily solve for d_{12} and $\mathbf{C}$ separately, because monomials containing both d_{12} and elements of $\mathbf{C}$ appear in the system. Instead, we form a square system of polynomial equations based on the available measurement constraints which we solve using the method described in Section 4.9.4.

Specifically, multiplying (4.43) by $\mathbf{C}^T$ yields

$$d_{12}\mathbf{C}^T\mathbf{b}_1 + {}^2\mathbf{p}_4 + d_{34}\mathbf{b}_4 - \mathbf{C}^T{}^1\mathbf{p}_3 = 0. \tag{4.101}$$

Then we eliminate d_{34} by forming the dot product with two unit vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ both of which are perpendicular to $\mathbf{b}_4$, and obtain two equations:

$$d_{12}\mathbf{v}_i^T\mathbf{C}^T\mathbf{b}_1 - \mathbf{v}_i^T\mathbf{C}^T{}^1\mathbf{p}_3 + \mathbf{v}_i^T{}^2\mathbf{p}_4 = 0,\ i = 1,\ 2. \tag{4.102}$$

Expanding the distance constraints (4.94) and (4.95), we have

$$d_{12}^2 + 2d_{12}(\mathbf{b}_1^T\mathbf{C}^2\mathbf{p}_6 - \mathbf{b}_1^T{}^1\mathbf{p}_5) - 2^1\mathbf{p}_5^T\mathbf{C}^2\mathbf{p}_6 + \epsilon_1 = 0 \tag{4.103}$$

$$d_{12}^2 + 2d_{12}(\mathbf{b}_1^T\mathbf{C}^2\mathbf{p}_8 - \mathbf{b}_1^T{}^1\mathbf{p}_7) - 2^1\mathbf{p}_7^T\mathbf{C}^2\mathbf{p}_8 + \epsilon_2 = 0 \tag{4.104}$$

where

$$\epsilon_1 = {}^2\mathbf{p}_6^T{}^2\mathbf{p}_6 + {}^1\mathbf{p}_5^T{}^1\mathbf{p}_5 - d_{56}^2,\ \epsilon_2 = {}^2\mathbf{p}_8^T{}^2\mathbf{p}_8 + {}^1\mathbf{p}_7^T{}^1\mathbf{p}_7 - d_{78}^2.$$

Equations (4.102)–(4.104) and the unit-norm constraint (4.100) form a square system of five polynomial equations allowing us to solve for the relative pose.

4.9.3 System 13: Measurements $\{\mathbf{b}_1;\ d_{34};\ d_{56};\ d_{78};\ d_{9,10}\ \}$

From the bearing and distances measurements, we have the following constraints [see (4.103)], $i = 2, \ldots, 5$

$$d_{12}^2 + 2d_{12}(\mathbf{b}_1^T \mathbf{C}^2 \mathbf{p}_{2i} - \mathbf{b}_1^{T\,1}\mathbf{p}_{2i-1}) - 2\,{}^1\mathbf{p}_{2i-1}^T \mathbf{C}^2 \mathbf{p}_{2i} + \epsilon_i = 0$$

where $\epsilon_i = {}^2\mathbf{p}_{2i}^{T\,2}\mathbf{p}_{2i} + {}^1\mathbf{p}_{2i-1}^T{}^1\mathbf{p}_{2i-1} - d_{2i-1,2i}^2$. Together with the unit-norm constraint (4.100), we have a square system of five polynomial equations to solve for the relative pose.

4.9.4 Symbolic-Numeric Solution Method

In this section, we describe Reid and Zhi's symbolic-numeric method [63] for solving systems of multivariate polynomial equations and its application to our problems. This method computes the solutions via the multiplication matrix. The eigenvalues and eigenvectors of the multiplication matrix directly relate to the solutions of the system of polynomial equations. Specifically, if a system of polynomial equations $F = \{f_1, ..., f_n\}$ in n variables $x_1, ..., x_n$ has m distinct solutions, then the remainder space of any polynomial divided by F is a m dimensional vector space spanned by the basis $\mathbf{x}_B = [b_1, ..., b_m]^T$. Elements of the basis $\mathbf{x}_B$ are monomials or even polynomials. Multiplication of any polynomial and the basis $\mathbf{x}_B$ is the sum of the remainders and polynomial combinations of f_i, $i = 1, ...n$. For example, multiplying x_1 with $\mathbf{x}_B$ yields

$$x_1\mathbf{x}_B = \begin{bmatrix} x_1 b_1 \\ \vdots \\ x_1 b_m \end{bmatrix} = \mathbf{M}_{x_1}\mathbf{x}_B + \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mn} \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} \tag{4.105}$$

where h_{ji}'s, $j = 1, ..., m$ and $i = 1, ..., n$ are polynomials, and $\mathbf{M}_{x_1}$ is the multiplication matrix. Evaluating (4.105) at the roots of F, $\{\mathbf{z}_j : f_i(\mathbf{z}_j) = 0,\ i = 1, ..., n\ j = 1, ..., m\}$, yields

$$z_1\mathbf{x}_B(\mathbf{z}_j) = \mathbf{M}_{x_1}\mathbf{x}_B(\mathbf{z}_j). \tag{4.106}$$

As evident, the eigenvalues of $\mathbf{M}_{x_1}$ are the x_1-coordinates of the solutions, and its eigenvectors are the basis vectors $\mathbf{x}_B$ evaluated at the solutions.

The classical method for computing the multiplication matrix requires computing the Gröbner basis [15] of the ideal defined by F in order to determine the basis $\mathbf{x}_B$ and compute the remainders of $x_1\mathbf{x}_B$ divided by F. However, algorithms for computing the Gröbner basis of a system of polynomial equations with floating point coefficients often fail, because numerical round off errors accumulate quickly. Recent techniques for computing the multiplication matrix avoid the step of computing the Gröbner basis, and instead seek to generate the multiplication matrix using polynomial combinations of the original polynomials of F. These methods are more stable numerically. Reid and Zhi's symbolic-numeric method falls into this second category.

We start by expanding the equations of the original system by multiplying them with monomials up to a certain total degree to form a new system

$$\mathbf{\Xi}_t\mathbf{x}_t = \mathbf{\Xi}_t[x_1^t, x_1^{t-1}x_2, ..., x_n^2, x_1, ..., x_n, 1]^T = \mathbf{0} \tag{4.107}$$

where $\mathbf{\Xi}_t$ is a matrix containing all the coefficients of the expanded polynomial equations, and $\mathbf{x}_t$ is a vector of monomials up to total degree t, whose dimension is $n_t = \binom{n+t}{n}$. Therefore, the solutions of the system of polynomial equations must lie in the right null space of $\mathbf{\Xi}_t$, which will be used to construct the multiplication matrix. The total degree t is determined via a so-called involutive-form test. This test also tells us how many solutions the system has, as well as the dimension of the remainder space.

The involutive test includes prolongation and projection operations. A single prolongation of a system F means to expand it to one higher total degree. Prolongation up to total degree t is denoted by $F^{(t)}$. After each prolongation, we compute the null space of the coefficient matrix $\mathbf{\Xi}_t$. A single projection means to remove rows of the vectors spanning the null space of $\mathbf{\Xi}_t$ such that they correspond to monomials with one lower total degree, i.e., $\mathbf{x}_t$ reduces to $\mathbf{x}_{t-1}$. One projection of a system F is denoted by $\pi(F)$, and higher projection orders are denoted by $\pi^i(F)$, $i = 2, \ldots, t$.

A system F is involutive at order t and projection order ℓ if and only if $\pi^{\ell}(F^{(t)})$ satisfies [63]:

$$dim\ \mathbf{B}_{t-\ell}^{(t)} = dim\ \mathbf{B}_{t-\ell-1}^{(t)} = dim\ \mathbf{B}_{t-\ell-1}^{(t+1)} \tag{4.108}$$

where $\mathbf{B}_{t-\ell}^{(t)}$, $\mathbf{B}_{t-\ell-1}^{(t)}$, and $\mathbf{B}_{t-\ell-1}^{(t+1)}$ are the null spaces of $\pi^{\ell}(F^{(t)})$, $\pi^{\ell+1}(F^{(t)})$, and $\pi^{\ell+1}(F^{(t+1)})$ respectively, and dim denotes the dimension of the corresponding null space. From now on, we will focus on $\mathbf{B}_{t-\ell}^{(t)}$ and $\mathbf{B}_{t-\ell-1}^{(t)}$ and for clarity drop the superscript (t).

This involutive test only needs to be performed once offline. Once we know the total degree t to which to expand the system, we use it to construct $\mathbf{\Xi}_t$ for all instances of the problem. Note that $\mathbf{\Xi}_t$ is stored in symbolic form and is used to compute the multiplication matrix with respect to an unknown variable, in our case ${q_4}^5$.

Once the expanded system coefficient matrix $\mathbf{\Xi}_t$ is known, the multiplication matrix $\mathbf{M}_{q_4}$ is constructed from the right null space $\mathbf{B}_t$ of $\mathbf{\Xi}_t$. Specifically, we first perform ℓ projections on $\mathbf{B}_t$ by removing the first $\binom{n+t}{n} - \binom{n+t-\ell}{n}$ rows of $\mathbf{B}_t$. The remaining portion of $\mathbf{B}_t$ is spanned by the columns of $\mathbf{B}_{t-\ell}$, which we further partition as

$$\mathbf{B}_{t-\ell} = \begin{bmatrix} \mathbf{B}_r \\ \\ \mathbf{B}_{t-\ell-1} \end{bmatrix} \tag{4.109}$$

where $\mathbf{B}_{t-\ell-1}$ corresponds to monomials up to total degree $t-\ell-1$ and $\mathbf{B}_r$ corresponds to monomials of total degree exactly $t-\ell$. Since $\mathbf{B}_t$ is the null space of $\mathbf{\Xi}_t$, $\mathbf{x}_t(\mathbf{z}_j)$ is spanned by $\mathbf{B}_t$. Similarly, $\mathbf{x}_{t-\ell}(\mathbf{z}_j)$ and $\mathbf{x}_{t-\ell-1}(\mathbf{z}_j)$ are spanned by $\mathbf{B}_{t-\ell}$ and $\mathbf{B}_{t-\ell-1}$, respectively. Furthermore, the column dimensions of $\mathbf{B}_{t-\ell}$ and $\mathbf{B}_{t-\ell-1}$ are both m, and thus,

$$\mathbf{x}_{t-\ell}(\mathbf{z}_j) = \mathbf{B}_{t-\ell}\mathbf{c}_j \tag{4.110}$$

$$\mathbf{x}_{t-\ell-1}(\mathbf{z}_j) = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{x}_{t-\ell}(\mathbf{z}_j) = \mathbf{B}_{t-\ell-1}\mathbf{c}_j \tag{4.111}$$

[5] In general, we can choose any variable as the multiplier as long as the solutions for this variable are distinct (see Ch. 2, Proposition 4.7 in [15]). In our specific problems at hand, we cannot choose d_{12}, because its solution multiplicity is at least two, since each solution for d_{12} corresponds to solutions $\bar{q}$ and $-\bar{q}$.

where $\mathbf{c}_j$ is some coefficient vector.

We then form the SVD of $\mathbf{B}_{t-\ell-1}$

$$\mathbf{B}_{t-\ell-1} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \end{bmatrix} \mathbf{V}^T \tag{4.112}$$

such that $\mathbf{U}_1$ is of dimension $\binom{n+t-\ell-1}{n} \times m$, and $\mathbf{S}$ is a square diagonal matrix. Moreover, based on the involutive test, $\mathbf{B}_{t-\ell-1}$ is of rank m, i.e., $\mathbf{S}$ is invertible.

Following [63], we choose the basis for the remainder space as,

$$\mathbf{x}_B = \mathbf{U}_1^T \mathbf{x}_{t-\ell-1} \tag{4.113}$$

and compute the multiplication matrix as

$$\mathbf{M}_{q_4} = \mathbf{U}_1^T \mathbf{B}_{q_4} \mathbf{V} \mathbf{S}^{-1} \tag{4.114}$$

where $\mathbf{B}_{q_4}$ is formed by the rows of $\mathbf{B}_{t-\ell}$ corresponding to monomials $q_4 \mathbf{x}_{t-\ell-1}$. Therefore, when evaluating $q_4 \mathbf{x}_{t-\ell-1}$ at the solution, we have

$$q_4 \mathbf{x}_{t-\ell-1}(\mathbf{z}_j) = \mathbf{B}_{q_4} \mathbf{c}_j. \tag{4.115}$$

It is easy to verify that $\mathbf{M}_{q_4}$ is indeed a multiplication matrix by checking whether $\mathbf{x}_B(\mathbf{z}_j)$ is an eigenvector of $\mathbf{M}_{q_4}$:

$$\begin{aligned}
\mathbf{M}_{q_4} \mathbf{x}_B(\mathbf{z}_j) &\overset{(4.114)}{=} \mathbf{U}_1^T \mathbf{B}_{q_4} \mathbf{V} \mathbf{S}^{-1} \mathbf{U}_1^T \mathbf{x}_{t-\ell-1}(\mathbf{z}_j) \\
&\overset{(4.111)}{=} \mathbf{U}_1^T \mathbf{B}_{q_4} \mathbf{V} \mathbf{S}^{-1} \mathbf{U}_1^T \mathbf{B}_{t-\ell-1} \mathbf{c}_j \\
&\overset{(4.112)}{=} \mathbf{U}_1^T \mathbf{B}_{q_4} \mathbf{c}_j \\
&\overset{(4.115)}{=} q_4 \mathbf{U}_1^T \mathbf{x}_{t-\ell-1}(\mathbf{z}_j) \\
&\overset{(4.113)}{=} q_4 \mathbf{x}_B(\mathbf{z}_j).
\end{aligned}$$

We can recover the solution $\mathbf{z}_j$ from an eigenvector $\boldsymbol{\xi}_j = c\mathbf{U}_1^T \mathbf{x}_{t-\ell-1}(\mathbf{z}_j)$ of $\mathbf{M}_{q_4}$ by

writing

$$\begin{aligned}\boldsymbol{\xi}'_j &= \mathbf{U}_1\boldsymbol{\xi}_j = c\mathbf{U}_1\mathbf{U}_1^T\mathbf{x}_{t-\ell-1}(\mathbf{z}_j) \\ &\overset{(4.111)}{=} c\mathbf{U}_1\mathbf{U}_1^T\mathbf{B}_{t-\ell-1}\mathbf{c}_j \overset{(4.112)}{=} c\mathbf{U}_1\mathbf{U}_1^T\mathbf{U}_1\mathbf{S}\mathbf{V}^T\mathbf{c}_j \\ &= c\mathbf{U}_1\mathbf{S}\mathbf{V}^T\mathbf{c}_j \overset{(4.112)}{=} c\mathbf{B}_{t-\ell-1}\mathbf{c}_j \\ &\overset{(4.111)}{=} c\mathbf{x}_{t-\ell-1}(\mathbf{z}_j) \end{aligned} \tag{4.116}$$

and by removing the scaling factor c, we can read off the solutions from the elements of $\mathbf{x}_{t-\ell-1}(\mathbf{z}_j)$.

In summary, the main steps to compute the multiplication matrix are listed in the following:

1. Compute the null space of $\boldsymbol{\Xi}_t$, so that the columns of matrix $\mathbf{B}_t$ span its null space.

2. Perform ℓ and $\ell+1$ projections on $\mathbf{B}_t$, i.e., take the rows of $\mathbf{B}_t$ that correspond to monomials up to total degree $t-\ell$ and $t-\ell-1$ to form the new matrices $\mathbf{B}_{t-\ell}$ and $\mathbf{B}_{t-\ell-1}$, respectively.

3. Compute the SVD of $\mathbf{B}_{t-\ell-1} = [\mathbf{U}_1\ \mathbf{U}_2]\cdot[\mathbf{S}\ \mathbf{0}]^T\cdot\mathbf{V}^T$.

4. Form the multiplication matrix of q_4 as $\mathbf{M}_{q_4} = \mathbf{U}_1^T\mathbf{B}_{q_4}\mathbf{V}\mathbf{S}^{-1}$, where $\mathbf{B}_{q_4}$ are the rows of $\mathbf{B}_{t-\ell}$ corresponding to monomials $q_4\mathbf{x}_{t-\ell-1}$.

5. Compute the eigenvectors $\boldsymbol{\xi}_i$ of $\mathbf{M}_{q_4}$, and recover all solutions from the elements of $\mathbf{U}_1\boldsymbol{\xi}_i$. The elements of this vector correspond to monomials in $\mathbf{x}_{t-\ell-1}$ evaluated at the solutions after proper scaling.

For System 11, the involutive condition is met at total degree $t=8$ and projection order $\ell=2$. As shown in Table 4.2, the number of solutions is 32. Since using quaternions introduces double solutions, the number of distinct solutions for the relative pose is 16. The size of the expanded coefficient matrix $\boldsymbol{\Xi}_8$ is 560×495.

Table 4.2: $dim\ \pi^{\ell}(F^{(t)})$ for System 11

	t = 4	t = 5	t = 6	t = 7	t = 8	t = 9
$\ell = 0$	40	52	59	63	66	70
$\ell = 1$	25	39	45	47	48	50
$\ell = 2$	13	25	33	33	**32**	32
$\ell = 3$	5	13	25	33	**32**	**32**
$\ell = 4$	1	5	13	25	32	32
$\ell = 5$	0	1	5	13	25	32
$\ell = 6$	0	0	1	5	13	25
$\ell = 7$	0	0	0	1	5	13
$\ell = 8$	0	0	0	0	1	5
$\ell = 9$	0	0	0	0	0	1

Both Systems 12 and 13 reach the involutive form at total degree $t = 8$, and they both have 5 variables, therefore the expanded coefficient matrices are both of size 1470×1287. However, System 12 reaches the involutive form at projection order $\ell = 3$ and has 16 solutions, while System 13 is involutive at $\ell = 1$ and has 28 solutions for the relative pose.

