CONTROL AND NAVIGATION OF MULTI-VEHICLE SYSTEMS USING VISUAL MEASUREMENTS

SHIYU ZHAO

(M.Eng., Beihang University, 2009)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

NUS GRADUATE SCHOOL FOR INTEGRATIVE

SCIENCES AND ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2014

Declaration

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

SHIYU ZHAO
2 January 2014

Acknowledgments

When looking back on the past four years at National University of Singapore, I am surprised see that I have grown up in many ways. I would like to thank everyone who has helped me along the way of my PhD.

First of all, I would like to express my heartfelt gratitude to my main supervisor, Professor Ben M. Chen, who taught me essential skills to survive in academia. I will always remember his patient guidance, selfless support, and precious edification. I am also grateful to my co-supervisors, Professor Tong H. Lee and Dr. Chang Chen, for their kind encouragement and generous help. I sincerely thank my Thesis Advisory Committee members, Professor Jianxin Xu and Professor Delin Chu, for the time and efforts they have spent on advising my research work.

Special thanks are given to Dr. Kemao Peng and Dr. Feng Lin, who are not only my colleagues but also my best friends. I appreciate their cordial support on my PhD study. I also would like to express my gratitude to the NUS UAS Research Group members including Xiangxu Dong, Fei Wang, Kevin Ang, Jinqiang Cui, Swee King Phang, Kun Li, Shupeng Lai, Peidong Liu, Yijie Ke, Kangli Wang, Di Deng, and Jing Lin. It is my honor to be a member of this harmonious and vigorous research group. I also wish to thank Professor Kai-Yew Lum at National Chi Nan University and Professor Guowei Cai at Khalifa University for their help on my research.

Finally, I need to thank my wife Jie Song and my parents. Without their wholehearted support, it would be impossible for me to finish my PhD study.

Contents

Summary

Computer vision techniques have been widely applied to control and navigation of autonomous vehicles nowadays. It is worth noting that vision inherently is a *bearing-only* sensing approach: it is easy for vision to obtain the bearing of a target relative to the camera, but much harder to obtain the distance from the target to the camera. Due to the bearing-only property of visual sensing, many interesting research topics arise in control and navigation of multi-vehicle systems using visual measurements. In this thesis, we will study several important ones of these topics.

The thesis consists of three parts. The topic addressed in each part is an interdisciplinary topic of control/navigation and computer vision. The three parts are summarized as below.

1) The first part of the thesis studies optimal placement of sensor networks for target localization and tracking. When localizing a target using multiple sensors, the placement of the sensors can greatly affect the target localization accuracy. Although optimal placement of sensor networks has been studied by many researchers, most of the existing results are only applicable to 2D space. Our main contribution is that we proved the necessary and sufficient conditions for optimal placement of sensor networks in both 2D and 3D spaces. We have also established a unified framework for analyzing optimal placement of different types of sensor networks.

2) The second part of the thesis investigates bearing-only formation control. Although a variety of approaches have been proposed in the literature to solve vision-based formation control, very few of them can be applied to practical applications. That is mainly because the conventional approaches treat vision

as a powerful sensor and hence require complicated vision algorithms, which heavily restrict real-time and robust implementations of these approaches in practice. Motivated by that, we treat vision as a bearing-only sensor and then formulate vision-based formation control as bearing-only formation control. This formulation poses minimal requirements on the end of vision and can provide a practical solution to vision-based formation control. In our work, we have proposed a distributed control law to stabilize cyclic formations using bearing-only measurements. We have also proved the local formation stability and local collision avoidance.

3) The third part of the thesis explores vision-based navigation of unmanned aerial vehicles (UAVs). This part considers two scenarios. In the first scenario, we assume the environment is unknown. The visual measurements are fused with the measurements of other sensors such as a low-cost inertial measurement unit (IMU). Our proposed vision-based navigation system is able to: firstly online estimate and compensate the unknown biases in the IMU measurements; secondly provide drift-free velocity and attitude estimates which are crucial for UAV stabilization control; thirdly reduce the position drift significantly compared to pure inertial navigation. In the second scenario, we assume there are artificial landmarks in the environment. The vision system is required to estimate the position of the UAV relative to the artificial landmarks without the assistance of any other sensors. In our work, the artificial landmarks are chosen as circles with known diameters. We have developed a robust and real-time vision system to navigate a UAV based on the circles. This vision system has been applied to the 2013 International UAV Grand Prix and helped us making a great success in this competition.

List of Tables

List of Figures

Chapter 1

Introduction

New advancements in the fields of computer vision and embedded systems have boosted the applications of computer vision to the area of control and navigation. Computer vision including 3D vision techniques have been investigated extensively up to now. However, due to the unique properties of visual measurements, many novel interesting problems emerge in vision-based control and navigation systems.

Vision inherently is a bearing-only sensing approach. Given an image and the associated intrinsic parameters of the camera, it is straightforward to compute the bearing of each pixel in the image. As a result, it is trivial for vision to obtain the bearing of a target relative to the camera once the target can be recognized in the image. It would be, however, much harder for vision to obtain the range from the target to the camera. Estimating the target range poses high requirements for both hardware and software of the vision system. First, in order to obtain the target range, geometric information of the vehicle is required, or the vehicle needs to carry a pre-designed artificial marker whose geometry is perfectly known. Second, pose estimation algorithms are required in order to estimate the target range. Range estimation will increase the computational burden significantly. The burden will be particularly high when estimating the positions of multiple targets. In summary, the bearing-only property of visual measurements plays a key role in many vision-based control and navigation tasks.

This thesis consists of three parts and four chapters. As illustrated in Figure 1.1, the topic addressed in each part is an interdisciplinary topic of computer

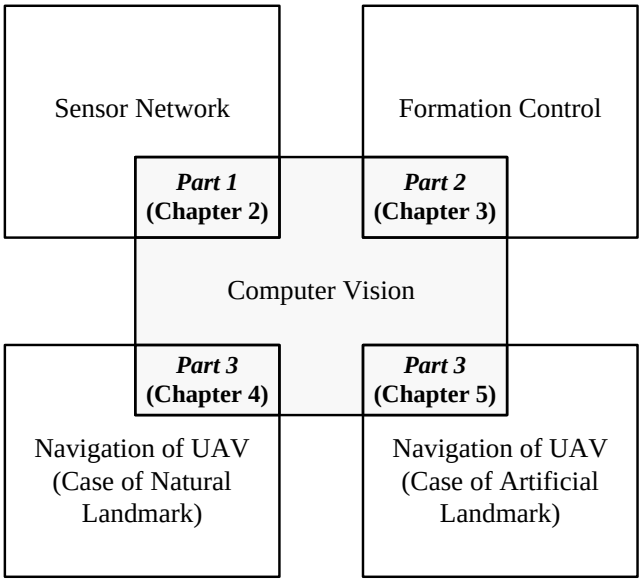


Figure 1.1: An illustration of the organization of the thesis.

vision and control/navigation. The visual measurement is the core of all the topics. Specifically, the first part (Chapter 2) addresses optimal placement of sensor networks for target localization, which is an interdisciplinary topic of sensor network and computer vision. The second part (Chapter 3) focuses on bearing-only formation control, which is an interdisciplinary topic of formation control and computer vision. The third part (Chapter 4 and Chapter 5) explores vision-based navigation of UAVs, which is an interdisciplinary topic of UAV navigation and computer vision.

1.1 Background

As aforementioned, it is easy for vision to obtain the bearing but hard to obtain the range of a target. As a result, if vision is treated as a bearing-only sensing approach, the burden on the end of vision can be significantly reduced, and consequently the reliability and efficiency of the vision system can be greatly enhanced. In fact, vision can be practically treated as a bearing-only sensor in some multi-vehicle systems.

In multi-vehicle cooperative target tracking, suppose each vehicle carries a monocular camera to measure the bearing of the target. If the multiple vehicles/cameras are deployed in a general placement, the target position can be determined cooperatively from the multiple bearing measurements. Cooperative

target localization/tracking by sensor networks is a mature research area. However, it is still an unsolved problem how to place the sensors in 3D space such that the target localization uncertainty can be minimized. When localizing a target from noisy measurements of multiple sensors, the placement of the sensors can significantly affect the estimation accuracy. In Chapter 2, we investigate the optimal sensor placement problem. One main contribution of our work is that we propose and prove the necessary and sufficient conditions for optimal sensor placement in both 2D and 3D spaces. Our research result was initially developed for bearing-only sensor networks, but later extended to range-only and received-strength-signal (RSS) sensor networks.

In cooperative target tracking, the bearing measurements are ultimately used for target position estimation. As a comparison, in multi-vehicle formation control, the bearing measurements can be directly used for formation stabilization while no position estimation is required.

It is necessary for each vehicle obtaining certain information such as positions of their neighbors in multi-vehicle formation control. The information exchange can be realized by vision. In the conventional framework for vision-based formation control, it is commonly assumed that vision is a very powerful sensor which can provide the relative positions of the neighbors. This assumption is practically unreasonable because it poses high requirements for both hardware and software of the vision system. Treating vision as a bearing-only sensing approach is a practically meaningful solution to vision-based formation control. In Chapter 3, vision-based formation control is formulated to a bearing-only formation control problem. We propose a distributed bearing-only control law to stabilize cyclic formations. It is proved that the control law can guarantee local exponential or finite-time stability.

The burden on the end of vision can be greatly reduced if vision can be treated as a bearing-only sensing approach. However, estimation of the target range cannot be always avoided in practice. We have to estimate the target range in many cases such as vision-based navigation of unmanned aerial vehicles (UAVs). My thesis will address vision-based navigation using natural and artificial landmarks, respectively.

In Chapter 4, we investigate navigation of UAVs using natural landmarks. Inertial measurement units (IMUs) are common sensors used for UAV navigation. The measurements of low-cost IMUs usually are corrupted by high noises and large biases. As a result, pure inertial navigation based on low-cost IMUs would drift rapidly. In practice, inertial navigation is usually aided by the global positioning system (GPS) to achieve drift-free navigation. However, GPS is unavailable in certain environments. In addition to GPS, vision is also a popular technique to aid inertial navigation. Chapter 4 addresses vision-aided navigation of UAVs in unknown and GPS-denied environments. We design and implement a navigation system based on a minimal sensor suite including vision to achieve drift-free attitude and velocity estimation.

Chapter 5 will present a vision-based navigation system using artificial landmarks. The navigation system can be used for cargo transporting by UAVs between moving platforms, and was successfully applied to the 2013 International UAV Innovation Grand Prix (UAVGP), held in Beijing, China, September 2013. The UAVGP competition contains several categories such as Rotor-Wing Category and Creativity Category. We next briefly introduce the tasks required by the Rotor-Wing Category that we have participated in. Two platforms moving on the ground are used to simulate two ships. Four circles are drawn on each platform. Four buckets are initially placed, respectively, inside the four circles on one platform. The weight of each bucket is about 1.5 kg. The competition task requires a UAV to transfer the four buckets one by one from one platform to the other. In addition to bucket transferring, the UAV should also perform autonomous taking off, target searching, target following and landing. The entire task must be completed by the UAV fully autonomously without any human intervention. Our team from the Unmanned Aircraft Systems (UAS) Group at National University of Singapore has successfully completed the entire task and made a great success in the competition. The great success is partially due to the vision-based navigation system presented in Chapter 5.

1.2 Literature Review

Optimal placement of sensor networks has been investigated extensively up to now. The existing studies can be characterized from the following several aspects.

In the literature, there are generally two kinds of mathematical formulations for optimal sensor placement problems. One is *optimal control* [97, 106, 96, 86] and the other is *parameter optimization* [11, 12, 13, 38, 83, 118, 37, 64, 88]. The optimal control formulations are usually adopted for cooperative path planning problems [97, 106, 96], the aim of which is to estimate the target position on one hand and plan the path of sensor platforms to minimize the estimation uncertainty on the other hand. These problems are also referred to simultaneous localization and planning (SLAP) [106]. The disadvantage of this kind of formulation is that the optimal control with various constraints generally can only be solved by numerical methods. Analytical properties usually cannot be obtained. Optimal sensor placement problems are also widely formulated as parameter optimization problems [11, 12, 13, 38, 83, 118, 37, 64, 88]. The target estimation uncertainty is usually characterized by the Fisher information matrix (FIM). In contrast to optimal control formulations, parameter optimization formulations can be solved analytically. The analytical solutions are important because they can provide valuable insights into the impact of sensor placements on target localization/tracking uncertainty. Many studies have shown that target tracking performance can be improved when sensors are steered to form an optimal placement. In our work, we only focus on determining optimal placements and will not address target tracking. One may refer to [83] for an example that illustrates the application optimal sensor placements to cooperative target tracking.

Until now, most of the existing results have been only concerned with optimal sensor placements in 2D space [11, 12, 13, 38, 83, 118, 64]. Very few studies have tackled 3D cases [88]. Analytical characterization of generic optimal sensor placements in 3D is still an open problem. Furthermore, the existing work on optimal sensor placement has addressed many sensor types such as bearing-only [11, 38, 122], range-only [83, 11, 66], RSS [13], time-of-arrival (TOA) [11, 12], time-difference-of-arrival (TDOA) [11, 64], and Doppler [14]. However, these

types of sensor networks are addressed individually in the literature. A unified framework for analyzing different types of sensor networks is still lacking.

Unlike optimal sensor placement, bearing-only formation control is still a new research topic that has not attracted much attention yet.

We next review studies related to bearing-only formation control from the following two aspects. The first aspect is what kinds of measurements are used for formation control. In conventional formation control problems, it is commonly assumed that each vehicle can obtain the *positions* of their neighbors via, for example, wireless communications. It is notable that the position information inherently consists of two kinds of partial information: *bearing* and *distance*. Formation control using bearing-only [89, 5, 10, 8, 41, 49] or distance-only measurements [21, 20] has become an active research topic in recent years. The second aspect is how the desired formation is constrained. In recent years, control of formations with inter-vehicle distance constraints has become a hot research topic [94, 74, 36, 117, 107, 63]. Recently researchers also investigated control of formations with bearing/angle constraints [5, 10, 8, 41, 49, 9]. Formations with a mix of bearing and distance constraints has also been studied by [42, 15].

From the point of view of the above two aspects, the problem studied in our work can be stated as *control of formations with angle constraints using bearing-only measurements*. This problem is a relatively new research topic. Up to now only a few special cases have been solved. The work in [89] proposed a distributed control law for balanced circular formations of unit-speed vehicles. The proposed control law can globally stabilize balanced circular formations using bearing-only measurements. The work in [5, 10, 8] studied distributed control of formations of three or four vehicles using bearing-only measurements. The global stability of the proposed formation control laws was proved by employing the Poincare-Bendixson theorem. But the Poincare-Bendixson theorem is only applicable to the scenarios involving only three or four vehicles. The work in [41] investigated formation shape control using bearing measurements. Parallel rigidity was proposed to formulate bearing-based formation control problems. A bearing-based control law was designed for a formation of three nonholonomic

vehicles. Based on the concept of parallel rigidity, the research in [49] proposed a distributed control law to stabilize bearing-constrained formations using bearing-only measurements. However, the proposed control law in [49] requires communications among the vehicles. That is different from the problem considered in our work where we assume there are no communications between any vehicles and each vehicle cannot share their bearing measurements with their neighbors. The work in [9, 15] designed control laws that can stabilize generic formations with bearing (and distance) constraints. However, the proposed control laws in [9, 15] require position instead of bearing-only measurements. In summary, although several frameworks have been proposed in [42, 41, 49, 15] to solve bearing-related formation control tasks, it is still an open problem to design a control law that can stabilize generic bearing-constrained formations using bearing-only measurements.

In cooperative target tracking or vision-based formation control, it is practically possible to treat vision as a bearing-only sensing approach. However, we have retrieve range information from visual measurements in many cases such as vision-based navigation of UAVs. Hence it is determined by the specific application whether vision can be treated as a bearing-only sensor. We next review the literature on vision-based navigation of UAVs. We first consider the case of unknown environments and the UAV is navigated based on natural landmarks. Then we consider the case of known environments where the UAV is navigated based on artificial landmarks.

The existing vision-based navigation tasks can be generally categorized to two kinds of scenarios. In the first kind of scenarios, maps or landmarks of the environments are available [120, 119, 114, 90, 59, 27]. Then the states of the UAV can be estimated without drift using image registration or pose estimation techniques. In the second kind of scenarios, maps or landmarks of the environments are not available. Visual odometry [27, 18, 67, 104] and simultaneous localization and mapping (SLAM) [69, 70, 18, 108, 17] are two popular techniques for vision-based navigation in unmapped environments. Given an image sequence taken by the onboard camera, the inter-frame motion of the UAV can be retrieved from pairs of consecutive images. Then visual odometry can

estimate the UAV states by accumulating these inter-frame motion estimates. However, the states estimated in this way will drift over time due to accumulation errors. As a comparison, SLAM not only estimates the UAV states, but also simultaneously builds up a map of the environment. Visual odometry usually discards the past vision measurements, but SLAM stores the past vision measurements in the map and consequently uses them to refine the current state estimation. Thus SLAM potentially can give better navigation accuracy than visual odometry. However, maintaining a map requires high computational and storage resources, which makes it difficult to implement real-time SLAM over onboard systems of small-scale UAVs. Moreover, SLAM is not able to completely remove drift without loop closure. But loop closure is not available in many navigation tasks in practice. Therefore, compared to SLAM, visual odometry is more efficient and suitable for navigating small-scale UAVs especially when mapping is not required. In this work we will adopt a visual odometry scheme to build up a real-time vision-based navigation system.

The particular vision technique used in our navigation system is *homography*, which has been successfully applied to a variety of UAV navigation tasks [27, 18, 67, 90, 59, 124, 123]. We recommend [82, Section 5.3] for a good introduction to homography. Suppose the UAV is equipped with a downward-looking monocular camera, which can capture images of the ground scene during flight. When the ground is planar, a 3 by 3 homography matrix can be computed from the feature matchings of two consecutive images. A homography matrix carries certain useful motion information of the UAV. The conventional way to retrieve the information is to decompose the homography matrix [18, 67]. However, homography decomposition has several disadvantages. For example, the decomposition gives two physically possible solutions. Other information is required to disambiguate the correct solution. More importantly, the homography estimated from two images certainly has estimation errors. These errors would propagate through the decomposition procedure and may cause large errors in the finally decomposed quantities. To avoid homography decomposition, the work in [27, 59] uses IMU measurements to eliminate the rotation in the homography and then retrieves the translational information only. Note drift-free attitude

estimation is not an issue in [27, 59]. But in our work the attitude (specifically the pitch and roll angles) of the UAV cannot be directly measured by any sensors. Thus we have to fully utilize the information carried by a homography to tackle the drift-free attitude estimation problem. It is notable that the homography carries the information of the pitch and roll angles if the ground plane is horizontal. For indoor environments, the floor surfaces normally are horizontally planar; for outdoor environments, the ground can be treated as a horizontal plane when the UAV flies at a relatively high altitude. By assuming the ground as a horizontal plane, we will show homography plays a key role in drift-free attitude and velocity estimation. Other vision-based methods such as horizontal detection [32] can also estimate attitude (roll and pitch angles) but the velocity cannot be estimated simultaneously.

In our work on vision-based navigation using artificial landmarks, we use circles with known diameters as the artificial landmarks. In order to accomplish the navigation task using circles, we need to solve the three key problems: ellipse detection, ellipse tracking, and circle-based pose estimation.

Ellipse detection has been investigated extensively up to now [47, 1, 4, 84, 121]. We choose ellipse fitting [47, 1] as the core of our ellipse detection algorithm. That is mainly because ellipse fitting is very efficient compared to, for example, Hough transform based ellipse detection algorithms [4, 84]. Our work adopts the well-implemented algorithm, the OpenCV function *fitEllipse*, for ellipse fitting. Since a contour cannot be determined as an ellipse or not merely by ellipse fitting, we present a three-step procedure to robustly detect ellipses. The procedure consists of 1) pre-processing, 2) ellipse fitting and 3) post-processing. The pre-processing is based on affine moment invariants (AMIs) [48]; the post-processing is based on the algebraic error between the contour and the fitted ellipse. The three-step procedure is not only robust against non-elliptical contours, but also can detect partially occluded ellipses.

In practical applications, multiple ellipses may be detected in an image, but we may be only interested in one of them. After certain initialization procedure, the ellipse of interest needs to be tracked over the image sequence such that the pose of the corresponding circle can be estimated continuously. There are several

practical challenges for tracking an ellipse in the competition task. Firstly, the areas enclosed by the ellipses are similar to each other in both color and shape. As a result, pattern matching methods based only on color, shape or feature points are not able to distinguish the target ellipse. Secondly, in order to track the target ellipse consistently, the frame rate of the image sequence must be high. This requires the tracking algorithm to be sufficiently efficient. Considering these challenges, we choose the efficient image tracking method CAMShift [2] as the core of our tracking algorithm. The proposed algorithm can robustly track the target ellipse even when its scale, shape or even color is dynamically varying.

The application of circles in camera calibration and pose estimation has been investigated extensively [57, 71, 65, 110, 40, 76]. However, the existing work mainly focused on the cases of concentric circles [71, 65, 76, 40], while the aim of our work is to do pose estimation based only on *one single circle*. The topic addressed in [110] is similar to ours, but it is concluded in [110] that other information such as parallel lines are required to estimate the pose of a single circle. From a practical point of view, we can successfully solve the single-circle-based pose estimation problem in our work by adopting a reasonable assumption. Based on that assumption, we propose an accurate and efficient algorithm that can estimate the position of the circle center from a single circle. The necessary and sufficient conditions for the adopted assumption are also proved.

1.3 Contributions of the Thesis

We next summarize the contributions of each chapter.

Chapter 2 studies optimal placement of sensor networks for target localization and tracking. We present a unified framework to analyze optimal placements of bearing-only, range-only, and RSS sensor networks. We prove the necessary and sufficient conditions for optimal placements in 2D and 3D spaces. It is shown that there are two kinds of optimal sensor placements: regular and irregular. An irregular optimal placement problem can be converted to a regular one in a lower dimensional space. A number of important analytical properties of optimal sensor placements are explored. We propose a gradient control law that not only

verifies our analytical analysis, but also provides a convenient numerical method to construct optimal placements. Since the existing results in the literature are mainly applicable to 2D cases, our work for both 2D and 3D cases is a significant generalization of the existing studies.

Chapter 3 addresses bearing-only formation control, a new research topic that has not attracted much attention yet. Bearing-only formation control provides a novel and practical solution for implementing vision-based formation control tasks. We investigate an important special case: cyclic formations with underlying graphs as cycles. We design a distributed control law which merely requires local bearing measurements. It is proved that the control law guarantees local exponential or finite-time formation stability. Collision avoidance between any vehicles can also be locally guaranteed. The stability analysis based on Lyapunov approaches should be useful for future research on more complicated bearing-based formation control problems.

Chapter 4 investigates vision-based navigation of UAVs using natural landmarks. Specifically, we propose a novel homography-based vision-aided inertial navigation system to provide drift-free velocity and attitude estimates. The observability analysis of the proposed navigation system suggests that the velocity, attitude and unknown biases are all observable as expected when the UAV speed is nonzero. Comprehensive simulations and flight experiments verify the effectiveness and robustness of the proposed navigation system.

Chapter 5 studies a vision-based navigation task for UAVs using artificial landmarks. Specifically, we propose reliable and efficient vision algorithms for ellipse detection, ellipse tracking, and circle-based pose estimation. A series of experiments and the great success of our team in UAVGP verify the efficiency, accuracy, and reliability of the proposed vision system. In addition to the specific tasks proposed by UAVGP, the proposed algorithms are also applied to a wide range of vision-based navigation and guidance tasks such as vision-based autonomous takeoff and landing, target following and vision-based formation control of UAVs.

Chapter 2

Optimal Placement of Sensor Networks for Target Tracking

2.1 Introduction

In this chapter, we will study optimal placement of sensor networks for target tracking. Although this topic has been investigated extensively, several key problems are still unsolved as discussed in Section 1.2. In our work, we adopt the parameter optimization formulation to analyze optimal placements. Our aim is to analytically determine the optimal sensor-target geometry based on an initial estimate of the target position. In practice, the initial estimate can be obtained by using, for example, Kalman filter. The optimal placement deployed based on the initial estimate is supposed to be able to improve the consequent target localization/tracking accuracy. It should be noted that we will not discuss target estimation or practical applications of optimal sensor placements in our work. Interested readers may refer to [83, Section 4] for a comprehensive example that illustrates how optimal sensor placements can be applied to cooperative target tracking.

The main contributions of our research are summarized as below.

1) We generalize the existing results in [11, 38, 83, 13] from 2D to 3D. The generalization is non-trivial. Maximizing the determinant of the FIM has been widely adopted as the criterion for optimal placements in 2D. This

criterion can be interpreted as maximizing the target information gathered by the sensors. However, this criterion cannot be directly applied to 3D cases because the determinant of the FIM is hardly analytically tractable in 3D cases. Motivated by that, we propose a *new criterion* for optimal sensor placement. This new criterion plays a key role in the generalization of the existing results from 2D to 3D.

2) In our work, we consider three types of sensor networks: bearing-only, range-only, and RSS-based. Optimal placements of these senor networks have been analyzed *individually* in the literature. We present a *unified* framework for analyzing optimal placement of these sensor networks. The results presented in this chapter are applicable to the three types of sensor networks.

3) Based on recently developed frame theory, we prove the necessary and sufficient conditions of optimal placement of sensor networks in 2D and 3D spaces. This is the most important result of our research.

4) A number of important properties of optimal sensor placements are explored. We also present a centralized gradient control law that can construct 2D and 3D optimal sensor placements numerically.

The chapter is organized as follows. Section 2.2 introduces preliminaries to frame theory. Section 2.3 presents a unified mathematical formulation for optimal placement problems of bearing-only, range-only, and RSS sensors in 2D and 3D. In Section 2.4, we present necessary and sufficient conditions for optimal placements. Section 2.5 further explores a number of important properties of optimal placements. Section 2.6 proposes a gradient control law that can be used to automatically deploy optimal sensor placements.

2.2 Preliminaries to Frame Theory

Frames can be defined in any Hilbert space. Here we are only interested in d-dimensional Euclidean space $\mathbb{R}^d$ with $d \geq 2$. Let $\|\cdot\|$ be the Euclidean norm of a vector or the Frobenius norm of a matrix. As shown by [6, 23, 72, 73], a set of vectors $\{\varphi_i\}_{i=1}^n$ in $\mathbb{R}^d$ $(n \geq d)$ is called a frame if there exist constants

$0 < a \leq b < +\infty$ so that for all $x \in \mathbb{R}^d$

$$a\|x\|^2 \leq \sum_{i=1}^{n} \langle x, \varphi_i \rangle^2 \leq b\|x\|^2, \tag{2.1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors. The constants a and b are called the *frame bounds*. A frame $\{\varphi_i\}_{i=1}^n$ is called *unit-norm* if $\|\varphi_i\| = 1$ for all $i \in \{1, \ldots, n\}$. Denote $\Phi = [\varphi_1, \ldots, \varphi_n] \in \mathbb{R}^{d \times n}$. Because $\langle x, \varphi_i \rangle^2 = (x^{\mathrm{T}} \varphi_i)^2 = x^{\mathrm{T}} \varphi_i \varphi_i^{\mathrm{T}} x$, inequality (2.1) can be rewritten as

$$a\|x\|^2 \leq x^{\mathrm{T}} \Phi \Phi^{\mathrm{T}} x \leq b\|x\|^2,$$

where the matrix $\Phi\Phi^{\mathrm{T}} = \sum_{i=1}^n \varphi_i \varphi_i^{\mathrm{T}}$ is called the *frame operator*. The frame bounds a and b obviously are the smallest and largest eigenvalues of $\Phi\Phi^{\mathrm{T}}$, respectively. Since $a > 0$, $\Phi\Phi^{\mathrm{T}}$ is positive definite and hence Φ is of full row rank. Therefore, the frame $\{\varphi_i\}_{i=1}^n$ spans $\mathbb{R}^d$. It is well known that d vectors in $\mathbb{R}^d$ form a basis if they span $\mathbb{R}^d$. Frame essentially is a generalization of the concept of basis. Unlike a basis, a frame have $n - d$ redundant vectors. The constant n/d is referred as the *redundancy* of the system. When $n/d = 1$, a frame would degenerate to a basis.

Tight frame is a particularly important concept in frame theory. A frame is *tight* when $a = b$. From (2.1) it is easy to see the frame $\{\varphi_i\}_{i=1}^n$ is tight when

$$\sum_{i=1}^{n} \varphi_i \varphi_i^{\mathrm{T}} = aI_d. \tag{2.2}$$

Taking trace on both sides of (2.2) yields $a = \sum_{i=1}^n \|\varphi_i\|^2 / d$. It is an important and fundamental problem in frame theory to construct a tight frame $\{\varphi_i\}_{i=1}^n$ that solves (2.2) with specified norms. This problem is also recognized as notoriously difficult [24]. One approach to this problem is to characterize tight frames as the minimizers of the *frame potential*

$$\mathrm{FP}\left(\{\varphi_i\}_{i=1}^n\right) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\varphi_i^{\mathrm{T}} \varphi_j\right)^2. \tag{2.3}$$

Frame potential was first proposed by [6] for unit-norm frames, and then gener-

alized by [23] for frames with arbitrary norms.

We can find tight frames by minimizing the frame potential. The following concept of *irregularity* is crucial for characterizing the minimizers of the frame potential [23, 72].

Definition 2.1 (**Irregularity**). *For any positive non-increasing sequence* $\{c_i\}_{i=1}^n$ *with* $c_1 \geq \cdots \geq c_n > 0$*, and any integer* d *satisfying* $1 \leq d \leq n$*, denote* k_0 *as the smallest nonnegative integer* k *for which*

$$c_{k+1}^2 \leq \frac{1}{d-k} \sum_{i=k+1}^{n} c_i^2. \tag{2.4}$$

The integer k_0 *is called the irregularity of* $\{c_i\}_{i=1}^n$ *with respect to* d*.*

Remark 2.1. *The irregularity of a sequence is evaluated with respect to a particular positive integer* d*. The irregularity of a given sequence may be different when evaluated with respect to different positive integers. In this chapter, we will omit mentioning this integer when the context is clear.*

Because the index $k = d - 1$ always makes (2.4) hold, the irregularity k_0 always exists and satisfies

$$0 \leq k_0 \leq d - 1.$$

When $k_0 = 0$, inequality (2.4) degenerates to the *fundamental inequality* [23]

$$\max_{j=1,\ldots,n} c_j^2 \leq \frac{1}{d} \sum_{i=1}^{n} c_i^2. \tag{2.5}$$

In this chapter we call the sequence $\{c_i\}_{i=1}^n$ *regular* when $k_0 = 0$, and *irregular* when $k_0 \neq 0$. The fundamental inequality (2.5) intuitively implies: a sequence is regular when no element is much larger than the others. Next we show several examples to illustrate the concept of irregularity.

Example 2.1. *Consider a sequence* $\{c_i\}_{i=1}^n$ *with* $c_1 = \cdots = c_n = c$ *and any* $d \leq n$*. The fundamental inequality* (2.5) *holds because* $1/d \sum_{i=1}^n c_i^2 = nc^2/d \geq c^2$*. Thus* $\{c_i\}_{i=1}^n$ *is regular with respect to any integer* $d \leq n$*. This result will be frequently used in the sequel.*

Example 2.2. *Consider a sequence* $\{c_i\}_{i=1}^4 = \{10, 1, 1, 1\}$ *and* $d = 3$. *Note the feature of this sequence is that one element is much larger than the others. Because* $10^2 > 1/3(10^2 + 1 + 1 + 1)$, *the sequence is irregular with respect to* $d = 3$. *In order to determine the irregularity* k_0, *we need to further check if* $\{c_i\}_{i=2}^4 = \{1, 1, 1\}$ *is regular with respect to* $d - 1 = 2$. *Since the elements of* $\{c_i\}_{i=2}^4$ *equal to each other,* $\{c_i\}_{i=2}^4$ *is regular with respect to* 2 *as shown in Example 2.1. Hence the irregularity of* $\{c_i\}_{i=1}^4$ *with respect to* $d = 3$ *is* $k_0 = 1$. *This example illustrates one important result: a sequence is irregular if certain element is much larger than the others.*

Example 2.3. *Consider a sequence* $\{c_i\}_{i=1}^4 = \{10, 10, 1, 1\}$ *and* $d = 2$ *or* 3. *When* $d = 2$, *we have* $10^2 < 1/2(10^2 + 10^2 + 1 + 1)$. *Hence* $\{c_i\}_{i=1}^4$ *is regular with respect to* $d = 2$. *When* $d = 3$, *we have* $10^2 > 1/3(10^2 + 10^2 + 1 + 1)$, $10^2 > 1/2(10^2 + 1 + 1)$ *and* $1 < 1/1(1 + 1)$. *Hence* $\{c_i\}_{i=1}^4$ *is irregular with respect to* $d = 3$ *and the irregularity is* $k_0 = 2$. *This example shows that a sequence may be regular with respect to one integer but irregular with respect to another.*

The minimizers of the frame potential in (2.3) are characterized by the following lemma [23], which will be used to prove the necessary and sufficient conditions of optimal placements.

Lemma 2.1. *In* $\mathbb{R}^d$, *given a positive non-increasing sequence* $\{c_i\}_{i=1}^n$ *with irregularity as* k_0, *if the norms of the frame* $\{\varphi_i\}_{i=1}^n$ *are specified as* $\|\varphi_i\| = c_i$ *for all* $i \in \{1, \ldots, n\}$, *any minimizer of the frame potential in* (2.3) *is of the form*

$$\{\varphi_i\}_{i=1}^n = \{\varphi_i\}_{i=1}^{k_0} \cup \{\varphi_i\}_{i=k_0+1}^n,$$

where $\{\varphi_i\}_{i=1}^{k_0}$ *is an orthogonal set, and* $\{\varphi_i\}_{i=k_0+1}^n$ *is a tight frame in the orthogonal complement of the span of* $\{\varphi_i\}_{i=1}^{k_0}$. *Any local minimizer is also a global minimizer.*

From Lemma 2.1, a minimizer of the frame potential consists of an orthogonal set $\{\varphi_i\}_{i=1}^{k_0}$ and a tight frame $\{\varphi_i\}_{i=k_0+1}^n$. The partition of the two sets is determined by the irregularity of the specified frame norms $\{c_i\}_{i=1}^n$. When the

irregularity $k_0 = 0$, it is clear that a minimizer of the frame potential is a tight frame. As a corollary of Lemma 2.1, the following result [23] gives the existence condition of the solutions to (2.2).

Lemma 2.2. *In $\mathbb{R}^d$, given a positive sequence $\{c_i\}_{i=1}^n$, there exists a tight frame $\{\varphi_i\}_{i=1}^n$ with $\|\varphi_i\| = c_i$ for all $i \in \{1, \ldots, n\}$ solving (2.2) if and only if $\{c_i\}_{i=1}^n$ is regular.*

2.3 Problem Formulation

Consider one target and n sensors in $\mathbb{R}^d$ ($d = 2$ or 3 and $n \geq d$). The n sensors are of one of the following sensor types: bearing-only, range-only, and RSS. Sensor networks with mixed sensor types are not considered in this chapter. Following [11, 38, 83, 13], we assume that an initial target position estimate $p \in \mathbb{R}^d$ is available. The optimal placement will be determined based on this initial estimate. Denote the position of sensor i as $s_i \in \mathbb{R}^d$, $i \in \{1, \ldots, n\}$. Then $r_i = s_i - p$ denotes the position of sensor i relative to the target. The sensor-target placement can be fully described by $\{r_i\}_{i=1}^n$. Our aim is to determine the optimal $\{r_i\}_{i=1}^n$ such that certain objective function can be optimized. The distance between sensor i and the target is given by $\|r_i\|$. The unit-length vector

$$g_i = \frac{r_i}{\|r_i\|}$$

represents the bearing of sensor i relative to the target.

2.3.1 Sensor Measurement Model and FIM

For any sensor type in Table 2.1, the measurement model of sensor i can be expressed as

$$z_i = h_i(r_i) + v_i,$$

where $z_i \in \mathbb{R}^m$ denotes the measurement of sensor i, the function $h_i(r_i) : \mathbb{R}^d \to \mathbb{R}^m$ is determined by the sensor type as shown in Table 2.1, and $v_i \in \mathbb{R}^m$ is the additive measurement noise. We assume v_i to be a zero-mean Gaussian noise

Table 2.1: Measurement models and FIMs of the three sensor types.

Sensor type	Measurement model	FIM	Coefficient
Bearing-only	$h_i(r_i) = \dfrac{r_i}{\|r_i\|}$	$F = \sum_{i=1}^{n} c_i^2 (I_d - g_i g_i^{\mathrm{T}})$	$c_i = \dfrac{1}{\sigma_i \|r_i\|}$
Range-only	$h_i(r_i) = \|r_i\|$	$F = \sum_{i=1}^{n} c_i^2 g_i g_i^{\mathrm{T}}$	$c_i = \dfrac{1}{\sigma_i}$
RSS	$h_i(r_i) = \ln \|r_i\|$	$F = \sum_{i=1}^{n} c_i^2 g_i g_i^{\mathrm{T}}$	$c_i = \dfrac{1}{\sigma_i \|r_i\|}$

with covariance as $\Sigma_i = \sigma_i^2 I_m \in \mathbb{R}^{m \times m}$, where I_m denotes the $m \times m$ identity matrix. By further assuming the measurement noises of different sensors are uncorrelated, the FIM given by n sensors is expressed as

$$F = \sum_{i=1}^{n} \left(\frac{\partial h_i}{\partial p} \right)^{\mathrm{T}} \Sigma_i^{-1} \frac{\partial h_i}{\partial p}, \tag{2.6}$$

where $\partial h_i / \partial p$ denotes the Jacobian of $h_i(r_i) = h_i(s_i - p)$ with respect to p. For a detailed derivation of the FIM formula in (2.6), we refer to [11, Section 3].

The measurement models of bearing-only, range-only, and RSS sensors are given in Table 2.1. The measurement of a bearing-only sensor is conventionally modeled as one angle (azimuth) in 2D or two angles (azimuth and altitude) in 3D. The drawback of this kind of model is that the model complexity increases dramatically as the dimension increases. As a result, this conventional model is not suitable for analyzing 3D optimal placements. Note that a unit-length vector essentially characterizes a bearing and is very suitable to represent a bearing-only measurement. Thus we model the measurement of a bearing-only sensor as a *unit-length vector* pointing from the target to the sensor. As will be shown later, this new bearing-only measurement model will greatly simplify the formulation of optimal bearing-only placement problems in 2D and 3D. The measurement model of range-only sensors in Table 2.1 is the same as the one given by [11]. The measurement model of RSS sensors in Table 2.1 is a modified version of the one in [13]. Without loss of generality, we simplify the model in [13] by omitting certain additive and multiplicative constants.

By substituting $h_i(r_i)$ into (2.6), we can calculate the FIMs of the three sensor types. The calculation is straightforward and omitted here. The FIMs have

been calculated and given in Table 2.1. As will be shown later, the coefficients $\{c_i\}_{i=1}^n$ in the FIM are crucial for determining optimal placements. Following [11, 38, 13, 83], we assume the coefficient c_i to be *arbitrary but fixed*. (i) For bearing-only or RSS sensors, as $c_i = 1/(\sigma_i \|r_i\|)$, both σ_i and $\|r_i\|$ are assumed to be fixed. Otherwise, if $\|r_i\|$ is unconstrained, the placement will be optimal when $\|r_i\|$ approaches zero. To avoid this trivial solution, it is reasonable to assume $\|r_i\|$ to be fixed. (ii) For range-only sensors, as $c_i = 1/\sigma_i$, only σ_i is assumed to be fixed. Hence $\|r_i\|$ will have no influence on the optimality of the placements for range-only sensors.

To end this subsection, we would like to point out that the FIMs given in Table 2.1 are consistent with the ones given in [11, 38, 13, 83] in 2D cases. To verify that, we can substitute $g_i = [\cos\theta_i, \sin\theta_i]^{\mathrm{T}} \in \mathbb{R}^2$ into the FIMs in Table 2.1.

2.3.2 A New Criterion for Optimal Placement

The existing work on optimal senor placement has adopted various objective functions such as $\det F$, $\mathrm{tr}\, F$, and $\mathrm{tr}\, F^{-1}$. These objective functions are respectively referred as D-, T-, and A-optimality criteria in the field of optimal experimental design [100]. The most popular criterion used for optimal sensor placement is to maximize $\det F$, which can be interpreted as minimizing the volume of the uncertainty ellipsoid characterized by F^{-1}. However, this criterion is not suitable for analyzing optimal placements in 3D space because $\det F$ is hardly analytically tractable in $\mathbb{R}^3$. In order to analytically characterize optimal placements in $\mathbb{R}^2$ and $\mathbb{R}^3$, we next introduce a new criterion that is closely related to the conventional one.

Denote $\{\lambda_i\}_{i=1}^d$ as the eigenvalues of F. Let $\bar{\lambda} = 1/d \sum_{i=1}^d \lambda_i$. Since $\sum_{i=1}^d \lambda_i = \mathrm{tr}\, F$, it is easy to examine that $\bar{\lambda}$ is an invariant quantity for any F given in Table 2.1. In this chapter, we will minimize the new objective function $\|F - \bar{\lambda} I_d\|^2$, which is of strong analytical tractability. Note $\|F - \bar{\lambda} I_d\|^2 = \sum_{i=1}^d (\lambda_i - \bar{\lambda})^2$. Hence minimizing $\|F - \bar{\lambda} I_d\|^2$ actually is to minimize the diversity of the eigenvalues of F. The following result shows that the new criterion has a close connection with the conventional one.

Lemma 2.3. *For any one of the three sensor types given in Table 2.1, we have*

$$\det F \leq \bar{\lambda}^d,$$

where the equality holds if and only if

$$\|F - \bar{\lambda} I_d\|^2 = 0.$$

Proof. For any one of the three sensor types, the FIM F is symmetric positive (semi) definite. Hence λ_j is real and nonnegative. From the FIMs shown in Table 2.1, we have $\sum_{j=1}^{d} \lambda_j = \operatorname{tr} F = (d-1) \sum_{i=1}^{n} c_i^2$ for bearing-only sensors, and $\sum_{j=1}^{d} \lambda_j = \operatorname{tr} F = \sum_{i=1}^{n} c_i^2$ for range-only or RSS sensors. Note $\{c_i\}_{i=1}^{n}$ is assumed to be fixed. Hence $\sum_{j=1}^{d} \lambda_j$ is an invariant quantity. By the inequality of arithmetic and geometric means, the conventional objective function $\det F$ satisfies

$$\det F = \prod_{j=1}^{d} \lambda_j \leq \left(\frac{1}{d} \sum_{j=1}^{d} \lambda_j \right)^d = \bar{\lambda}^d,$$

where the equality holds if and only if $\lambda_j = \bar{\lambda}$ for all $j \in \{1, \ldots, d\}$, which means

$$F = \bar{\lambda} I_d \iff \|F - \bar{\lambda} I_d\|^2 = 0.$$

In short, $\det F$ is maximized to its upper bound $\bar{\lambda}^d$ if and only if $\|F - \bar{\lambda} I_d\|^2 = 0$. □

Loosely speaking, Lemma 2.3 suggests that minimizing $\|F - \bar{\lambda} I_d\|^2$ is equivalent to maximizing $\det F$. We next further examine the relationship between the new and conventional criterions case by case.

1) In $\mathbb{R}^2$, we have $\det F = 1/2((\operatorname{tr} \mathcal{F})^2 - \operatorname{tr}(F^2)) = 1/2\left(4\bar{\lambda}^2 - \|F\|^2\right)$ and $\|F - \bar{\lambda} I_2\|^2 = \operatorname{tr}(F - \bar{\lambda} I_2)^2 = \|F\|^2 - 2\bar{\lambda}^2$, which suggest

$$\|F - \bar{\lambda} I_2\|^2 = -2 \det F + 2\bar{\lambda}^2.$$

Because $2\bar{\lambda}^2$ is constant, minimizing $\|F - \bar{\lambda} I_2\|^2$ is rigorously equivalent to

maximizing $\det F$ in $\mathbb{R}^2$. As a result, our analysis based on the new criterion will be consistent with the 2D results in [11, 38, 83, 13].

2) In $\mathbb{R}^3$, if $\|F - \bar{\lambda} I_3\|^2$ is able to achieve zero, then $\det F$ can be maximized to its upper bound as shown in Lemma 2.3. In this case the new criterion is still rigorously equivalent to the conventional one.

3) In $\mathbb{R}^3$, $\|F - \bar{\lambda} I_3\|^2$ is not able to reach zero in certain *irregular* cases (see Section 2.4 for the formal definition of irregular). In these cases $\det F$ and $\|F - \bar{\lambda} I_3\|^2$ may not be optimized simultaneously. But as will be shown later, the analysis of irregular cases in $\mathbb{R}^3$ based on the new criterion is a reasonable extension of the analysis of irregular cases in $\mathbb{R}^2$.

2.3.3 Problem Statement

We now formally state the optimal sensor placement problem that we are going to solve.

Problem 2.1. *Consider one target and n sensors in $\mathbb{R}^d$ ($d = 2$ or 3 and $n \geq d$). The sensors involve only one of the three sensor types in Table 2.1. Given arbitrary but fixed positive coefficients $\{c_i\}_{i=1}^n$, find the optimal placement $\{g_i^*\}_{i=1}^n$ such that*

$$\{g_i^*\}_{i=1}^n = \underset{\{g_i\}_{i=1}^n \subset \mathbb{S}^{d-1}}{\arg\min} \|F - \bar{\lambda} I_d\|^2, \tag{2.7}$$

where $\mathbb{S}^{d-1}$ denotes the unit sphere in $\mathbb{R}^d$.

Remark 2.2. *The sensor-target placement can be fully described by $\{r_i\}_{i=1}^n$. Recall $\|r_i\|$ is assumed to be fixed for bearing-only or RSS sensors, and $\|r_i\|$ has no effect on the placement optimality for range-only sensors. Thus for any sensor type, the optimal sensor placement can also be fully described by $\{g_i\}_{i=1}^n$. That means we only need to determine the optimal sensor-target bearings $\{g_i^*\}_{i=1}^n$ to obtain the optimal placement.*

Although the FIMs of different sensor types may have different formulas as shown in Table 2.1, the following result shows that substituting the FIMs of the

three sensor types into (2.7) will lead to an identical objective function. The following result is important because it enables us to *unify* the formulations of optimal sensor placement for the three sensor types.

Lemma 2.4. *Consider one target and n sensors in $\mathbb{R}^d$ ($d = 2$ or 3 and $n \geq d$). The sensors involve only one of the three sensor types in Table 2.1. The problem defined in* (2.7) *is equivalent to*

$$\{g_i^*\}_{i=1}^n = \underset{\{g_i\}_{i=1}^n \subset \mathbb{S}^{d-1}}{\arg\min} \|G\|^2, \tag{2.8}$$

where $G = \sum_{i=1}^n c_i^2 g_i g_i^{\mathrm{T}}$.

Proof. If all sensors are bearing-only, the FIM is $F = \sum_{i=1}^n c_i^2 (I_d - g_i g_i^{\mathrm{T}})$ and then $\bar{\lambda} = 1/d \sum_{j=1}^d \lambda_j = \operatorname{tr} F/d = (d-1)/d \sum_{i=1}^n c_i^2$. Hence

$$\begin{aligned} \|F - \bar{\lambda} I_d\| &= \left\| \sum_{i=1}^n c_i^2 (I_d - g_i g_i^{\mathrm{T}}) - \frac{d-1}{d} \sum_{i=1}^n c_i^2 I_d \right\| \\ &= \left\| -\sum_{i=1}^n c_i^2 g_i g_i^{\mathrm{T}} + \frac{1}{d} \sum_{i=1}^n c_i^2 I_d \right\|. \end{aligned}$$

If all sensors are range-only or RSS, the FIM is $F = \sum_{i=1}^n c_i^2 g_i g_i^{\mathrm{T}}$ and then $\bar{\lambda} = 1/d \sum_{j=1}^d \lambda_j = \operatorname{tr} F/d = 1/d \sum_{i=1}^n c_i^2$. Hence

$$\|F - \bar{\lambda} I_d\| = \left\| \sum_{i=1}^n c_i^2 g_i g_i^{\mathrm{T}} - \frac{1}{d} \sum_{i=1}^n c_i^2 I_d \right\|.$$

Therefore, for any sensor type in Table 2.1, the new objective function can be rewritten as

$$\begin{aligned} \|F - \bar{\lambda} I_d\|^2 &= \left\| \sum_{i=1}^n c_i^2 g_i g_i^{\mathrm{T}} - \frac{1}{d} \sum_{i=1}^n c_i^2 I_d \right\|^2 \\ &= \|G\|^2 - \frac{1}{d} \left(\sum_{i=1}^n c_i^2 \right)^2. \end{aligned} \tag{2.9}$$

Because $1/d(\sum_{i=1}^n c_i^2)^2$ is constant, minimizing $\|G\|^2$ is equivalent to minimizing $\|F - \bar{\lambda} I_d\|^2$. □

One primary aim of this work is to solve the parameter optimization problem (2.8). It should be noted that we must clearly know the type of the sensors such

that the coefficients $\{c_i\}_{i=1}^n$ in G can be calculated correctly. Once $\{c_i\}_{i=1}^n$ are calculated, the sensor types will be transparent to us. As a consequence, the analysis of optimal sensor placement in the sequel of the chapter will apply to all the three sensor types.

Remark 2.3. *In this work, we only consider homogeneous sensor networks. But it is worthwhile noting that Lemma 2.4 actually is also valid for a heterogeneous sensor network which contains both range-only and RSS sensors. That is because the FIMs of the two sensor types have the same formula, and the total FIM would simply be the sum of the two respective FIMs of range-only and RSS sensors. As a result, the analysis in the rest of this chapter also applies to heterogeneous sensor networks that contain both range-only and RSS sensors. In the heterogeneous case, the coefficient c_i should be calculated correctly according to the type of sensor i.*

2.3.4 Equivalent Sensor Placements

Before solving (2.8), we identify a group of placements that result in the same value of $\|G\|^2$.

Proposition 2.1. *The objective function $\|G\|^2$ is invariant to the sign of g_i for all $i \in \{1, \ldots, n\}$ and any orthogonal transformations over $\{g_i\}_{i=1}^n$.*

Proof. First, $g_i g_i^{\mathrm{T}} = (-g_i)(-g_i)^{\mathrm{T}}$ for all $i \in \{1, \ldots, n\}$, hence $\|G\|^2$ is invariant to the sign of g_i. Second, let $U \in \mathbb{R}^{d\times d}$ be an orthogonal matrix satisfying $U^{\mathrm{T}}U = I_d$. Applying U to $\{g_i\}_{i=1}^n$ yields $\{g_i' = Ug_i\}_{i=1}^n$. Then we have $G' = \sum_{i=1}^n c_i^2 g_i'(g_i')^{\mathrm{T}} = \sum_{i=1}^n c_i^2 (Ug_i)(Ug_i)^{\mathrm{T}} = UGU^{\mathrm{T}}$. Since G and G' are both symmetric, we have $\|G'\|^2 = \mathrm{tr}\,(UGU^{\mathrm{T}}UGU^{\mathrm{T}}) = \mathrm{tr}\,(G^2) = \|G\|^2$. □

Geometrically speaking, changing the sign of g_i means flipping sensor i about the target, and an orthogonal transformation represents a rotation, reflection or both combined operation over all sensors. Therefore, Proposition 2.1 implies that these geometric operations cannot affect the value of $\|G\|^2$. Furthermore, it is straightforward to examine that $\det F$ is also invariant to these geometric operations. It is noticed that the invariance to the sign change of g_i was originally

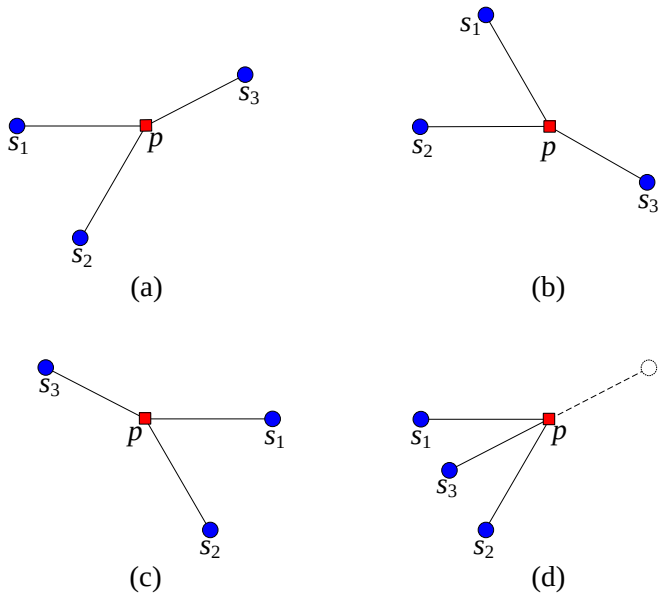


Figure 2.1: Examples of equivalent placements ($d = 2, n = 3$): (a) Original placement. (b) Rotate all sensors about the target 60 degrees clockwise. (c) Reflect all sensors about the vertical axis. (d) Flipping the sensor s_3 about the target.

recognized in [38] for 2D bearing-only sensor placements. By Proposition 2.1, we define the following equivalence relationship.

Definition 2.2 (**Equivalent placements**). *Given arbitrary but fixed coefficients* $\{c_i\}_{i=1}^n$*, two placements* $\{g_i\}_{i=1}^n$ *and* $\{g_i'\}_{i=1}^n$ *are called equivalent if they are differed by indices permutation, flipping any sensors about the target, or any global rotation, reflection or both combined over all sensors.*

Due to the equivalence, there always exist an infinite number of equivalent optimal placements minimizing $\|G\|^2$. If two optimal placements are equivalent, they lead to the same objective function value. But the converse statement is not true in general. In Section 2.5.3, we will give the condition under which the converse is true. Examples of 2D equivalent placements are given in Figure 2.1.

2.4 Necessary and Sufficient Conditions for Optimal Placement

In this section, we prove the necessary and sufficient conditions for optimal placements solving (2.8). Recall $G = \sum_{i=1}^{n} c_i^2 g_i g_i^{\mathrm{T}}$. Then we have

$$\begin{aligned} \|G\|^2 &= \sum_{i=1}^{n}\sum_{j=1}^{n}(c_i c_j g_i^{\mathrm{T}} g_j)^2 \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}(\varphi_i \varphi_j)^2, \end{aligned}$$

where $\varphi_i = c_i g_i$ and $\|\varphi_i\| = c_i$ for any $i \in \{1, \ldots, n\}$. The vectors $\{\varphi_i\}_{i=1}^{n}$ actually form a frame in $\mathbb{R}^d$. Then the objective function $\|G\|^2$ is the frame potential of $\{\varphi_i\}_{i=1}^{n}$ as shown in (2.3), and the matrix G is the frame operator. Furthermore, since $\|\varphi_i\| = c_i$, the coefficient sequence $\{c_i\}_{i=1}^{n}$ will fully determine the minimizers of $\|G\|^2$. According to the irregularity of $\{c_i\}_{i=1}^{n}$, optimal placements can be categorized as regular and irregular as shown below.

