UNIVERSITY OF TECHNOLOGY SYDNEY
Centre for Autonomous Systems, Faculty of Engineering and Information Technology

Autonomous Navigation and Planning Technology for Quad-rotors Unmanned Aerial Vehicle (UAV) System

by

Yongbo Chen

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2021

Certificate of Authorship/Originality

I certify that the work in this thesis has not been previously submitted for a degree nor has it been submitted as a part of the requirements for other degree except as fully acknowledged within the text.

I also certify that this thesis has been written by me. Any help that I have received in my research and in the preparation of the thesis itself has been fully acknowledged. In addition, I certify that all information sources and literature used are quoted in the thesis.

Yongbo Chen

April 30, 2021

Production Note:
Signature removed prior to publication.

ABSTRACT

Autonomous Navigation and Planning Technology for Quad-rotors Unmanned Aerial Vehicle (UAV) System

by

Yongbo Chen

In an unknown environment, a robot needs to keep estimating its pose and, simultaneously, building a map of its surrounding environment using only on-board sensors. This problem is called as simultaneous localization and mapping (SLAM), which is one of the key robotics problems that have been studied in the past decades. Meanwhile, many related research problems, including active SLAM, semantic SLAM and so on, are studied to further extend the applications of SLAM. This thesis aims to investigate the graph structure of SLAM and applies it in the related problems, including anchor selection and active SLAM, with the applications for Quad-rotors UAV system. The thesis is composed of three parts:

First, we explore the relation between the graphical structure of 2D and 3D pose-graph SLAM and Fisher information matrix (FIM), Cramér-Rao lower bounds (CRLB), and its optimal design metrics (T-/D-optimality). Based on the assumption of isotropic Langevin noise for rotation and block-isotropic Gaussian noise for translation, the FIM and CRLB are derived and shown to be closely related to the graph structure, in particular, the weighted Laplacian matrix. We also prove that the total node degrees and the weighted number of spanning trees, as two graph connectivity metrics, are closely related to the trace and determinant of FIM, respectively. We also present upper and lower bounds for the D-optimality metric, which can be efficiently computed and are almost independent of the estimation results. The proposed conclusions are verified with several well-known datasets.

Second, we consider 2D/3D pose-graph SLAM problem when accurate ground

truth for some poses, termed anchors, can be obtained. We present a high-efficient algorithm for the problem of choosing a set of anchored poses from a set of possible or potential poses, that minimizes estimated error in pose-graph SLAM. Using the tree-connectivity, the anchor selection problem is re-formulated as a sub-matrix selection problem for reduced weighted Laplacian matrix and belongs to maximization problem of a sub-modular function with a cardinality-fixed constraint. Two improved greedy methods, using Cholesky decomposition, approximate minimum degree permutation (AMDP), order re-use, and rank-1 update technologies, are presented to solve this problem with a performance guarantee between the chosen subset and the optimal solution. Simulations with public-domain datasets and real-world experiments are presented to demonstrate the efficiency of the proposed techniques.

Third, as an application of the graph structure results, based on map joining, two active SLAM methods with two different frameworks are presented: one for 2D feature-based SLAM and the other one for 3D pose-graph SLAM. For the 2D feature-based SLAM, we present a detached method based on model predictive control (MPC) framework. For the uncertainty minimization problem, a non-convex constrained least-squares problem is presented to approximate the original problem using graph topology. Using convex relaxation, it is further transformed into a convex problem, and then solved by a convex optimization method and a rounding procedure based on the singular value decomposition (SVD). For the area coverage problem, it is solved by the sequential quadratic programming (SQP) method. For the 3D pose-graph active SLAM problem, weighted node degree (T-optimality metric) and weighted tree-connectivity (D-optimality metric) are introduced to choose a candidate trajectory and several key poses. With the help of the key poses, a sampling-based path planning method and a continuous-time trajectory optimization method are combined hierarchically. In simulations and experiments, we validate these two approaches by comparing against existing methods, and we demonstrate the off/on-line planning part using a quad-rotor unmanned aerial vehicle (UAV).

Acknowledgements

In past two years, as a dual PhD student both at University of Technology Sydney (UTS) and Beijing Institute of Technology (BIT), I have gone through a mysterious research journey, filled with countless meaningful moments of research failure and triumph. When I am close to the finish line, I really realize that I cannot make this advance without the help of my surrounding people and my two universities UTS and BIT.

At first, I would like to thank my main supervisor A/Prof. Shoudong Huang. He guides me into the robotics research area and significantly helps me to complete my research. Without his help, I am almost impossible to finish this thesis. His vision, wisdom, and guidance along the way have helped me to go ahead on the right direction. Thank you for asking the right questions at the right time, for your enduring trust, patience, unconditional, and great support.

I want to thank Prof. Gamini Dissanayake, my co-supervisor Dr. Zhao, and Prof. Robert Fetch. In my research process, they spend much precise time to help the revision of my submitted papers. Many useful comments are given to revise my papers. Their vision, wisdom, and guidance along the way have helped me to improve my research level.

I also want to thank my supervisor at BIT, Prof. Jinqiao Yu. My life is changed because of his broad-mind and forethought. He gives me the opportunity to apply the China Scholarship Council (CSC) scholarship and join Centre for Autonomous Systems (CAS), UTS as a dual-PhD student. His support opens a gate for me, and then I find the direction of my own career. At the beginning of my PhD studying, he also offers many academic suggestions to me, which speeds up my steps in the academic area. He is my unexpected helper.

After four magnificent years, I will explore my own research areas in the coming days. Special thanks to my amazing friend, Jun Wang, with whom I have shared two years of unforgettable memories. Thanks to my wonderful friends and labmates in CAS. In particular, thanks to Kasra Khosoussi, Teng Zhang, Liyang Liu, Fang Bai, Jingwei Song, Yanghao Zhang, Jiaheng Zhao, Tianming Wang, Huan Yu, Wenjie Lu, Chanyeol Yoo, Karthick Thiyagarajan, Mahdi Hassan, Tomoyuki Shiozaki, Cedric Le Gentil, Prof. Dikai Liu, A/Prof. Sarath Kodagoda, A/Prof. Jaime Valls Miro, Dr. Teresa Vidal-Calleja, Dr. Alen Alempijevic, and many others.

As a dual-PhD student, I also want to thank my labmates in my chinese cooperative network formulation lab (CNFL), at BIT, including: Guanchen Luo, Qibo Deng, Shengqing Yang, Yafei Wang, Nuo Xu, Linlin Wang, Siyu Zhang, Xiaolong Su, Huchao jiang, Yuanchuang Shen, Fangzheng Chen, Xiaolin Ai, Weidong Liang, Di Yang, and many others.

I would like to express my gratitude to UTS, BIT and CSC for offering the great chance to take part in the international PhD visiting program. By their helps, I can come to Australia to improve my research skills. I have learned and achieved a lot during this wonderful period.

Finally, I want to thank my relatives: my grandparents, my parents, my parents in law, my wife, my little brother, and my son. In this journey, I have to stay apart from you, which is the most difficult part for me. Thank you very much for your support. Without your support, it is really hard to image that I can finish this thesis.

Yongbo Chen

Sydney, Australia, 2021.

Contents

List of Figures

Chapter 1

Introduction

With tremendous developments in multiple sciences and technologies, such as mechanism, sensors, computer sciences, optimization methods and so on, many kinds of quad-rotors UAVs, which are also called Air Robot, are developed to apply in a wide variety of unknown environments, such as indoor environment, forest environment and signal shielding situation. The autonomous navigation technology offers the accurate pose estimation and a consistent representation of its surrounding map. This problem is named as simultaneous localization and mapping (SLAM). In SLAM both the trajectory of the platform and the location of all landmarks in surrounding map are estimated on-line without the need for any a priori knowledge [42]. Research on SLAM has more than 30 years history, if considering the early years, when the formulation of the SLAM problem is not agreed on by all scholars in robotics area. So far, the SLAM problem has become one of the most popular research areas in mobile robotics. As the basis of the perception and planning technologies, the SLAM method becomes more significant for the quad-rotors in an indoor environment, compared with the other technologies, like structure design and control.

Meanwhile, multiple derivate SLAM problems, especially active SLAM, are booming. Because of the limited information contained in each obtained measurement, active SLAM problem, combined the SLAM and path planning capabilities, is a decision making problem where a robot's trajectory is chosen to improve both mapping and localization results, and at the same time, to perform other tasks such as coverage or exploration. As a normally large-scale decision-making problem, it is

considered as one of the most challenging problems of mobile vehicles.

This thesis focuses on the intrinsic structure of the SLAM problem and its applications in related planning technologies for mobile robots especially for UAVs.

1.1 Motivation

Even though the research on SLAM has such a long history, there still exists some basic structure questions behind the well-known SLAM problem. One of the key questions is why our solvers may fail miserably in many challenging scenarios and even some simpler cases? The intuitive answers for the question are too restrictive assumptions and their intrinsic influence of the SLAM structure. Another key question is that, based on current SLAM understanding, can we find a way to improve its performance? The additional prior knowledge introduction, like anchor selection, and the active SLAM are two powerful directions. Concerned on both two questions, this thesis aims to find some structure understandings about the SLAM problem.

This thesis is comprised of three main parts, and their motivations are respectively:

1. It is well known that SLAM can be represented as a graph, whose nodes are poses and landmarks, and their edges represent the relative measurements. As a graph network, its topology directly impacts the estimation quality. A "well-connected" graph will obviously lead to better SLAM result, but the following questions are how we can clarify the meaning of the "well-connected" graph, and identify which graph is "well-connected". If we can clearly define the meaning of "well-connected", we might apply it as a powerful tool to improve the performance in multiple assignment and planning tasks (e.g. anchor selection and active SLAM) related to SLAM. Inspired by the two-dimension (2D) results presented in [78], the

first part of this thesis aims to extend them into three-dimension (3D) case using Lie group representation and specific noise assumption.

2. The standard SLAM problems assume no prior knowledge on robot poses. The ground truth information is typically included in the first pose and is spread to other poses based on pairwise measurements. It is now well known that in the best achievable SLAM result, the estimate error for the pose that is furthest away from the origin can be very large, depending on the sensor range, sensor accuracy, and the graph connectivity. Clearly, if the true locations of some poses in the graph are available, the accuracy of the estimates can be significantly improved. Practically, this may be achieved in a number of ways. In indoor environments, it may be possible to stop the robot at previously defined, known locations. Outdoor, it may be possible to pause for a sufficient length of time to obtain an accurate Global Positioning System (GPS) fix. In comparison, 2D/3D pose-graph SLAM with multiple anchors* can obtain much better results than with only one anchor. Thus it might be necessary to use multiple anchors to improve the accuracy in SLAM especially when a highly accurate map is required. The second part of this thesis aims to formulate a solvable anchor selection, find a high-efficient and performance-guarantee solving algorithm and present several potential applications.

3. The growing number of SLAM methods work on the challenging scenarios using limited sensors and platform. It is easy to understand that it is really hard to obtain an accurate SLAM result under some unreasonable robot actions, which leads to some unreliable measurement networks. Intuitively speaking, the revisit of the previous trajectory may lead to more measurements benefit for the connected graph and estimated results. The more, the better. However, the end-less revisit trajectory in a small area may break the sparsity of the SLAM problem and against

*anchor — Here an anchor means a pose with its global location known. Also different from sub-map joining method [64, 146], the anchored poses considered in this paper have their global information, which brings more information to the whole SLAM system.

one of the key purpose of robot: exploration. Therefore, the active SLAM performs a decision making problem in order to find a balance between visiting new places (exploration) and reducing the uncertainty by re-visiting known areas (exploitation) [17]. As the third part of this thesis, we present two different active SLAM methods using graph topology obtained by the first two parts.

1.2 Main contributions

The main contributions of this thesis are summarized below:

i. ***Chapter 3*** :

- Derivation of the formula of the FIM and the CRLB of the synchronization on the group of rigid body motions, $\mathbb{R}^2 \times SO(2)$ and $\mathbb{R}^3 \times SO(3)$, based on the assumption of isotropic Lagevin noise (for rotation) and block-isotropic Guassian noise (for translation);
- Derivation of the relationship between the FIM of 3D pose-graph SLAM and the weighted Laplacian matrix;
- Extension of the analysis results of the D-optimality metric in [78] and [80] from 2D pose-graph SLAM into 3D case with $\mathbb{R}^3 \times SO(3)$ relative-pose measurements;
- Comparison of the D-optimality and T-optimality metrics of 2D and 3D pose-graph SLAM from the graphical perspective.

ii. ***Chapter 4*** :

- The approximation from the D-optimality metric based anchor selection problem to the sub-matrix selection of the weighted Lapalcian matrix, which greatly reduces its computational complexity.

- The new sub-matrix selection is proved to be a non-negative non-normalized non-monotone sub-modular optimization problem, and the greedy-based method is used.
- The performance bounds, based on the sub-modularity of the sub-matrix selection problem, about the obtained solution and the optimal solution, are derived.
- The high-efficiency method using lazy evaluation, Cholesky decomposition, AMDP, order re-use and rank-1 update technologies to greatly speed up the anchor selection process, is developed.

iii. ***Chapter 5*** :

- The application of submap joining (Linear SLAM) in active SLAM for improving the computational efficiency.
- The simplification of the original objective function using the incident matrices of the 2D feature-based SLAM graph as well as the strong positive correlation between the weighted number of spanning trees and the *D-opt* criterion.
- The application of a convex method based on spectrahedral description, half space representation, and a singular value decomposition rounding procedure.
- A theorem showing the relationship of the eigenvalues between the information matrices of the submap and joining global map.

iv. ***Chapter 6*** :

- Hierarchical evaluation with a fast but general-performance method using weighted node degree for all candidate paths, and a slow but fruitful method using weighted tree-connectivity for a selected set of paths.

- Application of sub-map planning and estimation, to improve computational efficiency.
- Application of real-time sampling-based planning and continuous-time optimization in active SLAM.
- Introduction of key poses to build a hierarchical framework, including the RRT-connect method and continuous-time trajectory planning method.

1.3 Thesis organization

This thesis is organized as follows:

- ***Chapter 2***: In this chapter, some significant contributions in SLAM and active SLAM areas are briefly reviewed.
- ***Chapter 3***: In this chapter, the intrinsic property of 2D/3D pose-graph SLAM is connected with the graph topology of the measurement network based on D/T optimal design metric of the FIM and CRLB.
- ***Chapter 4***: In this chapter, anchor selection problem is formulated as a sub-modular optimization problem and solved by an efficient greedy-based method.
- ***Chapter 5 and 6***: In these chapters, two different active SLAM frameworks are developed based on graph topology and sub-map joining.
- ***Chapter 7***: In this chapter, the conclusions of all chapters are summarized and presented.

1.4 Publication

Conference Papers

1) Yongbo Chen, Shoudong Huang, Robert Fitch, Liang Zhao, Huan Yu and Di Yang. "On-line 3D active pose-graph SLAM based on key poses using graph topology and sub-maps." IEEE International Conference on Robotics and Automation (ICRA), ICRA 2019, pp. 169-175.

2) Yongbo Chen, Shoudong Huang, Robert Fitch, and Jianqiao Yu. "Efficient active SLAM based on submap joining, graph topology and convex optimization." IEEE International Conference on Robotics and Automation (ICRA), ICRA 2018, pp. 5159-5166.

3) Yongbo Chen, Shoudong Huang, Robert Fitch, and Jianqiao Yu. "Efficient active SLAM based on submap joining." Australasian Conference on Robotics and Automation, ACRA. 2017.

Journal papers

1) Yongbo Chen, Liang Zhao, Yanghao Zhang, Shoudong Huang, and Gamini Dissanayake. "Anchor selection for pose-graph SLAM with multiple anchors based on graph topology and sub-modular optimization." IEEE Transactions on Robotics, Accepted.

2) Yongbo Chen, Shoudong Huang, Liang Zhao, and Gamini Dissanayake. "Cramér-Rao bounds and optimal design metrics for pose-graph SLAM." IEEE Transactions on Robotics, 37(2), 627-641, 2021.

3) Yongbo Chen, Shoudong Huang and Robert Fitch. "Active SLAM for mobile robots with area coverage and obstacle avoidance." IEEE/ASME Transactions on Mechatronics, 25(3), 1182-1192, 2020.

Other publications (not closely-related to this thesis)

1) Yongbo Chen, Jianqiao Yu, Xiaolong Su and Guanchen Luo. “Path planning for multi-UAV formation.” Journal of Intelligent & Robotic Systems, 77(1), 229-246, 2015.

2) Yongbo Chen, Guanchen Luo, Yuesong Mei, Jianqiao Yu, and Xiaolong Su. “UAV path planning using artificial potential field method updated by optimal control theory.” International Journal of Systems Science, 47(6), 1407-1420, 2016.

3) Yongbo Chen, Jianqiao Yu, Yuesong Mei, Yafei Wang and Xiaolong Su. “Modified central force optimization (MCFO) algorithm for 3D UAV path planning.” Neurocomputing, 171, 878-888, 2016.

4) Yongbo Chen, Jianqiao Yu, Yuesong Mei, Siyu Zhang, Xiaolin Ai and Zhengyue Jia. “Trajectory optimization of multiple quad-rotor UAVs in collaborative assembling task.” Chinese Journal of Aeronautics, 29(1), 184-201, 2016.

5) Yongbo Chen, Yuesong Mei, Jianqiao Yu, Xiaolong Su and Nuo Xu. “Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm.” Neurocomputing, 266, 445-457, 2017.

6) Yongbo Chen, Zhenyue Jia, Xiaolin Ai, Di Yang and Jianqiao Yu. “A modified two-part wolf pack search algorithm for the multiple traveling salesmen problem.” Applied Soft Computing, 61(1), 714-725, 2017.

7) Yongbo Chen, Di Yang and Jianqiao Yu. “Multi-UAV task assignment with parameter and time-Sensitive uncertainties using modified two-Part wolf pack search algorithm.” IEEE Transactions on Aerospace and Electronic Systems, 54(6), 2853-2872, 2018.

8) Yongbo Chen, Brenton Leighton, Huishen Zhu, Xijun Ke, Songtao Liu, and Liang Zhao. “Submap-based indoor navigation system for the Fetch robot.”

IEEE access, 8, 81479-81491, 2020.

9) Yongbo Chen, Liang Zhao, Ki Myung Brian Lee, Chanyeol Yoo, Shoudong Huang, and Fitch, Robert. "Broadcast your weaknesses: cooperative active pose-graph SLAM for multiple robots." IEEE Robotics and Automation Letters, 5(2), 2200–2207, 2020.

10) Yongbo Chen, Liang Zhao, Yanghao Zhang, Shoudong Huang, "Dense isometric non-rigid shape-from-motion based on graph optimization and edge selection." IEEE Robotics and Automation Letters, 5(4), 5889-5896, 2020.

10) Yongbo Chen, Yanghao Zhang, Shaifali Parashar, Liang Zhao, Shoudong Huang, "Conformal Non-Rigid Structure-from-Motion using Differential Geometry and Separable Structure", submitted to IEEE/CVF International Conference on Computer Vision 2021.

Chapter 2

Literature Survey on SLAM and Active SLAM

The Simultaneous Localization and Mapping (SLAM) problem asks if it is possible for a mobile robot, such as a quad-rotor UAV, being placed in an unknown environment with an unknown location for the robot to be able to build a map of the surrounding environment in real-time and simultaneously work out its own poses, using on-board sensors [126]. As mentioned previously, the journey of SLAM research, which has been lasted for more than 30 years, began with the search for rigorous estimation-theoretic formulation. Now, it has been developed into a large and complicated system with several well-known question formulations, a number of efficient solvers, many successful applications in a wide range of scenarios and a group of open source software communities. Many scholars make great contributions to this developing and growing research area. In this chapter, we briefly review the history of SLAM and highlight some of the most outstanding contributions. Currently, the process of solving a SLAM problem can be divided into SLAM front-end and SLAM back-end. Our survey focuses on SLAM back-end, which means to apply estimation or optimization techniques to solve the SLAM problems. Of course, SLAM front-end is becoming more and more popular. For readers who are interesting in SLAM front-end, please refer to famous surveys [6, 17].

In addition to the SLAM survey, we also make a short review of the development of active SLAM research area. The active SLAM is to choose good trajectories for improving the SLAM result and performing the given tasks, such as coverage. It is a derivate decision-making problem from SLAM area, because of the performance

limitation of the existing SLAM algorithms. It presents a well-known dilemma for the activity of the robot in an unknown space [17]; it is difficult to strike a balance between visiting new places and revisiting known areas to reduce map uncertainty. The active SLAM method has more than 15 years history. As an increasingly popular topic, there has been many scholars making much important contributions. As a brief literature survey, we also pick out the key contributions in the active SLAM area.

2.1 SLAM from filter to optimization

2.1.1 Filter-based SLAM

As a statistical estimation problem, the first group of the valid SLAM methods is filter-based SLAM. Based on the filter framework, the SLAM method needs to estimate spatial quantities (e.g., robot pose, landmark position) from noisy sensor measurements and imperfect models [77]. The early estimation-theoretic framework and the way to deal with the spatial and geometric uncertainties are built by Smith and Cheeseman et al. [122] and Durrant-Whyte et al. [43]. The research about a degree of convergence between the Kalman-fllter (KF) based SLAM methods and the probabilistic localization and mapping methods is introduced by Thrun et al. [124]. They are the bases of the first widely-used generation of SLAM solution EKF-SLAM [40] introduced by Dissanayake et al.. Compared with the previous work, this work has a deeper understand on the convergence and steady-state behavior of Kalman filter-based SLAM [77]. At the same time, this paper also formulates a systemic and complete SLAM implement, including map management, data association, and landmark initialization. EKF-SLAM can be seen as the first break-though in SLAM area, which provides the first complete SLAM back-end for the early 2D laser sensors. Meanwhile, EKF-SLAM supports the first real-time visual monocular SLAM system [38], which is a milestone to the real application of the monocular camera in

SLAM.

However, EKF-SLAM has two majors drawbacks: (1) computational cost and (2) inconsistency. In typical EKF-SLAM, the covariance matrix of the estimated poses and landmarks are saved and updated in every step. With the growing of the number of the observed landmarks, the quadratic time and space complexity of EKF-SLAM will become un-acceptable for real-time requirement. Without the help of the other approximated technologies, like sub-map joining, EKF-SLAM can only be utilized on-line in small-scale SLAM with limited number of landmarks. The other drawback is the inconsistency, which is caused by linearization of the nonlinear measurement and motion models. The estimation error is larger than actual error usually and greatly depends on the initial covariance assumption. The first scholars to state this phenomenon are Julier and Uhlmann et al. [70]. In the later work, this question is further investigated empirically and theoretically in [58] and [56]. In [58], Huang and Dissanayake et al. analysis the special situation with one-step motion to show that the inconsistency is caused by the fact that the filter Jacobians at different time instants are evaluated using different estimates for the same state variables. As an extension, Huang et al. explore this fact in the general case and analyses the inconsistency from the perspective of observability [56].