Since System 11 has a much smaller expanded coefficient matrix than System 12 and 13, its solution can be computed much faster. In our (unoptimized) Matlab implementation, solving System 11 takes 1.139 seconds, while solving System 12 and 13 takes 3.236 seconds and 4.397 seconds, respectively. Also note that our symbolic-numerical solvers are much faster than the iterative numerical solver, PHCpack [87]. It take PHC 2.983 seconds, 5.355 seconds, and 6.487 seconds to solve System 11, 12, and 13, respectively.

4.9.5 System 14: Measurements $\{d_{12};\ d_{34};\ d_{56};\ d_{78};\ d_{9,10};\ d_{11,12}\}$

From the six distance measurements, we have the following equations:

$$\mathbf{p}^T\mathbf{p} - d_{12}^2 = 0$$
$$(\mathbf{p} + \mathbf{C}^2\mathbf{p}_{2i} - {}^1\mathbf{p}_{2i-1})^T(\mathbf{p} + \mathbf{C}^2\mathbf{p}_{2i} - {}^1\mathbf{p}_{2i-1}) - d_{2i-1,2i}^2 = 0$$

where $i = 2, \ldots, 5$. This problem has been studied extensively in the literature, see [88, 45, 84] for its solution.

4.10 Simulation Results

We have evaluated the performance of our algorithms in simulation for different values of inter-robot measurement noise variance, but omit tests on noise in the robots' ego-motion estimates, because the effect of perturbing the robots' ego-motion estimates is very similar to that of perturbing the inter-robot measurements.

The data for our simulations are generated as follows. First, we generate random robot trajectories in 3D, with the two robots starting at initial positions 1 m $\sim$ 2 m apart from each other, and moving 3 m $\sim$ 6 m between obtaining distance and/or bearing measurements. We perturb the true bearing direction to generate the bearing measurements. The perturbed bearing vectors are uniformly distributed in a cone with the true bearing as its main axis. The angle between the true vector and the boundary of the cone is defined as σ_b. The noise in the distance measurement is assumed zero-mean white Gaussian with standard deviation $\sigma_d = 10\sigma_b$ m.

We conduct Monte Carlo simulations for different values of σ_b, and report the averaged results of 1000 trials per setting for each system. We report the error in position as the 2-norm of the difference between the true and the estimated position[6] . To evaluate the error in the relative orientation, we use a multiplicative error model for the quaternion corresponding to the rotation matrix. In particular, true orientation, $\bar{q}$, estimated orientation, $\hat{\bar{q}}$, and error quaternion, $\delta\bar{q}$ are related via [83]:

$$\bar{q} = \delta\bar{q} \otimes \hat{\bar{q}} \tag{4.117}$$

where $\delta\bar{q}$ describes the small rotation that makes the estimated and the true orientation

[6] Since we focus on assessing the accuracy of the minimal problem solver, out of the multiple solutions, we choose as estimate the one closest to the true value. In practice, additional measurements are used to uniquely determine the position [93].

coincide. Using the small-angle approximation, the error quaternion can be written as

$$\delta\bar{q} \simeq \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}^T & 1 \end{bmatrix}^T \Leftrightarrow \mathbf{C} \simeq (\mathbf{I}_3 - \lfloor \delta\boldsymbol{\theta} \times \rfloor)\hat{\mathbf{C}} \tag{4.118}$$

and the 2-norm of $\delta\boldsymbol{\theta}$ is used to evaluate the orientation error.

Fig. 4.5 shows the orientation error and position error, respectively, as a function of the bearing noise σ_b for Systems 1, 2 and 5–13. The curves depict the median of the error in the 1000 trials, and the vertical bars show the 25 and 75 percentiles. As expected, the error increases as the standard deviation of the noise increases. We also see that the 75 percentiles are growing much faster than the 25 percentiles for most systems except for the position error of Systems 1, 6, 7, and 10. This indicates that the probability of having larger error in the relative pose estimate increases dramatically with the variance of the measurement noise. In contrast, the distribution of the position errors of Systems 1, 6, 7, and 10 remains almost the same, because it is directly measured from the distance d_{12} and bearing $\mathbf{b}_1$. Furthermore, systems with more bearing measurements achieve better accuracy than systems with more distance measurements. In particular, Systems 10 and 13 perform significantly worse than the other two systems in their group [see Figs. 4.5(e) and 4.5(g)]. Finally, we see that in the absence of measurement noise, we can recover the relative pose perfectly.

Lastly, we note that in all practical cases the solutions of the minimal problems should be used in conjunction with RANSAC [26] to perform outlier rejection followed by nonlinear least squares so as to improve the estimation accuracy using all available measurements.

4.11 Experimental Results

In this section, we describe a real-world experiment performed to further validate our extrinsic calibration algorithms. In the experiment, we use a Point Grey Dragonfly camera and an IDS uEye camera to mimic two robots moving in 3D (see Fig. 4.6). The

Table 4.3: Minimal Systems' Estimation Errors Compared to the Least-Squares Solution

	$\mathbf{p}$	$\mathbf{q}$	$\|\|\delta\mathbf{p}\|\|$ (m)	$\|\|\delta\boldsymbol{\theta}\|\|$ (rad)
LS	-0.1618, 0.0864, 0.5122	-0.0923, -0.9744, 0.0900, 0.1841	0	0
sys1	-0.1465, 0.1198, 0.5795	-0.4776, -0.8581, 0.0148, 0.1878	0.0766	0.4008
sys2	-0.1491, 0.1219, 0.5897	-0.0885, -0.9818, 0.0626, 0.1558	0.0862	0.0402
sys5	-0.1282, 0.1048, 0.5070	-0.1225, -0.9778, 0.0589, 0.1596	0.0387	0.0499
sys6	-0.1465, 0.1198, 0.5795	-0.1948, -0.9805, 0.0255, 0.0084	0.0766	0.135
sys7	-0.1465, 0.1198, 0.5795	-0.0706, -0.9937, 0.0409, 0.0774	0.0766	0.1207

two cameras are moved by hand such that they face each other and at the same time observe point features on a board placed behind each camera. These features are used to compute the cameras' ego-motion. Specifically, as depicted in Fig. 4.7, $\{C_1\}$ and $\{C_2\}$ denote the frames of reference for the Dragonfly and the uEye cameras, respectively. The pose of $\{C_1\}$ ($\{C_2\}$) in the global frame of reference $\{G_1\}$ ($\{G_2\}$) is determined by tracking features with known 3D coordinates in frame $\{G_1\}$ ($\{G_2\}$). In particular, we employ the direct least-squares solution for the PnP problem [31] to solve the camera pose given the image coordinates of known 3D points. After the pose of each camera in its global frame of reference is known, its transformation with respect to its initial frame (i.e., the quantities ${}^1\mathbf{p}_{2i+1}, {}^1_{2i+1}\mathbf{C}$, and ${}^2\mathbf{p}_{2i+2}, {}^2_{2i+2}\mathbf{C}$, $i = 1, \ldots, 4$) is easily obtained.

The robot-to-robot distance and bearing measurements are obtained from images of the ping-pong ball mounted on top of each camera. Specifically, we first extract the edges of the ping-pong ball using the Canny edge detector [11]. Then we fit a circle to the edge pixels using least squares. From the center and radius of the circle, we can measure both the bearing and range to the center of the ball from the camera, because the radius of the ball is known to be 20 mm. Finally, given the transformation between the camera $\{C_1\}$ to the ball $\{B_1\}$, and $\{C_2\}$ to $\{B_2\}$, we can compute the range and bearing between the two cameras. Therefore, we have both range and two bearings at all time steps, which allows us to pick and choose any measurement combination

necessary for evaluating our extrinsic calibration algorithms.

Due to lighting conditions and motion blur, there are outliers in the ego-motion and robot-to-robot measurements. We perform RANSAC to eliminate outliers using the solutions of System 2 as hypothesis generator. We then refine the solution of the minimal System 2 using all the inliers by employing a least-squares algorithm, after which the accuracy of the robot-to-robot transformation is significantly improved.

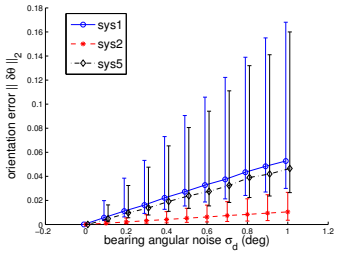
To visualize the accuracy of the robot-to-robot relative pose estimates, we transform point features expressed in the global frame $\{G_2\}$ into $\{G_1\}$ using the estimated robot-to-robot transformation and the poses of the two cameras in their respective global frames. Then, we project all 3D features back onto a test image that has both feature boards in its field of view (see Fig. 4.8). Note that in this image, we can only see the front of the board with the 4 black squares, while the one with the 18 squares is facing in the opposite direction. The camera pose of this image is determined by using the 16 corner features of the 4 black squares. The reprojection of these 16 features (reflecting the accuracy of the PnP solution) are marked by blue stars, and the reprojection of the 4×18 features (reflecting the accuracy of the estimated robot-to-robot transformation) on the other board are marked by red crosses. We can see that the blue stars are at the corners of the 4 square targets, so the camera pose is very accurate. However, the red crosses in Fig. 4.8(a) are not precisely reprojected onto the white board. This is because the initial solution of System 2 has large errors in the robot-to-robot transformation. The robot-to-robot transformation becomes significantly more accurate after a least-squares refinement. In Fig. 4.8(b), the red crosses are all on the white board, although they are slightly shifted to the left.

Since the least-squares solution is the most accurate estimate of the robot-to-robot transformation, we use it as ground truth to assess the accuracy of the the minimal solutions of Systems 1–2 and 5–7. The computed estimates and the corresponding error norms are shown in Table 4.3. The systems with mostly bearing measurements (Systems 2 and 5) appear to have the highest orientation accuracy. The orientation

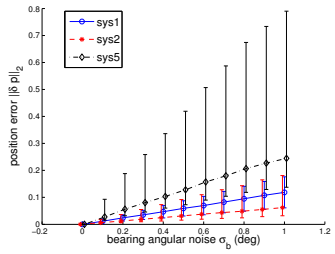
error of System 1 is particularly large (0.4008 rad). This is mainly due to the error in the robot-to-robot distance measurements. Finally, we should note that as in the simulation results, System 2 is the most resilient to measurement noise.

4.12 Summary

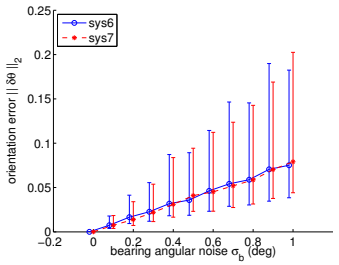
In this chapter, we address the problem of computing relative robot-to-robot 3D translation and rotation using any combination of inter-robot measurements and robot egomotion estimates. We have shown that there exist 14 base minimal systems which result from all possible combinations of inter-robot measurements. Except the two singular cases, Systems 3 and 4, we presented closed-form (algebraic) and analytical solutions to the remaining ones (see Fig. 4.1). A key advantage of the described methods is that they are significantly faster than other pure numerical approaches, such as homotopy continuation [87], since they require no iterations. Moreover, they can be used in conjunction with RANSAC for outlier rejection and for computing an initial estimate for the unknown robot-to-robot 3D transformation, which can be later refined using nonlinear least squares.



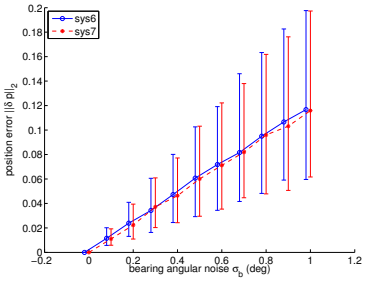
(a) Sys 1, 2, and 5: Orientation error



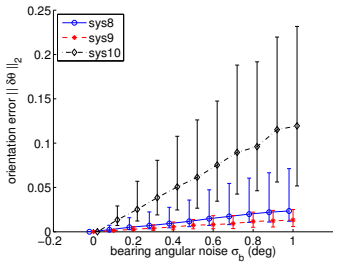
(b) Sys 1, 2, and 5: Position error



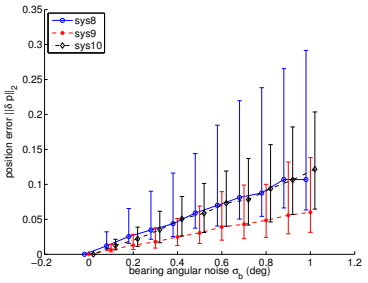
(c) Sys 6 and 7: Orientation error



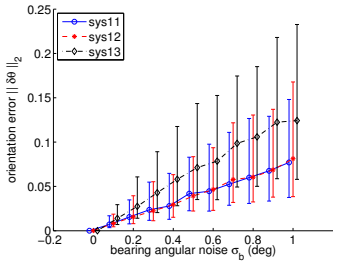
(d) Sys 6 and 7: Position error



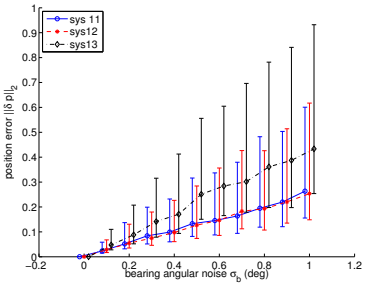
(e) Sys 8, 9, and 10: Orientation error



(f) Sys 8, 9, and 10: Position error



(g) Sys 11, 12 and 13: Orientation error



(h) Sys 11, 12, and 13: Position error

Figure 4.5: Orientation and position errors as functions of the bearing-measurement noise. The plots show the median and 25–75% quartiles in 1000 trials.

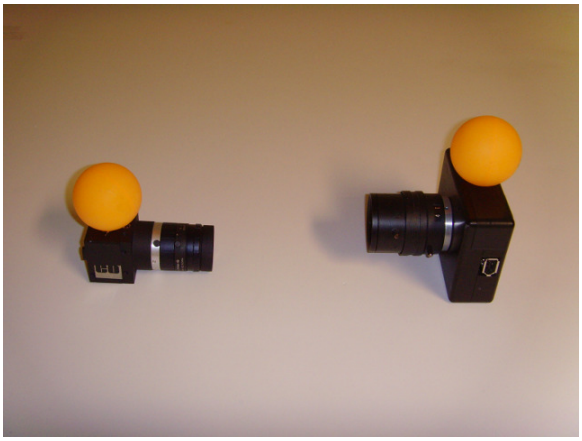


Figure 4.6: The two cameras used in the experiment. The Point Grey Dragonfly camera is shown on the right, and the IDS uEye camera is on the left. The two ping-pong balls mounted on top of the two cameras are used as visual targets for measuring the distance and bearing between the two cameras.

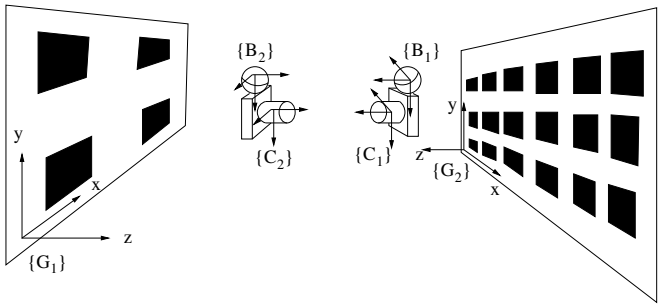
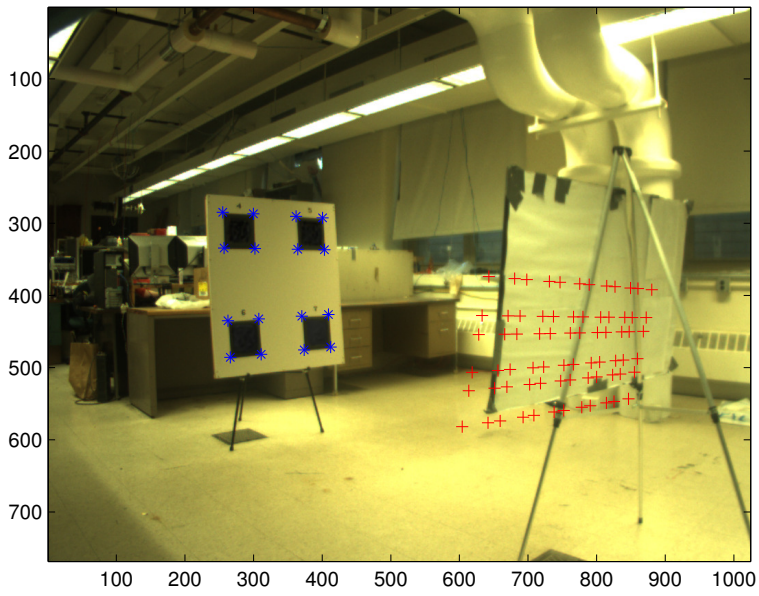
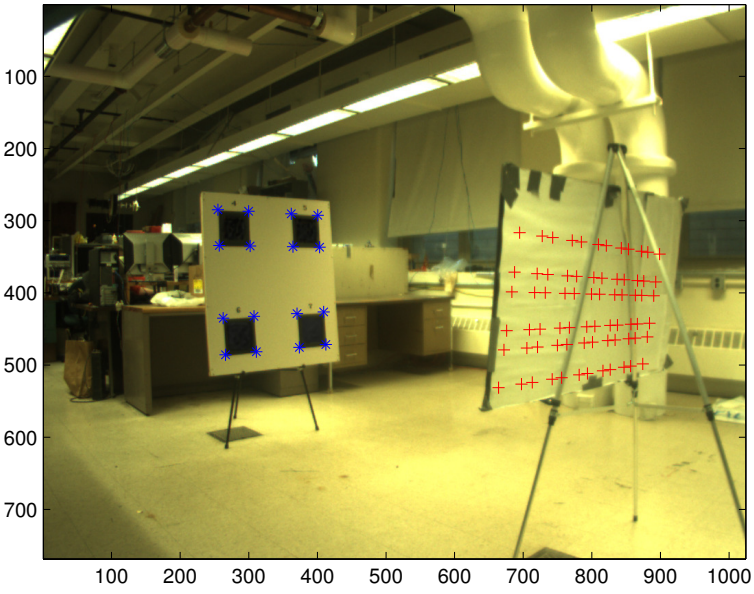


Figure 4.7: The experimental setup. The Dragonfly camera, $\{C_1\}$, and the uEye camera, $\{C_2\}$, are looking at each other. The Dragonfly (uEye) camera uses the known features in $\{G_1\}$ ($\{G_2\}$) to compute its ego-motion. The Dragonfly (uEye) measures the distance and bearing to the uEye (Dragonfly) by detecting the ping-pong ball $\{B_2\}$ ($\{B_1\}$).