When $\{c_i\}_{i=1}^{n}$ is regular, the necessary and sufficient condition of optimal placement is given below. The 2D version of the following result has been proposed in [11, 38, 13].

Theorem 2.1 (**Regular optimal placement**). *In $\mathbb{R}^d$ with $d = 2$ or 3, if the positive coefficient sequence $\{c_i\}_{i=1}^{n}$ is regular, then the objective function $\|G\|^2$ satisfies*

$$\|G\|^2 \geq \frac{1}{d}\left(\sum_{i=1}^{n} c_i^2\right)^2. \tag{2.10}$$

The lower bound of $\|G\|^2$ is achieved if and only if

$$\sum_{i=1}^{n} c_i^2 g_i g_i^{\mathrm{T}} = \frac{1}{d}\sum_{i=1}^{n} c_i^2 I_d. \tag{2.11}$$

Proof. Let $\{\mu_j\}_{j=1}^{d}$ be the eigenvalues of G. Then $\sum_{j=1}^{d} \mu_j = \operatorname{tr} G = \sum_{i=1}^{n} c_i^2$ is

constant. Let $\bar{\mu} = 1/d \sum_{j=1}^{d} \mu_j = 1/d \sum_{i=1}^{n} c_i^2$. It is obvious that

$$\|G\|^2 = \sum_{j=1}^{d} \mu_j^2 \geq d\bar{\mu}^2 = \frac{1}{d} \left(\sum_{i=1}^{n} c_i^2 \right)^2 . \tag{2.12}$$

The lower bound of $\|G\|^2$ is achieved if and only if $\mu_j = \bar{\mu}$ for all $j \in \{1, \ldots, d\}$, which implies $G = \bar{\mu} I_d$ and hence equation (2.11). By denoting $\varphi_i = c_i g_i$, equation (2.11) becomes $\sum_{i=1}^{n} \varphi_i \varphi_i^{\mathrm{T}} = 1/d \sum_{i=1}^{n} c_i^2 I_d$ which is the same as (2.2). Thus a regular optimal placement solving (2.11) corresponds to a tight frame. Because $\{c_i\}_{i=1}^{n}$ is regular, by Lemma 2.2 there exist optimal placements solving (2.11). □

We call a placement *regular* when its coefficient sequence is regular, and *regular optimal* when it solves (2.11). To obtain a regular optimal placement, we need further to solve (2.11). Details of the solutions to (2.11) will be given in Section 2.5.1. When $\|G\|^2$ reaches its lower bound given in (2.12), it is straightforward to see $\|F - \bar{\lambda} I_d\|^2 = 0$ by (2.9). Thus by Lemma 2.3, the conventional objective function $\det F$ would also be maximized to its upper bound. Then we have the following result.

Corollary 2.1. *In $\mathbb{R}^d$ with $d = 2$ or 3, a regular optimal placement not only minimizes the new objective functions $\|G\|^2$ and $\|F - \bar{\lambda} I_d\|^2$, but also maximizes the conventional one* $\det F$.

When $\{c_i\}_{i=1}^{n}$ is irregular, (2.11) will have no solution by Lemma 2.2. Then the the necessary and sufficient condition of optimal placement is given below. The 2D version of the following result has been proposed in [11, 38, 13].

Theorem 2.2 (**Irregular optimal placement**). *In $\mathbb{R}^d$ with $d = 2$ or 3, if the positive coefficient sequence $\{c_i\}_{i=1}^{n}$ is irregular with irregularity as $k_0 \geq 1$, without loss of generality $\{c_i\}_{i=1}^{n}$ can be assumed to be a non-increasing sequence, and then the objective function $\|G\|^2$ satisfies*

$$\|G\|^2 \geq \sum_{i=1}^{k_0} c_i^4 + \frac{1}{d - k_0} \left(\sum_{i=k_0+1}^{n} c_i^2 \right)^2 . \tag{2.13}$$

The lower bound of $\|G\|^2$ is achieved if and only if

$$\{g_i\}_{i=1}^n = \{g_i\}_{i=1}^{k_0} \cup \{g_i\}_{i=k_0+1}^n, \tag{2.14}$$

where $\{g_i\}_{i=1}^{k_0}$ is an orthogonal set, and $\{g_i\}_{i=k_0+1}^n$ forms a regular optimal placement in the $(d-k_0)$-dimensional orthogonal complement of $\{g_i\}_{i=1}^{k_0}$.

Proof. Recall $\|G\|^2$ is the frame potential of the frame $\{\varphi_i\}_{i=1}^n$ where $\varphi_i = c_i g_i$. From Lemma 2.1, the minimizer of $\|G\|^2$ is of the following form: $\{c_i g_i\}_{i=1}^{k_0}$ is an orthogonal set, and $\{c_i g_i\}_{i=k_0+1}^n$ is a tight frame (i.e., a regular optimal placement) in the orthogonal complement of $\{c_i g_i\}_{i=1}^{k_0}$.

Let $\Phi_1 = [\varphi_1, \ldots, \varphi_{k_0}] \in \mathbb{R}^{d\times k_0}$, $\Phi_2 = [\varphi_{k_0+1}, \ldots, \varphi_n] \in \mathbb{R}^{d\times(n-k_0)}$, and $\Phi = [\Phi_1, \Phi_2] \in \mathbb{R}^{d\times n}$. When $\{g_i\}_{i=1}^n$ is of the form in (2.14), the columns of Φ_1 are orthogonal to those of Φ_2. Then

$$\|G\|^2 = \operatorname{tr}(\Phi^{\mathrm{T}}\Phi)^2 = \operatorname{tr}(\Phi_1^{\mathrm{T}}\Phi_1)^2 + \operatorname{tr}(\Phi_2^{\mathrm{T}}\Phi_2)^2.$$

Because $\{g_i\}_{i=1}^{k_0}$ is an orthogonal set, we have $\operatorname{tr}(\Phi_1^{\mathrm{T}}\Phi_1)^2 = \sum_{i=1}^{k_0} \|\varphi_i\|^4 = \sum_{i=1}^{k_0} c_i^4$. Because $\{g_i\}_{i=k_0+1}^n$ is a regular optimal placement in a $(d-k_0)$-dimensional subspace, we have $\operatorname{tr}(\Phi_2^{\mathrm{T}}\Phi_2)^2 = 1/(d-k_0)(\sum_{i=k_0+1}^n c_i^2)^2$ by Theorem 2.1. Therefore, when $\{g_i\}_{i=1}^n$ is of the form in (2.14), the objective function $\|G\|^2$ reaches its lower bound as shown in (2.13). □

We call a placement *irregular* when its coefficient sequence is irregular, and *irregular optimal* when it is of the form in (2.14). In Theorem 2.2, $\{g_i\}_{i=k_0+1}^n$ is a regular optimal placement in a $(d-k_0)$-dimensional space. Thus Theorem 2.2 implies that an irregular optimal placement problem would be eventually converted to a regular one in a lower dimensional subspace.

As shown by Theorems 2.2, the irregularity of $\{c_i\}_{i=1}^n$ plays a key role in determining optimal placements. Recall the irregularity k_0 of an irregular sequence with respect to d satisfies $1 \le k_0 \le d-1$. As $d = 2$ or 3 in our work, it is possible to enumerate all the kinds of irregular optimal placements. Specifically, in $\mathbb{R}^2$, we have $d = 2$ and hence $k_0 = 1$; in $\mathbb{R}^3$, we have $d = 3$ and hence $k_0 = 1$ or 2. Thus there exist only *three* kinds of irregular optimal placements in $\mathbb{R}^2$

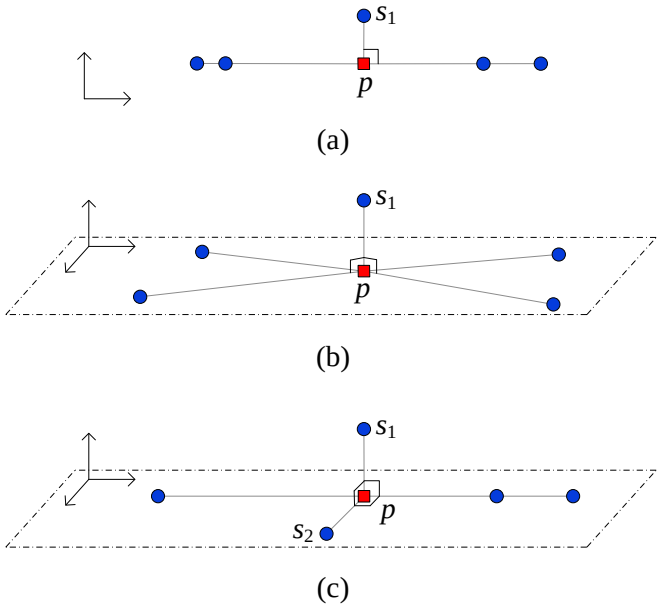


Figure 2.2: An illustration of the three kinds of irregular optimal placements in $\mathbb{R}^2$ and $\mathbb{R}^3$. (a) $d = 2$, $k_0 = 1$; (b) $d = 3$, $k_0 = 1$; (c) $d = 3$, $k_0 = 2$.

and $\mathbb{R}^3$. By Theorem 2.2, these three kinds of irregular optimal placements can be intuitively described as below.

1) Irregular optimal placement in $\mathbb{R}^2$ with irregularity $k_0 = 1$: the vector g_1 is orthogonal to $\{g_i\}_{i=2}^n$, and $\{g_i\}_{i=2}^n$ are collinear. See an illustration in Figure 2.2 (a).

2) Irregular optimal placement in $\mathbb{R}^3$ with irregularity $k_0 = 1$: the vector g_1 is orthogonal to $\{g_i\}_{i=2}^n$, and $\{g_i\}_{i=2}^n$ form a regular optimal placement in the 2D plane perpendicular to g_1. See an illustration in Figure 2.2 (b).

3) Irregular optimal placement in $\mathbb{R}^3$ with irregularity $k_0 = 2$: the vectors g_1, g_2 and $\{g_i\}_{i=3}^n$ are mutually orthogonal, and $\{g_i\}_{i=3}^n$ are collinear. See an illustration in Figure 2.2 (c).

Up to this point, Theorems 2.1 and 2.2 clearly indicate that the cruciality of the coefficients $\{c_i\}_{i=1}^n$ in determining optimal sensor placements. The coefficient c_i actually is the weight for sensor i. The larger the weight c_i is, the more sensor i contributes to the FIM. Recall $c_i = 1/\sigma_i$ for range-only sensors, and $c_i = 1/(\sigma_i \|r_i\|)$ for bearing-only or RSS sensors. Hence for range-only sensors, the measurement noise level of a sensor can affect its weight; for bearing-only

or RSS sensors, both measurement noise level and sensor-target range can affect the weight of a sensor and hence the optimal placement. In addition, a sequence $\{c_i\}_{i=1}^n$ is irregular only if certain c_i's are much larger than the others. Since large c_i implies small σ_i (and small $\|r_i\|$), a placement is irregular only if certain sensors can give much more accurate measurements (and are much closer to the target) than the others.

To make our analysis more general, we do not assume σ_i's to be identical in this work. But it is also meaningful to check the special case that $\sigma_i = \sigma_j$ for all $i \neq j$. First, for bearing-only or RSS-based sensors, the coefficient is $c_i = 1/(\sigma_i \|r_i\|)$. Then when $\sigma_i = \sigma_j$ for all $i \neq j$, from the fundamental inequality (2.5), a regular sequence $\{c_i\}_{i=1}^n$ implies

$$\max_{j=1,\ldots,n} \frac{1}{\|r_j\|^2} \leq \frac{1}{d} \sum_{i=1}^{n} \frac{1}{\|r_i\|^2}, \tag{2.15}$$

which geometrically means no sensor is much closer to the target than the others. The 2D version of inequality (2.15) has been proposed in [11, 38, 13]. Second, for range-only sensors, the coefficient is $c_i = 1/\sigma_i$. If $\sigma_i = \sigma_j$ for all $i \neq j$, then $c_i = c_j$. Hence $\{c_i\}_{i=1}^n$ is regular with respect to any $d \leq n$ as shown in Example 2.1.

We next consider an important special case $n = d$, i.e., the sensor number equals to the dimension of the space. This case is important because the optimal placement will be independent to the coefficients $\{c_i\}_{i=1}^n$ in this case. The optimal placement in the case of $n = d = 2$ has been solved by [11, 38, 13, 83].

Theorem 2.3. *In $\mathbb{R}^d$ with $d = 2$ or 3, if $n = d$, the objective function $\|G\|^2$ satisfies*

$$\|G\|^2 \geq \sum_{i=1}^{d} c_i^4.$$

The lower bound of $\|G\|^2$ is achieved if and only if $\{g_i\}_{i=1}^d$ is an orthogonal basis of $\mathbb{R}^d$.

Proof. Since $G = \sum_{i=1}^{d} c_i^2 g_i g_i^{\mathrm{T}}$ and $g_i^{\mathrm{T}} g_i = 1$ for all $i \in \{1, \ldots, n\}$, we have

$$\begin{aligned}
\|G\|^2 &= \mathrm{tr}\,(G^2) \\
&= \sum_{i=1}^{d} \sum_{j=1}^{d} c_i^2 c_j^2 (g_i^{\mathrm{T}} g_j)^2 \\
&= \sum_{i=1}^{d} \sum_{j=1, j\neq i}^{d} c_i^2 c_j^2 (g_i^{\mathrm{T}} g_j)^2 + \sum_{i=1}^{d} c_i^4 \\
&\geq \sum_{i=1}^{d} c_i^4,
\end{aligned}$$

where the equality holds if and only if $g_i^{\mathrm{T}} g_j = 0$ for all $i, j \in \{1, \ldots, d\}$ and $i \neq j$. □

Theorem 2.3 actually can be proved as a corollary of Theorems 2.1 and 2.2. But as shown above, we can also directly prove it in a straightforward way *without* employing frame theory. This can be explained from the point of view of redundancy. Recall the constant n/d reflects the redundancy of the system. When $n/d = 1$, the system has no redundancy and hence frames are no longer necessary for the optimality analysis.

2.5 Analytical Properties of Optimal Placements

In this section, we further explore a number of analytical properties of optimal placements in 2D and 3D. Theorem 2.2 implies that an irregular optimal placement problem can be eventually converted to a regular one in a lower dimensional space. Hence we will only focus on regular optimal placements.

2.5.1 Explicit Construction

A number of methods to explicitly construct some special 2D optimal placements have been proposed in [11, 38, 13, 83]. However, the construction of generic optimal placements in 2D or 3D is still an open problem. In our work, as stated in Theorem 2.1, constructing a regular optimal placement that solves (2.11) is equivalent to constructing a tight frame. Thus we successfully convert the optimal sensor placement problem to a tight frame problem. Note tight frame

construction has already been well studied in the literature on frame theory. Therefore, one can construct generic optimal placements (i.e., tight frames) of an arbitrary number of sensors in 2D or 3D by referring to the literature on tight frame construction. We will not discuss the construction of tight frames in detail here. Interested readers may refer to [44, 25, 24], to name a few.

The necessary and sufficient condition for 2D optimal placements has already been proposed in [11, 38, 13], where the sufficiency proof, however, is not given. Next we present a complete proof *without* employing frame theory. In the meantime, more importantly we propose an algorithm for explicitly constructing arbitrary 2D regular optimal placements. The following lemma can be found in [11, 38, 13, 83, 6, 51, 45].

Lemma 2.5. *In $\mathbb{R}^2$, the unit-length vector g_i can be written as $g_i = [\cos\theta_i, \sin\theta_i]^{\mathrm{T}}$. Then* (2.11) *is equivalent to*

$$\sum_{i=1}^{n} c_i^2 \bar{g}_i = 0, \tag{2.16}$$

where $\bar{g}_i = [\cos 2\theta_i, \sin 2\theta_i]^{\mathrm{T}}$.

Proof. Substituting $g_i = [\cos\theta_i, \sin\theta_i]^{\mathrm{T}}$ into (2.11) gives

$$\sum_{i=1}^{n} c_i^2 \begin{bmatrix} \frac{1}{2}\cos 2\theta_i & \frac{1}{2}\sin 2\theta_i \\ \frac{1}{2}\sin 2\theta_i & -\frac{1}{2}\cos 2\theta_i \end{bmatrix} = 0,$$

which is equivalent to (2.16). □

By Lemma 2.5, the matrix equation (2.11) is simplified to a vector equation (2.16). In order to construct $\{g_i\}_{i=1}^n$ solving (2.11), we can first construct $\{\bar{g}_i\}_{i=1}^n$ solving (2.16). Once $\bar{g}_i = [\cos 2\theta_i, \sin 2\theta_i]^{\mathrm{T}}$ is obtained, g_i can be retrieved as $g_i = \pm[\cos\theta_i, \sin\theta_i]^{\mathrm{T}}$. Note the sign changes of g_i will give equivalent optimal placements as stated in Proposition 2.1.

Theorem 2.4. *In $\mathbb{R}^2$, given a positive sequence $\{c_i\}_{i=1}^n$, there exists $\{\bar{g}_i\}_{i=1}^n$ with*

$\|\bar{g}_i\| = 1$ *solving* (2.16) *if and only if*

$$\max_{j=1,\ldots,n} c_j^2 \leq \frac{1}{2}\sum_{i=1}^{n} c_i^2. \tag{2.17}$$

Proof. Necessity: If $\sum_{i=1}^{n} c_i^2 \bar{g}_i = 0$, then $c_j^2 \bar{g}_j = \sum_{i\neq j} c_i^2 \bar{g}_i$ for all $j \in \{1, \ldots, n\}$. Hence $c_j^2 = \|c_j^2 \bar{g}_j\| = \|\sum_{i\neq j} c_i^2 \bar{g}_i\| \leq \sum_{i\neq j} \|c_i^2 \bar{g}_i\| = \sum_{i\neq j} c_i^2$. Then adding c_j^2 on both sides of the inequality gives $2c_j^2 \leq \sum_{i=1}^{n} c_i^2$.

Sufficiency: If $c_j^2 \leq 1/2 \sum_{i=1}^{n} c_i^2$ for all $j \in \{1, \ldots, n\}$, it is obvious that there always exists an index n_0 $(2 \leq n_0 \leq n)$ such that

$$c_1^2 + \cdots + c_{n_0-1}^2 \leq \frac{1}{2}\sum_{i=1}^{n} c_i^2, \tag{2.18}$$

$$c_1^2 + \cdots + c_{n_0-1}^2 + c_{n_0}^2 \geq \frac{1}{2}\sum_{i=1}^{n} c_i^2. \tag{2.19}$$

When $n_0 < n$, denote

$$\begin{aligned} \ell_1 &= c_1^2 + \cdots + c_{n_0-1}^2, \\ \ell_2 &= c_{n_0}^2, \\ \ell_3 &= c_{n_0+1}^2 + \cdots + c_n^2. \end{aligned} \tag{2.20}$$

Obviously $\ell_1 + \ell_2 + \ell_3 = \sum_{i=1}^{n} c_i^2$. From (2.17), $c_{n_0} \leq 1/2 \sum_{i=1}^{n} c_i^2$ and hence $\ell_1 + \ell_3 \geq \ell_2$. From (2.18), $\ell_1 \leq 1/2 \sum_{i=1}^{n} c_i^2$ and hence $\ell_2 + \ell_3 \geq \ell_1$. From (2.19), $\ell_1 + \ell_2 \geq 1/2 \sum_{i=1}^{n} c_i^2$ and hence $\ell_1 + \ell_2 \geq \ell_3$. Therefore, ℓ_1, ℓ_2 and ℓ_3 satisfy the triangle inequality and can form a triangle. Choose $\bar{g}_1 = \cdots = \bar{g}_{n_0-1}$. Then $\sum_{i=1}^{n_0-1} c_i^2 \bar{g}_i = \ell_1 \bar{g}_1$. Choose $\bar{g}_{n_0+1} = \cdots = \bar{g}_n$. Then $\sum_{i=n_0+1}^{n} c_i^2 \bar{g}_i = \ell_3 \bar{g}_n$. Then (2.16) becomes

$$\ell_1 \bar{g}_1 + \ell_2 \bar{g}_{n_0} + \ell_3 \bar{g}_n = 0. \tag{2.21}$$

We can choose $\bar{g}_1, \bar{g}_{n_0}$ and $\bar{g}_n$ that align with the three sides of the triangle with side length as ℓ_1, ℓ_2 and ℓ_3, respectively (see Figure 2.3). Then (2.21) and consequently (2.16) can be solved. When $n_0 = n$, the above proof is still valid. In this case, we have $\ell_3 = 0$ and $\ell_1 = \ell_2$, and (2.21) becomes $\bar{g}_1 + \bar{g}_{n_0} = 0$. □

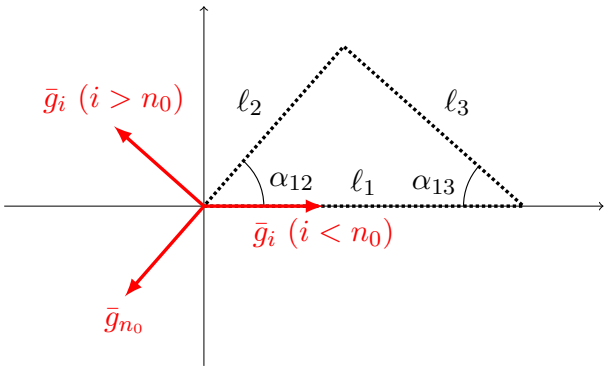


Figure 2.3: A geometric illustration of Algorithm 2.1.

Algorithm 2.1 Construction of 2D regular optimal placements $\{g_i\}_{i=1}^n$ with coefficients $\{c_i\}_{i=1}^n$.

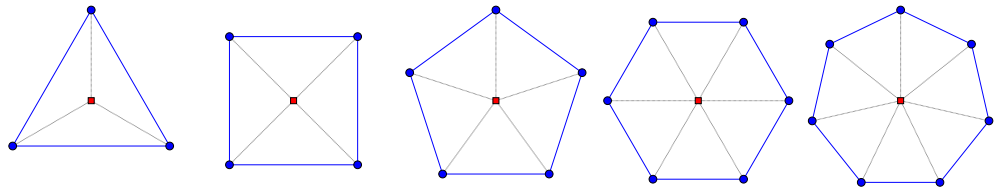
1: Choose n_0 satisfying (2.18) and (2.19). Then compute ℓ_1, ℓ_2 and ℓ_3 in (2.20).
2: Compute interior angles α_{12} and α_{13} of the triangle with side lengths as ℓ_1, ℓ_2 and ℓ_3 (See Figure 2.3).
3: Choose $g_i = [1,0]^{\mathrm{T}}$ for $i \in \{1,\dots,n_0-1\}$, $g_{n_0} = [\cos((\pi+\alpha_{12})/2), \sin((\pi+\alpha_{12})/2)]^{\mathrm{T}}$, and $g_i = [\cos((\pi-\alpha_{13})/2), \sin((\pi-\alpha_{13})/2)]^{\mathrm{T}}$ for $i \in \{n_0+1,\dots,n\}$.

From the proof of Theorem 2.4, a method for explicitly constructing 2D regular optimal placements can be summarized in Algorithm 2.1. The following example illustrates Algorithm 2.1.

Example 2.4. *In $\mathbb{R}^2$, consider six bearing-only sensors with sensor-target ranges respectively as $\|r_1\| = 5$, $\|r_2\| = 6$, $\|r_3\| = 7$, $\|r_4\| = 8$, $\|r_5\| = 9$, and $\|r_6\| = 10$. The measurement noise variance is $\sigma_i = 1$ for all $i \in \{1,\dots,6\}$. Recall $c_i = 1/(\sigma_i\|r_i\|)$ for bearing-only sensors. Then $c_1^2 = 0.0400$, $c_2^2 = 0.0278$, $c_3^2 = 0.0204$, $c_4^2 = 0.0156$, $c_5^2 = 0.0123$, $c_6^2 = 0.0100$, and $1/2\sum_{i=1}^6 c_i^2 = 0.0631$. It is easy to check the sequence $\{c_i\}_{i=1}^6$ is regular. Because $c_1^2 < 1/2\sum_{i=1}^6 c_i^2$ and $c_1^2 + c_2^2 > 1/2\sum_{i=1}^6 c_i^2$, choose $n_0 = 2$. Hence $\ell_1 = 0.0400$, $\ell_2 = 0.0278$, and $\ell_3 = 0.0584$. Then $\alpha_{12} = 2.0560\,\mathrm{rad}$ and $\alpha_{13} = 0.4344\,\mathrm{rad}$. As instructed in Algorithm 2.1, choose $g_1 = [1,0]^{\mathrm{T}}, g_2 = [0.8563, -0.5165]^{\mathrm{T}}, g_3 = \cdots = g_6 = [0.2155, 0.9765]^{\mathrm{T}}$. Then it can be verified that $\sum_{i=1}^6 c_i^2 g_i g_i^{\mathrm{T}} = 1/2\sum_{i=1}^6 c_i^2 I_2$.*

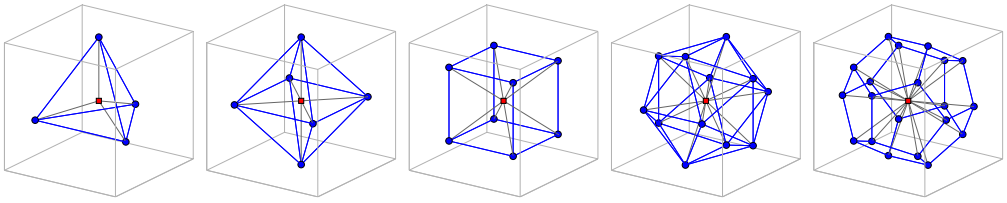
2.5.2 Equally-weighted Optimal Placements

The coefficient c_i actually is the weight for sensor i. Hence we call a placement *equally-weighted* if $c_1 = \cdots = c_n$. In the equally-weighted case, all sensors



(a) (b) (c) (d) (e)

Figure 2.4: Examples of 2D equally-weighted optimal placements: regular polygons. Red square: target; blue dots: sensors.



(a) (b) (c) (d) (e)

Figure 2.5: Examples of 3D equally-weighted optimal placements: Platonic solids. Red square: target; blue dots: sensors. (a) Tetrahedron, $n = 4$. (b) Octahedron, $n = 6$. (c) Hexahedron, $n = 8$. (d) Icosahedron, $n = 12$. (e) Dodecahedron, $n = 20$.

play equal roles for target localization. For bearing-only or RSS sensors, the placement is equally-weighted when $\sigma_i = \sigma_j$ and $\|r_i\| = \|r_j\|$ for all $i \neq j$ as $c_i = 1/(\sigma_i \|r_i\|)$. The corresponding geometry is that all sensors are restricted on a 2D circle or a 3D sphere centered at the target. For range-only sensors, the placement is equally-weighted when $\sigma_i = \sigma_j$ for all $i \neq j$ as $c_i = 1/\sigma_i$. As shown in Example 2.1, $\{c_i\}_{i=1}^n$ is regular with respect to any $d \leq n$ if $c_1 = \cdots = c_n$. Hence equally-weighted placements must be regular.

Equally-weighted placements are important because they often arise in practice and have some important special properties. In the equally-weighted case, (2.11) is simplified to $\sum_{i=1}^n g_i g_i^{\mathrm{T}} = n/dI_d$, which implies that an equally-weighted optimal placement is essentially a unit-norm tight frame [6, 45]. In $\mathbb{R}^2$, an equally-weighted placement is optimal if n ($n \geq 3$) sensors are located at the vertices of an n-side regular polygon [6, 45, 11, 38, 13, 83] as shown in Figure 2.4. In $\mathbb{R}^3$, an equally-weighted placement is optimal if n sensors are located at the vertices of a Platonic solid [6, 45]. There are only five Platonic solids as shown in Figure 2.5. It should be noted that equally-weighted optimal placements are not limited to regular polygons or Platonic solids. In Section 2.5.4 we will show more examples of equally-weighted optimal placements.

2.5.3 Uniqueness

Due to placement equivalence, there exist at least an infinite number of equivalent optimal placements minimizing $\|G\|^2$. It is interesting to ask whether all optimal placements that minimize $\|G\|^2$ are mutually equivalent, or in other words, whether the optimal placement is *unique* up to the equivalence. We next give the conditions under which the answer is positive.

According to Theorem 2.3, it is clear that the optimal placement is unique in the case of $n = d$. We next show the regular optimal placement is also unique in the case of $n = d + 1$ (i.e., three sensors in $\mathbb{R}^2$ or four sensors in $\mathbb{R}^3$). The uniqueness will be proved by construction, which is inspired by the work in [51] on unit-norm tight frames.

Theorem 2.5. *In $\mathbb{R}^d$ with $d = 2$ or 3, if $n = d + 1$, given a regular coefficient sequence $\{c_i\}_{i=1}^{d+1}$, the regular optimal placement $\{g_i\}_{i=1}^{d+1}$ is unique up to the equivalence in Definition 2.2.*

Proof. Suppose $\{g_i\}_{i=1}^{d+1}$ is a regular optimal placement solving (2.11). Denote $\varphi_i = c_i g_i$ and $\Phi = [\varphi_1, \ldots, \varphi_{d+1}] \in \mathbb{R}^{d\times(d+1)}$. Then (2.11) can be written in matrix form as $\Phi\Phi^{\mathrm{T}} = 1/d \sum_{i=1}^{d+1} c_i^2 I_d$. Hence Φ has mutually orthogonal *rows* with row norm as $\sqrt{1/d \sum_{i=1}^{d+1} c_i^2}$. Let $x = [x_1, \ldots, x_{d+1}] \in \mathbb{R}^{d+1}$ be a vector in the orthogonal complement of the row space of Φ. Assume $\|x\| = \sqrt{1/d \sum_{i=1}^{d+1} c_i^2}$. Adding x^{T} after the last row of Φ yields an augmented matrix $\Phi_{\mathrm{aug}} = \left[\Phi^{\mathrm{T}}, x\right]^{\mathrm{T}} \in \mathbb{R}^{(d+1)\times(d+1)}$. It is clear that $\Phi_{\mathrm{aug}}\Phi_{\mathrm{aug}}^{\mathrm{T}} = \frac{1}{d}\sum_{i=1}^{d+1} c_i^2 I_{d+1}$. Thus Φ_{aug} is a scaled orthogonal matrix and its *columns* are mutually orthogonal. The jth column of Φ_{aug} is $[\varphi_j^{\mathrm{T}}, x_j]^{\mathrm{T}} \in \mathbb{R}^{d+1}$ for all $j \in \{1, \ldots, d+1\}$. Note the column norm of Φ_{aug} is $\sqrt{1/d \sum_{i=1}^{d+1} c_i^2}$. Then we have $\|\varphi_j\|^2 + x_j^2 = 1/d \sum_{i=1}^{d+1} c_i^2$ and hence

$$x_j = \pm\sqrt{\frac{1}{d}\sum_{i=1}^{d+1} c_i^2 - c_j^2}. \tag{2.22}$$

The regularity of $\{c_i\}_{i=1}^{d+1}$ ensures $1/d \sum_{i=1}^{d+1} c_i^2 - c_j^2 \geq 0$.

By reversing the above proof, we can obtain an explicit construction algorithm for optimal placement with $n = d+1$ as shown in Algorithm 2.2. The rest is to prove the constructed optimal placements are mutually equivalent. First, given

a vector $x \in \mathbb{R}^{d+1}$ satisfying (2.22), let Φ and Φ' be two different bases of the orthogonal complement of x. Due to orthogonality, there exists an orthogonal matrix $U \in \mathbb{R}^{(d+1)\times(d+1)}$ such that

$$U \begin{bmatrix} \Phi \\ x^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \Phi' \\ x^{\mathrm{T}} \end{bmatrix}. \tag{2.23}$$

Write U as

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}, \tag{2.24}$$

where $U_{11} \in \mathbb{R}^{d\times d}, U_{12} \in \mathbb{R}^{d\times 1}, U_{21} \in \mathbb{R}^{1\times d}$, and $U_{22} \in \mathbb{R}$. Substituting (2.24) into (2.23) gives $U_{21}\Phi + (U_{22} - 1)x^{\mathrm{T}} = 0$. Since the rows of Φ and x^{T} are linearly independent, we have $U_{21} = 0$, $U_{22} = 1$. Thus $U_{12} = 0$ and $U_{11}\Phi = \Phi'$. Therefore, the placements described by Φ and Φ' are differed only by an orthogonal transformation U_{11}. From Definition 2.2, the two placements are equivalent. Second, let $E \in \mathbb{R}^{(d+1)\times(d+1)}$ be a diagonal matrix with diagonal entries as 1 or -1. Given arbitrary x and x' both satisfying (2.22), there exists an E such that $x' = Ex$. Note E is also an orthogonal matrix. It can be analogously proved that the optimal placements would be differed by an orthogonal transformation and a number of flipping of sensors about the target. From Definition 2.2, these placements are also equivalent. □

From the proof of Theorem 2.5, a method for explicitly constructing the unique regular optimal placement in the case of $n = d + 1$ can be summarized as Algorithm 2.2. The following example illustrates Algorithm 2.2.

Example 2.5. *In $\mathbb{R}^3$, consider four bearing-only sensors with sensor-target ranges respectively as $\|r_1\| = 20$, $\|r_2\| = 21$, $\|r_3\| = 22$, and $\|r_4\| = 23$. The measurement noise variance of the ith sensor is $\sigma_i = 0.01$ with $i \in \{1, \ldots, 4\}$. Recall $c_i = 1/(\sigma_i\|r_i\|)$ for bearing-only sensors. Then $c_1^2 = 25.00$, $c_2^2 = 22.68$, $c_3^2 = 20.66$, $c_4^2 = 18.90$ and $1/3\sum_{i=1}^4 c_i^2 = 29.08$. The sequence $\{c_i\}_{i=1}^4$ is regular. From (2.22), choose $x = [2.02, 2.53, 2.90, 3.19]^{\mathrm{T}}$. Compute the SVD of x and use*

Algorithm 2.2 Construction of the unique regular optimal placement $\{g_i\}_{i=1}^{d+1}$ with coefficients $\{c_i\}_{i=1}^{d+1}$.

1: Choose $x = [x_1, \ldots, x_{d+1}] \in \mathbb{R}^{d+1}$ with $x_j = \pm\sqrt{1/d \sum_{i=1}^{d+1} c_i^2 - c_j^2}$ for $i \in \{1, \ldots, d+1\}$.
2: Use the singular value decomposition (SVD) to numerically compute an orthogonal basis of the orthogonal complement of x. Let $x = U\Sigma V^{\mathrm{T}}$ be an SVD of x, where $U \in \mathbb{R}^{(d+1)\times(d+1)}$ is an orthogonal matrix.
3: Let u_i denote the ith column of U. Then $x = \pm\sqrt{1/d \sum_{i=1}^{d+1} c_i^2} u_1$, and Φ can be constructed as

$$\Phi = \sqrt{\frac{1}{d}\sum_{i=1}^{d+1} c_i^2}\,[u_2, \ldots, u_{d+1}]^{\mathrm{T}} \in \mathbb{R}^{d\times(d+1)}. \tag{2.25}$$

4: Compute $g_i = \varphi_i / c_i$ for $i \in \{1, \ldots, d+1\}$.

(2.25) *to compute* Φ *as*

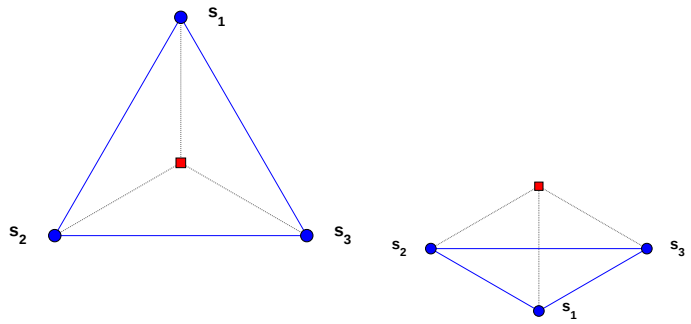
$$\Phi = \begin{bmatrix} -2.5307 & 4.5286 & -0.9906 & -1.0891 \\ -2.9016 & -0.9906 & 4.2568 & -1.2487 \\ -3.1901 & -1.0891 & -1.2487 & 4.0197 \end{bmatrix}.$$

It can be verified $\sum_{i=1}^{4} c_i^2 g_i g_i^{\mathrm{T}} = \Phi\Phi^{\mathrm{T}} = 1/3 \sum_{i=1}^{4} c_i^2 I_3$.

Figure 2.6 and Figure 2.7 show examples of unique optimal placements. Suppose all sensors have the same measurement noise standard deviation. Then the regular triangle in Figure 2.6 (a) is equally-weighted optimal as shown in Section 2.5.2. By Theorem 2.5, the regular triangle placement is also unique. Thus the two equivalent placements in Figure 2.6 represent all possible forms of the equally-weighted optimal placements with $n = 3$ in $\mathbb{R}^2$. Analogously, the three equivalent placements in Figure 2.7 present all possible forms of the equally-weighted optimal placements with $n = 4$ in $\mathbb{R}^3$.

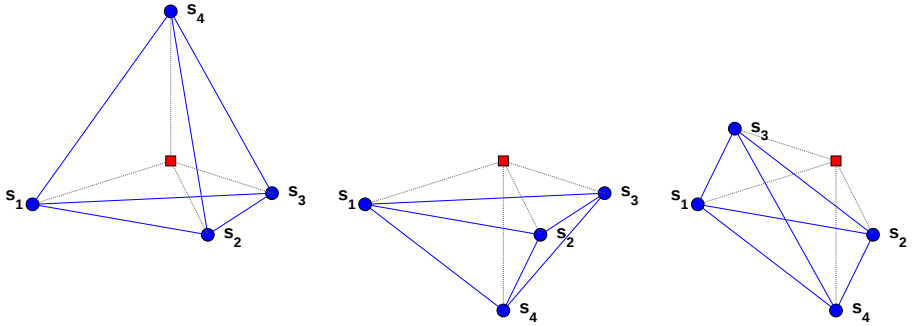
When $n > d + 1$, the regular optimal placement may not be unique. In the next subsection, we will give examples to show the optimal placement may not be unique when $n \geq 4$ in $\mathbb{R}^2$ or $n \geq 6$ in $\mathbb{R}^3$. Now a question remains: whether the regular optimal placement with $n = 5$ in $\mathbb{R}^3$ is unique. The answer is negative. The following gives an explanation as well as an algorithm for explicitly constructing regular optimal placements with $n = 5$ in $\mathbb{R}^3$.

Suppose the sequence $\{c_i\}_{i=1}^{5}$ is regular with respect to $d = 3$. Denote $\varphi_i =$



(a) (b)

Figure 2.6: The unique equally-weighted optimal placements with $n = 3$ in $\mathbb{R}^2$. Red square: target; blue dots: sensors. (a) Regular triangle. (b) Flip s_1 about the target.



(a) (b) (c)

Figure 2.7: The unique equally-weighted optimal placements with $n = 4$ in $\mathbb{R}^3$. Red square: target; blue dots: sensors. (a) Regular tetrahedron. (b) Flip s_4 about the target. (c) Flip s_4 and s_3 about the target.

$c_i g_i$ and $\Phi = [\varphi_1, \ldots, \varphi_5] \in \mathbb{R}^{3\times 5}$. Then (2.11) becomes $\Phi\Phi^{\mathrm{T}} = 1/3 \sum_{i=1}^5 c_i^2 I_3$. There always exists $\Phi' = [\varphi'_1, \ldots, \varphi'_5] \in \mathbb{R}^{2\times 5}$ in the orthogonal complement of the row space of Φ such that

$$\begin{bmatrix} \Phi \\ \Phi' \end{bmatrix} \begin{bmatrix} \Phi^{\mathrm{T}} & \Phi'^{\mathrm{T}} \end{bmatrix} = \frac{1}{3} \sum_{i=1}^5 c_i^2 I_5,$$

which implies $\|\varphi_j\|^2 + \|\varphi'_j\|^2 = 1/3 \sum_{i=1}^5 c_i^2$ and $\Phi'\Phi'^{\mathrm{T}} = 1/3 \sum_{i=1}^5 c_i^2 I_2$. Thus $\{\varphi'_j\}_{j=1}^5$ represents a 2D regular optimal placement with $\|\varphi'_j\| = \sqrt{1/3 \sum_{i=1}^5 c_i^2 - c_j^2}$ for all $j \in \{1, \ldots, 5\}$ (it can be verified $\{\|\varphi'_j\|\}_{j=1}^5$ is regular with respect to $d = 2$). Therefore, to obtain Φ, we can first construct $\{\varphi'_j\}_{j=1}^5$ using Algorithm 2.1 for example, and then find Φ in the orthogonal complement of the row space of Φ'. Since $\{\varphi'_i\}_{i=1}^5$ may have non-equivalent solutions, $\{\varphi_i\}_{i=1}^5$ would not be unique up to the equivalence.

2.5.4 Distributed Construction

When there are a large number of sensors, it might be inconvenient to design the optimal placement involving all sensors. The following property can be applied to construct large-scale optimal placements in a distributed manner. The 2D versions of the following result have been proposed in [11, 38, 13].

Theorem 2.6. *The union of multiple disjoint regular optimal placements in* $\mathbb{R}^d$ *(*$d = 2$ *or* 3*) is still a regular optimal placement in* $\mathbb{R}^d$*.*

Proof. In $\mathbb{R}^d$, consider multiple disjoint regular optimal placements: $\{c_i, g_i\}_{i\in\mathcal{I}_k}$ with $\mathcal{I}_k$ as the index set of the kth placement ($k = 1, \ldots, q$). The term disjoint as used here means that different placements share no common sensors. Define $\mathcal{I} = \bigcup_{k=1}^{q} \mathcal{I}_k$. If $|\cdot|$ denotes the cardinality of a set, then $|\mathcal{I}| = \sum_{k=1}^{q} |\mathcal{I}_k|$.

For the kth placement, since $\{c_i, g_i\}_{i\in\mathcal{I}_k}$ is regular optimal in $\mathbb{R}^d$, from Theorem 2.1 we have

$$\sum_{i\in\mathcal{I}_k} c_i^2 g_i g_i^{\mathrm{T}} = \frac{1}{d} \sum_{i\in\mathcal{I}_k} c_i^2 I_d.$$

For the union placement $\{c_i, g_i\}_{i\in\mathcal{I}}$, we have

$$\begin{aligned}\sum_{j\in\mathcal{I}} c_j^2 g_j g_j^{\mathrm{T}} &= \sum_{k=1}^{q} \sum_{i\in\mathcal{I}_k} c_i^2 g_i g_i^{\mathrm{T}} \\ &= \frac{1}{d} \sum_{k=1}^{q} \sum_{i\in\mathcal{I}_k} c_i^2 I_d \\ &= \frac{1}{d} \sum_{j\in\mathcal{I}} c_j^2 I_d.\end{aligned}$$

By Theorem 2.1, the union placement is regular optimal in $\mathbb{R}^d$. □

Theorem 2.6 implies that a large-scale regular optimal placement can be constructed in a distributed manner: firstly divide the large-scale placement into a number of disjoint regular sub-placements, secondly construct each regular optimal sub-placement, and finally combine these optimal sub-placements together to obtain a large regular optimal placement. We call this kind of method as *distributed construction*. Because the combination of the optimal sub-placements

can be arbitrary, distributed construction will lead to an infinite number of optimal placements for the large system. These optimal placements have the same FIM and $\|G\|^2$, but they are generally *non-equivalent*. Theorem 2.6 also implies that only regular placements can be possibly divided into some regular subsets.

Figure 2.8 gives examples of optimal placements generated by distributed construction. Suppose all sensors have the same measurement noise standard deviation. The placement with $n = 6$ in Figure 2.8 (a), (b) or (c) is a combination of two regular triangles with $n = 3$ (the sensors with the same color form a triangle optimal placement). The placement with $n = 8$ in Figure 2.8 (d) or (e) is a combination of two regular optimal placements with $n = 4$ as shown in Figure 2.7 (c), which is equivalent to the regular tetrahedron. Thus by Theorem 2.6 all placements in Figure 2.8 are regular optimal.

The distributed construction method is suitable for (but not limited to) constructing equally-weighted optimal placements. That is because a equally-weighted placement can be easily divided into some regular subsets. Suppose we have an equally-weighted placements in $\mathbb{R}^d$. Its coefficient sequence $\{c_i\}_{i=1}^n$ satisfies $c_i = c_j$ for all $i \neq j$. Then $\{c_i\}_{i=1}^n$ can be divided into a number of subsets. As long as the cardinality of each subset is no smaller than d, the subsets are all regular (refer to Example 2.1). We next present two examples to show how to divide equally-weighted placements into subsets. (i) For any integer $n \geq 4$, it is obvious that there exist nonnegative integers m_1 and m_2 such that n can be decomposed as $n = 2m_1 + 3m_2$. Thus in $\mathbb{R}^2$ we can always distributedly construct an equally-weighted optimal placement with $n \geq 4$ by using the ones with $n = 2$ or 3. (ii) For any integer $n \geq 6$, there exist nonnegative integers m_1, m_2 and m_3 such that $n = 3m_1 + 4m_2 + 5m_3$. Thus in $\mathbb{R}^3$ we can always distributedly construct an equally-weighted optimal placement with $n \geq 6$ by using the ones with $n = 3$, 4 or 5. Note distributed construction yields an infinite number of non-equivalent optimal placements. Hence the above two examples also imply that equally-weighted placements with $n \geq 4$ in $\mathbb{R}^2$ or $n \geq 6$ in $\mathbb{R}^3$ are not unique.

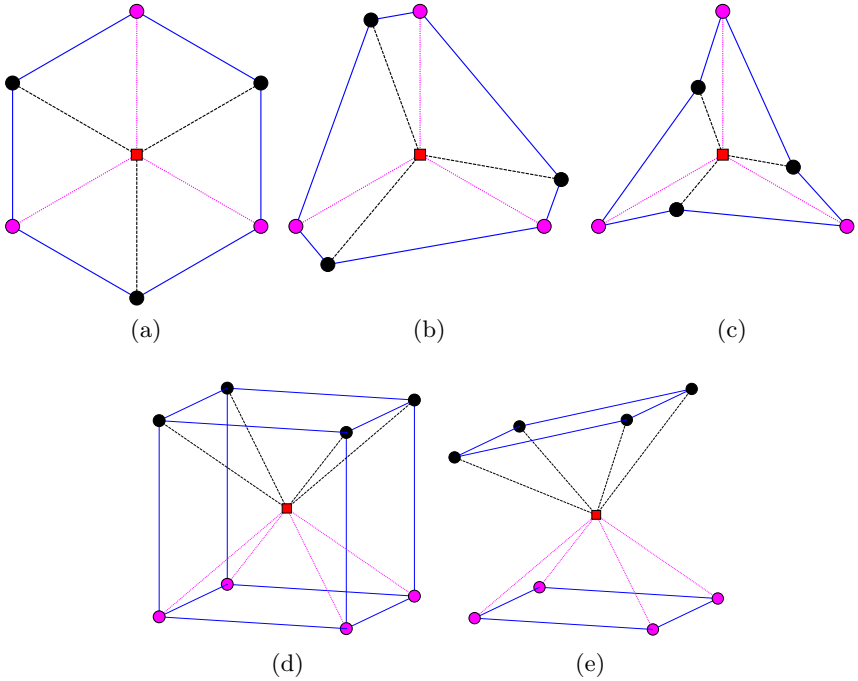


Figure 2.8: Examples of distributedly constructed optimal placements. Red square: target; dots: sensors.

2.6 Autonomously Deploy Optimal Sensor Placement

In addition to analytically determining optimal sensor placements, it is also practically important to study how to autonomously deploy the optimal placements. There are generally three approaches to autonomous optimal sensor deployment. (i) One approach is to use the optimal control formulation mentioned above. (ii) The second approach is to develop numerical methods to solve the parameter optimization formulation. For example, by numerically maximizing the determinant of the FIM, sensor trajectory optimization algorithms are obtained in [37, 96]. Compared to the optimal control approach, the second approach is easy to be designed and implemented, and various constraints of sensor trajectories can be easily included. (iii) The third approach is to design autonomous deployment algorithms based directly on analytical results of optimal sensor placement. For example, the work in [83] proposed a distributed motion coordination algorithm to autonomously deploy sensors for target tracking. Loosely speaking, that algorithm is designed based on the fact that a placement is optimal if range-only

sensors are uniformly distributed around the target. However, one limitation of the approach used in [83] is that the uniformly distributed placement is merely a special optimal placement as pointed out in Section 2.5.2. There are an infinite number of optimal placements that can optimize the objective function. These optimal placements may be more appropriate than the uniformly distributed one for certain initial placements and sensor trajectory constraints. More importantly, the uniformly distributed placement is optimal only if the sensors are equally weighted. If some of the sensors can provide more accurate measurements than the others, the uniformly distributed placement will not be optimal any more. With the above analysis, we will adopt the numerical optimization approach to solve autonomous sensor deployment problems in this chapter.

Based on the previous analytical results, we first propose a gradient control law to autonomously deploy optimal sensor placements in 2D and 3D. Considering the trajectories of the mobile sensors may be constrained in practice, we further propose a control strategy that can deploy optimal sensor placements while fulfilling the trajectory constraints. When there are trajectory constraints, the corresponding control strategy is only applicable to range-only sensors; the control law without sensor trajectory constraints is applicable to any of the three sensor types.

2.6.1 Gradient Control without Trajectory Constraints

Assume the motion model of sensor i to be $\dot{s}_i = u_i$, where $u_i \in \mathbb{R}^d$ is the control input. Then we have $\dot{r}_i = u_i$ because $r_i = s_i - p$ and the target position estimation p is given. Let $r = [r_1^{\mathrm{T}}, \ldots, r_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{dn}$. Denote β as the constant lower bound of $\|G\|^2$ in Theorems 2.1 and 2.2. Then the optimal placement set is $\mathcal{E}_0 = \{r \in \mathbb{R}^{dn} : \|G\|^2 - \beta = 0\}$. Choose the Lyapunov function as $V(r) = 1/4(\|G\|^2 - \beta)$. Clearly V is positive definite with respect to $\mathcal{E}_0$. Denote $\partial V/\partial r_i$ as the Jacobian of V with respect to r_i. Then we have

$$\dot{V} = \sum_{i=1}^{n} \frac{\partial V}{\partial r_i} \dot{r}_i = \sum_{i=1}^{n} \frac{c_i^2}{\|r_i\|} g_i^{\mathrm{T}} G P_i \dot{r}_i,$$

where $P_i = I_d - g_i g_i^{\mathrm{T}}$ is an orthogonal projection matrix satisfying $P_i^{\mathrm{T}} = P_i$, $P_i^2 = P_i$, and $\mathrm{Null}(P_i) = \mathrm{span}\{g_i\}$. $\mathrm{Null}(\cdot)$ denotes the null space of a matrix. Design the gradient control law as

$$\dot{r}_i = -P_i G g_i. \tag{2.26}$$

Then

$$\dot{V} = -\sum_{i=1}^{n} \frac{c_i^2}{\|r_i\|} \|P_i G g_i\|^2 \leq 0$$

and $\dot{V} = 0$ when $P_i G g_i = 0$ for all $i \in \{1, \ldots, n\}$.

Proposition 2.2. *For any initial condition $r(0) \in \mathbb{R}^{dn}$ with $\|r_i(0)\| \neq 0$ for all $i \in \{1, \ldots, n\}$, the solution to the nonlinear r-dynamics* (2.26) *asymptotically converges to the set*

$$\mathcal{E} = \{r \in \mathcal{S} : P_i G g_i = 0, \quad i = 1, \ldots, n\},$$

where $\mathcal{S} = \{r \in \mathbb{R}^{dn} : \|r_i\| = \|r_i(0)\|,\ i = 1, \ldots, n\}$.

Proof. The time derivative of $\|r_i\|$ is

$$\frac{\mathrm{d}\|r_i\|}{\mathrm{d}t} = \frac{r_i^{\mathrm{T}}}{\|r_i\|} \dot{r}_i = -g_i^{\mathrm{T}} P_i G g_i = 0. \tag{2.27}$$

The last equality uses the fact $g_i^{\mathrm{T}} P_i = 0$. By (2.27) we have $\|r_i(t)\| \equiv \|r_i(0)\| \neq 0$. Hence $\mathcal{S}$ is a positive invariant set with respect to the r-dynamics. The set $\mathcal{S}$ consists of a group of spheres in $\mathbb{R}^d$ centered at the origin. Thus $\mathcal{S}$ is compact. Note $\dot{V} = 0$ and $\dot{r} = 0$ for all points in $\mathcal{E}$. By the invariance principle [68], every solution starting in $\mathcal{S}$ asymptotically converges to $\mathcal{E}$. □

By Proposition 2.2, the r-dynamics converge either to the optimal placement set $\mathcal{E}_0$ or the set $\mathcal{E} \setminus \mathcal{E}_0$. By introducing Lagrange multipliers γ_i, $i = 1, \ldots, n$, the constrained optimization problem (2.8) is equivalent to minimizing the Lagrangian function $L = \|G\|^2 + \sum_{i=1}^{n} \gamma_i (g_i^{\mathrm{T}} g_i - 1)$. By calculating $\partial L / \partial g_i = 0$, we can show that $\mathcal{E}$ is the *critical point* set, which consists of not only minimizers of

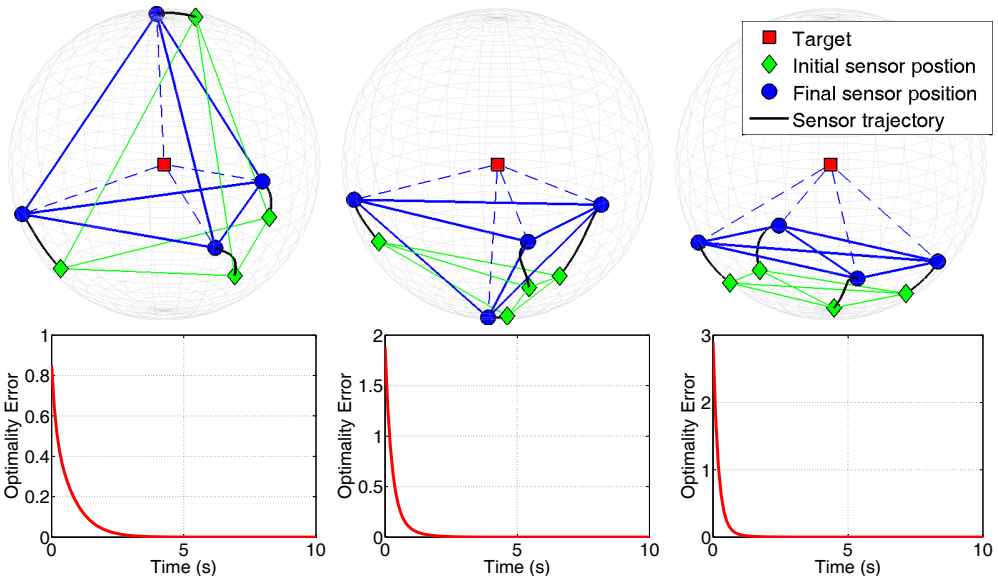
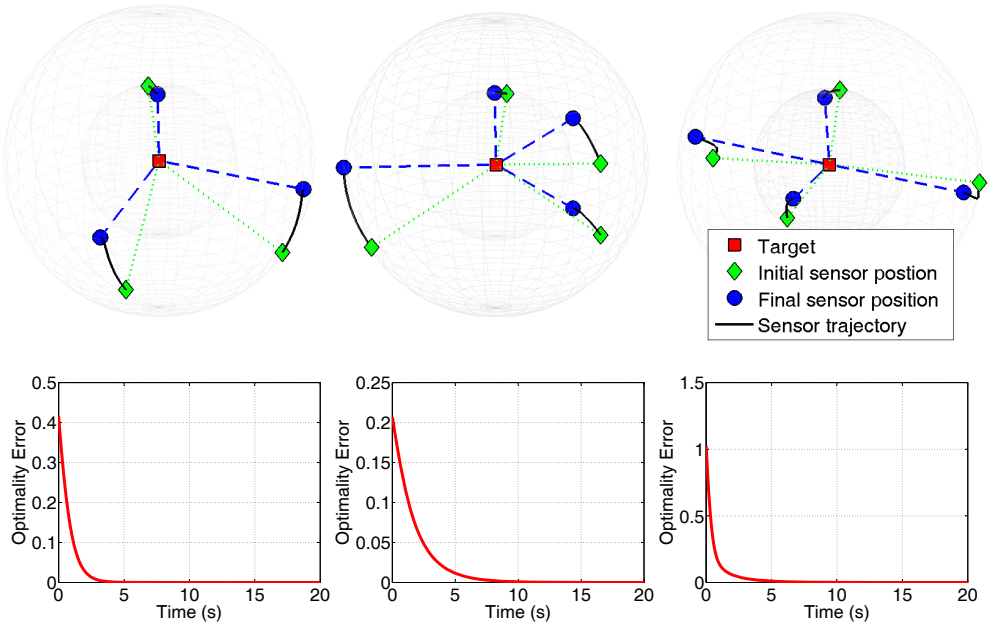


Figure 2.9: Gradient control of equally-weighted (regular) placements with $n = 4$ in $\mathbb{R}^3$.