Significant progresses have been made to deal with these two drawbacks, especially the inconsistency problem, which has been completely solved by the invariant-EKF (IEKF) SLAM. IEKF SLAM method is first presented by Barrau and Bonnabel et al. [8], and then its convergence and consistency analysis are discussed by Zhang and Huang et al. [143]. IEKF aims to formulate the landmarks and poses in a consistent manifold to achieve the improved consistency. Its performance is comparable with the current optimization based SLAM in the perception of the estimated error [144].

For the computational cost problem, another family of on-line SLAM algorithms, called extended information filter (EIF) SLAM [127], can partly solve this problem because of the sparsity of the Fisher information matrix (FIM). The EIF is the dual of EFK because of the canonical representation of the covariance matrix and FIM. Even though the EIF has the better ability in the memory cost, its computational complexity is higher than EKF SLAM with a cubic complexity, because of the equation solving process of the information vector and FIM. Beside the duality between EIF and EKF, EIF SLAM also shows that the sparsity is a significant intrinsic property of the SLAM problem. With the tremendous development of the sparse linear algebra researches, this point begins to play an important role in the second generation of SLAM solutions: optimization-based SLAM.

Because of the prediction step of the EIF SLAM, its FIM is not exactly sparse. The author calls it approximately sparse, which makes the computational complexity of the EIF SLAM high [127]. If we can delete the motion model of the robot, the FIM will be exactly sparse leading much lower computational complexity, which is similar to one iteration update of the optimization-based SLAM. Inspired by this idea, Wang et al. present D-SLAM algorithm which is a decoupled solution framework to SLAM [138]. In this method, they parameter the SLAM problem using features instead of both features and poses using some non-linear transform functions between features and poses. The SLAM problem is decoupled into map updating and localization two parts using exactly sparse EIF for mapping and EKF, the kidnapped robot solution and the covariance intersection for localization. This method leads to a solution with only $O(n)$ computational complexity with accepted information loss.

Another important family of on-line filter-based SLAM algorithms is the particle filter based SLAM algorithms. The most famous particle filter based SLAM is the FastSLAM algorithm, introduced by Montemerlo [103]. It is influenced by earlier

probabilistic mapping experiments of Murphy et al. [107] and Thrun et al. [125]. Besides the better real-time ability compared with EKF SLAM, one of the main advantages of the FastSLAM algorithm is that it can deal with the SLAM problem with the non-linear process model and non-Gaussian pose distribution [42]. It uses the Rao-Blackwellisation to make the map landmarks become independent, which greatly reduce the computational complexity of the particle filters. Instead of a joint map covariance with quadratic complexity, in the FastSLAM algorithm, the map is represented as a set of independent distributions*. Nevertheless, FastSLAM 1.0/2.0, especially FastSLAM 1.0, likes any other particle filtering algorithm, has two inevitable drawbacks, namely the filter degeneracy and sample impoverishment problem. For the sample impoverishment problem, in FastSLAM 2.0, Montemerlo and Thrun et al. partly solve this problem by sampling from the optimal proposal distribution [102]. Before the current optimization-based SLAM method, the FastSLAM algorithm is one of the most efficient algorithms.

2.1.2 Optimization-based SLAM

In fact, in very early years, the optimization-based SLAM [98] has been initially formulated and solved by a nonlinear least squares techniques, whose global minimizer corresponds to the maximum likelihood estimation under the assumption of Gaussian noise. However, compared with the filter-based SLAM, this formulation is not considered by most scholars, because of its drawback in computational complexity. Based on this work, Gutmann and Konolige et al. extend its formulation into an incremental way and present a more reliable loop-closure detection [53]. The other important point in their work is that they suggest to utilize the sparsity of the SLAM structure.

Now the other name of the optimization-based SLAM is called graph-optimization

*Note, FastSLAM still linearizes the observation model.

SLAM. This name partly originates from Folkesson and Christensen's work [48]. As the understanding and technologies deepens of the sparsity and graph structure of the SLAM problem, the optimization-based SLAM becomes computational acceptable. Based on sparse QR and Cholesky decomposition of the Jacobian and information matrix, Dellaert and Kaess et al. [39] present an important sparse SLAM solver: Square Root SAM. In this work, they also emphasize employing a good fill-reducing variable ordering. Based on this work, they further develop the classical second SLAM solvers: incremental smoothing and mapping (iSAM) [75] and iSAM2 [74]. In iSAM, they apply the incremental sparse QR factorization method based on the givens rotations. Further, in iSAM2, they introduce the Bayes tree data structure into the incremental update process of variable reordering and re-linearilzation. The covariance recovery using several similar technologies is presented in their following work [73]. The important point of iSAM1 and iSAM2 is the incremental idea in solving SLAM, which is very suitable for the most widely used SLAM application, navigation.

The initial highly efficient optimization SLAM method is based on the Gaussian Newton (GN) method. Because of the large-scale of the SLAM problem and some special singularity rotation representations, the Jaccobin matrices in SLAM may get closed to the non-full rank, which leads to the singularity problem for the GN method. Hence, the Levenberg-Marquardt (LM) and Powell's dogleg method are introduced in SLAM problem and similarly bundle adjustment problem [97, 118]. As an example, Parallel Tracking and Mapping (PTAM) [83] is the first visual SLAM system which implemented bundle adjustment method in real-time visual SLAM system. Benefit from the development of the optimization method and the deepen understanding of the graph structure of the SLAM, multiple standard tool-packs have been set up, such as: g2o [90], GTSAM, Ceres [2] and SLAM++ [66]. All these tool-packs are very efficient which can deal with more than 1000 nodes

problem in just few seconds. They are widely used in current complete SLAM system like cartographer [55], ORB-SLAM [105], ORB-SLAM2 [106] and so on. Currently, even though the gradient-based SLAM has become more and more mature, there is an obvious drawback for this kind of methods: the global optimality. An easy and intuitive question is whether exists a method that can guarantee the globally optimal solution. This point is the current direction in SLAM back-end. The partial answer is Yes in some special situations.

The early thinking about this point can be traced back to about 10 years ago. Huang et al. aim to explore the relationship between the SLAM and a linear least squares problem [60]. The linear least squares problem leads to the global solution and closed form of the estimation problem. They find an interesting phenomenon: In some SLAM problems with spherical (isotropic) noise covariance matrices, the GN-based SLAM has surprisingly good convergence results from very poor initial values. In their later work, they deeply explore this point at the one-step SLAM [61, 134, 135] and multiple step SLAM [136] with spherical covariance matrices. In these work, they prove that the simplified one-step problems have at most two minims, one of which appears only when the data are extremely noisy. Based on this point, they also develop a dimensionality-reduction scheme by exploiting the partially-linear structure of standard measurement models [77]. For the convergence of the GN-based SLAM, Carlone et al. show some parameters in the motion and sensor models will affect the convergence of the solution [18].

Thanks to these explorations in the global optimality of the SLAM, the convex relaxation methods appear. The first one aims to use the convex relaxation method to solve the SLAM problem is the semidefinite programming (SDP) relaxation for the 2D pose using Euler angle, presented by Liu and Huang et al.[95]. They formulate the SLAM problem as a quadratically constrained quadratic program (QCQP) which admits a straightforward SDP convex relaxation. They mapped the solution of the

convex program into a feasible suboptimal estimation for SLAM. Commonly, the solution of the relaxation result is just an approximate solution, so they use that estimation to initialize conventional nonlinear least squares solvers. Then, inspired by this work, David and Charles use the convex relaxation of the Lie group to solve both 2D and 3D problem with a uniform formulation [117]. The result of this method is comparable with other gradient-based SLAM. In the latest references, they further present a convex optimization method, using the decoupled structure of the SLAM problem and manifold method, called SE-Sync [116]. Based on this work, Briales and Gonzalez et al. further propose a similar and more-efficient method, named Cartan-sync [11]. For these convex-relaxation SLAM methods, their computational complexity is comparable with the GN-based SLAM.

Meanwhile, a performance verification technology is involved in this kind of relaxed method. It is well known that normal SLAM is a non-convex optimization problem, whose duality gap is not zero. We can get the optimal solution of the relaxed method because of its convexity. If the objective function of the obtained SLAM solution gets closed to the one of relaxed optimal solution, it means that the obtained solution is almost the global optimal solution of the SLAM problem. Based on this point, a verification technology is pointed out by Carlone et al.. They first apply the duality theory to verify the 2D SLAM using Euler angle [20]. Then, They further extend it into 2D and 3D case using Lagrangian dual problem and rotation matrix representation [22]. Finally, they summarize and systematize all used methods and technologies in their later work [19]. It seems that the duality gap can almost reach zero, for some real-world datasets. In [137], Wang and Huang also obtain the similar conclusion based on the exploration in two objective functions leads by different noise assumptions.

Besides these unconstrained formulation, there exists a constrained formulation by the introduction of the loop constraints based on measurement network, presented

by Bai et al. [3]. The benefit of this framework is that it can identify whether the obtained measurements is relatively correct based on χ^2 difference test [4] and make the result more robust.

In short, now, the optimization based SLAM is the most commonly used formulation in SLAM. Compared with the filter based SLAM, it does not exist the consistency problem and has a faster solving speed based on a large number of researches on SLAM.

2.1.3 Map joining and initialization

As a supplement of the previous SLAM survey, in this small section, we further introduce the sub-map joining methods and initialization technologies in SLAM area.

The sub-map joining idea can be used in both filter-based or optimization based SLAM methods. SLAM methods using multiple sub-maps can limit the number of landmarks in any single local map, which help to build each local map takes $O(1)$ time. Then if the global map is required, the local maps can be fused together with some approximations and assumptions. In the early years, when EKF and EIF SLAM is the main tool to solve the SLAM problem, sub-map joining method is presented to improve the running-time ability and the local consistency [141, 45, 62]. By the help of sub-maps, the local consistency of the SLAM method can be improved by preventing large global uncertainties. By starting a new local map, it can limit the uncertainty in any local map smaller than a certain threshold [111].

Now sub-map idea is more common used in the optimization based SLAM in some mature tools, like cartographer [55]. Within each local map, it can also marginalize out some of the variables before joining the local maps together (e.g. only keep the features [63]) [57]. The application of the sub-maps helps to reduce the dimension of the state vector of the map joining problem, which is significantly lower than that

of the global SLAM problem. By this way, the accessibility of the good real-time ability is easy to be understand. Very recently, the state-of-art sub-map joining method, called Linear SLAM [145, 146] presents a framework that a sequence of local maps can be joined together either sequentially or in a more efficient divide and conquer manner by only performing nonlinear coordinate transformations to the common pose and solving fusion process using linear least squares. Linear SLAM is high-efficiency with an acceptable information loss in the linearizion of the nonlinear coordinate transformations.

Except the sub-map SLAM, the next important technology is the initialization technology for the optimization based SLAM. Solving the nonlinear least squares problem in SLAM usually needs an initialization scheme to provide an initial guess. The way to obtain it needs to be fast, reliable and had better to get closed to the globally optimal solution. A good initialization method can help to reduce the iteration of the optimization method. Meanwhile, for a nonlinear optimization problem, a good initial value helps to converge to the globally optimal solution.

The simplest initialization way is to directly use the odometry information and one of the observations to compute the initial value of all the robot poses and the features. However, this way is not very reliable and robust especially in the large-scale problems with many outliers. This initial value may be far from accurate and useless, which usually leads to a local minimum.

In order to get a good initialization result, Konolige et al. get the initialization result by composing pairwise measurements along the shortest-path spanning tree rooted at the initial pose, which leads to a better initial value [84]. But for the large-scale problem, this method is still not very accurate. Because of the decoupled structure of the SLAM problem, Carlone and Tron et al. have proposed the strategy of initializing the rotations first, and then initializing the translation. This initial-

ization method is very robust for large-scale pose graph problem [23]. Later, Briales and Javier et al. apply the Lagrangian duality to perform the initialization of 3D pose graph optimization [12]. Very recently, two kinds of initialization methods are combined with the state-of-art technology SE-sync to enhance its running ability, which makes it even faster than other GN-based tool-pack, like GTSAM [108, 115].

In fact, one of the sub-map strategies is to utilize the results from the sub-maps to obtain a high-quality initial value for the global SLAM optimization [111]. So the sub-map results can be used as an initialization method.

Besides above technologies, a high-quality initial value could be obtained through incremental SLAM. In general, the SLAM method performs a navigation role in the robot moving process. It is very natural to compute the SLAM problem incrementally and apply the SLAM result from the previous step as an initial value for the later one. In iSAM and iSAM2, this point is used very efficiently [75, 74].

2.2 Active SLAM development

The name of active SLAM is first introduced by Leung and Huang et al. [93]. If involving its famous origination, called active perception [7], it has more than 30 years history. The initial motivation of this kind of active SLAM methods is that because the performance of the current SLAM method is limited, the better motion of robot may introduce the better measurement, which further gets the better estimated results. As presented in work [17], the active SLAM problem presents a well-known dilemma for the activity of the robot in an unknown space; it is difficult to strike a balance between visiting new places and revisiting known areas to reduce map uncertainty.

Different from traditional path planners [34], the active SLAM assumes uncertain robot motion and lack of the global environment knowledge. The estimation problem

needs to be solved during the planning process as an important part. Lacking the accurate SLAM results, the planning problem will have no basis. Even though the techniques for SLAM become more and more mature, the awful measurements may cause the large drift, even failure in SLAM. A good trajectory for SLAM is a powerful way to make sure that the platform gets measurements as good as it needs.

There are several well-known approaches for active SLAM, such as the Model Predictive Control (MPC) framework [59] and the partially observably Markov decision process (POMDP) formalism [71]. These methods all attempt to select the best future action from a set of alternatives. It entails three basic issues, (i) how to generate the available actions set, (ii) how to evaluate the effect of each candidate action, (iii) how to select the best action.

The ways to generate the available actions set can be easily divided into two parts, including the dynamic based methods and geometry based methods. As the name suggests, the future candidate trajectories in the dynamic based methods are obtained using the robot models or some potential information fields. The popular MPC framework [93], which belongs to control theoretic approaches, and the POMDP formalism [72], which in general is known to be computationally intractable, both belong to the dynamic based methods. Huang and Ngai et al. explore the possibility and necessity of multi-step trajectory planning in EKF SLAM [59]. Then, Leung and Huang et al. further apply the similar framework to the EIF SLAM [94]. For the POMDP-based methods, in order to lead tractable solutions, Bayesian optimization [100], efficient Gaussian beliefs propagation [110] and generalized belief space approach [68] are used in active SLAM based on some approximations.

The geometry based active SLAM methods are inspired by the path planning algorithms and combined with the information evaluation. It aims to decouple the path planning part with the selection part, which greatly reduces the computational

complexity. Kim and Eustice apply the A* method, which is a widely-used 2D grid-based path planning method, in the active visual SLAM using a underwater platform [81]. Similarly, as famous fast path planning methods, RRT* [132] and D* [101] are utilized to generate the candidate trajectory in active pose-graph SLAM. However, these geometry based methods are difficult to meet the dynamic constraints for the real system in a complex environment, which reduces their practicability. So the dynamic based methods are more popular in the existing literature.

Second, after obtaining the candidate actions, the evaluation indexes, which can predict the estimated error, are also very important. In active SLAM, the Theory of Optimal Experimental Design (TOED) [112], including A-opt, D-opt, E-opt and T-opt, for the information matrices of the estimated vector after executing the future actions is a common way to evaluate the actions in terms of estimation accuracy. In the early years, because of the low computational complexity and intuitive physical meaning, the A-opt metric, which is the trace function of the covariance matrix, is applied commonly in active SLAM [59]. However, after performing some explorations in comparison among all metrics [27], Carrillo and Reid et al. obtain an initial conclusion; the D-opt metric is more suitable for the fulfilling of the active SLAM than A-opt metric and works on a similar performance with entropy. In their later work [26], they analysis and further extend this conclusion by the theoretical way based on Euler angle representation. The conclusion refreshes into that, compared with the other metric, only the D-opt metric does not lose monotonicity in the exploration of the robot [26]. However, one key point is misunderstood in this process. Some later scholars show that the decision element of the monotonicity is representation of the rotation part of the SLAM rather than the choice of the metrics. Kim et al. show that when using the Lie group, the monotonicity of different metrics can be remained [82]. Very recently, a summarization work on this problem is presented by Rodríguez-Arévalo and Neira et al. [114]. In short, in most

situations using different representation, the D-opt metric, which is to maximize the determinant of the information matrix, is the most popular criterion to use.

The last issue, which is the selection of the best future actions, is a non-convex optimal control with multiple constraints, whose globally optimal solution is usually hard to obtain. In general, it is difficult to use the indirect methods to solve an optimal control problem whose objective function is the D-opt metric. So the direct methods are very common in active SLAM. Some researchers choose future waypoints from a small subset of locations, based on optimal design metrics, to reduce the size of the search space. This approach leads the problem into the discrete optimization domain. Even based on the small search area, selecting candidate actions based on these TOED metrics is also computationally expensive. Therefore, most of the current work in this area tries to reduce the computational complexity of the objective function of this problem. Based on the D-opt metric, Indelman et al. aim to search the best actions on a conservative sparse information space, which is an approximation of the metrics of covariance matrix [67]. By this way, it can limit the computational complexity involved with evaluating impact of a candidate action in $O(n)$, for an n-dimensional state. Then, Kopitkov and Indelman et al. develop a computationally efficient approach for decision making under uncertainty by applying the matrix determinant lemma and reusing calculation among all candidates, based on the D-opt metric [85, 86].

Based on the theoretical basis of the active SLAM method, Khosoussi and Huang et al. point out some intrinsic properties on the graph structure of the SLAM, which have a great potential applications in robotics area. They present an important low-bound of the D-opt metric of the FIM corresponding to the weighted number of the reduced weighted Laplacian matrix of the SLAM measurement network [59], also named tree-connectivity [80, 79]. The importance of this new approximated metric is only shown as the decrease of the computational cost, but also it illuminates a

key graph structure of the SLAM; the estimated result performance can be directly decided by the graph structure of the measurement network without solving it. This graph topology has been used widely used in active SLAM [30], loop closure detection [50], measurement selection [78] and pairwise matches among observations [46], and so on. In this thesis, we extend the results to 3D case.

2.3 Summary

With a long-time development, in the SLAM area, some significant questions have been answered, while many new and interesting questions have been raised, because of new applications, new platforms, and new computational tools. In many real-world applications, from self-driving cars to mobile devices, the SLAM technology plays an important role in the situations where global navigation systems are unavailable or do not provide sufficient accuracy.

Along with the development of SLAM, SLAM back-end develops and changes from filter based methods to the optimization methods. The purpose of the SLAM back-end changes from finding an available solution framework to the local-optimal framework, and further to the global-optimal and robust framework. The SLAM back-end becomes more and more mature. Of course, there are also many important future research topics, such as, failure detection, convergence properties, and so on. I believe in the nearby future many of them will have good answers, similar to the previous answered questions. The main limitations of the current SLAM back-end methods are the limited robustness ability and result accuracy. Focusing on these limitations, my research in this thesis aims to explore the graph topology in SLAM back-end algorithm and apply it in anchor selection problem to improve the result accuracy.

For the active SLAM, it is still in a very initial research stage. Most of the current work focuses on the existing methods from information theory, control method and

path planning method. More theoretical knowledge needs to be detected and used. It is almost an open problem for us. The main limitations of the current active SLAM methods are the high-computational complexity and complex application scenario. Focusing on these limitations, we aim to present some practical methods with low-computational complexity and wide application range.

Chapter 3

Cramér-Rao Bounds and Optimal Design Metrics for Pose-graph SLAM

As we saw in Chapter 2, the main challenges of the SLAM back-end are the limited robustness and the result accuracy. As a novel back-end algorithm, the convex optimization based method, like SE-Sync method [116], aims to the globally optimal solution of SLAM. Assuming we can get the 'optimal' solution, the result accuracy also needs to be determined based on the result evaluated metrics, such as FIM and covariance matrix. Without these tools, it is really hard to evaluate the confidence level and accuracy of obtained results and then further explore the uncertainty planning part. Hence, our goal in this chapter is to derive the mathematical formulation of the FIM, covariance matrix, and their important relations to Cramér-Rao lower bounds, and then reveal and analyze the impact of the topology of the graphical representation of SLAM on the estimation error of the maximum likelihood estimator, which will be used in the later chapters.

Notations Throughout this chapter, bold lowercase and uppercase letters are reserved for vectors and matrices, respectively. Sets are shown by uppercase letters. $\boldsymbol{S}_1 \succeq \boldsymbol{S}_2$ means $\boldsymbol{S}_1 - \boldsymbol{S}_2$ is positive semidefinite. The Kronecker product is denoted by $\otimes$. trace($\star$) and det($\star$) represent the trace and determinant value of the matrix $\star$. We denote by diag($\boldsymbol{M}_1, ..., \boldsymbol{M}_k$) the block-diagonal matrix with matrices $\boldsymbol{M}_1, ..., \boldsymbol{M}_k$ as blocks on its main diagonal. The squared Frobenius norm is $\| \star \|_F^2 = \text{trace}(\star^\top \cdot \star)$. $\| \star \|_{eig}$ means the biggest eigenvalue of $\star$; For a symmetric positive definite matrix $\star$, $\| \star \|_{eig} = \| \star \|_2^2$. $\text{dist}(\star, \bullet) = \| \log(\star^\top \cdot \bullet) \|_F$ is the geodesic

distance between $\star$ and $\bullet$ in $SO(n)$, $SO(n)$ (special orthogonal group) is defined as: $SO(n) \triangleq \{\boldsymbol{R} \in \mathbb{R}^{n\times n} : \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}_{n\times n}, \det(\boldsymbol{R}) = 1\}$. The vector norm is $\|\star\|^2 = \star^\top \cdot \star$, for a vector $\star$. $|\star|$ means the cardinality of the set $\star$. $\mathbb{E}\{\star\}$ means the mathematical expectation of $\star$. $[n]$ denotes the set $\{i \in \mathbb{N}^+ : i \leq n\}$. $\nabla_\star \bullet$ means the partial derivative of a function $\bullet$ with parameter $\star$. $\star \ltimes \bullet$ means the semi-product group of the group $\star$ and the group $\bullet$.