(a) Initial solution of System 2



(b) Solution of System 2 after least-squares refinement

Figure 4.8: Reprojected 3D feature points. The blue stars "*" are the back-projections of features in $\{G_1\}$. The red crosses "+" are the back-projections of features in $\{G_2\}$ which are expressed in frame $\{G_1\}$ using the estimated robot-to-robot transformation from: (a) the minimal solution of System 2; (b) the least-squares solution of System 2 initialized using the solution to the corresponding minimal problem.

Chapter 5

Relative Pose Estimation from Feature Point Matches and a Directional Correspondence

In this chapter, we present methods for determining the robot-to-robot relative pose using feature point correspondences and one directional reference. We provide efficient closed-form solutions for the minimal and analytical solutions for the over-determined problems. Parts of this chapter have been published in a journal paper [55].

5.1 Introduction

In the previous chapter, we described methods for determining the relative robot-to-robot pose using combinations of inter-robot distance and bearing measurements. These methods do not require any external references, therefore can be applied in any environment. However, robots often navigate in environments that are full of texture and structures which have been widely used for localization. Specifically, the 5-DOF relative camera pose (where the scale of the translation is unobservable) can be determined by

five or more feature correspondences [57]. The main challenge of using feature correspondences is detecting and matching feature points. Especially, cluttered environments with independently moving objects yield many erroneous feature correspondences which have to be removed. As mentioned in the previous chapter, RANSAC [26] is a reliable algorithm for removing outliers. In RANSAC, it is desirable to have a hypothesis generator that uses a small number of data points to generate a finite set of candidate solutions, since this minimizes the probability of choosing an outlier as part of the data.

In this work, we first derive a closed-form solution to the "three-plus-one" problem of estimating the 5-DOF relative camera pose from 3 point correspondences and one common direction. Subsequently, we derive an exact and an approximate analytical solution to the least-squares version of this problem, when more than 3 point correspondences are available.

5.2 Related Work

Nister [57] first presented an analytical solution to the minimal problem of determining the 5-DOF camera-to-camera pose using 5 point correspondences. This algorithm employs the idea of elimination and back-substitution, which requires computing the null space of a 5×9 matrix, performing a Gaussian elimination on a dense 10×20 matrix, and solving a 10^{th} order univariate polynomial. Finally, by back-substitution, the essential matrix is found and a total of 10 solutions of the camera relative pose are recovered from the essential matrix.

Directional references are often available from points at infinity or from the gravity measured by accelerometers, and can be used to reduce the number of feature point correspondences required for computing the relative camera pose. Since 2-DOF of rotation can be determined using a directional reference, the problem is simplified to that of computing 1-DOF of rotation and 2-DOF of translation from 3 point correspondences.

The method presented by Fraundorfer *et al.* [28] is similar to that of Nister [57]. Specifically, they form a system of multi-variate polynomials with 4 solutions, which requires a null-space extraction and a Gauss-Jordan elimination of a dense 10×6 matrix to obtain the coefficients of a 4^{th} order univariate polynomial. Kalantari *et al.* [43] follow a different formulation that uses three variables to represent the translation and the tangent half-angle formula to represent the unknown rotation. Subsequently, they employ the hidden variable technique [15] to compute the coefficients of a 6^{th} order univariate polynomial and compute 6 solutions for the relative pose. However, their formulation has a singularity (rotation of 180 degrees) and 2 of the 6 solutions are spurious.

In this chapter, we provide a fast, closed-form solution to the three-plus-one problem by directly solving for the rotation and translation, instead of the essential matrix. In particular, the coefficients of the 4^{th} order polynomial are derived in closed-form as functions of the point correspondences. Finally, we describe one exact and one approximate analytical solution to the least-squares version of the same problem when $N \geq 3$ point correspondences are available.

5.3 Problem Description

Consider two cameras observing N 3D point features whose image projections are represented by the homogeneous 3D vectors $\mathbf{p}_i = [p_{ix},\ p_{iy},\ 1]^T$ and $\mathbf{p}'_i = [p'_{ix},\ p'_{iy},\ 1]^T$, $i = 1, ..., N$, expressed in the two cameras' frames of references $\{1\}$ and $\{2\}$, respectively. In addition, there is a directional reference measured in these two frames, $\mathbf{b}_1$ in frame $\{1\}$ and $\mathbf{b}_2$ in frame $\{2\}$. Our objective is to determine the rotation matrix $\mathbf{C}$ and the up-to-scale translation $\mathbf{p} = [x,\ y,\ z]^T$ between these two camera views.

The two directional measurements provide us the following constraint (see Fig. 5.1)

$$\mathbf{b}_2 = \mathbf{C}\mathbf{b}_1. \tag{5.1}$$

This directional constraint has identical structure with the robot-to-robot mutual bearing constraint (4.7) in Section 4.4.3. Therefore, we can apply the same method to solve

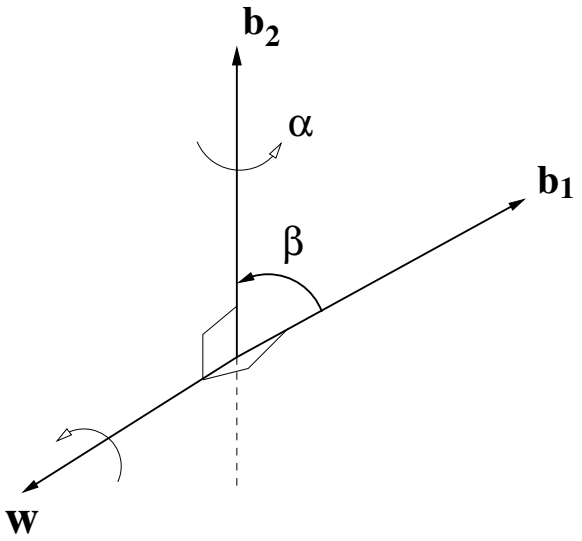


Figure 5.1: A rotation that satisfies the constraint $\mathbf{b}_2 = \mathbf{C}\mathbf{b}_1$, where $\mathbf{w} = \frac{\mathbf{b}_1 \times \mathbf{b}_2}{||\mathbf{b}_1 \times \mathbf{b}_2||}$.

this problem. First, we employ the results of Lemmas 17 and 18 to determine 2-DOF of rotation. The remaining degrees of freedom will be subsequently computed using feature point correspondences.

Using Lemma 18, we parameterize the unknown rotation matrix $\mathbf{C}$ as:

$$\mathbf{C} = \mathbf{C}(\mathbf{b}_2, \alpha)\mathbf{C}(\mathbf{w}, \beta) \tag{5.2}$$

where $\mathbf{w} = \frac{\mathbf{b}_1 \times \mathbf{b}_2}{||\mathbf{b}_1 \times \mathbf{b}_2||}$. From Lemma 17, the rotation angle β is computed as $\beta = Atan2(||\mathbf{b}_1 \times \mathbf{b}_2||, \mathbf{b}_1^T \mathbf{b}_2)$.

The remaining unknown rotation angle α and the translation $\mathbf{p}$ are determined by the feature correspondences:

$$\mathbf{p}_i'^T \lfloor \mathbf{p} \times \rfloor \mathbf{C}(\mathbf{b}_2, \alpha)\mathbf{C}(\mathbf{w}, \beta)\mathbf{p}_i = 0 \quad i = 1, ..., N. \tag{5.3}$$

In order to simplify our derivations hereafter, we first transform all measurements to a new frame of reference $\{e\}$, in which $\mathbf{w}$ and $\mathbf{b}_2$ are canonical vectors, i.e., $\mathbf{w} = [1\ 0\ 0]^T$ and $\mathbf{b}_2 = [0\ 1\ 0]^T$. The rotation matrix $\mathbf{C}_e = [\mathbf{w}\ \mathbf{b}_2\ \mathbf{w} \times \mathbf{b}_2]$ transforms the vectors

expressed in the new frame $\{e\}$ back to its original frame. Then the rotation matrix $\mathbf{C}$ is described by

$$\begin{aligned}\mathbf{C} &= \mathbf{C}_e\mathbf{C}_y(\alpha)\mathbf{C}_x(\beta)\mathbf{C}_e^T \\ &= \mathbf{C}_e \begin{bmatrix} c\alpha & 0 & s\alpha \\ 0 & 1 & 0 \\ -s\alpha & 0 & c\alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\beta & -s\beta \\ 0 & s\beta & c\beta \end{bmatrix} \mathbf{C}_e^T \end{aligned} \tag{5.4}$$

And the translation vector $\mathbf{p}$ is

$$\mathbf{p} = \mathbf{C}_e\bar{\mathbf{p}} \tag{5.5}$$

where $\bar{\mathbf{p}} = [\bar{x},\ \bar{y},\ \bar{z}]^T$ is the translation expressed in the canonical frame. Substituting the expressions for $\mathbf{C}$ and $\mathbf{p}$ from (5.4) and (5.5) in (5.3), we have

$$\begin{aligned}(\mathbf{C}_e^T\mathbf{p}_i')^T\lfloor\bar{\mathbf{p}}\times\rfloor\mathbf{C}_y(\alpha)\mathbf{C}_x(\beta)\mathbf{C}_e^T\mathbf{p}_i = 0 \\ \Rightarrow \bar{\mathbf{p}}_i'^T\lfloor\bar{\mathbf{p}}\times\rfloor\mathbf{C}_y(\alpha)\bar{\mathbf{p}}_i = 0\end{aligned} \tag{5.6}$$

where

$$\bar{\mathbf{p}}_i = \mathbf{C}_x(\beta)\mathbf{C}_e^T\mathbf{p}_i = [\bar{p}_{ix},\ \bar{p}_{iy}, \bar{p}_{iz}]^T \tag{5.7}$$

$$\bar{\mathbf{p}}_i' = \mathbf{C}_e^T\mathbf{p}_i' = [\bar{p}_{ix}',\ \bar{p}_{iy}', \bar{p}_{iz}']^T \tag{5.8}$$

Expanding the above equations, we have the following bilinear polynomial equations:

$$(a_{i1}c + a_{i2}s + a_{i3})\bar{x} + (a_{i4}c - a_{i5}s)\bar{y} + (a_{i2}c - a_{i1}s - a_{i6})\bar{z} = 0,\ i = 1...N \tag{5.9}$$

where $c := c\alpha$, $s := s\alpha$, and

$$\begin{aligned} a_{i1} &= -\bar{p}_{iz}\bar{p}_{iy}' & a_{i4} &= \bar{p}_{iz}\bar{p}_{ix}' - \bar{p}_{ix}\bar{p}_{iz}' \\ a_{i2} &= \bar{p}_{ix}\bar{p}_{iy}' & a_{i5} &= \bar{p}_{ix}\bar{p}_{ix}' + \bar{p}_{iz}\bar{p}_{iz}' \\ a_{i3} &= \bar{p}_{iy}\bar{p}_{iz}' & a_{i6} &= \bar{p}_{iy}\bar{p}_{ix}'. \end{aligned}$$

Our objective is to determine the rotation angle α and the translation $\bar{\mathbf{p}}$ from the equations in (5.9) and the trigonometric constraint $c^2 + s^2 = 1$, which is a system of multivariate polynomial equations in c, s, $\bar{x}$, $\bar{y}$, and $\bar{z}$.

5.4 Minimal Solution

We first present a closed-form solution to the minimal problem where there exist 3 point feature matches between the two camera views. We solve this system by elimination and back-substitution. We first employ an elimination procedure to obtain a 4^{th} order univariate polynomial in s, which we solve to find 4 solutions for s. Then, by back-substitution, we find 4 solutions for c. Each solution of s and c corresponds to two solutions for the translation with $\pm\bar{\mathbf{p}}$. Therefore, we have a total of 8 solutions for the relative rotation matrix and translation vector.

The main steps of the elimination procedure are as follows:

1. Rewrite the three epipolar (5.3) constraints as linear functions of $\bar{x}$, $\bar{y}$, and $\bar{z}$ in matrix form [see (5.10)].

2. Eliminate $\bar{x}$, $\bar{y}$, and $\bar{z}$ by computing the determinant of the coefficient matrix [see (5.10)], which yields a quadratic polynomial in c and s.

3. Finally, use the Sylvester resultant to eliminate one of the remaining two unknowns, say c, and obtain a 4^{th} order polynomial in s.

Specifically, rewriting the epipolar constraints in a matrix form, we have:

$$\begin{bmatrix} a_{11}c + a_{12}s + a_{13} & a_{14}c - a_{15}s & a_{12}c - a_{11}s - a_{16} \\ a_{21}c + a_{22}s + a_{23} & a_{24}c - a_{25}s & a_{22}c - a_{21}s - a_{26} \\ a_{31}c + a_{32}s + a_{33} & a_{34}c - a_{35}s & a_{32}c - a_{31}s - a_{36} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.10}$$

The determinant of the coefficient matrix is a quadratic polynomial in c and s.

$$\begin{aligned} & g_1c^3 + g_2c^2s + g_1cs^2 + g_2s^3 + g_3c^2 + g_4cs + g_5s^2 + g_6c + g_7s = 0 \\ \Rightarrow\ & g_1c(c^2+s^2) + g_2s(c^2+s^2) + g_3c^2 + g_4cs + g_5s^2 + g_6c + g_7s = 0 \\ \Rightarrow\ & g_3c^2 + g_4cs + g_5s^2 + (g_1+g_6)c + (g_2+g_7)s = 0 \end{aligned} \tag{5.11}$$

where

$$\begin{aligned} g_1 &= -a_{12}a_{24}a_{31} + a_{12}a_{21}a_{34} + a_{22}a_{14}a_{31} - a_{11}a_{22}a_{34} - a_{32}a_{14}a_{21} + a_{32}a_{11}a_{24} \\ g_2 &= a_{21}a_{15}a_{32} - a_{11}a_{25}a_{32} + a_{31}a_{12}a_{25} - a_{12}a_{21}a_{35} - a_{31}a_{15}a_{22} + a_{11}a_{22}a_{35} \\ g_3 &= a_{23}a_{12}a_{34} - a_{31}a_{14}a_{26} + a_{31}a_{16}a_{24} + a_{33}a_{14}a_{22} - a_{33}a_{12}a_{24} - a_{11}a_{24}a_{36} \\ &\quad + a_{11}a_{26}a_{34} + a_{13}a_{24}a_{32} - a_{13}a_{22}a_{34} + a_{21}a_{14}a_{36} - a_{21}a_{16}a_{34} - a_{23}a_{14}a_{32} \\ g_4 &= -a_{33}a_{15}a_{22} + a_{11}a_{25}a_{36} - a_{11}a_{26}a_{35} - a_{12}a_{24}a_{36} + a_{12}a_{26}a_{34} - a_{13}a_{24}a_{31} \\ &\quad - a_{13}a_{25}a_{32} + a_{13}a_{22}a_{35} + a_{13}a_{21}a_{34} - a_{21}a_{15}a_{36} + a_{21}a_{16}a_{35} + a_{22}a_{14}a_{36} \\ &\quad - a_{22}a_{16}a_{34} + a_{23}a_{14}a_{31} + a_{23}a_{15}a_{32} - a_{23}a_{12}a_{35} - a_{23}a_{11}a_{34} + a_{31}a_{15}a_{26} \\ &\quad - a_{31}a_{16}a_{25} - a_{32}a_{14}a_{26} + a_{32}a_{16}a_{24} - a_{33}a_{14}a_{21} + a_{33}a_{12}a_{25} + a_{33}a_{11}a_{24} \\ g_5 &= -a_{23}a_{15}a_{31} + a_{23}a_{11}a_{35} + a_{32}a_{15}a_{26} - a_{32}a_{16}a_{25} + a_{33}a_{15}a_{21} - a_{33}a_{11}a_{25} \\ &\quad + a_{12}a_{25}a_{36} - a_{12}a_{26}a_{35} + a_{13}a_{25}a_{31} - a_{13}a_{21}a_{35} - a_{22}a_{15}a_{36} + a_{22}a_{16}a_{35} \\ g_6 &= -a_{23}a_{16}a_{34} - a_{33}a_{14}a_{26} + a_{33}a_{16}a_{24} - a_{13}a_{24}a_{36} + a_{13}a_{26}a_{34} + a_{23}a_{14}a_{36} \\ g_7 &= -a_{23}a_{15}a_{36} + a_{23}a_{16}a_{35} + a_{33}a_{15}a_{26} - a_{33}a_{16}a_{25} + a_{13}a_{25}a_{36} - a_{13}a_{26}a_{35} \end{aligned} \tag{5.12}$$

In the final step, we employ the Sylvester resultant to eliminate one of the remaining two variables from equations (5.11) and the trigonometric constraint $c^2 + s^2 = 1$. The Sylvester resultant is the determinant of the coefficient matrix of the following system:

$$\begin{bmatrix} g_3 & g_4s + g_1 + g_6 & g_5s^2 + (g_2+g_7)s & 0 \\ 0 & g_3 & g_4s + g_1 + g_6 & g_5s^2 + (g_2+g_7)s \\ 1 & 0 & s^2 - 1 & 0 \\ 0 & 1 & 0 & s^2 - 1 \end{bmatrix} \begin{bmatrix} c^3 \\ c^2 \\ c \\ 1 \end{bmatrix} = 0 \tag{5.13}$$

which is a univariate polynomial in s

$$\sum_{i=0}^{4} h_i s^i = 0 \tag{5.14}$$

where

$$\begin{aligned}
h_0 &= -g_1^2 - 2g_1g_6 - g_6^2 + g_3^2 \\
h_1 &= 2g_3g_2 - 2g_4g_6 + 2g_3g_7 - 2g_4g_1 \\
h_2 &= -g_4^2 + g_1^2 + g_6^2 + g_2^2 + g_7^2 - 2g_3^2 + 2g_1g_6 + 2g_2g_7 + 2g_3g_5 \\
h_3 &= 2g_4g_1 + 2g_4g_6 + 2g_5g_2 + 2g_5g_7 - 2g_3g_2 - 2g_3g_7 \\
h_4 &= g_4^2 + g_5^2 + g_3^2 - 2g_3g_5
\end{aligned}$$

We can find the 4 solutions for s by solving in closed-form [23, Ch. 4.16] the 4^{th} order univariate polynomial equation (5.14). Back-substituting the solutions of s into (5.11), we compute the corresponding solutions for c. Note that each solution of s corresponds to one solution for c, because we can reduce the order of (5.11) to linear in c, once s is known, by replacing the quadratic term c^2 with $1 - s^2$. The solution for c is:

$$c = \frac{(g_3 - g_5)s^2 - (g_2 + g_7)s - g_3}{g_4 s + g_1 + g_6}. \tag{5.15}$$

After c and s are found, we determine the corresponding solutions for $\bar{x}$, $\bar{y}$, and $\bar{z}$ by computing the null space of the coefficient matrix in (5.10). Finally, we recover $\mathbf{p}$ from (5.5).