$\|G\|^2$ (i.e., optimal placements) but also saddle points and maximizers of $\|G\|^2$ (i.e., non-optimal placements). The sets $\mathcal{E}_0$ and $\mathcal{E}$ are equilibrium manifolds. It is noticed that nonlinear stabilization problems involving equilibrium manifolds also emerge in formation control area recently [74, 36, 107]. It is possible to conduct strict stability analysis including identifying the attractive region of $\mathcal{E}_0$ by using center manifold theory [74, 107] or differential geometry [36]. But that will be non-trivial because the geometric structure of $\mathcal{E}_0$ is extremely complicated as shown in [39].

Figure 2.9 and Figure 2.10 show several optimal placements obtained by the proposed gradient control law. Due to space limitations, we only show 3D examples. The three final converged placements in Figure 2.9 are actually the three regular optimal ones shown in Figure 2.7. The three final placements in Figure 2.10 are the two as illustrated in Figure 2.2 (b) and (c). Clearly the numerical results are consistent with our previous analysis. The *optimality error* refers to the difference between $\|G\|^2$ and its lower bound given in (2.10) or (2.13). The optimality error can be used as a numerical indicator to evaluate the optimality of a placement. As shown in Figure 2.9 and Figure 2.10, the optimality errors all converge to zero.



(a) $n = 3$, $k_0 = 1$ (b) $n = 4$, $k_0 = 1$ (c) $n = 4$, $k_0 = 2$

Figure 2.10: Gradient control of irregular placements in $\mathbb{R}^3$.

2.6.2 Gradient Control with Trajectory Constraints

The mobile sensors usually cannot move freely in practical applications. For example, the trajectory of each sensor may be constrained in the boundary of a set in $\mathbb{R}^d$. In order to autonomously deploy an optimal sensor placement, we need to solve two problems: (i) minimize the objective function $\|G\|^2$, and in the meantime (ii) fulfill the trajectory constraint of each mobile sensor. Hence we define two artificial potentials for autonomous sensor deployment. First, define $V_I = \|G\|^2$ as the inter-sensor potential. The placement is optimal when V_I is minimized. Second, define V_E as the external potential corresponding to the sensor trajectory constraints. The sensor trajectory constraints are fulfilled if and only if $V_E = 0$. By combining the two potentials we have the total potential as

$$V = k_I V_I + k_E V_E,$$

where $k_I, k_E > 0$. The gradient control law for sensor i is designed as

$$\dot{s}_i = -(k_I \nabla_{s_i} V_I + k_E \nabla_{s_i} V_E), \tag{2.28}$$

where ∇_{s_i} denotes the gradient with respect to s_i. The target is assumed to be stationary. Hence we have $\nabla_{s_i} V_I = \nabla_{r_i} V_I$ as $r_i = s_i - p$.

Remark 2.4. *Our previous analytical results are based on one assumption that the coefficient for each sensor is constant. This assumption should also be satisfied while the sensors are moving. When there are no trajectory constraints, the assumption can be satisfied when the sensors are moving on a circle or sphere. But when there are trajectory constraints, the assumption may be invalid when the trajectory constraints are fulfilled. Hence control law* (3.14) *only applies to range-only sensors. That is because the coefficients of range-only sensors are merely function of their noise covariances which are assumed to be independent to the sensor trajectories.*

Inter-sensor Force $\nabla_{s_i} V_I$

Recall $G = \sum_{i=1}^{n} c_i^2 g_i g_i^{\mathrm{T}}$ and $G^{\mathrm{T}} = G$. Then the differential of V_I is

$$\begin{aligned}
\mathrm{d}V_I &= \mathrm{d}\mathrm{tr}\left(G^2\right) \\
&= 2\mathrm{tr}\left(G\mathrm{d}G\right) \\
&= 2\mathrm{tr}\left(G\sum_{i=1}^{n} c_i^2 \mathrm{d}g_i g_i^{\mathrm{T}}\right) + 2\mathrm{tr}\left(G\sum_{i=1}^{n} c_i^2 g_i \mathrm{d}g_i^{\mathrm{T}}\right) \\
&= 2\sum_{i=1}^{n} c_i^2 g_i^{\mathrm{T}} G \mathrm{d}g_i + 2\sum_{i=1}^{n} c_i^2 \mathrm{d}g_i^{\mathrm{T}} G g_i \\
&= 4\sum_{i=1}^{n} c_i^2 g_i^{\mathrm{T}} G \mathrm{d}g_i. \qquad (2.29)
\end{aligned}$$

Furthermore, from $g_i = r_i / \|r_i\|$ we have

$$\mathrm{d}g_i = \frac{1}{\|r_i\|} P_i \mathrm{d}r_i, \qquad (2.30)$$

where $P_i = I_d - g_i g_i^{\mathrm{T}} \in \mathbb{R}^{d\times d}$ is an orthogonal projection matrix. Substituting (2.30) into (2.29) yields

$$\nabla_{s_i} V_I = \nabla_{r_i} V_I = \frac{4c_i^2}{\|r_i\|} P_i G g_i. \qquad (2.31)$$

Remark 2.5. *The implementation of* (2.31) *requires* all-to-all communications

among the sensors due to the term G. However, it is possible to implement (2.31) *in a distributed way. To do that, each sensor need to maintain an estimate $\hat{G}_i$ of G, and $\hat{G}_i$ should converge to G given a non-complete underlying graph. The distributed estimation of G actually is an average consensus problem which has been investigated extensively (see, for example, [95, 80, 102]).*

External Force $\nabla_{s_i} V_E$

The external potential V_E is chosen according to practical application requirements. Here we use two specific scenarios, one of which is 2D and the other is 3D, to demonstrate how control strategy (3.14) can be applied to solve practical problems.

1) A 2D Scenario Suppose the sensors can only move on the boundary of an ellipse, while the stationary target is located inside the ellipse (see Figure 2.11). The position of sensor i must satisfy

$$(s_i - s_0)^{\mathrm{T}} Q (s_i - s_0) = 1,$$

where $s_0 \in \mathbb{R}^2$ is the center of the ellipse and $Q = \mathrm{diag}\{1/a^2, 1/b^2\} \in \mathbb{R}^{2\times 2}$ with $a > 0$ and $b > 0$ as the lengths of the two semi-axes, respectively. According to the constraint, we choose V_E as

$$V_E = \sum_{i=1}^{n} \left[(s_i - s_0)^{\mathrm{T}} Q (s_i - s_0) - 1 \right]^2. \tag{2.32}$$

Then it is straightforward to obtain

$$\nabla_{s_i} V_E = 4 \left[(s_i - s_0)^{\mathrm{T}} Q (s_i - s_0) - 1 \right] Q (s_i - s_0). \tag{2.33}$$

The implementation of (2.33) can be distributed because it only requires the information of sensor i. The parameters of the ellipse such as Q and s_0 should be also known by sensor i.

2) A 3D Scenario

We now consider a 3D scenario which can be applied to cooperative air and

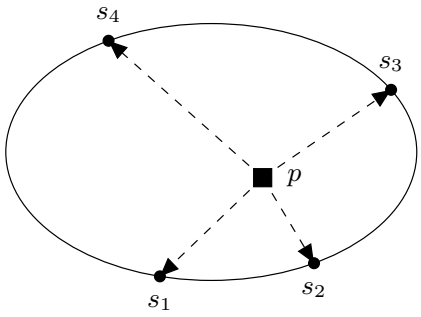


Figure 2.11: An illustration of the 2D scenario where all mobile sensors move on the boundary of an ellipse.

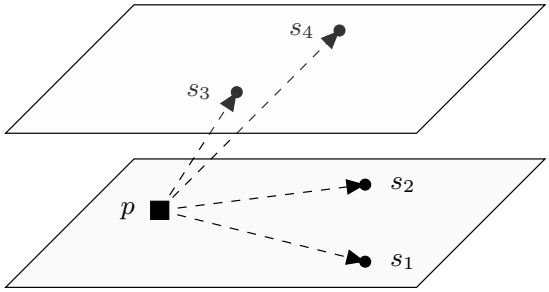


Figure 2.12: An illustration of the 3D scenario where each sensor moves at a fixed altitude.

ground surveillance [52, 60]. Suppose there are multiple UAVs and unmanned ground vehicles (UGVs). Each vehicle is equipped with a range-only sensor, which can measure the range between the vehicle and a target. The UAVs fly at fixed altitudes and the UGVs move on the ground with altitude as zero (see Figure 2.12). Denote ℓ_i as the altitude of vehicle i. Then the external potential V_E can be chosen as

$$V_E = \sum_{i=1}^{n} (e_3^{\mathrm{T}} s_i - \ell_i)^2, \tag{2.34}$$

where $e_3 = [0, 0, 1]^{\mathrm{T}} \in \mathbb{R}^3$. Clearly all the altitude constraints are satisfied if and only if $V_E = 0$. It is straightforward to obtain

$$\nabla_{s_i} V_E = 2(e_3^{\mathrm{T}} s_i - \ell_i) e_3. \tag{2.35}$$

Compatibility of V_I and V_E

The proposed gradient control law (3.14) guarantees that the gradient $k_I \nabla_{s_i} V_I + k_E \nabla_{s_i} V_E$ for all i will converge to zero. However, $k_I \nabla_{s_i} V_I + k_E \nabla_{s_i} V_E = 0$ does not simply imply $\nabla_{s_i} V_I = \nabla_{s_i} V_E = 0$. To have that implication, the external

potential V_E and the inter-sensor potential V_I should be *compatible* with each other. By compatible, we mean $k_I\nabla_{s_i}V_I + k_E\nabla_{s_i}V_E = 0$ for all i if only if $\nabla_{s_i}V_I = \nabla_{s_i}V_E = 0$ for all i. If V_E is not compatible with V_I, it is possible that $k_I\nabla_{s_i}V_I + k_E\nabla_{s_i}V_E = 0$ while $\nabla_{s_i}V_I \neq 0$ and $\nabla_{s_i}V_E \neq 0$. Since $\nabla_{s_i}V_I = 0$ is a necessary condition for optimal placements, the final converged placement may be non-optimal in non-compatible cases.

We next prove $V_I = \|G\|^2$ is compatible with V_E given in (2.32) or (2.34). To do that, we will show $\nabla_{s_i}V_I$ in (2.31) is impossible to be parallel to $\nabla_{s_i}V_E$ in (2.33) or (2.35). Firstly consider $\nabla_{s_i}V_I$ in (2.31). Since $P_i = I_d - g_i g_i^{\mathrm{T}}$, we have $\mathrm{Null}\,(P_i) = \mathrm{span}\{g_i\}$ and hence $P_i r_i = 0$. As a result,

$$r_i^{\mathrm{T}}\nabla_{s_i}V_I = 0,$$

which means $\nabla_{s_i}V_I \perp r_i$. Secondly, consider $\nabla_{s_i}V_E$ given in (2.33) and (2.35).

1) In the 2D scenario, the vector $\nabla_{s_i}V_E$ is the *normal vector* of the ellipse at point s_i. Geometrically it is clear that $\nabla_{s_i}V_E$ is impossible to be orthogonal to r_i for any p inside the ellipse and any s_i on the ellipse. As a result, $\nabla_{s_i}V_E$ is not parallel to $\nabla_{s_i}V_I$, and hence $k_I\nabla_{s_i}V_I + k_E\nabla_{s_i}V_E = 0$ if and only if $\nabla_{s_i}V_I = \nabla_{s_i}V_E = 0$ for all i.

2) In the 3D scenario, the vector $\nabla_{s_i}V_E$ is the *normal vector* of the horizontal planes. Define $\mathcal{I} = \{1, \ldots, n\}$, $\mathcal{I}_{>0} = \{i \in \mathcal{I} : \ell_i > 0\}$ and $\mathcal{I}_{=0} = \{i \in \mathcal{I} : \ell_i = 0\}$. Geometrically it is clear that if $\ell_i > 0$, then $\nabla_{s_i}V_E$ is impossible to be orthogonal to r_i. Hence $\nabla_{s_i}V_E$ is not parallel to $\nabla_{s_i}V_I$ for all $i \in \mathcal{I}_{>0}$. As a result, $k_I\nabla_{s_i}V_I + k_E\nabla_{s_i}V_E = 0$ for all $i \in \mathcal{I}_{>0}$ if and only if $\nabla_{s_i}V_I = \nabla_{s_i}V_E = 0$ for all $i \in \mathcal{I}_{>0}$. Since the inter-sensor forces are *mutual*, if $\nabla_{s_i}V_I = 0$ for all $i \in \mathcal{I}_{>0}$, then we have $\sum_{i\in\mathcal{I}_{=0}} \nabla_{s_i}V_I = 0$. Furthermore, since r_i for all $i \in \mathcal{I}_{=0}$ is located in the plane with $\ell_i = 0$, we have that $\nabla_{s_i}V_I$ is within the plane and cannot be parallel to $\nabla_{s_i}V_E$. Thus $k_I\nabla_{s_i}V_I + k_E\nabla_{s_i}V_E = 0$ for all $i \in \mathcal{I}_{=0}$ if and only if $\nabla_{s_i}V_I = \nabla_{s_i}V_E = 0$ for all $i \in \mathcal{I}_{=0}$. Therefore, $k_I\nabla_{s_i}V_I + k_E\nabla_{s_i}V_E = 0$ for all $i \in \mathcal{I}$ if and only if $\nabla_{s_i}V_I = \nabla_{s_i}V_E = 0$ for all $i \in \mathcal{I}$.

In fact, if the sensors are constrained on the smooth boundary of an arbitrary

convex set in $\mathbb{R}^d$ and the target is located inside the convex set, the norm vector $\nabla_{s_i} V_E$ is impossible to be parallel to $\nabla_{s_i} V_I$. Hence the corresponding V_E will always be compatible with $V_I = \|G\|^2$. We only consider the case that sensors are constrained on the boundary of a set in this chapter. But more complicated issues such as obstacle avoidance and collision avoidance among sensors could also be solved by introducing more external potentials. At last, it should be noted that the compatibility of V_I and V_E does not guarantee that the final converged placement is optimal. That is because $\nabla_{s_i} V_I = 0$ for all i is only necessary but not sufficient for a placement to be optimal.

2.6.3 Simulation Results

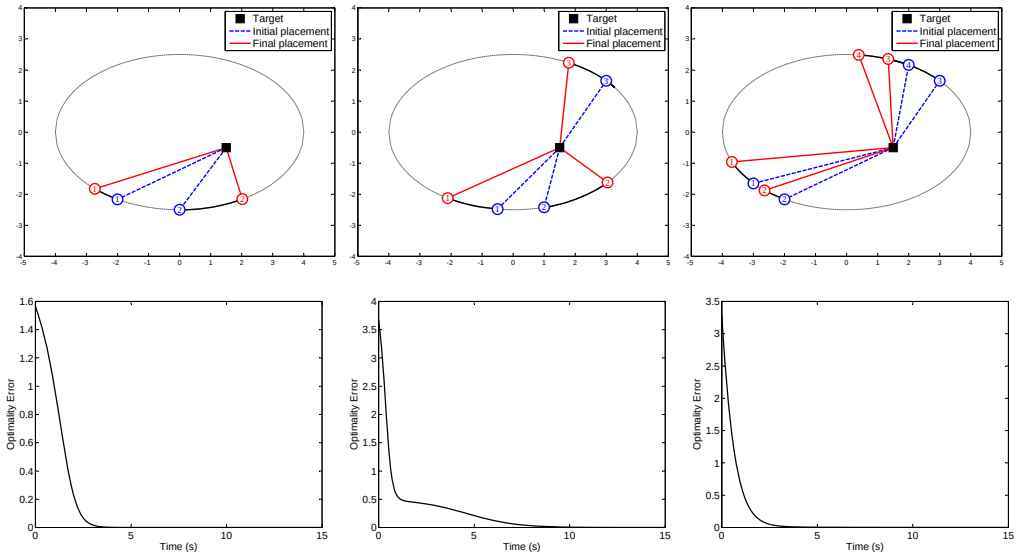
We next present several scenarios to verify the proposed gradient control law with sensor trajectory constraints. It is assumed that the sensor network has all-to-all communications.

Autonomous Sensor Deployment for Static Targets

We first consider static targets.

2D Scenario In the 2D scenario, all sensors are constrained on the boundary of an ellipse while the target is located inside the ellipse. The external force $\nabla_{s_i} V_E$ is given in (2.33). The standard deviations of the sensor noises are assumed to be one, i.e., $\sigma_i = 1$ for all i. Note the final optimal placement is determined only by the relative values of $\{\sigma_i\}_{i=1}^n$ but not the absolute values. As shown in Figure 2.13, given appropriate initial placements, the control strategy can steer sensors to optimal placements while ensuring the sensor trajectory constraints fulfilled. The optimality error shown in Figure 2.13 refers to the difference between the objective function $\|G\|^2$ and its lower bound given in (2.10) or (2.13). As can be seen, the control strategy can efficiently reduce the optimality errors to zero.

Numerically it is clear the final placements in Figure 2.13 are optimal because the optimality errors converge to zero. We next *analytically* examine the optimality of the final placements based on the previous analytical results. As

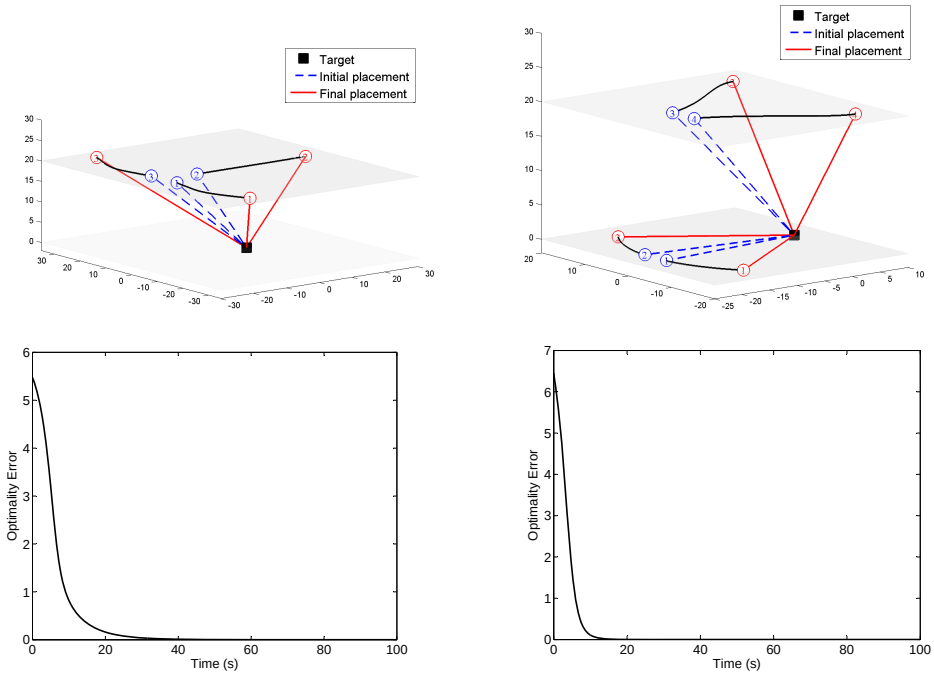


(a) $n = 2$ (b) $n = 3$ (c) $n = 4$

Figure 2.13: Sensor trajectory and optimality error for the 2D scenario.

shown in Figure 2.13(a), the angle subtended by the two sensors in the final placement is 90 deg. Hence we have $g_1 \perp g_2$ in the final placement. Then the final placement is optimal by Theorem 2.3. As shown in Figure 2.13(b), the sensors are steered to equally spaced angular positions around the target. The angle subtended by any two sensors in the final placement is 120 deg. This kind of equally spaced angular placements are optimal as concluded in Section 2.5.2. At the first glance, it is intuitively unclear whether the final placement in Figure 2.13(c) is optimal. The final placement of the four sensors can be viewed as a combination of two sub-placements: the sub-placement of sensors 1 and 3, and the sub-placement of sensors 2 and 4. Note the angle subtended by sensors 1 and 3 and the one by sensors 2 and 4 are both 90 deg in the final placement. That means $g_1 \perp g_3$ and $g_2 \perp g_4$, and the two sub-placements are (regular) optimal, respectively. By Theorem 2.6, the final placement Figure 2.13(c) is optimal.

3D Scenario In the 3D scenario, it is assumed that each sensor is carried by a UAV or UGV that moves at a fixed altitude. The standard deviations of the sensor noises are assumed to be one, i.e., $\sigma_i = 1$ for all i. Simulation results with $n = 3$ and 4 are respectively shown in Figure 2.14(a) and Figure 2.14(b). As shown in Figure 2.14, the optimality errors in the two cases are both reduced to zero efficiently.



(a) $n = 3$ (b) $n = 4$

Figure 2.14: Sensor trajectory and optimality error for the 3D scenario.

In Figure 2.14(a), the angle subtended by any two sensors is 90 deg in the final placement. That means g_1, g_2 and g_3 are mutually orthogonal in the final placement. By Theorem 2.3, the final placement is optimal. The final optimal placement shown in Figure 2.14(b) actually is the one given in Figure 2.7. In both cases of $n = 3$ and $n = 4$, the final converged placements are optimal and the sensor position constraints are simultaneously fulfilled.

Autonomous Sensor Deployment for Dynamic Targets

We next apply the proposed control strategy to track a dynamic target. We consider a 3D cooperative target tracking scenario where there are three UAVs flying at the same altitude. Each UAV carries a range-only sensor to measure the distance to the ground target. In the simulation, the target moves on a non-flat ground. The 3D maneuvering motion of the target is given as

$$p(t) = \begin{bmatrix} 0.5t \\ 10\sin(\pi/5t) \\ 3 - 3\cos(\pi/20t) \end{bmatrix}.$$

See Figure 2.15 for the target trajectory.

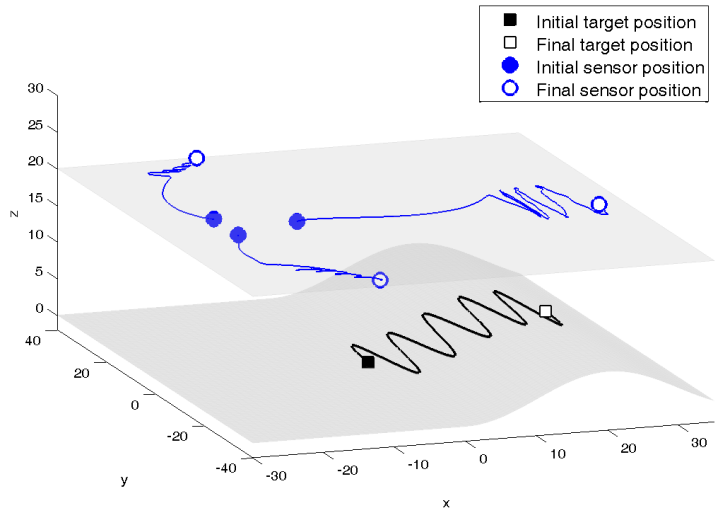


Figure 2.15: Autonomous optimal sensor deployment to track a dynamic target. The target moves on the non-flat ground and the three UAVs fly at a fixed altitude.

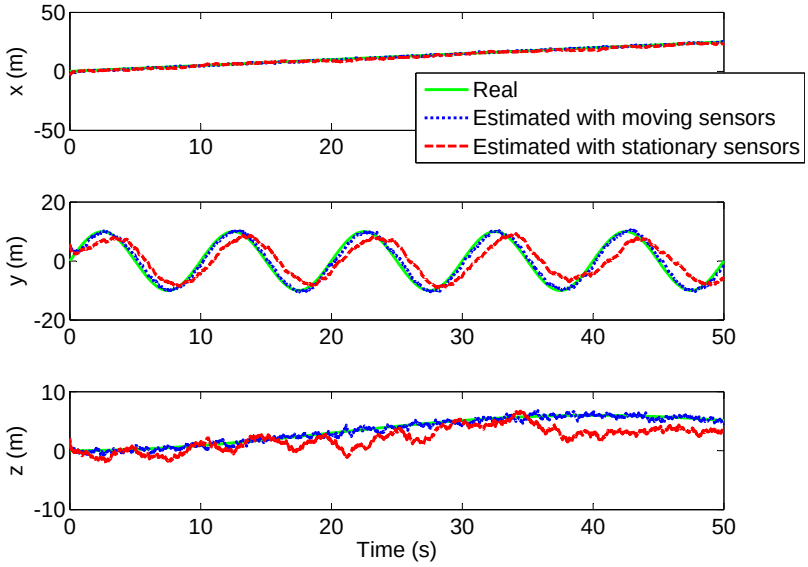


Figure 2.16: Target position estimation results by stationary and moving sensors.

In this cooperative target tracking scenario, all UAVs need to share their range measurements to estimate the target position on one hand; and the UAVs need to move to form an optimal placement that can improve the target estimation accuracy on the other hand. The autonomous optimal sensor deployment algorithm is summarized as Algorithm 2.3.

The real initial position of the target is $p(0) = [0, 0, 0]^{\mathrm{T}}$, while the initial target estimate for the EKF is $\hat{p}(0) = [-4, 5, 3]^{\mathrm{T}}$. The standard deviation of noise w_i is set as three meter. By Algorithm 2.3, the trajectories of the three moving sensors are obtained as shown in Figure 2.15. The behaviors of the

Algorithm 2.3 Autonomous Optimal Sensor Deployment for Target Tracking

1: Obtain an initial estimate of the target position $\hat{p}(0)$.
2: At time t, estimate the target position $\hat{p}(t)$ using a centralized extended Kalman filter (EKF). The EKF is established based on the following process and measurement models:

$$\begin{aligned} \dot{p} &= v, \\ m_i &= \|p - s_i\| + w_i, \quad i = 1, \ldots, n \end{aligned}$$

where $m_i \in \mathbb{R}$ is the measurement of sensor i, and $v \in \mathbb{R}^3$ and $w_i \in \mathbb{R}$ for all i are white Gaussian noises.
3: Based on $\hat{p}(t)$, autonomously deploy the optimal sensor placement using the control strategy (3.14).
4: At time $t + 1$, go to Step 2.

three sensors in Figure 2.15 can be explained as the following: the three sensors attempt to form a placement where each angle subtended at the target by any two sensors is 90 deg. In order to illustrate how optimal placements can improve target tracking performance, we compare the target estimation results in the cases of moving and stationary sensors. In the case of stationary sensors, the three sensors stay statically at the initial placement shown in Figure 2.15. The initial placement is obviously not optimal. The target estimation results in the two cases are shown in Figure 2.16. Note the parameters of the EKF are the same in the two cases. As can be seen, the target can be tracked accurately when the sensors are steered to maintain an optimal placement, while the tracking performance is worse when the sensors are stationary.

Chapter 3

Bearing-only Formation Control

3.1 Introduction

As discussed in the previous chapter, vision can be practically treated as a bearing-only sensor in cooperative target tracking and localization. In this chapter, we will show that vision-based formation control is another area where vision can be thought of as a bearing-only sensor. More specifically, vision-based formation control can be formulated as a bearing-only formation control problem. Unlike optimal sensor placement, bearing-only formation control is a very new research topic and has not attracted much attention yet. Moreover, the bearing measurements are ultimately used to estimate the target position in cooperative target tracking and localization. As a comparison, no position estimation will be involved in bearing-only formation control and the control will be directly implemented based on bearing measurements.

There are a number of challenging theoretical problems regarding bearing-only formation control.

The first problem we need to consider is how to properly utilize the bearing measurements for control. There are generally two approaches. The first approach is that each vehicle uses its bearing measurements to estimate/track the positions of their neighbors. One may refer to [92] for bearing-only target tracking algorithms. Once the neighbors' positions have been estimated, they

can be used for control. Hence in the first approach, the formation control is still based on position information and conventional control laws can be applied. But several problems need to be noticed. Firstly, since the positions are estimated from bearings, this approach leads to a coupled nonlinear estimation and control problem, whose stability needs to be rigorously analyzed. Secondly, position tracking using bearing-only measurements requires certain observability conditions, details of which are out of the scope of this work. Intuitively speaking, in order to localize a vehicle from bearing measurements, we need to measure the bearings of the vehicle from different angles. However, most of the practical formation control tasks require relative static vehicle positions. Without relative motion, it is theoretically impossible for a vehicle to estimate its neighbors' positions from bearings. As a result, considering this limitation of the first approach, we will follow [5, 41] and adopt the second approach, which is to directly implement formation control laws based on bearing measurements.

Collision avoidance is a key issue in all kinds of formation control tasks. This issue is especially important in bearing-only formation control as inter-vehicle distances are unmeasurable and uncontrollable. In order to prove collision avoidance, we need to analyze the dynamics of the inter-vehicle distances in the absence of distance measurements. As will be shown later, the distance- and angle-dynamics of the formation are strongly coupled with each other. To rigorously prove the formation stability, we need to analyze the two dynamics simultaneously. Furthermore, asymptotic convergence of the angle-dynamics would be insufficient to analyze the distance-dynamics. It is necessary to prove exponential or finite-time convergence rate, which makes the problem more challenging.

Another challenging and interesting problem is the scale control of a formation. In fact, the scale of a formation is uncontrollable with bearing-only measurements, and inter-vehicle distance measurements are required to control the formation scale. One possible approach to formation scale control is to consider mixed bearing and distance constraints/measurements. We will leave formation scale control for future research. In our work, we will not consider distance measurements or constraints. Finally, global stability analysis of bearing-based

formation control undoubtedly is a challenging and meaningful research topic. When position measurements are available for formation control, a globally stable control law has been proposed in [29] to stabilize formations in arbitrary dimensions with fixed topology. However, when only bearing measurements are available, up to now control laws that guarantee global stability is only applicable to formations of three or four vehicles [5, 10, 8].

As a first step towards solving generic bearing-based formation control, the work in this chapter studies an important special case, cyclic formation, whose underlying information flow is described by an undirected cycle graph. In a cyclic formation, each vehicle has exactly two neighbors. The angle subtended at each vehicle by their two neighbors is pre-specified in the desired formation. The control objective is to steer each vehicle in the plane such that the angles converge to the pre-specified values. The main contributions of this chapter are summarized as below.

1) We propose a distributed control law that can stabilize cyclic formations merely using local bearing measurements. Compared to the existing work [5, 8], the proposed control law can handle cyclic formations with an *arbitrary* number of vehicles. In addition, this chapter does not make parallel rigidity assumptions [42, 9, 41] on the desired formation.

2) We prove that the proposed control law ensures local exponential or finite-time stability. The exponential or finite-time stability can be easily switched by tuning a parameter in the control law. The stability analysis is based on Lyapunov approaches and significantly different from those in [5, 8].

3) The dynamics of the inter-vehicle distances is analyzed in the absence of distance measurements. It is proved that the distance between any vehicles can neither approach zero nor infinity. Collision avoidance between any vehicles (no matter if they are neighbors or not) can be locally guaranteed.

If the vehicle number is larger than three, the shape of a cyclic formation would be indeterminate. To well define the shape of a formation of more than three vehicles, more complicated underlying graphs of the formation, such as

rigid graphs, are required. More complicated cases are out of the scope of this chapter and will be studied in the future.

The chapter is organized as follows. Notations and preliminaries are presented in Section 3.2. The control objective and proposed control law are given in Section 3.3. The stability analysis of the continuous and discontinuous cases are presented in Sections 3.4 and 3.5, respectively. Simulations are given in Section 3.6 to verify the effectiveness and robustness of the control law.

3.2 Notations and Preliminaries

3.2.1 Notations

The eigenvalues of a symmetric positive semi-definite matrix $A \in \mathbb{R}^{n\times n}$ are denoted as $0 \leq \lambda_1(A) \leq \lambda_2(A) \leq \cdots \leq \lambda_n(A)$. Let $\mathbf{1} = [1, \ldots, 1]^{\mathrm{T}} \in \mathbb{R}^n$, and I be the identity matrix with appropriate dimensions. Denote $[\,\cdot\,]_{ij}$ as the entry at the ith row and jth column of a matrix, and $[\,\cdot\,]_i$ as the ith entry of a vector. Let $|\cdot|$ be the absolute value of a real number, and $\|\cdot\|_p$ be the p-norm of a vector. For the sake of simplicity, we omit the subscript when $p = 2$, i.e., denoting $\|\cdot\|$ as the 2-norm. The null space of a matrix is denoted as $\mathrm{Null}\,(\cdot)$. The angle between two vectors is denoted as $\angle(\cdot, \cdot)$.

Given an arbitrary angle $\alpha \in \mathbb{R}$, the 2 by 2 rotation matrix

$$R(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

geometrically rotates a vector in $\mathbb{R}^2$ counterclockwise through an angle α about the origin. It is easy to see that for all nonzero $x \in \mathbb{R}^2$: (i) $x^{\mathrm{T}}R(\alpha)x > 0$ when $\alpha \in (-\pi/2, \pi/2) \pmod{2\pi}$; (ii) $x^{\mathrm{T}}R(\alpha)x = 0$ when $\alpha = \pm\pi/2 \pmod{2\pi}$; (iii) and $x^{\mathrm{T}}R(\alpha)x < 0$ when $\alpha \in (\pi/2, 3\pi/2) \pmod{2\pi}$. Moreover, we have $R^{-1}(\alpha) = R^{\mathrm{T}}(\alpha) = R(-\alpha)$ and $R(\alpha_1)R(\alpha_2) = R(\alpha_1 + \alpha_2)$ for arbitrary angles α_1 and α_2. Finally, for any $x \in \mathbb{R}^2$, denote $x^{\perp} = R(\pi/2)x$. Clearly $x^{\mathrm{T}}x^{\perp} = 0$.

3.2.2 Graph Theory

An *undirected graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a vertex set $\mathcal{V} = \{1, \ldots, n\}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where an edge is an unordered pair of distinct vertices. The undirected edge between vertices i and j is denoted as (i, j) or (j, i). If $(i, j) \in \mathcal{E}$, then i and j are called to be adjacent. A path from i to j in a graph is a sequence of distinct nodes starting with i and ending with j such that consecutive vertices are adjacent. If there is a path between any two vertices in $\mathcal{G}$, then $\mathcal{G}$ is said to be connected. The set of neighbors of vertex i is denoted as $\mathcal{N}_i = \{j \in \mathcal{V} : \ (i, j) \in \mathcal{E}\}$. An undirected *cycle* is a connected graph where every vertex has exactly two neighbors.

An *orientation* of an undirected graph is the assignment of a direction to each edge. An *oriented graph* is an undirected graph together with a particular orientation. A directed edge (i, j) in the oriented graph points from vertex i to vertex j. The *incidence matrix* E of an oriented graph is the $\{0, \pm 1\}$-matrix with rows indexed by edges and columns by vertices. More specifically, suppose (j, k) is the ith directed edge of the oriented graph. Then the entry of E in the ith row and kth column is 1, the one in the ith row and jth column is -1, and the others in the ith row are zero. Thus we have $E\mathbf{1} = 0$ by definition. Moreover, if the graph is connected, we have $\operatorname{rank}(E) = n - 1$ [50, Theorem 8.3.1] and hence $\operatorname{Null}(E) = \operatorname{span}\{\mathbf{1}\}$.

3.2.3 Nonsmooth Stability Analysis

Next we introduce some useful concepts and facts regarding discontinuous dynamic systems [46, 26, 98, 3, 30, 28].

Filippov Differential Inclusion

Consider the dynamic system

$$\dot{x}(t) = f\left(x(t)\right), \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a measurable and essentially locally bounded function. The Filippov differential inclusion [46] associated with the system (3.1) is

$$\dot{x} \in \mathcal{F}[f](x), \tag{3.2}$$

where $\mathcal{F}[f] : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ is defined by

$$\mathcal{F}[f](x) = \bigcap_{r>0} \bigcap_{\mu(S)=0} \overline{\text{co}}\left\{f\left(B(x,r) \setminus S\right)\right\}. \tag{3.3}$$

In (3.3), $\overline{\text{co}}$ denotes convex closure, $B(x,r)$ denotes the open ball centered at x with radius $r > 0$, and $\mu(S) = 0$ means that the Lebesgue measure of the set S is zero. The set-valued map $\mathcal{F}[f]$ associates each point x with a set. Note $\mathcal{F}[f](x)$ is multiple valued only if $f(x)$ is discontinuous at x.

A Filippov solution of (3.1) on $[0, t_1] \subset \mathbb{R}$ is defined as an absolutely continuous function $x : [0, t_1] \to \mathbb{R}^n$ that satisfies (3.2) for almost all $t \in [0, t_1]$. If $f(x)$ is measurable and essentially locally bounded, the existence of Filippov solutions can be guaranteed [30, Lemma 2.5] [28, Proposition 3] though the uniqueness cannot. The interested reader is referred to [28, p. 52] for the uniqueness conditions of Filippov solutions. A solution is called maximal if it cannot be extended forward in time. A set Ω is said to be weakly invariant (respectively strongly invariant) for (3.1), if for each $x(0) \in \Omega$, Ω contains at least one maximal solution (respectively all maximal solutions) of (3.1).

Generalized Gradient

Suppose $V : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz function. If $V(x)$ is differentiable at x, denote $\nabla V(x)$ as the gradient of $V(x)$ with respect to x. Let M_V be the set where $V(x)$ fails to be differentiable. The generalized gradient [26, 30, 28] of $V(x)$ is defined as

$$\partial V(x) = \text{co}\left\{\lim_{i \to +\infty} \nabla V(x_i) \mid x_i \to x,\, x_i \notin S \cup M_V\right\},$$

where co denotes convex hull and S is an arbitrary set of Lebesgue measure zero. The generalized gradient is a set-valued map. If $V(x)$ is continuously

differentiable at x, then $\partial V(x) = \{\nabla V(x)\}$.

Given any set $S \subseteq \mathbb{R}^n$, let $\mathrm{Ln} : 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$ be the set-valued map that associates S with the set of least-norm elements of $\overline{S}$. If S is convex, $\mathrm{Ln}(S)$ is singleton. In this chapter, we only apply Ln to generalized gradients which are always convex. For a locally Lipschitz function $V(x)$, $\mathrm{Ln}(\partial V) : \mathbb{R}^n \to \mathbb{R}^n$ is called the generalized gradient vector field. The following fact [28, Proposition 8]

$$\mathcal{F}\left[\mathrm{Ln}\left(\partial V(x)\right)\right] = \partial V(x) \tag{3.4}$$

will be very useful in our work. A point x is called a critical point if $0 \in \partial V(x)$. For a critical point x, it is obvious that $\mathrm{Ln}(\partial V(x)) = \{0\}$.

Set-valued Lie Derivative

The evolution of a locally Lipschitz function $V(x)$ along the solutions to the differential inclusion $\dot{x} \in \mathcal{F}[f](x)$ can be characterized by the set-valued Lie derivative [3, 30, 28], which is defined by

$$\widetilde{\mathcal{L}}_{\mathcal{F}} V(x) = \left\{\ell \in \mathbb{R} \mid \exists \xi \in \mathcal{F}[f](x),\ \forall \zeta \in \partial V(x),\ \xi^{\mathrm{T}}\zeta = \ell\right\}.$$

With a slight abuse of notation, we also denote $\widetilde{\mathcal{L}}_f V(x) = \widetilde{\mathcal{L}}_{\mathcal{F}} V(x)$. The set-valued Lie derivative may be empty. When $\widetilde{\mathcal{L}}_{\mathcal{F}} V(x) = \emptyset$, we take $\max \widetilde{\mathcal{L}}_{\mathcal{F}} V(x) = -\infty$ (see [3, 30, 28]).

A function $V : \mathbb{R}^n \to \mathbb{R}$ is called regular [28, p. 57] at x if the right directional derivative of $V(x)$ at x exists and coincides with the generalized directional derivative of $V(x)$ at x. Note a locally Lipschitz and convex function is regular. The following two lemmas are useful for proving the stability of discontinuous systems using nonsmooth Lyapunov functions. The next result can be found in [105, 3, 30].

Lemma 3.1. *Let $V : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz and regular function. Suppose the initial state is x_0 and let $\Omega(x_0)$ be the connected component of $\{x \in \mathbb{R}^n \mid V(x) \leq V(x_0)\}$ containing x_0. Assume the set $\Omega(x_0)$ is bounded.*

If $\max \widetilde{\mathcal{L}}_f V(x) \leq 0$ *or* $\widetilde{\mathcal{L}}_f V(x) = \emptyset$ *for all* $x \in \Omega(x_0)$*, then* $\Omega(x_0)$ *is strongly invariant for* (3.1)*. Let*

$$Z_{f,V} = \{x \in \mathbb{R}^n \mid 0 \in \widetilde{\mathcal{L}}_f V(x)\}. \tag{3.5}$$

Then any solution of (3.1) *starting from* x_0 *converges to the largest weakly invariant set* M *contained in* $\overline{Z}_{f,V} \cap \Omega(x_0)$*. Furthermore, if the set* M *is a finite collection of points, then the limit of all solutions starting from* x_0 *exists and equals one of them.*

The next result can be found in [98, 30].

Lemma 3.2. *Let* $V : \mathbb{R}^n \to \mathbb{R}$ *be a locally Lipschitz and regular function. Suppose the initial state is* x_0 *and let* S *be a compact and strongly invariant set for* (3.1)*. If* $\max \widetilde{\mathcal{L}}_f V(x) \leq -\kappa < 0$ *almost everywhere on* $S \backslash Z_{f,V}$*, then any solution of* (3.1) *starting at* $x_0 \in S$ *reaches* $Z_{f,V} \cap S$ *in finite time. The convergence time is upper bounded by* $\left(V(x_0) - \min_{x \in S} V(x)\right)/\kappa$*.*

3.2.4 Useful Lemmas

We next prove and introduce some useful results.

Lemma 3.3. *Let* $\mathcal{U} \triangleq \{x \in \mathbb{R}^n : \ x \neq 0$ *and nonzero entries of* x *are not of the same sign*$\}$*. Suppose* $A \in \mathbb{R}^{n \times n}$ *is a positive semi-definite matrix with* $\lambda_1(A) = 0$ *and* $\lambda_2(A) > 0$*. If* $\mathbf{1} = [1, \ldots, 1]^{\mathrm{T}} \in \mathbb{R}^n$ *is an eigenvector associated with the zero eigenvalue of* A*, then*

$$\inf_{x \in \mathcal{U}} \frac{x^{\mathrm{T}} A x}{x^{\mathrm{T}} x} = \frac{\lambda_2(A)}{n}.$$

Remark 3.1. *By the definition of* $\mathcal{U}$*, any* $x \in \mathcal{U}$ *should at least contain one positive entry and one negative entry. If the nonzero entries of* x *are all positive or negative, then* $x \notin \mathcal{U}$*.*

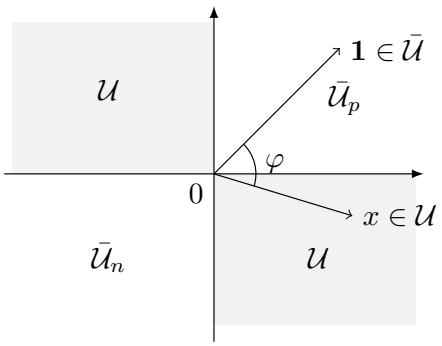


Figure 3.1: A 2D illustration for the proof of Lemma 3.3.

Proof. By orthogonally projecting $x \in \mathcal{U}$ to $\mathbf{1}$ and the orthogonal complement of $\mathbf{1}$, we decompose x as

$$x = x_0 + x_1,$$

where $x_0 \in \text{Null}\,(A)$ and $x_1 \perp \text{Null}\,(A)$. Let φ be the angle between $\mathbf{1}$ and x. Then we have $\|x_0\| = \cos\varphi\|x\|$ and $\|x_1\| = \sin\varphi\|x\|$. As a result,

$$\begin{aligned} x^{\mathrm{T}}Ax &= x_1^{\mathrm{T}}Ax_1 \\ &\geq \lambda_2(A)x_1^{\mathrm{T}}x_1 \\ &= \lambda_2(A)\sin^2\varphi\|x\|^2. \end{aligned} \tag{3.6}$$

By the definition of $\mathcal{U}$, any x in $\mathcal{U}$ would not be in span$\{\mathbf{1}\}$. That means $\varphi \neq 0$ or π and hence $\sin\varphi \neq 0$. We next identify the positive infimum of $\sin\varphi$.

Define $\bar{\mathcal{U}}_p = \{x \in \mathbb{R}^n : \ x \neq 0$ and nonzero entries of x are all positive$\}$ and $\bar{\mathcal{U}}_n = \{x \in \mathbb{R}^n : \ x \neq 0$ and nonzero entries of x are all negative$\}$. Let $\bar{\mathcal{U}} = \{0\} \cup \bar{\mathcal{U}}_p \cup \bar{\mathcal{U}}_n$. Clearly $\mathcal{U} \cup \bar{\mathcal{U}} = \mathbb{R}^n$. It is easy to see $\bar{\mathcal{U}}$ is a closed set and hence $\mathcal{U}$ is an open set. Figure 3.1 shows a 2D example to illustrate the above notations. Denote $\partial\mathcal{U}$ as the boundary of $\mathcal{U}$. The vector $\mathbf{1} \in \bar{\mathcal{U}}$ is isolated from any $x \in \mathcal{U}$ by $\partial\mathcal{U}$. Then we have $\inf_{x\in\mathcal{U}}\varphi = \min_{x\in\partial\mathcal{U}}\angle(x,\mathbf{1})$ and $\sup_{x\in\mathcal{U}}\varphi = \max_{x\in\partial\mathcal{U}}\angle(x,\mathbf{1})$. In fact, the boundary $\partial\mathcal{U}$ is formed by the hyper-planes $[x]_i = 0$ with $i \in \{1,\ldots,n\}$. Denote $p_i \in \mathbb{R}^n$ as the orthogonal projection of $\mathbf{1}$ on the hyper-plane $[x]_i = 0$. Then $\min_{x\in\partial\mathcal{U}}\angle(x,\mathbf{1}) = \angle(p_i,\mathbf{1})$ and $\max_{x\in\partial\mathcal{U}}\angle(x,\mathbf{1}) = \angle(-p_i,\mathbf{1})$. Note the ith entry of p_i is zero and the others are one. It can be calculated that $\cos\angle(\pm p_i,\mathbf{1}) = \pm\sqrt{n-1}/\sqrt{n}$ and hence

$\sin\angle(\pm p_i, \mathbf{1}) = 1/\sqrt{n}$. Thus

$$\inf_{x\in\mathcal{U}} \sin\varphi = \frac{1}{\sqrt{n}},$$

substituting which into (3.6) yields

$$\inf_{x\in\mathcal{U}} \frac{x^{\mathrm{T}}Ax}{x^{\mathrm{T}}x} = \frac{\lambda_2(A)}{n}.$$

□

Lemma 3.4. *Let $x(t)$ be a real positive scalar variable of $t \in [0, +\infty)$. Given any positive constants α and k, if the time derivative of $x(t)$ satisfies*

$$|\dot{x}(t)| \leq \alpha \exp\left(\int_0^t -\frac{k}{x(\tau)}\mathrm{d}\tau\right), \quad t \in [0, +\infty), \tag{3.7}$$

then $x(t)$ for all $t \in [0, +\infty)$ has a finite upper bound.

Proof. The proof consists of three steps.

Step 1: Prove the special case of $\alpha = 1$ and $k \in (0, 1)$.

The idea of the proof is to repeatedly utilize inequality (3.7) and the following inequality

$$x(t) \leq x(0) + \int_0^t |\dot{x}(\tau)|\mathrm{d}\tau. \tag{3.8}$$

First of all, because $x > 0$, we have $-k/x < 0$ and hence by (3.7) we have

$$|\dot{x}(t)| \leq \exp(0) = 1,$$

substituting which into (3.8) gives

$$x(t) \leq x(0) + \int_0^t 1\mathrm{d}\tau = t + c,$$

where $c = x(0)$. Substituting the above inequality back into (3.7) yields

$$|\dot{x}(t)| \leq \exp\left(\int_0^t -\frac{k}{\tau + c}\mathrm{d}\tau\right)$$

$$= \exp\left(-k \ln \frac{t+c}{c}\right)$$
$$= \left(\frac{c}{t+c}\right)^k.$$

Again by (3.8) we have

$$\begin{aligned} x(t) &\leq x(0) + \int_0^t \left(\frac{c}{\tau+c}\right)^k \mathrm{d}\tau \\ &= x(0) + \frac{c^k}{1-k}\left[(t+c)^{1-k} - c^{1-k}\right] \\ &< \frac{c^k}{1-k}(t+c)^{1-k}, \end{aligned} \tag{3.9}$$

where the last inequality uses the fact $c = x(0)$, $1-k < 1$ and hence $x(0) - c/(1-k) < 0$. Denote $\mu = (1-k)/c^k$. Substituting (3.9) into (3.7) gives

$$\begin{aligned} |\dot{x}(t)| &< \exp\left(\int_0^t -\mu k(\tau+c)^{k-1}\mathrm{d}\tau\right) \\ &= \exp\left(-\mu(t+c)^k + \mu c^k\right) \\ &= e^{1-k}e^{-\mu(t+c)^k}. \end{aligned}$$

One more by (3.8) we have

$$\begin{aligned} x(t) &\leq x(0) + e^{1-k}\int_0^t e^{-\mu(\tau+c)^k}\mathrm{d}\tau \\ &\leq x(0) + e^{1-k}\int_0^{+\infty} e^{-\mu(\tau+c)^k}\mathrm{d}\tau. \end{aligned} \tag{3.10}$$

Let $s = \mu(\tau+c)^k$. Then $\mathrm{d}\tau = (1/k)\mu^{-1/k}s^{1/k-1}\mathrm{d}s$. The above integral becomes

$$\begin{aligned} \int_0^{+\infty} e^{-\mu(\tau+c)^k}\mathrm{d}\tau &= \frac{1}{k}\mu^{-\frac{1}{k}}\int_{\mu c^k}^{+\infty} e^{-s}s^{\frac{1}{k}-1}\mathrm{d}s \\ &< \frac{1}{k}\mu^{-\frac{1}{k}}\int_0^{+\infty} e^{-s}s^{\frac{1}{k}-1}\mathrm{d}s \\ &= \frac{1}{k}\mu^{-\frac{1}{k}}\Gamma\left(\frac{1}{k}\right), \end{aligned} \tag{3.11}$$

where $\Gamma(1/k)$ is the well-known *Gamma function* and it has a positive value at $1/k > 0$. By substituting (3.11) into (3.10), we find a finite upper bound for $x(t)$

as

$$x(t) < \gamma = x(0) + \frac{1}{k}\mu^{-\frac{1}{k}} e^{1-k}\Gamma\left(\frac{1}{k}\right).$$

Step 2: Prove the special case of $\alpha = 1$ and $k \in [1, +\infty)$.

Consider a constant $k_0 \in (0, 1)$. Then $k > k_0$. Since $x(t) > 0$ for all $t \in [0, +\infty)$, we have

$$\int_0^t -\frac{k}{x(\tau)}\mathrm{d}\tau < \int_0^t -\frac{k_0}{x(\tau)}\mathrm{d}\tau,$$

which implies

$$|\dot{x}(t)| \leq \exp\left(\int_0^t -\frac{k}{x(\tau)}\mathrm{d}\tau\right) < \exp\left(\int_0^t -\frac{k_0}{x(\tau)}\mathrm{d}\tau\right).$$

Then by Step 1 there exists a finite upper bound γ such that $x(t) < \gamma$ for all $t \in [0, +\infty)$.

Step 3: Prove the generic case of $\alpha \in (0, +\infty)$ and $k \in (0, +\infty)$.

Note the combination of Step 1 and Step 2 indicates that $x(t)$ is bounded above if $\alpha = 1$ and $k \in (0, +\infty)$. When $\alpha \in (0, +\infty)$, inequality (3.7) can be rewritten as

$$\left|\left(\frac{x(t)}{\alpha}\right)'\right| \leq \exp\left(\int_0^t -\frac{k/\alpha}{x(\tau)/\alpha}\mathrm{d}\tau\right).$$

By Step 1 and Step 2, we know $x(t)/\alpha$ is bounded above, and so is $x(t)$. □

Lemma 3.5 ([112, Lemma 2])**.** *Let $x_1, \ldots, x_n \geq 0$. Given $p \in (0, 1]$, then*

$$\left(\sum_{i=1}^n x_i\right)^p \leq \sum_{i=1}^n x_i^p \leq n^{1-p}\left(\sum_{i=1}^n x_i\right)^p.$$

Lemma 3.6 ([58, Corollary 5.4.5])**.** *Let $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ be any two vector norms on $\mathbb{R}^n$. Then there exist finite positive constants C_m and C_M such that $C_m\|x\|_\alpha \leq \|x\|_\beta \leq C_M\|x\|_\alpha$ for all $x \in \mathbb{R}^n$.*

3.3 Problem Formulation

3.3.1 Control Objective

Consider n ($n \geq 3$) vehicles in $\mathbb{R}^2$. Denote the position of vehicle i as $z_i \in \mathbb{R}^2$. The dynamics of each vehicle is modeled as

$$\dot{z}_i = u_i,$$

where $u_i \in \mathbb{R}^2$ is the control input to be designed. This chapter focuses on cyclic formations (see Figure 3.2), whose underlying information flow is described by an undirected cycle graph. In a cyclic formation, each vehicle has exactly two neighbors. Denote θ_i as the angle at vehicle i subtended by its two neighbors (see Figure 3.2). The angle θ_i is specified as $\theta_i^* \in [0, 2\pi)$ in the desired formation. The desired angles $\{\theta_i^*\}_{i=1}^n$ should be feasible such that there exist $\{z_i\}_{i=1}^n$ ($z_i \neq z_j$ for $i \neq j$) to realize the desired formation. We make the following assumptions on $\{\theta_i^*\}_{i=1}^n$ and $\{z_i(0)\}_{i=1}^n$.