3.1 Related work

Pose graph SLAM is the synchronization on the group of rigid body motions in two-dimensional (2D) plane and three-dimensional (3D) space, $\mathbb{R}^2 \times SO(2)$ and $\mathbb{R}^3 \times SO(3)$. Multiple estimation problems, including pose-graph SLAM, fall into this category [17]. These synchronization problems in general give rise to a weighted graph representation. In essence, there is a correlation between the graphical structure of the 2D/3D pose-graph SLAM problem and its corresponding measurement network.

In the state-of-the-art algorithms of SLAM, SE-Sync [116] and Cartan-Sync [11] show outstanding computational efficiency (more than an order of magnitude faster) compared with the GN based approach under the assumption that the rotation noise obeying the isotropic Langevin distribution on $SO(2)$ and $SO(3)$. Nevertheless, the FIM is not presented in the literature. Beyond that, from the graphical point of view, we know that adding relative measurements between the poses, which is equivalent to introducing new edges to the corresponding graph, helps to reduce the uncertainty of the estimator. In 2D pose-graph SLAM, the FIM has been shown to be closely related to the graph structure of the measurements network, in particular, the weighted Laplacian matrix [79] [78] assuming Gaussian noise on the relative pose orientation and ignored the wrap-around issue.

It is known that, in a flat Euclidean space, the classical CRLB result for any

unbiased estimator provides us with a simple but strong relation between the covariance matrix $\boldsymbol{C}$ and the FIM $\boldsymbol{F}$: $\boldsymbol{C} \succeq \boldsymbol{F}^{-1}$[37]. Because of the non-flat property of the parameter space of 3D pose-graph SLAM, its CRLB does not follow this simple expression. The curvature terms of the space will be introduced to show the rigorous CRLB.

Because of the sparseness advantage of the FIM, the TOED, including A-optimality, D-optimality, E-optimality, and T-optimality on the FIM are widely used in decision making under uncertainty and belief space planning, with applications including autonomous driving, surveillance, and active SLAM [17]. For example, in [140], the D-optimality, E-optimality and T-optimality metrics are used in active visual object reconstruction, and the T-optimality metric is applied to solve the sensor selection problem in Large Sensor Networks [121] [96]. The TOED is closely related to the graphical structure of the block design [5]. In 2D pose/feature-graph SLAM with the block-isotropic Gaussian noise, the D-optimality metric can be bounded by a expression related to the weighted number of the spanning trees [78].

Although many efficient optimization algorithms have been developed, the achievable estimated uncertainty is not well studied. Boumal [9] proposes the FIM for the estimation problems when the actual parameter space is a Riemannian sub-manifolds or a Riemannian quotient manifold. In [9], Boumal also shows two simple examples based on isotropic Gaussian noise: synchronization on the group of translation $\mathbb{R}^n$ and simple version of synchronization on the group of rotation $SO(3)$. In his later work [10], the conclusions are extended into the general rotation group $SO(n)$ based on several kinds of Gaussian-like, but non-Gaussian, noises. Especially for the isotropic Langevin noise, which has attracted some robotics researchers' attention. As the state-of-the-art back-end algorithms, the convex relaxation based SE-Sync [116] and Cartan-Sync [11] are built based on the assumption of Gaussian noise (for translation) and isotropic Langevin noise (for rotation). To the best of our knowl-

edge, the FIM for pose-graph SLAM based on these noises, whose parameter space is the product manifold $\mathbb{R}^n \times SO(n),\ n = 2, 3$, has not been analyzed.

CRLB, as a classical tool in estimation theory [113], provides a lower bound on the variance of any unbiased estimator for an estimated problem [10]. The traditional CRLB is defined in a flat Euclidean space. Simith [123] extends the theory of CRLB into the general non-flat manifold. Because the FIM will become singular when no anchor is provided in the estimation problem, Xavier and Barroso [142] use the pseudoinverse of the FIM for this anchor-free case. Based on these new extended tools, Boumal presents the CRLB for the synchronization of rotations $SO(n)$ in both anchored and anchor-free cases. Pose-graph SLAM is an anchored estimation problem in a product manifold $\mathbb{R}^n \times SO(n),\ n = 2, 3$ commonly.

In fact, there are two different Lie group representations corresponding to 2D/3D pose-graph SLAM: $\mathbb{R}^n \times SO(n)$ and $SE(n) = \mathbb{R}^n \ltimes SO(n)$. The direct product of $SO(n)$ and $\mathbb{R}^n$ manifolds $\mathbb{R}^n \times SO(n)$ can be represented as a $(2n+1)\times(2n+1)$ matrix with $n + \frac{n(n-1)}{2}$-dimensional minimal representation. The $\mathbb{R}^n$ and $SO(n)$ group can be considered as two separated parts easily, because of the separated Riemannian structure (decided by inner product) [130]. The special Euclidean group $SE(n)$ is isomorphic to $\mathbb{R}^n \times SO(n)$ but with different Riemannian structure (complex and coupled inner product) [13]. It can be represented as a $(n+1)\times(n+1)$ homogeneous transformation matrix with $n + \frac{n(n-1)}{2}$-dimensional minimal representation. Because of the different Riemannian structure, these two groups have the different tangent spaces and gradient forms, which leads to different FIM and CRLB. Compared with $SE(3)$, the direct product group $\mathbb{R}^n \times SO(n)$ can remain bi-invariance property by the simple combined Riemannian metric, which is defined as the sum of the metrics of $\mathbb{R}^n$ and $SO(n)$. However, the manifold $SE(3)$ does not have any bi-invariant metric, which makes the more complicated formulations for these concepts, such as curvature terms. Meanwhile, many recent popular SLAM pose-graph optimization

methods, such as SE-sync method and initialization techniques [23], separate the SLAM problem into two parts: the rotation estimation and the rest linear least squares problem for translation. Their objective functions are built based on the direct product group $\mathbb{R}^n \times SO(n)$. Hence, in this thesis, we only consider our problem on the direct product group $\mathbb{R}^n \times SO(n)$.

In [79, 78, 80], the authors explore the impact of the graphical structure of SLAM on some of the desirable attributes of some estimation problems: linear sensor network (SN), compass-SLAM and 2D pose-graph SLAM with block-isotropic Gaussian noise. In linear-SN and compass-SLAM, the FIM is proportional to the reduced Laplacian matrix of the corresponding graph, which helps to directly connect the optimal design of the FIM with the structure of the measurement graph. For 2D pose-graph SLAM with the block-isotropic Gaussian noise, the authors state that the D-optimality metric of the FIM can be bounded by a expression related to the weighted number of spanning trees of the measurement graph (weighted tree-connectivity). Based on the lower bounds, a new near-t-optimal graph synthesis framework is put forward for the measurement selection, pose-graph pruning problems, and D-optimality-aware SLAM front-end.

3.2 Synchronization on $\mathbb{R}^n \times SO(n)$, pose-graph SLAM

3.2.1 Synchronization on $\mathbb{R}^n \times SO(n)$

Synchronization on the group of the rigid body motions in 2D plane and 3D space, $\mathbb{R}^n \times SO(n),\ n = 2, 3$, is the problem of estimating a set of positions $\boldsymbol{x}_0,\ \boldsymbol{x}_1, \cdots,\ \boldsymbol{x}_{n_p} \in \mathbb{R}^n$ and rotations $\boldsymbol{R}_0,\ \boldsymbol{R}_1, \cdots,\ \boldsymbol{R}_{n_p} \in SO(n)$ from noisy measurements of some relative rotations $\boldsymbol{R}_j \boldsymbol{R}_i^\top$ and relative coordinate transformations $\boldsymbol{R}_i^\top (\boldsymbol{x}_j - \boldsymbol{x}_i)$.

In 2D/3D pose-graph SLAM, considering the anchored pose, the parameter space

is: $\mathcal{P} = \{\mathbb{R}^n \times \cdots \times \mathbb{R}^n\}_{n_p+1} \times \{SO(n) \times \cdots \times SO(n)\}_{n_p+1}$. We can create a new rotation graph $\mathcal{G}_1 = (\mathcal{V}, \mathcal{F})$, whose nodes only represent the rotations of the poses, by the pose graph $\mathcal{G}$. The rotation graph $\mathcal{G}_1$ is un-directed, because the measurements (i, j) introduce the same information for i-th node and j-th node.

For the pose-graph edge, $(i, j) \in \mathcal{E}$, we have a measurement $\boldsymbol{p}_{ij} \in \mathbb{R}^n$ of the relative noisy measurement between i-th and j-th poses:

$$\boldsymbol{p}_{ij} = \boldsymbol{R}_i^\top (\boldsymbol{x}_j - \boldsymbol{x}_i) + \boldsymbol{y}_{ij} \tag{3.1}$$

where $\boldsymbol{y}_{ij}$ is a random vector whose distributed function $f_{ij} : \mathbb{R}^n \to \mathbb{R}^+$ meets an isotropic Gaussian distribution:

$$\begin{aligned} f_{ij}(\boldsymbol{y}_{ij}) =& \frac{1}{(2\pi)^{n/2}\ \det(\boldsymbol{\Sigma}_{ij})^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{y}_{ij}^\top \boldsymbol{\Sigma}_{ij}^{-1} \boldsymbol{y}_{ij})\right) \\ \boldsymbol{\Sigma}_{ij} =& \delta_{ij}^2 \boldsymbol{I}_{n\times n} \end{aligned} \tag{3.2}$$

where δ_{ij} means the isotropic variance value. We write $\boldsymbol{y}_{ij} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{ij})$ to mean that $\boldsymbol{y}_{ij}$ is a random vector with probability density functions (PDF) (3.2).

For the edge in the rotation graph $(i, j) \in \mathcal{F}$, we have the relative noisy rotation $\boldsymbol{H}_{ij} \in SO(n)$ between $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$:

$$\boldsymbol{H}_{ij} = \boldsymbol{Z}_{ij} \boldsymbol{R}_j \boldsymbol{R}_i^\top \tag{3.3}$$

where $\boldsymbol{Z}_{ij}$ is a random rotation whose distributed function $\widehat{f_{ij}} : SO(n) \to \mathbb{R}^+$ meets

an isotropic Langevin distribution with mean $\boldsymbol{I}_{n\times n}$ and concentration $\kappa_{ij} \geq 0$ [10]:

$$\begin{aligned} \widehat{f}_{ij}(\boldsymbol{Z}_{ij}) =& \frac{1}{c_n(\kappa_{ij})} \exp\left(\kappa_{ij}\ \mathrm{trace}(\boldsymbol{Z}_{ij})\right) \\ c_2(\kappa_{ij}) =& I_0(2\kappa_{ij}) \\ c_3(\kappa_{ij}) =& \exp(\kappa_{ij})\left(I_0(2\kappa_{ij}) - I_1(2\kappa_{ij})\right) \\ I_v(2\kappa_{ij}) =& \frac{1}{2\pi}\int_{-\pi}^{\pi} \exp\left(2\kappa_{ij}\cos(\theta)\right)\cos(v\theta)d\theta \end{aligned} \tag{3.4}$$

where $c_n(\kappa_{ij}),\ n = 2, 3$ is a normalization constant such that $\widehat{f}_{ij}$ has unit mass. $I_v(2\kappa_{ij}),\ v = \{0, 1, 2, \cdots\} \in \mathbb{Z}$ means the modified Bessel functions [139]. We write $\boldsymbol{Z}_{ij} \sim Lang(\boldsymbol{I}_{n\times n}, \kappa_{ij})$ to mean that $\boldsymbol{Z}_{ij}$ is a random variable with PDF (3.4). This PDF meets the assumptions shown in [10]: 1. smoothness and support; 2. independence; 3. invariance assumptions.

Under the independence assumption, the log-likelihood of the estimator $\boldsymbol{\theta} = \boldsymbol{x} \times \boldsymbol{R} = (\boldsymbol{x}_0, \cdots, \boldsymbol{x}_{n_p}, \boldsymbol{R}_0, \cdots, \boldsymbol{R}_{n_p}) \in \mathcal{P}$, given the measurements $\boldsymbol{p}_{ij}$ and $\boldsymbol{H}_{ij}$, is given by:

$$L(\boldsymbol{y}; \boldsymbol{\theta}) = \sum_{(i,j)\in\mathcal{E}} \log f_{ij}(\boldsymbol{p}_{ij} - \boldsymbol{R}_i^\top(\boldsymbol{x}_j - \boldsymbol{x}_i)) + \frac{1}{2}\sum_{(i,j)\in\mathcal{F}} \log \widehat{f_{ij}}(\boldsymbol{H}_{ij}\boldsymbol{R}_i\boldsymbol{R}_j^\top). \tag{3.5}$$

The coefficient $\frac{1}{2}$ is used to balance the information in the un-directed rotation graph $\mathcal{G}_1$ and the directed pose-graph $\mathcal{G}$, satisfying $(i, j) \in \mathcal{F} \Leftrightarrow j \in V_i$, and $\sum_{(i,j)\in\mathcal{F}} \log \widehat{f_{ij}}(\boldsymbol{H}_{ij}\boldsymbol{R}_i\boldsymbol{R}_j^\top) = \sum_i \sum_{j\in V_i} \log \widehat{f_{ij}}(\boldsymbol{H}_{ij}\boldsymbol{R}_i\boldsymbol{R}_j^\top) = 2\sum_{(i,j)\in\mathcal{E}} \log \widehat{f_{ij}}(\boldsymbol{H}_{ij}\boldsymbol{R}_i\boldsymbol{R}_j^\top)$.

By introducing the PDF functions (3.2), (3.4), finding the maximum of the log-likelihood function (3.5) is equivalent to:

$$\max_{\boldsymbol{\theta}\in\mathcal{P}} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij}\mathrm{trace}(\boldsymbol{H}_{ij}\boldsymbol{R}_i\boldsymbol{R}_j^\top) - \sum_{(i,j)\in\mathcal{E}} \frac{\delta_{ij}^{-2}}{2}\|\boldsymbol{p}_{ij} - \boldsymbol{R}_i^\top(\boldsymbol{x}_j - \boldsymbol{x}_i)\|_2^2. \tag{3.6}$$

3.2.2 Geometry of the parameter spaces

The FIM is a classical tool for estimation problems on Euclidean spaces. In order to define the FIM on a manifold $\mathbb{R}^n \times SO(n),\ n = 2, 3$, we need to define some notions and tools to describe the parameter spaces for the synchronization.

Tangent space on $SO(n)$ As a Lie group, the dimension of its least representation is $d = \frac{n(n-1)}{2}$. We can admit a tangent space $\mathcal{T}_{\boldsymbol{Q}}SO(n),\ n = 2, 3$ for each rotation:

$$\mathcal{T}_{\boldsymbol{Q}}SO(n) = \boldsymbol{Q}so(n) \triangleq \{\boldsymbol{Q\Omega} : \boldsymbol{\Omega} \in \mathbb{R}^{n\times n}, \boldsymbol{\Omega}^\top + \boldsymbol{\Omega} = 0\} \tag{3.7}$$

where $\boldsymbol{Q} \in SO(n) \in \mathbb{R}^{n\times n}$.

Inner product on $\mathbb{R}^n$ **and** $SO(n)$ Based on the Riemannian metric of the manifold $\mathbb{R}^n \times SO(n)$, we define the inner products on the tangent space on $\mathbb{R}^n$ and $SO(n)$ respectively are:

$$\begin{cases} \langle \boldsymbol{\rho}_1, \boldsymbol{\rho}_2 \rangle_{\boldsymbol{X}} = \boldsymbol{\rho}_1^\top \boldsymbol{\rho}_2 & \boldsymbol{\rho}_1, \boldsymbol{\rho}_2 \in \mathbb{R}^n \\ \langle \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2 \rangle_{\boldsymbol{R}} = \mathrm{trace}(\boldsymbol{\Omega}_1^\top \boldsymbol{\Omega}_2) & \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2 \in \mathcal{T}_{\boldsymbol{Q}}SO(n). \end{cases} \tag{3.8}$$

Because $(\boldsymbol{Q\Omega}_1)^\top \boldsymbol{Q\Omega}_2 = \boldsymbol{\Omega}_1^\top \boldsymbol{\Omega}_2$, we omit the tangent subscripts $\boldsymbol{Q}$ in above equation and all related inner product equations of this paper, for better readability. The Riemannian metric of $\mathbb{R}^n \times SO(n)$ is the sum of the Riemannian metrics of $\mathbb{R}^n$ and $SO(n)$, which helps to maintain the bi-invariance property. $\forall(\boldsymbol{\rho}, \boldsymbol{Q}) \in \mathbb{R}^n \times SO(n),\ \forall(\boldsymbol{\rho}_i, \boldsymbol{\Omega}_i) \in \mathcal{T}_{(\boldsymbol{\rho},\boldsymbol{Q})}(\mathbb{R}^n \times SO(n)),\ i = 1, 2$, the specific expression of the Riemannian metric of $\mathbb{R}^n \times SO(n)$ is defined as:

$$\langle (\boldsymbol{\rho}_1, \boldsymbol{\Omega}_1), (\boldsymbol{\rho}_2, \boldsymbol{\Omega}_2) \rangle := \langle \boldsymbol{\rho}_1, \boldsymbol{\rho}_2 \rangle_{\boldsymbol{X}} + \langle \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2 \rangle_{\boldsymbol{R}}. \tag{3.9}$$

Gradient on $SO(n)$ Let $h : SO(n) \to \mathbb{R}$ be a differentiable function, we can define the gradient of h by:

$$\begin{aligned} \text{grad } h(\boldsymbol{Q}) &= \boldsymbol{Q}\text{skew}(\boldsymbol{Q}^\top \nabla h(\boldsymbol{Q})), \\ \text{skew}(\star) &\triangleq (\star - \star^\top)/2, \end{aligned} \tag{3.10}$$

where $\nabla h(\boldsymbol{Q})$ means the gradient of h seen as a Euclidean function in $\mathbb{R}^{n\times n}$.

The directional derivative of h at $\boldsymbol{Q}$ along $\boldsymbol{Q\Omega}$ can be written as:

$$< \text{grad } h(\boldsymbol{Q}), \boldsymbol{Q\Omega} >_{\boldsymbol{R}} \tag{3.11}$$

where $\boldsymbol{Q\Omega}$ is a tangent vector in the tangent space.

3.2.3 Definition of FIM

Definition [10]: Let $\boldsymbol{\theta} \in \mathcal{P}$ be unknown parameter and $f(\boldsymbol{y};\boldsymbol{\theta})$ be the PDF of the measurement $\boldsymbol{y}$ conditioned by $\boldsymbol{\theta}$ (the measurement noises are shown in the PDFs (3.2) and (3.4)). Based on the log-likelihood function $L(\boldsymbol{y};\boldsymbol{\theta}) = \log f(\boldsymbol{y};\boldsymbol{\theta})$ shown in (3.5) and the orthonormal basis, the (i,j)-th element of the FIM is defined as:

$$\boldsymbol{F}_{(i,j)} = \mathbb{E}\left\{\langle \text{grad}L(\boldsymbol{y};\boldsymbol{\theta}), e_i\rangle \cdot \langle \text{grad}L(\boldsymbol{y};\boldsymbol{\theta}), e_j\rangle\right\}, \tag{3.12}$$

where e_i and e_j are the i-th and j-th bases of the tangent space of the parameters. Expectations are taken w.r.t. the measurement $\boldsymbol{y}$. They will be defined based on the parameter space [9]. It is noted that the FIM is directly decided by the bases. For a parameter space, there may exist different bases.

3.3 FIM for 2D pose-graph SLAM

As shown in Section 3.2.2, the tools, including the tangent space, the inner product and the gradient, are completely suitable for both the 2D ($\mathbb{R}^2 \times SO(2)$) and 3D ($\mathbb{R}^3 \times SO(3)$) parameter space. In this part, we will show the orthonormal basis for the group $\mathbb{R}^2 \times SO(2)$.

In 2D case, define*:

$$\boldsymbol{E} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \tag{3.13}$$

the orthogonal basis $\boldsymbol{E}^{\boldsymbol{x},\boldsymbol{R}} = (\boldsymbol{E}^{\boldsymbol{x}}_{0,1}, \boldsymbol{E}^{\boldsymbol{x}}_{0,2} \cdots, \boldsymbol{E}^{\boldsymbol{x}}_{n_p,1}, \boldsymbol{E}^{\boldsymbol{x}}_{n_p,2}, \boldsymbol{E}^{\boldsymbol{R}}_{0}, \cdots, \boldsymbol{E}^{\boldsymbol{R}}_{n_p})$ of the tangent space $\mathcal{T}_{(\boldsymbol{x},\boldsymbol{R})}\mathcal{P}$ can be fixed as:

$$\begin{aligned}
\boldsymbol{E}^{\boldsymbol{x}}_{i,k} &= ({\boldsymbol{E}^{X}_{i,k}}^{\top}; \mathbf{0}_{2(n_p+1)\times 2}),\ i \in \{0,\ 1, \cdots,\ n_p\},\ k = 1, 2 \\
\boldsymbol{E}^{\boldsymbol{R}}_{j} &= (\mathbf{0}_{2(n_p+1)\times 1}; {\boldsymbol{E}^{R}_{j}}^{\top}),\ j \in \{0,\ 1, \cdots,\ n_p\}, \\
\boldsymbol{E}^{X}_{i,k} &= (\mathbf{0}_{1\times 2}, \cdots, \mathbf{0}_{1\times 2}, \underbrace{\underbrace{1}_{k-\text{th}}}_{i-\text{th}}, 0, \mathbf{0}_{1\times 2}, \cdots, \mathbf{0}_{1\times 2})_{1\times 2(n_p+1)}, \\
\boldsymbol{E}^{R}_{j} &= (\mathbf{0}_{2\times 2}, \cdots, \mathbf{0}_{2\times 2}, \underbrace{\boldsymbol{R}_j\boldsymbol{E}}_{j-\text{th}}, \mathbf{0}_{2\times 2}, \cdots, \mathbf{0}_{2\times 2})_{2\times 2(n_p+1)},
\end{aligned} \tag{3.14}$$

where $\boldsymbol{E}^{\boldsymbol{x}}_{i,k}$ and $\boldsymbol{E}^{\boldsymbol{R}}_{j}$ are corresponding to the k-axis coordinate of the i-th pose $(\boldsymbol{x}_i, \boldsymbol{R}_i)$ and the rotation parameter of the j-th pose $(\boldsymbol{x}_j, \boldsymbol{R}_j)$.