5.5 Least-squares Solutions

We now consider the case where 2-DOF of the rotation have been computed as described in Section 5.3 and we are interested in computing the least-squares (LS) solution for the remaining 1-DOF of rotation and the 2-DOF for translation. We start by assuming that all outliers have been removed using RANSAC and we have $N > 3$ point correspondences

$\mathbf{p}_i$ and $\mathbf{p}'_i$, $i = 1, ..., N$. The LS cost function that we seek to minimize in order to compute $\mathbf{C}_y(\alpha)$ and $\bar{\mathbf{p}}$ is:

$$\min_{\alpha, \bar{\mathbf{p}}} J = \frac{1}{N} \sum (\bar{\mathbf{p}}_i'^T \lfloor \bar{\mathbf{p}} \times \rfloor \mathbf{C}_y(\alpha) \bar{\mathbf{p}}_i)^2. \tag{5.16}$$

where $\bar{\mathbf{p}}_i$, $\bar{\mathbf{p}}'_i$ and $\bar{\mathbf{p}}$ are defined by equations (5.7), (5.8) and (5.5), respectively. In this section, we will provide two analytical solutions to this least-squares problem. The first one seeks to minimize the cost function J directly with respect to α and $\bar{\mathbf{p}}$. The second one transforms the original optimization problem into an eigenvalue minimization problem and approximates it by minimizing the corresponding matrix determinant. Interestingly, we will show, in simulation, that this approximate solution outperforms the exact solution due to its better numerical stability.

5.5.1 Exact Solution

In order to find the global minimum of the cost function J, we first determine *all critical/stationary points* by solving a system of polynomial equations, and evaluate their objective function values. Then the optimal solution is the critical point whose objective value is the smallest.

We parameterize[1] the translation using $\bar{\mathbf{p}} = [x, y, 1]^T$ in order to simplify the derivation of the gradient of J.

$$\nabla_x J = \frac{2}{N} \sum (\bar{\mathbf{p}}_i'^T \lfloor \bar{\mathbf{p}} \times \rfloor \mathbf{C}_y(\alpha) \bar{\mathbf{p}}_i)(a_{i1}c + a_{i2}s + a_{i3}) = 0 \tag{5.17}$$

$$\nabla_y J = \frac{2}{N} \sum (\bar{\mathbf{p}}_i'^T \lfloor \bar{\mathbf{p}} \times \rfloor \mathbf{C}_y(\alpha) \bar{\mathbf{p}}_i)(a_{i4}c - a_{i5}s) = 0 \tag{5.18}$$

$$\nabla_\alpha = \frac{2}{N} \sum (\bar{\mathbf{p}}_i'^T \lfloor \bar{\mathbf{p}} \times \rfloor \mathbf{C}_y(\alpha) \bar{\mathbf{p}}_i)(-a_{i1}xs + a_{i2}xc - a_{i4}ys - a_{i5}yc - a_{i2}s - a_{i1}c) = 0 \tag{5.19}$$

Together with the trigonometric constraint $c^2 + s^2 = 1$, we have a system of 4 equations in 4 unknowns $[c,\ s,\ x,\ y]$. The stationary points can be found by solving this system of

[1] This is done by dividing both sides of (5.6) by $\bar{z}$ and setting $x = \bar{x}/\bar{z}$ and $y = \bar{y}/\bar{z}$ before substituting in (5.16).

polynomial equations. To do so, we employ an elimination and back-substitution procedure similar to the minimal solution in Section 5.4. The main steps of this elimination procedure are as follows:

1. Solve for x and y as functions of c and s using equations (5.17) and (5.18) which are linear in x and y

$$\begin{bmatrix} \sum(a_{i1}c + a_{i2}s + a_{i3})\bar{\mathbf{p}}_i^T \mathbf{C}_y(\alpha)^T \lfloor \bar{\mathbf{p}}_i' \times \rfloor^T \\ \sum(a_{i4}c - a_{i5}s)\bar{\mathbf{p}}_i^T \mathbf{C}_y(\alpha)^T \lfloor \bar{\mathbf{p}}_i' \times \rfloor^T \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{0}$$

$$\Rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{l_3} \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \tag{5.20}$$

   where l_1, l_2, and l_3 are 4^{th} order polynomials in c and s.

2. Substitute the above expressions for x and y in equation (5.19) and multiply with l_3 to remove the denominator, which yields a 10^{th} order polynomial f in c and s.

3. Finally, use the Sylvester resultant of f and of the trigonometric constraint, $s^2 + c^2 = 1$, to eliminate one of the remaining two unknowns, say c, and obtain a 14^{th} order polynomial in s.

After we compute the 14 solutions for s using the companion matrix, we back-substitute them in f to solve for c. Since this 10^{th} order polynomial can be reduced to linear in c using the trigonometric constraint $c^{2n} = (1 - s^2)^n$, $n = 1, ..., 5$, every solution of s corresponds to one solution of c. Finally, x and y can be uniquely determined for each pair of c and s using (5.20).

5.5.2 Approximate Solution

The exact solution requires solving a 14^{th} order polynomial and multiple back-substitution steps, which is numerically unstable. In this section, we will present an approximate solution that mitigates the problem of numerical instability.

We first reformulate the least-squares problem as a matrix eigenvalue minimization problem. From (5.16), we have

$$\begin{aligned} J &= \frac{1}{N}\sum(\bar{\mathbf{p}}^T \lfloor \bar{\mathbf{p}}'_i \times \rfloor \mathbf{C}_y(\alpha)\bar{\mathbf{p}}_i)^2 \\ &= \bar{\mathbf{p}}^T \left[\frac{1}{N}\sum \lfloor \bar{\mathbf{p}}'_i \times \rfloor \mathbf{C}_y(\alpha)\bar{\mathbf{p}}_i \bar{\mathbf{p}}_i^T \mathbf{C}_y(\alpha)^T \lfloor \bar{\mathbf{p}}'_i \times \rfloor^T\right] \bar{\mathbf{p}} \\ &= \bar{\mathbf{p}}^T \mathbf{A} \bar{\mathbf{p}} \end{aligned} \tag{5.21}$$

For $||\bar{\mathbf{p}}|| = 1$, the minimal value of J is the smallest eigenvalue of $\mathbf{A}$, i.e., $\min J = \min eig(\mathbf{A})$. However, it is very difficult to find the value of α that minimizes the smallest eigenvalue of $\mathbf{A}$ analytically. On the other hand, $\mathbf{A}$ is positive semi-definite and in the ideal (noise free) case, $\mathbf{A}$ should be rank deficient, i.e., $\det(\mathbf{A}) = 0$. In practice, however, $\det(\mathbf{A}) > 0$ due to noise in the measurements. Based on this fact, we seek to find an approximate solution for α by minimizing the determinant of $\mathbf{A}$ instead of its smallest eigenvalue. In the following, we compute all the critical/stationary points of $\det(\mathbf{A})$ and choose as solution for α the one corresponding to the smallest determinant.

The determinant of $\mathbf{A}$ is a 6^{th} order polynomial in c and s:

$$\begin{aligned} \det(\mathbf{A}) = {} & k_1c^6 + k_2c^5s + k_3c^4s^2 + k_4c^3s^3 + k_5c^2s^4 + k_6cs^5 + k_7s^6 + k_8c^5 + k_9c^4s \\ & + k_{10}c^3s^2 + k_{11}c^2s^3 + k_{12}cs^4 + k_{13}s^5 + k_{14}c^4 + k_{15}c^3s + k_{16}c^2s^2 + k_{17}cs^3 \\ & + k_{18}s^4 + k_{19}c^3 + k_{20}c^2s + k_{21}cs^2 + k_{22}s^3 + k_{23}c^2 + k_{24}cs + k_{25}s^2 \end{aligned} \tag{5.22}$$

where the $k'_i s$ are functions of the measurements $\mathbf{p}_i$ and $\mathbf{p}'_i$. Taking derivative with respect to α, we obtain:

$$\begin{aligned} \nabla_\alpha(\det(\mathbf{A})) = {} & k_2c^6 + [(2k_3 - 6k_1)s + k_9]c^5 + [(3k_4 - 5k_2)s^2 + (2k_{10} - 5k_8)s + k_{15}]c^4 \\ & + [(4k_5 - 4k_3)s^3 + (3k_{11} - 4k_9)s^2 + (2k_{16} - 4k_{14})s + k_{20}]c^3 \\ & + [(5k_6 - 3k_4)s^4 + (4k_{12} - 3k_{10})s^3 + (3k_{17} - 3k_{15})s^2 + (2k_{21} - 3k_{19})s + k_{24}]c^2 \\ & + [(6k_7 - 2k_5)s^5 + (5k_{13} - 2k_{11})s^4 + (4k_{18} - 2k_{16})s^3 + (3k_{22} - 2k_{20})s^2 \\ & + (2k_{25} - 2k_{23})s]c \\ & - k_6s^6 - k_12s^5 - k_{17}s^4 - k_{21}s^3 - k_{24}s^2 = 0 \end{aligned} \tag{5.23}$$

Next, we employ the Sylvester resultant to eliminate c from equation (5.23) and the trigonometric constraint $s^2 + c^2 = 1$, which yields an 8^{th} order polynomial in s.

$$l_1 s^8 + l_2 s^7 + l_3 s^6 + l_4 s^5 + l_5 s^4 + l_6 s^3 + l_7 s^2 + l_8 s + l_9 = 0 \tag{5.24}$$

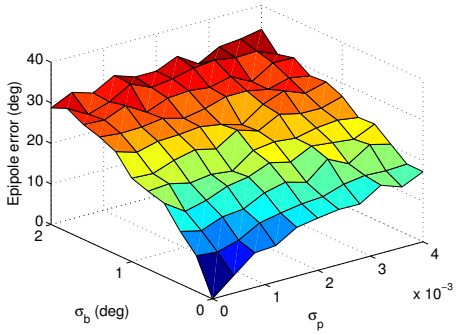
After computing the 8 solutions for s, we determine the solutions for c using (5.23). The translation vector $\bar{\mathbf{p}}$ is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}$.

5.6 Simulation Results

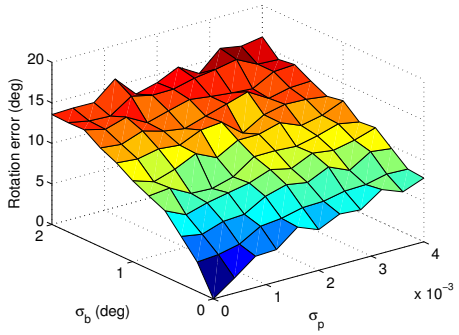
We evaluate our closed-form solution for the three-plus-one problem and compare the performance of the exact and approximate least-squares solutions in simulation. We have conducted Monte Carlo simulations for different measurement noise variance settings and camera motion configurations, and report the average results of 1000 runs. In each Monte Carlo experiment, we randomly generate 100 3D points and project them onto the two normalized image planes. The transformations between the two camera frames are randomly generated with a unit 3D vector $\mathbf{p}$ and a rotational matrix $\mathbf{C}$. The two camera configurations tested are the forward and sideways motions. In the forward motion, the translation along the z-axis is about 10 times larger than the other two axes, while in the sideways motion, the translation along the x-axis is about 10 times larger than the other two axes. The image points are corrupted by Gaussian noise on the normalized image plane with standard deviation σ_p from 0 to 4×10^{-3}, while the directional measurements are perturbed with a rotation around a random axis with a small angle drawn from a Gaussian distribution with standard deviation σ_b from 0 to 2 degrees.

Fig. 5.2 shows the performance of the minimal solver with both directional and image noise. In particular, Fig. 5.2(a) shows the absolute error in translation, represented by the angle between the estimated and true translation vector $\mathbf{p}$, and Fig. 5.2(b) shows

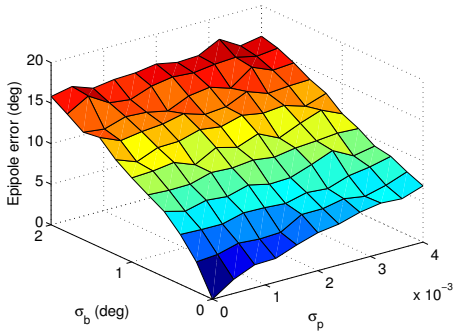
the absolute error in rotation, represented by the rotational angle between the estimated and true rotation matrix $\mathbf{C}$, when the camera is moving forward. Fig. 5.2(c) and 5.2(d) show the same kind of errors when the camera is moving sideways. As evident moving sideways provides higher estimation accuracy as compared to a forward motion.



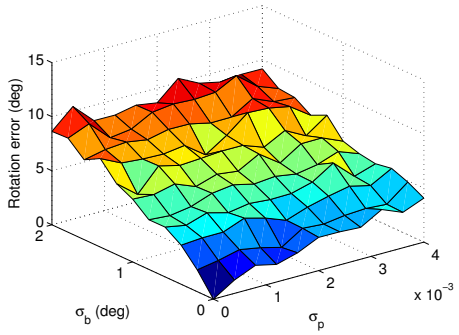
(a) Forward Motion Translation Error



(b) Forward Motion Rotation Error



(c) Sideways Motion Translation Error

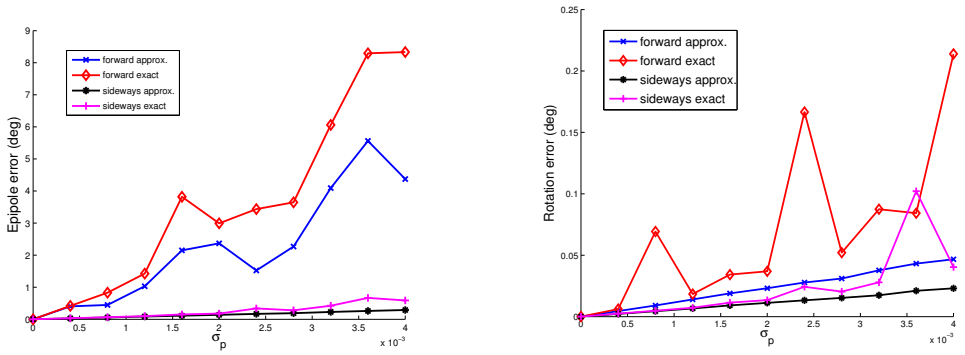


(d) Sideways Motion Rotation Error

Figure 5.2: Simulation results of the 3 plus 1 minimal solution. The plots show the mean absolute error of translation and rotation as functions of the directional and camera measurement noise (in normalized image coordinates) averaged over 1000 Monte Carlo trials.

Fig. 5.3 shows the results of the least-squares solutions with only camera measurement noise. Note that the approximate solution is much more accurate than the exact solution in both forward and sideways motions. In particular, the absolute rotation errors of the approximate solution increase almost linearly with the image noise variance,

while the exact solution shows more variations in its accuracy, because it requires solving a 14^{th} order polynomial. In contrast, the approximate solution is numerically more stable than the exact solution because it only requires solving an 8^{th} order polynomial.



(a) Least-squares Translation Error

(b) Least-squares Rotation Error

Figure 5.3: Simulation results for the closed-form least-squares solution. The plots show the mean absolute errors of translation and rotation as functions of camera measurement noise (in normalized image coordinates) averaged over 1000 Monte Carlo trials.

5.7 Summary

In this chapter, we addressed the problem of determining the relative position and orientation of a camera from image feature point matches and a directional correspondence. In particular, we provided an efficient closed-form solution to the minimal problem with three point correspondences and one directional constraint. Additionally, we presented two analytical solutions to the least-squares problem using N feature points and one directional correspondence. The first solution seeks all the critical points of the least-squares cost function, which involves solving a 14^{th} order univariate polynomial equation. The optimal solution is the one with the minimal cost function value. Although this solution is mathematically exact, it suffers from numerical instability due to the requirement of solving high order polynomials and error propagation in subsequent

back-substitution steps. The second solution employs an approximation to the original cost function. Specifically, the optimal solution is the smallest eigenvalue of a positive semidefinite matrix which is a function of the camera orientation. Instead of solving for the smallest eigenvalue, we minimize the determinant of this matrix. As a result, we only need to solve an 8^{th} order univariate polynomial. The resulting approximate solution showed significantly better numerical stability and provided more accurate estimates of the relative camera pose.