Assumption 3.1. *In the desired formation, $\theta_i^* \neq 0$ and $\theta_i^* \neq \pi$ for all $i \in \{1, \ldots, n\}$.*

Remark 3.2. *Assumption 3.1 means no three consecutive vehicles in the desired formation are collinear. The collinear case is a theoretical difficulty in many formation control problems (see, for example, [74, 36, 63, 8]). In practice, bearings are usually measured by optical sensors such as cameras. Hence vehicle i cannot measure the bearings of its two neighbors simultaneously when $\theta_i = 0$ due to line-of-sight occlusion. On the other hand, the field-of-view of a monocular camera is usually less than 180 degrees. Hence vehicle i cannot measure the bearings of its two neighbors simultaneously either when $\theta_i = \pi$ due to limited field-of-view. Thus Assumption 3.1 is reasonable from the practical point of view.*

Assumption 3.2. *In the initial formation, no two vehicles coincide with each other, i.e., $z_i(0) \neq z_j(0)$ for all $i \neq j$.*

The formation control objective is summarized as below.

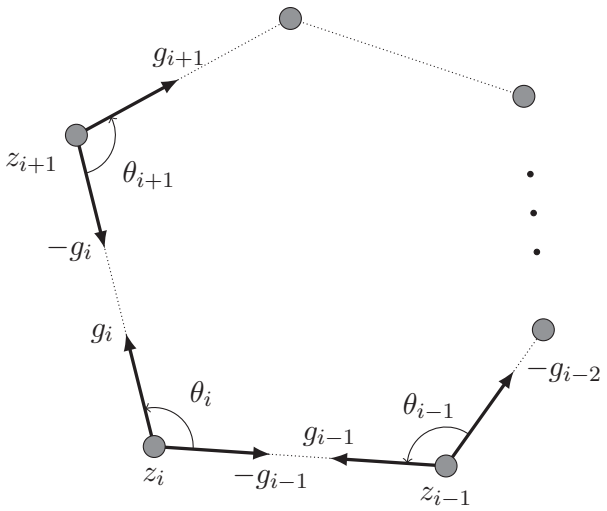


Figure 3.2: An illustration of cyclic formations.

Problem 3.1. *Under Assumptions 3.1 and 3.2, design control input u_i for vehicle i ($i = 1, \ldots, n$) based only on the local bearing measurements of its two neighbors such that the formation is steered from its initial position $\{z_i(0)\}_{i=1}^n$ to a finite final position $\{z_i(t_f)\}_{i=1}^n$ where $\theta_i(t_f) = \theta_i^*$. The final converged time t_f can be either infinite or finite. During the formation evolution, collision avoidance between any vehicles should be guaranteed.*

3.3.2 Control Law Design

We next define some notations and then propose our formation control law. In the cyclic formation, we can have $\mathcal{N}_i = \{i-1, i+1\}$ for $i \in \{1, \ldots, n\}$ by indexing the vehicles properly (see Figure 3.2). Then vehicle i can measure the bearings of vehicles $i-1$ and $i+1$. The indices $i+1$ and $i-1$ are taken modulo n. Denote

$$e_i \triangleq z_{i+1} - z_i \tag{3.12}$$

as the edge vector pointing from vehicle i to vehicle $i+1$. Then the unit-length vector

$$g_i \triangleq \frac{e_i}{\|e_i\|}$$

characterizes the relative bearing between vehicles $i+1$ and i (see Figure 3.2). Thus the bearings measured by vehicle i include g_i and $-g_{i-1}$. The control input

u_i will be designed as a function of g_i and $-g_{i-1}$.

The angle $\theta_i \in [0, 2\pi)$ is defined in the following way (see Figure 3.2): rotating $-g_{i-1}$ counterclockwise through an angle θ_i about vehicle i yields g_i, which can be expressed as

$$g_i = R(\theta_i)(-g_{i-1}).$$

When θ_i is defined in the above way, the angles θ_i and θ_{i+1} are on the same side of e_i for all $i \in \{1, \ldots, n\}$. As a result, the quantity $\sum_{i=1}^{n} \theta_i$ is invariant to the positions of the vehicles because the sum of the interior or exterior angles of a polygon is constant. Thus if $\sum_{i=1}^{n} \theta_i(0) = \sum_{i=1}^{n} \theta_i^*$, then $\sum_{i=1}^{n} \theta_i(t) \equiv \sum_{i=1}^{n} \theta_i^*$.

The angle error for vehicle i, which will be used for feedback control, is defined as

$$\varepsilon_i \triangleq \cos\theta_i - \cos\theta_i^* = -g_i^{\mathrm{T}} g_{i-1} - \cos\theta_i^*. \tag{3.13}$$

The nonlinear control law for vehicle i is designed as

$$u_i = \mathrm{sgn}(\varepsilon_i)|\varepsilon_i|^a (g_i - g_{i-1}), \tag{3.14}$$

where $a \in [0, 1]$ and $\mathrm{sgn}(\varepsilon_i)$ is defined by

$$\mathrm{sgn}(\varepsilon_i) = \begin{cases} 1 & \text{if } \varepsilon_i > 0 \\ 0 & \text{if } \varepsilon_i = 0 \\ -1 & \text{if } \varepsilon_i < 0 \end{cases}.$$

In the special case of $a = 1$, control law (3.14) becomes $u_i = \varepsilon_i(g_i - g_{i-1})$ because $\mathrm{sgn}(\varepsilon_i)|\varepsilon_i| = \varepsilon_i$. In the special case of $a = 0$, control law (3.14) becomes $u_i = \mathrm{sgn}(\varepsilon_i)(g_i - g_{i-1})$.

Remark 3.3. *It should be noted that* $\mathrm{sgn}(\varepsilon_i)|\varepsilon_i|^a$ *is* continuous *in* ε_i *for* $a \in (0, 1]$. *That is because* $\lim_{\varepsilon_i \to 0^+} \mathrm{sgn}(\varepsilon_i)|\varepsilon_i|^a = \lim_{\varepsilon_i \to 0^-} \mathrm{sgn}(\varepsilon_i)|\varepsilon_i|^a = 0$. *Therefore, the control law is continuous in* ε_i *if* $a \in (0, 1]$. *But the control law is discontinuous if* $a = 0$.

Clearly (3.14) is a *distributed* control law as it only relies on the bearings of vehicle i's neighbors. Moreover, although g_i and g_{i-1} in (3.14) are expressed in a global coordinate frame, the control law can be implemented based on the local bearings measured in the local coordinate frame of vehicle i. To see that, denote R_i as the rotation transformation from a global frame to the local frame of vehicle i. Then the bearings of vehicles $i-1$ and $i+1$ measured in the local frame are $R_i(-g_{i-1})$ and $R_i g_i$, respectively. Note ε_i defined in (3.13) is invariant to R_i. Then substituting $R_i(-g_{i-1})$ and $R_i g_i$ into (3.14) gives $u_{i,\text{local}} = \text{sgn}(\varepsilon_i)|\varepsilon_i|^a R_i(g_i - g_{i-1})$. Converting $u_{i,\text{local}}$ into the global frame would yield the same control input value given by (3.14).

As will be shown later, control law (3.14) ensures local exponential stability if $a = 1$, and local finite-time stability if $a \in [0, 1)$. Loosely speaking, finite-time stability means ε_i for all i converges to zero in finite time. See [7] or [54, Section 4.6] for a formal definition of finite-time stability of nonlinear systems. Besides fast convergence, finite-time stability can also bring benefits such as disturbance rejection and robustness against uncertainties [7, 115, 22].

3.4 Stability Analysis of the Continuous Case

If $a \in (0, 1]$, the control law (3.14) is continuous in ε, and the formation stability can be proved based on traditional Lyapunov approaches. We first propose a continuously differentiable Lyapunov function and then show its time derivative under control law (3.14) is non-positive.

3.4.1 Lyapunov Function

Denote $\varepsilon = [\varepsilon_1, \ldots, \varepsilon_n]^{\mathrm{T}} \in \mathbb{R}^n$ and $z = [z_1^{\mathrm{T}}, \ldots, z_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2n}$. It is straightforward to see from (3.14) that $\varepsilon = 0$ implies $\dot{z} = 0$ and then $\dot{\varepsilon} = 0$. Hence $\varepsilon = 0$ is an equilibrium of the ε-dynamics. Consider the Lyapunov function

$$V(\varepsilon) = \frac{1}{a+1} \sum_{i=1}^{n} |\varepsilon_i|^{a+1}.$$

Clearly V is positive definite with respect to $\varepsilon = 0$. In the special case of $a = 1$, we have $V = 1/2\varepsilon^{\mathrm{T}}\varepsilon$, which is a quadratic function of ε.

We next show V is *continuously differentiable* in ε. (i) If $\varepsilon_i > 0$, $\frac{\partial|\varepsilon_i|^{a+1}}{\partial\varepsilon_i} = \frac{\partial\varepsilon_i^{a+1}}{\partial\varepsilon_i} = (a+1)\varepsilon_i^a = (a+1)\mathrm{sgn}(\varepsilon_i)|\varepsilon_i|^a$ and hence $\lim_{\varepsilon_i\to 0^+}\frac{\partial|\varepsilon_i|^{a+1}}{\partial\varepsilon_i} = 0$. (ii) If $\varepsilon_i < 0$, $\frac{\partial|\varepsilon_i|^{a+1}}{\partial\varepsilon_i} = \frac{\partial(-\varepsilon_i)^{a+1}}{\partial\varepsilon_i} = -(a+1)(-\varepsilon_i)^a = (a+1)\mathrm{sgn}(\varepsilon_i)|\varepsilon_i|^a$ and hence $\lim_{\varepsilon_i\to 0^-}\frac{\partial|\varepsilon_i|^{a+1}}{\partial\varepsilon_i} = 0$. From (i) and (ii) we have

$$\frac{\partial|\varepsilon_i|^{a+1}}{\partial\varepsilon_i} = (a+1)\mathrm{sgn}(\varepsilon_i)|\varepsilon_i|^a, \quad \forall\varepsilon_i\in\mathbb{R}. \tag{3.15}$$

Note $\mathrm{sgn}(\varepsilon_i)|\varepsilon_i|^a$ is continuous in ε_i for $a\in(0,1]$. Thus $|\varepsilon_i|^{a+1}$ is continuously differentiable in ε_i. As a result, V is continuously differentiable in ε.

3.4.2 Time Derivative of V

We next derive the time derivative of V under control law (3.14) and show it is non-positive. For the sake of simplicity, denote

$$\sigma_i \triangleq \mathrm{sgn}(\varepsilon_i)|\varepsilon_i|^a$$

and $\sigma = [\sigma_1,\ldots,\sigma_n]^{\mathrm{T}}\in\mathbb{R}^n$. Then control law (3.14) can be rewritten as $\dot{z}_i = \sigma_i(g_i - g_{i-1})$, and (3.15) becomes $\partial|\varepsilon_i|^{a+1}/\partial\varepsilon_i = (a+1)\sigma_i$. The time derivative of V is

$$\begin{aligned}
\dot{V} &= \frac{1}{a+1}\sum_{i=1}^{n}\frac{\partial|\varepsilon_i|^{a+1}}{\partial\varepsilon_i}\dot{\varepsilon}_i \\
&= \sum_{i=1}^{n}\sigma_i\dot{\varepsilon}_i \qquad \text{(By (3.15))} \\
&= \sum_{i=1}^{n}\sigma_i(-g_i^{\mathrm{T}}\dot{g}_{i-1} - g_{i-1}^{\mathrm{T}}\dot{g}_i) \qquad \text{(By (3.13))} \\
&= \sum_{i=1}^{n}\sigma_i(-g_i^{\mathrm{T}}\dot{g}_{i-1}) + \sum_{i=1}^{n}\sigma_i(-g_{i-1}^{\mathrm{T}}\dot{g}_i) \\
&= \sum_{i=1}^{n}\sigma_{i+1}(-g_{i+1}^{\mathrm{T}}\dot{g}_i) + \sum_{i=1}^{n}\sigma_i(-g_{i-1}^{\mathrm{T}}\dot{g}_i) \\
&= -\sum_{i=1}^{n}(\sigma_{i+1}g_{i+1} + \sigma_i g_{i-1})^{\mathrm{T}}\dot{g}_i.
\end{aligned} \tag{3.16}$$

Since $g_i = e_i/\|e_i\|$, we have

$$\dot{g}_i = \frac{\dot{e}_i}{\|e_i\|} - \frac{e_i}{\|e_i\|^2}\frac{\mathrm{d}\|e_i\|}{\mathrm{d}t} = \frac{1}{\|e_i\|}\left(I - \frac{e_i}{\|e_i\|}\frac{e_i^{\mathrm{T}}}{\|e_i\|}\right)\dot{e}_i \triangleq \frac{1}{\|e_i\|}P_i\dot{e}_i, \tag{3.17}$$

where $P_i = I - g_i g_i^{\mathrm{T}}$. Matrix P_i plays an important role in the stability analysis in this chapter. Geometrically P_i is an *orthogonal projection matrix* which can orthogonally project any vector onto the orthogonal compliment of g_i. The algebraic properties of P_i are listed below.

Lemma 3.7. *Matrix P_i satisfies:*

(i) $P_i^{\mathrm{T}} = P_i$ *and* $P_i^2 = P_i$.

(ii) P_i *is positive semi-definite.*

(iii) $\mathrm{Null}\,(P_i) = \mathrm{span}\{g_i\}$.

Proof. (i) The two properties are trivial to check.

(ii) For any $x \in \mathbb{R}^2$, since $P_i^2 = P_i$ and $P_i^{\mathrm{T}} = P_i$, we have $x^{\mathrm{T}} P_i x = x^{\mathrm{T}} P_i^{\mathrm{T}} P_i x = \|P_i x\|^2 \geq 0$.

(iii) First, it is easy to see $P_i g_i = 0$ and hence $g_i \in \mathrm{Null}\,(P_i)$. Second, for any $x \in \mathbb{R}^2$, we have $P_i x = x - (g_i^{\mathrm{T}} x) g_i$. Clearly $P_i x = 0$ only if x is parallel to g_i. Thus $\mathrm{Null}\,(P_i) = \mathrm{span}\{g_i\}$.

□

Furthermore, from (3.12) and control law (3.14), we have

$$\begin{aligned} \dot{e}_i &= \dot{z}_{i+1} - \dot{z}_i \\ &= \sigma_{i+1} g_{i+1} + \sigma_i g_{i-1} - (\sigma_{i+1} + \sigma_i) g_i. \end{aligned} \tag{3.18}$$

Because $P_i g_i = 0$, substituting the above $\dot{e}_i$ back into (3.17) gives

$$\dot{g}_i = \frac{1}{\|e_i\|} P_i (\sigma_{i+1} g_{i+1} + \sigma_i g_{i-1}).$$

Substituting the above $\dot{g}_i$ back into (3.16) yields

$$\dot{V} = -\sum_{i=1}^{n} \frac{1}{\|e_i\|} (\sigma_{i+1} g_{i+1} + \sigma_i g_{i-1})^{\mathrm{T}} P_i (\sigma_{i+1} g_{i+1} + \sigma_i g_{i-1}) \leq 0. \tag{3.19}$$

Now we can claim the equilibrium $\varepsilon = 0$ is at least Lyapunov stable.

We next derive the matrix form of (3.19), which will be useful to prove exponential and finite-time stability. To do that, we need the following lemma.

Lemma 3.8. *Let $g_i^\perp = R(\pi/2)g_i$. It is obvious that $\|g_i^\perp\| = 1$ and $(g_i^\perp)^{\mathrm{T}} g_i = 0$. Furthermore,*

(i) $P_i = g_i^\perp (g_i^\perp)^{\mathrm{T}}$.

(ii) $(g_i^\perp)^{\mathrm{T}} g_j = -(g_j^\perp)^{\mathrm{T}} g_i$ *for all* $i \neq j$.

(iii) $(g_i^\perp)^{\mathrm{T}} g_{i-1} = \sin\theta_i$. *As a result,* $(g_i^\perp)^{\mathrm{T}} g_{i-1} > 0$ *if* $\theta_i \in (0, \pi)$; *and* $(g_i^\perp)^{\mathrm{T}} g_{i-1} < 0$ *if* $\theta_i \in (\pi, 2\pi)$.

Proof. (i) Denote $G_i = [g_i, g_i^\perp] \in \mathbb{R}^{2\times 2}$. It is easy to examine that G_i is an orthogonal matrix satisfying $G_i^{\mathrm{T}} G_i = G_i G_i^{\mathrm{T}} = I$. Hence we have

$$g_i g_i^{\mathrm{T}} + g_i^\perp (g_i^\perp)^{\mathrm{T}} = G_i G_i^{\mathrm{T}} = I.$$

Thus $g_i^\perp (g_i^\perp)^{\mathrm{T}} = I - g_i g_i^{\mathrm{T}} = P_i$.

(ii) $(g_i^\perp)^{\mathrm{T}} g_j = g_i^{\mathrm{T}} R^{\mathrm{T}}(\pi/2) g_j = g_i^{\mathrm{T}} R(-\pi/2) g_j = g_i^{\mathrm{T}} R(-\pi) R(\pi/2) g_j = g_i^{\mathrm{T}} R(-\pi) g_j^\perp = g_i^{\mathrm{T}}(-I) g_j^\perp = -(g_j^\perp)^{\mathrm{T}} g_i$.

(iii) By the definition of θ_i, we have $g_i = R(\theta_i)(-g_{i-1})$ and hence $g_{i-1} = -R(-\theta_i) g_i$. Then

$$\begin{aligned}
(g_i^\perp)^{\mathrm{T}} g_{i-1} &= -g_i^{\mathrm{T}} R\left(-\frac{\pi}{2}\right) R(-\theta_i) g_i \\
&= -g_i^{\mathrm{T}} R\left(-\frac{\pi}{2} - \theta_i\right) g_i \\
&= -\|g_i\| \left\| R\left(-\frac{\pi}{2} - \theta_i\right) g_i \right\| \cos\left(-\frac{\pi}{2} - \theta_i\right) \\
&= \sin\theta_i.
\end{aligned}$$

Then it is straightforward to have the rest results in Lemma 3.8(iii). □

Substituting $P_i = g_i^\perp (g_i^\perp)^{\mathrm{T}}$ as shown in Lemma 3.8(i) into (3.19) yields

$$\begin{aligned}
\dot{V} &= -\sum_{i=1}^{n} \frac{1}{\|e_i\|} \left((g_i^\perp)^{\mathrm{T}} (\sigma_{i+1} g_{i+1} + \sigma_i g_{i-1}) \right)^2 \\
&\leq -\frac{1}{\sum_{i=1}^{n} \|e_i\|} \sum_{i=1}^{n} \left(\sigma_{i+1} (g_i^\perp)^{\mathrm{T}} g_{i+1} + \sigma_i (g_i^\perp)^{\mathrm{T}} g_{i-1} \right)^2 \\
&= -\frac{1}{\sum_{i=1}^{n} \|e_i\|} \|\xi\|^2, \qquad (3.20)
\end{aligned}$$

where

$$
\begin{aligned}
\xi &= \begin{bmatrix} \sigma_2(g_1^\perp)^{\mathrm{T}} g_2 + \sigma_1 (g_1^\perp)^{\mathrm{T}} g_n \\ \vdots \\ \sigma_1 (g_n^\perp)^{\mathrm{T}} g_1 + \sigma_n (g_n^\perp)^{\mathrm{T}} g_{n-1} \end{bmatrix} \\
&= \begin{bmatrix} (g_1^\perp)^{\mathrm{T}} g_n & (g_1^\perp)^{\mathrm{T}} g_2 & 0 & \dots & 0 \\ 0 & (g_2^\perp)^{\mathrm{T}} g_1 & (g_2^\perp)^{\mathrm{T}} g_3 & \dots & 0 \\ 0 & 0 & (g_3^\perp)^{\mathrm{T}} g_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (g_n^\perp)^{\mathrm{T}} g_1 & 0 & \dots & 0 & (g_n^\perp)^{\mathrm{T}} g_{n-1} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_n \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & \dots & 0 & 1 \end{bmatrix}}_{E \in \mathbb{R}^{n \times n}} \times \\
&\qquad \underbrace{\begin{bmatrix} (g_1^\perp)^{\mathrm{T}} g_n & 0 & 0 & \dots & 0 \\ 0 & (g_2^\perp)^{\mathrm{T}} g_1 & 0 & \dots & 0 \\ 0 & 0 & (g_3^\perp)^{\mathrm{T}} g_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & (g_n^\perp)^{\mathrm{T}} g_{n-1} \end{bmatrix}}_{D \in \mathbb{R}^{n \times n}} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_n \end{bmatrix}. \qquad (3.21)
\end{aligned}
$$

The last equality above uses the fact that $(g_i^\perp)^{\mathrm{T}} g_{i-1} = -(g_{i-1}^\perp)^{\mathrm{T}} g_i$ given by Lemma 3.8(ii). Substituting (3.21) into (3.20) yields

$$
\dot{V} \leq -\frac{1}{\sum_{i=1}^n \|e_i\|} \sigma^{\mathrm{T}} D^{\mathrm{T}} E^{\mathrm{T}} E D \sigma. \qquad (3.22)
$$

Inequality (3.22) is very important and will be used to prove the exponential and finite-time stability of the control law in the next section. We would like to mention that D is a diagonal matrix and $E\mathbf{1} = 0$. It can be easily checked that E is the incidence matrix of an oriented cycle graph. Thus we have $\mathrm{rank}(E) = n-1$ [50, Theorem 8.3.1] and hence $\mathrm{Null}\,(E^{\mathrm{T}}E) = \mathrm{Null}\,(E) = \mathrm{span}\{\mathbf{1}\}$.

3.4.3 Exponential and Finite-time Stability Analysis

Based on inequality (3.22) obtained in the previous section, we next prove the exponential and finite-time stability of control law (3.14). The proof of our main result consists of three relatively independent steps, each of which will be summarized as a proposition. As aforementioned, the inter-vehicle distance dynamics is a theoretical difficulty. We will particularly analyze this issue in the second and third steps. More specifically, the second step shows that the distance between any two vehicles cannot approach infinity; the third step proves that the distance between any two vehicles (no matter if they neighbors or not) cannot approach zero during formation evolution.

At this point, it is still unclear whether any vehicles may collide with each other during formation evolution. Nevertheless, we can always assume there is a "collision time" $T_c \in (0, +\infty)$, at which at least two vehicles collide with each other. Note T_c could be infinity. If T_c is infinity, there would be no collision between any vehicles during the whole formation evolution. In fact, we will later prove T_c to be infinity given sufficiently small initial error ε_0. But at this point we are only able to claim that inequality (3.22) is valid only for $t \in [0, T_c)$.

Denote $\Omega(c) \triangleq \{\varepsilon \in \mathbb{R}^n : V(\varepsilon) \leq c\}$ with $c > 0$ as the level set of $V(\varepsilon)$. Note V can be written as $V = 1/(a+1)\|\varepsilon\|_{a+1}^{a+1}$ where $\|\cdot\|_{a+1}$ is the $(a+1)$-norm. Hence $\Omega(c)$ is compact [58, Corollary 5.4.8]. Because $\dot{V} \leq 0$ as shown in (3.22), the level set $\Omega\left(V(\varepsilon_0)\right)$ is also positively invariant with respect to (3.14).

Proposition 3.1. *Under Assumptions 3.1 and 3.2, if the initial error ε_0 is sufficiently small, then there exists a positive constant K such that*

$$\dot{V} \leq -\frac{K}{\sum_{i=1}^{n} \|e_i\|} V^{\frac{2a}{a+1}}, \quad \forall t \in [0, T_c). \tag{3.23}$$

Proof. Suppose $\varepsilon \neq 0 \Leftrightarrow \sigma \neq 0$. Rewrite $\sigma^{\mathrm{T}} D^{\mathrm{T}} E^{\mathrm{T}} E D \sigma$ on the right hand side of (3.22) as

$$\sigma^{\mathrm{T}} D^{\mathrm{T}} E^{\mathrm{T}} E D \sigma = \underbrace{\left(\frac{\sigma^{\mathrm{T}} D^{\mathrm{T}} E^{\mathrm{T}} E D \sigma}{\sigma^{\mathrm{T}} D^{\mathrm{T}} D \sigma}\right)}_{\text{term 1}} \underbrace{\left(\frac{\sigma^{\mathrm{T}} D^{\mathrm{T}} D \sigma}{V^{\frac{2a}{a+1}}}\right)}_{\text{term 2}} V^{\frac{2a}{a+1}}. \tag{3.24}$$

Step 1: analyze term 2 in (3.24). At the equilibrium point $\varepsilon = 0$ (i.e., $\theta_i = \theta_i^*$ for all i), we have $[D]_{ii} = (g_i^{\perp})^{\mathrm{T}} g_{i-1} \neq 0$ because $\theta_i^* \neq 0$ or π as stated in Assumption 3.1. Thus by continuity we have $[D]_{ii} \neq 0$ for every point in $\Omega(V(\varepsilon_0))$ if ε_0 is sufficiently small. Then $D^{\mathrm{T}}D = D^2$ is positive definite and hence $\lambda_1(D^{\mathrm{T}}D) > 0$ for all $\varepsilon \in \Omega(V(\varepsilon_0))$. Since $\Omega(V(\varepsilon_0))$ is compact, there exists a lower bound $\underline{\lambda}_1(D^{\mathrm{T}}D) > 0$ such that $\lambda_1(D^{\mathrm{T}}D) \geq \underline{\lambda}_1(D^{\mathrm{T}}D)$ and consequently

$$\sigma^{\mathrm{T}} D^{\mathrm{T}} D \sigma \geq \underline{\lambda}_1(D^{\mathrm{T}}D)\sigma^{\mathrm{T}}\sigma \tag{3.25}$$

for all $\varepsilon \in \Omega(V(\varepsilon_0))$. In addition, since $2a/(a+1) \in (0, 1]$, we have

$$\begin{aligned} V^{\frac{2a}{a+1}} &= \left(\frac{1}{a+1}\right)^{\frac{2a}{a+1}} \left(\sum_{i=1}^{n} |\varepsilon_i|^{a+1}\right)^{\frac{2a}{a+1}} \\ &\leq \left(\frac{1}{a+1}\right)^{\frac{2a}{a+1}} \sum_{i=1}^{n} |\varepsilon_i|^{2a} \quad \text{(By Lemma 3.5)} \\ &= \left(\frac{1}{a+1}\right)^{\frac{2a}{a+1}} \sum_{i=1}^{n} \sigma_i^2 \quad \text{(By } |\varepsilon_i|^{2a} = \sigma_i^2\text{)} \\ &= \left(\frac{1}{a+1}\right)^{\frac{2a}{a+1}} \sigma^{\mathrm{T}}\sigma. \end{aligned} \tag{3.26}$$

Thus (3.25) and (3.26) imply

$$\begin{aligned} \frac{\sigma^{\mathrm{T}} D^{\mathrm{T}} D \sigma}{V^{\frac{2a}{a+1}}} &\geq \frac{\underline{\lambda}_1(D^{\mathrm{T}}D)\sigma^{\mathrm{T}}\sigma}{\left(\frac{1}{a+1}\right)^{\frac{2a}{a+1}} \sigma^{\mathrm{T}}\sigma} \\ &= (a+1)^{\frac{2a}{a+1}} \underline{\lambda}_1(D^{\mathrm{T}}D) \end{aligned} \tag{3.27}$$

for all $\varepsilon \in \Omega(V(\varepsilon_0)) \setminus \{0\}$.

Step 2: analyze term 1 in (3.24). Define

$$w_i = \frac{\cos\theta_i - \cos\theta_i^*}{\theta_i - \theta_i^*}.$$

Note $\lim_{\theta_i \to \theta_i^*} w_i = -\sin\theta_i^*$ by L'Hôpital's rule. Thus w_i is well defined even if $\theta_i - \theta_i^* = 0$. Denote $\delta_i \triangleq \theta_i - \theta_i^*$ and recall $\varepsilon_i = \cos\theta_i - \cos\theta_i^*$. Then we have

$$\left.\begin{array}{l} DW\text{: same sign} \\ \delta\text{: not same sign} \end{array}\right\} \Longrightarrow DW\delta = D\varepsilon\text{: not same sign} \Longrightarrow D\sigma\text{: not same sign} \Longrightarrow D\sigma \in \mathcal{U}$$

Figure 3.3: Illustrate how to obtain $D\sigma \in \mathcal{U}$.

$\varepsilon_i = w_i\delta_i$, whose matrix form is

$$\varepsilon = W\delta,$$

where $W = \mathrm{diag}\{w_1, \ldots, w_n\} \in \mathbb{R}^{n\times n}$ and $\delta = [\delta_1, \ldots, \delta_n]^{\mathrm{T}} \in \mathbb{R}^n$. On one hand, when ε_0 is sufficiently small, we have θ_i is sufficiently close to θ_i^* such that both θ_i and θ_i^* are in either $(0, \pi)$ or $(\pi, 2\pi)$ for all $\varepsilon \in \Omega(V(\varepsilon_0))$. It can be examined that $w_i < 0$ when $\theta_i, \theta_i^* \in (0, \pi)$, and $w_i > 0$ when $\theta_i, \theta_i^* \in (\pi, 2\pi)$. On the other hand, $[D]_{ii} = (g_i^{\perp})^{\mathrm{T}} g_{i-1} > 0$ when $\theta_i \in (0, \pi)$, and $[D]_{ii} = (g_i^{\perp})^{\mathrm{T}} g_{i-1} < 0$ when $\theta_i \in (\pi, 2\pi)$ as shown in Lemma 3.8(iii). Thus we always have

$$[D]_{ii} w_i < 0$$

for all $i \in \{1, \ldots, n\}$ and all $\varepsilon \in \Omega(V(\varepsilon_0))$, which means the diagonal entries of DW *are of the same sign*. However, because $\sum_i^n \theta_i \equiv \sum_i^n \theta_i^* \Leftrightarrow \sum_{i=1}^n \delta_i = 0$, the nonzero entries in δ *are not of the same sign*. Hence the nonzero entries of $D\varepsilon = DW\delta$ *are not of the same sign*. Furthermore, because σ_i has the same sign as ε_i, the nonzero entries of $D\sigma$ *are not of the same sign* either. Thus $D\sigma \in \mathcal{U}$ where $\mathcal{U}$ is defined in Lemma 3.3. The above arguments are illustrated intuitively in Figure 3.3. Recall $\mathrm{Null}\,(E^{\mathrm{T}}E) = \mathrm{Null}\,(E) = \mathrm{span}\{\mathbf{1}\}$. Therefore, by Lemma 3.3 we have

$$\frac{\sigma^{\mathrm{T}} D^{\mathrm{T}} E^{\mathrm{T}} E D\sigma}{\sigma^{\mathrm{T}} D^{\mathrm{T}} D\sigma} > \frac{\lambda_2(E^{\mathrm{T}}E)}{n}. \tag{3.28}$$

Step 3: substituting (3.27) and (3.28) into (3.24) yields

$$\sigma^{\mathrm{T}} D^{\mathrm{T}} E^{\mathrm{T}} E D\sigma \geq \underbrace{\frac{\lambda_2(E^{\mathrm{T}}E)}{n}(a+1)^{\frac{2a}{a+1}} \underline{\lambda}_1(D^{\mathrm{T}}D)}_{K} V^{\frac{2a}{a+1}}. \tag{3.29}$$

Then (3.23) can be obtained by substituting (3.29) into (3.22). Note (3.22) holds for all $t \in [0, T_c)$, and so does (3.23). □

Proposition 3.1 requires ε_0 to be sufficiently small, but does not give any explicit condition of ε_0. In order to determine the region of convergence, we next give a *sufficient condition* of ε_0 which ensures the validity of Proposition 3.1. The proof of Proposition 3.1 requires ε_0 to be sufficiently small such that (i) $[D]_{ii} \neq 0$ and (ii) both θ_i and θ_i^* are in either $(0, \pi)$ or $(\pi, 2\pi)$ for all $\varepsilon \in \Omega(V(\varepsilon_0))$. Since $[D]_{ii} = 0$ if and only if $\theta_i = 0$ or π, condition (ii) implies condition (i). Denote $\Delta_i = \min\{\theta_i^*, |\theta_i^* - \pi|, 2\pi - \theta_i^*\}$ and $\bar{\varepsilon}_i = \min\{|\cos(\theta_i^* + \Delta_i) - \cos\theta_i^*|, |\cos(\theta_i^* - \Delta_i) - \cos\theta_i^*|\}$. Then we have the following sufficient condition. *If ε_0 satisfies*

$$V(\varepsilon_0) < \frac{1}{a+1} \min_i \bar{\varepsilon}_i^{a+1}, \tag{3.30}$$

then condition (ii) can be satisfied and hence Proposition 3.1 is valid. To see that, for any $j \in \{1, \ldots, n\}$, we have $\frac{1}{a+1}|\varepsilon_j(t)|^{a+1} \leq \frac{1}{a+1}\sum_{i=1}^n |\varepsilon_i(t)|^{a+1} = V(\varepsilon(t)) \leq V(\varepsilon_0) < \frac{1}{a+1}\min_i \bar{\varepsilon}_i^{a+1} \leq \frac{1}{a+1}\bar{\varepsilon}_j^{a+1}$. Thus $|\varepsilon_j(t)| < \bar{\varepsilon}_j$ for all $t \in [0, T_c)$. Since the cosine function is monotone in $(0, \pi)$ or $(\pi, 2\pi)$, we have $|\varepsilon_j(t)| < \bar{\varepsilon}_j \implies |\theta_i(t) - \theta_i^*| < \Delta_i$ and hence condition (ii) is valid. It should be noted that $\Delta_i \neq 0$ because $\theta_i^* \neq 0$ or π. Therefore, $\bar{\varepsilon}_i > 0$ and hence the set of ε_0 that satisfies (3.67) is always nonempty.

Since the inter-vehicle distances are not controlled directly, we cannot simply rule out the possibility that $\sum_{i=1}^n \|e_i\|$ in (3.23) may go to infinity. Based on Proposition 3.1, we next further prove $\sum_{i=1}^n \|e_i\|$ is bounded above by a finite positive constant.

Proposition 3.2 (Finite Inter-vehicle Distance)**.** *Under Assumptions 3.1 and 3.2, if* (3.23) *holds and the initial error ε_0 is sufficiently small such that $V(\varepsilon_0) \leq 1$, then there exists a finite constant $\gamma > 0$ such that*

$$\sum_{i=1}^n \|e_i(t)\| \leq \gamma, \quad \forall t \in [0, T_c),$$

which holds even if $T_c = +\infty$. As a result, (3.23) *implies*

$$\dot{V} \leq -\frac{K}{\gamma} V^{\frac{2a}{1+a}}, \quad \forall t \in [0, T_c). \tag{3.31}$$

Proof. Denote $\rho(t) \triangleq \sum_{i=1}^n \|e_i(t)\|$ for the sake of simplicity. The time derivative

of ρ is

$$\begin{aligned}
\dot{\rho} &= \sum_{i=1}^{n} \frac{\mathrm{d}\|e_i\|}{\mathrm{d}t} \\
&= \sum_{i=1}^{n} g_i^{\mathrm{T}} \dot{e}_i \\
&= \sum_{i=1}^{n} g_i^{\mathrm{T}} [\sigma_{i+1}(g_{i+1} - g_i) + \sigma_i(g_{i-1} - g_i)] \quad \text{(By (3.18))} \\
&= \sum_{i=1}^{n} [\sigma_{i+1}(g_i^{\mathrm{T}} g_{i+1} - 1) + \sigma_i(g_i^{\mathrm{T}} g_{i-1} - 1)] \\
&= \sum_{i=1}^{n} \sigma_i(g_{i-1}^{\mathrm{T}} g_i - 1) + \sum_{i=1}^{n} \sigma_i(g_i^{\mathrm{T}} g_{i-1} - 1) \\
&= 2\sum_{i=1}^{n} \sigma_i(g_i^{\mathrm{T}} g_{i-1} - 1) \\
&= v^{\mathrm{T}} \sigma,
\end{aligned}$$

where $v = [v_1, \ldots, v_n]^{\mathrm{T}} \in \mathbb{R}^n$ with $v_i = 2(g_i^{\mathrm{T}} g_{i-1} - 1)$. By the Cauchy-Schwarz inequality, we have

$$|\dot{\rho}| = |v^{\mathrm{T}} \sigma| \leq \|v\| \|\sigma\| \leq \beta \|\sigma\|, \tag{3.32}$$

where β is the maximum of $\|v\|$ over the compact set $\Omega(V(\varepsilon_0))$.

Furthermore, note

$$\begin{aligned}
V^{\frac{2a}{1+a}} &= \left(\frac{1}{a+1}\right)^{\frac{2a}{a+1}} \left(\sum_{i=1}^{n} |\varepsilon_i|^{a+1}\right)^{\frac{2a}{a+1}} \\
&\geq \left(\frac{1}{a+1}\right)^{\frac{2a}{a+1}} \frac{1}{n^{\frac{1-a}{1+a}}} \sum_{i=1}^{n} |\varepsilon_i|^{2a} \quad \text{(By Lemma 3.5)} \\
&= \left(\frac{1}{a+1}\right)^{\frac{2a}{a+1}} \frac{1}{n^{\frac{1-a}{1+a}}} \|\sigma\|^2,
\end{aligned}$$

which implies

$$\|\sigma\|^2 \leq \underbrace{(a+1)^{\frac{2a}{a+1}} n^{\frac{1-a}{1+a}}}_{\kappa} V^{\frac{2a}{a+1}}. \tag{3.33}$$

Substituting (3.33) into (3.32) yields

$$|\dot{\rho}| \leq \beta\sqrt{\kappa}V^{\frac{a}{a+1}}. \tag{3.34}$$

On the other hand, if ε_0 is sufficiently small such that $V(\varepsilon_0) \leq 1$, then $V^{\frac{2a}{1+a}} \geq V$ for all $\varepsilon \in \Omega(V(\varepsilon_0))$ as $2a/(1+a) \leq 1$. Thus (3.23) implies

$$\dot{V} \leq -\frac{K}{\rho}V^{\frac{2a}{1+a}} \leq -\frac{K}{\rho}V$$

for $\varepsilon \in \Omega(V(\varepsilon_0))$. By the comparison lemma [68, Lemma 3.4], the above inequality suggests

$$V(t) \leq V(0)\exp\left(\int_0^t -\frac{K}{\rho(\tau)}\mathrm{d}\tau\right). \tag{3.35}$$

Substituting (3.35) into (3.34) yields

$$|\dot{\rho}| \leq \beta\sqrt{\kappa}V(0)^{\frac{a}{a+1}}\exp\left(\int_0^t -\frac{\frac{a}{a+1}K}{\rho(\tau)}\mathrm{d}\tau\right). \tag{3.36}$$

Note (3.36) holds for $t \in [0, T_c)$.

Based on (3.36) we draw the following conclusions. (i) If T_c is infinity, (3.36) holds for $t \in [0, +\infty)$. By Lemma 3.4 there exists a finite constant that bounds $\rho(t)$ above for all $t \in [0, +\infty)$. (ii) If T_c is finite, it is obvious that $\rho(t)$ is finite for all $t \in [0, T_c)$ because the speed of each vehicle is finite. In either case, denote γ as the finite upper bound of ρ. Then it is evident to have (3.31) from (3.23). □

Remark 3.4. *Since the formation is a cycle, the distance between any two vehicles (even they are not neighbors) is smaller than* $\sum_{i=1}^{n}\|e_i\|$. *Hence Proposition 3.2 implies that the distance between any vehicles is always finite during the whole formation evolution.*

Collision avoidance is an important problem in various formation control tasks. It is especially important for bearing-based formation control as the inter-vehicle distances are unmeasurable and uncontrollable. Based on the results of Proposition 3.2, we next further prove no vehicles will collide with each other under control law (3.14) during the whole formation evolution.

Proposition 3.3. *[Collision Avoidance] Under Assumptions 3.1 and 3.2, if* (3.31) *holds and the initial error* ε_0 *is sufficiently small such that* $V(\varepsilon_0) \leq 1$*, then there exists a positive constant* η *such that*

$$\sum_{i=1}^{n} \|z_i(t) - z_i(0)\| \leq \eta \|\varepsilon_0\|_{a+1}^a, \quad \forall t \in [0, T_c). \tag{3.37}$$

Furthermore, if ε_0 *satisfies*

$$\|z_j(0) - z_k(0)\| > \eta \|\varepsilon_0\|_{a+1}^a \tag{3.38}$$

for all $j, k \in \{1, \ldots, n\}$ *and* $j \neq k$*, then* $T_c = +\infty$ *and the distance between any two vehicles is bounded below by a positive constant during the whole formation evolution.*

Proof. We first prove (3.37). The quantity $\sum_{i=1}^{n} \|z_i(t) - z_i(0)\|$ actually characterizes the "distance" from the formation at time t to the initial formation. Recall

$$z_i(t) - z_i(0) = \int_0^t \sigma_i (g_i - g_{i-1}) \mathrm{d}\tau$$

by control law (3.14). Then we have

$$\begin{aligned}
\sum_{i=1}^{n} \|z_i(t) - z_i(0)\| &= \sum_{i=1}^{n} \left\| \int_0^t \sigma_i (g_i - g_{i-1}) \mathrm{d}\tau \right\| \\
&\leq \sum_{i=1}^{n} \int_0^t |\varepsilon_i|^a \|g_i - g_{i-1}\| \mathrm{d}\tau \\
&\leq 2 \int_0^t \sum_{i=1}^{n} |\varepsilon_i|^a \mathrm{d}\tau \quad \text{(Becuause } \|g_i - g_{i-1}\| \leq \|g_i\| + \|g_{i-1}\| = 2\text{)} \\
&\leq 2n^{1-a} \int_0^t \|\varepsilon(t)\|_1^a \mathrm{d}\tau \quad \text{(By Lemma 3.5)} \\
&\leq 2n^{1-a} C \int_0^t \|\varepsilon(t)\|_{a+1}^a \mathrm{d}\tau. \quad \text{(By Lemma 3.6)}
\end{aligned} \tag{3.39}$$

If ε_0 is sufficiently small such that $V(\varepsilon_0) \leq 1$ and hence $V(t) \leq 1$ for all $t \in [0, T_c)$, then $V^{\frac{2a}{1+a}} \geq V$ as $2a/(1+a) \leq 1$. Consequently (3.31) implies

$$\dot{V} \leq -\frac{K}{\gamma} V^{\frac{2a}{1+a}} \leq -\frac{K}{\gamma} V,$$

which suggests

$$V(t) \leq V(0)e^{-\frac{K}{\gamma}t}, \quad \forall t \in [0, T_c). \tag{3.40}$$

Substituting $V = 1/(a+1)\|\varepsilon\|_{a+1}^{a+1}$ into (3.40) yields

$$\|\varepsilon(t)\|_{a+1} \leq \|\varepsilon(0)\|_{a+1} e^{-\frac{K}{(a+1)\gamma}t}.$$

Substituting the above inequality into (3.39) gives

$$\begin{aligned}
\sum_{i=1}^{n} \|z_i(t) - z_i(0)\| &\leq 2n^{1-a}C \int_0^t \|\varepsilon(0)\|_{a+1}^a e^{-\frac{aK}{(a+1)\gamma}\tau} \mathrm{d}\tau \\
&= 2n^{1-a}C\|\varepsilon(0)\|_{a+1}^a \frac{(a+1)\gamma}{aK}\left(1 - e^{-\frac{aK}{(a+1)\gamma}t}\right) \\
&\leq \underbrace{\frac{2n^{1-a}C(a+1)\gamma}{aK}}_{\eta} \|\varepsilon(0)\|_{a+1}^a
\end{aligned} \tag{3.41}$$

for all $t \in [0, T_c)$.

With the above preparation, we now prove collision avoidance by contradiction. Assume vehicles j and k collide at a *finite* time T_c, which means

$$z_j(T_c) = z_k(T_c). \tag{3.42}$$

Note vehicles j and k are not necessarily neighbors. However, since $z_j(t) - z_k(t) \equiv z_j(0) - z_k(0) - [z_k(t) - z_k(0)] - [z_j(0) - z_j(t)]$, the distance between vehicles j and k satisfies

$$\begin{aligned}
\|z_j(t) - z_k(t)\| &\geq \|z_j(0) - z_k(0)\| - \|z_k(t) - z_k(0)\| - \|z_j(t) - z_j(0)\| \\
&\geq \|z_j(0) - z_k(0)\| - \sum_{i=1}^{n} \|z_i(t) - z_i(0)\| \\
&\geq \|z_j(0) - z_k(0)\| - \eta\|\varepsilon(0)\|_{a+1}^a \quad \text{(By (3.41))} \\
&> 0, \quad \forall t \in [0, T_c).
\end{aligned} \tag{3.43}$$

The last inequality is by the condition (3.38). Inequality (3.43) indicates that the distance between any two vehicles is bounded below by a positive constant

for all $t \in [0, T_c)$. Clearly (3.42) conflicts with (3.43). Thus we have $T_c = +\infty$ and collision avoidance between any vehicles can be ensured. □

Remark 3.5. *As shown in* (3.42) *and* (3.43)*, it is not assumed that vehicles j and k are neighbors. Hence collision avoidance is guaranteed between any vehicles no matter if they are neighbors or not.*

We next summarize Propositions 3.1, 3.2 and 3.3 and give the main stability results as below.

Theorem 3.1. *Under Assumptions 3.1 and 3.2, the equilibrium $\varepsilon = 0$ is locally exponentially stable by control law* (3.14) *if $a = 1$, and locally finite-time stable if $a \in (0, 1)$. Collision avoidance between any vehicles (no matter if they are neighbors or not) is locally guaranteed.*

Proof. By Propositions 3.2 and 3.3, we have

$$\dot{V} \leq -\frac{K}{\gamma} V^{\frac{2a}{1+a}}, \quad \forall t \in [0, +\infty), \tag{3.44}$$

given sufficiently small $\varepsilon(0)$. From (3.44) we conclude: (i) If $a \in (0, 1)$ and hence $2a/(1 + a) \in (0, 1)$, the solution to (3.14) starting from $\Omega(V(\varepsilon_0))$ converges to $\varepsilon = 0$ in finite time [7, Theorem 4.2]. (ii) If $a = 1$ and hence $2a/(1 + a) = 1$, the equilibrium $\varepsilon = 0$ is locally exponentially stable [54, Theorem 3.1]. Collision avoidance has already been proved in Proposition 3.3. □

Remark 3.6. *As shown in Propositions 3.1, 3.2 and 3.3, if ε_0 satisfies* (3.67)*,* (3.38) *and $V(\varepsilon_0) \leq 1$, then the convergence and collision avoidance can be guaranteed. Note the right hand side of* (3.67) *is less than one. Hence* (3.67) *implies $V(\varepsilon_0) \leq 1$. As a result, we can obtain a convergence region from* (3.67) *and* (3.38)*. But this convergence region may be conservative. The real convergence region is not necessarily small, which will be illustrated by simulations.*

Up to this point, we have been primarily focusing on the convergence of $\varepsilon(t)$. It should be noted that the convergence of $\varepsilon(t)$ does not simply imply the formation $\{z_i(t)\}_{i=1}^n$ converges to a finite final position. But this issue can be solved by the exponential or finite-time convergence rate. Specifically, control

law (3.14) implies that $z_i(t) = z_i(0) + \int_0^t \sigma_i(g_i - g_{i-1})$. Since ε_i converges to zero exponentially or in finite time, the function $\sigma_i(g_i - g_{i-1})$ is integrable even if $t \to +\infty$. As a result, $\{z_i(t)\}_{i=1}^n$ will converge to a finite position and control law (3.14) successfully solves Problem 3.1.

Remark 3.7. *The exponential or finite-time stability not only shows the fast convergence rate of the proposed control law, but also is necessarily useful for proving the finite position of the final converged formation. It is notable that similar problems also appear in control of distance-constrained formations [35, Section V], where the exponential convergence rate of distance dynamics is first proved and then used to prove the formation converging to a finite final position.*

At last, we characterize a number of important behaviors of the formation evolution. (i) Inequality (3.37) intuitively indicates that the final converged formation would not move far away from the initial formation if the initial angle errors are small. (ii) From control law (3.14), it is obvious that $\dot{z} = 0$ if $\varepsilon = 0$. It intuitively means that the vehicles will stop moving once the angles achieve the desired values. (iii) Another important behavior of the formation is that $\dot{z} = 0$ if $\dot{\varepsilon} = 0$. That is because $\dot{\varepsilon} = 0 \Rightarrow \dot{V} = 0 \Rightarrow V = 0 \Rightarrow \varepsilon = 0 \Rightarrow \dot{z} = 0$. The intuitive interpretation is that control law (3.14) cannot change the positions of the vehicles without changing the angles in the formation.

3.5 Stability Analysis of the Discontinuous Case

In the case of $a = 0$, the control law (3.14) will be discontinuous in ε. Then we need to use the non-smooth analysis tools to prove the formation stability.

3.5.1 Error Dynamics

We next derive the error dynamics of the closed-loop system under control law (3.14). Denote $\varepsilon = [\varepsilon_1, ..., \varepsilon_n]^{\mathrm{T}} \in \mathbb{R}^n$. Recall g_i is defined as $g_i = e_i/\|e_i\|$. Then the time derivative of g_i is

$$\dot{g}_i = \frac{1}{\|e_i\|} P_i \dot{e}_i, \tag{3.45}$$

where $P_i = I - g_i g_i^{\mathrm{T}}$. Based on (3.45), we obtain the dynamics of ε as below.

Theorem 3.2. *The ε-dynamics under control law* (3.14) *is*

$$\dot{\varepsilon} = -A\mathrm{sgn}(\varepsilon), \tag{3.46}$$

where $A \in \mathbb{R}^{n\times n}$ and all of the entries of A are zero except

$$\begin{aligned}
[A]_{i(i-1)} &= \frac{1}{\|e_{i-1}\|} g_i^{\mathrm{T}} P_{i-1} g_{i-2}, \\
[A]_{ii} &= \frac{1}{\|e_{i-1}\|} g_i^{\mathrm{T}} P_{i-1} g_i + \frac{1}{\|e_i\|} g_{i-1}^{\mathrm{T}} P_i g_{i-1}, \\
[A]_{i(i+1)} &= \frac{1}{\|e_i\|} g_{i-1}^{\mathrm{T}} P_i g_{i+1},
\end{aligned} \tag{3.47}$$

for all $i \in \{1, \ldots, n\}$.

Proof. Recall $e_i = z_{i+1} - z_i$. Then substituting control law (3.14) into $\dot{e}_i = \dot{z}_{i+1} - \dot{z}_i$ yields

$$\begin{aligned}
\dot{e}_i &= \dot{z}_{i+1} - \dot{z}_i \\
&= \mathrm{sgn}(\varepsilon_{i+1})(g_{i+1} - g_i) - \mathrm{sgn}(\varepsilon_i)(g_i - g_{i-1}) \\
&= \mathrm{sgn}(\varepsilon_{i+1}) g_{i+1} + \mathrm{sgn}(\varepsilon_i) g_{i-1} - [\mathrm{sgn}(\varepsilon_{i+1}) + \mathrm{sgn}(\varepsilon_i)]\, g_i.
\end{aligned} \tag{3.48}$$

Substituting (3.48) into (3.45) and using the fact that $P_i g_i = 0$ gives

$$\dot{g}_i = \frac{1}{\|e_i\|} P_i \left[\mathrm{sgn}(\varepsilon_{i+1}) g_{i+1} + \mathrm{sgn}(\varepsilon_i) g_{i-1}\right].$$

Recall $\varepsilon_i = -g_i^{\mathrm{T}} g_{i-1} - \cos\theta_i^*$ as defined in (3.13) and θ_i^* is constant. Then by the above equation we have

$$\begin{aligned}
\dot{\varepsilon}_i &= -g_i^{\mathrm{T}} \dot{g}_{i-1} - g_{i-1}^{\mathrm{T}} \dot{g}_i \\
&= -\frac{1}{\|e_{i-1}\|} g_i^{\mathrm{T}} P_{i-1} \left[\mathrm{sgn}(\varepsilon_i) g_i + \mathrm{sgn}(\varepsilon_{i-1}) g_{i-2}\right] - \frac{1}{\|e_i\|} g_{i-1}^{\mathrm{T}} P_i \left[\mathrm{sgn}(\varepsilon_{i+1}) g_{i+1} + \mathrm{sgn}(\varepsilon_i) g_{i-1}\right] \\
&= -[A]_{i(i-1)} \mathrm{sgn}(\varepsilon_{i-1}) - [A]_{ii} \mathrm{sgn}(\varepsilon_i) - [A]_{i(i+1)} \mathrm{sgn}(\varepsilon_{i+1}),
\end{aligned} \tag{3.49}$$

where $[A]_{i(i-1)}$, $[A]_{ii}$ and $[A]_{i(i+1)}$ are given in (3.47). It is straightforward to see the matrix form of (3.49) is (3.46). □

We next prove the matrix A in (3.46) is symmetric positive semi-definite.

Corollary 3.1. *The matrix A in* (3.46) *is symmetric positive semi-definite. For any $x = [x_1, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^n$,*

$$x^{\mathrm{T}} A x = \sum_{i=1}^{n} \frac{1}{\|e_i\|} \left(g_{i+1} x_{i+1} + g_{i-1} x_i\right)^{\mathrm{T}} P_i \left(g_{i+1} x_{i+1} + g_{i-1} x_i\right) \geq 0. \tag{3.50}$$

Proof. In order to prove A is symmetric, we only need to prove $[A]_{(i+1)i} = [A]_{i(i+1)}$ for all i. By changing the index i in $[A]_{i(i-1)}$ in (3.47) to $i+1$, we obtain

$$[A]_{(i+1)i} = \frac{1}{\|e_i\|} g_{i+1}^{\mathrm{T}} P_i g_{i-1}.$$

It is clear that $[A]_{(i+1)i} = [A]_{i(i+1)}$ due to the symmetry of P_i. For any vector $x = [x_1, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^n$, we have

$$\begin{aligned}
x^{\mathrm{T}} A x &= \sum_{i=1}^{n} [A]_{i(i-1)} x_i x_{i-1} + [A]_{ii} x_i^2 + [A]_{i(i+1)} x_i x_{i+1} \\
&= \sum_{i=1}^{n} \left(\frac{1}{\|e_{i-1}\|} g_i^{\mathrm{T}} P_{i-1} g_{i-2}\right) x_i x_{i-1} + \sum_{i=1}^{n} \left(\frac{1}{\|e_{i-1}\|} g_i^{\mathrm{T}} P_{i-1} g_i\right) x_i^2 \\
&\qquad + \sum_{i=1}^{n} \left(\frac{1}{\|e_i\|} g_{i-1}^{\mathrm{T}} P_i g_{i-1}\right) x_i^2 + \sum_{i=1}^{n} \left(\frac{1}{\|e_i\|} g_{i-1}^{\mathrm{T}} P_i g_{i+1}\right) x_i x_{i+1} \\
&= \sum_{i=1}^{n} \left(\frac{1}{\|e_i\|} g_{i+1}^{\mathrm{T}} P_i g_{i-1}\right) x_{i+1} x_i + \sum_{i=1}^{n} \left(\frac{1}{\|e_i\|} g_{i+1}^{\mathrm{T}} P_i g_{i+1}\right) x_{i+1}^2 \\
&\qquad + \sum_{i=1}^{n} \left(\frac{1}{\|e_i\|} g_{i-1}^{\mathrm{T}} P_i g_{i-1}\right) x_i^2 + \sum_{i=1}^{n} \left(\frac{1}{\|e_i\|} g_{i-1}^{\mathrm{T}} P_i g_{i+1}\right) x_i x_{i+1} \\
&= \sum_{i=1}^{n} \frac{1}{\|e_i\|} \left(g_{i+1} x_{i+1} + g_{i-1} x_i\right)^{\mathrm{T}} P_i \left(g_{i+1} x_{i+1} + g_{i-1} x_i\right) \geq 0,
\end{aligned}$$

where the last inequality is due to the fact that P_i is positive semi-definite. □

3.5.2 Finite-time Stability Analysis

In this section we analyze the stability of the error dynamics (3.46). By employing a locally Lipschitz Lyapunov function and the nonsmooth analysis tools introduced in Section 3.2.3, we prove that the origin $\varepsilon = 0$ is locally finite-time stable with collision avoidance guaranteed. In addition to the dynamics of ε, we

also analyze the behaviors of the vehicle positions during formation evolution.