*It is noted that, for 2D case, we present the FIM using the orthogonal basis but not the orthonormal basis (orthonormal is a subset of orthogonal). Commonly, in the FIM definition (3.12), the basis used is the orthonormal basis. However, in most SLAM methods [**?**, **?**], the rotation matrix is represented by the minimal representation vectors $\boldsymbol{\phi}$ ($\exp(\boldsymbol{\phi}^{\wedge})$ is the corresponding rotation matrix, where $\star^{\wedge}$ means the skew-symmetric matrix of $\star$), which means that, for simplicity, the orthogonal basis (3.13) (3.14) is used in the 2D case. Moreover, for the 2D rotation matrix, its corresponding Euler angle value is approximately equal to the minimal Lie group representation using the basis (3.13) (3.14). Thus for the 2D case in this paper, we present the results using this orthogonal basis (3.13) (3.14) corresponding to the Euler angle. If using the orthonormal basis $\boldsymbol{E}/\sqrt{2}$, the derivation process is similar and the weight values $w_{ij}^{SO(2)}$ and ψ_i for rotation group part in the FIM (3.15) will be $w_{ij}^{SO(2)}/2$ and $\psi_i/2$.

Because of the group $\mathbb{R}^2 \times SO(2)$, there are four different parts in the FIM: One sub-matrix corresponding to the Euclidean space $\mathbb{R}^2$. One sub-matrix corresponding to $SO(2)$ Lie group, and other two sub-matrices related to the coupling part of $\mathbb{R}^2$ Euclidean space and $SO(2)$ Lie group. Based on the definition (3.12), we can prove the following theorem:

Theorem 1. *For the 2D case of the pose graph problem* (3.6)*, given the basis* (3.14)*, the FIM is:*

$$\boldsymbol{\mathcal{I}}_{2D} = \left[\begin{array}{c:c} \boldsymbol{L}_w^{\mathbb{R}^2} & \triangle_w^\top \\ \hdashline \triangle_w & \boldsymbol{L}_w^{SO(2)} + diag\{\psi_1, \cdots, \psi_{n_p}\} \end{array}\right], \tag{3.15}$$

where $\boldsymbol{L}_w^{\mathbb{R}^2}$ *is the sub-FIM corresponding to the Euclidean space* $\mathbb{R}^2$*, satisfying* $\boldsymbol{L}_w^{\mathbb{R}^2} = \boldsymbol{L}_{w_{\mathbb{R}}} \otimes \boldsymbol{I}_{2\times 2}$*.* $\boldsymbol{L}_{w_{\mathbb{R}}}$ *is the weighted Laplacian matrix, of which weight value* $w_{ij}^{\mathbb{R}}$ *for* (i,j)*-th edge is* δ_{ij}^{-2}*;* $\boldsymbol{L}_w^{SO(2)} + diag\{\psi_1, \cdots, \psi_{n_p}\}$ *is the sub-FIM corresponding to the* $SO(2)$ *Lie group, satisfying* $\boldsymbol{L}_w^{SO(2)} = \boldsymbol{L}_{w_{SO(2)}} \otimes \boldsymbol{I}_{d\times d}$*, where* d *is given in Section 3.2.2.* $\boldsymbol{L}_{w_{SO(2)}}$ *is the weighted Laplacian matrix, of which weight value* $w_{ij}^{SO(2)}$ *for* (i,j)*-th edge is* $2\kappa_{ij}\frac{I_1(2\kappa_{ij})}{I_0(2\kappa_{ij})}$*.* $\psi_i = \sum_{j\in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2,\ i = 1, 2, \cdots, n_p$*; The* (i, i_1)*-th block of the* $SO(2)$ *by* $\mathbb{R}^2$ *coupling sub-matrix* $\triangle_w$ *corresponding to the* $(n_p + 1 + i, i_1)$*-th block of the FIM is:*

$$(\triangle_w)_{i,i_1} = \begin{cases} \sum_{j\in V_i^+} \delta_{ij}^{-2} (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{E} & i = i_1 \\ \delta_{ii_1}^{-2} (\boldsymbol{x}_{i_1} - \boldsymbol{x}_i)^\top \boldsymbol{E} & (i, i_1) \in \mathcal{E} \\ \boldsymbol{0}_{1\times 2} & \textit{else.} \end{cases} \tag{3.16}$$

Proof. See Appendix B.1.1. □

Remark 1. *It is noted that, for 2D pose-graph SLAM, if the first node* $(\boldsymbol{x}_0,\ \boldsymbol{R}_0)$ *is anchored, the rows and columns corresponding to node* $(\boldsymbol{x}_0,\ \boldsymbol{R}_0)$ *of the FIM need to be deleted. So the FIM of the SLAM problem is related to the weighted reduced Laplacian matrix.*

3.3.1 Discussion on the FIM for 2D pose-graph SLAM

In the previous 2D work [79], [78] and [80], based on the block-isotropic Gaussian noise and the Euler angle, we get a similar formulation of the FIM by: $\boldsymbol{\mathcal{I}}_{2D} = \boldsymbol{J}^\top \boldsymbol{\Sigma}_{2D}^{-1} \boldsymbol{J}$, where $\boldsymbol{J}$ is the Jacobian matrix of the factors of the pose-graph SLAM, $\boldsymbol{\Sigma}_{2D}$ is the block-diagonal matrix whose non-zero blocks are the covariances of the factors of the 2D pose-graph SLAM. In fact, there are three important differences between this work with the previous work [79, 78, 80]:

Different noise assumptions The noise for the rotation part used in this work is different from the Gaussian noise. For the Gaussian noise, we have following result: If $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{P})$, we can get $\boldsymbol{G}\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{G}\boldsymbol{\mu}, \boldsymbol{G}\boldsymbol{P}\boldsymbol{G}^\top)$. So the FIM can be computed by $\boldsymbol{J}^\top \boldsymbol{\Sigma}_{2D}^{-1} \boldsymbol{J}$. This equation may not be suitable for other kinds of noises, so we need to compute the FIM based on its general definition shown in (3.12).

Lie group representation instead of Euler angle The previous work [79, 78, 80] is built based on the Euler angle representation for the orientation of the poses. The measurement of the rotation part is simply written as the subtraction of the orientation. Because of the non-uniqueness of the Euler angle and the periodicity of the trigonometric functions, the wraparound problem will lead the measurement function to be complex instead of a simple subtraction. In this work, the formulation is built based on the rotation group, so there is no wraparound issue.

Expandability of results from 2D to 3D Based on the Gaussian noise on the Euler angle, for the 3D case, the relative pose measurement can not be written as a simple subtraction formulation, which is greatly different from the 2D case, so the expandability of the original method is limited in 2D case. Based on the rigorous treatment of uncertainty on $\mathbb{R}^n \times SO(n)$ (isotropic Langevin noise for rotation and

block-isotropic Gaussian noise for translation), the similar but more complex results are obtained rigorously in the 2D case and further extended into the 3D case in the following section.

3.4 FIM for 3D pose-graph SLAM

As shown in Section 3.2.2, the tools, including the tangent space, the inner product and the gradient, are also suitable for the 3D ($\mathbb{R}^3 \times SO(3)$) parameter space. For the orthonormal basis, we will show its difference with (3.14) because of the higher dimensions. In 3D case, define[†]:

$$\boldsymbol{E}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}, \boldsymbol{E}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \boldsymbol{E}_3 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.17}$$

the orthonormal basis $\boldsymbol{E}^{\boldsymbol{x},\boldsymbol{R}} = (\boldsymbol{E}^{\boldsymbol{x}}_{0,1}, \boldsymbol{E}^{\boldsymbol{x}}_{0,2}, \boldsymbol{E}^{\boldsymbol{x}}_{0,3}, \cdots, \boldsymbol{E}^{\boldsymbol{x}}_{n_p,1}, \boldsymbol{E}^{\boldsymbol{x}}_{n_p,2}, \boldsymbol{E}^{\boldsymbol{x}}_{n_p,3}, \boldsymbol{E}^{\boldsymbol{R}}_{0,1}, \boldsymbol{E}^{\boldsymbol{R}}_{0,2}, \boldsymbol{E}^{\boldsymbol{R}}_{0,3}, \cdots, \boldsymbol{E}^{\boldsymbol{R}}_{n_p,1}, \boldsymbol{E}^{\boldsymbol{R}}_{n_p,2}, \boldsymbol{E}^{\boldsymbol{R}}_{n_p,3})$ of the tangent space $\mathcal{T}_{(\boldsymbol{x},\boldsymbol{R})}\mathcal{P}$ can be fixed as:

$$\begin{aligned}
\boldsymbol{E}^{\boldsymbol{x}}_{i,k} &= (\boldsymbol{E}^{X}_{i,k}{}^{\top}; \mathbf{0}_{9(n_p+1)\times 3}),\ i \in \{0, \cdots,\ n_p\},\ k = 1, 2, 3, \\
\boldsymbol{E}^{\boldsymbol{R}}_{j,k} &= (\mathbf{0}_{3(n_p+1)\times 1}; \boldsymbol{E}^{R}_{j,k}{}^{\top}),\ j \in \{0, \cdots,\ n_p\}, \\
\boldsymbol{E}^{X}_{i,k} &= (\mathbf{0}_{1\times 3}, \cdots, \mathbf{0}_{1\times 3}, \underbrace{\underbrace{1}_{k-\text{th}}, 0, 0}_{i-\text{th}}, \mathbf{0}_{1\times 3}, \cdots, \mathbf{0}_{1\times 3})_{1\times 3(n_p+1)}, \\
\boldsymbol{E}^{R}_{j,k} &= (\mathbf{0}_{3\times 3}, \cdots, \mathbf{0}_{3\times 3}, \underbrace{\boldsymbol{R}_j \boldsymbol{E}_k}_{3j+k-\text{th}}, \mathbf{0}_{3\times 3}, \cdots, \mathbf{0}_{3\times 3})_{3\times 9(n_p+1)}.
\end{aligned} \tag{3.18}$$

Similar to the FIM in Section 3.3, there are four parts in the FIM for the group $\mathbb{R}^3 \times SO(3)$ of the rigid body motions in 3D space. Based on the definition (3.12),

[†]Similar to the 2D case, in many applications, $\sqrt{2}\boldsymbol{E}_1$, $\sqrt{2}\boldsymbol{E}_2$, $\sqrt{2}\boldsymbol{E}_3$ are used in the bases of the manifold SO(3). In that case, the weight for rotation group part in the final FIM will be $2\omega_{ij}$, where ω_{ij} will be shown in (B.45).

we can prove the following theorem:

Theorem 2. *For the 3D case of the pose graph problem* (3.6)*, given the orthonormal basis* (3.18)*, the FIM is:*

$$\boldsymbol{\mathcal{I}}_{3D} = \left[\begin{array}{c|c} \boldsymbol{L}_w^{\mathbb{R}^3} & \triangle_w^{3D\top} \\ \hline \triangle_w^{3D} & \boldsymbol{L}_w^{SO(3)} + diag\{\boldsymbol{\Psi}_1, \cdots, \boldsymbol{\Psi}_{n_p}\} \end{array}\right], \tag{3.19}$$

where $\boldsymbol{L}_w^{\mathbb{R}^3}$ *is the sub-FIM corresponding to the Euclidean space* $\mathbb{R}^3$*, satisfying* $\boldsymbol{L}_w^{\mathbb{R}^3} = \boldsymbol{L}_{w_{\mathbb{R}}} \otimes \boldsymbol{I}_{3\times 3}$*.* $\boldsymbol{L}_{w_{\mathbb{R}}}$ *is the same as that in Theorem 1;* $\boldsymbol{L}_w^{SO(3)} + diag\{\boldsymbol{\Psi}_1, \cdots, \boldsymbol{\Psi}_{n_p}\}$ *is the sub-FIM corresponding to the* $SO(3)$ *Lie group, satisfying* $\boldsymbol{L}_w^{SO(3)} = \boldsymbol{L}_{w_{SO(3)}} \otimes \boldsymbol{I}_{d\times d}$*, where* d *is given in Section 3.2.2.* $\boldsymbol{L}_{w_{SO(3)}}$ *is the weighted Laplacian matrix, of which weight value* $w_{ij}^{SO(3)}$ *for* (i,j)*-th edge is* $\frac{1}{3}\frac{\kappa_{ij}^2(2I_0(2\kappa_{ij}) - I_1(2\kappa_{ij}) - 2I_2(2\kappa_{ij}) + I_3(2\kappa_{ij}))}{2I_0(2\kappa_{ij}) - 2I_1(2\kappa_{ij})}$*.* $\boldsymbol{\Psi}_i$ *satisfies:*

$$\begin{aligned} \boldsymbol{\Psi}_i &= \left[\begin{array}{ccc} \psi_i^{11} & \psi_i^{12} & \psi_i^{13} \\ \psi_i^{12} & \psi_i^{22} & \psi_i^{21} \\ \psi_i^{13} & \psi_i^{21} & \psi_i^{33} \end{array}\right], \ i = 1, 2, \cdots, n_p \\ \psi_i^{kl} &= \sum_{j\in V_i^+} \delta_{ij}^{-2} (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{I}_{3\times 3}^{k,l} \boldsymbol{R}_i^\top (\boldsymbol{x}_i - \boldsymbol{x}_j); \end{aligned} \tag{3.20}$$

Let $\boldsymbol{\varsigma}_{ij}^k = (\delta_{ij}^{-2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{E}_k \boldsymbol{R}_i^\top)^\top$*,* $k = 1, 2, 3$*, we have the* (i, i_1)*-th block of the* $SO(3)$ *by* $\mathbb{R}^3$ *coupling sub-matrix* $\triangle_w^{3D}$ *corresponding to the* $(n_p + 1 + i, i_1)$*-th block of the FIM:*

$$(\triangle_w^{3D})_{i,i_1} = \begin{cases} \left[\begin{array}{ccc} \sum_{j\in V_i^+} \boldsymbol{\varsigma}_{ij}^1 & \sum_{j\in V_i^+} \boldsymbol{\varsigma}_{ij}^2 & \sum_{j\in V_i^+} \boldsymbol{\varsigma}_{ij}^3 \end{array}\right]^\top & i = i_1 \\ \left[\begin{array}{ccc} -\boldsymbol{\varsigma}_{ii_1}^1 & -\boldsymbol{\varsigma}_{ii_1}^2 & -\boldsymbol{\varsigma}_{ii_1}^3 \end{array}\right]^\top & (i, i_1) \in \mathcal{E} \\ \boldsymbol{0}_{3\times 3} & else. \end{cases} \tag{3.21}$$

Proof. See Appendix B.1.2. □

Similarly, for the 3D case, if the first node is anchored, the corresponding rows and columns of the FIM need to be deleted. So the anchored FIM is related to the weighted reduced Laplacian matrix. The following CRLB and discussions on optimality metrics are focused on the pose-graph SLAM, so the FIM is defined based on the anchored situation.

3.5 CRLB for pose-graph SLAM

Classical CRLB gives a lower bound on the covariance matrix $\boldsymbol{C}$ of any unbiased estimator for an estimation problem in $\mathbb{R}^n$. In terms of the FIM $\boldsymbol{F} = \boldsymbol{\mathcal{I}}_{nD},\ n = 2, 3$ of that problem, the classical result reads $\boldsymbol{C} \succeq \boldsymbol{F}^{-1}$. However, because our parameter space $\mathcal{P}$ is a manifold instead of a flat Euclidean space, the CRLB takes up the more general form $\boldsymbol{C} \succeq \boldsymbol{F}^{-1} +$ curvature terms [123].

Inspired by [10], we also show that when the signal-to-noise ratio (SNR) is large enough, the curvature terms will become negligible. The CRLB is the asymptotic bound, which means only the leading-order curvature term has been computed. For SLAM, the non-flat property of the parameter space comes from the rotation group, so, in order to limit the Taylor truncation error, the SNR snr is heuristicly defined as[‡]:

$$snr = \frac{(n_p + 1 - |\mathcal{A}|)\mathbb{E}\{\text{dist}^2(\boldsymbol{Z}^{\boldsymbol{R}}_{uni}, \boldsymbol{I}_{3\times 3})\}}{9\text{trace}(\boldsymbol{I}^{SO(3)^{-1}})}, \tag{3.22}$$

where $\boldsymbol{I}^{SO(3)^{-1}}$ is the sub-matrix of the inverse function of the FIM $\boldsymbol{\mathcal{I}}_{3D}{}^{-1}$ corresponding to the $SO(3)$ Lie group, $|\mathcal{A}|$ means the number of anchored nodes (in general $|\mathcal{A}| = 1$ for SLAM), the expectation is taken w.r.t. $\boldsymbol{Z}^{\boldsymbol{R}}_{uni}$, uniformly distributed over $SO(3)$. For the numerator, $n_p + 1 - |\mathcal{A}|$ means the number of the

[‡]Because the SNR will be used in the 3D case only, we only show the formulation for 3D pose-graph.

estimated poses and $\mathbb{E}\{\text{dist}^2(\boldsymbol{Z}_{uni}^{\boldsymbol{R}}, \boldsymbol{I}_{3\times3})\}$ shows a suitable constant, which means the expected squared-error for a bad rotational estimator based on some uniformly random measurements (no information is available in the random estimator). The denominator is defined because the SNR is inversely proportional to the trace of the the rotational block of the inverse matrix of the FIM. When the quantity of the rotational measurements increases (large-scale $\boldsymbol{I}^{SO(3)}$), the SNR *snr* will be large. It is well-known that it's the rotational part specifically that makes the SLAM problem nontrivial. When the robot orientation is assumed to be known, the SLAM problem can be simplified as a linear-Gaussian estimation problem, called Compass-SLAM problem [78], whose globally-optimal solution can be computed easily by solving a linear least squares problem and the curvature terms will be zero. Hence, we do not consider anything about the performance of the position estimates in the defined SNR (3.22) since the position estimate has no effect on the curvature term.

3.5.1 CRLB for 2D pose-graph SLAM

Before the discussion about the CRLB of the synchronization of the manifolds $\mathbb{R}^2 \times SO(2)$, we introduce a lemma [24]:

Lemma 1. *Let $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$ be the product of two Riemannian manifolds, R be its curvature tensor, and R_1, R_2 be curvature tensors for $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively, then one can relate R, R_1 and R_2 by:*

$$R(X, Y) = R_1(X_1, Y_1) + R_2(X_2, Y_2), \tag{3.23}$$

where $X_i, Y_i \in \mathcal{T}(\mathcal{M}_i), i = 1, 2$ and $X = X_1 + X_2, Y = Y_1 + Y_2 \in \mathcal{T}(\mathcal{M})$, $\mathcal{T}(\star)$ means the tangent space of $\star$.

Based on Lemma 1, we can prove that the curvature tensor of the parameter space $\mathcal{P}_1 = \{\mathbb{R}^2 \times \cdots \times \mathbb{R}^2\}_{n_p} \times \{SO(2) \times \cdots \times SO(2)\}_{n_p}$ is equal to the sum

of the multiple curvature tensors of the group $\{\mathbb{R}^2 \times \cdots \times \mathbb{R}^2\}_{n_p}$ and the group $\{SO(2) \times \cdots \times SO(2)\}_{n_p}$. As the (product) Lie group, the curvature tensors of the parameter space $\mathcal{P}_1$ on the tangent space $\mathcal{T}_{(\boldsymbol{x},\boldsymbol{R})}\mathcal{P}_1$ is given by a simple formula [10]:

$$\langle R(\bar{\boldsymbol{X}}, \bar{\boldsymbol{\Omega}})\bar{\boldsymbol{\Omega}},\ \bar{\boldsymbol{X}}\rangle = \frac{1}{4}\|[\bar{\boldsymbol{X}}, \bar{\boldsymbol{\Omega}}]\|_F^2, \tag{3.24}$$

where $[\bar{\boldsymbol{X}}, \bar{\boldsymbol{\Omega}}]$ is the Lie bracket of $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{\Omega}}$, two vectors (not necessarily orthonormal) in the tangent space $\mathcal{T}_{(\boldsymbol{x},\boldsymbol{R})}\mathcal{P}_1$. Because of the Lie brackets and the bases (3.14), it is easy to find that these two groups are both flat and their curvature tensors are $\mathbf{0}$. So the curvature tensor of the space $\mathcal{P}_1$ is $\mathbf{0}$.

The CRLB for 2D pose-graph SLAM on $\mathbb{R}^2 \times SO(2)$ is:

$$\boldsymbol{C} \succeq \boldsymbol{F}^{-1} + \text{curvature terms} = \boldsymbol{F}^{-1} + \mathbf{0} = \boldsymbol{F}^{-1}. \tag{3.25}$$

So we can see that, for the parameter space $\mathcal{P}_1$, its CRLB formula is the same as the classical CRLB result for the flat Euclidean space shown as $\boldsymbol{C} \succeq \boldsymbol{F}^{-1}$.

3.5.2 CRLB for 3D pose-graph SLAM

Different from the $\mathbb{R}^2 \times SO(2)$ group, the manifold $\mathbb{R}^3 \times SO(3)$ is not a flat space. So we need to compute the curvature terms. Based on Lemma 1, we known that the curvature tensor of the parameter space $\mathcal{P}_3 = \{\mathbb{R}^3 \times \cdots \times \mathbb{R}^3\}_{n_p} \times \{SO(3) \times \cdots \times SO(3)\}_{n_p}$ is equal to the sum of the curvature tensor of the manifold $\{\mathbb{R}^3 \times \cdots \times \mathbb{R}^3\}_{n_p}$ and the manifold $\mathcal{P}_2 = \{SO(3) \times \cdots \times SO(3)\}_{n_p}$. The Euclidean space $\mathbb{R}^3$ is flat with a $\mathbf{0}$ curvature tensor, so the curvature tensor of the space $\mathcal{P}_3$ is determined by the curvature tensor of the manifold $\mathcal{P}_2$.

Based on Theorem 4 in [123], CRLB follows: $\boldsymbol{C} \geq \boldsymbol{F}^{-1} - \frac{1}{3}(R_m(\boldsymbol{F}^{-1})\boldsymbol{F}^{-1} + \boldsymbol{F}^{-1}R_m(\boldsymbol{F}^{-1}))$, where the operator R_m: $\mathbb{R}_{6n_p \times 6n_p} \to \mathbb{R}_{6n_p \times 6n_p}$ involves the Riemannian curvature tensor of the parameter space. The operator $R_m(\boldsymbol{F}^{-1}) = \frac{1}{4}\text{diag}\{\mathbf{0},$

ddiag($\boldsymbol{I}^{SO(3)^{-1}}$)} of the parameter space $\mathcal{P}_2$ for the anchored rotation synchronization has been shown by [10].