Chapter 6

Optimal Motion Strategies

In this chapter, we seek optimal robot motion strategies that minimize the uncertainty of the relative robot pose estimates. In particular, we study the case of a team of robots moving in formation and analyze the trade-off between maintaining the formation and improving the accuracy of the relative pose estimates. Part of this chapter has been published in a conference paper [95]. This is work in collaboration with Ke Zhou.

6.1 Introduction

Robots are often required to move in a formation. For example, in military missions, aerial vehicles fly in formations for mutual defense and reconnaissance; in transportation, vehicle platooning saves fuel and increases the throughput of transportation networks [8]; and in cooperative object manipulation tasks, multiple robots form a formation in order to move a large payload [74].

Due to its broad applications, formation control has been extensively studied, and there exist many approaches for maintaining formation, such as behavior-based [7], virtual structure [80], and leader-follower-based [17] methods. All these methods rely on on-board sensors to provide information about the robots' relative positions and orientations (poses), so as to reduce the deviations from the desired poses. However,

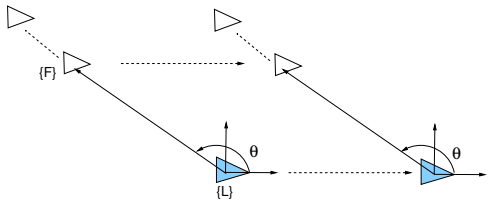


Figure 6.1: A leader, L, and a follower, F, are moving on parallel straight lines. The relative pose between the leader and follower are unobservable when the robots only measure the bearing θ, because the follower can be anywhere along the line of sight of the leader.

sensors usually do not measure the relative pose directly, but only measure the range or bearing from one robot to its neighbors. Therefore, an estimator is often employed for estimating the relative pose using inter-robot measurements. Clearly, successful formation control requires accurate relative pose estimates.

Interestingly, robot motion affects the accuracy of relative pose estimates. Early experimental results of Rekleitis *et al.* [65] show that random robot motion improves the accuracy of the relative pose estimates as compared to moving in a formation. In addition, Trawny and Barfoot [82] also find that the optimal motion strategy is not to move in a formation.

The estimation accuracy in robot formations degrades because the system is unobservable. Specifically, when the robots move in formation their range or bearing measurements are constant over time and thus do not provide sufficient information to determine the robots' relative poses (see Fig. 6.1). However, the system becomes observable when the robots move arbitrarily [50, 93, 48], because knowledge of each robot's time-varying motions along with the range or bearing measurements provides additional information about their relative poses. As a result, random trajectories show better performance than moving in formation. Hence, there is a trade-off between maintaining a rigid formation and improving the estimation accuracy.

To analyze this trade-off, we seek the optimal next-time step follower position in a leader-follower formation. In order to balance the requirement of moving in formation

and improving the estimation accuracy, the follower is allowed to move away from the desired position within a given range. The specific estimator employed in this work is the Extended Kalman Filter (EKF), and the optimization criterion used is the weighted trace of the posterior covariance matrix. Our main contribution is that we find the global optimal of this optimization problem analytically by first computing all critical points (i.e., the points which satisfy the Karush-Kuhn-Tucker (KKT) necessary optimality conditions) and select as global minimum the one that yields the lowest cost.

The remainder of the chapter is organized as follows. In the following section, we describe related localization approaches in robot formation. In Section 6.2, we formulate the optimal motion strategy problem and present our solution in Section 6.3 followed by simulation results (Section 6.4). Finally, in Section 6.5, we draw our conclusions and suggest future research directions.

6.2 Problem Formulation

In a leader-follower formation, the leader moves on a desired straight line trajectory, while the follower keeps its position $\mathbf{p} = [x,\ y]^T$ and orientation ϕ constant with respect to the leader[1] . We define the state vector as the follower's position and orientation with respect to the leader, $\mathbf{x} = [x,\ y,\ \phi]^T$, and denote the desired follower state as $\mathbf{x}_o = [x_o,\ y_o,\ \phi_o]^T$. Furthermore, the follower measures the distance d, or bearing β to the leader (see Fig. 6.2). The follower estimates its state using these measurements, so as to maintain the formation. Due to the trade-off between maintaining the formation and improving the estimation accuracy discussed before, we allow the follower's position to deviate from the desired position $\mathbf{p}_o$ and move within a circle of radius r centered at $\mathbf{p}_o$. Our objective is to determine where the follower should move to at the next time step to record a relative measurement, such that the estimation uncertainty is minimized.

[1] In this chapter, we only consider the case with one follower. The same algorithm can be applied to the case of multiple followers.

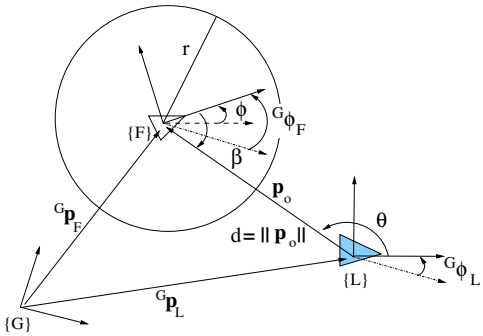


Figure 6.2: A leader-follower formation: the follower F is at the desired position $\mathbf{p}_o$ with respect to the leader L. The follower measures either the distance d, or bearing β to the leader. In order to gain observability and minimize the relative pose estimation error, the follower is allowed to move inside the circle centered at $\mathbf{p}_o$ with radius r.

In the following sections, we will first present the system kinematic model[2] , the measurement model, and then formulate the optimization problem that needs to be solved.

6.2.1 System Kinematic Model

We assume that both the leader and the follower are non-holonomic vehicles whose linear and rotational velocities are $(v_L,\ \omega_L)$ and $(v_F,\ \omega_F)$, respectively. In addition, we focus on the case where the leader is moving on a straight line with constant v_L, and $\omega_L = 0$, because, most likely, it is the dominant motion when a team of robots are moving towards a distant destination. The directions of the linear velocities are along the x-axes of the vehicles' body frames. Hence, these linear velocities expressed in a fixed global frame $\{G\}$ are described by

$$^G\dot{\mathbf{p}}_L = {}^G_L\mathbf{C} \cdot \mathbf{e}_1 v_L, \quad ^G\dot{\mathbf{p}}_F = {}^G_F\mathbf{C} \cdot \mathbf{e}_1 v_F. \tag{6.1}$$

where $\mathbf{e}_1 = [1\ 0]^T$, ${}^G_L\mathbf{C}$ and ${}^G_F\mathbf{C}$ are the rotation matrices that project vectors expressed in frame $\{L\}$ and $\{F\}$ to frame $\{G\}$, respectively.

[2] Note that our optimization framework can be applied with any system model. The specific one presented hereafter is used in our simulations to evaluate the performance of our method.

The rotation matrices ${}^G_L\mathbf{C}$, ${}^G_F\mathbf{C}$, and ${}^L_F\mathbf{C}$ satisfy the following relationship:

$$ {}^G_F\mathbf{C}({}^G\phi_F) = {}^G_L\mathbf{C}({}^G\phi_L) \cdot {}^L_F\mathbf{C}(\phi) \tag{6.2} $$

$$ \Rightarrow {}^G\phi_F = {}^G\phi_L + {}^L\phi_F. \tag{6.3} $$

Differentiating (6.3) with respect to time, yields the kinematic model for the orientation:

$$ \dot{\phi} = {}^L\dot{\phi}_F = {}^G\dot{\phi}_F - {}^G\dot{\phi}_L = \omega_F - \omega_L = \omega_F. \tag{6.4} $$

Similarly, we derive the kinematic model for the robots' relative position $\mathbf{p}$ by taking the time derivative of the following geometric constraint

$$ \mathbf{p} = {}^G_L\mathbf{C}^T({}^G\mathbf{p}_F - {}^G\mathbf{p}_L) \tag{6.5} $$

$$ \Rightarrow \dot{\mathbf{p}} = {}^G_L\mathbf{C}^T({}^G\dot{\mathbf{p}}_F - {}^G\dot{\mathbf{p}}_L). \tag{6.6} $$

Substituting (6.1) and (6.2) in (6.6), we obtain

$$ \dot{\mathbf{p}} = {}^L_F\mathbf{C} \cdot \mathbf{e}_1 v_F - \mathbf{e}_1 v_L \tag{6.7} $$

Rearranging terms in (6.4) and (6.7), yields the following continuous-time model:

$$ \begin{aligned} \dot{x} &= \cos(\phi) v_F - v_L \\ \dot{y} &= \sin(\phi) v_F \\ \dot{\phi} &= \omega_F. \end{aligned} \tag{6.8} $$

6.2.2 State and Covariance Propagation

The discrete-time kinematic equations of the follower's pose are obtained by discretizing (6.8) using Euler's forward method.

$$ \begin{aligned} x_{k+1} &= x_k + \cos(\phi_k) v_{F_k} \delta t - v_L \delta t \\ y_{k+1} &= y_k + \sin(\phi_k) v_{F_k} \delta t \\ \phi_{k+1} &= \phi_k + \omega_{F_k} \delta t \end{aligned} \tag{6.9} $$

We employ the EKF to propagate the estimated follower's pose[3] using the control inputs (odometry measurements) $v_{F_{mk}}$ and $\omega_{F_{mk}}$

$$\begin{aligned}
\hat{x}_{k+1|k} &= \hat{x}_{k|k} + \cos(\hat{\phi}_{k|k}) v_{F_{mk}} \delta t - v_L \delta t \\
\hat{y}_{k+1|k} &= \hat{y}_{k|k} + \sin(\hat{\phi}_{k|k}) v_{F_{mk}} \delta t \\
\hat{\phi}_{k+1|k} &= \hat{\phi}_{k|k} + \omega_{F_{mk}} \delta t
\end{aligned} \tag{6.10}$$

By linearizing (6.10), the error-propagation equation for the follower's pose is readily derived as:

$$\begin{aligned}
\begin{bmatrix} \tilde{x}_{k+1|k} \\ \tilde{y}_{k+1|k} \\ \tilde{\phi}_{k+1|k} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & -\sin(\hat{\phi}_{k|k}) v_{F_{mk}} \delta t \\ 0 & 1 & \cos(\hat{\phi}_{k|k}) v_{F_{mk}} \delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}_{k|k} \\ \tilde{y}_{k|k} \\ \tilde{\phi}_{k|k} \end{bmatrix} \\
&\quad + \begin{bmatrix} \cos(\hat{\phi}_{k|k}) \delta t & 0 \\ \sin(\hat{\phi}_{k|k}) \delta t & 0 \\ 0 & \delta t \end{bmatrix} \begin{bmatrix} w_{v_{F_k}} \\ w_{\omega_{F_k}} \end{bmatrix} \\
\Leftrightarrow \quad \tilde{\mathbf{x}}_{k+1|k} &= \mathbf{\Phi}_k \tilde{\mathbf{x}}_{k|k} + \mathbf{G}_k \mathbf{w}_k
\end{aligned} \tag{6.11}$$

where $w_{v_{F_k}}$ and $w_{\omega_{F_k}}$ are white, zero-mean, Gaussian noise sequences with variances $\sigma^2_{v_{F_k}}$ and $\sigma^2_{\omega_{F_k}}$, affecting the linear and rotational velocity inputs, respectively.

From equation (6.11), the error-state covariance propagation equation can be written as:

$$\mathbf{P}_{k+1|k} = \mathbf{\Phi}_k \mathbf{P}_{k|k} \mathbf{\Phi}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \tag{6.12}$$

where $\mathbf{Q}_k = \text{diag}(\sigma^2_{v_{F_k}}, \sigma^2_{\omega_{F_k}})$.

[3] In the rest of the paper, the "hat" symbol "ˆ" denotes the estimated value of a quantity. The subscript $l|j$ refers to the estimate of the quantity at time step l after incorporating all measurements up to time-step j. The error between the actual value of the quantity and its estimate is denoted by the "tilde" symbol "˜" . The relationship between the actual value x and the estimate $\hat{x}$ is $\tilde{x} = x - \hat{x}$.

6.2.3 Measurement Update

The follower employs the inter-robot measurements to perform pose updates in the EKF. The inter-robot measurement at time step k can be described by the generally nonlinear model

$$z_{k+1} = h(\mathbf{x}_{k+1}, n_{k+1}) \tag{6.13}$$

where n_k is the Gaussian noise with covariance $\mathbf{R}$. In the EKF, we employ linearization to obtain the measurement error equation

$$\tilde{z}_{k+1} = \mathbf{H}_{k+1}\tilde{\mathbf{x}}_{k+1|k} + \Gamma_{k+1}n_{k+1} \tag{6.14}$$

where

$$\mathbf{H}_{k+1} = \nabla_{\mathbf{x}_{k+1}} h(\mathbf{x}_{k+1}, n_{k+1})|_{\hat{\mathbf{x}}_{k+1|k},0}$$
$$\Gamma_{k+1} = \nabla_{n_{k+1}} h(\mathbf{x}_{k+1}, n_{k+1})|_{\hat{\mathbf{x}}_{k+1|k},0}.$$

The Jacobian matrices $\mathbf{H}_{k+1}$ and Γ_{k+1} are time varying and depend on the state estimates. Thus, the location where the robot records the measurement will affect the accuracy of the state estimates. In this chapter, we consider two types of inter-robot measurements, namely the inter-robot range and bearing.

Range measurements

The range measurement between the leader and the follower is described by the following model (see Fig 6.2):

$$z_{k+1} = d_{k+1} + n_{d_{k+1}} = \sqrt{x_{k+1}^2 + y_{k+1}^2} + n_{d_{k+1}} \tag{6.15}$$

where the noise $n_{d_{k+1}}$ is zero-mean Gaussian process with variance $\mathbf{R}_d = \sigma_d^2$.

The measurement Jacobian for the distance measurement is a 1×3 vector:

$$\begin{aligned}\mathbf{h}_{d_{k+1}} &= \begin{bmatrix} \frac{\hat{x}_{k+1|k}}{\hat{d}_{k+1|k}} & \frac{\hat{y}_{k+1|k}}{\hat{d}_{k+1|k}} & 0 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\hat{\theta}_{k+1|k}) & \sin(\hat{\theta}_{k+1|k}) & 0 \end{bmatrix}. \end{aligned} \tag{6.16}$$

where $\hat{d}_{k+1|k} = \sqrt{\hat{x}^2_{k+1|k} + \hat{y}^2_{k+1|k}}$.

Bearing measurements

The relative bearing from the follower to the leader is described by (see Fig 6.2):

$$\begin{aligned} z_{k+1} &= \beta_{k+1} + n_{\beta_{k+1}} = \theta_{k+1} - \phi_{k+1} + \pi + n_{\beta_{k+1}} \\ &= \arctan\left(\frac{y_{k+1}}{x_{k+1}}\right) - \phi_{k+1} + \pi + n_{\beta_{k+1}} \end{aligned} \tag{6.17}$$

where $n_{\beta_{k+1}}$ is zero-mean Gaussian noise with variance $\mathbf{R}_\beta = \sigma^2_\beta$.

The measurement Jacobian for the bearing measurement has the following structure:

$$\begin{aligned} \mathbf{h}_{\beta_{k+1}} &= \begin{bmatrix} -\frac{\hat{y}_{k+1|k}}{\hat{d}^2_{k+1|k}} & \frac{\hat{x}_{k+1}}{\hat{d}^2_{k+1}} & -1 \end{bmatrix} \\ &= \frac{1}{\hat{d}_{k+1|k}} \begin{bmatrix} -\sin(\hat{\theta}_{k+1|k}) & \cos(\hat{\theta}_{k+1|k}) & -\hat{d}_{k+1|k} \end{bmatrix}. \end{aligned} \tag{6.18}$$

Posterior Covariance

The inter-robot measurements are processed by the EKF to update the follower's pose estimate. The covariance update equation is

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|k}\mathbf{H}^T_{k+1}\mathbf{S}^{-1}_{k+1}\mathbf{H}_{k+1}\mathbf{P}_{k+1|k} \tag{6.19}$$

where $\mathbf{S}_{k+1} = \mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\mathbf{H}^T_{k+1} + \mathbf{R}_{k+1}$. In this equation, $\mathbf{H}_{k+1}$ and $\mathbf{R}_{k+1}$ are the measurement Jacobian matrix and noise covariance matrix at time-step $k+1$, respectively. The posterior covariance indicates the accuracy of the estimate and it is a function of the propagated state. However, the actual value of the next step measurement is not required for computing the posterior covariance, therefore it is possible to optimize the robot motion to achieve higher accuracy. In the next section, we will formulate the robot motion strategy as an optimization problem. The optimization variable is the next time-step position $\hat{\mathbf{p}}_{k+1|k}$.

6.2.4 Optimization Problem Formulation

The cost function is defined as the weighted sum of the diagonal elements of the posterior covariance,

$$c = \text{tr}(\mathbf{W}\mathbf{P}_{k+1|k+1}\mathbf{W}) \tag{6.20}$$

where the weighting matrix $\mathbf{W} = \text{diag}(1, 1, d_o)$ with $d_o = ||\mathbf{p}_o||_2$. Since $\mathbf{\Phi}_k$ [see (6.11)] is a function of the control input $v_{F_{mk}}$, the prior covariance $\mathbf{P}_{k+1|k}$ [see (6.12)] is also a function of the optimization variable $\hat{\mathbf{p}}_{k+1|k}$, which makes solving the optimization problem very challenging. However, since the robots are required to move in formation, the follower can only deviate from the nominal position $\mathbf{p}_o$ by a small distance r. So we approximate the prior covariance by a constant matrix $\mathbf{P}$ which is chosen to be the prior covariance when the follower moves to the desired position $\mathbf{p}_o$. From here on, we will drop the subscripts for the state, covariance, and measurement Jacobian, when the meaning of each quantity is clear in the context to simplify our notation, e.g., $\mathbf{p} = \hat{\mathbf{p}}_{k+1|k}$ and $\mathbf{P} = \mathbf{P}_{k+1|k}$.