We first consider the problem of collision avoidance. On one hand, Assumption 3.2 states that no vehicles coincide with each other in the initial formation, i.e., $z_i(0) \neq z_j(0)$ for any $i \neq j$. On the other hand, control law (3.14) implies that $\|\dot{z}_i\| \leq \|g_i - g_{i-1}\| \leq 2$, which means that the maximum speed of each vehicle is two. Therefore, any two vehicles are not able to collide with each other (no matter they are neighbors or not) for all $t \in [0, T^*)$ where

$$T^* \triangleq \frac{\min_{i \neq j} \|z_i(0) - z_j(0)\|}{4}.$$

In the sequel, we will only consider $t \in [0, T]$ with $T < T^*$. We will prove that the system can be stabilized within the finite time interval $[0, T]$.

Consider the Lyapunov function

$$V(\varepsilon) = \sum_{i=1}^{n} |\varepsilon_i|, \tag{3.51}$$

which is positive definite in ε. Note $V(\varepsilon)$ is locally Lipschitz and convex. Hence $V(\varepsilon)$ is also regular.

Theorem 3.3. *For the error dynamics* (3.46) *and Lyapunov function* (3.51)*, the Filippov differential inclusion is* $\dot{\varepsilon} \in -A\partial V(\varepsilon)$*. The set-valued Lie derivative is given by*

$$\widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon) = \{\ell \in \mathbb{R} \mid \exists \eta \in \partial V(\varepsilon),\ \forall \zeta \in \partial V(\varepsilon),\ -\zeta^{\mathrm{T}} A \eta = \ell\}. \tag{3.52}$$

When $\widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon) \neq \emptyset$*, for any* $\ell \in \widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon)$*, there exsits* $\eta \in \partial V(\varepsilon)$ *such that*

$$\ell = -\eta^{\mathrm{T}} A \eta \leq 0. \tag{3.53}$$

Proof. Step 1: calculate the generalized gradient. By definition we have the generalized gradient as

$$\partial V(\varepsilon) = \{\eta = [\eta_1, \ldots, \eta_n]^{\mathrm{T}} \in \mathbb{R}^n \mid \eta_i = \operatorname{sgn}(\varepsilon_i) \text{ if } \varepsilon_i \neq 0 \text{ and } \eta_i \in [-1, 1] \text{ if } \varepsilon_i = 0 \text{ for } i \in \{1, \ldots, n\}\}.$$

Because $|\eta_i| = |\mathrm{sgn}(\varepsilon_i)| = 1$ if $\varepsilon_i \neq 0$, we have the obvious but important fact that

$$\|\eta\| \geq 1, \quad \forall \eta \in \partial V(\varepsilon),\ \forall \varepsilon \neq 0. \tag{3.54}$$

Additionally, if $\varepsilon_i \neq 0$, then $\mathrm{Ln}(\{\mathrm{sgn}(\varepsilon_i)\}) = \{\mathrm{sgn}(\varepsilon_i)\}$; and if $\varepsilon_i = 0$, then $\mathrm{Ln}([-1,1]) = \{0\} = \{\mathrm{sgn}(0)\}$. Thus we have the following useful property

$$\mathrm{Ln}(\partial V(\varepsilon)) = \{\mathrm{sgn}(\varepsilon)\}. \tag{3.55}$$

Step 2: calculate the Filippov differential inclusion. Since $\|e_i(t)\| \neq 0$ for all i and all $t \in [0,T]$, the matrix A in (3.14) is continuous. Then by [98, Theorem 1, 5)], the Filippov differential inclusion associated with the system (3.46) can be calculated as

$$\dot{\varepsilon} \in \mathcal{F}[-A\mathrm{sgn}(\varepsilon)] = -A\mathcal{F}[\mathrm{sgn}(\varepsilon)]. \tag{3.56}$$

Substituting (3.55) into (3.56) yields

$$\mathcal{F}[\mathrm{sgn}(\varepsilon)] = \mathcal{F}[\mathrm{Ln}(\partial V(\varepsilon))] = \partial V(\varepsilon),$$

where the last equality uses the fact (3.4). Thus the Filippov differential inclusion in (3.56) can be rewritten as

$$\dot{\varepsilon} \in -A\partial V(\varepsilon). \tag{3.57}$$

Step 3: calculate the set-valued Lie derivative. The set-valued Lie derivative of $V(\varepsilon)$ with respect to (3.57) is given by

$$\begin{aligned}\widetilde{\mathcal{L}}_{-A\partial V}V(\varepsilon) &= \{\ell \in \mathbb{R} \mid \exists \xi \in -A\partial V(\varepsilon),\ \forall \zeta \in \partial V(\varepsilon),\ \zeta^{\mathrm{T}}\xi = \ell\} \\ &= \{\ell \in \mathbb{R} \mid \exists \eta \in \partial V(\varepsilon),\ \forall \zeta \in \partial V(\varepsilon),\ -\zeta^{\mathrm{T}}A\eta = \ell\}.\end{aligned}$$

The set $\widetilde{\mathcal{L}}_{-A\partial V}V(\varepsilon)$ could be empty. When $\widetilde{\mathcal{L}}_{-A\partial V}V(\varepsilon) \neq \emptyset$, for any $\ell \in \widetilde{\mathcal{L}}_{-A\partial V}V(\varepsilon)$, there exists $\eta \in \partial V$ such that $\ell = -\zeta^{\mathrm{T}}A\eta$ for all $\zeta \in \partial V$. In

particular, by choosing $\zeta = \eta$ we have $\ell = -\eta^{\mathrm{T}} A\eta \leq 0$. Note $-\eta^{\mathrm{T}} A\eta \leq 0$ is due to the fact that A is a positive semi-definite matrix as shown in Lemma 3.1. Therefore, we have either $\widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon) = \emptyset$ or $\max \widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon) \leq 0$. □

We are ready to prove the formation stability based on Theorem 3.3. Note if $\widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon) = \emptyset$, we have $\max \widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon) = -\infty$ (see Section 3.2.3). Hence we need only to focus on the case of $\widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon) \neq \emptyset$.

Theorem 3.4. *Consider the set-valued Lie derivative given in* (3.52)*. When* $\widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon) \neq \emptyset$*, for any* $\ell \in \widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon)$*, there exsits* $\eta \in \partial V(\varepsilon)$ *such that*

$$\ell \leq -\frac{1}{\sum_{i=1}^{n} \|e_i\|} \eta^{\mathrm{T}} D^{\mathrm{T}} E^{\mathrm{T}} E D \eta, \tag{3.58}$$

where

$$E = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & \dots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{n \times n},$$

$$D = \begin{bmatrix} (g_1^{\perp})^{\mathrm{T}} g_n & 0 & 0 & \dots & 0 \\ 0 & (g_2^{\perp})^{\mathrm{T}} g_1 & 0 & \dots & 0 \\ 0 & 0 & (g_3^{\perp})^{\mathrm{T}} g_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & (g_n^{\perp})^{\mathrm{T}} g_{n-1} \end{bmatrix} \in \mathbb{R}^{n \times n}. \tag{3.59}$$

Proof. By (3.50), we can rewrite $\ell = -\eta^{\mathrm{T}} A\eta$ in (3.53) as

$$\begin{aligned} \ell &= -\sum_{i=1}^{n} \frac{1}{\|e_i\|} \left(g_{i+1}\eta_{i+1} + g_{i-1}\eta_i\right)^{\mathrm{T}} P_i \left(g_{i+1}\eta_{i+1} + g_{i-1}\eta_i\right) \\ &\leq -\frac{1}{\sum_{i=1}^{n} \|e_i\|} \sum_{i=1}^{n} \left(g_{i+1}\eta_{i+1} + g_{i-1}\eta_i\right)^{\mathrm{T}} P_i \left(g_{i+1}\eta_{i+1} + g_{i-1}\eta_i\right) \\ &= -\frac{1}{\sum_{i=1}^{n} \|e_i\|} \sum_{i=1}^{n} \left[\left(g_{i+1}\eta_{i+1} + g_{i-1}\eta_i\right)^{\mathrm{T}} g_i^{\perp}\right]^2 \quad \text{(By Lemma 3.8(ii))} \\ &= -\frac{1}{\sum_{i=1}^{n} \|e_i\|} \sum_{i=1}^{n} \left[(g_i^{\perp})^{\mathrm{T}} g_{i+1}\eta_{i+1} + (g_i^{\perp})^{\mathrm{T}} g_{i-1}\eta_i\right]^2 \end{aligned}$$

$$= -\frac{1}{\sum_{i=1}^{n} \|e_i\|} h^{\mathrm{T}} h, \tag{3.60}$$

where

$$\begin{aligned} h &= \begin{bmatrix} (g_1^{\perp})^{\mathrm{T}} g_2 \eta_2 + (g_1^{\perp})^{\mathrm{T}} g_n \eta_1 \\ \vdots \\ (g_n^{\perp})^{\mathrm{T}} g_1 \eta_1 + (g_n^{\perp})^{\mathrm{T}} g_{n-1} \eta_n \end{bmatrix} \\ &= \begin{bmatrix} (g_1^{\perp})^{\mathrm{T}} g_n & (g_1^{\perp})^{\mathrm{T}} g_2 & 0 & \dots & 0 \\ 0 & (g_2^{\perp})^{\mathrm{T}} g_1 & (g_2^{\perp})^{\mathrm{T}} g_3 & \dots & 0 \\ 0 & 0 & (g_3^{\perp})^{\mathrm{T}} g_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (g_n^{\perp})^{\mathrm{T}} g_1 & 0 & \dots & 0 & (g_n^{\perp})^{\mathrm{T}} g_{n-1} \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \vdots \\ \eta_n \end{bmatrix} \\ &= ED\eta \end{aligned} \tag{3.61}$$

with E and D given in (3.59). The last equality of (3.61) uses the fact that $(g_i^{\perp})^{\mathrm{T}} g_{i-1} = -(g_{i-1}^{\perp})^{\mathrm{T}} g_i$ as shown in Lemma 3.8(iii). Substituting (3.61) into (3.60) gives (3.58). □

Note that D is a diagonal matrix and E actually is an incidence matrix of a directed and connected cycle graph. We now present the main stability result.

Theorem 3.5. *Under Assumptions 3.2 and 3.1, the equilibrium $\varepsilon = 0$ of system (3.46) is locally finite-time stable. Collision avoidance between any vehicles (no matter they are neighbors or not) can be locally guaranteed.*

Proof. Consider the time interval $[0, T]$ with $T < T^*$. Then $\|e_i(t)\| \neq 0$ and $\|e_i(t)\| \neq +\infty$ for all $t \in [0, T]$. We will prove that ε can converge to zero in the finite time interval $[0, T]$ if $\varepsilon(0)$ is sufficiently small.

Let $\Omega(\varepsilon(0)) \triangleq \{\varepsilon \in \mathbb{R}^n \mid V(\varepsilon) \leq V(\varepsilon(0))\}$. Since $V(\varepsilon) = \sum_{i=1}^{n} |\varepsilon_i| = \|\varepsilon\|_1$, the level set $\Omega(\varepsilon(0))$ is connected and compact. Because $\widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon) = \emptyset$ or $\max \widetilde{\mathcal{L}}_{-A\partial V} V(\varepsilon) \leq 0$ for any $\varepsilon \in \Omega(\varepsilon(0))$ as proved in Theorem 3.3, we have that $\Omega(\varepsilon(0))$ is strongly invariant to (3.46) over $[0, T]$ by Lemma 3.1.

Step 1: prove the nonzero entries of $D\eta$ do not have the same sign.

Denote $\delta_i = \theta_i - \theta_i^*$ and $\delta = [\delta_1, \dots, \delta_n]^{\mathrm{T}} \in \mathbb{R}^n$. Consider the case of $\varepsilon \neq 0$

and hence $\delta \neq 0$. Because $\sum_{i=1}^n \theta_i \equiv \sum_{i=1}^n \theta_i^*$, we have $\sum_{i=1}^n \delta_i \equiv 0$. Thus the nonzero entries of δ *do not have the same sign* if $\delta \neq 0$. Let

$$w_i \triangleq \frac{\cos\theta_i - \cos\theta_i^*}{\theta_i - \theta_i^*}.$$

Then $\varepsilon_i = w_i\delta_i$ and hence

$$\varepsilon = W\delta,$$

where $W = \mathrm{diag}\{w_1, \ldots, w_n\} \in \mathbb{R}^{n\times n}$. Since $\lim_{\theta_i\to\theta_i^*} w_i = -\sin\theta_i^*$ by L'Hôpital's rule, the equations $\varepsilon_i = w_i\delta_i$ and $\varepsilon = W\delta$ are always valid even when $\theta_i - \theta_i^* = 0$.

Suppose $V(\varepsilon(0))$ is sufficiently small such that $\theta_i(0)$ is sufficiently close to θ_i^* and hence $\theta_i, \theta_i^* \in (0,\pi)$ or $\theta_i, \theta_i^* \in (\pi, 2\pi)$ for all $\varepsilon \in \Omega(\varepsilon(0))$. Then it is easy to see that $w_i < 0$ if $\theta_i, \theta_i^* \in (0,\pi)$, and $w_i > 0$ if $\theta_i, \theta_i^* \in (\pi, 2\pi)$. On the other hand, recall $(g_i^\perp)^{\mathrm{T}} g_{i-1} > 0$ when $\theta_i \in (0,\pi)$, and $(g_i^\perp)^{\mathrm{T}} g_{i-1} < 0$ when $\theta_i \in (\pi, 2\pi)$ as shown in Lemma 3.8(iv). Thus we have

$$(g_i^\perp)^{\mathrm{T}} g_{i-1} w_i < 0$$

for all $i \in \{1, \ldots, n\}$. Since $[D]_{ii} = (g_i^\perp)^{\mathrm{T}} g_{i-1}$, the above inequality implies that the diagonal entries of DW *have the same sign.* Thus as the nonzero entries in δ do not have the same sign, the nonzero entries of $DW\delta = D\varepsilon$ *do not have the same sign* either. Furthermore, because $\eta_i = \mathrm{sgn}(\varepsilon_i)$ if $\varepsilon_i \neq 0$, the nonzero entry ε_i has the same sign with η_i. As a result, the nonzero entries of $D\eta$ *do not have the same sign.* Thus we have $D\eta \in \mathcal{U}$ with $\mathcal{U}$ defined in Lemma 3.3.

Step 2: determine the negative upper bound of ℓ.

Note E is an incidence matrix of a directed and connected cycle graph. By [50, Theorem 8.3.1], we have $\mathrm{rank}(E) = n-1$ and $\mathrm{Null}\,(E^{\mathrm{T}}E) = \mathrm{Null}\,(E) = \mathrm{span}\{\mathbf{1}\}$. Thus inequality (3.58) implies

$$\begin{aligned}
\ell &\leq -\frac{1}{\sum_{i=1}^n \|e_i\|} \frac{\lambda_2(E^{\mathrm{T}}E)}{n} \|D\eta\|^2 \quad \text{(By Lemma 3.3)} \\
&\leq -\frac{1}{\sum_{i=1}^n \|e_i\|} \frac{\lambda_2(E^{\mathrm{T}}E)}{n} \lambda_1(D^2) \|\eta\|^2
\end{aligned}$$

$$\leq -\frac{1}{\sum_{i=1}^{n}\|e_i\|}\frac{\lambda_2(E^{\mathrm{T}}E)}{n}\lambda_1(D^2), \tag{3.62}$$

where the last inequality uses the fact $\|\eta\| \geq 1$ if $\varepsilon \neq 0$ as shown in (3.54).

Now we analyze the two terms, $\sum_{i=1}^{n}\|e_i\|$ and $\lambda_1(D^2)$, in (3.62). (i) Over the finite time interval $[0, T]$, the quantity $\sum_{i=1}^{n}\|e_i\|$ cannot go to infinity because the vehicle speed is finite. Hence there exists a constant $\gamma > 0$ such that $\sum_{i=1}^{n}\|e_i\| \leq \gamma$. (ii) Since D is diagonal, we have $\lambda_1(D^2) = \min_i [D]_{ii}^2$. At the equilibrium point $\varepsilon = 0$ (i.e., $\theta_i = \theta_i^*$ for all i), we have $[D]_{ii} = (g_i^{\perp})^{\mathrm{T}} g_{i-1} \neq 0$ because $\theta_i^* \neq 0$ or π as stated in Assumption 3.1. By continuity, we can still have $[D]_{ii} \neq 0$ for all $\varepsilon \in \Omega(\varepsilon(0))$ if $\varepsilon(0)$ is sufficiently small. Because $\Omega(\varepsilon(0))$ is compact, there exists a lower bound β such that $\lambda_1(D^2) \geq \beta$ for all $\varepsilon \in \Omega(\varepsilon(0))$. By (i) and (ii), inequality (3.62) can be rewritten as

$$\ell \leq -\frac{\beta\lambda_2(E^{\mathrm{T}}E)}{\gamma n} \triangleq -\kappa < 0, \quad \forall \varepsilon \in \Omega(\varepsilon(0)) \setminus \{0\}. \tag{3.63}$$

Step 3: draw the stability conclusion.

If $\varepsilon = 0$ we have $0 \in \widetilde{\mathcal{L}}_{-A\partial V}V(\varepsilon)$ because of (3.52) and the fact that $0 \in \partial V(0)$; if $\varepsilon \neq 0$ we have $0 \notin \widetilde{\mathcal{L}}_{-A\partial V}V(\varepsilon)$ because $\max \widetilde{\mathcal{L}}_{-A\partial V}V(\varepsilon) < 0$ by (3.63). Thus by the definition (3.5), we have

$$Z_{-A\mathrm{sgn}(\varepsilon),V(\varepsilon)} = \{0\}. \tag{3.64}$$

Based on (3.63), (3.64) and Lemma 3.2, any solution of (3.46) starting from $\varepsilon(0)$ converges to $\varepsilon = 0$ in finite-time, and the convergence time is upper bounded by $V(\varepsilon(0))/\kappa$. Thus if $V(\varepsilon(0))$ satisfies

$$\frac{V(\varepsilon(0))}{\kappa} < T < T^*, \tag{3.65}$$

then the system can be stabilized within the time interval $[0, T]$ during which collision avoidance between any vehicles can be guaranteed. □

While the local formation stability is proved in Theorem 3.5, the convergence region of the equilibrium $\varepsilon = 0$ is not given. We next give a sufficient condition on $\varepsilon(0)$ to guarantee the convergence and collision avoidance.

Corollary 3.2. *Let* $\Delta_i \triangleq \min\{\theta_i^*, |\theta_i^* - \pi|, 2\pi - \theta_i^*\}$ *where* $i \in \{1, \ldots, n\}$. *There exists* ξ *such that* $0 < \xi < \min_i \Delta_i$. *Let* $\bar{\varepsilon}_i \triangleq \min\{|\cos(\theta_i^* + \xi) - \cos\theta_i^*|, |\cos(\theta_i^* - \xi) - \cos\theta_i^*|\}$, $\zeta \triangleq \min_i\{\theta_i^* - \xi, |\pi - \theta_i^*| - \xi, 2\pi - \theta_i^* - \xi\}$ *and* $\gamma \triangleq \sum_{i=1}^n \|e_i(0)\| + 4T$. *Under Assumptions 3.2 and 3.1, the proposed control law guarantees the convergence of* ε *to zero in* $[0, T]$ *with collision avoidance between any vehicles if*

$$V(\varepsilon(0)) < \min\left\{\min_i \bar{\varepsilon}_i, \frac{\sin^2\zeta\lambda_2(E^{\mathrm{T}}E)}{\gamma n}T\right\}. \tag{3.66}$$

Proof. The proof of Theorem 3.5 requires $\varepsilon(0)$ to be sufficiently small such that the following three conditions hold: (i) $\lambda_1(D^2) = \min_i [D^2]_{ii} > 0$ for all $\varepsilon \in \Omega(\varepsilon(0))$; (ii) θ_i^* and $\theta(t)$ for all $t \in [0, T]$ are both in $(0, \pi)$ or $(\pi, 2\pi)$; (iii) $V(\varepsilon(0))/\kappa < T$. Note condition (iii) ensures the collision avoidance.

Step 1: analyze condition (i). Recall $[D]_{ii} = (g_i^{\perp})^{\mathrm{T}} g_{i-1} = \sin\theta_i$ as proved in Lemma 3.8. Hence $\min_i [D^2]_{ii} > 0$ if $\theta_i(t) \neq 0$ and $\theta_i(t) \neq \pi$ for all $t \in [0, T]$. Thus condition (ii) implies condition (i).

Step 2: analyze condition (ii). Denote $\Delta_i \triangleq \min\{\theta_i^*, |\theta_i^* - \pi|, 2\pi - \theta_i^*\}$. There exists ξ such that $0 < \xi < \min_i \Delta_i$. Let $\bar{\varepsilon}_i \triangleq \min\{|\cos(\theta_i^* + \xi) - \cos\theta_i^*|, |\cos(\theta_i^* - \xi) - \cos\theta_i^*|\}$. Then we have the following sufficient condition: if $\varepsilon(0)$ satisfies

$$V(\varepsilon(0)) < \min_i \bar{\varepsilon}_i, \tag{3.67}$$

then condition (ii) holds. To see that, for any $j \in \{1, \ldots, n\}$, we have $|\varepsilon_j(t)| \leq \sum_{i=1}^n |\varepsilon_i(t)| = V(\varepsilon(t)) \leq V(\varepsilon(0)) < \min_i \bar{\varepsilon}_i \leq \bar{\varepsilon}_j$. Thus $|\varepsilon_j(t)| < \bar{\varepsilon}_j$ for all $t \in [0, T]$. Since the cosine function is monotone in $(0, \pi)$ or $(\pi, 2\pi)$, we have $|\varepsilon_j(t)| < \bar{\varepsilon}_j \implies |\theta_i(t) - \theta_i^*| < \xi$ and hence condition (ii) holds. It should be noted $\bar{\varepsilon}_i \neq 0$ and hence the set of $\varepsilon(0)$ that satisfies (3.67) is always nonempty.

Further define $\zeta \triangleq \min_i\{\theta_i^* - \xi, |\pi - \theta_i^*| - \xi, 2\pi - \theta_i^* - \xi\}$. Then $|\theta_i(t) - \theta_i^*| < \xi$ implies $\theta_i(t) > \zeta$, $|\pi - \theta_i(t)| > \zeta$ and $2\pi - \theta_i(t) > \zeta$ for all $t \in [0, T]$. Thus $[D^2]_{ii} = \sin^2\theta_i > \sin^2\zeta$. Hence we have $\beta = \sin^2\zeta$, where β is the lower bound of $\lambda_1(D^2)$ as defined in the proof of Theorem 3.5.

Step 3: analyze condition (iii). We first identify an upper bound of $\sum_{i=1}^n \|e_i\|$. Since the speed of each vehicle is bounded above by two. It is easy to see $\sum_{i=1}^n \|e_i(t)\| \leq \sum_{i=1}^n \|e_i(0)\| + 4T \triangleq \gamma$ for all $t \in [0, T]$. Therefore, we have κ

defined in (3.63) as

$$\kappa = \frac{\sin^2 \zeta \lambda_2(E^{\mathrm{T}}E)}{\gamma n},$$

substituting which into (3.65) yields

$$V(\varepsilon(0)) < \frac{\sin^2 \zeta \lambda_2(E^{\mathrm{T}}E)}{\gamma n} T.$$

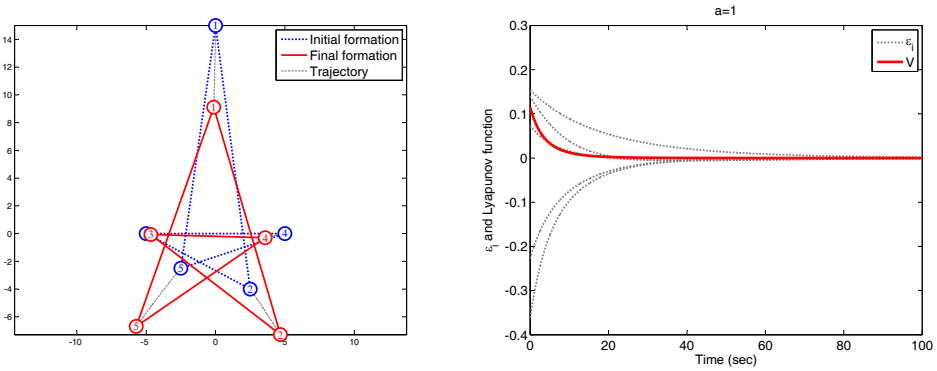
Therefore, if $V(\varepsilon(0))$ satisfies (3.66), then the three conditions are satisfied. By Theorem 3.5, the convergence of ε and collision avoidance between any vehicles can be guaranteed. □

Up to this point, the stability of the ε-dynamics has been proved. From control law (3.14), it is trivial to see that $\dot{z}_i = 0$ if $\varepsilon_i = 0$. Hence each vehicle will converge to a finite stationary finial position within finite time. Additionally, suppose the target formation is achieved at time $t_f < V(\varepsilon(0))/\kappa$. Since $\|\dot{z}_i\| \leq \|g_i - g_{i-1}\| \leq 2$, we have $\|z_i(t_f) - z_i(0)\| \leq 2t_f \leq 2V(\varepsilon(0))/\kappa$. Therefore, the final converged position $z_i(t_f)$ will be very close to its initial position $z_i(0)$ if the initial angle error $\varepsilon(0)$ is sufficiently small. In other words, the final converged formation will not be far away from the initial formation given small initial angle errors.

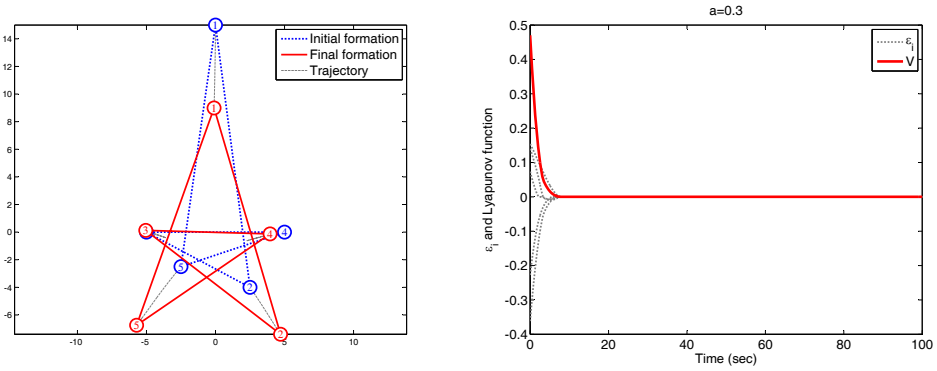
3.6 Simulation Results

Simulation results are presented in this section to verify the effectiveness and robustness of the proposed control law.

The desired formation in Figure 3.4 is a non-convex star polygon with $n = 5$. The angle at each vertex in the desired formation is $\theta_1^* = \cdots = \theta_5^* = 36$ deg. As can be seen, the proposed control law can effectively reduce the angle errors. The desired formation in Figure 3.5 is a ten-side polygon, where the angle at each vertex is $\theta_1^* = \cdots = \theta_{10}^* = 144$ deg. In the stability analysis, we assume the initial error $\varepsilon(0)$ is sufficiently small such that $\theta_i, \theta_i^* \in (0, \pi)$ or $(\pi, 2\pi)$ for all points in $\Omega(V(\varepsilon_0))$. However, this assumption is not satisfied in the example where $\theta_i(0) = \pi$ for $i = 2, 3, 7, 8, 10$ and $\theta_5(0) \in (\pi, 2\pi)$ but $\theta_5^* \in (0, \pi)$. As can



(a) $a = 1$

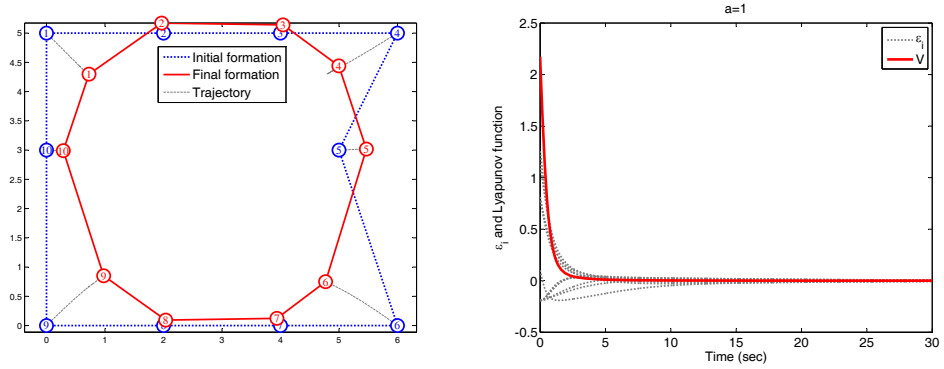


(b) $a = 0.3$

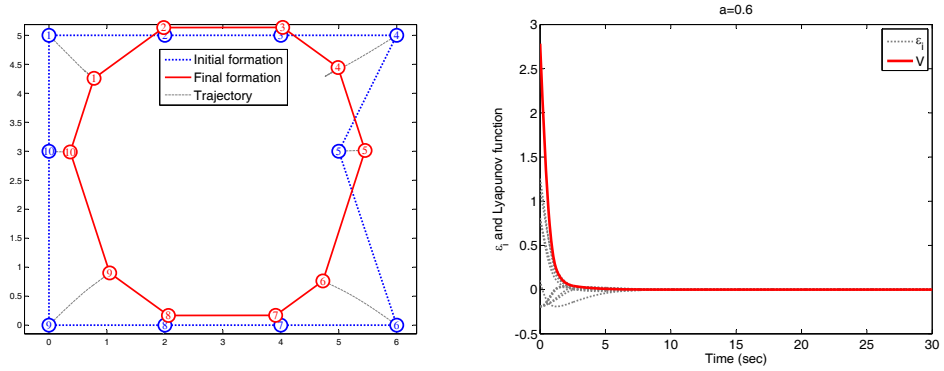
Figure 3.4: Formation and angle error evolution with $n = 5$ and $\theta_1^* = \cdots = \theta_n^* = 36$ deg.

be seen, the desired formation can still be achieved. The simulation suggests the convergence region of the desired formation by the proposed control law is not necessarily small. As shown in both Figure 3.4 and Figure 3.5, the angle errors and the Lyapunov function converge to zero in finite time if $a < 1$. Figures 3.6, 3.7, 3.8 and 3.9 respectively show the formation control of three, four, five and eight vehicles. As can be seen, the control law can stabilize the formation in finite time if $a = 0$.

Figure 3.10 demonstrates the robustness of the proposed control law against measurement noises and vehicle motion failure. In Figure 3.10(b), we add an error to each ε_i to simulate measurement noises. Each error is randomly drawn from a normal distribution with mean 0 and standard deviation 1. In Figure 3.10(b), vehicle 4 fails to move. As can be seen, the proposed control law still performs well in the presence of measurement noises or motion failure of one vehicle.

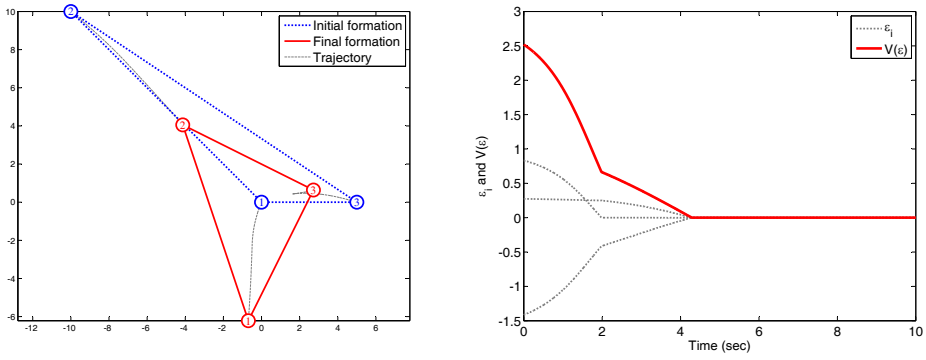


(a) $a = 1$



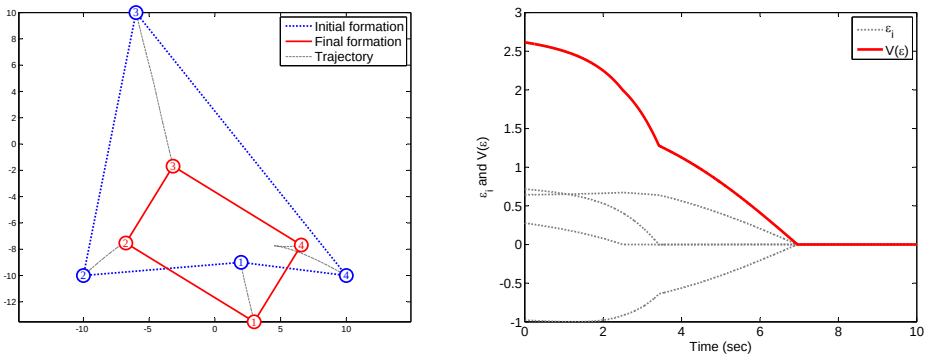
(b) $a = 0.6$

Figure 3.5: Formation and angle error evolution with $n = 10$ and $\theta_1^* = \cdots = \theta_n^* = 144$ deg.



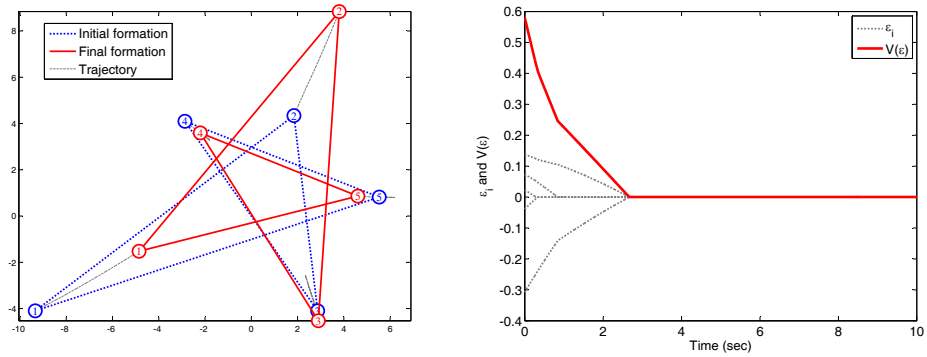
(a) Vehicle trajectory (b) Angle error and Lyapunov function

Figure 3.6: Control results by the proposed control law with $n = 3$, $\theta_1^* = \theta_2^* = 45$ deg and $\theta_3^* = 90$ deg.



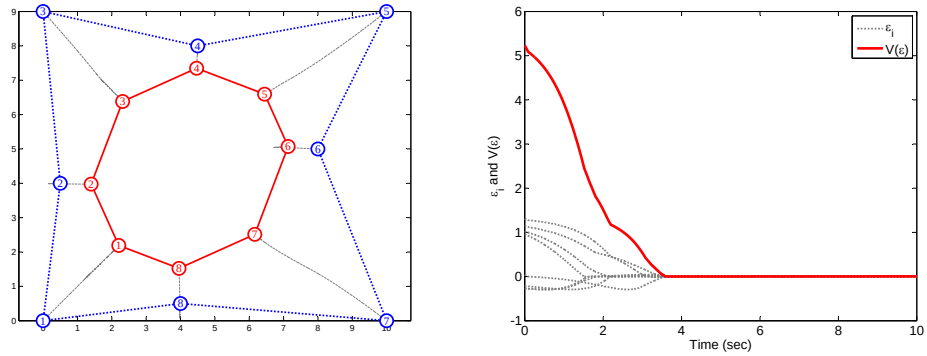
(a) Vehicle trajectory (b) Angle error and Lyapunov function

Figure 3.7: Control results by the proposed control law with $n = 4$ and $\theta_1^* = \cdots = \theta_4^* = 90$ deg.



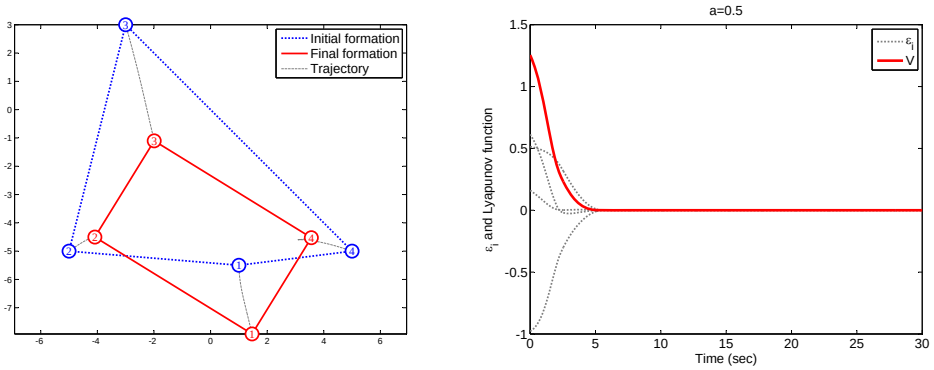
(a) Vehicle trajectory (b) Angle error and Lyapunov function

Figure 3.8: Control results by the proposed control law with $n = 5$ and $\theta_1^* = \cdots = \theta_5^* = 36$ deg.

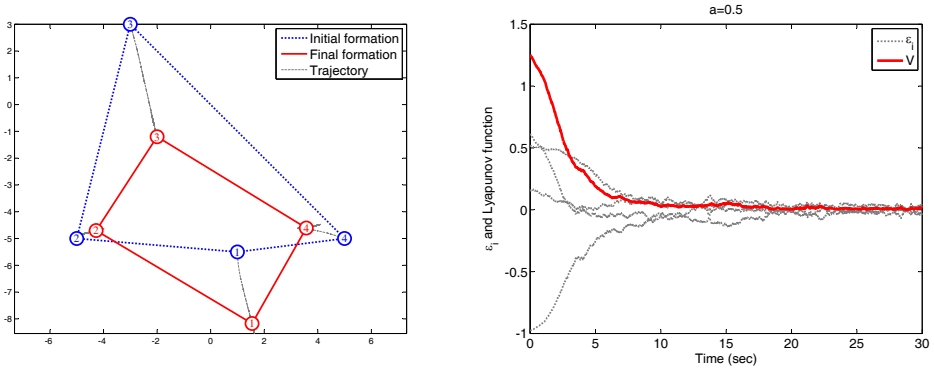


(a) Vehicle trajectory (b) Angle error and Lyapunov function

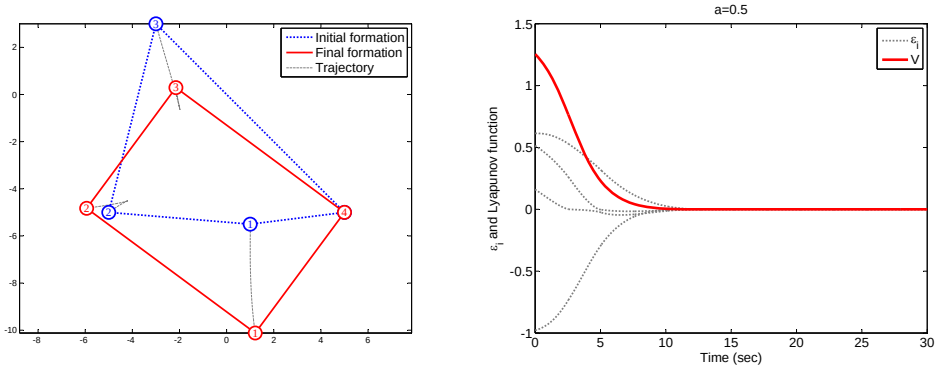
Figure 3.9: Control results by the proposed control law with $n = 8$ and $\theta_1^* = \cdots = \theta_8^* = 135$ deg.



(a) Ideal case



(b) In the presence of measurement noise



(c) In the presence of vehicle motion failure (Vehicle 4 fails to move)

Figure 3.10: An illustration of the robustness of the proposed control law against measurement noise and vehicle motion failure. $n = 4$ and $\theta_1^* = \cdots = \theta_4^* = 90$ deg.

Chapter 4

Vision-based Navigation using Natural Landmarks

4.1 Introduction

Chapters 2 and 3 addressed cooperative target tracking and vision-based formation control, respectively. It was shown that vision can be treated as a bearing-only sensor for these two topics. In this and the next chapters, we will study another important application of vision: vision-based navigation of UAVs. Vision will not be treated as a bearing-only sensor for navigation tasks. We need to extract as much information as possible from vision instead of just bearing information. In this chapter, we consider vision-based navigation using natural landmarks, while the next chapter will address vision-based navigation using artificial landmarks.

Computer vision techniques have been successfully applied to various UAV navigation tasks (see, for example, [16] and the references therein). Compared to inertial navigation systems, vision systems generally have two disadvantages. First, the reliability of vision systems usually is low. That is because vision systems heavily rely on environmental conditions. Inappropriate light condition or insufficient environmental features can easily cause failures of vision systems. Second, the update rate of vision systems usually is low. That is because image processing and vision algorithms usually are very time-consuming. As the on-board computers of small-scale UAVs typically have very limited computational

and storage resources, the update rate of onboard vision systems generally is much lower than that of IMUs. Therefore, since UAVs require reliable and high-update-rate state estimates, pure vision navigation is not suitable for navigation and control of UAVs. Hence in our navigation system we will adopt the vision-aided inertial navigation scheme.

In this chapter, we propose a homography-based vision-aided inertial navigation system for navigating small-scale UAVs in unmapped and GPS-denied environments. The proposed navigation system adopts four types of commonly used sensors: a low-cost IMU, a downward-looking monocular camera, a barometer and a compass. The measurements from different sensors are fused by an EKF. It is assumed the IMU measurements are corrupted by white noises and unknown constant biases. As a result, the position, velocity and attitude estimated by IMU dead reckoning would all drift rapidly. The main purpose of this work is to: firstly online estimate and compensate the unknown IMU biases; secondly provide drift-free velocity and attitude estimates which are crucial for UAV stabilization control; thirdly reduce the position drift significantly compared to pure inertial navigation. Since no global references such as maps are available, we do not aim at obtaining drift-free position estimates. But our vision-aided navigation system will be able to significantly reduce the position drift such that the UAV can perform acceptable long-duration navigation in unmapped and GPS-denied environments.

The vision measurement is a 3 by 3 homography matrix, which can be computed from pairs of consecutive images taken by the onboard camera. In contrast to the existing work [27, 18, 67], we avoid decomposing the homography to retrieve the useful information therein. Instead, the homography matrix is converted to a 9 by 1 vector and directly fed into the EKF as the vision measurement. By assuming the ground is a horizontal plane, we show that the homography plays a key role in drift-free velocity and attitude (specifically pitch and roll angles) estimation. The observability of the proposed navigation system is further analyzed. When the UAV speed is nonzero, we theoretically show the velocity, attitude and unknown constant biases are all observable. Comprehensive simulations and real flight experiments have been conducted to verify the

effectiveness and robustness of the proposed navigation system. The simulation and experimental results show the proposed navigation system can provide drift-free velocity and attitude estimates. The drift of the position estimate is also significantly reduced compared to pure inertial navigation.

This chapter is organized as follows. The homography-based vision-aided navigation system is designed in Section 4.2. Then the observability of the proposed navigation system is analyzed in Section 4.3. Simulation and real flight experimental results are shown in Sections 4.4 and 4.5, respectively.

Notations: Let $\mathbf{e}_i$ with $i \in \{1, 2, 3\}$ be the ith column of the 3 by 3 identity matrix $\mathbf{I}_{3\times 3}$. The operator vec converts a matrix to a column vector by stacking its column vectors below one another. Given a matrix A, the null space and range space of A are denoted as $\text{Null}\,(A)$ and $\text{Range}(A)$, respectively. For any $\mathbf{x} = [x_1, x_2, x_3]^{\mathrm{T}} \in \mathbb{R}^3$, the skew-symmetric matrix associated with $\mathbf{x}$ is denoted as

$$[\mathbf{x}]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}.$$

4.2 Design of the Vision-aided Navigation System

The navigation system contains two main sensors: an IMU and a monocular camera. The IMU measures the acceleration (also known as specific force) and angular rate of the UAV. We assume the IMU measurements are corrupted by zero-mean Gaussian white noises and unknown constant biases. The unknown biases may vary every time the IMU is initialized. Hence they need to be estimated online and then compensated in the navigation algorithm. The monocular camera is directed downward and can capture images of the ground scene during flight. The vision measurement, a 3 by 3 homography matrix, can be computed from each pair of consecutive images captured by the camera. We assume each of the entries of the homography matrix is corrupted by a zero-mean Gaussian white noise. In addition to the IMU and camera, we also consider two auxiliary sensors: a barometer and a compass, which can respectively measure the altitude

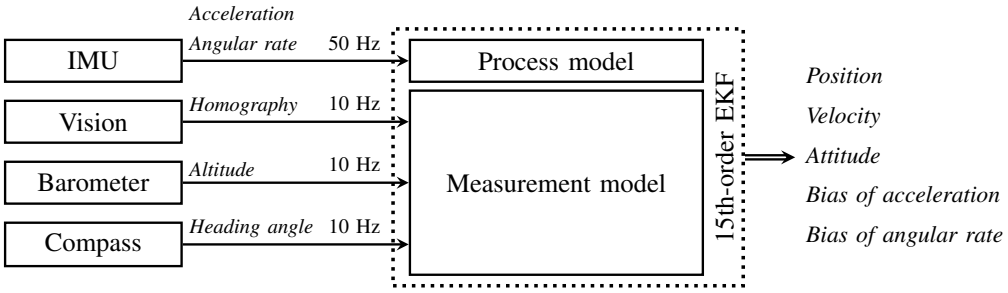


Figure 4.1: The structure of the proposed vision-aided navigation system.

and heading angle of the UAV.

The measurements of the above four types of sensors are fused by a 15th-order EKF. The 15 states to be estimated are: 3-dimensional position, 3-dimensional velocity, 3-dimensional attitude (roll, pitch and heading), 3-dimensional acceleration bias and 3-dimensional angular rate bias. In the rest of this section, we will develop the process model and the measurement model for the EKF. The IMU measurements enter the EKF through the process model as inputs; the measurements of the vision, barometer and compass enter the EKF through the measurement model. In our navigation system, the sampling rates of the IMU and the vision system are 50 Hz and 10 Hz, respectively. Denote T_{s} and T_{v} respectively as the sampling periods of the IMU and the vision system. Hence $T_{\mathrm{s}} = 0.02$ sec and $T_{\mathrm{v}} = 0.1$ sec. The structure of our navigation system is illustrated in Figure 4.1.

There are three reference frames in the navigation system: camera frame, body frame and navigation frame. For the sake of simplicity, we can install the camera on the UAV properly such that the axes of the camera frame are parallel to those of the body frame and the origins of the two frames are very close. Thus it can be assumed that the camera frame coincides with the body frame. The navigation frame is a local North-East-Down frame fixed on the ground plane. As the ground is assumed to be a horizontal plane, the ground plane coincides with the x-y plane of the navigation frame. The body and navigation frames are respectively denoted by subscripts b and n. We use a slash to represent a transformation from one frame to the other. For example, subscript n/b represents a transformation from the body frame to the navigation frame.

4.2.1 Process Model

We now design the process model of the vision-aided navigation system. Let $\mathbf{p}_{\mathrm{n}} = [p_{\mathrm{n},x}, p_{\mathrm{n},y}, p_{\mathrm{n},z}]^{\mathrm{T}} \in \mathbb{R}^3$ and $\mathbf{v}_{\mathrm{n}} = [v_{\mathrm{n},x}, v_{\mathrm{n},y}, v_{\mathrm{n},z}]^{\mathrm{T}} \in \mathbb{R}^3$ respectively denote the position and velocity of the UAV in the navigation frame. The attitude represented by Euler angles (roll, pitch and heading) is denoted as $\boldsymbol{\rho} = [\phi, \theta, \psi]^{\mathrm{T}} \in \mathbb{R}^3$. The kinematic model of the UAV is

$$\begin{bmatrix} \dot{\mathbf{p}}_{\mathrm{n}} \\ \dot{\mathbf{v}}_{\mathrm{n}} \\ \dot{\boldsymbol{\rho}} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{\mathrm{n}} \\ \mathbf{R}_{\mathrm{n/b}}\, \mathbf{a}_{\mathrm{nb}} + g\mathbf{e}_3 \\ \mathbf{L}_{\mathrm{n/b}}\, \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n}} \end{bmatrix}, \tag{4.1}$$

where $\mathbf{a}_{\mathrm{nb}} \in \mathbb{R}^3$ and $\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n}} \in \mathbb{R}^3$ respectively denote the acceleration and angular rate of the UAV; and g represents the local gravitational acceleration. The transformation matrices $\mathbf{R}_{\mathrm{n/b}}$ and $\mathbf{L}_{\mathrm{n/b}}$ are given by

$$\mathbf{R}_{\mathrm{n/b}} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \tag{4.2}$$

$$\mathbf{L}_{\mathrm{n/b}} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix},$$

where $s_* = \sin(*)$, $c_* = \cos(*)$ and $t_* = \tan(*)$.

Let $\mathbf{a}_{\mathrm{nb,IMU}}$ and $\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n,IMU}}$ be the IMU measurements of the acceleration and angular rate, respectively. Then we have

$$\mathbf{a}_{\mathrm{nb,IMU}} = \mathbf{a}_{\mathrm{nb}} - \mathbf{b}_a - \mathbf{w}_a, \tag{4.3}$$

$$\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n,IMU}} = \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n}} - \mathbf{b}_\omega - \mathbf{w}_\omega, \tag{4.4}$$

where $\mathbf{b}_a = [b_{a,x}, b_{a,y}, b_{a,z}]^{\mathrm{T}} \in \mathbb{R}^3$ and $\mathbf{b}_\omega = [b_{\omega,x}, b_{\omega,y}, b_{\omega,z}]^{\mathrm{T}} \in \mathbb{R}^3$ are unknown but constant measurement biases; and $\mathbf{w}_a \in \mathbb{R}^3$ and $\mathbf{w}_\omega \in \mathbb{R}^3$ are zero-mean Gaussian white noises.