Following the same formulation as in [10], we can get the curvature tensor of 3D pose-graph SLAM:

$$\begin{aligned} \boldsymbol{C} &\succeq \boldsymbol{F}^{-1} - \frac{1}{12}(\text{ddiag}(\widetilde{\boldsymbol{L}})\boldsymbol{F}^{-1} + \boldsymbol{F}^{-1}\text{ddiag}(\widetilde{\boldsymbol{L}})), \\ \widetilde{\boldsymbol{L}} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}^{SO(3)^{-1}} \end{bmatrix}_{6n_p \times 6n_p}. \end{aligned} \tag{3.26}$$

It is easy to find that when *snr* is large enough, compared with $\boldsymbol{F}^{-1}$, the function ddiag($\widetilde{\boldsymbol{L}}$)$\boldsymbol{F}^{-1}$ + $\boldsymbol{F}^{-1}$ddiag($\widetilde{\boldsymbol{L}}$) is much smaller. So it is negligible.

3.6 Optimal experimental design metrics for pose-graph SLAM

It is known that the TOED, including T-optimality and D-optimality, for the FIM of the estimated state vector is a common way to evaluate the actions in terms of estimation accuracy. In this section, we will discuss and compare the optimal experimental design metrics of 2D/3D pose-graph SLAM and show the bounds of these indexes, which are much easier to be computed than the accurate index for the FIM.

3.6.1 T-optimality design metric

T-optimality design metric for the synchronization on $\mathbb{R}^2 \times SO(2)$

Based on (3.15), we can get the T-optimality design metric of the FIM:

$$\text{trace}(\boldsymbol{\mathcal{I}}_{2D}) = \text{trace}(\boldsymbol{L}_w^{\mathbb{R}^2}) + \text{trace}(\boldsymbol{L}_w^{SO(2)}) + \sum_{i=1}^{n_p} \sum_{j \in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2 \tag{3.27}$$

T-optimality design metric for the synchronization on $\mathbb{R}^3 \times SO(3)$

Based on (3.19), we can get the T-optimality design metric of the FIM:

$$\text{trace}(\boldsymbol{\mathcal{I}}_{3D}) = \text{trace}(\boldsymbol{L}_w^{\mathbb{R}^3}) + \text{trace}(\boldsymbol{L}_w^{SO(3)}) + \sum_{i=1}^{n_p} \text{trace}(\boldsymbol{\Psi}_i) \tag{3.28}$$

We can find that, in these three parts, only one part $\sum_{i=1}^{n_p} \text{trace}(\boldsymbol{\Psi}_i)$ includes the state vector $\boldsymbol{x}_i$, $\boldsymbol{x}_j$, $\boldsymbol{R}_i$. In fact, because of the special structure about $\boldsymbol{\Psi}_i$, we have: $\sum_{i=1}^{n_p} \text{trace}(\boldsymbol{\Psi}_i) = \sum_{i=1}^{n_p} \sum_{j \in V_i^+} \text{trace}(\boldsymbol{\Psi}_{(i,j)}) = \sum_{i=1}^{n_p} \sum_{j \in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2$. The proof is shown in Appendix B.1.6.

Further analysis about the T-optimality design metric

We know that the measurement function is: $\boldsymbol{p}_{ij} = \boldsymbol{R}_i^\top (\boldsymbol{x}_j - \boldsymbol{x}_i) + \boldsymbol{y}_{ij}$. If the solution is the "optimal" or "near-optimal" solution with the small noise, we have:

$$\delta_{ij}^{-2} \|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2 = \delta_{ij}^{-2} \|\boldsymbol{p}_{ij} - \boldsymbol{y}_{ij}\|^2 \approx \delta_{ij}^{-2} \|\boldsymbol{p}_{ij}\|^2 \tag{3.29}$$

Introduce (3.29) into (3.27) and (3.29), we have:

$$\text{trace}(\boldsymbol{\mathcal{I}}_{nD}) \approx \text{trace}(\boldsymbol{L}_w^{\mathbb{R}^n}) + \text{trace}(\boldsymbol{L}_w^{SO(n)}) + \sum_{i=1}^{n_p} \sum_{j \in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{p}_{ij}\|^2, \; n = 2, 3 \tag{3.30}$$

Based on (3.30), we can see that the trace function of the FIM is weakly related to the state vector obtained by the estimated result. So, in general, the T-optimality can be easily computed using above equation without considering the SLAM result. For many real word datasets, compared with the other parts, $\sum_{i=1}^{n_p} \sum_{j \in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{p}_{ij}\|^2$

is relatively small in general. We have:

$$\begin{aligned}\operatorname{trace}(\boldsymbol{\mathcal{I}}_{nD}) \rightarrow & \operatorname{trace}(\boldsymbol{L}_w^{\mathbb{R}^n}) + \operatorname{trace}(\boldsymbol{L}_w^{SO(n)}) \\ = & \sum_{i=1}^{n_p} \left(\sum_{j \in V_i} \omega_{ij} + n \sum_{j_1 \in V_i^+} \sigma_{ij_1}^{-2} + n \sum_{j_2 \in V_i^-} \sigma_{j_2 i}^{-2} \right) \\ = & \sum_{i=1}^{n_p} \sum_{j \in V_i} \left(\omega_{ij} + n \widehat{\sigma}_{ij}^{-2} \right)\end{aligned} \tag{3.31}$$

where $\widehat{\sigma}_{ij}$ means the covariance values corresponding to translation part, which are connected by the i-th and the j-th pose (in and out).

3.6.2 D-optimality design metric

D-optimality design is to use the log-determinant of the covariance matrix as an objective function. Due to the sparse structure of the FIM, we always compute $\log(\det(\boldsymbol{C}))$ via $\log(\det(\boldsymbol{F}))$: $\log(\det(\boldsymbol{C})) \approx \log(\det(\boldsymbol{F}^{-1})) = -\log(\det(\boldsymbol{F}))$. Some references show that the D-optimality metric can keep monotonicity during the exploration. Beside this, the D-optimality metric and the entropy have an explicit relationship [27]. D-optimality is a useful metric for quantifying the uncertainty of the robot and the generated map in an active SLAM context. However, the determinant function of a matrix is really expensive. In this part, we will derive the bounds of the D-optimality metric which are easier to compute and can be used to approximate the original metric.

Some results about D-optimality design metric for the synchronization of the group $\mathbb{R}^2 \times SO(2)$ based on the block-isotropic Gaussian noise has been discussed in [78]. Because the situation using the isotropic Langevin noise on $\mathbb{R}^2 \times SO(2)$ is similar to the ones of [78], in this part, we only show the extended result in the group $\mathbb{R}^3 \times SO(3)$.

Theorem 3. *Considering the 3D pose-graph SLAM problem in Section 5.2.2, its*

D-optimality design metric $\log(\det(\boldsymbol{\mathcal{I}}_{3D}))$ *of the FIM has a lower bound lb and an upper bound ub:*

$$lb \leq \log(\det(\boldsymbol{\mathcal{I}}_{3D})) \leq ub, \tag{3.32}$$

where $lb = \log(\det(\boldsymbol{L}_w^{\mathbb{R}^3}))+\log(\det(\boldsymbol{L}_w^{SO(3)}))$, $ub = \log(\det(\boldsymbol{L}_w^{\mathbb{R}^3}))+\sum_{i=1}^{3n_p}\log(\lambda_i(\boldsymbol{L}_w^{SO(3)})+\lambda_\infty)$, $\lambda_\infty = \|diag\{\boldsymbol{\Psi}_1,\cdots,\boldsymbol{\Psi}_{n_p}\}\|_{eig}$ *and* $\lambda_i(\boldsymbol{L}_w^{SO(3)})$ *means the i-th eigenvalue of* $\boldsymbol{L}_w^{SO(3)}$.

Proof. Based on the Schur's determinant formula [78] and (3.19), because $\boldsymbol{L}_w^{\mathbb{R}^3}$ is invertible, we have:

$$\begin{aligned}&\log(\det(\boldsymbol{\mathcal{I}}_{3D})) = \log(\det(\boldsymbol{L}_w^{\mathbb{R}^3})) + \log(\det(\boldsymbol{L}_w^{SO(3)}+\\ &\text{diag}\{\boldsymbol{\Psi}_1,\cdots,\boldsymbol{\Psi}_{n_p}\} - \triangle_w^{3D}\boldsymbol{L}_w^{\mathbb{R}^{3-1}}\triangle_w^{3D^\top})).\end{aligned} \tag{3.33}$$

Similar to the proof of Theorem 3 in [78], we can show $\triangle_w^{3D}\boldsymbol{L}_w^{\mathbb{R}^{3-1}}\triangle_w^{3D^\top} \succeq 0$ and $\text{diag}\{\boldsymbol{\Psi}_1,\cdots,\boldsymbol{\Psi}_{n_p}\} - \triangle_w^{3D}\boldsymbol{L}_w^{\mathbb{R}^{3-1}}\triangle_w^{3D^\top} \succeq 0$ are orthogonal projection matrices, then we have:

$$\begin{aligned}&lb = \log(\det(\boldsymbol{L}_w^{\mathbb{R}^3})) + \log(\det(\boldsymbol{L}_w^{SO(3)})) \leq\\ &\log\left(\det(\boldsymbol{\mathcal{I}}_{3D})\right) \leq \log(\det(\boldsymbol{L}_w^{\mathbb{R}^3})) + \log\left(\det\left(\boldsymbol{L}_w^{SO(3)}\right.\right.\\ &\left.\left.+\,\text{diag}\{\boldsymbol{\Psi}_1,\cdots,\boldsymbol{\Psi}_{n_p}\}\right)\right) < ub = \log(\det(\boldsymbol{L}_w^{\mathbb{R}^3}))\\ &+\log(\det(\boldsymbol{L}_w^{SO(3)} + \lambda_\infty \boldsymbol{I}_{3n_p\times 3n_p}))\\ &= \log(\det(\boldsymbol{L}_w^{\mathbb{R}^3})) + \sum_{i=1}^{3n_p}\log(\lambda_i(\boldsymbol{L}_w^{SO(3)}) + \lambda_\infty).\end{aligned} \tag{3.34}$$

□

Some discussions about λ_∞ are shown as follows:

Corollary 1. *The biggest eigenvalue* λ_∞ *in Theorem 3 is smaller than* $\max_{i=1,2,\cdots,n_p}\frac{1}{2}\sum_{j\in V_i^+}\delta_{ij}^{-2}\|\boldsymbol{x}_j - \boldsymbol{x}_i\|_2^2$.

Proof. Based on the Rayleigh quotients [54] and Appendix B.1.6, we know that the variational description of the maximal eigenvalue $\lambda_\infty(\boldsymbol{\Psi}_i)$ of the real symmetric matrix $\boldsymbol{\Psi}_i,\ i = 1, \cdots, n_p$ is:

$$\begin{aligned}
&\lambda_\infty(\boldsymbol{\Psi}_i) = \max_{\boldsymbol{x}\neq 0} \frac{\boldsymbol{x}^\top \boldsymbol{\Psi}_i \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \\
&\max_{\boldsymbol{x}\neq 0} \frac{\boldsymbol{x}^T \sum_{j\in V_i^+} \boldsymbol{\Psi}_{(i,j)} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \leq \sum_{j\in V_i^+} \max_{\boldsymbol{x}\neq 0} \frac{\boldsymbol{x}^\top \boldsymbol{\Psi}_{(i,j)} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \\
&= \sum_{j\in V_i^+} \lambda_\infty(\boldsymbol{\Psi}_{(i,j)}) = \sum_{j\in V_i^+} \frac{1}{2} \delta_{ij}^{-2} \|\boldsymbol{x}_j - \boldsymbol{x}_i\|_2^2.
\end{aligned} \tag{3.35}$$

All eigenvalues of a block diagonal matrix are the eigenvalues of all block matrices on the diagonal [54], so we have:

$$\begin{aligned}
\lambda_\infty &= \max_{i=1,2,\cdots,n_p} \lambda_\infty(\boldsymbol{\Psi}_i) \\
&\leq \max_{i=1,2,\cdots,n_p} \frac{1}{2} \sum_{j\in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{x}_j - \boldsymbol{x}_i\|_2^2.
\end{aligned} \tag{3.36}$$

□

3.6.3 Discussion and comparison

Efficiency of the metric

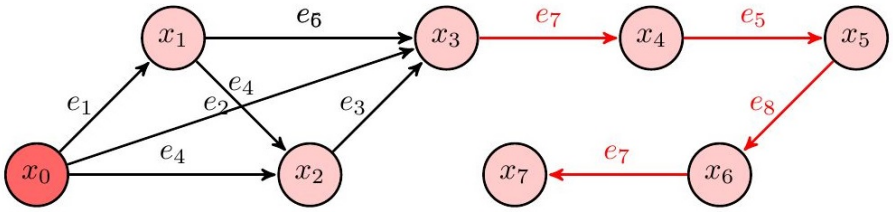
T-optimality metric The parameters ω_{ij} and $\widehat{\sigma}_{ij}$ are constant. We can assume that these two parameters are the same in every measurement process. Set: $c^\omega = \omega_{ij} + 3\widehat{\sigma}_{ij}^{-2}$, based on (3.31), we have:

$$\operatorname{trace}(\boldsymbol{\mathcal{I}}_{nD}) \approx \sum_{i=1}^{n_p} \sum_{j\in V_i} c^\omega = c^\omega \sum_{i=1}^{n_p} |V_i|,\ n = 2, 3 \tag{3.37}$$

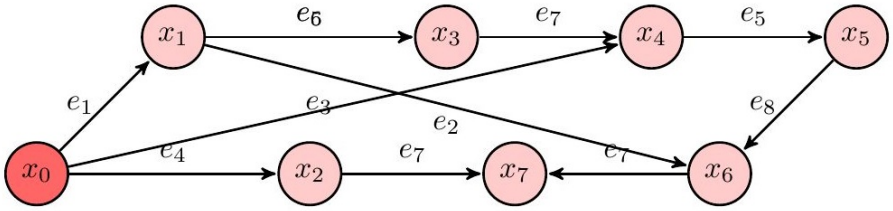
where $\sum_{i=1}^{n_p} |V_i|$ means the total node degree of the graph.

Based on (3.37), we can get a simple conclusion: For fixed number of pose, min-

imizing the estimated uncertainty is equivalent to gathering as many measurements as possible. Moreover, we can also see the limitation of the T-optimality metric: when two graphs have the same total node degree and same node number, the T-optimality metric can not be used to compute their uncertainty levels. Let's see the two graphs in Figure 4.1:



(a) G_1 (pose-graph 1).



(b) G_2 (pose-graph 2).

Figure 3.1 : Two examples of pose-graphs (with $\boldsymbol{x}_0$ as anchor)

Because of the same nodes and total node degree, graph 1 and graph 2 have the similar T-optimality objective function based on (3.34). However, it is easy to know that graph 2 is much better than graph 1 because of the loop closure. The uncertainty of poses $\boldsymbol{x}_4$ to $\boldsymbol{x}_7$ in graph 1 is larger than the ones in graph 2. So in some situations, the T-optimality will lost its efficiency.

D-optimality metric Based on (3.34) and [78], we have the lower bound ($lb = \log(\det(\boldsymbol{L}_w^{\mathbb{R}^n})) + \log(\det(\boldsymbol{L}_w^{SO(n)}))$) and the upper bound ($ub = \log(\det(\boldsymbol{L}_w^{\mathbb{R}^n})) + \sum_{i=1}^{dn_p} \log(\lambda_i(\boldsymbol{L}_w^{SO(n)}) + \lambda_\infty)$). For most pose-graph problems, the relative translations $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$ are usually small, so we have $ub \to lb$. Finally, we can get: $\log(\det(\boldsymbol{\mathcal{I}}_{nD})) \to lb = \log(\det(\boldsymbol{L}_w^{\mathbb{R}^n})) + \log(\det(\boldsymbol{L}_w^{SO(n)}))$. We can get the following

lemma:

Theorem 4. *Considering the 2D/3D pose-graph SLAM problem in Section 5.2.2,* $\xi \triangleq \max\limits_{i=1,2,\cdots,n_p} \sum_{j\in V_i^+} \delta_{ij}^{-2}\|\boldsymbol{x}_j-\boldsymbol{x}_i\|_2^2$ *(2D) or* $\xi \triangleq \max\limits_{i=1,2,\cdots,n_p} \frac{1}{2}\sum_{j\in V_i^+} \delta_{ij}^{-2}\|\boldsymbol{x}_j-\boldsymbol{x}_i\|_2^2$ *(3D) and* $\lambda_{min}(\boldsymbol{L}_w^{SO(n)})$ *is the minimal eigenvalue of* $\boldsymbol{L}_w^{SO(n)}$*, define* $\varepsilon = \log(\det(\boldsymbol{\mathcal{I}}_{nD})) - \log(\det(\boldsymbol{L}_w^{\mathbb{R}^n})) - \log(\det(\boldsymbol{L}_w^{SO(n)}))$*. Then we have,*

$$0 \leq \varepsilon \leq dn_p \log(1+\xi/\lambda_{min}(\boldsymbol{L}_w^{SO(n)})). \tag{3.38}$$

Proof. Based on the equations (3.34) and (3.36), we have:

$$\begin{aligned}
\varepsilon &= \log(\det(\boldsymbol{\mathcal{I}}_{nD})) - \log(\det(\boldsymbol{L}_w^{\mathbb{R}^n})) - \log(\det(\boldsymbol{L}_w^{SO(n)})) \\
&\leq \log(\det(\boldsymbol{L}_w^{SO(n)} + \xi \boldsymbol{I}_{dn_p\times dn_p})) - \log(\det(\boldsymbol{L}_w^{SO(n)})) \\
&= \log\prod_{i=1}^{dn_p} \frac{\lambda_i(\boldsymbol{L}_w^{SO(n)})+\xi}{\lambda_i(\boldsymbol{L}_w^{SO(n)})} \leq \log\left(1+\xi/\lambda_{min}(\boldsymbol{L}_w^{SO(n)})\right)^{dn_p} \\
&= dn_p \log\left(1+\xi/\lambda_{min}(\boldsymbol{L}_w^{SO(n)})\right),
\end{aligned} \tag{3.39}$$

where $\lambda_i(\boldsymbol{L}_w^{SO(n)}),\ i=1,\ 2,\cdots,\ dn_p$ are the eigenvalues of $\boldsymbol{L}_w^{SO(n)}$, $\lambda_{min}(\boldsymbol{L}_w^{SO(n)})$ is the minimal eigenvalue of $\boldsymbol{L}_w^{SO(n)}$. □

Based on the Kirchhoff's matrix tree theorem [28], it is easy to know that $\det(\boldsymbol{L}_w^{\mathbb{R}^n})$ and $\det(\boldsymbol{L}_w^{SO(n)})$ are equivalent to the weighted number of the spanning trees of the translation graph and the rotation graph. So, for a weighted graph $\mathcal{G}$ in pose-graph SLAM, the D-optimality design of the weighted Laplacian matrix is almost equal to maximize the weighted number of spanning trees of the graph $\mathcal{G}$ (also named weighted tree connectivity).

The 2D SLAM results with the Gaussian noise about the relationship between the D-optimality metric with the weighted tree connectivity have been discussed in [78]. In this thesis, we extend it into the 3D pose-graph SLAM situation. Thus, results

and all the further algorithms are derived from these results (including $k - ESP^{+}$ problem [78]) can be extended to 3D case.

Computational Complexity

T-optimality metric After constructing the FIM, the computational complexity of the trace function of the FIM is $O(n_p)$. In some special scenarios and applications, such as the active SLAM, some parts and some functions of the FIM can be reused and form the incremental method. The complexity can be further reduced to $O(L_p)$, where L_p is the predicted horizon. It is easy to find that the T-optimality metric is also a cost-effective metric.

D-optimality metric We can see that the bounds lb and ub of the D-optimality metric of the FIM are almost independent of the values of the pose $\boldsymbol{x}_i$ and $\boldsymbol{R}_i$, which leads to robust performance. Beyond that the update and operations on these bounds are easier than the real D-optimality metric of the FIM. In this part, we talk about the computational complexity of these bounds in 3D case (2D case is similar):

For the lower bound, we have two parts: $\log(\det(\boldsymbol{L}_w^{\mathbb{R}^3}))$ and $\log(\det(\boldsymbol{L}_w^{SO(3)}))$. Using the sparse Cholesky decomposition with a good fill-reducing permutation (Algorithm 1 in [78]), they can be computed much faster than the log-determinant function of the dense matrix $O(n_p^3)$; However, because $\boldsymbol{L}_w^{\mathbb{R}^3}$ and $\boldsymbol{L}_w^{SO(3)}$ have the same sparse structure, we can simply modify the original Algorithm 1 in [78] by the more efficient new algorithm (Algorithm 1):

In Algorithm 1, $\widehat{\boldsymbol{L}}_w^{SO(3)}$ and $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^3}$ are the weighted Laplacian matrix corresponding to the rotation graph and the translation graph without using Kronecker product, i. e. $\boldsymbol{L}_w^{SO(3)} = \widehat{\boldsymbol{L}}_w^{SO(3)} \otimes \boldsymbol{I}_{3\times 3}$ and $\boldsymbol{L}_w^{\mathbb{R}^3} = \widehat{\boldsymbol{L}}^{\mathbb{R}^3_w} \otimes \boldsymbol{I}_{3\times 3}$. $(\star)_{i,i}$ means the i-th diagonal element of the matrix $\star$. At the same time, the same order $\boldsymbol{l}$ can also be used to

Algorithm 1 Lower bound computation for D-optimality metric (3D case)

procedure LB($\widehat{\boldsymbol{L}}_w^{SO(3)}, \widehat{\boldsymbol{L}}_w^{\mathbb{R}^3}$)
  $\boldsymbol{l} \leftarrow COLAMD(\widehat{\boldsymbol{L}}_w^{SO(3)})$ ▷ Column approximate minimum degree permutation
  $\boldsymbol{L}_1 \leftarrow SparseCholesky(\widehat{\boldsymbol{L}}_w^{SO(3)}(\boldsymbol{l},\boldsymbol{l}))$ ▷ Sparse Cholesky factor based on $\boldsymbol{l}$ for $\widehat{\boldsymbol{L}}_w^{SO(3)}$
  $\boldsymbol{L}_2 \leftarrow SparseCholesky(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^3}(\boldsymbol{l},\boldsymbol{l}))$ ▷ Sparse Cholesky factor based on the same $\boldsymbol{l}$ for $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^3}$
  return $2 \cdot 3 \cdot \sum_i (\log(\boldsymbol{L}_1)_{i,i} + \log(\boldsymbol{L}_2)_{i,i})$

compute the upper bound.