The objective function can be simplified by

$$\begin{aligned} &\underset{\mathbf{p}}{\text{argmin}} \ \text{tr}(\mathbf{W}\mathbf{P}_{k+1|k+1}\mathbf{W}) \\ =&\underset{\mathbf{p}}{\text{argmin}} \ -\text{tr}(\mathbf{W}\mathbf{P}\mathbf{H}^T\mathbf{S}^{-1}\mathbf{H}\mathbf{P}\mathbf{W}) \end{aligned} \tag{6.21}$$

Defining $f = -\text{tr}(\mathbf{W}\mathbf{P}\mathbf{H}^T\mathbf{S}^{-1}\mathbf{H}\mathbf{P}\mathbf{W})$, we formulate the formation optimization problem as

- Optimization Problem 1 (Π_1)

$$\begin{aligned} \underset{\mathbf{p}}{\text{min.}} \ & f \\ \text{s.t.} \ & g = \|\mathbf{p} - \mathbf{p}_o\|_2 - r \leq 0 \end{aligned} \tag{6.22}$$

Remark 20. *The optimization problem* Π_1 *is a nonlinear programming problem since*

both the objective function f and the constraint g are nonlinear functions of the optimization variable. Furthermore, Π_1 is not a convex problem, because the objective function is not convex.

6.3 Problem Solution

As mentioned in the previous section, the optimization problem Π_1 is *not convex*, and thus it has multiple local minima. A naive approach is to discretize the feasible set and find the point which yields the minimum value for the function f. However, such an approach provides no guarantees for finding the global optimum, due to limitations of the grid resolution.

Instead, we find the global optimal solution as follows: We first determine *all critical/stationary points* (i.e., those points which satisfy the Karush-Kuhn-Tucker (KKT) necessary optimality conditions [9, Ch. 3]) analytically and evaluate their objective values. Then, as optimal solution, we select the critical point whose objective value is the smallest.

6.3.1 Distance-only

For the distance-only case, the measurement Jacobian $\mathbf{h}_d$ [see (6.16)] only depends on the bearing angle θ from the leader to the follower, hence the objective function f also only depends on θ. Therefore, we change the optimization variable from the position $\mathbf{p}$ to the bearing angle θ, and reformulate problem Π_1 to the optimization problem Π_2.

- Optimization Problem 2 (Π_2)

$$\begin{aligned} \max_{\theta} \quad & f_d(\theta) \\ \text{s.t.} \quad & \theta_{min} \leq \theta \leq \theta_{max} \end{aligned} \tag{6.23}$$

where θ_{min} is the minimum and θ_{max} is the maximum bearing angle that satisfies the constraint g. From the optimal solution θ^* of problem Π_2, we choose the optimal next

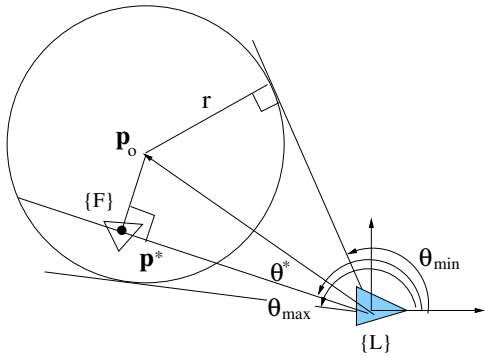


Figure 6.3: Optimal solution for the distance-only case. Since the measurement Jacobian $\mathbf{h}_d$ is a function of only θ, the cost function values are the same for all points on the line with angle θ. From the optimal $\theta^* \in [\theta_{min}, \theta_{max}]$, we determine the optimal next step position $\mathbf{p}^*$ as the nearest point to $\mathbf{p}_o$.

step position $\mathbf{p}^*$ as the nearest point on the line to $\mathbf{p}_o$ (see Fig. 6.3).

The global optimal θ^* can appear in the following two cases: (i) the constraint is inactive, then θ^* satisfies $\nabla_\theta f_d(\theta) = 0$; (ii) the constraint is active, then $\theta^* = \theta_{min}$, or $\theta^* = \theta_{max}$. We examine both cases, and determine the global optimal θ^* by searching for the one with the smallest objective value. In the following, we present our solution for $\nabla_\theta f_d(\theta) = 0$.

The objective function $f_d(\theta)$ is a fractional function whose numerator and denominator are both linear in $\sin(2\theta)$ and $\cos(2\theta)$.

$$\begin{aligned} f_d(\theta) &= -\frac{u}{v} = -\frac{\mathrm{tr}(\mathbf{W}\mathbf{P}\mathbf{h}_d^T\mathbf{h}_d\mathbf{P}\mathbf{W})}{\mathbf{h}_d\mathbf{P}\mathbf{h}_d^T + \mathbf{R}_d} \\ &= -\frac{a_1\sin(2\theta) + a_2\cos(2\theta) + a_3}{a_4\sin(2\theta) + a_5\cos(2\theta) + a_6}\,, \end{aligned} \tag{6.24}$$

where the prior covariance is a constant matrix with elements:

$$\mathbf{P} = \begin{bmatrix} p_1 & p_4 & p_5 \\ p_4 & p_2 & p_6 \\ p_5 & p_6 & p_3 \end{bmatrix}, \tag{6.25}$$

and

$$
\begin{aligned}
a_1 &= p_1 p_4 + p_2 p_4 + d_o^4 p_5 p_6 \\
a_2 &= (p_1^2 - p_2^2 + d_o^4 p_5 - d_o^4 p_6)/2 \\
a_3 &= (p_1^2 + p_2^2 + 2p_4^2 + d_o^4 p_5^2 + d_o^4 p_6^2)/2 \\
a_4 &= p_4,\ a_5 = (p_1 - p_2)/2,\ a_6 = (p_1 + p_2)/2 + \sigma_d^2 \ .
\end{aligned}
$$

Using the quotient rule, the gradient of the objective function f_d is

$$
\nabla_\theta f_d = -\frac{u'v - uv'}{v^2} = -\frac{2(\sqrt{b_1^2 + b_2^2}\sin(2\theta + \varphi) + b_3)}{v^2} \ ,
$$

where

$$
\begin{aligned}
b_1 &= a_3 a_5 - a_2 a_6,\ b_2 = a_1 a_6 - a_3 a_4 \\
b_3 &= a_1 a_5 - a_2 a_4,\ \varphi = atan2(b_2, b_1).
\end{aligned}
$$

Since the denominator v is always nonzero, $\nabla_\theta f_d = 0$ is equivalent to

$$
\sqrt{b_1^2 + b_2^2}\sin(2\theta + \varphi) + b_3 = 0 \ .
$$

For $\theta \in (-\pi - \frac{\varphi}{2}, \pi - \frac{\varphi}{2}]$, there are four critical points for $2\theta + \varphi \in (-2\pi, 2\pi]$,

$$
\begin{aligned}
2\theta_1^* + \varphi &= \pi - \alpha, \quad \theta_2^* + \varphi = -\pi - \alpha \\
2\theta_3^* + \varphi &= \alpha, \qquad \theta_4^* + \varphi = -2\pi + \alpha \\
\alpha &= \arcsin(-\frac{b_3}{\sqrt{b_1^2 + b_2^2}})
\end{aligned}
$$

two of which are the minima:

$$
\theta_1^* = \frac{\pi - \alpha - \varphi}{2}, \quad \theta_2^* = \frac{-\pi - \alpha - \varphi}{2}.
$$

If θ_1^* or θ_2^* satisfies the constraint (6.23), then the optimal solution is θ_1^* or θ_2^*. Otherwise, the optimal solution is θ_{min} or θ_{max} (both of which can be easily computed from the geometry of Fig. 6.3) whichever yields a smaller value for the objective function.

6.3.2 Bearing-Only

For the bearing-only case, the measurement Jacobian $\mathbf{h}_\beta$ [see (6.17)] depends on both θ and d. Therefore, contrary to the distance-only case, the objective function for the bearing-only case is an explicit function of both d and θ. Using the same parameterization employed in the distance-only case, we express the optimization variable $\mathbf{p}$ using polar coordinates, i.e., $\mathbf{p} := d[\cos(\theta)\ \ \sin(\theta)]^T$. Transforming the optimization problem Π_1 for the bearing-only case into polar coordinates, results in the following problem

- Optimization Problem 3 (Π_3)

$$\min_{d,\theta}.\ \ f_\beta = -\frac{c_u\, d^{\,2} + 2\,\mathbf{b}_u^T\,\boldsymbol{\alpha}\, d + \boldsymbol{\alpha}^T\mathbf{A}_u\boldsymbol{\alpha}}{c_v\, d^{\,2} + 2\,\mathbf{b}_v^T\,\boldsymbol{\alpha}\, d + \boldsymbol{\alpha}^T\mathbf{A}_v\boldsymbol{\alpha}} \tag{6.26}$$

$$\text{s. t.}\ \ g_\beta = (d\cos(\theta) - x_o)^2 + (d\sin(\theta) - y_o)^2 - r^2 \leq 0\ , \tag{6.27}$$

where $\boldsymbol{\alpha} = [-\sin(\theta)\ \ \cos(\theta)]^T$ and $\mathbf{A}_u, \mathbf{b}_u, c_u, \mathbf{A}_v, \mathbf{b}_v$, and c_v are known parameters expressed in terms of $\mathbf{P}$, d_o, and σ_b^2 (see [91] for their specific expressions).

In order to solve Π_3, we proceed as follows: We first determine *all critical/stationary points* by solving systems of polynomial equations using an elimination procedure, and evaluate their objective values. Then, as optimal solution for Π_3 we select the critical point whose objective value is the smallest.

To proceed, we first construct the Lagrange function [9]

$$\mathcal{L}(d, \theta, \lambda) = f_\beta(d, \theta) + \lambda\ g_\beta(d, \theta)\ .$$

Based on the KKT necessary conditions, the critical points d^*, θ^*, and the associated Lagrange multiplier λ^* must satisfy

$$\nabla f_\beta(d^*, \theta^*) + \lambda^* \nabla g_\beta(d^*, \theta^*) = \mathbf{0}_{2\times 1} \tag{6.28}$$

$$\lambda^* g_\beta(d^*, \theta^*) = 0 \tag{6.29}$$

$$\lambda^* \geq 0\ , \tag{6.30}$$

where $\nabla f_\beta = [\nabla_d f_\beta \;\; \nabla_\theta f_\beta]^T$ and $\nabla g = [\nabla_d g_\beta \;\; \nabla_\theta g_\beta]^T$.

As shown in [91], both f_β and its derivative ∇f_β are *rational* functions with respect to d^*, $\cos(\theta^*)$, and $\sin(\theta^*)$, while g_β and ∇g_β are polynomials in those three variables. Thus, (6.28) can be transformed into a *polynomial* equation in d^*, $\cos(\theta^*)$, and $\sin(\theta^*)$ by requiring the numerator to be equal to zero. Therefore, computing all critical points of Π_3 is equivalent to solving the polynomial system defined by (6.28)-(6.29).

Similar to the distance-only problem, we examine two cases: (i) the constraint (6.27) is *inactive*, i.e., $g_\beta(d^*, \theta^*) < 0$; (ii) the constraint is *active*, i.e., $g_\beta(d^*, \theta^*) = 0$, and determine the global optimal solution by searching for the critical point that has the smallest objective function value.

Inactive Constraint

We first compute the set of all stationary points, denoted as $\mathbf{\Xi}_u$, by assuming that the constraint is inactive, or equivalently,

$$g_\beta(d^*, \theta^*) < 0 \;, \quad \forall \;\; (d^*, \theta^*) \in \mathbf{\Xi}_u \;. \tag{6.31}$$

Combining (6.31) and the complementary slackness condition (6.29), yields $\lambda^* = 0$. Hence, the term $\lambda^* \nabla g_\beta(d^*, \theta^*)$ in (6.28) vanishes. Thus, (6.28) and (6.29) are simplified into

$$\nabla f_\beta(d^*, \theta^*) = \begin{bmatrix} \nabla_d f_\beta(d^*, \theta^*) \\ \nabla_\theta f_\beta(d^*, \theta^*) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \;. \tag{6.32}$$

As shown in [91], (6.32) is equivalent to a system of two multivariate polynomial equations in the variables d^*, $\cos(\theta^*)$, and $\sin(\theta^*)$. The first polynomial, derived from $\nabla_d f_\beta(d^*, \theta^*) = 0$ and denoted as $\mathcal{F}_d$, has degree three; whereas the second polynomial $\mathcal{F}_\theta$, obtained through $\nabla_\theta f_\beta(d^*, \theta^*) = 0$, is of degree four.

In order to solve $\mathcal{F}_d = \mathcal{F}_\theta = 0$, we first treat $\cos(\theta^*)$ and $\sin(\theta^*)$ as parameters and

rewrite the $\mathcal{F}_d$ and $\mathcal{F}_\theta$ as polynomials in d^* [91]

$$\mathcal{F}_d = \zeta_2 (d^*)^2 + \zeta_1 d^* + \zeta_0 \tag{6.33}$$

$$\mathcal{F}_\theta = \gamma_3 (d^*)^3 + \gamma_2 (d^*)^2 + \gamma_1 d^* + \gamma_0 \ , \tag{6.34}$$

where $\zeta_i,\ i = 0, 1, 2$ and $\gamma_j,\ j = 0, \ldots, 3$ are bivariate polynomials in $\cos(\theta^*)$ and $\sin(\theta^*)$ [91].

We first eliminate the variable d^* using the Sylvester resultant [16, Ch. 3]. The Sylvester matrix of $\mathcal{F}_d$ and $\mathcal{F}_\theta$ with respect to d^*, denoted as $\mathbf{Syl}(\mathcal{F}_d, \mathcal{F}_\theta; d^*)$, is the following 5×5 matrix

$$\mathbf{Syl}(\mathcal{F}_d, \mathcal{F}_\theta; d^*) = \begin{bmatrix} \zeta_2 & & & \gamma_3 & \\ \zeta_1 & \zeta_2 & & \gamma_2 & \gamma_3 \\ \zeta_0 & \zeta_1 & \zeta_2 & \gamma_1 & \gamma_2 \\ & \zeta_0 & \zeta_1 & \gamma_0 & \gamma_1 \\ & & \zeta_0 & & \gamma_0 \end{bmatrix} .$$

The resultant of $\mathcal{F}_d$ and $\mathcal{F}_\theta$ with respect to d^*, denoted as $\mathbf{Res}(\mathcal{F}_d, \mathcal{F}_\theta; d^*)$, is the determinant of the Sylvester matrix $\mathbf{Syl}(\mathcal{F}_d, \mathcal{F}_\theta; d^*)$. Furthermore, since $\zeta_i,\ i = 0, 1, 2$ and $\gamma_j,\ j = 0, \ldots, 3$ are bivariate polynomials of $\cos(\theta^*)$ and $\sin(\theta^*)$, $\mathbf{Res}(\mathcal{F}_d, \mathcal{F}_\theta; d^*)$ is also a bivariate polynomial of $\cos(\theta^*)$ and $\sin(\theta^*)$:

$$\mathcal{R}_u \left(\cos(\theta^*), \sin(\theta^*)\right) := \mathbf{Res}(\mathcal{F}_d, \mathcal{F}_\theta; d^*) = 0 \ . \tag{6.35}$$

Together with the trigonometric identity

$$\mathcal{I} := \cos^2(\theta^*) + \sin^2(\theta^*) - 1 = 0 \ , \tag{6.36}$$

we employ the Sylvester resultant again to eliminate one of the variables, $\cos(\theta^*)$, to obtain a univariate polynomial in $\sin(\theta^*)$:

$$\mathcal{S}_u := \mathbf{Res}\left(\mathcal{R}_u, \mathcal{I}; \cos(\theta^*)\right) = \sum_{j=0}^{11} \delta_j \sin^{2j}(\theta^*) = 0 \ , \tag{6.37}$$

where $\delta_j,\ j = 0, \ldots, 11$ are known coefficients expressed in terms of $\mathbf{P}$, d_o, and σ_b^2. Note that (6.37) does not contain odd-degree terms of $\sin(\theta^*)$. Therefore, (6.37) is equivalent to the following eleventh-order univariate polynomial in the variable $s := \sin^2(\theta^*)$

$$\mathcal{S}_u(s) = \sum_{j=0}^{11} \delta_j s^j = 0 \ . \tag{6.38}$$

The roots of the univariate polynomial $\mathcal{S}_u(s)$ correspond to the 11 eigenvalues of the associated 11×11 companion matrix $\mathbf{\Delta}$ [22]:

$$\mathbf{\Delta} = \begin{bmatrix} 0 & & & -\delta_0/\delta_{11} \\ 1 & 0 & & -\delta_1/\delta_{11} \\ & \ddots & & \vdots \\ & & 1 & -\delta_{10}/\delta_{11} \end{bmatrix} \ .$$

Note that since $s = \sin^2(\theta^*)$, we only need to consider the real solutions between 0 and 1 of (6.38). Once s is determined, both $\cos(\theta^*)$ and $\sin(\theta^*)$ can be computed using the trigonometric identity, which can have at most 4 solutions for θ^*. Finally, for each θ^*, we compute the coefficients $\zeta_j,\ j = 0, 1, 2$ [see (6.33)], and solve for d^* from (6.33), which can have at most 2 solutions. Thus, the set $\mathbf{\Xi}_u$ consisting of all critical points (d^*, θ^*), has at most 88 elements. However, we only need to consider those critical points satisfying the constraint $g(d^*, \theta^*) < 0$.