Based on kinematic model (4.1), the position, velocity and attitude of the

UAV can be estimated by integrating $\mathbf{a}_{\text{nb,IMU}}$ and $\boldsymbol{\omega}^{\text{b}}_{\text{b/n,IMU}}$. This kind of manner is known as *dead reckoning*. However, the states estimated by dead reckoning would drift rapidly due to the measurement biases and noises. Note IMU measurement biases $\mathbf{b}_a$ and $\mathbf{b}_\omega$ may change every time the IMU is initialized. Hence they must be online estimated and compensated. To that end, we augment the state vector by adding the unknown biases. From (4.1), (4.3) and (4.4), the nonlinear process model of the navigation system is given by

$$\begin{bmatrix} \dot{\mathbf{p}}_{\text{n}} \\ \dot{\mathbf{v}}_{\text{n}} \\ \dot{\boldsymbol{\rho}} \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{b}}_\omega \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{\text{n}} \\ \mathbf{R}_{\text{n/b}}(\mathbf{a}_{\text{nb,IMU}} + \mathbf{b}_a + \mathbf{w}_a) + g\mathbf{e}_3 \\ \mathbf{L}_{\text{n/b}}(\boldsymbol{\omega}^{\text{b}}_{\text{b/n,IMU}} + \mathbf{b}_\omega + \mathbf{w}_\omega) \\ \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} \end{bmatrix},$$

which can be rewritten in a compact form as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u} + \mathbf{b} + \mathbf{w}), \tag{4.5}$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_{\text{n}} \\ \mathbf{v}_{\text{n}} \\ \boldsymbol{\rho} \\ \mathbf{b}_a \\ \mathbf{b}_\omega \end{bmatrix}, \quad \mathbf{f}(\mathbf{x}, \mathbf{u} + \mathbf{b} + \mathbf{w}) = \begin{bmatrix} \mathbf{f}_{\mathbf{p}} \\ \mathbf{f}_{\mathbf{v}} \\ \mathbf{f}_{\boldsymbol{\rho}} \\ \mathbf{f}_{\mathbf{b}_a} \\ \mathbf{f}_{\mathbf{b}_\omega} \end{bmatrix},$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{a}_{\text{nb,IMU}} \\ \boldsymbol{\omega}^{\text{b}}_{\text{b/n,IMU}} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_a \\ \mathbf{b}_\omega \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_\omega \end{bmatrix}. \tag{4.6}$$

Clearly the IMU measurements $\mathbf{a}_{\text{nb,IMU}}$ and $\boldsymbol{\omega}^{\text{b}}_{\text{b/n,IMU}}$ enter the process model as inputs. Let

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}, \quad \mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{w}}$$

be the Jacobians of $\mathbf{f}(\mathbf{x}, \mathbf{u} + \mathbf{b} + \mathbf{w})$ with respect to $\mathbf{x}$ and $\mathbf{w}$, respectively. It

can be calculated that

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{\partial \mathbf{f_v}}{\partial \boldsymbol{\rho}} & \mathbf{R}_{\mathrm{n/b}} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{\partial \mathbf{f}_{\boldsymbol{\rho}}}{\partial \boldsymbol{\rho}} & \mathbf{0}_{3\times3} & \mathbf{L}_{\mathrm{n/b}} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}_{15\times15}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{R}_{\mathrm{n/b}} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{L}_{\mathrm{n/b}} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}_{15\times6}, \tag{4.7}$$

where $\partial \mathbf{f_v}/\partial \boldsymbol{\rho}$ and $\partial \mathbf{f}_{\boldsymbol{\rho}}/\partial \boldsymbol{\rho}$ are given by

$$\frac{\partial \mathbf{f_v}}{\partial \boldsymbol{\rho}} = \begin{bmatrix} \frac{\partial \mathbf{R}_{\mathrm{n/b}}}{\partial \phi}\mathbf{a}_{\mathrm{nb}} & \frac{\partial \mathbf{R}_{\mathrm{n/b}}}{\partial \theta}\mathbf{a}_{\mathrm{nb}} & \frac{\partial \mathbf{R}_{\mathrm{n/b}}}{\partial \psi}\mathbf{a}_{\mathrm{nb}} \end{bmatrix}_{3\times3},$$
$$\frac{\partial \mathbf{f}_{\boldsymbol{\rho}}}{\partial \boldsymbol{\rho}} = \begin{bmatrix} \frac{\partial \mathbf{L}_{\mathrm{n/b}}}{\partial \phi}\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n}} & \frac{\partial \mathbf{L}_{\mathrm{n/b}}}{\partial \theta}\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n}} & \mathbf{0}_{3\times1} \end{bmatrix}_{3\times3},$$

with

$$\frac{\partial \mathbf{R}_{\mathrm{n/b}}}{\partial \phi} = \begin{bmatrix} 0 & c_\phi s_\theta c_\psi + s_\phi s_\psi & -s_\phi s_\theta c_\psi + c_\phi s_\psi \\ 0 & c_\phi s_\theta s_\psi - s_\phi c_\psi & -s_\phi s_\theta s_\psi - c_\phi c_\psi \\ 0 & c_\phi c_\theta & -s_\phi c_\theta \end{bmatrix},$$
$$\frac{\partial \mathbf{R}_{\mathrm{n/b}}}{\partial \theta} = \begin{bmatrix} -s_\theta c_\psi & s_\phi c_\theta c_\psi & c_\phi c_\theta c_\psi \\ -s_\theta s_\psi & s_\phi c_\theta s_\psi & c_\phi c_\theta s_\psi \\ -c_\theta & -s_\phi s_\theta & -c_\phi s_\theta \end{bmatrix},$$
$$\frac{\partial \mathbf{R}_{\mathrm{n/b}}}{\partial \psi} = \begin{bmatrix} -c_\theta s_\psi & -s_\phi s_\theta s_\psi - c_\phi c_\psi & -c_\phi s_\theta s_\psi + s_\phi c_\psi \\ c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ 0 & 0 & 0 \end{bmatrix},$$
$$\frac{\partial \mathbf{L}_{\mathrm{n/b}}}{\partial \phi} = \begin{bmatrix} 0 & c_\phi t_\theta & -s_\phi t_\theta \\ 0 & -s_\phi & -c_\phi \\ 0 & c_\phi/c_\theta & -s_\phi/c_\theta \end{bmatrix},$$
$$\frac{\partial \mathbf{L}_{\mathrm{n/b}}}{\partial \theta} = \begin{bmatrix} 1 & c_\phi/c_\theta^2 & -s_\phi/c_\theta^2 \\ 0 & 0 & 0 \\ 0 & c_\phi s_\theta/c_\theta^2 & -s_\phi s_\theta/c_\theta^2 \end{bmatrix}.$$

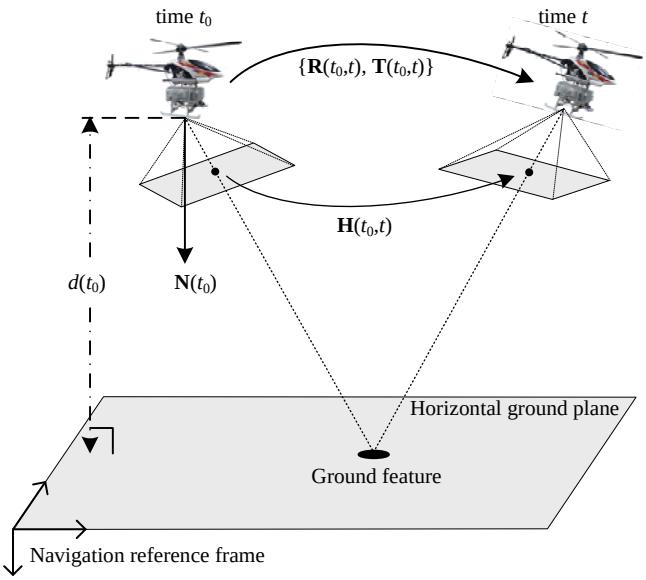


Figure 4.2: An illustration of the quantities $\mathbf{R}(t_0, t)$, $\mathbf{T}(t_0, t)$, $\mathbf{N}(t_0)$ and $d(t_0)$ in $\mathbf{H}(t_0, t)$.

4.2.2 Vision Measurement: Homography

The onboard camera is directed downward and can capture images of the ground scene during flight. The vision measurement, a 3 by 3 homography matrix, can be computed from two consecutive images. More specifically, let t and $t_0 = t - T_\mathrm{v}$ denote the current and the last sampling time instances, respectively. Given two images captured respectively at t_0 and t, the corresponding features of the two images are related by a homography matrix $\mathbf{H}(t_0, t) \in \mathbb{R}^{3\times3}$. Let $\mathbf{R}(t_0, t) \in \mathbb{R}^{3\times3}$ and $\mathbf{T}(t_0, t) \in \mathbb{R}^3$ respectively be the rotation and translation of the UAV from time t_0 to time t. Denote $\mathbf{N}(t_0) \in \mathbb{R}^3$ as the unit-length normal vector of the ground plane resolved in the camera frame at time t_0. Let $d(t_0) > 0$ be the distance between the UAV and the ground plane at time t_0. Since the ground plane coincides with the x-y plane of the navigation frame, $d(t_0)$ is the altitude of the UAV at time t_0. These variables are all illustrated in Figure 4.2.

For ease of presentation, the time variables in $\mathbf{H}(t_0, t)$, $\mathbf{R}(t_0, t)$, $\mathbf{T}(t_0, t)$, $\mathbf{N}(t_0)$ and $d(t_0)$ will be omitted in the sequel. The homography $\mathbf{H}$ is expressed by [82, Section 5.3] [56, Chapter 13]

$$\mathbf{H} = \mathbf{R} + \frac{1}{d}\mathbf{T}\mathbf{N}^\mathrm{T}. \tag{4.8}$$

Furthermore, $\mathbf{R}$, $\mathbf{T}$, $\mathbf{N}$ and d can be written by the UAV states as

$$\mathbf{R} = \mathbf{R}_{\mathrm{b/n}}(t)\mathbf{R}_{\mathrm{b/n}}^{\mathrm{T}}(t_0), \tag{4.9}$$

$$\mathbf{T} = \mathbf{R}_{\mathrm{b/n}}(t)\left[\mathbf{p}_{\mathrm{n}}(t_0) - \mathbf{p}_{\mathrm{n}}(t)\right], \tag{4.10}$$

$$\mathbf{N} = \mathbf{R}_{\mathrm{b/n}}(t_0)\mathbf{e}_3, \tag{4.11}$$

$$d = -\mathbf{e}_3^{\mathrm{T}}\mathbf{p}_{\mathrm{n}}(t_0), \tag{4.12}$$

where $\mathbf{R}_{\mathrm{b/n}}$ denotes the rotation from the navigation frame to the body frame and $\mathbf{R}_{\mathrm{b/n}} = \mathbf{R}_{\mathrm{n/b}}^{\mathrm{T}}$. It is notable that (4.11) and (4.12) are valid only if the ground plane is horizontal. As shown by (4.8) to (4.12), the homography $\mathbf{H}$ clearly is a nonlinear function of the UAV states at both time t and t_0.

We next qualitatively analyze the useful information carried by a homography matrix.

Attitude Information

In (4.8) both $\mathbf{R}$ and $\mathbf{N}$ contain certain attitude information of the UAV. First, $\mathbf{R}$ represents the rotation of the UAV from t_0 to t. The UAV attitude can be estimated by accumulating $\mathbf{R}$. But the attitude estimated in that way will drift because of the accumulating error. Second, $\mathbf{N}$ represents the normal vector of the horizontal ground plane resolved in the camera frame. From (4.2) and (4.11), we have

$$\mathbf{N} = \begin{bmatrix} -s_\theta \\ s_\phi c_\theta \\ c_\phi c_\theta \end{bmatrix}.$$

Clearly $\mathbf{N}$ contains the roll and pitch angles. Thus it is possible to obtain drift-free estimates of the roll and pitch angles from $\mathbf{N}$. In our work we will use $\mathbf{N}$ instead of $\mathbf{R}$ for drift-free attitude estimation. Note the heading angle is not contained in the homography. Hence it is impossible to obtain drift-free estimates of the heading angle from the homography. In our navigation system, the heading angle will be directly measured by a compass, which is a commonly used sensor for UAVs nowadays.

Velocity Information

The term $\mathbf{T}$ in (4.8) represents the translation of the UAV from time t_0 to t, and hence can be used to estimate the UAV velocity. But it can be observed from (4.10) that $\mathbf{T}$ is expressed in the camera frame. In order to obtain the UAV velocity in the navigation frame, we need to convert $\mathbf{T}$ to the navigation frame. Note the rotation from the camera frame (i.e., body frame here) to the navigation frame is described by the UAV attitude. Hence drift-free velocity estimation requires drift-free attitude estimation. Another issue regarding the velocity estimation is the altitude d. From (4.8), the magnitude of $\mathbf{T}$ will be inaccurately scaled if d is inaccurate. Thus drift-free velocity estimation also requires accurate altitude measurements. Since the altitude is also crucial for the flight safety of the UAV, we will use a barometer to directly measure the altitude of the UAV.

Up to this point, it is clear what information is carried by a homography. The next is how to retrieve the information. One conventional method is to directly decompose $\mathbf{R}$, $\mathbf{T}$ and $\mathbf{N}$ from $\mathbf{H}$. Interested readers may refer to [82, Section 5.3.3] for homography decomposition algorithms. A decomposition would give two physically possible solutions. Prior knowledge of the UAV motion may be required to disambiguate the two solutions. On the other hand, a homography is computed from feature matchings of two images. Hence feature location errors or mismatchings would certainly cause errors in the homography. These errors would propagate through the decomposition procedure to the decomposed $\mathbf{R}$, $\mathbf{T}$ and $\mathbf{N}$. The error propagation may become very severe in certain situations [27]. Hence the errors in the homography matrix must be treated in a proper manner.

In our navigation system, we can avoid the homography decomposition by fully utilizing the IMU measurements. Note the angular rate of the UAV can be measured by the IMU. As will be shown later, the term $\mathbf{R}$ can be computed accurately using the angular rate measurements. By eliminating $\mathbf{R}$ from $\mathbf{H}$, we can obtain $\mathbf{T}\mathbf{N}^{\mathrm{T}}/d$, which is a rank-one matrix. Of course, one may use low rank approximation techniques to decompose $\mathbf{T}/d$ and $\mathbf{N}$. However, due to the measurement errors, the low rank decomposition would be ill-posed when

the translation $\mathbf{T}$ is very small. In the extreme case of $\mathbf{T} = \mathbf{0}$, it would be theoretically impossible to retrieve $\mathbf{N}$. Hence in our work we will directly feed the measurements of $\mathbf{H}$ and $\mathbf{R}$ into the EKF without any decompositions. In that way, the measurement errors can be well handled by the EKF.

4.2.3 Measurement Model

We now present the measurement model of the vision-aided navigation system. Let $\mathbf{H}$ and $\mathbf{H}_{\text{vis}}$ denote the true homography and the homography estimated by the vision system, respectively. In our navigation system, we convert $\mathbf{H}_{\text{vis}}$ to vec$\mathbf{H}_{\text{vis}} \in \mathbb{R}^9$ and then directly feed it into the EKF. Thus the nonlinear measurement model of the vision system is

$$\mathbf{y}_{\text{vis}} = \text{vec}\mathbf{H}_{\text{vis}} = \text{vec}\mathbf{H} + \mathbf{n}_{\text{vis}},$$

where $\mathbf{n}_{\text{vis}} \in \mathbb{R}^9$ is assumed to be a zero-mean Gaussian white noise.

From (4.8) to (4.12), it is clear vec$\mathbf{H}$ is a nonlinear function of both $\mathbf{x}(t)$ and $\mathbf{x}(t_0)$. Since the state vector in the process model is $\mathbf{x}(t)$, we need to express vec$\mathbf{H}$ as a nonlinear function of $\mathbf{x}(t)$ only. Otherwise, if the state vector of the EKF contains both $\mathbf{x}(t)$ and $\mathbf{x}(t_0)$, the EKF will be not only high-order but also very complicated. To that end, we propose the following approximations to express vec$\mathbf{H}$ as a function of $\mathbf{x}(t)$.

Approximating R

We now show $\mathbf{R}$ can be accurately computed from the angular rate measurements of the IMU. The dynamics of $\mathbf{R}_{\text{b/n}}$ is given by

$$\dot{\mathbf{R}}_{\text{b/n}} = -\left[\boldsymbol{\omega}^{\text{b}}_{\text{b/n}}\right]_{\times} \mathbf{R}_{\text{b/n}}. \tag{4.13}$$

Recall $\mathbf{R} = \mathbf{R}_{\text{b/n}}(t)\mathbf{R}^{\text{T}}_{\text{b/n}}(t_0)$ as shown in (4.9). Hence $\mathbf{R}$ is irrelevant to the UAV states, and can be computed from $\boldsymbol{\omega}^{\text{b}}_{\text{b/n}}$ according to (4.13). However, equation (4.13) has no closed-form solutions unless $\boldsymbol{\omega}^{\text{b}}_{\text{b/n}}$ is (piecewise) constant. Since the sampling period $T_{\text{v}} = 0.1$ sec is very short, the angular rate can be treated

as constant over the time interval $[t_0, t]$. Then $\mathbf{R}$ can be approximated by

$$\begin{aligned}\mathbf{R} &\approx \exp\left\{-\left[\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n}}(t)\right]_{\times} T_{\mathrm{v}}\right\} \\ &\approx \exp\left\{-\left[\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n,IMU}}(t)\right]_{\times} T_{\mathrm{v}}\right\}. \end{aligned} \tag{4.14}$$

Remark 4.1. *The approximation error of* $\mathbf{R}$ *in* (4.14) *is caused by the measurement errors in* $\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n},IMU}(t)$ *and the assumption that* $\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n}}$ *is constant over* $[t_0, t]$. *But the approximation of* $\mathbf{R}$ *given by* (4.14) *can be very accurate because the time interval* $T_v = 0.1$ *sec is very short and the measurement errors in* $\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{b/n},IMU}(t)$ *usually are small.*

Remark 4.2. *The error of* $\mathbf{R}$ *will result in small errors in the estimates of the velocity and attitude. But the error will* not *cause drift in these estimates.*

Remark 4.3. *The matrix exponential on the right hand side of* (4.14) *can be rigorously computed by Rodrigues' rotation formula [82, Theorem 2.9, p. 27].*

Approximating T

Assume the UAV velocity to be constant over the time interval $[t_0, t]$. Then $\mathbf{p}_{\mathrm{n}}(t) - \mathbf{p}_{\mathrm{n}}(t_0) \approx \mathbf{v}_{\mathrm{n}}(t) T_{\mathrm{v}}$ and hence $\mathbf{T}$ in (4.10) can be approximated by

$$\mathbf{T} \approx -\mathbf{R}_{\mathrm{b/n}}(t)\mathbf{v}_{\mathrm{n}}(t) T_{\mathrm{v}}. \tag{4.15}$$

Approximating N

Recall $\mathbf{N}$ is the normal vector of the ground plane at time t_0. Since $\mathbf{R}_{\mathrm{b/n}}(t_0) = \mathbf{R}^{\mathrm{T}}\mathbf{R}_{\mathrm{b/n}}(t)$, we have $\mathbf{N}$ in (4.11) as

$$\mathbf{N} \approx \mathbf{R}^{\mathrm{T}}\mathbf{R}_{\mathrm{b/n}}(t)\mathbf{e}_3, \tag{4.16}$$

where $\mathbf{R}$ is given by (4.14).

Approximating d

Recall d is the altitude of the UAV at time t_0. Since $\mathbf{p}_{\mathrm{n}}(t) - \mathbf{p}_{\mathrm{n}}(t_0) \approx \mathbf{v}_{\mathrm{n}}(t)T_{\mathrm{v}}$, we can approximately write d in (4.12) as

$$d \approx -\mathbf{e}_3^{\mathrm{T}} \left[\mathbf{p}_{\mathrm{n}}(t) - \mathbf{v}_{\mathrm{n}}(t)T_{\mathrm{v}}\right]. \tag{4.17}$$

By using the above approximations of $\mathbf{R}$, $\mathbf{T}$, $\mathbf{N}$ and d, we can express vec$\mathbf{H}$ as a nonlinear function of $\mathbf{x}(t)$ only. With a little abuse of notation, rewrite the state vector of the EKF as $\mathbf{x} = [x_1, ..., x_{15}]^{\mathrm{T}}$. Then the Jacobian of vec$\mathbf{H}$ with respect to $\mathbf{x}$ is

$$\begin{aligned}
\mathbf{C}_{\mathrm{vis}} &= \frac{\partial \mathrm{vec}\mathbf{H}}{\partial \mathbf{x}} \\
&= \left[\frac{\partial \mathrm{vec}\mathbf{H}}{\partial x_1}, \ldots, \frac{\partial \mathrm{vec}\mathbf{H}}{\partial x_{15}}\right]_{9\times 15} \\
&= \left[\mathrm{vec}\frac{\partial \mathbf{H}}{\partial x_1}, \ldots, \mathrm{vec}\frac{\partial \mathbf{H}}{\partial x_{15}}\right]_{9\times 15},
\end{aligned} \tag{4.18}$$

where $\partial\mathbf{H}/\partial x_i$ for $i = 1, \ldots, 15$ is given by

$$\frac{\partial \mathbf{H}}{\partial x_i} = -\frac{1}{d^2}\frac{\partial d}{\partial x_i}\mathbf{T}\mathbf{N}^{\mathrm{T}} + \frac{1}{d}\left[\frac{\partial \mathbf{T}}{\partial x_i}\mathbf{N}^{\mathrm{T}} + \mathbf{T}\left(\frac{\partial \mathbf{N}}{\partial x_i}\right)^{\mathrm{T}}\right]. \tag{4.19}$$

The partial derivatives on the right hand side of (4.19) can be calculated from (4.15), (4.16) and (4.17). These partial derivatives are listed as below.

$$\begin{aligned}
\frac{\partial \mathbf{H}}{\partial x_{\mathrm{n}}} &= \mathbf{0}_{3\times 3}, \\
\frac{\partial \mathbf{H}}{\partial y_{\mathrm{n}}} &= \mathbf{0}_{3\times 3}, \\
\frac{\partial \mathbf{H}}{\partial z_{\mathrm{n}}} &= \frac{1}{d^2}\mathbf{T}\mathbf{N}^{\mathrm{T}}, \\
\frac{\partial \mathbf{H}}{\partial v_{\mathrm{n},x}} &= -\frac{T_{\mathrm{v}}}{d}\mathbf{R}_{\mathrm{b/n}}\mathbf{e}_1\mathbf{N}^{\mathrm{T}}, \\
\frac{\partial \mathbf{H}}{\partial v_{\mathrm{n},y}} &= -\frac{T_{\mathrm{v}}}{d}\mathbf{R}_{\mathrm{b/n}}\mathbf{e}_2\mathbf{N}^{\mathrm{T}}, \\
\frac{\partial \mathbf{H}}{\partial v_{\mathrm{n},z}} &= -\frac{T_{\mathrm{v}}}{d}\mathbf{R}_{\mathrm{b/n}}\mathbf{e}_3\mathbf{N}^{\mathrm{T}} - \frac{T_{\mathrm{v}}}{d^2}\mathbf{T}\mathbf{N}^{\mathrm{T}},
\end{aligned}$$

$$\frac{\partial \mathbf{H}}{\partial \phi} = \frac{1}{d}\left(-\frac{\partial \mathbf{R}_{\mathrm{b/n}}}{\partial \phi}\mathbf{v}_{\mathrm{n}} T_{\mathrm{v}} \mathbf{N}^{\mathrm{T}} + \mathbf{T} \begin{bmatrix} 0 \\ c_\theta c_\phi \\ -c_\theta s_\phi \end{bmatrix}^{\mathrm{T}} \mathbf{R} \right),$$

$$\frac{\partial \mathbf{H}}{\partial \theta} = \frac{1}{d}\left(-\frac{\partial \mathbf{R}_{\mathrm{b/n}}}{\partial \theta}\mathbf{v}_{\mathrm{n}} T_{\mathrm{v}} \mathbf{N}^{\mathrm{T}} + \mathbf{T} \begin{bmatrix} -c_\theta \\ -s_\phi s_\theta \\ -c_\phi s_\theta \end{bmatrix}^{\mathrm{T}} \mathbf{R} \right),$$

$$\frac{\partial \mathbf{H}}{\partial \psi} = \frac{1}{d}\left(-\frac{\partial \mathbf{R}_{\mathrm{b/n}}}{\partial \psi}\mathbf{v}_{\mathrm{n}} T_{\mathrm{v}} \mathbf{N}^{\mathrm{T}} \right).$$

The other derivatives not listed above are all zero.

In addition to the vision measurement model, the measurement models of the compass and barometer are respectively given by

$$y_{\mathrm{comp}} = \psi + n_{\mathrm{comp}} = \mathbf{C}_{\mathrm{comp}}\mathbf{x} + n_{\mathrm{comp}},$$

$$y_{\mathrm{baro}} = p_{\mathrm{n},z} + n_{\mathrm{baro}} = \mathbf{C}_{\mathrm{baro}}\mathbf{x} + n_{\mathrm{baro}},$$

where $n_{\mathrm{comp}} \in \mathbb{R}$ and $n_{\mathrm{baro}} \in \mathbb{R}$ are assumed to be zero-mean Gaussian white noises and

$$\mathbf{C}_{\mathrm{comp}} = \begin{bmatrix} \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix}_{1\times 15}, \tag{4.20}$$

$$\mathbf{C}_{\mathrm{baro}} = \begin{bmatrix} \mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix}_{1\times 15}. \tag{4.21}$$

To sum up, the nonlinear measurement model of the vision-aided navigation system is

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{n}, \tag{4.22}$$

where

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_{\mathrm{vis}} \\ y_{\mathrm{comp}} \\ y_{\mathrm{baro}} \end{bmatrix}_{11\times 1}, \quad \mathbf{h}(\mathbf{x}) = \begin{bmatrix} \mathrm{vec}\mathbf{H} \\ \psi \\ p_{\mathrm{n},z} \end{bmatrix}_{11\times 1}, \quad \mathbf{n} = \begin{bmatrix} \mathbf{n}_{\mathrm{vis}} \\ n_{\mathrm{comp}} \\ n_{\mathrm{baro}} \end{bmatrix}_{11\times 1}. \tag{4.23}$$

The Jacobian of $\mathbf{h}(\mathbf{x})$ with respect to $\mathbf{x}$ is given by

$$\mathbf{C} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{C}_{\text{vis}} \\ \mathbf{C}_{\text{comp}} \\ \mathbf{C}_{\text{baro}} \end{bmatrix}_{11\times 15}. \tag{4.24}$$

4.2.4 Extended Kalman Filtering

We have established the continuous process model (4.5) and the measurement model (4.22). The corresponding Jacobians have been given in (4.7) and (4.24). Now it is ready to apply the EKF to fuse the measurements of the IMU, vision, compass and barometer. The procedure of the EKF is standard, but we list the equations here for readers' reference.

1) Predict state estimation:

$$\hat{\mathbf{x}}_{k,k-1} = \hat{\mathbf{x}}_{k-1} + T_{\text{s}}\mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1} + \hat{\mathbf{b}}_{k-1} + \mathbf{0}),$$

where $\mathbf{x}$ and $\mathbf{f}$ are given in (4.6).

2) Predict error covariance:

$$\mathbf{P}_{k,k-1} = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^{\text{T}} + \mathbf{G}_k \mathbf{Q}_{\text{cov}} \mathbf{G}_k^{\text{T}},$$

where the state transition matrix $\mathbf{F}$ is

$$\mathbf{F} = \mathbf{I} + T_{\text{s}}\mathbf{A},$$

and the noise input matrix $\mathbf{G}$ is

$$\mathbf{G} = T_{\text{s}}\mathbf{B}.$$

The matrices $\mathbf{A}$ and $\mathbf{B}$ are given in (4.7), and $\mathbf{Q}_{\text{cov}}$ is the covariance of the process noise $\mathbf{w}$ in (4.6).

3) Compute the Kalman gain:

$$\mathbf{K}_k = \mathbf{P}_{k,k-1}\mathbf{C}_k^{\mathrm{T}}(\mathbf{C}_k\mathbf{P}_{k,k-1}\mathbf{C}_k^{\mathrm{T}} + \mathbf{R}_{\mathrm{cov}})^{-1},$$

where $\mathbf{C}$ is given in (4.24), and $\mathbf{R}_{\mathrm{cov}}$ is the covariance of the measurement noise $\mathbf{n}$ in (4.23).

4) Update state estimation:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k,k-1} + \mathbf{K}_k[\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k,k-1})],$$

where $\mathbf{y}$ and $\mathbf{h}$ are given in (4.23).

5) Update error covariance:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\mathbf{P}_{k,k-1}.$$

In practice, light condition changing or insufficient features of the ground scene can cause extremely large homography estimation errors. These inaccurate homography estimates must be detected and rejected. Otherwise, they may cause large errors or even instability of the EKF. Motivated by that, we adopt *innovation filtering* [53, Section 15.3] in our navigation system. Innovation filtering is also called spike filtering, measurement gating or prefiltering. Its basic idea is to compare the real measurement given by sensors with the predicted one given by EKF. If the discrepancy between them exceeds a threshold, then the real measurement for that iteration is rejected. The innovation filter is straightforward to implement and requires little computational resource. In our flight experiments, innovation filtering can effectively detect spikes of vision measurements and hence significantly improve the robustness of the navigation system. Details of innovation filtering are omitted here. Interested readers may refer to [53, Section 15.3].

4.3 Observability Analysis of the Vision-aided Navigation System

In this section we analyze the observability of the proposed navigation system. Since process model (4.5) and measurement model (4.22) are highly nonlinear, we will mainly consider the observability of linearized systems. More specifically, the nonlinear models (4.5) and (4.22) will be linearized under two typical flight conditions: hovering and straight and steady level (SSL) flight. Note hovering is a flight mode only possible for rotorcrafts such as quadrotor UAVs.

When the UAV is in SSL flight or hovering condition, the UAV states are approximately given by

$$\begin{aligned} &\phi = \theta = \psi = 0, \\ &\boldsymbol{\omega}_{\mathrm{b/n}}^{\mathrm{b}} = \mathbf{0}_{3\times 1}, \\ &\mathbf{a}_{\mathrm{nb}} = -g\mathbf{e}_3, \\ &\mathbf{v}_{\mathrm{n}} = \kappa\mathbf{e}_1, \end{aligned} \tag{4.25}$$

where $\kappa \geq 0$ represents the UAV speed. When $\kappa > 0$, the UAV flies to the north with the speed as κ; when $\kappa = 0$, the UAV is hovering. As will be shown later, the value of the UAV speed κ can affect the observability of the system. Due to symmetry, the value of the heading angle has no influence on the observability analysis. For the sake of simplicity, we choose $\psi = 0$ which means the UAV is heading north.

Remark 4.4. *Although* (4.25) *is a simplified condition, the observability analysis under this condition can characterize the observability of the system under generic flight conditions. That will be verified later by simulation and flight experimental results.*

Substituting the linearized condition (4.25) into (4.7) gives

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & g[\mathbf{e}_3]_\times & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}. \tag{4.26}$$

Substituting the linearized condition (4.25) into (4.14), (4.15) and (4.16) yields $\mathbf{R} = \mathbf{I}_{3\times3}$, $\mathbf{T} = -T_\mathrm{v}\mathbf{v}_\mathrm{n}$ and $\mathbf{N} = \mathbf{e}_3$, respectively. Further substituting the values of $\mathbf{R}$, $\mathbf{T}$ and $\mathbf{N}$ into (4.18) gives

$$\mathbf{C}_\mathrm{vis} = \frac{1}{\alpha}\begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\kappa\mathbf{e}_1\mathbf{e}_2^\mathrm{T} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \kappa\mathbf{e}_1\mathbf{e}_1^\mathrm{T} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \kappa[\mathbf{e}_1]_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}. \tag{4.27}$$

where $\alpha = -d/T_\mathrm{v}$. While calculating (4.27), we omit the small terms containing T_v^2/d^2 or T_v/d^2 considering T_v is small. The detailed derivation of (4.27) is given as below. Substituting the condition (4.25) into the partial derivatives of $\mathbf{H}$ gives

$$\begin{aligned} \frac{\partial\mathbf{H}}{\partial v_{\mathrm{n},x}} &= -\frac{T_\mathrm{v}}{d}\mathbf{e}_1\mathbf{e}_3^T, \\ \frac{\partial\mathbf{H}}{\partial v_{\mathrm{n},y}} &= -\frac{T_\mathrm{v}}{d}\mathbf{e}_2\mathbf{e}_3^T, \\ \frac{\partial\mathbf{H}}{\partial v_{\mathrm{n},z}} &= -\frac{T_\mathrm{v}}{d}\mathbf{e}_3\mathbf{e}_3^T, \\ \frac{\partial\mathbf{H}}{\partial\phi} &= -\frac{T_\mathrm{v}}{d}\left(-[\mathbf{e}_1]_\times\mathbf{v}_\mathrm{n}\mathbf{e}_3^\mathrm{T} + \mathbf{v}_\mathrm{n}\mathbf{e}_2^\mathrm{T}\right), \\ \frac{\partial\mathbf{H}}{\partial\theta} &= -\frac{T_\mathrm{v}}{d}\left(-[\mathbf{e}_2]_\times\mathbf{v}_\mathrm{n}\mathbf{e}_3^\mathrm{T} - \mathbf{v}_\mathrm{n}\mathbf{e}_1^\mathrm{T}\right), \\ \frac{\partial\mathbf{H}}{\partial\psi} &= -\frac{T_\mathrm{v}}{d}\left(-[\mathbf{e}_3]_\times\mathbf{v}_\mathrm{n}\mathbf{e}_3^\mathrm{T}\right). \end{aligned}$$

Thus

$$\begin{bmatrix} \mathrm{vec}\dfrac{\partial\mathbf{H}}{\partial v_{\mathrm{n},x}} & \mathrm{vec}\dfrac{\partial\mathbf{H}}{\partial v_{\mathrm{n},y}} & \mathrm{vec}\dfrac{\partial\mathbf{H}}{\partial v_{\mathrm{n},z}} \end{bmatrix} = -\frac{T_\mathrm{v}}{d}\begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} \end{bmatrix},$$

$$\begin{bmatrix} \operatorname{vec}\dfrac{\partial \mathbf{H}}{\partial \phi} & \operatorname{vec}\dfrac{\partial \mathbf{H}}{\partial \theta} & \operatorname{vec}\dfrac{\partial \mathbf{H}}{\partial \psi} \end{bmatrix} = -\frac{T_{\mathrm{v}}}{d}\begin{bmatrix} \mathbf{0}_{3\times 1} & -\mathbf{v}_{\mathrm{n}} & \mathbf{0}_{3\times 1} \\ \mathbf{v}_{\mathrm{n}} & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \\ -[\mathbf{e}_1]_\times \mathbf{v}_{\mathrm{n}} & -[\mathbf{e}_2]_\times \mathbf{v}_{\mathrm{n}} & -[\mathbf{e}_3]_\times \mathbf{v}_{\mathrm{n}} \end{bmatrix}$$
$$= -\frac{T_{\mathrm{v}}}{d}\begin{bmatrix} -\mathbf{v}_{\mathrm{n}}\mathbf{e}_2^{\mathrm{T}} \\ \mathbf{v}_{\mathrm{n}}\mathbf{e}_1^{\mathrm{T}} \\ [\mathbf{v}_{\mathrm{n}}]_\times \end{bmatrix}.$$

All the other entries in $\mathbf{C}_{\mathrm{vis}}$ are zero. Hence we obtain (4.27).

Recall $\mathbf{C}_{\mathrm{comp}}$ and $\mathbf{C}_{\mathrm{baro}}$ are given (4.20) and (4.21). Then $\mathbf{C}$ can be obtained by substituting (4.20), (4.21) and (4.27) into (4.24). We next analyze the rank of the observability matrix

$$\mathcal{O} = \left[\mathbf{C}^{\mathrm{T}}, (\mathbf{CA})^{\mathrm{T}}, \cdots, (\mathbf{CA}^{14})^{\mathrm{T}}\right]^{\mathrm{T}} \tag{4.28}$$

under the condition (4.25).

4.3.1 Case 1: SSL Flight

We first consider the SSL flight condition with $\kappa > 0$. Substituting (4.26), (4.27), (4.20) and (4.21) into (4.28) yields the observability matrix

$$\mathcal{O}_{\mathrm{SSL}} = \left[\begin{array}{ccccc} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\kappa\mathbf{e}_1\mathbf{e}_2^{\mathrm{T}} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \kappa\mathbf{e}_1\mathbf{e}_1^{\mathrm{T}} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} & \kappa[\mathbf{e}_1]_\times & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ \hdashline \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\kappa\mathbf{e}_1\mathbf{e}_2^{\mathrm{T}} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \kappa\mathbf{e}_1\mathbf{e}_1^{\mathrm{T}} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & g[\mathbf{e}_3]_\times & \mathbf{I}_{3\times 3} & \kappa[\mathbf{e}_1]_\times \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \alpha\mathbf{e}_3^{\mathrm{T}} \\ \mathbf{0}_{1\times 3} & \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ \hdashline \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & g[\mathbf{e}_3]_\times \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times 3} \end{array}\right]_{26\times 15}. \tag{4.29}$$

The scale factor $1/\alpha$ and zero rows of $\mathcal{O}_{\text{SSL}}$ are omitted as they do not contribute to the rank of $\mathcal{O}_{\text{SSL}}$. By examining the row rank (or column rank), it is straightforward to see

$$\text{rank}(\mathcal{O}_{\text{SSL}}) = 13.$$

Hence there are *two* unobservable modes. In order to identify the two unobservable modes, we need to determine the unobservable subspace, i.e., the null space of $\mathcal{O}_{\text{SSL}}$. By observation we obtain an orthogonal basis of the unobservable subspace as

$$\text{Null}\left(\mathcal{O}_{\text{SSL}}\right) = \text{Range}\begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix}_{15\times2}. \tag{4.30}$$

Form (4.30), the two unobservable modes are

$$\mathbf{x}_{\text{SSL}}^{\text{unobs}} = \{p_{\text{n},x},\, p_{\text{n},y}\}.$$

Therefore, in the case of SSL flight, the position ($p_{\text{n},x}$ and $p_{\text{n},y}$) is unobservable, but the velocity, attitude and unknown biases are all observable. In fact, this is the best situation we can have because it is impossible to make the position observable without any global references such as maps or GPS. Although the position estimate still drifts, the drift will be significantly reduced compared to pure inertial navigation as will be shown later.

4.3.2 Case 2: Hovering

We now consider the hovering condition with $\kappa = 0$. By substituting $\kappa = 0$ into (4.29), $\mathcal{O}_{\text{SSL}}$ degenerates to

$$\mathcal{O}_{\text{hover}} = \left[\begin{array}{ccccc} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\text{T}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \alpha\mathbf{e}_3^{\text{T}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \hdashline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & g[\mathbf{e}_3]_\times & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\text{T}} \\ \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\text{T}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \hdashline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & g[\mathbf{e}_3]_\times \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\text{T}} & \mathbf{0}_{1\times3} \end{array}\right]_{14\times15}.$$

Note $\mathcal{O}_{\text{hover}}$ is merely a special case of $\mathcal{O}_{\text{SSL}}$. By counting the row rank (or column rank), it is straightforward to see

$$\text{rank}(\mathcal{O}_{\text{hover}}) = 11.$$

Hence there are *four* unobservable modes. Clearly the observability of the system degenerates when the UAV speed is zero. By observation we can identify an orthogonal basis of the null space of $\mathcal{O}_{\text{hover}}$ as

$$\text{Null}\left(\mathcal{O}_{\text{hover}}\right) = \text{Range}\left[\begin{array}{cccc} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{e}_1 & \mathbf{e}_2 \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & -g\mathbf{e}_2 & g\mathbf{e}_1 \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{array}\right]_{15\times4},$$

from which the unobservable modes can be determined as

$$\mathbf{x}_{\text{hover}}^{\text{unbos}} = \{p_{\text{n},x},\, p_{\text{n},y},\, \phi - gb_{a,y},\, \theta + gb_{a,x}\}.$$

In the case of hovering, the position ($p_{\text{n},x}$ and $p_{\text{n},y}$) is still unobservable. Moreover, $\phi - gb_{a,y}$ and $\theta + gb_{a,x}$ are also unobservable. In other words, ϕ and θ as well

as $b_{a,x}$ and $b_{a,y}$ become unobservable when the UAV speed is zero. This observation can be intuitively explained by using the homography formula (4.8): the attitude information of the pitch and roll angles originally comes from the normal vector $\mathbf{N}$ in the homography; when the UAV speed is zero, the translational vector $\mathbf{T}$ in (4.8) is zero; consequently the term $\mathbf{T}\mathbf{N}^{\mathrm{T}}$ will vanish and the attitude information carried by $\mathbf{N}$ cannot be retrieved. It can also be seen from (4.27) that the vision measurement actually is equivalent to a velocity measurement when $\kappa = 0$.

4.3.3 Numerical Rank Analysis

In the preceding subsections we have analytically determined the rank of the observability matrix. In fact, we can also numerically compute the singular values and then determine the rank of the observability matrix. The numerical rank analysis can provide us new insights into the observability of the navigation system.

Denote the singular values of $\mathcal{O}_{\mathrm{SSL}}$ as $\sigma_1 \geq \cdots \geq \sigma_{15} \geq 0$. When $\kappa > 0$ we have $\mathrm{rank}(\mathcal{O}_{\mathrm{SSL}}) = 13$ and hence $\sigma_{12} \geq \sigma_{13} > 0 = \sigma_{14} = \sigma_{15}$; when $\kappa = 0$ we have $\mathrm{rank}(\mathcal{O}_{\mathrm{hover}}) = 11$ and hence $\sigma_{12} = \cdots = \sigma_{15} = 0$. Clearly if $\kappa \to 0$ we would have $\sigma_{12} \to 0$ and $\sigma_{13} \to 0$. Thus it is reasonable to choose the ratio σ_1/σ_{13} as an indicator to evaluate the rank of $\mathcal{O}_{\mathrm{SSL}}$: if σ_1/σ_{13} is very large, $\mathcal{O}_{\mathrm{SSL}}$ is on the verge of rank deficiency, and the rank of $\mathcal{O}_{\mathrm{SSL}}$ almost degenerates to that of $\mathcal{O}_{\mathrm{hover}}$. Hence if σ_1/σ_{13} is very large, the observability of θ and ϕ would become *weak* though they are still observable. The term weak as used here intuitively means that the estimation of pitch and roll angles may converge slow.

The UAV speed κ is not the only parameter that affects σ_1/σ_{13}. In addition to κ, $\mathcal{O}_{\mathrm{SSL}}$ is also parameterized by the UAV altitude d.[1] Figure 4.3 shows how κ and d affect σ_1/σ_{13}. It is observed from Figure 4.3 that σ_1/σ_{13} is large when κ is small or d is large. In other words, the observability of θ and ϕ is weak if the UAV speed is slow or the altitude is large. This observation can be explained by the homography formula (4.8): when the speed is slow or the altitude is large,

[1]The altitude d is contained in $\alpha = -d/T_{\mathrm{v}}$. The vision sampling period T_{v} can also affect σ_1/σ_{13}. Here we only consider $T_{\mathrm{v}} = 0.1$ sec.

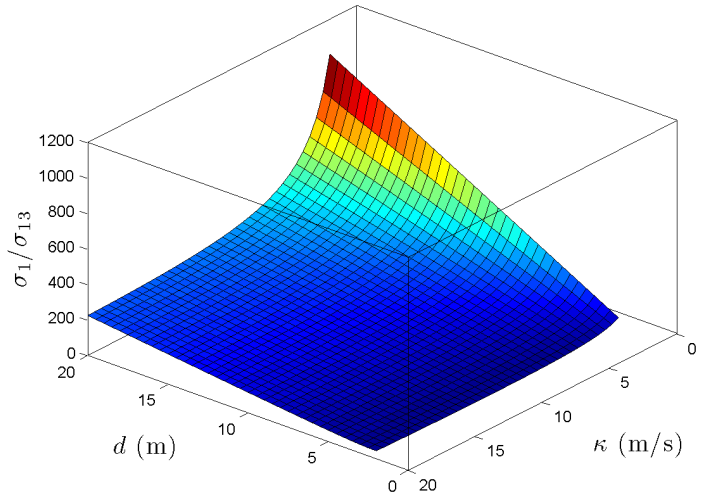


Figure 4.3: The ratio σ_1/σ_{13} is large when κ is small or d is large.

the term $\mathbf{T}\mathbf{N}^{\mathrm{T}}/d$ would almost vanish, which will cause difficulty to recover the attitude information in $\mathbf{N}$. This observation can also be explained from a more intuitive and fundamental point of view, which involves the bearing-only property of vision systems. Note monocular cameras are inherently bearing-only sensors because the depth information of the scene is lost during perspective projection when forming an image. Hence an image inherently only carries the bearings of the ground features. In our vision system, a homography is computed from two consecutive images. If the two images are captured from very different angles relative to the ground features, certain useful information (pitch and roll angles in our case) can be recovered from the bearings carried by the two images. Otherwise, if the UAV speed is slow or the UAV altitude is large, the two consecutive images are almost captured from the same angle relative to the ground features. Then the two images would not provide much new information other than bearings. Hence the pitch and roll angles become difficult to estimate. Interested readers may refer to [122, 91, 92] on the interesting and special properties of bearing-only measurements.

4.4 Comprehensive Simulation Results

In this section, we show comprehensive simulation results to verify the effectiveness of the proposed vision-aided navigation system.

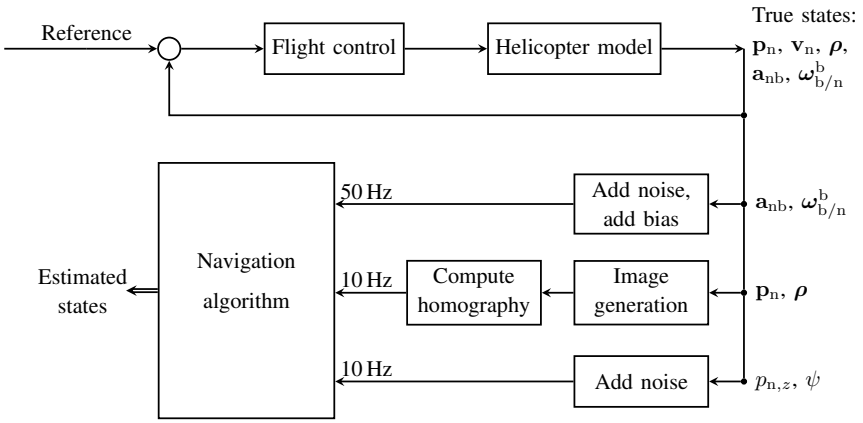


Figure 4.4: Block diagram of the simulation.

Table 4.1: Noise standard deviation and biases in the simulation.

Measurement	Noise standard deviation	Bias
Acceleration (m/s^2)	0.05 (for each entry)	$0.03g$ (for each entry)
Angular rate (rad/s)	0.02 (for each entry)	$\pi/180$ (for each entry)
Heading angle (degree)	1	None
Altitude (m)	2	None

4.4.1 Simulation Settings

The structure of our simulation program is illustrated by the block diagram in Figure 4.4. In the simulation we consider a 6-DOF nonlinear unmanned helicopter model and a flight control law, the details of which can be found in [85, 19]. Given a trajectory reference, true states of the UAV as well as the acceleration and angular rate can be computed. Then we add noises and biases to the true acceleration and angular rate to generate the simulated IMU measurements, and add noises to the true altitude and heading angle to generate the simulated measurements of barometer and compass, respectively. Table 4.1 shows the values of the biases and the standard deviations of the noises.

In our simulation, the homography matrices are computed from real images. We have developed an image generation program using Matlab and C++. A satellite image (3384×2766 pixels) is used to simulate the ground scene. By setting appropriate intrinsic parameters of the downward-looking camera, images (320×240 pixels) captured by the camera can be generated according to the position and attitude of the UAV relative to the satellite image. Samples of the generated images are given in Figure 4.5.

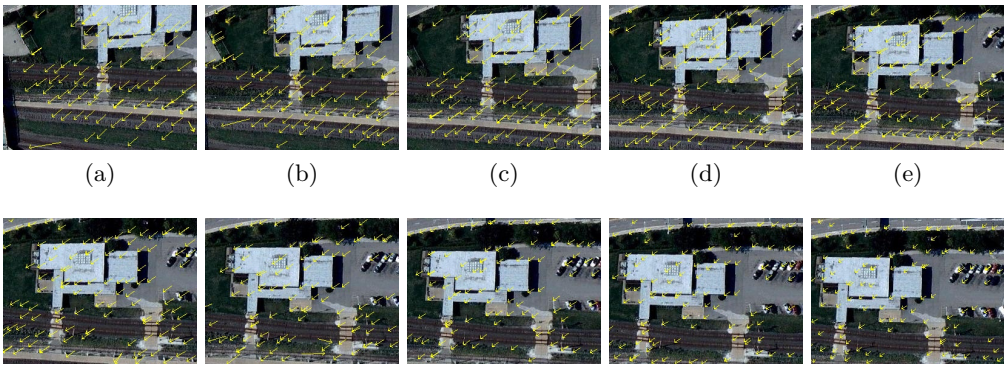


Figure 4.5: Samples of the generated images. The arrows in the images represent the detected optical flow.

In our work we employ OpenCV 2.3.1 to realize real-time vision processing. The following vision processing tasks are required to compute the homography matrix from two consecutive images: (i) extracting feature points in each image (OpenCV function *goodFeaturesToTrack*), (ii) matching feature points of the two images (OpenCV function *calcOpticalFlowPyrLK*) and (iii) computing homography from feature matchings (OpenCV function *findHomography*). Note the second largest singular value of a homography equals one [82, Lemma 5.18, p. 135]. Hence the homography given by function *findHomography* should be normalized before usage. In order to improve the homography estimation accuracy, we may utilize a number of auxiliary functions such as image undistortion (OpenCV function *undistort*), histogram equalization (OpenCV function *equalizeHist*) and sub-pixel feature location refinement (OpenCV function *cornerSubPix*).

4.4.2 Simulation Results

The trajectory reference of the UAV in the simulation is a sine wave with constant heading angle and altitude. This is a typical maneuvering flight motion called *slalom* [19]. Figure 4.5 shows samples of the generated images. The arrows in the images represent the detected optical flow. Homography matrices are computed from each pair of consecutive images. Since the true states of the UAV are known in the simulation, true homography matrices can also be computed. Thus we can obtain the errors of the estimated homography matrices (see Figure 4.6). As can be seen, the error of each entry of the homography is

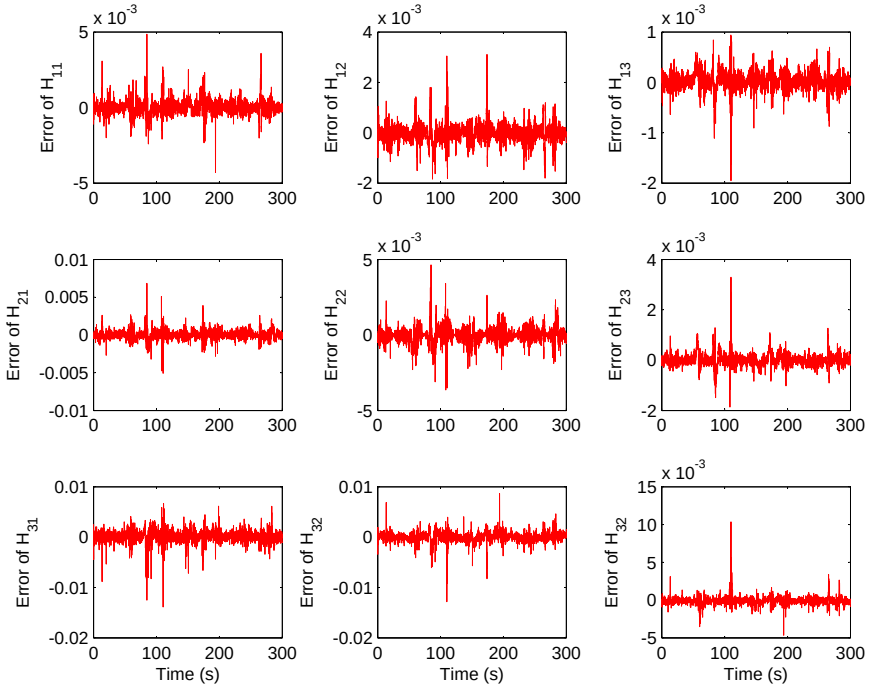


Figure 4.6: The errors of the homography matrices computed from the generated images.

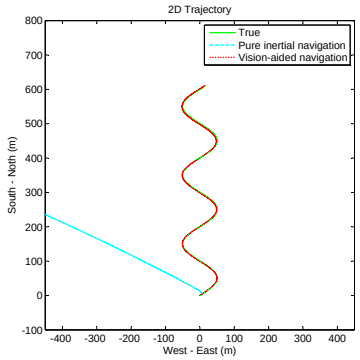
zero-mean and can be reasonably assumed as a zero-mean Gaussian white noise. Most of the homography estimates are accurate though there exist a number of spike measurements whose errors are much larger than the others. In practice, innovation filtering can be used to effectively detect these spikes.

The UAV states are shown in Figure 4.7. The green solid lines represent the true UAV states; the red dotted ones are the UAV states estimated by the vision-aided navigation system; and the cyan dashed ones are the estimates by pure inertial navigation. From Figure 4.7, we have the following important observations.

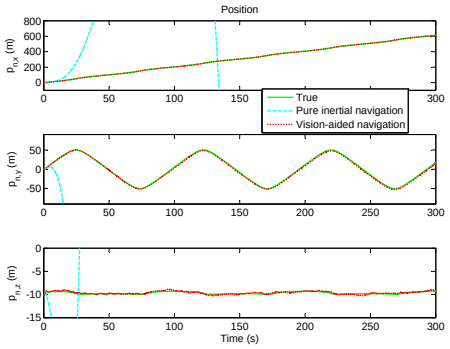
1) Due to the large biases in the acceleration and angular rate measurements, pure inertial navigation drifts very fast. Vision-aided navigation can significantly reduce the position drift as shown in Figure 4.7(a) and (b) though the position ($p_{\mathrm{n},x}$ and $p_{\mathrm{n},y}$) estimate still drifts slowly.

2) Vision-aided navigation provides drift-free estimates of velocity and attitude as shown in Figure 4.7(c) and (d).

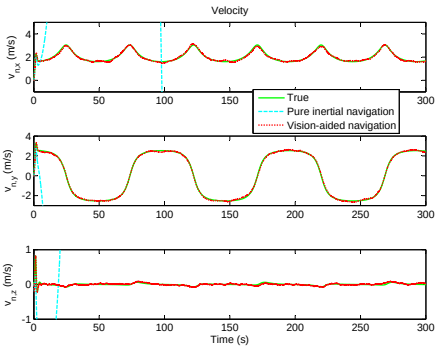
3) As shown in Figure 4.7(e) and (f), the biases of the IMU measurements can



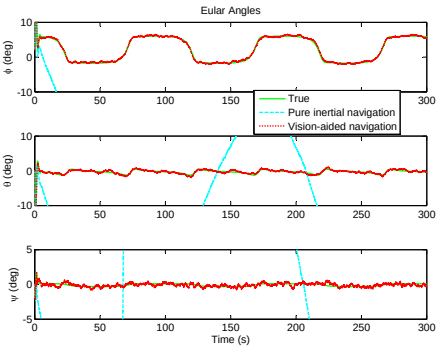
(a) 2D trajectory



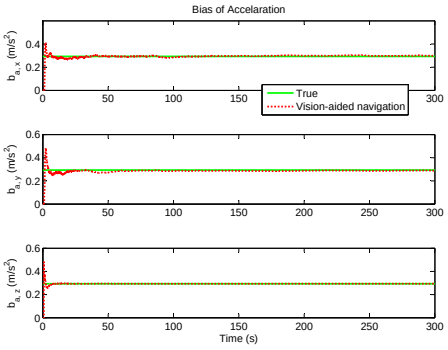
(b) Position



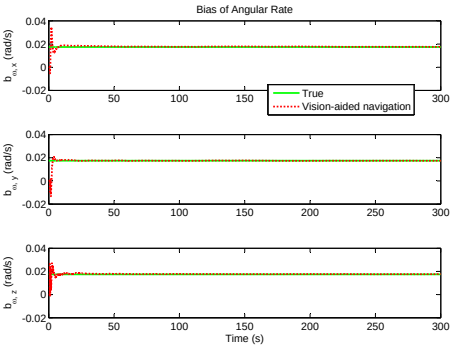
(c) Velocity



(d) Euler angle



(e) Bias of acceleration



(f) Bias of angular rate

Figure 4.7: Simulation results.

Figure 4.8: The quadrotor UAV and the flight test field.

be estimated accurately.

The above three observations are consistent with our observability analysis in Section 4.3. More specifically, observation 1) verifies the observability analysis that the the position ($p_{\mathrm{n},x}$ and $p_{\mathrm{n},y}$) is unobservable; observations 2) and 3) are consistent with the observability analysis that the velocity, attitude and biases are all observable. Due to space limitations, more simulation results are omitted here. From the simulation results under various flight conditions, it is noticed that the convergence of the estimates of θ, ψ, $b_{a,x}$ and $b_{a,y}$ would be slow when the UAV altitude is high or the UAV speed is slow. That is also consistent with the numerical observability analysis in Section 4.3.3.

4.5 Flight Experimental Results

In this section we present flight experimental results to verify the effectiveness and robustness of the proposed navigation system.

4.5.1 Platform and Experimental Settings

As shown in Figure 4.8, the flight experimental platform is an autonomous quadrotor UAV constructed by NUS UAS Team. The key specifications of the quadrotor UAV are listed in Table 4.2. The main onboard systems of this platform are introduced as below.

Table 4.2: Main specifications of the quadrotor UAV.

Specifications	Quadrotor UAV
Dimensions	$61 \times 61 \times 35$ ($l \times w \times h$) cm
No-load weight	1 kg
Maximum takeoff weight	3 kg
Power source	Lithium polymer battery
Flight endurance	15 mins

Navigation Sensor

The quadrotor UAV is equipped with a navigation sensor IG-500N, which contains an IMU, a GPS receiver and a barometer. The navigation sensor can provide a variety of measurements such as the acceleration and angular rate of the UAV. It can also give drift-free measurements of the position, velocity and attitude of the UAV in the presence of GPS. For the purpose of comparison, the UAV states (position, velocity and attitude) measured by the navigation sensor are treated as the *ground truth* in the flight experiments. For our vision-aided navigation system, we only use the measurements of acceleration, angular rate, heading angle and altitude. The update rate of the acceleration and angular rate is 50 Hz, while that of the heading angle and altitude is 10 Hz.