The above algorithm is similar to the Algorithm 1 in [78]. The only difference is to re-use the result of the column approximate minimum degree reordering algorithm because of the same sparse structure.

For the upper bound ub, there are three parts: $\log(\det(\boldsymbol{L}_w^{\mathbb{R}^3}))$, $\lambda_i(\boldsymbol{L}_w^{SO(3)})$ and λ_∞. For the part $\lambda_i(\boldsymbol{L}_w^{SO(3)})$, we can use the Lanczos algorithm [92] and Fast Multi-pole Method [36] for the sparse Hermitian matrix (Information matrix meets). The Lanczos algorithm can help to generate a tridiagonal real symmetric matrix from the matrix $\boldsymbol{L}_w^{SO(3)}$ in complexity $O(d_n m_p n_p)$ and $O(d_n n_p^2)$ if $(m_p = n_p)$, where d_n is the average number of nonzero elements in a row of the matrix $\boldsymbol{L}_w^{SO(3)}$, m_p is the number of iterations in the Lanczos algorithm (as default, let $m_p = n_p$). For tridiagonal matrices, the Fast Multi-pole Method compute all eigenvalues in just $O(n_p \log n_p)$ operations. So the computing complexity of $\lambda_i(\boldsymbol{L}_w^{SO(3)})$ is totally $O(dn_p^2)+O(n_p \log n_p)$; For the part λ_∞, we have shown the analytical results $\lambda_\infty \approx \max\limits_{i=1,2,\cdots,n_p} \frac{1}{2} \sum_{j \in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{p}_{ij}\|^2$.

The above discussion is based on the operations on the whole FIM. Similar to the T-optimality metric, in some special scenarios and applications, such as the active SLAM, the incremental computation using the matrix determinant and re-use of calculation method (rAMDL) has been introduced into the computation of the D-optimality metric of the FIM [86]. Without the loop-closure, its computational

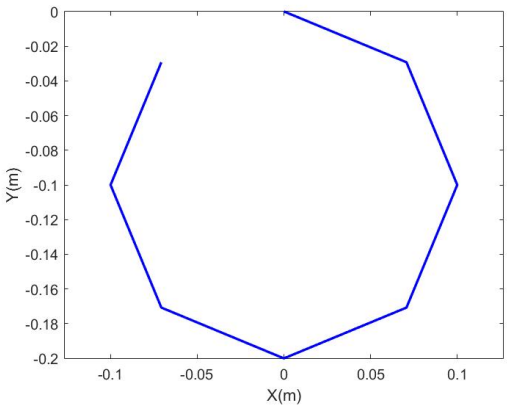


Figure 3.2 : Inital pose-graph

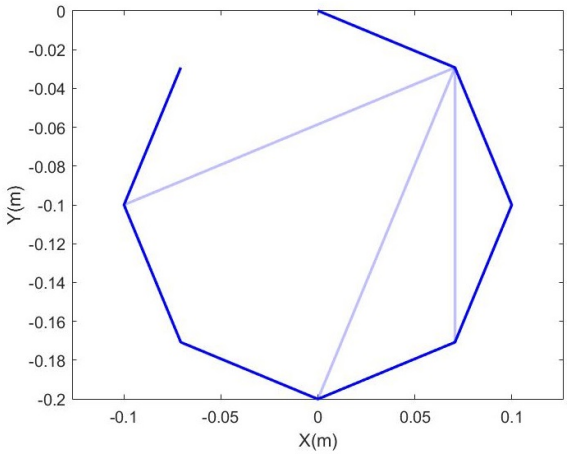


Figure 3.3 : Pose-graph added 3 measurements

complexity is reduced to $O(L_p^3)$, where L_p is the look-ahead step which is usually constant and independent of the pose number n_p. In future, we will try to apply the rAMDL method in the computation of these bounds to further improve the efficiency. This is one of our future research directions.

3.7 Simulation results

In this section, we validate the correctness of our conclusions and evaluate the performance of the bounds on a variety of synchronization problems. These experiments are based on a variety of 2D/3D rigid body motion synchronization problems drawn from pose-graph SLAM. The datasets and the estimation results in these simulation results are obtained based on the open source of the literature [116].

All of the following experiments are performed on a HP EliteDesk 800 G2 desktop with an Intel Core i5-6500 3.20 GHz processor and 8 GB of RAM running Windows 10 Enterprise. Our experimental implementations are written in MATLAB R2016a.

3.7.1 Efficency of the metrics

In this experiment, we verify the results in Section 3.6.1 and 3.6.3, which show that the T-optimality metric has a strong and direct relationship with the total node degree of the pose-graph and the D-optimality metric has a better performance than the T-optimality metric. We use a small example with eight nodes to simulate their relation. A regular octagon is constructed based on these eight poses. The variances of the translation δ_{ij}^2 and the concentrations κ_{ij} of all measurements are 1×10^{-4} and 5×10^3. The initial pose graph only includes the odometer without the additional measurements, as shown in Figure 3.2. Then, we begin to add the measurements gradually from one edge to 21 edges without repetition. It is noted that because the first pose is the anchor, the additional edges are not allowed to connected with the first pose. For example in Figure 3.3, we add 3 additional measurements. Then, for every graph, we can estimate the trace function and the log-determinant function of the FIM after using the SE-sync. The results (Black points) and their corresponding total node degree (Red Line) are shown in Figure 3.4 and 3.5.

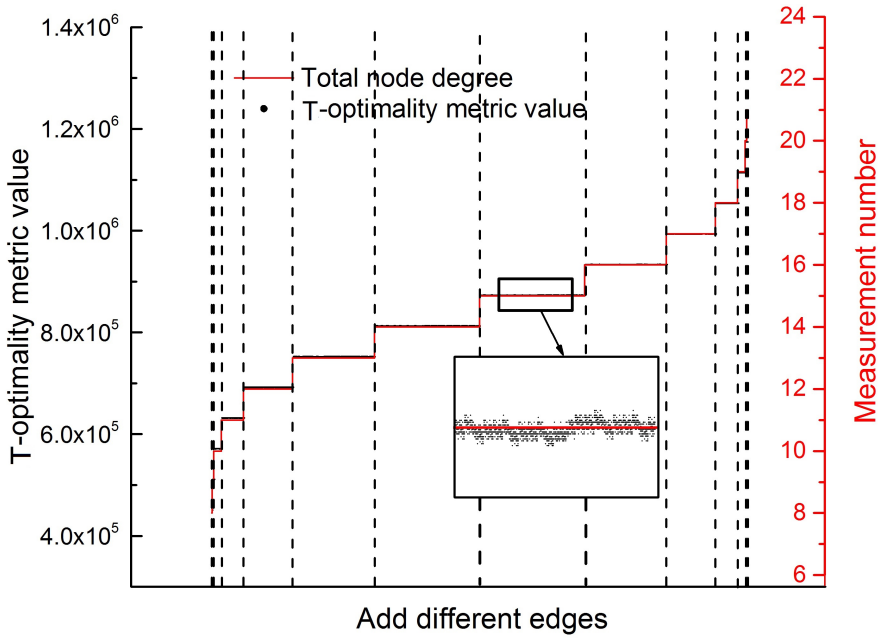


Figure 3.4 : Direct relationship between T-optimality metric with total node degree

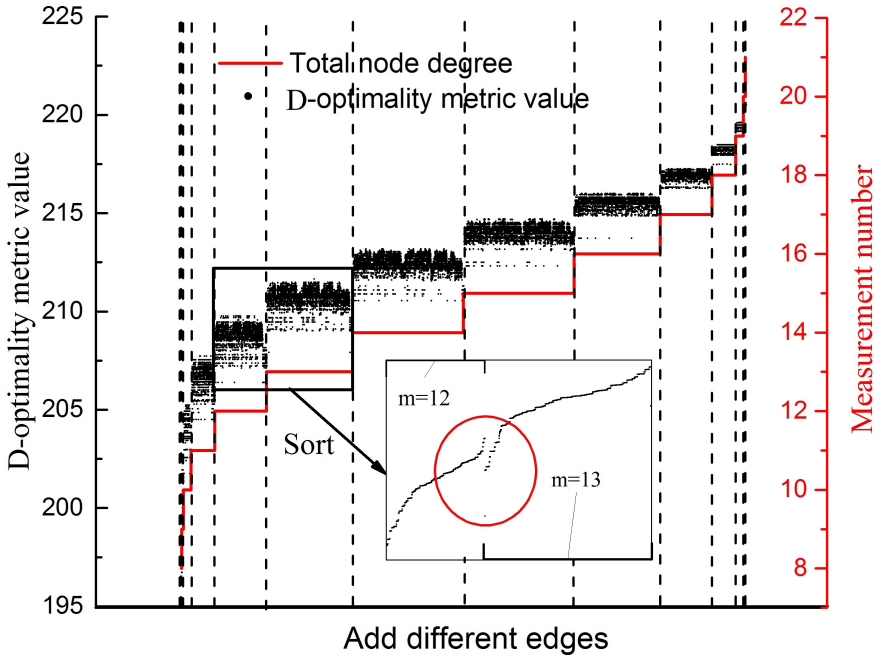


Figure 3.5 : Part relationship between D-optimality metric with total node degree

In Figure 3.4, because the term $\sum_{i=1}^{n_p} \sum_{j \in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{p}_{ij}\|^2$ is relatively small, we can find that the T-optimality metric is almost proportional to the total node degree of the whole pose graph. The T-optimality metric values of the graphs, which have the same node degree, are similar. It may lead the obvious local minimal problem in its application. In Figure 3.5, we can learn that the D-optimality metric has a weaker connection with the total node degree compared with the T-optimality metric. We sort the D-optimality metric values of the graphs with $m = 12$ and $m = 13$ measurements. Pay attention to the red circle part. We find that the D-optimality metric values of some graphs with more measurements are worse than the ones of the graphs with less measurements. This situation is common for many real pose-graph SLAM problems.

3.7.2 T-/D-optimality metrics in active SLAM application

In this simulation, in order to further explore the applied efficiency of the T-/D-optimality metrics, we compare them in the active SLAM task, as shown in Fig. 3.6.

Based on 100 random simulations with different designed way-points, we find that the optimal one selected by the T-optimality metric is commonly the same as the one picked out by the D-optimality metric (77%). Fig. 3.6 shows a result of which black and blue lines are respectively selected by the D-optimality metric and the T-optimality metric. The black path will lead more loops than the blue path, which leads to the smaller covariance value of the poses (black line: 1.2497×10^{-3} m, blue line: 1.2701×10^{-3} m). However, we also find that, when we pick out top 5 paths which have the big T-optimality metrics in every simulation as a set Ξ, the optimal one selected by the D-optimality metric will belong to Ξ with a very high probability (100%, 100 simulations). The above results show that these two metrics have a strong positive correlation, but still have some differences. The D-optimality metric is better than the T-optimality metric in the uncertainty evaluation.

Even though, sometimes, the T-optimality metric has the poorer performance than the D-optimality metric, it can be computed very fast. We can only use the diagonal elements of the FIM to compute the T-optimality metric. For the D-optimality metric, we need to construct the whole FIM and then compute the log-determinant function, which will obviously introduce the high computational complexity. It respectively costs 5.3934s and 0.4398s to compute the D-optimality metric and the T-optimality metric of 20 candidate paths with 170 poses and about 5206 measurements. The main advantage of the T-optimality metric is the better computational efficiency. So in our current work [31], we use the T-optimality metric or the weighted node degree to deal with the large-scale search for rough candidate actions, and the D-optimality or its lower bound is applied for sophisticated search within a small elite group.

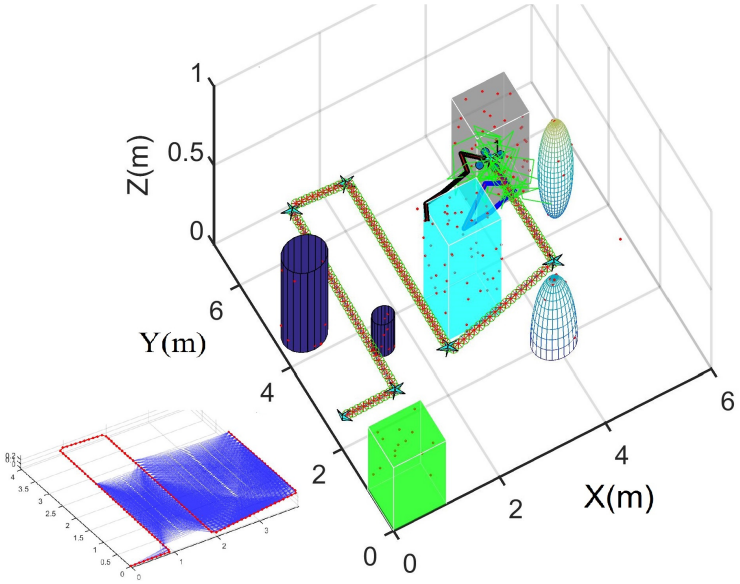


Figure 3.6 : Active SLAM task using two metrics. An unmanned aerial vehicle (UAV) moves from the first pentagram $(0, 2, 0.2)$, passes several pre-defined waypoints (blue pentagrams), and meanwhile performs the SLAM task in the whole process. The green circles, the red stars, and the red points are respectively the real UAV trajectory, the estimated trajectory, and the detected features. The features will be detected when they locate in the sensor range of the UAV and we can get the relative pose measurements based on the common features between two poses. Using SE-sync, the obtained pose graph with the red nodes and the blue edges is shown in the small left-down figure. Finally, when it reaches $(4, 4.5, 0.5)$, the UAV aims to select the future paths by evaluating the T-/D-optimality metrics. 20 random candidate paths (green lines) with the same number of the additional poses are generated and evaluated. The optimal paths based on different metrics are selected. The black and blue lines are the paths respectively selected by the D-optimality and T-optimality metrics.

3.7.3 Bound efficiency on D-optimality metric

In this set of experiments, we evaluate the bound efficiency and computational advantage on a variety of classical heterogeneous 3D pose-graph SLAM benchmarks. These datasets include: the synthetic datasets (3D: sphere, torus, and tiny/small/normal grid datasets) and the real-world datasets (2D: CSAIL, Intel Research Lab, manhattan (M3500), KITTI, city10000, and ais2klinik datasets; 3D: garage, cubicle, and rim datasets).

For the lower bound, the proposed Algorithm 1 is used to compute it. The largest

eigenvalue of the weighted Laplacian matrix is obtained using the QR decomposition with the same ordering shown in Algorithm 1. Results for these experiments are shown in Table 3.1 (2D) and Table 3.2 (3D). We demonstrate the number of the poses and the measurements, the log-determinant functions of the FIM, its upper and lower bounds, and their computational time.

Table 3.1 : Metric Results for the 2D SLAM datasets

			Original FIM		Upper bound		Lower bound	
Dataset	# Poses	# Measurements	Metric value	Time $[s]$	Value	Time $[s]$	Value	Time $[s]$
CSAIL	1045	1172	1.9860×10^{4}	2.3284×10^{-3}	1.9876×10^{4}	9.2391×10^{-4}	1.9858×10^{4}	3.6712×10^{-3}
Intel	1728	2512	3.0194×10^{4}	6.0744×10^{-3}	3.0352×10^{4}	4.0055×10^{-3}	3.0155×10^{4}	2.0743×10^{-3}
manhattan	3500	5453	7.1124×10^{4}	3.3530×10^{-1}	7.1193×10^{4}	7.3602×10^{-3}	7.1104×10^{4}	4.5105×10^{-3}
KITTI	4541	4677	1.1697×10^{5}	8.9039×10^{-3}	1.1703×10^{5}	4.6468×10^{-3}	1.1695×10^{5}	3.1092×10^{-3}
city10000	10000	20687	1.6667×10^{5}	9.0052×10^{-2}	1.6816×10^{5}	2.4292×10^{-2}	1.6520×10^{5}	2.0664×10^{-2}
ais2klinik	15115	16727	2.3642×10^{5}	3.6301×10^{-2}	2.3975×10^{5}	1.4693×10^{-2}	2.3567×10^{5}	1.3116×10^{-2}

Table 3.2 : Metric Results for the 3D SLAM datasets

Dataset	# Poses	# Measurements	Original FIM		Upper bound		Lower bound	
			Metric value	Time $[s]$	Value	Time $[s]$	Value	Time $[s]$
tiny-grid	9	11	2.4562×10^{2}	2.1558×10^{-4}	2.7561×10^{2}	2.9866×10^{-2}	2.4163×10^{2}	1.4115×10^{-4}
small-grid	125	297	4.4452×10^{3}	3.6373×10^{-3}	4.8488×10^{3}	3.1836×10^{-2}	4.3815×10^{3}	4.3020×10^{-4}
garage	1661	6275	2.0618×10^{4}	3.9511×10^{-2}	3.6618×10^{4}	3.6117×10^{-2}	1.5845×10^{4}	2.2989×10^{-3}
sphere	2500	4949	9.9105×10^{4}	7.4856×10^{-2}	1.0607×10^{5}	7.4023×10^{-2}	9.9054×10^{4}	6.7317×10^{-3}
torus	5000	9048	2.4986×10^{5}	1.9488×10^{-1}	2.6526×10^{5}	1.0156×10^{-1}	2.4985×10^{5}	1.6390×10^{-2}
cubicle	5750	16869	2.3729×10^{5}	2.1579×10^{-1}	3.1839×10^{5}	5.2156×10^{-2}	2.3685×10^{5}	1.4141×10^{-2}
grid	8000	22236	4.2613×10^{5}	3.2318×10^{0}	4.4902×10^{5}	1.5065×10^{-1}	4.2610×10^{5}	9.0280×10^{-2}
rim	10195	29743	4.7257×10^{5}	4.1626×10^{-1}	5.8622×10^{5}	6.5298×10^{-2}	4.7209×10^{5}	4.1260×10^{-2}

On each of these examples, the log-determinant function of the FIM is bounded within lb and ub correctly. We can also find that, the benefit from the great dimensionality reduction, the computation of the log-determinant function of the upper and lower bounds is much more expensive than the ones of the FIM. Because the dimension of the weighted Laplacian matrix is one-fourth (2D)/one-sixth (3D) of that of the full FIM, this great computational gap is sensible.

It is easy to find that, except the 'garage' dataset, the log-determinant function of the whole FIM gets closed to its lower bound (in 3%), because of the small $\sum_{i=1}^{n_p} \sum_{j \in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{p}_{ij}\|^2$. So in the real application of the D-optimality metric, such as the active pose-graph SLAM [131], we suggest using the tight lower bound to replace the original objective function.

Besides the computational time of obtaining these metrics, we also show the computational time of constructing the whole FIM, the weighted Laplacian matrix for $\mathbb{R}^2$ and $\mathbb{R}^3$ parts and the weighted Laplacian matrix for $SO(2)$ and $SO(3)$ part. They are used to compute the metrics. Table 4.3 and Table 4.4 show the computation time. The results show that the sparser weighted Laplacian matrices for the lower bound are easier to be constructed than the whole FIM.

Table 3.3 : Matrix constructing time for the 2D SLAM datasets

	Time $[s]$	
Dataset	Whole FIM	Weighted Laplacian matrices
CSAIL	8.8581×10^{-2}	1.5723×10^{-2}
Intel	1.3518×10^{-1}	1.9283×10^{-2}
manhattan	2.3982×10^{-1}	2.6794×10^{-2}
KITTI	2.2071×10^{-1}	2.5118×10^{-2}
city10000	7.3709×10^{-1}	6.3875×10^{-2}
ais2klinik	6.3033×10^{-1}	5.4222×10^{-2}

Table 3.4 : Matrix constructing time for the 3D SLAM datasets

	Time $[s]$	
Dataset	Whole FIM	Weighted Laplacian matrices
tiny-grid	7.1865×10^{-2}	2.0968×10^{-2}
small-grid	1.1677×10^{-1}	1.9557×10^{-2}
garage	9.6897×10^{-1}	5.3639×10^{-2}
sphere	8.3076×10^{-1}	4.8680×10^{-2}
torus	1.3884×10^{0}	7.3766×10^{-2}
cubicle	2.3300×10^{0}	1.1033×10^{-1}
grid	3.0566×10^{0}	1.1413×10^{-1}
rim	4.0647×10^{0}	1.7824×10^{-1}

3.7.4 Efficiency of CRLB

In this section, our main purpose is to validate that the CRLB is reachable using SE-sycn method (which belongs to maximum likelihood estimator (MLE)) [116]. These experiments are based on 'tiny-3Dgrid' and 'CSAIL' problems drawn from pose-graph SLAM [116].

In order to compute the covariance statistically, we need a ground truth and then sample noises to it. The ground truth is obtained by the optimization results of original datasets present in [116] using SE-sync. Then, we set these estimated poses as the ground truth. After obtaining the ground truth, the random noises obeying the isotropic Gaussian distribution and the isotropic Lagevin distribution are generated. We use 'normrnd' MATLAB function to generate the isotropic Gaussian distribution. For the isotropic Lagevin distribution, the noises are generated by the Acceptance-Rejection Method (ARM) [47]. Then we can add these noises into our relative measurements by the edge data using following equations:

$$\begin{aligned} \boldsymbol{H}_{ij} =& \boldsymbol{Z}_{ij}\boldsymbol{R}_j\boldsymbol{R}_i^{\top}, \boldsymbol{Z}_{ij} \sim Lang(\boldsymbol{I}_{n\times n}, k_c \cdot \kappa_{ij}) \\ \boldsymbol{p}_{ij} =& \boldsymbol{R}_i^{\top}(\boldsymbol{x}_j - \boldsymbol{x}_i) + \boldsymbol{y}_{ij}, \boldsymbol{y}_{ij} \sim \mathcal{N}(\boldsymbol{0}, {\delta_{ij}}^2 \boldsymbol{I}_{n\times n}) \end{aligned} \tag{3.40}$$

where k_c is the coefficient to determine the uncertainty level.