The next step is to evaluate the objective function $f_\beta(d, \theta)$ [see (6.26)] at all the points in $\mathbf{\Xi}_u$ and select the one with the smallest objective value as the global optimal solution of Π_3, for the case when the constraint is inactive. We refer to this optimal solution as (d_u^*, θ_u^*).

Active Constraint

When the constraint is active, i.e., $g_\beta = 0$, the complementary slackness condition (6.29) is automatically satisfied. Hence, (6.28) and (6.29) are simplified into

$$\nabla f_\beta(d^*, \theta^*) + \lambda^* \nabla g_\beta(d^*, \theta^*) = \mathbf{0}_{2\times 1} \tag{6.39}$$

$$g_\beta(d^*, \theta^*) = 0 \, . \tag{6.40}$$

We solve this systems of equation by an elimination procedure similar to the inactive constraint case. The difference is that the additional variable λ^* needs to be eliminated first. In order to do so, we multiply both sides of (6.39) with $(\nabla g_\beta(d^*, \theta^*))^T \mathbf{J}$, where $\mathbf{J} = \mathbf{C}(-\pi/2)$ and $\mathbf{C}(\cdot)$ is the 2×2 rotation matrix, and obtain a polynomial in only d^*, $\cos(\theta^*)$, and $\sin(\theta^*)$:

$$\mathcal{Q}(d^*, \cos(\theta^*), \sin(\theta^*)) = 0. \tag{6.41}$$

Combining (6.41) with (6.40) and the trigonometric identity (6.36), we conclude that the set of all stationary points, denoted as $\mathbf{\Xi}_c$, must satisfy

$$\mathcal{Q}(d^*, \cos(\theta^*), \sin(\theta^*)) = 0 \tag{6.42}$$

$$g_\beta(d^*, \cos(\theta^*), \sin(\theta^*)) = 0 \tag{6.43}$$

$$\mathcal{I}(\cos(\theta^*), \sin(\theta^*)) = 0 \, . \tag{6.44}$$

To solve the above system of polynomial equations, we employ the same strategy outlined in the inactive constraint case. Firstly, we use the Sylvester resultant to eliminate d^* from (6.42) and (6.43) and obtain an eighth-order bivariate polynomial $\mathcal{R}_c(\cos(\theta^*), \sin(\theta^*))$. Secondly, we eliminate $\cos(\theta^*)$ from $\mathcal{R}_c$ and $\mathcal{I}$, and arrive at a univariate polynomial in $\sin(\theta^*)$

$$\mathcal{S}_c\left(\sin(\theta^*)\right) = \sum_{j=0}^{8} \eta_j \sin^{2j}(\theta^*) = 0 \, , \tag{6.45}$$

where $\eta_j,\ j = 0, \ldots, 8$, are known coefficients expressed in terms of $\mathbf{P}$, $\mathbf{p}_o$, r, d_o, and σ_b^2. Since only even-degree terms appear in the above equation, it is equivalent to the following eighth-order univariate polynomial in $s := \sin^2(\theta^*)$

$$\mathcal{S}_c(s) = \sum_{j=0}^{8} \eta_j s^j = 0 \ , \tag{6.46}$$

Similar to the inactive constraint case, the roots of (6.46) can be calculated by computing the eigenvalues of its associated 8×8 companion matrix. Once s is determined, both $\cos(\theta^*)$ and $\sin(\theta^*)$ can be computed, which give at most 4 solutions for θ^*. Finally, for each θ^*, we solve for d^* from (6.40), which can have at most 2 solutions. Thus, the set $\mathbf{\Xi}_c$ consisting of all critical points (d^*, θ^*), has at most 64 elements.

The final step is to evaluate the objective function $f(d, \theta)$ [see (6.26)] at all the critical points in $\mathbf{\Xi}_c$ and select the one with the smallest objective value as the global optimal solution of Π_3, for the case when the constraint is active. We refer to this optimal solution as (d_c^*, θ_c^*).

Globally Optimal Solution for Π_3

Finally, the globally minimal solution $(d_{\text{opt}}, \theta_{\text{opt}})$ for Π_3 can be selected as

$$(d_{\text{opt}}, \theta_{\text{opt}}) = \begin{cases} (d_u^*, \theta_u^*) & : \quad f(d_u^*, \theta_u^*) \leq f(d_c^*, \theta_c^*) \\ (d_c^*, \theta_c^*) & : \quad f(d_u^*, \theta_u^*) > f(d_c^*, \theta_c^*) \end{cases}$$

6.4 Simulation Results

We have evaluated the performance of our motion strategy for a leader-follower formation in simulation. In particular, we consider the case where the leader moves on a straight line and the follower is supposed to maintain its position at $\mathbf{p}_o$ with respect to the leader. We have compared our motion strategy, the relaxed algebraic method (RAM), with four other strategies; namely (i) the "maintaining the formation" (MTF) strategy; (ii) a constrained random motion (CRM); (iii) the active control strategy of

Mariottini *et al.* [48] (MME); and (iv) grid-based search (GBS). The strategy MTF always moves the follower toward the desired position $\mathbf{p}_o$, while CRM moves the follower to a random position inside a circle centered at $\mathbf{p}_o$ with radius r. For GBS, the area inside the circle is discretized into cells, and the follower moves to the cell that has the lowest cost. GBS should be a benchmark for evaluating the performance of all motion strategies, if the discrete cell size is sufficiently small.

We have conducted Monte Carlo simulations with 50 trials with the following settings. The desired follower position is $\mathbf{p}_o = [-1, \ -2]^T$ for the robot with range measurements, and $\mathbf{p}_o = [-1, \ 2]^T$ for the robot with bearing measurements. The radius of the constraint circle is set to $r = 0.1||\mathbf{p}_o||_2$. The initial estimated pose of the follower is $\hat{\mathbf{x}}_{1|1} = [\mathbf{p}_o^T, \ 0]^T$, with covariance $\mathbf{P}_{1|1} = \text{diag}(0.1, \ 0.1, \ 4 \times 10^{-4})$. We use the discrete-time kinematic model (6.9) with $\delta t = 0.05$ s to simulate the leader and follower motions. The leader's linear velocity is set to 1 m/s, with rotational velocity 0 rad/s. The follower's control inputs, the linear and rotational velocities v_{Fm} and ω_{Fm}, are generated according to the next time-step position $\mathbf{p}_{k+1}$ determined by the specific motion strategy. In order to drive to $\mathbf{p}_{k+1}$, the follower first rotates with velocity ω_{Fm}, such that its heading points to $\mathbf{p}_{k+1}$. Then it executes a pure translation with velocity v_{Fm} to finally arrive at $\mathbf{p}_{k+1}$. The true follower's linear and rotational velocities are affected by zero-mean white Gaussian noise with standard deviation $\sigma_v = 0.2$ m/s and $\sigma_\omega = 0.1$ rad/s, respectively. At $\mathbf{p}_{k+1}$, the follower records a distance or bearing measurement to the leader, so as to update its pose estimate.

Fig. 6.4 shows the trajectories of RAM, GBS, and MME with one robot measuring distance to the leader and another robot measuring its bearing to the leader. The trajectories of MTF and CRM are not informative and therefore are omitted. RAM generates a zigzagging motion pattern which touches the circle constraint most of the time, while GBS also shows a similar motion pattern. The trajectory of MME is more irregular with larger position uncertainty depicted by the 3σ ellipses.

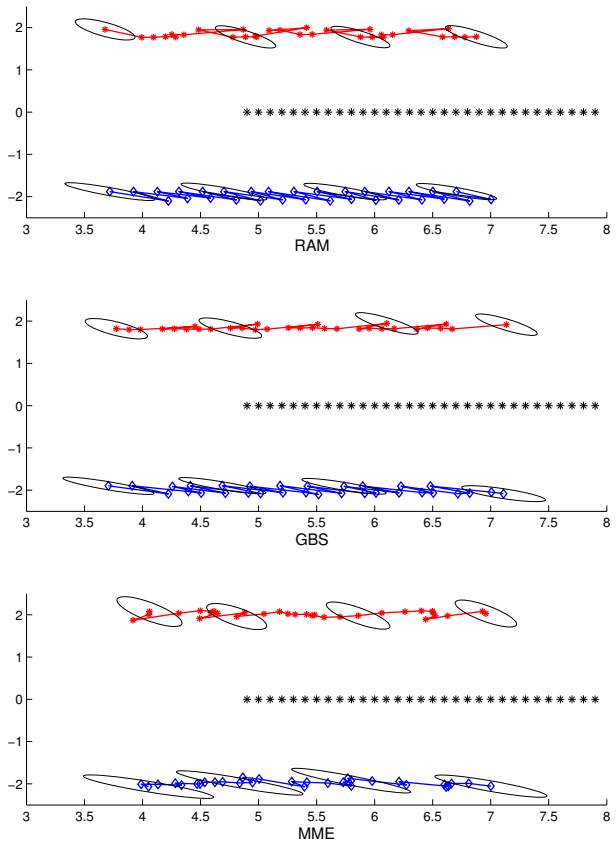


Figure 6.4: The estimated trajectories of the two followers and the leader, when employing RAM, GBS, or MME as motion strategy. The units of the x and y axes are meters. In each subplot, the follower equipped with a bearing sensor is on the top, the leader is in the middle, and the follower equipped with a range sensor is at the bottom. The ellipses denote the 3σ bounds for the follower's position uncertainty at the corresponding positions.

6.4.1 Distance-only case

Fig. 6.5 shows the follower's pose uncertainty of the aforementioned five motion strategies with distance measurement noise standard deviation $\sigma_d = 0.1$ m. The metric used for evaluating uncertainty is the weighted trace of the posterior covariance matrix

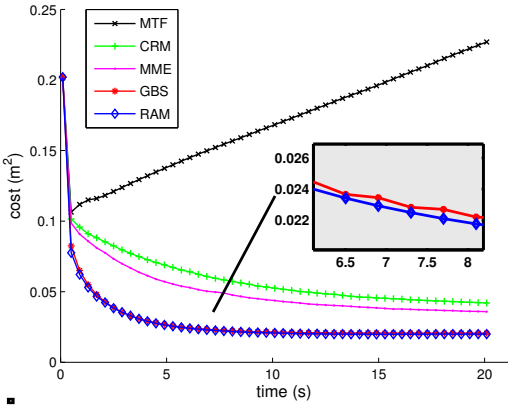


Figure 6.5: *Distance-only case, Monte Carlo simulations:* Average weighted trace of the follower's posterior covariance matrix in 50 experiments.

(see Section 6.2.4). Since the relative pose is unobservable when the follower moves in exact formation, the uncertainty of MTF increases unbounded over time. When the follower moves around the desired position, the relative pose becomes observable, which is demonstrated by the convergence of the uncertainty to almost steady-state values for the other four motion strategies. However, the specific motion strategy employed affects the estimate's uncertainty. CRM shows the highest uncertainty, closely followed by the MME strategy. This result is expected, because these two strategies are designed to gain observability, rather than minimize the uncertainty. Our motion strategy, RAM, is designed to minimize uncertainty and shows superior performance compared to CRM and MME. From the zoom-in view of the uncertainty curves of RAM and GBS, we can see that our strategy is even slightly better than the benchmark strategy GBS. This is due to the fact that given limitations on how small the grid-cell size may be, the GBS may not reach the global minimum.

6.4.2 Bearing-only case

We hereafter examine the performance of the RAM motion strategy for the bearing-only measurement model. In our simulation, the standard deviation of the bearing

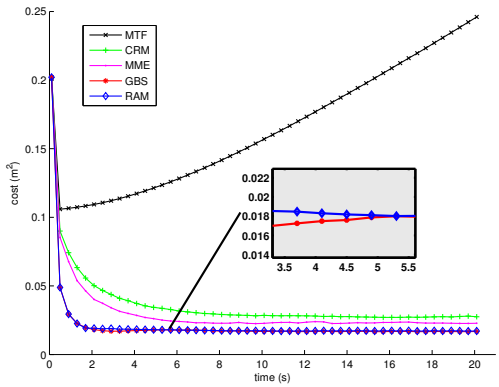


Figure 6.6: *Bearing-only case, Monte Carlo simulations:* Average weighted trace of the follower's posterior covariance matrix in 50 experiments.

measurement noise is set to $\sigma_\beta = 0.02$ rad.

The time evolution of the weighted trace of the follower's posterior covariance matrix (i.e., the cost function) averaged over 50 experiments is shown in Fig. 6.6. As expected, the performance of RAM and GBS is improved compared to the cases of CRM and MME, and is significantly better than that of the unobservable case MTF. Additionally, the uncertainty in the follower's pose estimates (weighted trace of the covariance matrix) achieved by the proposed RAM motion strategy is indistinguishable of that of the GBS.

6.5 Summary

In this chapter, we presented a new approach to generate optimal motion strategies for a leader-follower formation. Specifically, we considered the case where the follower measures the distance or bearing to the leader, while the leader is moving on a straight line. Since the follower's pose is unobservable, when it maintains a constant position with respect to the leader, we allowed the follower to deviate slightly from its desired position, and determined its optimal next time-step position. In particular, we minimized the weighted trace of the posterior covariance matrix, and computed the global optimal

solution analytically. Even though we focused on the case where the leader is moving on a straight line, our optimization method can be applied to arbitrary leader motions. The simulation results demonstrate superior localization accuracy compared to existing follower motion strategies, while showing indistinguishable performance compared to a grid-based search strategy.

As part of our future work, we plan to investigate the case where the robots have access to vector measurements. In this case, the optimization problem becomes significantly more challenging, due to the trade-off between different sources of information. In addition, we are also interested in solving for the optimal follower positions for more than one time step. Optimizing over a longer time horizon is expected to further increase the localization accuracy at the expense of increased complexity.

Chapter 7

Conclusion and Discussion

The work presented in the previous chapters has focused on introducing new methods for determining the initial robot-to-robot transformation in a multi-robot team, which is a critical prerequisite for any multi-robot sensor fusion tasks. In particular, we addressed the problem of motion-induced robot-to-robot extrinsic calibration, where combinations of distance and/or bearing measurements between pairs of robots collected at multiple vantage points are used to compute their relative pose. Our first main contribution is to provide efficient solutions for initializing the relative robot pose without external references. Our second contribution is to provide a closed-form solution for determining the relative transformation between two camera views using point features and directional correspondences. Moreover, since the locations where the robots collect measurements affect their accuracy, we studied the problem of localization in a leader-follower formation and determined the optimal motion strategy for the follower so that its localization uncertainty is minimized.

7.1 Summary of Contributions

Specifically, in Chapter 3 we first solved the problem of determining the 3-DOF relative pose of robots moving in a plane while measuring their distance to each other. Non-iterative algorithms for computing the initial estimate of the 3-DOF transformation were presented for the cases when 3, 4, and 5 distance measurements were available. We have shown that for nonsingular configurations the maximum number of solutions for the above cases are 6, 4, and 1, respectively. Furthermore, we studied the observability of the nonlinear system describing the robots' 3-DOF relative pose, given odometric and distance measurements, and determined the *necessary and sufficient* conditions on the motions of the robots for their relative pose to be observable.

In Chapter 4, we solved the more general problem of motion-induced robot-to-robot extrinsic calibration in 3D using combinations of distance and bearing measurements over multiple time steps (e.g., the robots measure distance at the first time step, bearing at the second time step, etc.). We proved that only 14 minimal systems need to be considered, out of a large number of possible measurement combinations, while all other problems, resulting from different combinations of distance and bearing measurements, can be solved using the solutions of these 14 systems. Furthermore, we provided closed-form and efficient analytical solutions to these 14 problems.

In Chapter 5, we presented a closed-form solution for determining the relative pose of a camera between two views using point features and directional correspondences. In particular, we provided an efficient algebraic solution for the minimal problem (i.e., based on 3 point features and 1 directional correspondence), which is often used in conjunction with RANSAC for outlier rejection. Moreover, we provided analytical solutions to the corresponding least-squares problem, where only inlier measurements are considered.

Finally, in Chapter 6 we studied the trade-off between localization accuracy and maintaining a leader-follower formation. Specifically, we employed algebraic geometry

methods to analytically compute the location where the follower should collect its next measurement, so that its localization uncertainty is minimized.

7.2 Future Research Directions

The work on optimal motion strategies for localization in leader-follower formations opens up a number of interesting research directions. First of all, a straightforward extension would be to study the same problem for robots moving in 3D using distance and/or bearing measurements. Secondly, it would be interesting to investigate optimal motion strategies for robot teams that do not have to adhere to formation constraints. In addition, it would be desirable to not only optimize the motion for pair of robots, but also to seek optimal motion strategies for the entire team. Due to the large number of optimization variables, such a problem would be very difficult to solve. Instead, it would be desirable to determine approximate solutions computable in real time that can quantify the loss of precision due to the relaxations invoked.

References

[1] "Autonomous ocean sampling network," http://www.mbari.org/aosn/.

[2] "Cluster ESA/NASA mission." [Online]. Available: http://sci.esa.int/science-e/www/area/index.cfm?fareaid=8

[3] A. G. Akritas, "Sylvester's form of resultant and the matrix-triangularization subresultant prs method," in *Proceedings of Conference on Computer Aided Proofs in Analysis*, Cincinnati, OH, Mar. 1989, pp. 5–11.

[4] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 578–589, May 2003.

[5] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, "A theory of network localization," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1663–1678, Dec. 2006.

[6] P. Bahl and V. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of IEEE Computer and Communications Society*, Tel Aviv, Israel, Mar. 26-30, 2000, pp. 775–784.

[7] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, Dec. 1998.

[8] J. Bender, "An overview of systems studies of automated highway systems," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 82–99, Feb. 1991.

[9] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[10] F. Beutler and U. D. Hanebeck, "Closed-form range-based posture estimation based on decoupling translation and orientation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, Mar. 18–23, 2005, pp. 989–992.