Flight Control System

The primary tasks of the flight control system include collecting measurements from various sensors, executing the proposed navigation algorithm and performing flight control laws. The flight control system also communicates with the ground control station for real-time monitoring and command issuing. The flight control computer is a Gumstix Overo Fire embedded computer with a 720 MHz processor. The navigation and control algorithms run at 50 Hz in the flight control computer. In order to improve the real-time performance, the original Linux operating system in the Gumstix is replaced by a QNX Neutrino real-time operating system. For details of the onboard software system and the ground control station, please refer to [33, 34, 81]. The quadrotor platform is able to perform autonomous hovering and way point following. For details of the modeling and control of the quadrotor UAV, please refer to [109, Sections II and III].

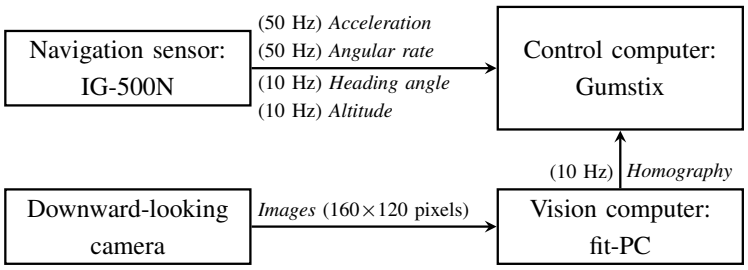


Figure 4.9: The connections between the onboard systems. The 15th-order EKF is executed in real-time in the control computer.

Vision System

The onboard vision system consists of a monocular downward-looking camera (Pointgrey Firefly) and an embedded vision computer (fit-PC2i). The weight of the entire vision system is about 350 g. The camera is directed downwards in order to capture images of the ground scene during flight. It captures images of 160×120 pixels at 10 FPS (frame per second). The parameters of the camera such as exposure and shutter speed can be customized for outdoor applications. A wide-angle lens is connected to the camera to enhance the image quality. The intrinsic parameters of the camera have been calibrated before flight tests. To process the images captured by the camera, a light-weight embedded computer, fit-PC2i, is adopted as the vision computer. This vision computer contains an Intel Atom Z530 1.6 GHz CPU, a solid-state drive (SSD), 1 GB memory and four USB 2.0 high-speed ports. Compared to hard disk drives, the SSD is less susceptible to physical vibration during flight. The image and vision processing algorithms are implemented on a Linux operating system installed in the vision computer. The camera and the vision computer are connected through a USB 2.0 port. Homography matrices are computed in the vision computer at 10 Hz and sent to the flight control computer through a RS232 full UART.

The vision algorithm for computing homography matrices has already been discussed in the last paragraph of Section 4.4.1. But since the onboard computational resource is very limited, we need to adjust the vision algorithm in order to realize real-time onboard vision processing. For example, the image size in simulations is 320×240 pixels, but we need to reduce the image size to 160×120 pixels in flight experiments to reduce computational burden. In addition, some

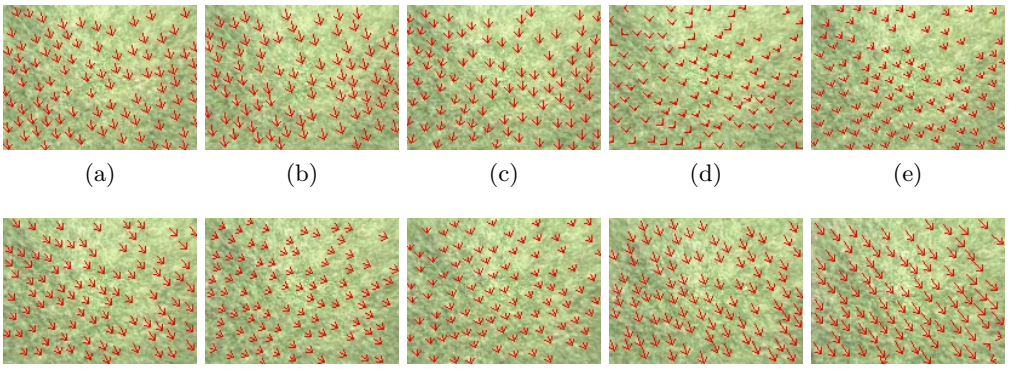


Figure 4.10: Samples of the consecutive images captured by the onboard camera. The arrows in the images represent the detected optical flow.

time-consuming functions such as sub-pixel feature location refinement have also been removed. In our flight experiments, we notice that the dynamically changing light condition in outdoor environments can easily cause large errors in homography estimates. It is experimentally proved that the histogram equalization function can effectively eliminate the affect of light condition changing.

The connections between the above onboard systems are illustrated in Figure 4.9.

4.5.2 Experimental Results

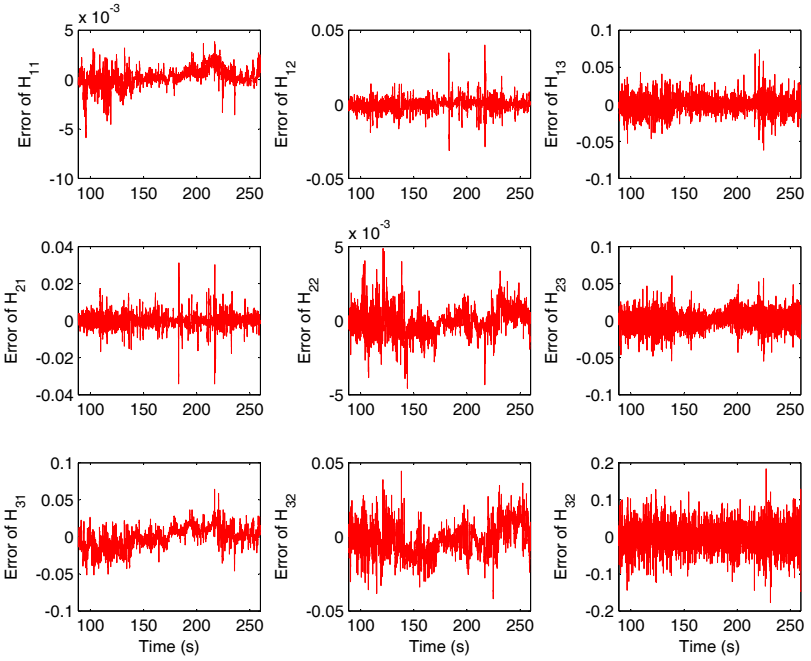
The flight experiments are conducted in a flat grass field (see Figure 4.8). Figure 4.10 shows samples of the images of the grass field captured by the onboard camera. As can be seen, features and optical flow can be smoothly detected in the images.

Next we show two sets of flight experimental results: open-loop and closed-loop. (i) In the open-loop flight experiment, the UAV is controlled manually. The flight data and the images captured by the onboard camera are used to test the vision-aided navigation algorithm. The term open-loop simply means the navigation results are not fed back for flight control. (ii) In the closed-loop flight experiment, the UAV flies autonomously to follow the pre-specified way points. The navigation results are fed back for autonomous flight control. The vision-aided navigation algorithm is executed in the onboard computer in real-time.

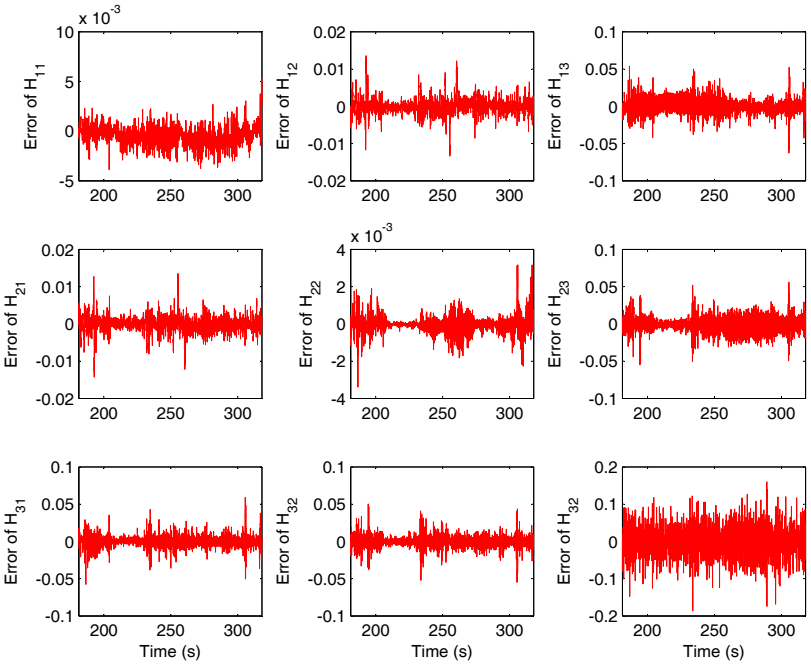
According to the position and attitude provided by GPS/IMU, we can com-

pute the true homography matrices, based on which the errors of the onboard estimated homography matrices can be obtained. As shown in Figure 4.11, the error of each entry of the homography can be reasonably assumed as a zero-mean Gaussian white noise. The homography errors in the flight experiments are much larger than those in the simulations. Nevertheless, the vision-aided navigation can still perform well.

Figures 4.12 and 4.13 show the open-loop and closed-loop flight experimental results, respectively. As can be seen, the vision-aided navigation can provide accurate and drift-free estimates of the velocity and attitude. The UAV can be successfully stabilized based on the vision-aided navigation. In Figure 4.12(a) or Figure 4.13(a), the 2D trajectory is plotted against the satellite image of the flight test field. Note the satellite image is only for illustration purposes and it is *not* used for navigation. In Figure 4.12(c) or Figure 4.13(c), the z-axis velocity $v_{\mathrm{n},z}$ given by the vision-aided navigation is inconsistent with the GPS data. After conducting a series of experiments, we have verified that the $v_{\mathrm{n},z}$ estimated by the vision-aided navigation is correct, while the $v_{\mathrm{n},z}$ given by GPS is inaccurate due to certain technical problems which will be solved in the future. The reason the vision-aided navigation can estimate $v_{\mathrm{n},z}$ accurately is because $p_{\mathrm{n},z}$ can be directly measured by the barometer.

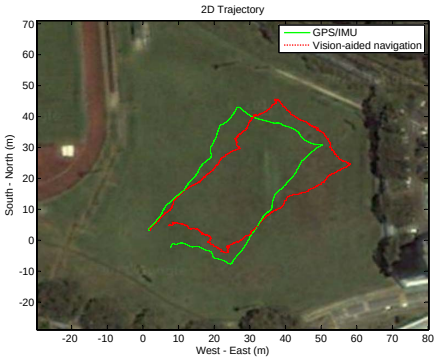


(a) Open-loop experiment

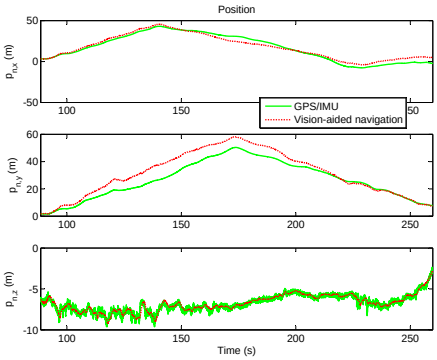


(b) Closed-loop experiment

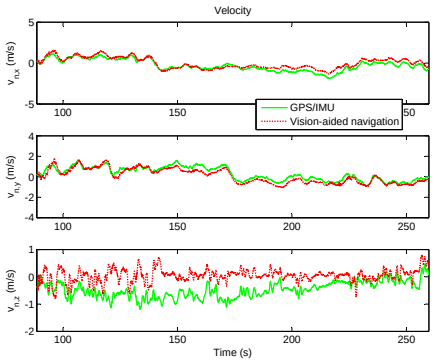
Figure 4.11: The errors of the homography estimates.



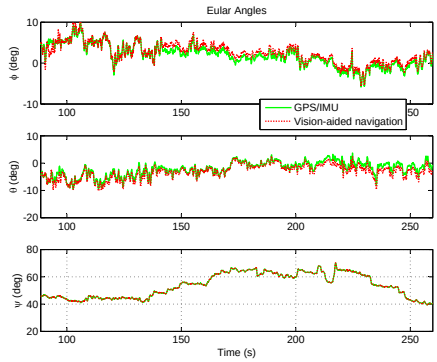
(a) 2D trajectory plotted against satellite image



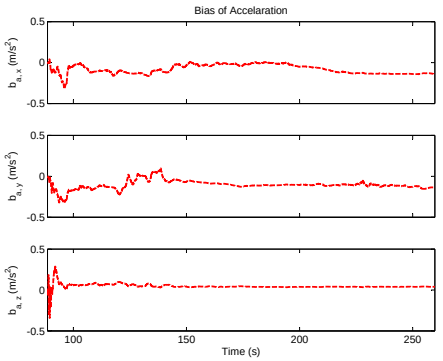
(b) Position



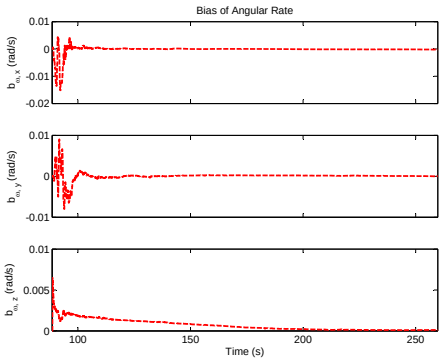
(c) Velocity



(d) Euler angle

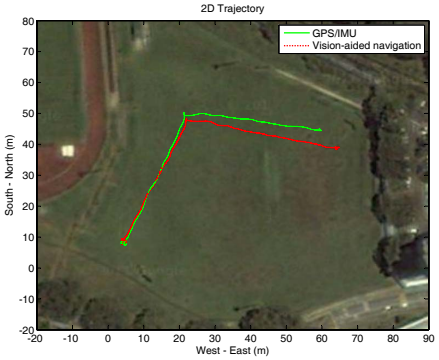


(e) Bias of acceleration

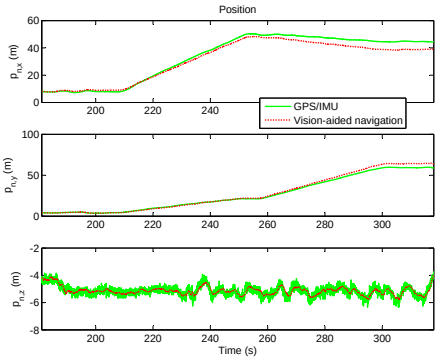


(f) Bias of angular rate

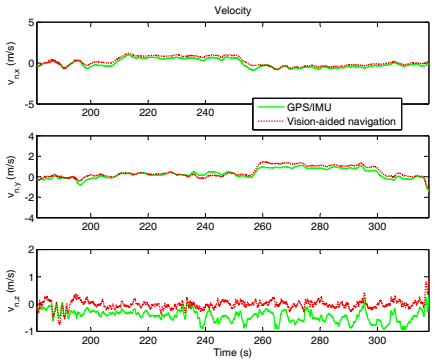
Figure 4.12: Open-loop flight experimental results.



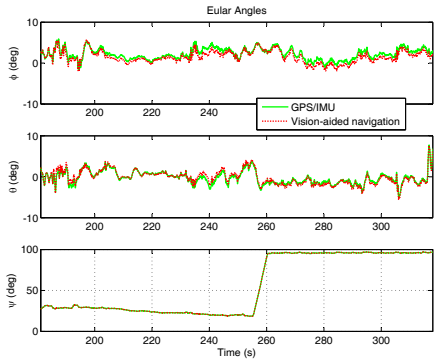
(a) 2D trajectory plotted against satellite image



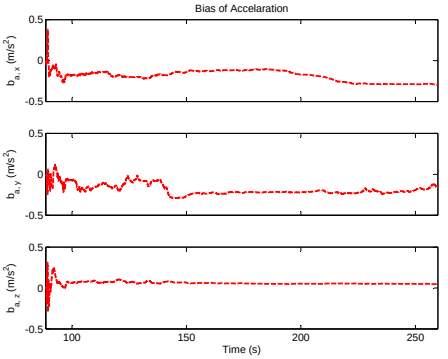
(b) Position



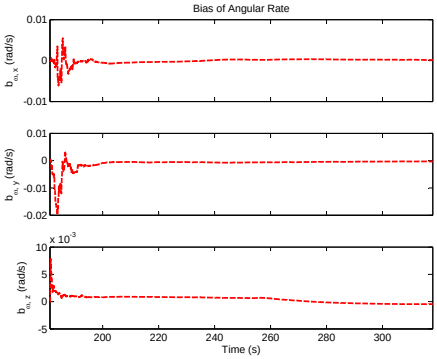
(c) Velocity



(d) Euler angle



(e) Bias of acceleration



(f) Bias of angular rate

Figure 4.13: Closed-loop autonomous flight experimental results.

Chapter 5

Vision-based Navigation using Artificial Landmarks

5.1 Introduction

The previous chapter considered vision-based navigation of UAVs using natural landmarks. In this chapter, we address vision-based navigation using artificial landmarks such as circles with known diameters. The navigation system presented in this chapter was motivated by the 2013 UAVGP competition.

In order to accomplish the competition tasks proposed by the 2013 UAVGP, our NUS UAS team has developed a fully autonomous unmanned helicopter system. The whole system consists of two main sub-systems: automatic control and vision processing. This chapter will only focus on the vision system; the control system will be addressed in detail elsewhere. The developed vision system is able to complete the following tasks. Firstly, detect all ellipses in the image, determine whether there is a bucket in the corresponding circle, and then select a target ellipse according to the current aim of the UAV; secondly, track the target ellipse over the image sequence consistently once it has been selected by certain initialization procedure; thirdly, estimate the position of the target circle relative to the UAV such that the UAV can be guided to the target circle to grab or drop a bucket.

We have developed a variety of algorithms for the vision system to complete the competition tasks. But this chapter will mainly focus on the core algorithms

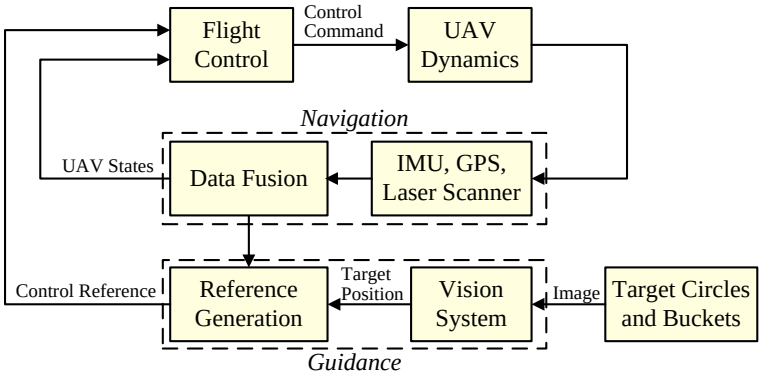
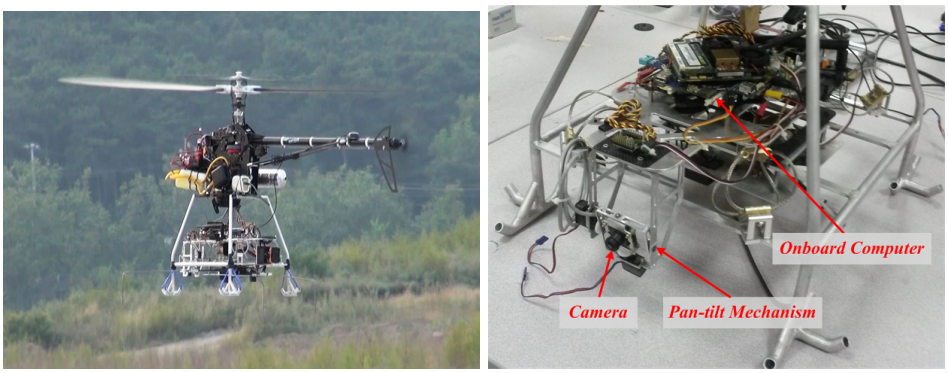


Figure 5.1: Guidance, navigation and control structure of the unmanned helicopter system.

on ellipse detection, ellipse tracking and circle-based pose estimation. These algorithms are not restricted to the specific competition tasks. They can also be applied to a wide range of vision-based guidance and navigation tasks such as autonomous takeoff and landing [76, 40, 116], target tracking and following [79, 62], visual servoing [111, 43, 99, 75], and vision-based formation control [31].

Although computer vision provides a promising approach to many UAV tasks [78, 93], too complicated and time-consuming vision algorithms are hardly applicable to practical real-time applications due to the very limited onboard computational resources. In our work, we design and implement our algorithms mainly from the practical point of view. The proposed algorithms can be conveniently implemented by the popular library OpenCV (the version we used is 2.3.1). No special functions of advanced vision processing or mathematical computation are required. The implementation details of the algorithms are presented in the hope that one can implement the algorithms for their own work by following the chapter.

The chapter is organized as follows. We first introduce the unmanned helicopter systems and the vision system in Section 5.2. The algorithms for ellipse detection, ellipse tracking and circle-based pose estimation are presented in Sections 5.3, 5.4 and 5.5, respectively. Experimental results are given in Section 5.6 to verify the proposed vision algorithms.



(a) Unmanned helicopter (b) Onboard vision system

Figure 5.2: The unmanned helicopter and the onboard vision system.

5.2 System Overview

The guidance, navigation and control structure of the unmanned helicopter system is given in Figure 5.1. It is worth noting that the navigation sensors on the UAV include an IMU, a GPS receiver and a laser scanner. The laser scanner is used to obtain the accurate altitude of the UAV. As shown in Figure 5.1, the role of the vision system is to provide guidance for the UAV. The input of the vision system is an image of the target circles and buckets; the output is the position of the target circle. The vision estimation results will be consequently used to guide the UAV to the target circle to grab or drop a bucket.

The unmanned helicopter developed by the our UAS Group at NUS is shown in Figure 5.2(a). As shown in Figure 5.2(b), the onboard vision system is composed of a camera, a pan-tilt mechanism and an onboard computer. We selected a wide-angle camera in order to cover a large area. After image undistortion, the horizontal angle of view is 87°, and the vertical one is 71°. The camera is mounted on a pan-tilt mechanism, which can be controlled in 50 Hz to ensure that the image plane is perfect horizontal. The pan-tilt mechanism can compensate the fast dynamical rotation of the UAV. As a result, the location of the target in the image changes smoothly which is essential for image tracking. The onboard vision computer is AscTec Mastermind with the CPU as Intel Core2Duo SL9400 (2x1.86 GHz). The whole vision system including various image processing and vision algorithms is run at 5 Hz in the onboard computer. The updating rate of the vision system can also be chosen to 10 Hz, but 5 Hz is already sufficient for

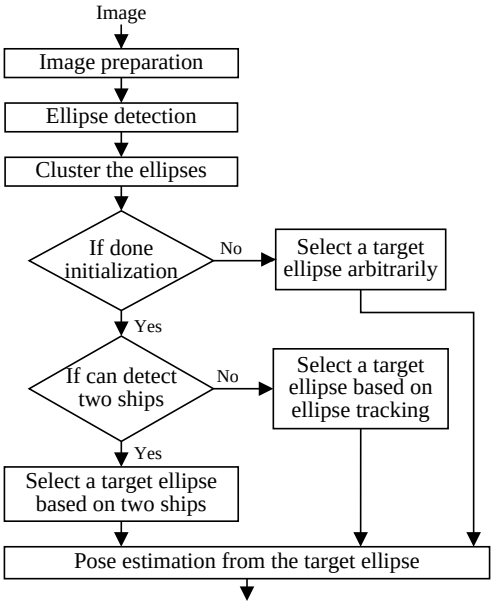


Figure 5.3: Flow chart of the vision system.

our applications.

The flow chart of the vision system is given in Figure 5.3. We next briefly explain each procedure in the flow chart.

1) *Image* The image size is 640x480 pixels.

2) *Image preparation* The purpose of the image pre-processing is to detect the contours corresponding to the circles. These detected contours will be consequently used for ellipse detection.

3) *Ellipse detection* Once the contours has been obtained by the previous step, the ones that correspond to ellipses will be detected. Ellipse detection is a key procedure and will be addressed in detail later.

4) *Ellipse clustering* The main aim for ellipse clustering is to decide whether the two ships are in the field of view. Since the four ellipses on each ship are of the same size and they are distributed evenly on the ship, their size and position information can be used for ellipse clustering. If two clusters of ellipses can be obtained, it can be concluded that the two ships are in the field of view.

5) *Initialization* The UAV takes off from a location far from the ships and is guided to the ships based on GPS. Once the vision system has detected

two ships, an initialization procedure will be triggered. The purpose of the initialization procedure is to select a proper target circle according to the UAVs current aim. The image tracking is also initialized in this procedure.

6) *Select a target ellipse arbitrarily* After the UAV takes off, it will be guided to the ship area by GPS. However, the GPS accuracy is not perfect and hence the UAV may not be guided to the exact desired area. Motivated by that, the vision system will provide the position of any detected circle before the initialization procedure such that the UAV can be guided to the exact ship area.

7) *Select a target ellipse based on two ships* If two ships are in the field of view, the vision system will select a target ellipse according to the current aim of the UAV. For example, suppose the buckets are initially placed on the right ship and they are required to be taken to the left one. If the current aim for the UAV is to grab a bucket, the vision system will automatically select one ellipse that corresponds to a circle on the right ship with bucket in it.

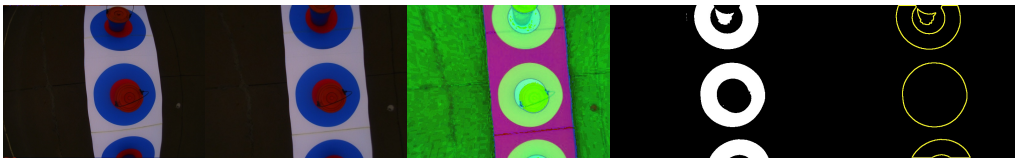
8) *Select a target ellipse based on ellipse tracking* In most of the cases, the two ships cannot be the field of view simultaneously. Then it is essential to track a target ellipse over the image sequence.

9) *Pose estimation from the target ellipse* Once the target ellipse has been selected in any of the above ways, the ellipse will be used to estimate the position of the circle center relative to the camera.

Ellipse detection, ellipse tracking and single-circle-based pose estimation are three key problems in the vision system. In the sequel of the chapter, we will present the details of our solutions to the three problems.

5.3 Ellipse Detection

In this section, we present the real-time and robust ellipse detection algorithm developed for the vision system.



(a) (b) (c) (d) (e)

Figure 5.4: An illustration of the preparation steps. (a) Original image; (b) Undistorted image; (c) Converting the image from RGB to HSV; (d) Color thresholding; (e) Detect contours.

5.3.1 Preparation

Before presenting the ellipse detection algorithm, we first introduce several preparation steps. The purpose of the preparation steps is to obtain the contours that correspond to the circles. These contours will be subsequently processed by the ellipse detection algorithm. Though important, the preparation steps are not the main focus of this chapter. One may choose different methods to detect contours according to their own specific tasks. Edge detection and color thresholding are two common methods. In our work, we detect contours by color thresholding because the color information of the circles is already given in the competition task.

The preparation steps mainly include image undistortion, converting RGB to HSV, color thresholding and contour detection. These steps are illustrated by Figure 5.4. Details of these steps are omitted here as they can be easily implemented by using OpenCV. In our work, we are only interested in the extreme outer contours and the inner ones are not detected (see, for example, the middle contour in Figure 5.4(e)).

5.3.2 A Three-step Ellipse Detection Procedure

Once the contours have been detected, each of them will be processed by the ellipse detection algorithm to see if it corresponds to an ellipse. The proposed ellipse detection algorithm contains three steps: 1) pre-processing based on AMIs, 2) ellipse fitting and 3) post-processing based on algebraic error. Ellipse fitting is the core of our algorithm. But ellipse fitting itself is not able to determine whether a contour corresponds to an ellipse or not. The purpose of the pre- and post-processing steps is to robustly eliminate non-elliptical contours. We

describe the three steps one by one in detail as follows.

Pre-processing based on AMIs

Moment invariants are very useful tools for pattern recognition. The most popular moment invariants were proposed by Hu [61]. Hu's moment invariants are invariant only under translation, rotation and scaling of the object. The work in [48] proposed AMIs that are invariant under general affine transformations. Since any ellipse can be obtained by applying an affine transformation to a circle, all ellipses and all circles have exactly the same AMIs. As a result, AMIs are powerful tools for ellipse detection [103, 87, 113].

Four AMIs were proposed in [48, Section 2.1]. In our work, we only use the first three as the fourth one is much more complicated and less reliable than the first three. The first three AMIs are expressed as

$$\begin{aligned} I_1 &= (\mu_{20}\mu_{02} - \mu_{11}^2)/\mu_{00}^4, \\ I_2 &= (\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^3 + 4\mu_{21}^3\mu_{03} - 3\mu_{21}^2\mu_{12}^2)/\mu_{00}^{10}, \\ I_3 &= \left(\mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2)\right)/\mu_{00}^7, \end{aligned} \tag{5.1}$$

where the central moment μ_{ij} is given by

$$\mu_{ij} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}(x-\bar{x})^i(y-\bar{y})^j\rho(x,y)\mathrm{d}x\mathrm{d}y. \tag{5.2}$$

In the above equation, $(\bar{x}, \bar{y})$ is the coordinate of the centroid and $\rho(x, y)$ is the density distribution function.

We now calculate the AMIs of ellipses and circles. Since all ellipses and circles have the same AMIs, for the sake of simplicity, we can calculate the AMIs of a special one: a circle centered at the origin with the radius as one unit. Suppose $\rho(x, y) = 1$ when (x, y) is on the circle, and $\rho(x, y) = 0$ otherwise. Then (5.2) can be rewritten as

$$\mu_{ij} = \int_0^{2\pi}\cos^i\theta\sin^j\theta\mathrm{d}\theta. \tag{5.3}$$

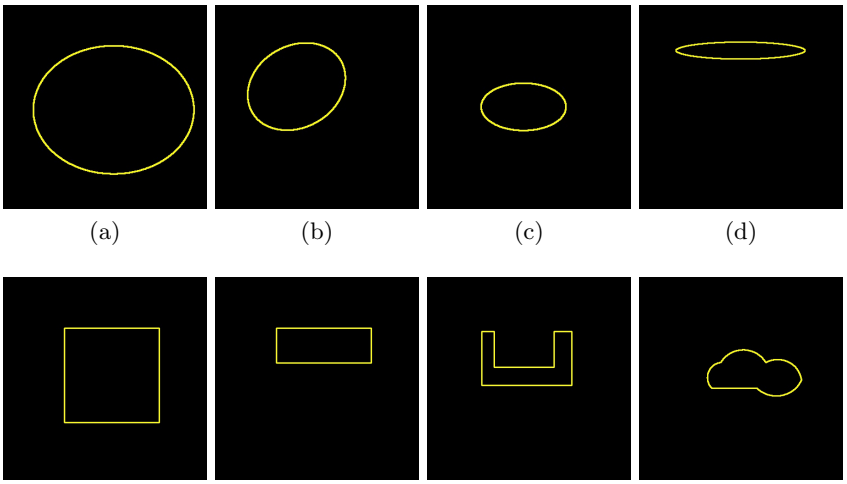


Figure 5.5: Examples to verify the AMIs given in (5.4).

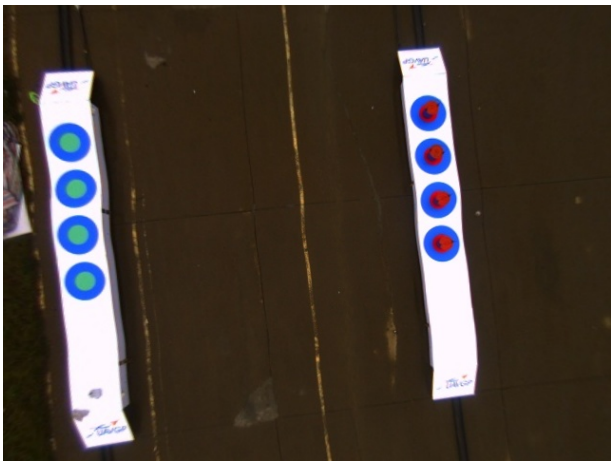
It can be calculated from (5.3) that $\mu_{00} = 2\pi$, $\mu_{11} = 0$, $\mu_{02} = \mu_{20} = \pi$ and $\mu_{12} = \mu_{21} = \mu_{03} = \mu_{30} = 0$. Substituting these values into (5.1) yields

$$\begin{aligned} I_1 &= \frac{1}{16\pi^2} \approx 0.006332, \\ I_2 &= I_3 = 0. \end{aligned} \tag{5.4}$$

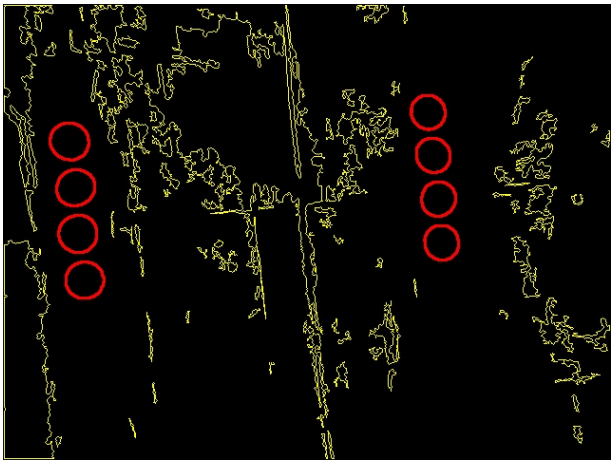
Thus an arbitrary circle or ellipse has its I_1, I_2 and I_3 as given in (5.4).

A number of artificial images are shown in Figure 5.5 to verify the theoretical values in (5.4). The contours in Figure 5.5(a)-(d) are elliptical contours with different shape, scale, translation and rotation. As can be seen from Table 5.1, their AMIs are very close to the theoretical values. As a comparison, the AMIs of the non-elliptical contours in Figure 5.5(e)-(h) are significantly different from the theoretical values. In addition, since an elliptical contour in an image always contains a finite number of points, the AMIs computed from the contour usually have errors. Generally speaking, the fewer the points contained in the contour, the larger the errors will be. It is observed that the elliptical contour in Figure 5.5(d) contains fewer points than the others. As a result, the error of its I_1 is relatively larger.

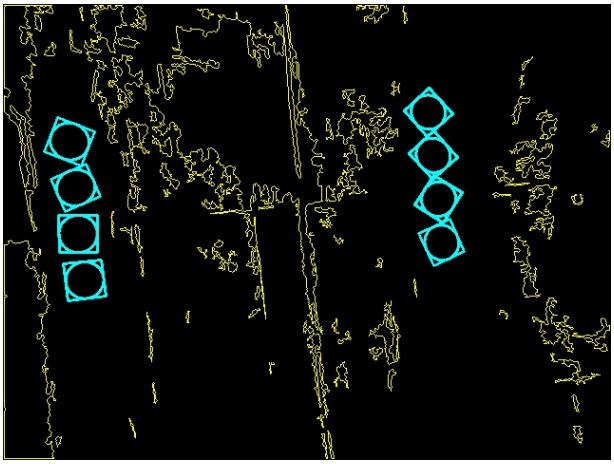
To sum up, the pre-processing for ellipse detection is to compute the AMIs of each contour and compare them with the theoretical values in (5.4). If the AMIs of a contour is sufficiently close to the theoretical values, then the contour



(a)



(b)



(c)

Figure 5.6: An example to illustrate the pre-processing and ellipse fitting. As can be seen, the AMIs can be used to robustly detect the elliptical contours in the presence of a large number of non-elliptical ones. (a) Color image; (b) Elliptical contours detected based on AMIs; (c) Fitted ellipses with rotated bounding boxes.

Table 5.1: The AMIs of the contours in Figure 5.5.

Image	(a)	(b)	(c)	(d)
I_1	0.006333	0.006333	0.006334	0.006338
I_2	0.000000	0.000000	0.000000	0.000000
I_3	0.000000	0.000000	0.000000	0.000000
Image	(e)	(f)	(g)	(h)
I_1	0.006944	0.006944	0.030369	0.007094
I_2	0.000000	0.000000	0.000003	0.000000
I_3	0.000000	0.000000	0.000114	-0.000003

can be classified as an ellipse. According to our experience, we recommend the thresholds for I_1, I_2 and I_3 to be ± 0.0003, ± 0.0000001 and ± 0.000001, respectively. At last, the OpenCV function *moments* can be used to compute all the central moments that are required to calculate AMIs. An example is shown in Figure 5.6 to verify the effectiveness and robustness of the pre-processing. As can be seen, the contours that correspond to ellipses are all successfully detected in the presence of a large number of non-elliptical contours.

Ellipse Fitting

There are a variety of ellipse fitting algorithms in the literature (see, for example, [47, 1]). In our work, we choose the ellipse fitting function, *fitEllipse*, implemented in OpenCV. Experiments show that this function is efficient and accurate enough for our work. Figure 5.6 gives an example to illustrate the ellipse fitting. As shown in Figure 5.6(c), all ellipses in the image are successfully obtained based on the elliptical contours given by the pre-processing.

Post-processing based on Algebraic Error

In the pre-processing, we usually set the thresholds to be a little loose such that all elliptical contours can be successfully detected. But as a result of the loose thresholds, non-elliptical contours may be falsely detected by the pre-processing. In order to improve the robustness of the algorithm, we adopt a post-processing procedure to remove these false detections. The post-processing calculates the algebraic error between a contour and its fitted ellipse. If the algebraic error is larger than a threshold, the fitted ellipse will be classified as a false detection.

The algebraic error between a contour and the fitted ellipse is defined as

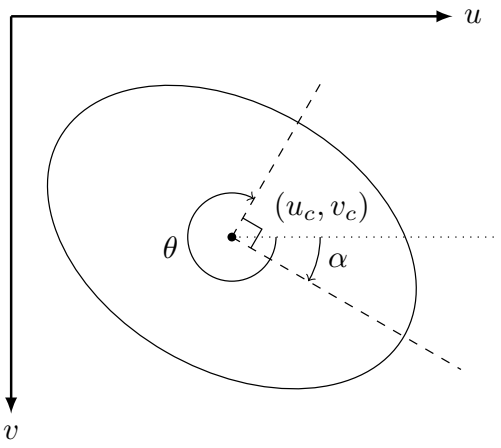


Figure 5.7: An illustration of the ellipse parameters and the angle returned by RotatedRect in OpenCV.

follows. Let (u, v) be the coordinate of a pixel point in the image coordinate frame. As shown in Figure 5.7, the u- and v-axis of the image frame defined in OpenCV are pointing right and downward, respectively. The origin of the image frame is located at the upper left corner of the image. Let (u_c, v_c) be the coordinate of the ellipse center, and a and b be the lengths of the semi-major and semi-minor axes, respectively. Denote α as the angle between the major axis of the ellipse and the u-axis of the image frame (see Figure 5.7). The angle α is positive clockwise based on the right-hand rule. Then the equation of the ellipse in the image frame is expressed as

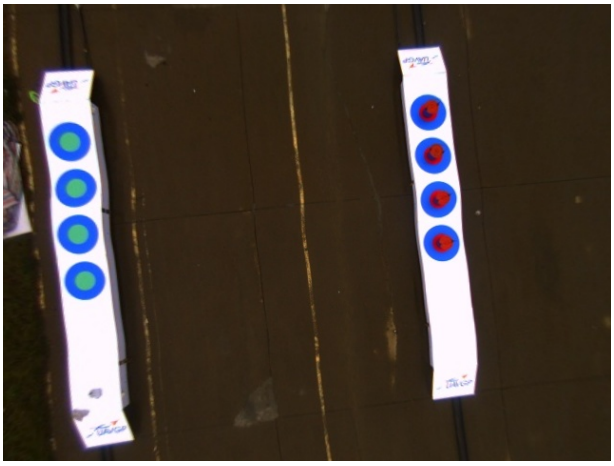
$$\frac{[(u-u_c)\cos\alpha + (v-v_c)\sin\alpha]^2}{a^2} + \frac{[(u-u_c)\sin\alpha - (v-v_c)\cos\alpha]^2}{b^2} = 1. \quad (5.5)$$

Based on (5.5), the algebraic error between a contour and the fitted ellipse is defined as

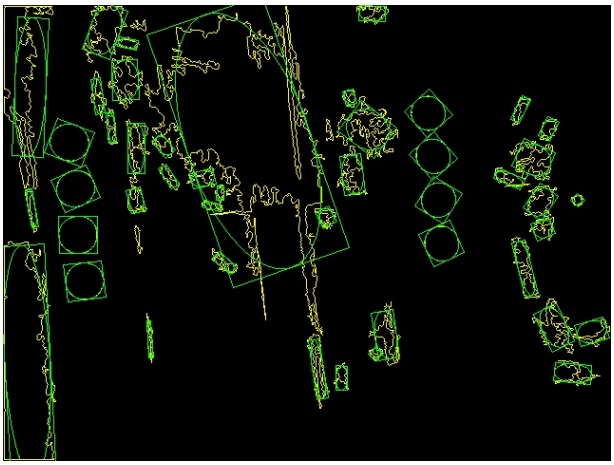
$$e_{\text{alg}} \triangleq \frac{1}{n}\sum_{i=1}^{n}\left|\frac{[(u_i-u_c)\cos\alpha + (v_i-v_c)\sin\alpha]^2}{a^2} + \frac{[(u_i-u_c)\sin\alpha - (v_i-v_c)\cos\alpha]^2}{b^2} - 1\right|, \quad (5.6)$$

where (u_i, v_i) with $i = 1, \ldots, n$ are the coordinates of the points on the contour.

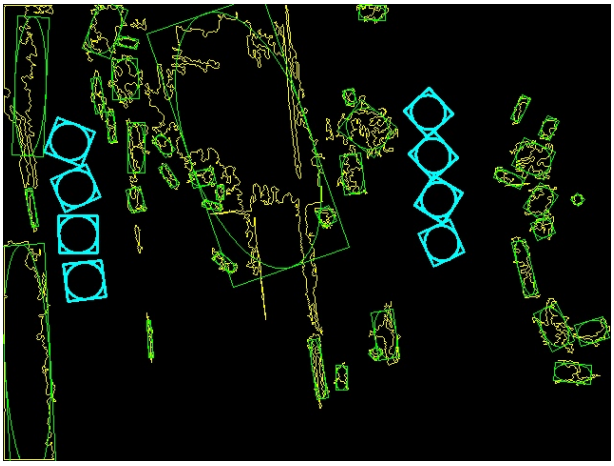
The post-processing will eliminate a fitted ellipse if its algebraic error is larger than a threshold. According to a variety of experiments, we recommend the threshold for the algebraic error to be set as 0.1. An example is given in



(a)



(b)



(c)

Figure 5.8: An example to illustrate the post-processing. (a) Color image; (b) Fitted ellipses for all contours (contours with too few points are excluded); (c) Good ellipses detected based on the algebraic error.

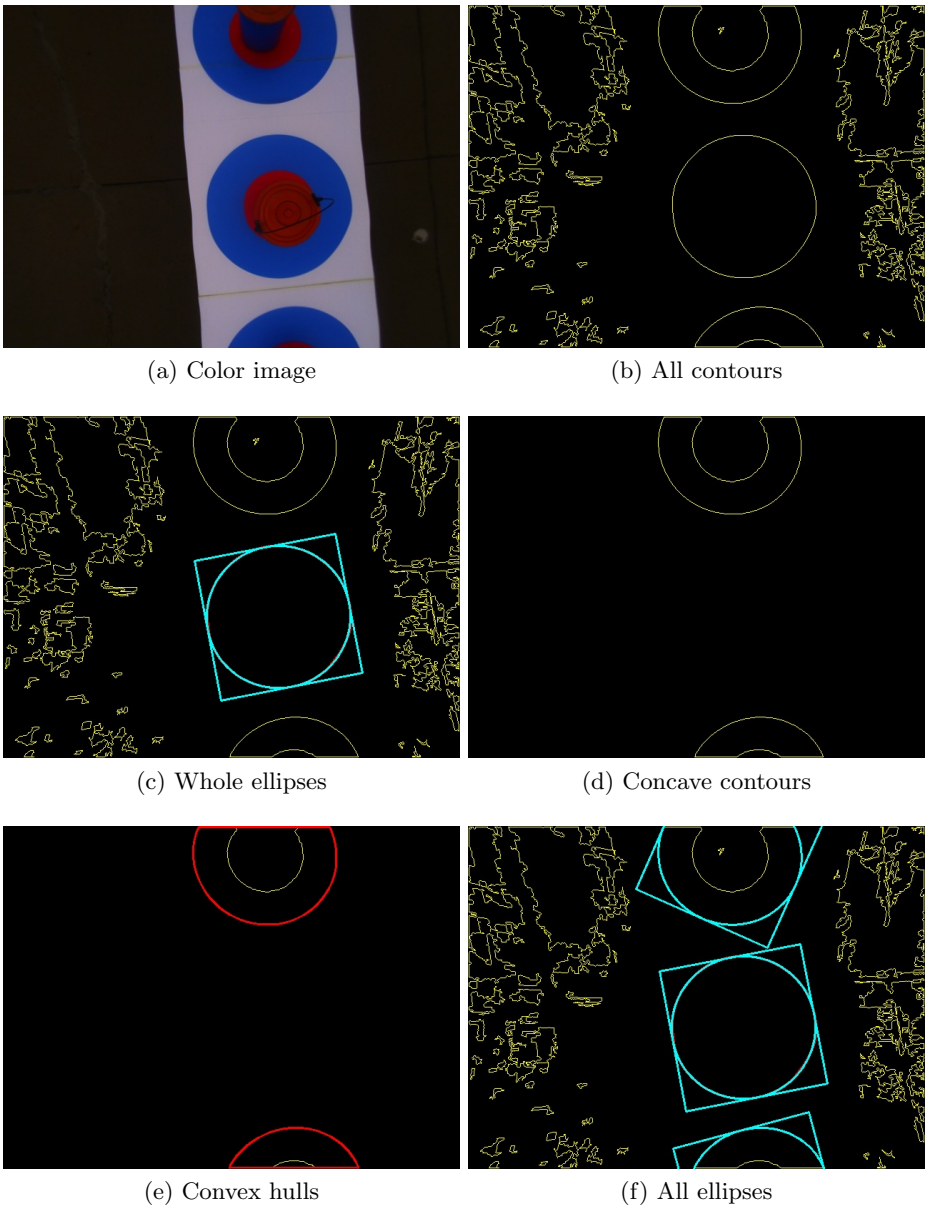
Figure 5.8 to illustrate the effectiveness and robustness of the post-processing. Note the ellipse detection in Figure 5.8 is based merely on ellipse fitting and post-processing, while pre-processing is not used. As shown in Figure 5.8(b), ellipses are fitted for all of the contours. As can be seen from Figure 5.8(c), all falsely fitted ellipses can be eliminated by the post-processing.

In practical implementation, an ellipse is described by a structure named *RotatedRect* in OpenCV. We can easily (but may not directly) obtain all the parameters required to compute the algebraic error from *RotatedRect*. Here we would like to highlight how to obtain the parameter α from *RotatedRect* since it is a little tricky. The angle returned by *RotatedRect* is not α; instead, it is θ as shown in Figure 5.7. According to a series of experiments we have conducted, the angle θ is the angle between the minor-axis of the ellipse and the u-axis of the image frame. Moreover, the angle θ is positive clockwise, and it is always in the interval $[135°, 315°)$. Therefore, the angle α can be obtained as $\alpha = \pi/2 + \theta$. It is worth noting that adding $k\pi$ to α with k as an integer does not affect the algebraic error in (5.6).

Discussion

It is not necessarily required that the pre- and post-processing procedures must be both adopted in the ellipse detection algorithm. As illustrated in Figure 5.6 and Figure 5.8, either pre- or post-processing is already sufficient to detect all the ellipses. But in order to improve the robustness of the algorithm, it is recommended to use both of them unless the computational resource is extremely limited.

More importantly, the pre-processing based on the AMIs can only detect whole ellipses but not partial ones. As a comparison, the post-processing based on the algebraic error can handle both whole and partial ellipses, but it is not able to judge whether an ellipse is whole or partial. Therefore, the pre- and post-processing procedures have their own advantages, which should be used flexibly in practice. We will show later how the algebraic error can be used to detect partially occluded ellipses.



(a) Color image (b) All contours

(c) Whole ellipses (d) Concave contours

(e) Convex hulls (f) All ellipses

Figure 5.9: An example to illustrate the detection of partially occluded ellipses.

5.3.3 Special Cases

We next solve two important special cases.

Partially Occluded Ellipse

In practice, it is common that only part of a circle can be seen by the camera due to occlusion or limited field of view. Then the contour corresponding to that circle will not be elliptical (see, for example, the top and the bottom contours in Figure 5.9(b)). We next show how to fit ellipses for these contours. The procedure is as the following. Firstly, run the pre- and post-processing to detect

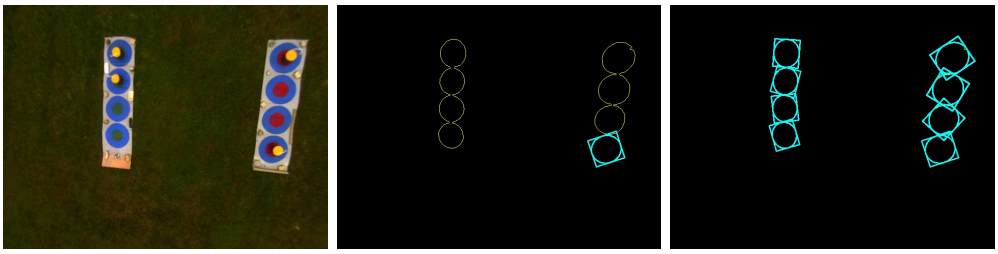
whole ellipses. Secondly, compute the convex hull of each of the contours that are not detected as whole ellipses. The OpenCV function *convexHull* can be conveniently used to do this. Thirdly, fit ellipse for the convex hull, and then compute the algebraic error between the fitted ellipse and the convex hull (not the original contour). If the algebraic error is too large, the fitted ellipse will be eliminated, otherwise it can be classified as an occluded ellipse.

Figure 5.9 shows an example to illustrate the above procedure. As can be seen from Figure 5.9(a), there exists three circles in the image. The middle circle can be entirely seen, but the top and the bottom ones are only partially observable. As a result, the image of the middle circle is a whole ellipse, while the images of the other two are concave contours (see Figure 5.9(b)). The whole ellipse can be successfully detected by the pre- and post-processing procedures while the concave ones cannot (see Figure 5.9(c)). But by employing the convex hull based method introduced above, the top and the bottom ellipses can also be successfully detected as shown in Figure 5.9(f).

Overlapped Ellipses

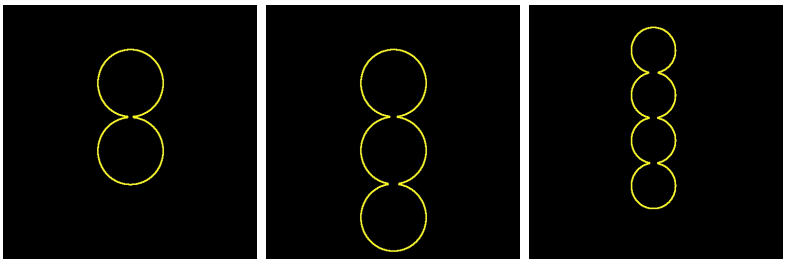
When solving the competition task, we encountered a very special problem: although the circles are not overlapped in the real world, their images may be slightly overlapped due to image blur or inappropriate color thresholding. Figure 5.10 shows a typical example where there are actually eight circles on the ground, but only three contours can be detected in the image. As shown in Figure 5.10(b), the left four circles are detected as one single contour, and so are the right top three. The pre- and post-processing can only detect the isolated ellipse (the right bottom one). We next propose an algorithm to tackle this special case.

The proposed algorithm consists of two steps. The first step is to detect the overlapped contours based on AMIs. As shown in Figure 5.11, the AMIs of the special contours are $I_1 \approx 0.008$ and $I_2 = I_3 = 0$. Hence by setting appropriate thresholds (we recommend ± 0.01 for I_1, ± 0.0000001 for I_2 and ± 0.000001 for I_3), these overlapped contours can be detected. In the second step, we need to determine how many ellipses exist in an overlapped contour. The basic idea is trial



(a) Color image (b) Whole ellipses (c) All ellipses

Figure 5.10: An example to illustrate the case of overlapped ellipses.



(a) (b) (c)

Figure 5.11: Three contours of slightly overlapped ellipses. The three cases are already sufficient for the competition task. (a) The contour corresponds to two overlapped ellipses: $I_1 = 0.008017$; (b) The contour corresponds to three overlapped ellipses: $I_1 = 0.008192$; (c) The contour corresponds to four overlapped ellipses: $I_1 = 0.008194$. The AMIs $I_2 = I_3 = 0$ for all the three contours.

and error. For example, we partition an overlapped contour into n ($n = 2, 3, 4$) parts and fit ellipse for each part. Then the algebraic error between the overlapped contour and the n fitted ellipses can be computed. If the algebraic error is smaller than a threshold, the n ellipses can be retained as corrected ones. False contours may be detected based on AMIs, but step two can effectively remove these false detections. As shown in Figure 5.10(c), the proposed algorithm can successfully detect all the eight ellipses.

5.3.4 Summary of the Ellipse Detection Algorithm

The proposed ellipse detection algorithm is summarized in Algorithm 5.1. Several important remarks are given here. First, Algorithm 5.1 can detect both whole and partial ellipses. The case of overlapped ellipses is not considered in Algorithm 5.1 as that is a very special case. Second, the thresholds for I_1, I_2 and I_3 in Algorithm 5.1 are experiential values. They might need to be tuned according to specific tasks in practice. Finally, as aforementioned, the pre-processing can detect whole ellipses, while the post-processing can detect both whole and

Algorithm 5.1 A Robust Real-time Ellipse Detection Algorithm

1: **Preparation:** Detect contours in the image using, for example, edge detection or color thresholding.

2: **Pre-processing:** For each contour, use the OpenCV function *moments* to compute its central moments μ_{00}, μ_{11}, μ_{02}, μ_{20}, μ_{12}, μ_{21}, μ_{03} and μ_{30}. Then substitute the central moments into (5.1) to calculate I_1, I_2 and I_3. Compare the calculated I_1, I_2 and I_3 with the values in (5.4). If

$$|I_1 - 1/(16\pi^2)| < 0.0003, \quad |I_2| < 0.0000001, \quad |I_3| < 0.000001, \tag{5.7}$$

then the contour is a good candidate for a whole ellipse.

3: **Ellipse fitting:**

a) For each contour that satisfies (5.7), use the OpenCV function *fitEllipse* to fit an ellipse.

b) For each contour that does not satisfy (5.7), use the OpenCV function *convexHull* to obtain the convex hull of the contour, and then fit an ellipse for the convex hull.

4: **Post-processing:**

a) For each contour that satisfies (5.7), compute its algebraic error e_{alg} as defined in (5.6).

b) For each contour that does not satisfy (5.7), compute its algebraic error e_{alg} between its convex hull and the fitted ellipse.

If $|e_{\text{alg}}| < 0.1$, then the fitted ellipse can be confirmed as a correct one.

partial ones. They need to be used flexibly in practice according to the specific task requirements.