For an estimated result, we can compute the trace function of covariance by:

$$\text{trace}(\boldsymbol{C}) = \mathbb{E}\{\sum_{i=1}^{n_p} \left(\text{dist}(\boldsymbol{R}_i, \bar{\boldsymbol{R}}_i)^2 + \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_2^2\right)\} \tag{3.41}$$

where $\bar{\star}$ means the ground truth of $\star$. Every measurement dataset can generate a value using (3.41), then repeat many times to obtain the mathematical expectation $\mathbb{E}\{\text{trace}(\boldsymbol{C})\}$ by average. Finally, we can compute and compare the CRLB and the average mean squared error (MSE) for every pose.

For the 2D situation, we use the 'CSAIL' dataset to obtain the average MSE and CRLB. The initial κ_{ij} and δ_{ij}^{-2} are respectively set as 150 and 140. The coefficient k_c changes from 1 to 20. The simulations are repeated 100 times to get the covariance matrix. The simulation results are shown in Fig. 3.7.

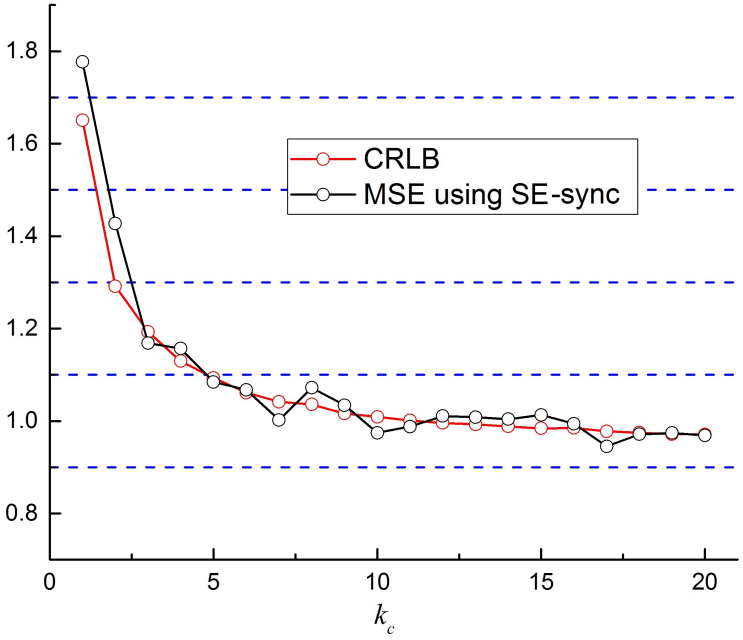


Figure 3.7 : Comparison of the CRLB with the mean squared error ($MSE = \frac{\text{trace}(\boldsymbol{C})}{n_p}$) of known estimators for the synchronization of $n_p = 1045$ poses with a complete measurement graph and one anchor. The SE-sync method appears to reach the CRLB.

For the 3D situation, we use the 'tiny-3Dgrid' dataset to obtain the covariance, CRLB with curvature terms and CRLB without curvature terms. κ_{ij}=12.5. Then the coefficient k_c changes from 1 to 20. Based on the Monte Carlo simulation, the processes are repeated 50 times. The simulation results are shown in Fig. 3.8. The

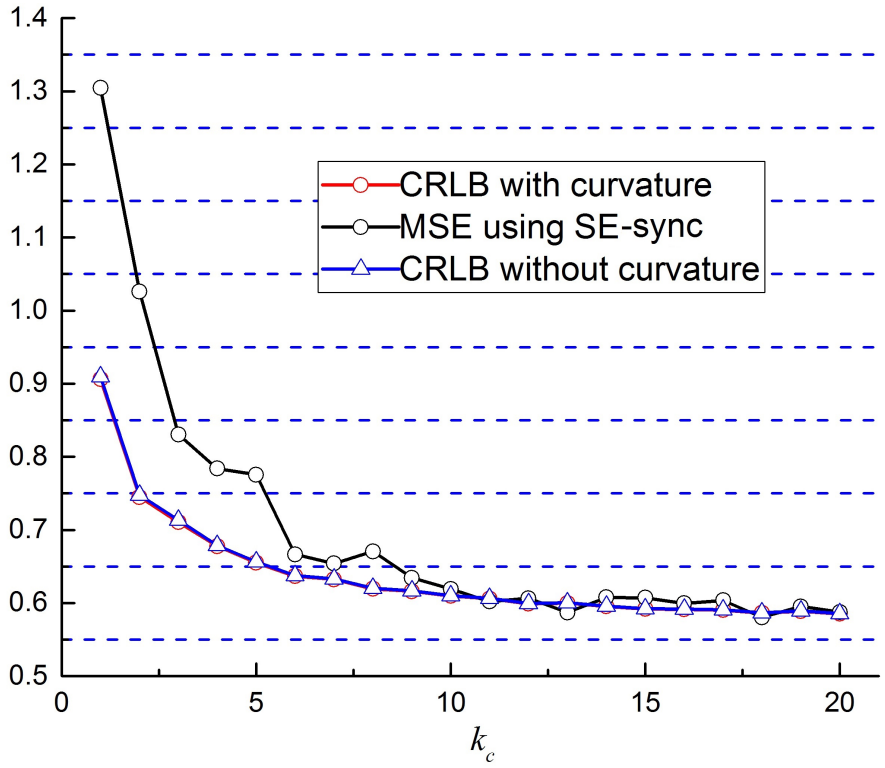


Figure 3.8 : Comparison of the CRLB with the mean squared error ($MSE = \frac{\text{trace}(\boldsymbol{C})}{n_p}$) of known estimators for the synchronization of $n_p = 9$ poses with a complete measurement graph and one anchor. The SE-sync method appears to reach the CRLB.

above two simulations show that the CRLB is reachable when the SNR is large enough.

3.8 Summary

In this chapter, based on the assumption of the isotropic Langevin noise for rotation and the block-isotropic Gaussian noise for translation, the FIM and CRLB of 2D/3D pose-graph SLAM, which are formulated as the synchronization on $\mathbb{R}^n \times$

$SO(n),\ n = 2, 3$, are derived and shown to be closely related to the weighted Laplacian matrix of pose-graph SLAM. Then, the TOED, including T-optimality and D-optimality are discussed. It shows that the T-optimality metric is almost directly determined by the total node degree, and the D-optimality design is equivalent to maximize the weighted number of the spanning trees of the pose-graph. We find that the T-optimality metric is cheaper than the D-optimality metric, but the D-optimality metric has a better performance. Furthermore, the lower and upper bounds of the D-optimality metric are presented, which are cheaper and can be used to replace the original metric. The experiment results, which verify our standpoints, are performed based on both the synthetic dataset and the real-world dataset.

Chapter 4

Anchor Selection based on Graph Topology and Sub-modular Optimization

As shown in Chapter 2, even though multiple kinds of SLAM methods, like ORB SLAM and SVO SLAM, have been presented and applied in some very challenging scenarios, compared with some outdoor and manual systems, like GPS and surveying and mapping systems, their robustness and accuracy may not be acceptable by recent applications, such as autonomous driving, autonomous robot manipulation system, and so on. Based on the limited sensors and estimation methods, it is really necessary to introduce some additional tools in the SLAM system to increase the whole information amount of the estimator. As shown in Fig. 4.7, using the Intel dataset, with the introduce of the anchored poses, the SLAM result becomes better and better. So our goal in this chapter aims to investigate the graph structure of SLAM as well as introduce and solve the anchor selection problem to improve the accuracy of the SLAM results. In this chapter, we propose an anchor selection algorithm based on graph topology and sub-modular optimization method.

4.1 Motivation and related work

In traditional SLAM, robot poses are typically estimated with respect to the robot starting position, which is regarded as the origin of the world coordinate frame. It is now well known that, as the robot travels away from the origin, errors in the estimation grows. Extent of error growth depends on the sensor range, the sensor accuracy, and the graph connectivity. However, if the true locations of some poses,

defined as anchors*, are available, the accuracy of the estimates can be significantly improved. Practically, this may be achieved in a number of ways although at some cost. In indoor environments, it may be possible to stop the robot at previously defined, known locations. Outdoors, it may be possible to pause for a sufficient length of time to obtain an accurate Global Positioning System (GPS) fix. Thus it might be necessary to use multiple anchors to improve the accuracy in SLAM especially when a highly accurate map is required. Clearly the number and the geometric distribution of the anchors have an impact on the accuracy of the resulting SLAM problem.

Naturally, the following question arises: given a set of poses that can possibly serve as anchors, how to select the best sub-set of anchors with given number that results in the most accurate SLAM result? In this chapter, this question is addressed in the context where the SLAM problem is formulated as a 2D/3D pose graph.

In recent works [78] and the previous chapter, the D-optimality metric of the FIM in 2D/3D pose-graph SLAM is shown to be closely related to weighted Laplacian matrix and the tree-connectivity of the pose-graph. Based on this conclusion, because of the lower computation complexity, instead of using the original D-optimality metric, the tree-connectivity metric is widely used in active SLAM [30, 68, 85], data exchange for coordinative SLAM (CSLAM) [128], and so on. How to select the edges of the measurement network of the SLAM problem to reduce the computational complexity and improve the accuracy of the result is described in [78] for 2D situations. In previous chapters, this result is extended to the 3D case. Inspired by [78], but unlike [78], the problem presented in this chapter requires selecting a set of anchors for a given measurement network, instead of selecting edges.

---

*anchor — Here an anchor means a pose with its global location and orientation known. Also different from sub-map joining method [64, 146], the anchored poses considered in this thesis have their global information, which brings more information to the whole SLAM system.

In this chapter, a sub-modular optimization framework to solve this anchor selection problem is proposed. Based on an approximation to the D-optimality metric, the anchor selection problem is transformed into a sub-matrix selection problem, which is proved to be a non-normalized non-monotone sub-modular optimization problem over a cardinality-fixed constraint. As a sub-modular function maximization problem, it is NP-hard. Then, the normal and random greedy methods are applied to obtain the near-optimal solution with a performance guarantee [87]. The bounds for the optimal value are obtained based on the obtained solution and Cauchy Interlacing Theorem [52]. Exploiting the matrix sparseness, a highly efficient lazy-greedy-based method, using Cholesky decomposition, approximate minimum degree permutation (AMDP), order re-use and rank-1 update technologies, is presented to quickly select the good potential anchored poses. Furthermore, some additional application scenarios, including the landmark setting in 2D/3D mapping and the CSLAM trajectory assignment with known initial positions, can use this anchor selection method. Finally, the simulations are presented to illustrate the proposed technique.

4.2 Anchor selection optimization problem

For the conventional SLAM problem, the first pose P_1 is anchored. The corresponding columns and rows of the FIM are deleted, which leads to a positive reduced matrix $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\{1\})$.

For the SLAM with multiple anchors, because the information for the known anchors with ground truths are infinite (without any uncertainty), the rows and columns of the FIM corresponding to the anchored poses are deleted, defined as $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a)$. It should be noted that, the final estimated poses $\boldsymbol{P'}$ will be different from the corresponding sub-part of the SLAM solutions $\boldsymbol{P}$ with only the first pose as the anchor. Meanwhile, with the same number of the anchors, the different

anchors also result in the different solutions. Hence, with the value changing of the obtained poses, the elements of the parts of the reduced FIM $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a)$, including $\triangle_w^{nD}, n = 2, 3$, ψ_i and $\boldsymbol{\Psi}_i$, are also changed, because they are dependent on the estimated poses $\{P_{i'_1},\ P_{i'_2},\ \cdots\}$. It means that $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a)$ is no longer a sub-matrix of the original matrix $\boldsymbol{\mathcal{I}}$. For the case with given set $\mathcal{N}_a$ of the anchors, in order to get its correct corresponding FIM $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a)$, the anchored SLAM problem is first solved using the iterative methods or the convex-based methods, and then Theorem 1 and Theorem 2 are applied by removing the rows and columns of the anchored poses.

In this part, we will formulate the anchor selection problem, prove its submodularity, and discuss its relations with edge selection and node-edge selection problems.

4.2.1 Optimization problem formulation

The purpose to select the anchored poses is to minimize the uncertainty of the SLAM result. Based on the D-optimality metric, given the fixed anchor number $|\mathcal{N}_a| = N$, the anchor selection problem is formulated as:

$$\begin{aligned} \underset{\mathcal{N}_a}{\text{maximize}} \quad & f_{cost}(\mathcal{N}_a) \triangleq \log(\det(\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a))) \\ \text{subject to} \quad & |\mathcal{N}_a| = N. \end{aligned} \tag{4.1}$$

Different anchors $\mathcal{N}_a$ will result in different SLAM solutions and different reduced FIMs $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a)$. Thus, to compute the accurate objective functions in (4.1) for a given candidate choice $\mathcal{N}_a$, we need to first obtain the estimated result by solving the whole pose-graph optimization process, then use the result to compute its corresponding reduced FIM, and finally get the objective function value $f_{cost}(\mathcal{N}_a)$. In solving general combinatorial optimization problems, the evaluation of the objective functions needs to be performed many times and is one of the most computational-

costly steps. Hence, it is computational unacceptable for directly solving the anchor selection based on the formulation (4.1).

Therefore, if we can find a way to avoid the complexity involved in computing the pose-graph SLAM solution, the computation of the objective function value of the anchor selection problem can be much faster and become computationally solvable. The following proposition shows that the D-optimality metric is close to the tree-connectivity of the pose-graph, which indicates that the graph structure of the SLAM measurement network plays a key role in evaluating the uncertainty of the SLAM result. Meanwhile, we note that the tree-connectivity of the pose-graph is independent of the computing of the SLAM solution, so we aim to replace the original D-optimality objective function with the tree-connectivity in order to avoid solving the pose-graph optimization problem when evaluating the objective function.

Based on Theorem 7, when $\delta/\lambda_1 \to 0^+$, $\log(\det(\boldsymbol{\mathcal{I}}))$ will get close to $\log(\det(\boldsymbol{L}_w^{\mathbb{R}^n}))$ $+\log(\det(\boldsymbol{L}_w^{SO(n)}))$. From [33, 78], in many real-world pose graph SLAM problems, the translation part δ is relatively small. So, we can approximate the anchor selection problem (4.1) by the following problem:

$$\begin{aligned} \underset{\mathcal{N}_a}{\text{maximize}} \quad & f_{obj}(\mathcal{N}_a) \triangleq \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a))) \\ \text{subject to} \quad & |\mathcal{N}_a| = N, \end{aligned} \tag{4.2}$$

where $\boldsymbol{\mathcal{L}} = \text{diag}\{\boldsymbol{L}_{w_{\mathbb{R}}} \otimes \boldsymbol{I}_{n\times n}, \boldsymbol{L}_{w_{SO(n)}} \otimes \boldsymbol{I}_{d\times d}\}$, thus $f_{obj}(\mathcal{N}_a) = n\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\mathcal{N}_a)))+$ $d\log(\det(\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\mathcal{N}_a)))$.

It is important to notice that the new problem (4.2) is independent of the final SLAM solution. So we can compute the objective function of the anchor selection problem without performing the SLAM optimization process.

4.2.2 Cardinality-fixed sub-modular optimization

In this part, our goal is to prove the sub-modularity of the approximate formulation (4.2). Before this, let's consider:

Definition 1. *[78] Let $\mathcal{T}_{\mathcal{G}}$ be the set of all spanning trees of $\mathcal{G}$. The weighted number $t_w(\mathcal{G})$ of $\mathcal{G}$, called tree-connectivity [78], is defined as:*

$$t_w(\mathcal{G}) \triangleq \sum_{\mathcal{T}\in\mathcal{T}_{\mathcal{G}}} \mathbb{V}(\mathcal{T}),\ \mathbb{V}(\mathcal{T}) = \prod_{e\in\mathcal{E}(\mathcal{T})} w(e), \tag{4.3}$$

where $w(e)$ means the weighted value of the edge e, $\mathbb{V}(\mathcal{T}):\ \mathcal{T}\to\mathbb{R}_+$ is the value of a spanning tree $\mathcal{T}\in\mathcal{T}_{\mathcal{G}}$, $\mathcal{E}(\mathcal{T})$ represents the set of all edges in $\mathcal{T}$.

A theorem about the tree-connectivity is as follows:

Theorem 5. *(Weighted Matrix-Tree Theorem [78]). For $\mathcal{G}=(\mathcal{V},\ \mathcal{E},\ w)$ with $w:\ \mathcal{E}\to\mathbb{R}^+$, we have $t_w(\mathcal{G})=\det(\boldsymbol{\mathcal{L}}_w^{\mathcal{G}})$, where $\boldsymbol{\mathcal{L}}_w^{\mathcal{G}}$ is the reduced weighted Laplacian matrix of $\mathcal{G}$ removing one row and one corresponding column.*

For the selection of the only one anchored pose, we have the following theorem:

Theorem 6. *If only one pose is anchored, the objective function in* (4.2) *is independent of the choice of the anchored pose.*

Proof. Assuming the i-th pose or the j-th pose is anchored, the objective function in (4.2) is decided by the reduced weighted Laplacian matrices $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\{i\})$, $\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\{i\})$, $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\{j\})$, and $\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\{j\})$. Based on Theorem 5, the determinant of the reduced weighted Laplacian matrix with one anchor is equal to the weighted number of spanning trees of its corresponding weighted graph (tree-connectivity) [28]. Because the weighted pose graph is the same, the weighted number of its spanning trees is constant. So we have: $\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\{i\}))=\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\{j\}))$ and $\det(\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\{i\}))=\det(\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\{j\}))$. The proof is completed. □

Without loss of generality, for the multiple anchor selection problem, we can first anchor the first pose P_1 (similar to the classical SLAM problems), and then choose the rest anchors. Then, we have the following theorem using some preliminaries about the sub-modularity shown in Appendix A.2.

Theorem 7. *When more than one anchor is chosen, the optimization problem* (4.2) *is a non-negative non-normalized non-monotone sub-modular optimization with a cardinality-fixed constraint.*

Proof. See Appendix B.2.1. □

Remark 2. *Benefiting from the approximation using graph topology from the optimization problem* (4.1) *to the problem* (4.2)*, Theorem 7 is related to the sub-modular property of the entropy maximization problem for selecting the most informative subset from a set of correlated random variables in [28]. However, in [28], the random variables correspond to a linear Gaussian estimation problem, such as linear sensor network and compass-SLAM [88, 78], and is essentially different from the common highly-nonlinear SLAM problem. Besides the differences of the focused problem setting, the anchor selection problem uses the anchored poses as the variables and needs to operate multiple rows and columns, instead of one row and one column. Theorem 7 is also related to the related work [76] using a determinant function, instead of using the log-determinant function, for the sub-matrix selection of the real symmetric positive semidefinite matrix, which shows the multiplicative sub-modular, instead of classical sub-modular, and it is applied in the problem related to the complete graph without connecting with the estimation problems. Besides, compared with these two references [28, 76], our anchor selection is not a monotone increasing optimization.*

4.2.3 Discussion on the connections and differences between anchor (nodes) selection, edge selection, and node-edge selection

The approach in this paper is inspired by the novel edge selection reference [78] and its related application [50]. The k edge selection problem (k-ESP) aims to select the optimal edges to be added/deleted for a given pose graph with fixed nodes to maximize the information gain or minimize the information lost. They also use the tree-connectivity to replace the D-optimality design metric of the FIM as the objective function. However, there are some important differences between the anchor selection and the edge selection problems.

The key difference is in the information aspect. The edge selection is similar to the information marginalization, which means only the local relative information from measurements is added/removed. However, in the anchor selection, the global information of the anchored poses is added. As shown in Fig. 4.1, we consider a base pose graph and one factor $\frac{1}{2\delta_{ij}^2}\|\boldsymbol{p}_{ij} - \boldsymbol{R}_i^\top(\boldsymbol{x}_j - \boldsymbol{x}_i)\|^2$ in the objective function. For simplification, only the weighted Laplacian matrix corresponding to the translation part $\boldsymbol{L}_{w_\mathbb{R}} \otimes \boldsymbol{I}_{n\times n}$, which is a sub-matrix of $\boldsymbol{\mathcal{L}}$ in (4.2), is considered. Figures in the first row show the generating process from the incident matrix and the diagonal matrix with weights to $\boldsymbol{L}_{w_\mathbb{R}} \otimes \boldsymbol{I}_{n\times n}$ corresponding to the considered factor. For the anchor selection method, even though pose P_j is regarded as the constant value and removed from the state vector, the measurement between P_i and P_j remains, and the uncertainty of pose P_i is still limited by this measurement (i, j).

Commonly, in edge selection [78], we do not add or delete the nodes. In [128], the edge selection is formulated as a data exchange problem in the loop closure detection of CSLAM. Considering its corresponding problem in our formulation, it means to select all the edges connected to those poses by adding some additional measurements to these edges but without anchoring the poses themselves. Similar to

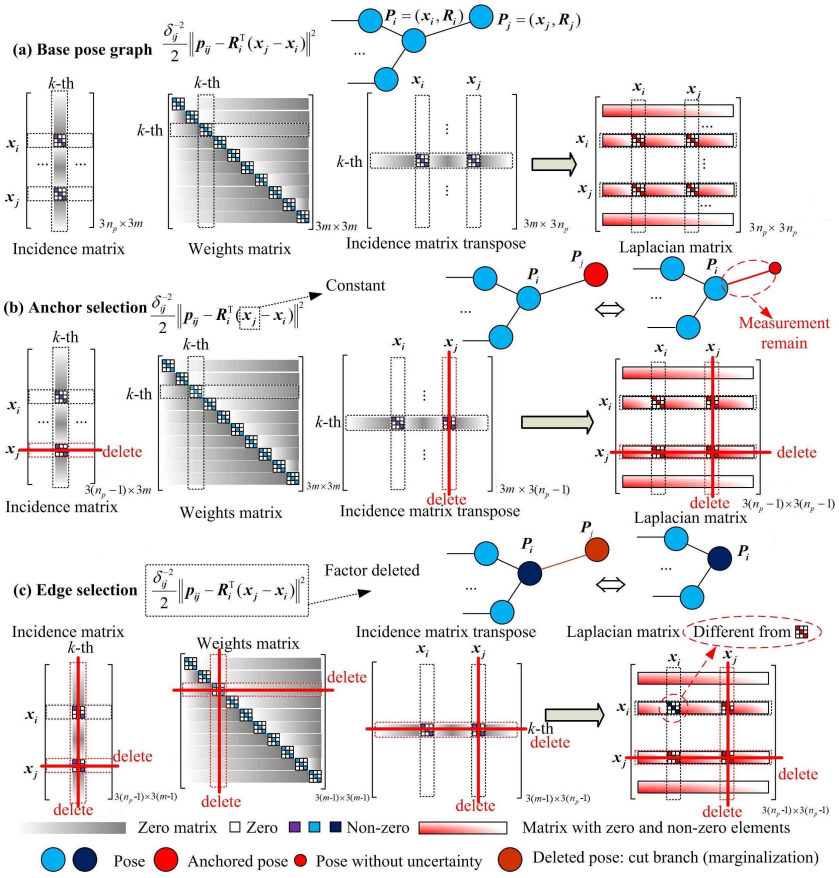


Figure 4.1 : The differences of changing the reduced weighted Laplacian matrices (translation only) calculated by incidence matrix and weights diagonal matrix: (a) Base pose graph; (b) Anchor selection: the anchored pose P_j is deleted; (c) Edge selection: both pose P_j and the edge (i, j) are deleted, which also resulting in a different P_i block in the weighted Laplacian matrix.

the common edge selection problem, the main difference between our anchor selection and the data exchange problem is also from the information aspect. The added measurements will make the connections between these poses and their connected poses stronger, which greatly strengthens the local structure of the pose graph. In other words, the uncertainty levels of these poses connected by these additional edges trend to converge and these poses can be considered as a local unit from the information point of view. However, because no global information is added, their uncertainty level may be still large if the other poses in the whole pose graph are weakly connected with this new local unit. The impact of this local unit on the

global pose graph is limited (Fig. 4.2). In some special situations, if edge selection results in the deletion of one pose P_j, which corresponds to deleting all the edges connected to this pose, then this situation is still different from anchoring pose P_j. This is because some useful information from the measurements (i, j) are marginalized, and the uncertainty of pose P_i increases. In this case, P_j is not estimated which is different from the case of anchoring P_j where the ground truth of P_j is known. This operation in edge selection may cause the sparser pose graph structure, as shown in Fig. 4.3.