[11] J. F. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

[12] J. L. Casti, "Recent developments and future perspectives in nonlinear system theory," *SIAM Review*, vol. 24, no. 3, pp. 301–331, Jul. 1982.

[13] H. H. Chen, "Pose determination from line-to-plane correspondences: existence condition and closed-form solutions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 530–541, Jun. 1991.

[14] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 39–64, Feb. 2006.

[15] D. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*, 2nd ed. Springer, 2004.

[16] ——, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, 3rd ed. New York: Springer, 2007.

[17] J. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 16–20 1998, pp. 2864–2869.

[18] M. Di Marco, A. Garulli, A. Giannitrapani, and A. Vicino, "Simultaneous localization and map building for a team of cooperating robots: a set membership approach," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 238–249, Apr. 2003.

[19] P. Dietmaier, "The Stewart-Gough platform of general geometry can have 40 real postures," in *Advances in Robot Kinematics: Analysis and Control*, J. Lenarcic and M. L. Husty, Eds. Kluwer Academic Publishers, 1998, pp. 7–16.

[20] J. Djugash, S. Singh, and P. I. Corke, "Further results with localization and mapping using range from radio," in *Proceedings of the International Conference on Field and Service Robotics*, Jul. 29–31, 2005.

[21] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of INFOCOM* 20^{th} *Annual Joint Conference of IEEE Computer and Communications Society*, Anchorage, AK, Apr. 22–26, 2001, pp. 1655–1663.

[22] A. Edelman and H. Murakami, "Polynomial roots from companion matrix eigenvalues," *Mathematics of Computation*, vol. 64, no. 210, pp. 763–776, Apr. 1995.

[23] G. Ehrlich, *Fundamental Concepts of Abstract Algebra*. Boston: PWS-Kent Publishing Company, 1991.

[24] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, computation, and randomization in

network localization," in *Proceedings of INFOCOM* 23rd *Annual Joint Conference of the IEEE Computer and Communications Societies*, Hong Kong, Mar. 7–11, 2004, pp. 2673–2684.

[25] J. Fenwick, P. Newman, and J. Leonard, "Cooperative concurrent mapping and localization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC, May 11–15, 2002, pp. 1810–1817.

[26] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[27] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, Jun. 2000.

[28] F. Fraundorfer, P. Tanskanen, and M. Pollefeys, "A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Crete, Greece, Sep. 5–11 2010, pp. 269–282.

[29] L. Girod and D. Estrin, "Robust range estimation using acoustic and multimodal sensing," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, HI, Oct. 29 - Nov. 3 2001, pp. 1312–1320.

[30] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Transactions on Automatic Control*, vol. 22, no. 5, pp. 728–740, Oct. 1977.

[31] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (DLS) solution for PnP," in *Proceedings of the International Conference on Computer Vision*, Barcelona, Spain, Nov. 6–13, 2011.

[32] Y. Hidaka, A. Mourikis, and S. Roumeliotis, "Optimal formations for cooperative localization of mobile robots," in *Proc. IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 18–22 2005, pp. 4126–4131.

[33] S. Higo, T. Yoshimitsu, and I. Nakatani, "Localization on small body surface by radio ranging," in *Proceedings of the 16th AAS/AIAA Space Flight Mechanics Conference*, Tampa, FL, Jan. 22–26, 2006.

[34] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, Apr. 1987.

[35] ——, "Closed-form solution of absolute orientation using orthonormal matrices," *Journal of the Optical Society of America A*, vol. 5, no. 7, pp. 1127–1135, Jul. 1988.

[36] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Putting the 'I' in 'team': an ego-centric approach to cooperative localization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sep. 14–19, 2003, pp. 868–874.

[37] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection," *International Journal of Robotics Research*, vol. 25, no. 5–6, pp. 431–447, May 2006.

[38] M. L. Husty, "An algorithm for solving the direct kinematics of general Stewart-Gough platforms," *Mechanism and Machine Theory*, vol. 31, no. 4, pp. 365–380, May 1996.

[39] C. Innocenti, "Forward kinematics in polynomial form of the general Stewart platform," *Journal of Mechanical Design*, vol. 123, no. 2, pp. 254–260, Jun. 2001.

[40] J. Jennings, G. Whelan, and W. Evans, "Cooperative search and rescue with a team of mobile robots," in *Proceedings of the International Conference on Advanced Robotics*, Monterey, CA, Jul. 7–9 1997, pp. 193–200.

[41] B. Jung and G. S. Sukhatme, "A generalized region-based approach for multi-target tracking in outdoor environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, Apr. 26–May 1, 2004, pp. 2189–2195.

[42] ——, "Tracking targets using multiple robots: The effect of environment occlusion," *Autonomous Robots*, vol. 13, no. 3, pp. 191–205, Nov. 2002.

[43] M. Kalantari, A. Hashemi, F. Jung, and J.-P. Guedon, "A new solution to the relative orientation problem using only 3 points and the vertical direction," *Journal of Math Imaging*, vol. 39, no. 3, pp. 259–268, Nov. 2011.

[44] R. Kurazume and S. Hirose, "Study on cooperative positioning system: optimum moving strategies for CPS-III," in *Proceeding of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 16–20 1998, pp. 2896–2903.

[45] T.-Y. Lee and J.-K. Shim, "Improved dialytic elimination algorithm for the forward kinematics of the general Stewart-Gough platform," *Mechanism and Machine Theory*, vol. 38, no. 6, pp. 563–577, Jun. 2003.

[46] R. Madhavan, K. Fregene, and L. E. Parker, "Distributed heterogeneous outdoor multi-robot localization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC, May 11–15, 2002, pp. 374–381.

[47] ——, "Distributed cooperative outdoor multirobot localization and mapping," *Autonomous Robots*, vol. 17, no. 1, pp. 23–39, Jul. 2004.

[48] G. L. Mariottini, S. Martini, and M. B. Egerstedt, "A switching active sensing strategy to maintain observability for vision-based formation control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12–17 2009, pp. 2637–2642.

[49] G. Mariottini, G. Pappas, D. Prattichizzo, and K. Daniilidis, "Vision-based localization of leader-follower formations," in *Proceedings of the IEEE International Conference on Decision and Control*, Seville, Spain, Dec. 12–15 2005, pp. 635–640.

[50] A. Martinelli and R. Siegwart, "Observability analysis for mobile robot localization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, Aug. 2–6, 2005, pp. 1471–1476.

[51] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. New York: Academic Press, 1979, vol. 1.

[52] M. Mazo Jr, A. Speranzon, K. H. Johansson, and X. Hu, "Multi-robot tracking of a moving object using directional sensors," in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, Apr. 26–May 1, 2004, pp. 1103–1108.

[53] F. M. Mirzaei and S. I. Roumeliotis, "Globally optimal pose estimation from line-to-line correspondences," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9 - 13, 2011, pp. 5581–5588.

[54] A. I. Mourikis, S. I. Roumeliotis, and J. W. Burdick, "SC-KF mobile robot localization: A stochastic cloning-kalman filter for processing relative-state measurements," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 717–730, Aug. 2007.

[55] O. Naroditsky, X. S. Zhou, J. Gallier, S. I. Roumeliotis, and K. Daniilidis, "Two efficient solutions for visual odometry using directional correspondence," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 4, pp. 818–824, 2012.

[56] J. Nie, "Sum of squares method for sensor network localization," *Computational Optimization and Applications*, vol. 43, no. 2, pp. 1573–2894, Jun. 2009.

[57] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[58] C.-H. Ou and K.-F. Ssu, "Sensor position determination with flying anchors in three-dimensional wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 9, pp. 1084–1097, Sep. 2008.

[59] L. E. Parker, "Distributed algorithms for multi-robot observation of multiple moving targets," *Autonomous Robots*, vol. 12, no. 3, pp. 231–255, May 2002.

[60] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992.

[61] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, Boston, MA, Aug. 6 - 11 2000, pp. 32–43.

[62] L. Quan and Z.-D. Lan, "Linear n-point camera pose determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 774–780, Aug. 1999.

[63] G. Reid and L. Zhi, "Solving polynomial systems via symbolic-numeric reduction to geometric involutive form," *Journal of Symbolic Computation*, vol. 44, no. 3, pp. 280–291, Mar. 2009.

[64] I. Rekleitis, G. Dudek, and E. Milios, "Probabilistic cooperative localization and mapping in practice," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sep.14-19, 2003, pp. 1907–1912.

[65] ——, "Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, Lausanne, Switzerland, Sep. 30– Oct. 5 2002, pp. 2690–2695.

[66] W. Respondek, "Geometry of static and dynamic feedback," Lecture Notes at the Summer School on Mathematical Control Theory, Trieste, Italy, Sep. 2001.

[67] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, Oct. 2002.

[68] S. I. Roumeliotis and J. W. Burdick, "Stochastic cloning: A generalized framework for processing relative state measurements," in *Proccedings of the IEEE International Conference on Robotics and Automation*, Washington D.C., May 11-15 2002, pp. 1788–1795.

[69] S. I. Roumeliotis and G. A. Bekey, "Distributed multi-robot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, Oct. 2002.

[70] W. J. Rugh, *Linear System Theory*, 2nd ed. Prentice Hall, 1996.

[71] A. Rynn, W. A. Malik, and S. Lee, "Sensor based localization for multiple mobile robots using virtual links," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, Oct. 27–31, 2003, pp. 1771–1776.

[72] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *Proceedings of the*

1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, Sep. 28, 2002.

[73] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proceedings of MobiHoc* 4^{th} *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Annapolis, MD, Jun. 1–3, 2003, pp. 201–212.

[74] J. Spletzer, A. K. Das, R. Fierro, C. J. Taylor, V. Kumar, and J. P. Ostrowski, "Cooperative localization and control for multi-robot manipulation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, HI, Oct. 29– Nov. 3 2001, pp. 631–636.

[75] D. Stewart, "A platform with six degrees of freedom: A new form of mechanical linkage which enables a platform to move simultaneously in all six degrees of freedom developed by elliott-automation," *Aircraft Engineering and Aerospace Technology*, vol. 38, no. 4, pp. 30–35, 1993.

[76] I. Stewart, *Galois Theory*, 3rd ed. Chapman and Hall/CRC Mathematics Series, 2003.

[77] A. Stroupe, M. Martin, and T. Balch, "Distributed sensor fusion for object position estimation by multi-robot systems," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 21–26, 2001, pp. 1092–1098.

[78] P. Sturm, "Algorithms for plane-based pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, SC, Jun. 13–15, 2000, pp. 706–711.

[79] H. Sugiyama, T. Tsujioka, and M. Murata, "Coordination of rescue robots for real-time exploration over disaster areas," in *Proceedings of the* 11^{th} *IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, Orlando, FL, May 5–7, 2008, pp. 170–177.

[80] K.-H. Tan and M. Lewis, "Virtual structures for high-precision cooperative mobile robotic control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, Japan, Nov. 4–8 1996, pp. 132–139.

[81] S. Thrun, "A probabilistic on-line mapping algorithm for teams of mobile robots," *International Journal of Robotics Research*, vol. 20, no. 5, p. 335, May 2001.

[82] N. Trawny and T. Barfoot, "Optimized motion strategies for cooperative localization of mobile robots," in *Proceeding of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, Apr. 26 – May 1 2004, pp. 1027–1032.

[83] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep. 2005-002, Jan. 2005.

[84] N. Trawny, X. S. Zhou, and S. I. Roumeliotis, "3D relative pose estimation from six distances," in *Proceedings of Robotics: Science and Systems*, Seattle, WA, Jun. 28 – Jul. 1, 2009, pp. 233–240.

[85] N. Trawny, X. S. Zhou, K. X. Zhou, and S. I. Roumeliotis, "3D relative pose estimation from distance-only measurements," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, Oct. 29 – Nov. 2, 2007, pp. 1071–1078.

[86] ——, "Interrobot transformations in 3D," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 225–243, Apr. 2010.

[87] J. Verschelde, "Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation," *ACM Transactions on Mathematical Software*, vol. 25, no. 2, pp. 251–276, 1999.

[88] C. W. Wampler, "Forward displacement analysis of general six-in-parallel SPS (Stewart) platform manipulators using soma coordinates," *Mechanism and Machine Theory*, vol. 31, no. 3, pp. 331–337, Apr. 1996.

[89] F. Zhang, B. Grocholsky, and V. Kumar, "Formations for localization of robot networks," in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, Apr. 26–May 1 2004, pp. 3369–3374.

[90] L. Zhang, X. Zhou, and Q. Cheng, "Landscape-3D: A robust localization scheme for sensor networks over complex 3D terrains," in *Proccedings of 31st IEEE Conference on Local Computer Networks*, Tampa, FL, Nov. 14–17, 2006, pp. 239–246.

[91] K. Zhou and S. I. Roumeliotis, "Optimized motion strategy for bearing-only localization in leader-follower formations," MARS Lab., Dept. of Computer Science and Engineering, University of Minnesota, Tech. Rep., Mar. 2011, http://mars.cs.umn.edu/tr/reports/Ke11.pdf.

[92] X. S. Zhou and S. I. Roumeliotis, "Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 9–15, 2006, pp. 1785–1792.

[93] ——, "Robot-to-robot relative pose estimation from range measurements," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1379–1393, Dec. 2008.

[94] ——, "Determining the robot-to-robot relative pose using range and/or bearing measurements," University of Minnesota, Dept. of Comp. Sci.

& Eng., MARS Lab, Tech. Rep., Feb. 2010. [Online]. Available: www.cs.umn.edu/~zhou/paper/14systech.pdf

[95] X. S. Zhou, K. X. Zhou, and S. I. Roumeliotis, "Optimized motion strategies for localization in leader-follower formations," in *Proceeding of the IEEE/RSJ International Conference on Intellent Robots and Systems*, San Francisco, California, Sep. 25–30 2011, pp. 98–105.

[96] X. S. Zhou and S. I. Roumeliotis, "Determining the robot-to-robot relative pose using range-only measurements," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 10 – 14, 2007, pp. 4025–4031.

[97] ——, "Determining the robot-to-robot relative pose using range-only measurements," University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Minneapolis, MN, Tech. Rep., May 2007. [Online]. Available: www.cs.umn.edu/~zhou/paper/distOnly.pdf

[98] ——, "Determining the robot-to-robot 3D relative pose using combinations of range and bearing measurements: 14 minimal problems and closed-form solutions to three of them," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 18–22, 2010, pp. 2983 – 2990.

[99] ——, "Determining the robot-to-robot 3D relative pose using combinations of range and bearing measurements (part II)," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9–13, 2011.

[100] ——, "Determining 3D relative transformations for any combination of range and bearing measurements," *IEEE Transactions on Robotics*, 2012, submitted.

Appendix

In what follows, we show that in equation (4.87), $\mathbf{C}(\mathbf{u}_1, \theta) = \mathbf{C}(\mathbf{u}_1, 180^\circ)$. Using the Rodrigues formula for $\mathbf{C}(\mathbf{v}_1, 180^\circ)$, we have

$$\begin{aligned}\mathbf{C}(\mathbf{u}_1, \theta) &= \mathbf{C}_1^T \mathbf{C}(\mathbf{v}_1, 180^\circ)\mathbf{C}_2 \\ &= \mathbf{C}(\mathbf{e}_2, \beta_1 - \beta_2)^T(-\mathbf{I} + 2\mathbf{v}_1\mathbf{v}_1^T)\mathbf{C}(\mathbf{e}_2, \beta_1 + \beta_2)\end{aligned}$$

Substituting $\mathbf{v}_1 = \mathbf{C}(\mathbf{e}_2, \beta_1)\mathbf{u}_1$ (see Fig. 4.4) in the above equation, we have

$$\begin{aligned}\mathbf{C}(\mathbf{u}_1, \theta) &= -\mathbf{C}(\mathbf{e}_2, 2\beta_2) + 2\mathbf{C}(\mathbf{e}_2, \beta_2)\mathbf{u}_1\mathbf{u}_1^T\mathbf{C}(\mathbf{e}_2, \beta_2) \\ &= \mathbf{C}(\mathbf{e}_2, \beta_2)(-\mathbf{I} + 2\mathbf{u}_1\mathbf{u}_1^T)\mathbf{C}(\mathbf{e}_2, \beta_2) \\ &= \mathbf{C}(\mathbf{e}_2, \beta_2)\mathbf{C}(\mathbf{u}_1, 180^\circ)\mathbf{C}(\mathbf{e}_2, \beta_2) \\ &= \mathbf{C}(\mathbf{e}_2, \beta_2)\mathbf{C}(\mathbf{u}_1, 180^\circ)\mathbf{C}(\mathbf{e}_2, \beta_2)\mathbf{C}(\mathbf{u}_1, -180^\circ) \\ &\quad \cdot \mathbf{C}(\mathbf{u}_1, 180^\circ) \qquad (7.1) \\ &= \mathbf{C}(\mathbf{e}_2, \beta_2)\mathbf{C}(\mathbf{C}(\mathbf{u}_1, 180^\circ)\mathbf{e}_2, \beta_2)\mathbf{C}(\mathbf{u}_1, 180^\circ) \qquad (7.2) \\ &= \mathbf{C}(\mathbf{e}_2, \beta_2)\mathbf{C}(-\mathbf{e}_2, \beta_2)\mathbf{C}(\mathbf{u}_1, 180^\circ) \\ &= \mathbf{C}(\mathbf{u}_1, 180^\circ)\end{aligned}$$

where from (7.1) to (7.2) we have used the relation

$$\mathbf{C}(\mathbf{u}_1, 180^\circ)\mathbf{C}(\mathbf{e}_2, \beta_2)\mathbf{C}(\mathbf{u}_1, -180^\circ) = \mathbf{C}(\mathbf{C}(\mathbf{u}_1, 180^\circ)\mathbf{e}_2, \beta_2).$$

Hence, $\theta = 180^\circ$.