5.4 Ellipse Tracking

After the ellipse detection procedure, multiple ellipses may be detected in an image. But we are only interested in one specific ellipse and the relative position of the corresponding circle. We next present an ellipse tracking algorithm which can track an ellipse of interest over the image sequence consistently. The details of the algorithm are given in Algorithm 5.2. Several important remarks are made as follows.

1) The scale and shape of the target ellipse may keep changing when the camera and the target circle are relatively moving. The color of the ellipse area may also vary a little when observed from different angles. Therefore, the histogram of the target ellipse area needs to be updated continuously. In the histogram-update law as shown in (5.8), the new histogram is a convex

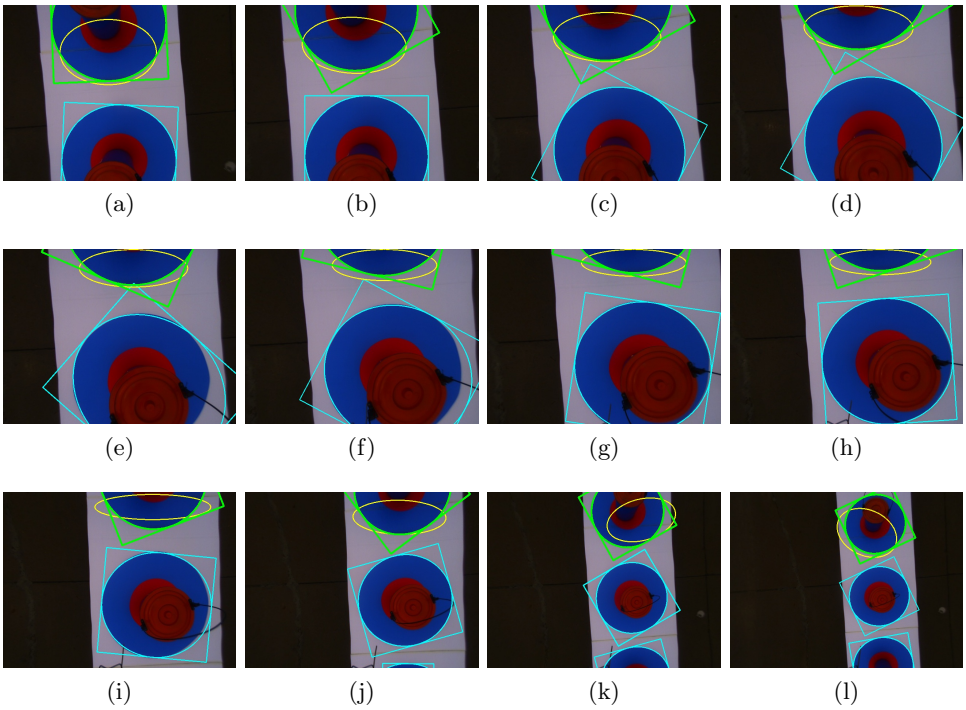


Figure 5.12: Examples to illustrate ellipse tracking over consecutive images. In each image, all ellipses have been detected and drawn in cyan. The tracked ellipse is highlighted in green. The yellow ellipse is the target area returned by CAMShift.

combination of the old one and the one of the ellipse area in frame k. If $w = 1$, there will be no histogram update; if $w = 0$, the histogram would be exactly the one of the area enclosed by the target ellipse in frame k. According to a series of experiments, we choose $w = 0.95$ in our work.

2) The histogram $\tilde{h}_k$ should be the one for the exact area enclosed by the target ellipse. But in order to speed up the histogram computation, $\tilde{h}_k$ actually is for the area enclosed by the tangent rectangle of the ellipse. The tangent rectangle contains not only the ellipse area but also some background area out of the ellipse. In practice, it is recommended to scale down the tangent rectangle a little in order reduce the effect of the background area on $\tilde{h}_k$.

3) The CAMShift is performed over the hue channel of the color image. That is because the ellipses are most distinguishable from the background in the hue channel in the competition task.

4) The initialization procedure for the first frame is omitted in Algorithm 5.2 as it may vary from task to task. For our specific task, there are totally eight circles. When the corresponding eight ellipses can be detected in the image,

Algorithm 5.2 Ellipse Tracking based on CAMShift

Require:

1: Update the tracking window: choose the initial tracking window for CAMShift as the tangent rectangle of the target ellipse in frame k.

2: Update the histogram: denote h_k and h_{k+1} as the histogram used for tracking in frame k and $k+1$, respectively. Let $\tilde{h}_k$ be the histogram of the target ellipse area in frame k. Then update h_{k+1} by

$$h_{k+1} = wh_k + (1-w)\tilde{h}_k, \tag{5.8}$$

where $w \in [0, 1]$ is a weight factor.

3: Compute the back projection image based on the updated histogram. This can be done by using the OpenCV function *calcBackProject*.

4: Obtain the final tracking window computed by CAMShift. This can be done by substituting the back projection image and the initial tracking window to the OpenCV function *CamShift*.

5: Find the ellipse that is located closest to the final tracking window given by CAMShift.

the initialization procedure is triggered to select one ellipse for tacking.

Figure 5.12 shows examples to illustrate the ellipse tracking over an image sequence. As can be seen, the target ellipse is tracked robustly though its position and scale keep changing. Moreover, since the target circle is only partially observable in some images, the histogram of the area enclosed by the partial ellipse is dynamically varying. But thanks to the histogram update law (5.8), the target ellipse can still be tracked robustly.

5.5 Single-Circle-based Pose Estimation

After the ellipse of interest has been obtained, it can be used to estimate the pose of the corresponding circle. Pose estimation from 3D-2D point correspondences has been investigated extensively (see, for example, [55, 101, 77] and the references therein). But unlike the existing studies in [71, 65, 76, 40, 110], we aim at recovering the pose of a *single circle*. The main contribution of our work is that we analyze the specific properties of circle-based pose estimation, and identify *four point correspondences* that can be used for pose estimation. Based on the four correspondences, various mature algorithms such as the OpenCV function *solvePnP* can be applied to solve the pose estimation problem.

5.5.1 Pose Estimation from Four Point Correspondences

We first introduce three coordinate frames involved in the pose estimation problem: world frame, camera frame and image frame. Without loss of generality, we set the world frame in the following way: the origin of the world frame coincides with the circle center; and the Z-axis of the world frame is orthogonal to the plane Π_1 that contains the circle. Thus the Z component of any points on the circle is zero, and the coordinate of the circle center is $(0, 0, 0)$ in the world frame. The camera frame has its origin located at the camera center and its z-axis orthogonal to the image plane Π_2. The image frame defined in OpenCV has been shown in Figure 5.7. The pose estimation problem we are going to solve is summarized as below.

Problem 5.1. *Suppose an ellipse has already been detected in an image, and the parameters of the ellipse in the image frame are known. The ellipse is the perspective projection of a circle. Given the diameter of the circle and the intrinsic parameters of the camera, estimate the coordinate of the circle center in the camera frame.*

Remark 5.1. *In our task, the circles are used for vision-based guidance, and we are only interested in the position of the circle center. The attitude of the camera relative to the circle is not of our interests because the attitude of the UAV can be obtained from GPS and inertial sensors. As will be shown later, we only need to estimate the translation from the world frame to the camera frame in order to solve Problem 5.1, while the rotation is not required.*

Consider an arbitrary point on the circle. Denote $p = [X, Y, Z]^{\mathrm{T}} \in \mathbb{R}^3$ and $q = [x, y, z]^{\mathrm{T}} \in \mathbb{R}^3$ as the coordinates of the point in the world and the camera frame, respectively. Let $R \in \mathbb{R}^{3\times3}$ and $T \in \mathbb{R}^3$ be the rotational and translational transformation from the world frame to the camera frame, respectively. Then we have

$$q = Rp + T. \tag{5.9}$$

Use the subscript c to denote the coordinate of the circle center. Then we have

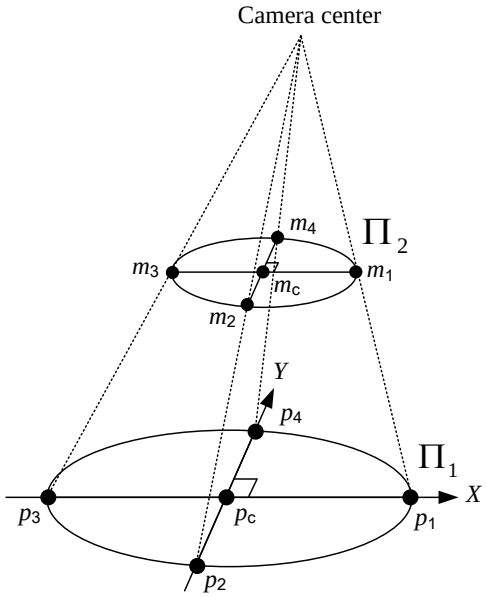


Figure 5.13: Perspective projection of a circle and the four point correspondences.

$p_c = [0, 0, 0]^{\mathrm{T}}$ and hence $q_c = T$. Therefore, the translation T is the one we need to estimate. The rotation R is not required to be estimated.

We next identify four sets of corresponding points on the ellipse and the circle, respectively. Let m_1, m_2, m_3 and m_4 be the four vertexes of the ellipse in the image (see Figure 5.13). Given the parameters of the ellipse such as (u_c, v_c), a, b, α as shown in Figure 5.7, the coordinates of the four points can be easily obtained as $m_1 = (u_c + a\cos\alpha, v_c + a\sin\alpha)$, $m_2 = (u_c - b\sin\alpha, v_c + b\cos\alpha)$, $m_3 = (u_c - a\cos\alpha, v_c - a\sin\alpha)$ and $m_4 = (u_c + b\sin\alpha, v_c - b\cos\alpha)$. Let p_1, p_2, p_3 and p_4 be the four corresponding points on the circle. Point p_i corresponds to m_i for $i = 1, \ldots, 4$. The following assumption on $p_1, \ldots, p_4$ is a key to our algorithm.

Assumption 5.1. *The four points $p_1, \ldots, p_4$ are evenly distributed on the circle, which means p_1, p_c and p_3 are collinear, p_2, p_c and p_4 are collinear and the line p_1p_3 is perpendicular to p_2p_4.*

The necessary and sufficient condition of Assumption 5.1 will be proved later. Under Assumption 5.1, we are able to set the world frame such that the four points $p_1, \ldots, p_4$ are located at the X- and Y-axes, respectively (see Figure 5.13). As a result, if the diameter of the circle is d, we have $p_1 = (d/2, 0, 0)$, $p_2 = (0, -d/2, 0)$, $p_3 = (-d/2, 0, 0)$ and $p_4 = (0, d/2, 0)$. It is worth noting

Algorithm 5.3 Single-Circle-based Pose Estimation from Four Point Correspondences

1: Obtain the four image points, which are the vertexes of the ellipse:

$$m_1 = (u_c + a\cos\alpha, v_c + a\sin\alpha), \quad m_2 = (u_c - b\sin\alpha, v_c + b\cos\alpha),$$
$$m_3 = (u_c - a\cos\alpha, v_c - a\sin\alpha), \quad m_4 = (u_c + b\sin\alpha, v_c - b\cos\alpha).$$

2: Obtain the four corresponding points on the circle:

$$p_1 = (d/2, 0, 0), \quad p_2 = (0, -d/2, 0), \quad p_3 = (-d/2, 0, 0), \quad p_4 = (0, d/2, 0).$$

3: Substitute the four correspondences $\{m_1, p_1\}$, $\{m_2, p_2\}$, $\{m_3, p_3\}$ and $\{m_4, p_4\}$ to the OpenCV function *solvePnP*. Suppose T is the translational vector returned by *solvePnP*. Then T is the translation from the world frame to the camera frame. Thus $q_c = T$.

that the orientation of the world frame does not affect $q_c = T$, and hence its orientation can be set freely. Thus we obtain four sets of point correspondences $\{m_1, p_1\}$, $\{m_2, p_2\}$, $\{m_3, p_3\}$ and $\{m_4, p_4\}$, which can be consequently used for pose estimation. Our pose estimation algorithm is summarized in Algorithm 5.3.

Although Algorithm 5.3 is designed for *single-circle-based* pose estimation, it can also be applied to *multi-circle-based* pose estimation problems. The basic idea is to estimate the coordinate of each circle center first and then estimate the pose of the entire multi-circle pattern. For example, in the competition task, there are totally eight circles on the two ships. The relative positions of the eight circles are known in advance. If we can detect an eight-circle pattern in the image, the pose (both rotation and translation) of the camera relative to the ships can be determined.

5.5.2 Analysis of Assumption 5.1

Assumption 5.1 plays a key role in the propose pose estimation algorithm. We now analyze when Assumption 5.1 is valid. The following theorem gives the necessary and sufficient condition of Assumption 5.1.

Theorem 5.1. *Assumption 5.1 holds if and only if the ellipse center coincides with the projection of the circle center.*

Proof. The proof is based on the fact that collinearity is preserved under perspective projection.

Necessity: Assumption 5.1 implies that p_c is collinear with p_1 and p_3. Thus the projection point of p_c is collinear with m_1 and m_3. Analogously the projection of p_c is also collinear with m_2 and m_4. As a result, the projection of p_c is the intersection of the lines m_1m_3 and m_2m_4. Denote m_c as the center of the ellipse. Since m_c is also the intersection of m_1m_3 and m_2m_4, we have m_c coincides with the projection of p_c.

Sufficiency: In order to prove the sufficiency, we only need to prove: first, the lines p_1p_3 and p_2p_4 intersect at p_c; second, the two lines p_1p_3 and p_2p_4 are perpendicular to each other. Firstly, if m_c coincides with the projection of p_c, we have p_1, p_3 and p_c are also collinear because m_1, m_3 and m_c are collinear. Similarly we have p_2, p_4 and p_c collinear. Therefore, the lines p_1p_3 and p_2p_4 intersect at p_c. Secondly, note $\|m_1-m_c\| = \|m_3-m_c\|$ and $\|p_1-p_c\| = \|p_3-p_c\|$, where $\|\cdot\|$ denotes the Euclidean norm of a vector. As a result, if m_c is the perspective projection of p_c, it is easy to prove that p_1p_3 is parallel to m_1m_3 based on similar triangles. Analogously, it can also be proved that p_2p_4 is parallel to m_2m_4. Thus, p_1p_3 is perpendicular to p_2p_4 because m_1m_3 is perpendicular to m_2m_4. □

It has been proved in [57] that the ellipse center generally does not coincide with the projection of the circle center. Next we further show when they coincide with each other.

Theorem 5.2. *The ellipse center coincides with the projection of the circle center if and only if the image plane is parallel to the plane that contains the circle.*

Proof. Sufficiency: See [57, Section 2.3]. We give a brief explanation here. The work in [57] does not state the sufficiency of Theorem 5.2 explicitly. But it is stated that the projection of the circle center is the same as the ellipse center if "the rotation is performed around the Z-axis" (see the paragraph below equation (16) in [57]). The condition that the image plane is parallel to the object plane in Theorem 5.2 actually is equivalent to the condition that "the rotation is performed around the Z-axis" in [57].

Necessity: If m_c is the projection of p_c, we know p_c is collinear with p_1 and

p_3. Furthermore, since $\|m_1 - m_c\| = \|m_3 - m_c\|$ and $\|p_1 - p_c\| = \|p_3 - p_c\|$, it is easy to prove that p_1p_3 is parallel to m_1m_3 based on similar triangles. It can be analogously proved that p_2p_4 is parallel to m_2m_4. Since the image plane Π_2 contains m_1m_3 and m_2m_4 while the plane Π_1 contains p_1p_3 and p_2p_4, the two planes are parallel to each other. □

Remark 5.2. *The work in [57] merely mentioned that the case of two parallel planes is one of the examples that guarantee the ellipse center to coincide with the projection of the circle center. The contribution of Theorem 5.2 is that it shows that this case is the only example.*

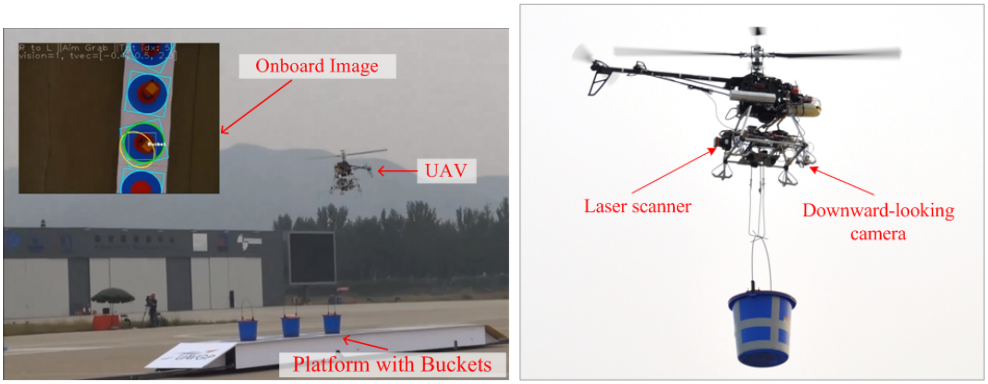
Theorems 5.1 and 5.2 clearly indicate that Assumption 5.1 is valid if and only if the image plane is parallel to the plane that contains the circle. For our competition task, the circles are placed horizontally on the ground. The onboard camera is installed on a pan-tilt mechanism such that the image plane can be automatically controlled to be horizontal. Thus Assumption 5.1 is always satisfied in our system. But we will later show by experimental results that even the assumption is not satisfied, the accuracy of the proposed four-point pose estimation algorithm is still good.

5.6 Experimental and Competition Results

Our vision system including the proposed algorithms on ellipse detection, ellipse tracking and circle-based pose estimation have been implemented on the unmanned helicopter for the 2013 UAVGP. Successful completion of all competition tasks is the strongest evidence for the accuracy and robustness of the vision system. A video of the onboard vision processing for the final round competition can be found at http://youtu.be/GSeafBsASTs.

5.6.1 Flight Data in the Competition

Figure 5.14 shows the helicopter UAV developed by our UAS Group at NUS. Although the UAV is equipped with a GPS, the accuracy of the GPS is much worse than that of the vision system. As a result, the GPS data cannot be used as the ground truth to verify the accuracy of the vision estimation. Fortunately,



(a) (b)

Figure 5.14: The helicopter UAV in the competition. (a) The UAV is approaching to a "ship" to grab a bucket. (b) The UAV is flying with a bucket.

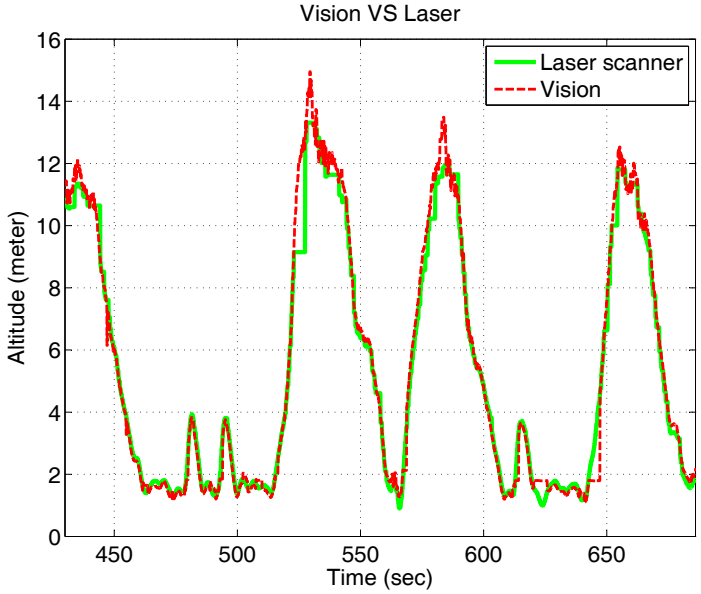


Figure 5.15: The altitude measurements given by the vision system and the laser scanner.

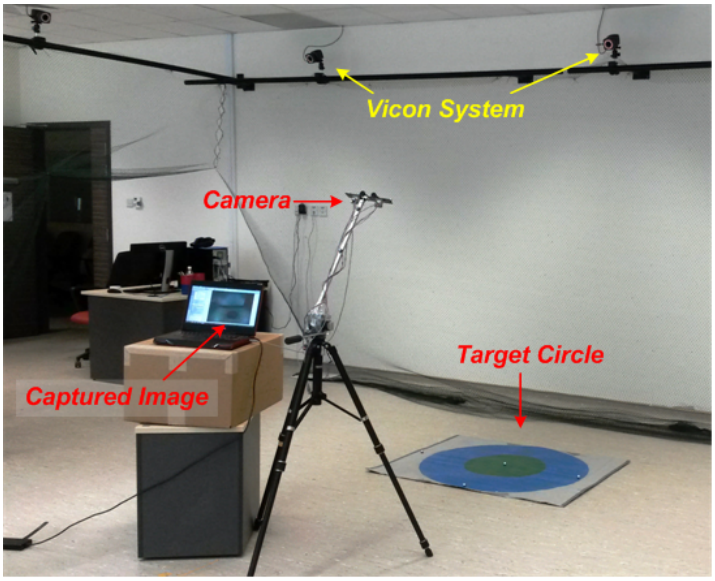


Figure 5.16: Experiment setup in a Vicon system to verify Algorithm 5.3.

a laser scanner is mounted at the back of the UAV to measure the accurate altitude (see Figure 5.14(b)). Since the vision system can estimate the vertical distance from the target circle to the UAV, we can estimate the UAV altitude from vision by adding an offset (the height of the ship which is known) to the vision estimates. Thus we can compare the altitude estimates by vision with the laser measurements. It is worth noting that the laser measurement is particularly accurate in general, and hence it can be viewed as the ground truth of the altitude. The flight data as shown in Figure 5.15 indicates that the vision estimates are very accurate as they coincide with the laser measurements very well.

5.6.2 Experiments for Algorithm 5.3

Experiments have been conducted to verify the accuracy of Algorithm 5.3. As shown in Figure 5.16, the experiments are conducted in a Vicon system. The Vicon system is used to obtain the ground truth of the range from the circle center to the camera center. The diameter of the target circle in our experiment is one meter.

The images of the target circle are given in Figure 5.17. Based on the detected ellipse, the coordinate of the circle center in the camera frame can be estimated by Algorithm 5.3. The estimation results are given in Table 5.2. As can be

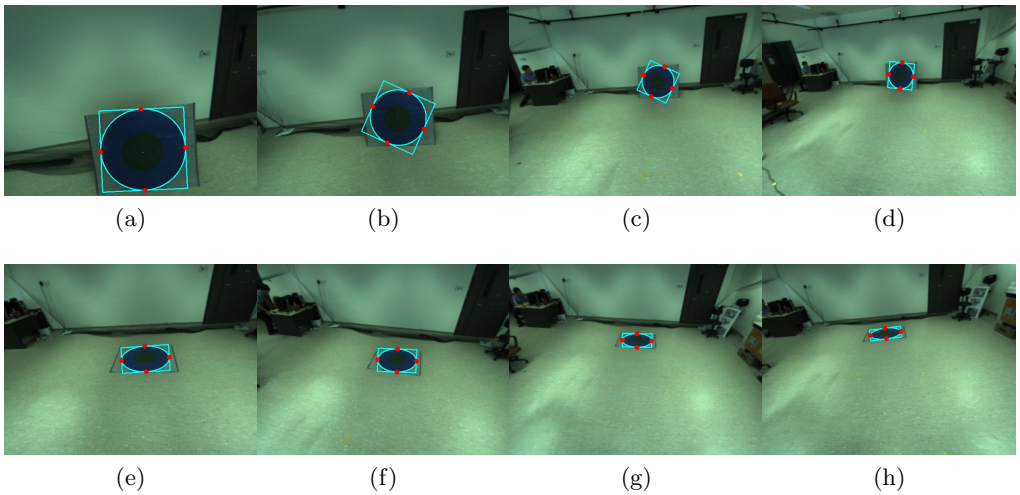


Figure 5.17: Images captured in the experiment. From (a) to (d), the target circle is placed almost vertically; from (e)-(h), the target circle is placed horizontally on the floor. The detected ellipse is drawn on each image. The four red dots drawn on each ellipse are the detected vertexes of the ellipse. The size of each image is 640 × 480 pixels.

Table 5.2: Pose estimation results using the images in Figure 5.17.

Image	$\|q_c\|$ by vision	$\|q_c\|$ by Vicon	Abs error of $\|q_c\|$
(a)	1.667 m	1.650 m	0.017 m
(b)	2.383 m	2.363 m	0.020 m
(c)	4.201 m	4.218 m	0.017 m
(d)	5.304 m	5.253 m	0.051 m
(e)	2.789 m	2.817 m	0.028 m
(f)	3.425 m	3.450 m	0.025 m
(g)	4.513 m	4.486 m	0.027 m
(h)	4.337 m	4.295 m	0.042 m

seen, the proposed algorithm can give very accurate estimates of the range from the circle center to the camera center. Specifically, when the range is shorter than 5 meter, the estimation error is smaller than 0.05 meter. It is worth noting that Assumption 5.1 is not satisfied for the cases in Figure 5.17(e)-(h), but the estimates are still very accurate in those cases.

5.6.3 Efficiency Test

We also have conducted experiments to test the efficiency of the proposed algorithms. The algorithms were tested on a laptop with its CPU as Intel Core i5-2520M 2.5 GHz. We used the proposed three algorithms to process 100 consecutive images (640x480 pixel) captured by the onboard camera in the competition. The average time consumption of each algorithm (in debug mode) is given in Table 5.3. As can be seen, the algorithms for ellipse detection, tracking

Table 5.3: Time consumption of each procedure in the vision system.

Algorithm	Time Consumption
Undistortion	0.0240 sec
RGB to HSV	0.0093 sec
Color Thresholding	0.0042 sec
Contour Detection	0.0011 sec
Ellipse Detection	0.0012 sec
Ellipse Tracking	0.0084 sec
Circle-based Pose Estimation	0.0030 sec

and pose estimation are very efficient. The most time-consuming procedure is image undistortion. In order to improve the efficiency of the image undistortion, we have written a new undistortion function to replace the OpenCV one *undistort*. The core idea is to use the OpenCV function *undistort* just once to build a lookup table. Then every image can be undistorted by simply indexing the lookup table, and the undistortion time can be reduced to around 0.005 sec.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis focused on the area of control and navigation of multi-vehicle systems using visual measurements. It presented both theoretical and practical contributions to this area.

One main theoretical contribution of this thesis is that we successfully solved the optimal placement of sensor networks for target tracking in 2D and 3D spaces. This problem is very important yet not solved by the existing studies. We successfully proved the necessary and sufficient conditions for optimal sensor placement in 2D and 3D spaces. We also established a unified framework for analyzing the optimal placement of different types of sensors.

One main novelty of the thesis is that we formulated a new formation control problem: control of multi-vehicle systems using bearing-only measurements. Our idea on bearing-only formation control was motivated by vision-based formation control. In conventional frameworks for vision-based formation control, it is assumed that vision can provide the relative positions of the neighbors of one vehicle. This assumption is, however, hard to be implemented in practice. If vision is only required to provide bearing measurements, the burden on the end of vision will be significantly reduced and consequently the efficiency and reliability of the vision system can be greatly enhanced. As a result, bearing-only formation control provides a novel and practical framework for implementing vision-based formation control tasks. The research presented in this thesis solved

an important special case: stabilization of cyclic formations using bearing-only measurements. Of course, more research on this topic is needed in the future.

In addition to the theoretical research on control of multi-vehicle systems, we also investigated an important application-oriented research topic: vision-based navigation of UAVs using either natural or artificial landmarks. Our main contrition for this part is that we presented systematic solutions to vision-based navigation: the proposed algorithms were not only analyzed theoretically but also verified by flight experiments. It was shown that the proposed navigation systems are efficient and reliable. Specifically, the vision-based navigation system presented in Chapter 5 was applied to the 2013 UAVGP competition and helped our team made a great success in the competition. Completing all the required competition tasks is the strongest evidence of the efficiency and reliability of our proposed navigation system.

6.2 Future Work

The research presented in this thesis suggests a number of interesting and important future research directions.

1) In this thesis, we addressed optimal placement of sensor networks for tracking one single target. It is of great practical interests to study optimal placement of sensor networks for covering an area instead of just one point. For example, suppose the sensor network is fixed while the target may move within a bounded area. Then it is interesting to study how to place the sensor network optimally such that the average target tracking accuracy can be maximized. It is conjectured that analytical solutions can be obtained only for certain areas with regular shapes.

2) We proposed a gradient control law to autonomously deploy optimal placements of sensor networks. But the proposed control law is a centralized one based on all-to-all communications. It is meaningful to study distributed optimization algorithms or control laws that can distributedly deploy optimal sensor placements. In addition, our work only focused on homogeneous sensor networks. It is also important to analytically characterize 2D or 3D optimal

placements of heterogeneous sensors in the future.

3) Our research on control of cyclic formations using bearing-only measurements is a first but important step towards solving generic bearing-based formation control problems. Formations with more complicated underlying graphs need to be studied in the future. Moreover, in order to control the formation scale, bearing-only constraints and measurements would be insufficient; distance constraints and measurements need be considered. Distributed control of formations with mixed bearing and distance constraints using mixed measurements is of both theoretical and practical importance.

4) Since bearing-only formation control is motivated by vision-based formation control tasks, implementation of the proposed control laws certainly is meaningful. There are a number of problems on the end of vision in practical implementation. For example, the field of view (FOV) of a camera usually is limited. With limited FOV, one vehicle may not be able to observe its neighbors continuously. In addition, it is also an interesting research topic whether and how stereo vision can be applied to vision-based formation control.

5) Cooperative vision-based navigation of multiple vehicles is a hot research topic in recent years. The vision-based navigation systems presented in this thesis, however, are mainly designed for navigating one single vehicle. It is meaningful to study how to apply the proposed vision-based navigation systems to cooperative navigation tasks.

Bibliography

[1] S. J. Ahn, W. Rauh, and H.-J. Warnecke. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. *Pattern Recognition*, 34:2283–2303, 2001.

[2] J. G. Allen, R. Y. D. Xu, and J. S. Jin. Object tracking using CamShift algorithm and multiple quantized feature spaces. In *Proceedings of the Pan-Sydney area workshop on Visual information processing*, pages 3–7, Darlinghurst, Australia, 2004.

[3] A. Bacciotti and F. Ceragioli. Stability and stabilization of discontinuous systems and nonsmooth Lyapunov functions. *ESAIM: Control, Optimisation and Calculus of Variations*, 4:361–376, January 1999.

[4] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[5] M. Basiri, A. N. Bishop, and P. Jensfelt. Distributed control of triangular formations with angle-only constraints. *Systems & Control Letters*, 59:147–154, 2010.

[6] J. J. Benedetto and M. Fickus. Finite normalized tight frames. *Advances in Computational Mathematics*, 18:357–385, 2003.

[7] S. P. Bhat and D. S. Bernstein. Finite-time stability of continuos autonomous systems. *SIAM Journal on Control and Optimization*, 38(3):751–766, 2000.

[8] A. N. Bishop. Distributed bearing-only quadrilateral formation control. In *Proceedings of the 18th IFAC World Congress*, pages 4507–4512, Milano, Italy, August 2011.

[9] A. N. Bishop. Stabilization of rigid formations with direction-only constraints. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 746–752, Orlando, FL, USA, December 2011.

[10] A. N. Bishop. A very relaxed control law for bearing-only triangular formation control. In *Proceedings of the 18th IFAC World Congress*, pages 5991–5998, Milano, Italy, August 2011.

[11] A. N. Bishop, B. Fidan, B. D. O. Anderson, K. Doğançay, and P. N. Pathirana. Optimality analysis of sensor-target localization geometries. *Automatica*, 46:479–492, 2010.

[12] A. N. Bishop, B. Fidan, B. D. O. Anderson, P. N. Pathirana, and K. Doğançay. Optimality analysis of sensor-target geometries in passive localization: Part 2 - time-of-arrival based localization. In *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 13–18, Melbourne, Australia, December 2007.

[13] A. N. Bishop and P. Jensfelt. An optimality analysis of sensor-target geometries for signal strength based localization. In *Proceedings of the 5th International Conference on Intelligent Sensors, Sensor Networks, and Information Processing*, pages 127–132, Melbourne, Australia, December 2009.

[14] A. N. Bishop and M. Smith. Remarks on the cramer-rao inequality for doppler-based target parameter estimation. In *Proceedings of the 6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 199–204, Brisbane, Australia, December 2010.

[15] A. N. Bishop, T. H. Summers, and B. D. O. Anderson. Stabilization of stiff formations with a mix of direction and distance constraints. In

Proceedings of the 2013 IEEE Multi-Conference on Systems and Control, 2013. to appear.

[16] F. Bonin-Font, A. Ortiz, and G. Oliver. Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53:263–296, 2008.

[17] M. Bryson and S. Sukkarieh. Observability analysis and active control for airborne SLAM. *IEEE Transactions on Aerospace and Electronic Systems*, 44(1):261–280, January 2008.

[18] F. Caballero, L. Merino, J. Ferruz, and A. Ollero. Vision-based odometry and SLAM for medium and high altitude flying UAVs. *Journal of Intelligent and Robotic Systems*, 54:137–161, 2009.

[19] G. Cai, B. M. Chen, and T. H. Lee. *Unmanned Rotorcraft Systems*. Springer, New York, 2011.

[20] M. Cao and A. S. Morse. An adaptive approach to the range-only station-keeping problem. *International Journal of Adaptive Control and Signal Processing*, 26:757–777, 2012.

[21] M. Cao, C. Yu, and B. D. O. Anderson. Formation control using range-only measurements. *Automatica*, 47:776–781, 2011.

[22] Y. Cao, W. Ren, and Z. Meng. Decentralized finite-time sliding mode estimators and their applications in decentralized finite-time formation tracking. *Systems & Control Letters*, 59:522–529, 2010.

[23] P. G. Casazza, M. Fickus, J. Kovačević, M. T. Leon, and J. C. Tremain. A physical interpretation of tight frames. In *Harmonic Analysis and Applications*, Applied and Numerical Harmonic Analysis, pages 51–76. Birkh 2006.

[24] P. G. Casazza, M. Fickus, and D. G. Mixon. Auto-tuning unit norm frames. *Applied and Computational Harmonic Analysis*, 32:1–15, 2012.

[25] P. G. Casazza and M. T. Leon. Existence and construction of finite tight frames. *Journal of Computational and Applied Mathematics*, 4(3):277–289, 2006.

[26] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Wiley, New York, 1983.

[27] G. Conte and P. Doherty. Vision-based unmanned aerial vehicle navigation using geo-referenced information. *EURASIP Journal on Advances in Signal Processing*, pages 1–18, 2009.

[28] J. Cortés. Discontinuous dynamical systems. *IEEE Control Systems Magazine*, 28(3):36–73, 2008.

[29] J. Cortés. Global and robust formation-shape stabilization of relative sensing networks. *Automatica*, 45:2754–2762, 2009.

[30] J. Cortés and F. Bullo. Coordination and geometric optimization via distributed dynamical systems. *SIAM Journal on Control and Optimization*, 44(5):1543–1574, 2005.

[31] M. Defoort, T. Floquet, A. Kökösy, and W. Perruquetti. Sliding-mode formation control for cooperative autonomous mobile robots. *IEEE Transactions on Industrial Electronics*, 55(11):3944–3953, November 2008.

[32] L. Di, T. Fromm, and Y. Chen. A data fusion system for attitude estimation of low-cost miniature UAVs. *Journal of Intelligent and Robotic Systems*, August 2011.

[33] X. Dong. *Development of sophisticated unmanned software systems and applications to UAV formation*. PhD thesis, National University of Singapore, 2012.

[34] X. Dong, B. M. Chen, G. Cai, H. Lin, and T. H. Lee. Development of a comprehensive software system for implementing cooperative control of multiple unmanned aerial vehicles. *International Journal of Robotics and Automation*, 26(1):49–63, 2011.

[35] F. Dörfler and B. Francis. Formation control of autonomous robots based on cooperative behavior. In *Proceedings of the 2009 European Control Conference*, pages 2432–2437, Budapest, Hungary, 2009.

[36] F. Dörfler and B. Francis. Geometric analysis of the formation problem for autonomous robots. *IEEE Transactions on Automatic Control*, 55(10):2379–2384, October 2010.

[37] K. Doğançay. Online optimization of receiver trajectories for scan-based emitter localization. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):1117–1125, July 2007.

[38] K. Doğançay and H. Hmam. Optimal angular sensor separation for AOA localization. *Signal Processing*, 88:1248–1260, 2008.

[39] K. Dykema and N. Strawn. Manifold structure of spaces of spherical tight frames. *International Journal of Pure and Applied Mathematics*, 28:217–256, 2006.

[40] D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart. Vision based position control for MAVs using one single circular landmark. *Journal of Intelligent and Robotic Systems*, 61:495–512, 2011.

[41] T. Eren. Formation shape control based on bearing rigidity. *International Journal of Control*, 85(9):1361–1379, 2012.

[42] T. Eren, W. Whiteley, A. S. Morse, P. N. Belhumeur, and B. D. O. Anderson. Sensor and network topologies of formations with direction, bearing and angle information between agents. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 3064–3069, Hawaii, USA, December 2003.

[43] Y. Fang, X. Liu, and X. Zhang. Adaptive active visual servoing of nonholonomic mobile robots. *IEEE Transactions on Industrial Electronics*, 59(1):486–497, January 2012.

[44] D. Feng, L. Wang, and Y. Wang. Generation of finite tight frames by householder transformations. *Advances in Computational Mathematics*, 24:297–309, 2006.

[45] M. Fickus. *Finite normalized tight frames and spherical equidistribution.* PhD thesis, University of Maryland, 2001.

[46] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides.* Kluwer Academic Publishers, 1988.

[47] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, May 1999.

[48] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Pattern Recognition*, 26(1):167–174, January 1993.

[49] A. Franchi and P. R. Giordano. Decentralized control of parallel rigid formations with direction constraints and bearing measurements. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 5310–5317, Hawaii, USA, December 2012.

[50] C. Godsil and G. Royle. *Algebraic Graph Theory.* Springer, New York, 2001.

[51] V. K. Goyal, J. Kovačević, and J. A. Kelner. Quantized frame expansions with erasures. *Applied and Computational Harmonic Analysis*, 10(3):203–233, May 2001.

[52] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas. Cooperative air and ground surveillance. *IEEE Robotics & Automation Magazine*, 13(3):16–25, September 2006.

[53] P. D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems.* Artech House, 2008.

[54] W. M. Haddad and V. Chellaboina. *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach.* Princeton University Press, 2008.

[55] R. M. Haralick, H. Joo, D. Lee, S. Zhuang, V. G. Vaidya, and M. B. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1426–1446, 1989.

[56] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[57] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, San Juan, Puerto Rico, 1997.

[58] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[59] A. M. Howard, B. M. Jones, and N. Serrano. Integrated sensing for entry, descent, and landing of a robotic spacecraft. *IEEE Transactions on Aerospace and Electronic Systems*, 47(1):295–304, January 2011.

[60] J. Hu, J. Xu, and L. Xie. Cooperative search and exploration in robotic networks. *Unmanned Systems*, 1(1):121–142, 2013.

[61] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962.

[62] Y. Hu, W. Zhao, and L. Wang. Vision-based target tracking and collision avoidance for two autonomous robotic fish. *IEEE Transactions on Industrial Electronics*, 56(5):1401–1410, May 2009.

[63] H. Huang, C. Yu, and Q. Wu. Autonomous scale control of multiagent formations with only shape constraints. *International Journal of Robust and Nonlinear Control*, 23(7):765–791, May 2013.

[64] J. T. Isaacs, D. J. Klein, and J. P. Hespanha. Optimal sensor placement for time difference of arrival localization. In *Procedings of the 48th Conference on Decision and Control*, pages 7878–7884, Shanghai, China, December 2009.

[65] G. Jiang and L. Quan. Detection of concentric circles for camera calibration. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, pages 333–340, Beijing, China, 2005.

[66] D. B. Jourdan and N. Roy. Optimal sensor placement for agent localization. In *Proceedings of the IEEE/ION Position, Location, and Navigation Symposium*, pages 128–139, San Diego, USA, April 2006.

[67] M. K. Kaiser, N. R. Gans, and W. E. Dixon. Vision-based estimation for guidance, navigation, and control of an aerial vehicle. *IEEE Transactions on Aerospace and Electronic Systems*, 46(3):137–161, July 2010.

[68] H. K. Khalil. *Nonlinear Systems, Third Edition.* Prentice Hall, 2002.

[69] J. Kim and S. Sukkarieh. Autonomous airborne navigation in unknown terrain environments. *IEEE Transactions on Aerospace and Electronic Systems*, 40:1031–1045, 2004.

[70] J. Kim and S. Sukkarieh. Real-time implementation of airborne inertial-SLAM. *Robotics and Autonomous Systems*, 55:62–71, 2007.

[71] J.-S. Kim, P. Gurdjos, and I.-S. Kweon. Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):637–642, 2005.

[72] J. Kovačević and A. Chebira. Life beyond bases: the advent of frames (part I). *IEEE Signal Processing Magazine*, 24(4):86–104, July 2007.

[73] J. Kovačević and A. Chebira. Life beyond bases: the advent of frames (part II). *IEEE Signal Processing Magazine*, 24(5):115–125, September 2007.

[74] L. Krick, M. E. Broucke, and B. A. Francis. Stabilization of infinitesimally rigid formations of multi-robot networks. *International Journal of Control*, 82(3):423–439, 2009.

[75] H. Lang, M. T. Khan, K.-K. Tan, and C. W. de Silva. Developments in visual servoing for mobile manipulation. *Unmanned Systems*, 1(1):143–162, July 2013.

[76] S. Lange, N. Sunderhauf, and P. Protzel. A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments. In *Proceedings of the 2009 International Conference on Advanced Robotics*, pages 1–6, Munich, Germany, June 2009.

[77] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate $O(n)$ solution to the PnP problem. *International Journal of Computer Vision*, 81(2):155–166, February 2009.

[78] F. Lin, K. Z. Y. Ang, F. Wang, B. M. Chen, T. H. Lee, B. Yang, M. Dong, X. Dong, J. Cui, S. K. Phang, B. Wang, D. Luo, K. Peng, G. Cai, S. Zhao, M. Yin, and K. Li. Development of an unmanned coaxial rotorcraft for the DARPA UAVForge Challenge. *Unmanned Systems*, 1(2):211–245, 2013.

[79] F. Lin, X. Dong, B. M. Chen, K. Y. Lum, and T. H. Lee. A robust real-time embedded vision system on an unmanned rotorcraft for ground target following. *IEEE Transactions on Industrial Electronics*, 59(2):1038–1049, February 2012.

[80] P. Lin and Y. Jia. Average consensus in networks of multi-agents with both switching topology and coupling time-delay. *Physica A: Statistical Mechanics and its Applications*, 387(1):303–313, January 2008.

[81] P. Liu, X. Dong, B. M. Chen, and T. H. Lee. Development of an enhanced ground control system for unmanned aerial vehicles. In *Proceedings of the IASTED International Conference on Engineering and Applied Science*, pages 136–143, Colombo, Sri Lanka, December 2012.

[82] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3D Vision*. Springer, New York, 2004.

[83] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42:661–668, 2006.

[84] R. A. McLaughlin. Randomized hough transform: Improved ellipse detection with comparison. *Pattern Recognition Letters*, 19:299–305, March 1998.

[85] E. Michaelsen, M. Kirchhof, and U. Stilla. Sensor pose inference from airborne videos by decomposing homography estimates. In *Proceedings of the 2011 Asian Control Conference*, pages 211–216, Kaohsiung, Taiwan, May 2011.

[86] B. M. Miller and E. Y. Rubinovich. *Impulsive Control in Continuous and Discrete-Continuous Systems.* Kluwer Academic/Plenum Publishres, New York, 2003.

[87] F. Mokhtarian and S. Abbasi. Shape similarity retrieval under affine transforms. *Pattern Recognition*, 35:31–41, 2002.

[88] D. Moreno-Salinas, A. M. Pascoal, and J. Aranda. Optimal sensor placement for underwater positioning with uncertainty in the target location. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pages 2308–2314, Shanghai, China, May 2011.

[89] N. Moshtagh, N. Michael, A. Jadbabaie, and K. Daniilidis. Bearing-only control laws for balanced circular formations of ground robots. In *Proceedings of Robotics: Science and Systems*, Zurich, Switzerland, June 2008.

[90] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2):264–280, April 2009.

[91] S. C. Nardone and V. J. Aidala. Observability criteria for bearings-only target motion analysis. *IEEE Transactions on Aerospace and Electronic Systems*, 17(2):162–166, March 1981.

[92] S. C. Nardone, A. G. Lindgren, and K. F. Gong. Fundamental properties and performance of conventional bearings-only target motion analysis. *IEEE Transactions on Automatic Control*, 29(9):775–787, September 1984.

[93] P. C. Niedfeldt, B. T. Carroll, J. A. Howard, R. W. Beard, B. S. Morse, and S. Pledgie. Enhanced UAS surveillance using a video utility metric. *Unmanned Systems*, 1(2):277–296, 2013.

[94] R. Olfati-Saber and R. M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, 2002.

[95] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, September 2004.

[96] Y. Oshman and P. Davidson. Optimization of observer trajectories for bearings-only target localization. *IEEE Transactions on Aerospace and Electronic Systems*, 35(3):892–902, July 1999.

[97] J. Ousingsawat and M. E. Campbell. Optimal cooperative reconnaissance using multiple vehicles. *Journal of Guidance, Control, and Dynamics*, 30(1):122–132, January–February 2007.

[98] B. Paden and S. S. Sastry. A calculus for computing Filippovs differential inclusion with application to the variable structure control of robot manipulators. *IEEE Transactions on Circuits and Systems*, 34(1):73–82, January 1987.

[99] D.-H. Park, J.-H. Kwon, and I.-J. Ha. Novel position-based visual servoing approach to robust global stability under field-of-view constraint. *IEEE Transactions on Industrial Electronics*, 59(12):4735–4752, December 2012.

[100] F. Pukelsheim. *Optimal Design of Experiments*. John Wiley, 1993.

[101] L. Quan and Z. Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, August 1999.

[102] W. Ren and Y. Cao. *Distributed Coordination of Multi-agent Networks*. Springer, New York, 2011.

[103] P. L. Rosin. Measuring shape: ellipticity, rectangularity, and triangularity. In *Proceedings of the 15th International Conference on Pattern Recognition*, pages 952–955, Barcelona, Spain, 2000.

[104] D. Scaramuzza and F. Fraundorfer. Visual odometry: Part I - the first 30 years and fundamentals. *IEEE Robotics and Automation Magazine*, 18(4):80–92, December 2011.

[105] D. Shevitz and B. Paden. Lyapunov stability theory of nonsmooth systems. *IEEE Transactions on Automatic Control*, 39(9):1910–1914, September 1994.

[106] A. J. Sinclair, R. J. Prazenica, and D. E. Jeffcoat. Optimal and feedback path planning for cooperative attack. *Journal of Guidance, Control, and Dynamics*, 31(6):1708–1715, November–December 2008.

[107] T. H. Summers, C. Yu, S. Dasgupta, and B. D. O. Anderson. Control of minimally persistent leader-remote-follower and coleader formations in the plane. *IEEE Transactions on Automatic Control*, 56(12):2778–2792, December 2011.

[108] C. N. Taylor, M. J. Veth, J. F. Raquet, and M. M. Miller. Comparison of two image and inertial sensor fusion techniques for navigation in unmapped environments. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2):946–958, April 2011.

[109] F. Wang, J. Cui, S. K. Phang, B. M. Chen, and T. H. Lee. A mono-camera and scanning laser range finder based UAV indoor navigation system. In *Proceedings of the International Conference on Unmanned Aircraft Systems*, pages 694–701, Atlanta, USA, May 2013.

[110] G. Wang, J. Wu, and Z. Ji. Single view based pose estimation from circle or parallel lines. *Pattern Recognition Letters*, 29:977–985, 2008.

[111] J. Wang and H. Cho. Micropeg and hole alignment using image moments based visual servoing method. *IEEE Transactions on Industrial Electronics*, 55(3):1286–1294, March 2008.

[112] L. Wang and F. Xiao. Finite-time consensus problems for networks of dynamic agents. *arXiv:math/0701724*, 2007.

[113] Y. Wang and E. K. Teoh. 2D affine-invariant contour matching using B-spline model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1853–1858, October 2007.

[114] A. D. Wu, E. N. Johnson, and A. A. Proctor. Vision-aided inertial navigation for flight control. *AIAA Journal of Aerospace Computing, Information and Communication*, 2(9):348–360, September 2005.

[115] F. Xiao, L. Wang, J. Chen, and Y. Gao. Finite-time formation control for multi-agent systems. *Automatica*, 45:2605–2611, 2009.

[116] S. Yang, S. A. Scherer, and A. Zell. An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *Journal of Intelligent and Robotic Systems*, 69:499–515, 2013.

[117] C. Yu, B. D. O. Anderson, S. Dasgupta, and B. Fidan. Control of minimally persistent formations in the plane. *SIAM Journal on Control and Optimization*, 48(1):206–233, 2009.

[118] H. Zhang. Two-dimensional optimal sensor placement. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(5):781–792, May 1995.

[119] J. Zhang, W. Liu, and Y. Wu. Novel technique for vision-based UAV navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 47(4):2731–2741, October 2011.

[120] J. Zhang, Y. Wu, W. Liu, and X. Chen. Novel approach to position and orientation estimation in vision-based UAV navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 46(2):687–700, April 2010.

[121] S.-C. Zhang and Z.-Q. Liu. A robust, real-time ellipse detector. *Pattern Recognition*, 38:273–287, 2005.

[122] S. Zhao, B. M. Chen, and T. H. Lee. Optimal placement of bearing-only sensors for target localization. In *Procedings of the 2012 Amarican Control Conference*, pages 5108–5113, Montreal, Canada, June 2012.

[123] S. Zhao, X. Dong, J. Cui, Z. Y. Ang, F. Lin, K. Peng, B. M. Chen, and T. H. Lee. Design and implementation of homography-based vision-aided inertial navigation of UAVs. In *Proceedings of the 32nd Chinese Control Conference*, pages 5101–5106, Xi'an, China, July 2013.

[124] S. Zhao, F. Lin, K. Peng, B. M. Chen, and T. H. Lee. Homography-based vision-aided inertial navigation of UAVs in unknown environments. In *Proceedings of the 2012 AIAA Guidance, Navigation, and Control Conference*, Minneapolis, Minnesota, USA, August 2012.

List of Author's Publications

Journal

[J1] F. Wang, P. Liu, **S. Zhao**, B. M. Chen, S. K. Phang, S. Lai, and T. H. Lee, "Development of an unmanned rotorcraft system for the International UAV Innovation Grand Prix," to be submitted for journal publication.

[J2] **S. Zhao**, F. Lin, K. Peng, X. Dong, B. M. Chen, and T. H. Lee, "Vision-aided navigation for small-scale UAVs in GPS-denied environments," submitted for journal publication.

[J3] **S. Zhao**, Z. Hu, M. Yin, K. Z. Y. Ang, P. Liu, F. Wang, X. Dong, F. Lin, B. M. Chen, and T. H. Lee, "A robust real-time vision system for an unmanned helicopter transporting cargoes between moving platforms," revised for IEEE Transactions on Industrial Electronics.

[J4] **S. Zhao**, F. Lin, K. Peng, B. M. Chen, and T. H. Lee, "Finite-time stabilization of cyclic formations using bearing-only measurements," *International Journal of Control*, vol. 87, no. 4, pp. 715-727, April 2014.

[J5] **S. Zhao**, B. M. Chen, and T. H. Lee, "Optimal deployment of mobile sensors for target tracking in 2D and 3D spaces," *Acta Automatica Sinica*, vol. 1, no. 1, pp. 50–56, February 2014.

[J6] **S. Zhao**, F. Lin, K. Peng, B. M. Chen, and T. H. Lee, "Distributed control of angle-constrained cyclic formations using bearing-only measurements," *Systems & Control Letters*, vol. 63, no. 1, pp. 12–24, January 2014.

[J7] **S. Zhao**, B. M. Chen, and T. H. Lee, "Optimal sensor placement for target localization and tracking in 2D and 3D," *International Journal of Control*,

vol. 86, no. 10, pp. 1687–1704, October 2013.

[J8] F. Lin, K. Z. Y. Ang, F. Wang, B. M. Chen, T. H. Lee, B. Yang, M. Dong, X. Dong, J. Cui, S. K. Phang, B. Wang, D. Luo, K. Peng, G. Cai, **S. Zhao**, M. Yin, and K. Li, "Development of an unmanned coaxial rotorcraft for the DARPA UAVForge challenge," *Unmanned Systems*, vol. 1, no. 2, pp. 247–258, September 2013.

Conference

[C1] F. Wang, P. Liu, **S. Zhao**, B. M. Chen, S. K. Phang, S. Lai, and T. H. Lee, "Guidance, navigation and control of an unmanned helicopter for automatic cargo transportation," submitted to *2014 Chinese Control Conference*.

[C2] **S. Zhao**, Z. Hu, M. Yin, K. Z. Y. Ang, P. Liu, F. Wang, X. Dong, F. Lin, B. M. Chen, and T. H. Lee, "A robust vision system for a UAV transporting cargoes between moving platforms," submitted to *2014 Chinese Control Conference*.

[C3] K. Peng, **S. Zhao**, F. Lin, and B. M. Chen, "Vision-based target tracking/following and estimation of target motion," in *Proceedings of the 2013 AIAA Guidance, Navigation and Control Conference*, (Boston, USA), August 2013.

[C4] **S. Zhao**, X. Dong, J. Cui, Z. Y. Ang, F. Lin, K. Peng, B. M. Chen, and T. H. Lee, "Design and implementation of homography-based vision-aided inertial navigation of UAVs," in *Proceedings of the 2013 Chinese Control Conference*, (Xian, China), pp. 5101–5106, July 2013.

[C5] **S. Zhao**, F. Lin, K. Peng, B. M. Chen, and T. H. Lee, "Distributed control of angle-constrained circular formations using bearing-only measurements," in *Proceedings of the 2013 Asian Control Conference*, (Istanbul, Turkey), pp. 1–6, June 2013.

[C6] **S. Zhao**, F. Lin, K. Peng, B. M. Chen, and T. H. Lee, "Finite-time stabilization of circular formations using bearing-only measurements," in *Pro-*

ceedings of the 2013 IEEE International Conference on Control & Automation, (Hangzhou, China), pp. 840–845, June 2013.

[C7] K. Peng, **S. Zhao**, F. Lin, and B. M. Chen, "Vision based stabilization for aircraft in unknown environment without GPS signal," in *Proceedings of the 2012 AIAA Guidance, Navigation and Control Conference*, (Minneapolis, USA), August 2012.

[C8] **S. Zhao**, F. Lin, K. Peng, B. M. Chen, and T. H. Lee, "Homography-based vision-aided inertial navigation of UAVs in unknown environments," in *Proceedings of the 2012 AIAA Guidance,Navigation and Control Conference*, (Minneapolis, USA), August 2012.

[C9] **S. Zhao**, B. M. Chen, and T. H. Lee, "Optimal placement of bearing-only sensors for target localization," in *Proceedings of the 2012 American Control Conference*, (Montreal, Canada), pp. 5108–5113, June 2012.