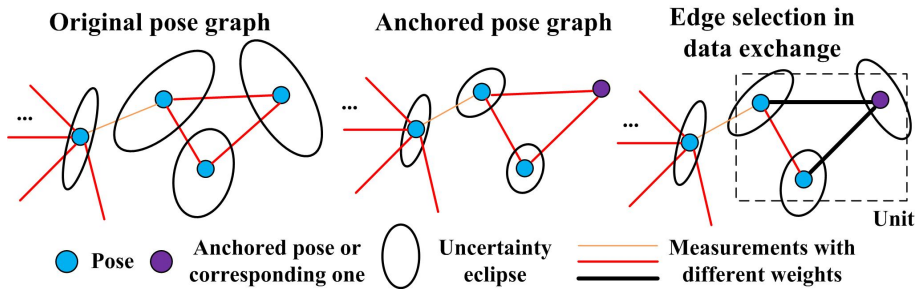


Figure 4.2 : The first and second figures are the original pose graph and the anchored pose graph with one anchor. The last figure is the corresponding edge selection case shown in data exchange [128]. For the last figure, because the measurements connected to the purple pose become very strong, the accuracy of the local poses will improve. The three poses can be considered as a local unit. Their accuracy will be decided by the other poses connect to this local unit. Especially when the local structure is weakly connected with other poses, no matter how accurate the added measurements (black measurements with large weights) are, the pose estimate accuracy is still poor and limited.

Meanwhile, because of using different variables in the optimization, the details of the problem formulation are also different. Commonly, in most applications, like sensor scheduling [78, 21], the candidate set for the edge selection is limited to make sure that no poses are deleted/added, which means the dimension of the state vector corresponding to the poses keeps as constant and only the values of the FIM corresponding to the poses connected the selected edge are changed. However, for the anchor selection problem, the dimension of the FIM will certainly reduce. The edge selection problem discussed in[78] is proved to be a normalized, monotone increasing

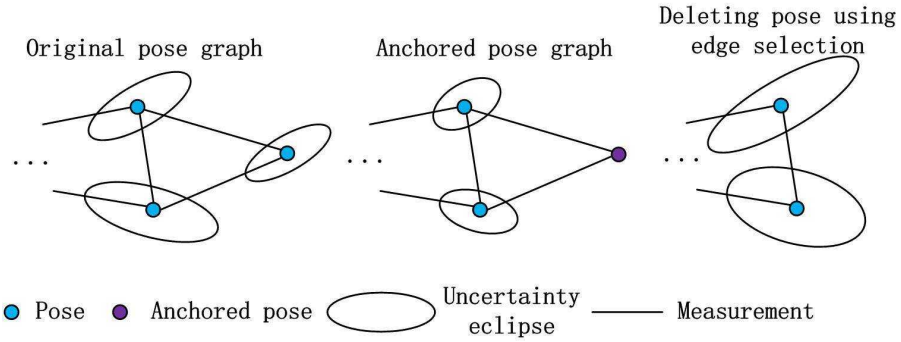


Figure 4.3 : The first figure is the original pose graph. The second figure is the anchored pose graph with an anchor in purple. Because the uncertainty of the purple anchor is reduced to zero, the connected measurements limit the related poses and the uncertainty of these poses also reduces. The last figure is to delete a pose using edge selection, which means to delete all the edges connected to a pose. Because the purple pose does not need to be estimated, the related measurements are also ignored in the pose graph. With the sparser pose-graph structure, the accuracy of the rest poses reduces.

sub-modular maximization problem, which shows different properties compared with ours. More details of the edge selection problem are shown in [78, 33].

Some other works such as [35] focus on the node-edge selection problem related to active SLAM. In active SLAM, the additional trajectory, which introduces the new poses and measurements to the base pose graph, is selected optimally to reduce the uncertainty of the SLAM solution. Similar to the edge selection problem and different from the anchor selection problem, it does not introduce any global ground truth information. Moreover, because of introducing more poses, the dimension of the FIM will become larger, which is different from the ones corresponding to edge selection (reduce or keep dimension) and anchor selection (reduce dimension).

For the conventional SLAM problem, the first pose $(\boldsymbol{x}_0, \boldsymbol{R}_0)$ is anchored. The corresponding columns and rows of the FIM are deleted, which leads to a positive reduced matrix $\boldsymbol{\mathcal{I}}_{nD}$.

The multiple anchored poses are defined as $\mathcal{N}_a = \{P_{i'}\} \in \mathcal{V}$. $\mathcal{V} \backslash \mathcal{N}_a$ means that the pose subset $\mathcal{N}_a$ is deleted from the whole pose set $\mathcal{V}$. For the SLAM with multiple

anchors, the anchored poses are known and their corresponding rows and columns of the FIM are deleted, defined as $\boldsymbol{\mathcal{I}}_{nD}(\mathcal{V}\backslash\mathcal{N}_a)$. It is noted that, with the changing of the finial estimated solution using different number of the anchored poses, the elements of the rest part of the FIM $\boldsymbol{\mathcal{I}}_{nD}(\mathcal{V}\backslash\mathcal{N}_a)$, including $\triangle_w^{nD}, n = 2, 3$, ψ_i and $\boldsymbol{\Psi}_i$, are also changed, because they are relative to the estimated poses P_i. It means that $\boldsymbol{\mathcal{I}}_{nD}(\mathcal{V}\backslash\mathcal{N}_a)$ is no longer a sub-matrix of the original matrix $\boldsymbol{\mathcal{I}}_{nD}$.

With the number of the anchored poses increasing, the SLAM results will become more accurate. However, the selection of the anchored poses will greatly affect the final result. It seems that the objective function of this anchor selection problem includes a state estimation sub-problem, which makes this problem hard to solve. This chapter aims to find the best anchored poses for the SLAM system given the cardinality of $\mathcal{N}_a$.

4.3 Greedy methods

As a non-normalized non-monotone sub-modular maximization problem with a cardinality-fixed constraint, the greedy optimization method is commonly used. There are several kinds of greedy methods with different performances. We will present them and discuss about the near optimal performance of the final solution.

4.3.1 Natural greedy algorithm

A natural greedy algorithm for this problem starts with an empty set, and then an element is added to maximize the marginal gain in the k-th iteration. A formal statement of the 1-step greedy algorithm is given as Algorithm 2 [14].

At the k-th iteration, using the log-determinant values of $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^3}$ and $\widehat{\boldsymbol{L}}_w^{SO(3)}$, the computational complexity of computing the objective function value $f_{obj}(\mathcal{N}_a \cup \{s_{n_p-k}\})$ is $O((n_p - k + 1)^3)$. The computational complexity of the whole greedy method will

Algorithm 2 1-step Greedy algorithm

Require: The new weighted Laplacian matrix $\boldsymbol{\mathcal{L}}$; the fixed anchored number N
Ensure: The sub-optimal anchored poses set $\mathcal{N}_a$

$\mathcal{N}_a \leftarrow \emptyset$;
while $|\mathcal{N}_a| \leq N$ **do** ▷ Cardinality-fixed constraint
  $s^*_{n_p-k} = \max_{s_{n_p-k} \in \mathcal{V} \setminus \mathcal{N}_a} f_{obj}(\mathcal{N}_a \cup \{s_{n_p-k}\})$;
  $\mathcal{N}_a \leftarrow \mathcal{N}_a \cup \{s^*_{n_p-k}\}$;
  $k \leftarrow k + 1$;
return $\mathcal{N}_a$ ▷ Selected set with good performance

be $O((n_p - N + 1)^4 + \cdots + (n_p)^4)$, which can be simplified as:

$$\begin{aligned} & O((n_p - N + 1)^4 + \cdots + (n_p)^4) \\ =& O\left((1^4 + \cdots + (n_p)^4) - (1^4 + \cdots + (n_p - N)^4)\right) \\ =& O(f_o(n_p) - f_o(n_p - N)), \end{aligned} \tag{4.4}$$

where $f_o(\star) = \frac{\star(\star+1)(6\star^3+9\star^2+\star-1)}{2}$, we can get $O(f_o(n_p) - f_o(n_p - N)) \approx O(6N^5 - 30N^4 n_p + 60N^3 n_p^2 - 60N^2 n_p^3 + 30N n_p^4)$. When $n_p >> N$, we can get that the computational complexity will be about $O\left(30N n_p^4\right)$.

4.3.2 Random greedy algorithm and continuous-double-greedy

Because $f_{obj}(\mathcal{N}_a)$ is a non-monotone sub-modular optimization function, the natural greedy algorithm does not have the performance guarantee. The combination of the random greedy method and continuous-double-greedy has the performance guarantee for non-monotone functions [15].

Random greedy algorithm

The random-greedy algorithm is very similar to the natural greedy algorithm [51]. The main difference is to randomly select among the top N highest-scoring elements, instead of selecting the best one. The specific random greedy algorithm

is shown in Algorithm 3.

Algorithm 3 Random greedy algorithm

Require: The weighted Laplacian matrix $\boldsymbol{\mathcal{L}}$; the fixed anchored number N
Ensure: The best anchored poses set $\mathcal{N}_a$

$\mathcal{N}_a \leftarrow \emptyset$;
while $|\mathcal{N}_a| \leq N$ **do** ▷ Cardinality-fixed constraint
  Select a subset $\mathcal{M}_k^*$ from $\mathcal{V} \setminus \mathcal{N}_a$ satisfying $\mathcal{M}_k^* = \max_{\mathcal{M}_k} \sum_{s_{n_p-k} \in \mathcal{M}_k, |\mathcal{M}_k|=N} f_{obj}(\mathcal{N}_a \cup \{s_{n_p-k}\})$;
  $s_{n_p-k}^* \leftarrow Uniform(\mathcal{M}_k^*)$ ▷ Randomly select one element from $\mathcal{M}_k^*$ following uniform distribution
  $\mathcal{N}_a \leftarrow \mathcal{N}_a \cup \{s_{n_p-k}^*\}$;
  $k \leftarrow k + 1$;
return $\mathcal{N}_a$ ▷ Selected set with good performance

The computational complexity of the random greedy algorithm is similar to the natural/lazy greedy algorithm. In fact, this simple algorithm might be considered a natural substitute for the classical greedy algorithm. For the cardinality-fixed monotone increasing sub-modular optimization problem, it retains the same tight guarantee with the natural/lazy one. For the non-monotone sub-modular optimization problem, it can give an approximation $(1 - \frac{N}{en_p})/e - \varepsilon$ *approximation*, where ε is an arbitrarily small positive constant, however the classical/lazy greedy algorithm does not have any performance guarantee.

Discussion on the selection of random greedy algorithm and continuous-double-greedy method

It is well-known that, for a non-monotone sub-modular optimization method, the combination using the best solution of the random greedy algorithm and continuous-double-greedy method can reach an approximation factor of:

$$\max \left\{ \frac{1 - \frac{N}{en_p}}{e} - \varepsilon, \left(1 + \frac{n_p}{2\sqrt{(n_p - N)N}}\right)^{-1} - o(1) \right\}, \tag{4.5}$$

where ε is an arbitrarily small positive constant [51].

For the general anchor selection problem, the number of the anchored poses N is relatively small, compared with the total number of poses n_p (usually lager than 10^3). Based on the approximation factor Eq.(4.5), we can find that, when $N/n_p \to 0^+$, $\frac{1-\frac{N}{en_p}}{e} - \varepsilon$ will be bigger than $\left(1 + \frac{n_p}{2\sqrt{(n_p-N)N}}\right)^{-1} - o(1)$, which means that the random-greedy algorithm has a better guarantee in the low N situation. So in this chapter, we only use the random greedy algorithm instead of the combination of two algorithms. In order to verify this choice, a simulation with $n_p = 1000$ poses to show the approximation guarantees changing with the number of anchored poses is illustrated as follows:

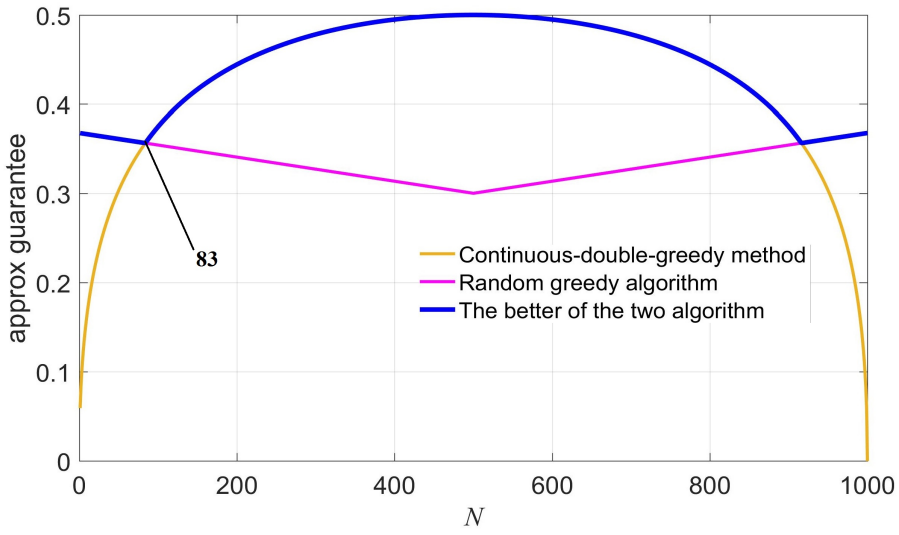


Figure 4.4 : Approximation guarantees of random greedy algorithm and continuous-double-greedy method

In Fig. 4.4, we can see that when $N \leq 83$, the random greedy algorithm can get the better approximation guarantee.

4.3.3 Discussion of near optimal performance

In this subsection, we will discuss the performances of the greedy method and random greedy method. Because of the sub-modularity and Cauchy Interlacing

Theorem, in fact, the optimal solution of this optimization problem is limited by some bounds.

Lemma 2. *The objective function $f_{obj}(\mathcal{N}_a)$ has a natural bound without considering the optimization method:*

$$\begin{aligned} \mathcal{F}_{LB} \leq f_{obj}(\mathcal{N}_a) \leq \mathcal{F}_{UB} \\ \mathcal{F}_{LB} = n\sum_{1}^{n_p-N}\log\lambda_i(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V})) + d\sum_{1}^{n_p-N}\log\lambda_i(\widehat{\boldsymbol{L}}_w^{SO(n)}(\mathcal{V})) \\ \mathcal{F}_{UB} = n\sum_{N+1}^{n_p}\log\lambda_i(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V})) + d\sum_{N+1}^{n_p}\log\lambda_i(\widehat{\boldsymbol{L}}_w^{SO(n)}(\mathcal{V})), \end{aligned} \tag{4.6}$$

where $d = \frac{n(n-1)}{2}$, the eigenvalues of the $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V})$ and $\widehat{\boldsymbol{L}}_w^{SO(n)}(\mathcal{V})$ are respectively sorted by the increased order and named $\lambda_1(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V})), \cdots, \lambda_{n_p}(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}))$ and $\lambda_1(\widehat{\boldsymbol{L}}_w^{SO(n)}(\mathcal{V}))$, $\cdots, \lambda_{n_p}(\widehat{\boldsymbol{L}}_w^{SO(n)}(\mathcal{V}))$.

Proof. The proof is given in Appendix B.2.2. □

Corollary 2. *The objective function $f_{obj}(\mathcal{N}_a)$ has a natural bound without considering the optimization method:*

$$\sum_{1}^{n_d(n_p-N)}\log\lambda_i(\boldsymbol{\mathcal{L}}) \leq f_{obj}(\mathcal{N}_a) \leq \sum_{n_dN+1}^{n_dn_p}\log\lambda_i(\boldsymbol{\mathcal{L}}), \tag{4.7}$$

where $n_d = n + d$, $n_p \geq 1$, the eigenvalues of the $\boldsymbol{\mathcal{L}}$ are sorted by the increased order and named $\lambda_1(\boldsymbol{\mathcal{L}}), \cdots, \lambda_{n_dn_p}(\boldsymbol{\mathcal{L}})$.

Proof. It can be directly obtained by Eq.(B.90). □

Based on this natural bound, when using the random greedy algorithm, we can get the tighter bound for the optimal solution:

Lemma 3. *Let $\mathcal{U}_{opt}$, $\mathcal{U}_{greedy}$ and $\mathcal{U}^*_{greedy}$ be the optimal value of (4.2), the objective function value obtained by the lazy greedy algorithm, and the mathematical*

expectation of the objective function value achieved by the random greedy algorithm, respectively. It holds:

$$\begin{aligned} \mathcal{U}_{LB} &\leq \mathcal{U}_{opt} \leq \mathcal{U}_{UB} \\ \mathcal{U}_{LB} &= \max\left\{\mathcal{U}_{greedy}, \mathcal{F}_{LB}\right\} \\ \mathcal{U}_{UB} &= \min\left\{\left(\frac{1-\frac{N}{en_p}}{e}-\varepsilon\right)^{-1}\mathcal{U}^{*}_{greedy}, \mathcal{F}_{UB}\right\}. \end{aligned} \tag{4.8}$$

Proof. The proof is given in Appendix B.2.3. □

Based on Lemma 3, we have the bounds to ensure the near-optimal performance of the solution. Because of the randomness, the natural greedy algorithm always has a better performance than the random greedy algorithm. So in practice, the lazy greedy algorithm is used to get a good result and meanwhile the random greedy algorithm is used to get the bound if it is required.

4.3.4 Algorithm speeded up by lazy evaluation, sparse Cholesky decomposition and order re-use

Sparse Cholesky decomposition

It is well known that to obtain the computational complex of the log-determinant function of a $n_m \times n_m$ dense matrix is $O(n_m^3)$. With the growth of n_m, its computational time becomes unacceptable for the general hardware system. Benefit from the sparse Cholesky decomposition, for the positive definite matrix, we have:

$$\log\det(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^2}(\mathcal{V}\backslash\mathcal{N}_a)) = \log\det(\boldsymbol{C}_1\boldsymbol{C}_1^{\top}) = 2\log\det(\boldsymbol{C}_1) = 2\sum_i \log(\boldsymbol{C}_1)_{i,i}, \tag{4.9}$$

where $\boldsymbol{C}_1$ is the lower triangular Cholesky factor of $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)$, $(\boldsymbol{C}_1)_{i,i}$ is the i-th diagonal element of $\boldsymbol{C}_1$.

Lazy evaluation

For many real-world pose-graph datasets, the number of poses n_p is about 10^3 to 10^5, so its computational complexity is unacceptable based on the natural greedy algorithm. Using the lazy evaluation, its computational process can speed up [87]. Instead of computing the objective function for each candidate element $s_{n_p-k} \in \mathcal{V} \setminus \mathcal{N}_a$, the lazy-greedy algorithm only needs to evaluate a part of them. It keeps the upper bounds of the candidate elements and sorts them in decreasing order. In each iteration, the lazy-greedy algorithm only needs to evaluate the element on top of the list and updates its upper bound. Because of the sub-modularity, for the objective function, $\Delta_k = f_{obj}(\mathcal{N}_a \cup \{s_{n_p-k}\}) - f_{obj}(\mathcal{N}_a)$ is the upper bound of the following functions $\Delta_{k+1} = f_{obj}((\mathcal{N}_a \cup \{s^*_{n_p-k}\}) \cup \{s_{n_p-k-1}\}) - f_{obj}(\mathcal{N}_a \cup \{s^*_{n_p-k}\})$, $\Delta_{k+2}, \cdots$. Many elements on the later part of the list do not need to be evaluated, because their upper bounds are too small.

Order re-use

In this process, in order to speed up the Cholesky decomposition process, the AMDP is used to permute the rows and columns of the symmetric sparse matrix $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)$. Meanwhile, because the structures of $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)$ and $\widehat{\boldsymbol{L}}_w^{SO(n)}(\mathcal{V}\backslash\mathcal{N}_a)$ are the similar, the order obtained by $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)$ can be also used for $\widehat{\boldsymbol{L}}_w^{SO(n)}(\mathcal{V}\backslash\mathcal{N}_a)$. Moreover, the AMDP can only be computed once for the initial matrix $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\{i\})$, and then the obtained order vector $\boldsymbol{p}$ can be re-used in all matrix using in this method by the following technology easily:

Assuming that after deleting i-th row and column of the matrix $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)$,

satisfying:

$$\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a) = \begin{bmatrix} \boldsymbol{L}^{1,1} & \boldsymbol{l}_1 & \boldsymbol{L}^{1,2} \\ \boldsymbol{l}_1^\top & l & \boldsymbol{l}_2 \\ \boldsymbol{L}^{2,1} & \boldsymbol{l}_2^\top & \boldsymbol{L}^{2,2} \end{bmatrix}, \tag{4.10}$$

$$\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\})) = \begin{bmatrix} \boldsymbol{L}^{1,1} & \boldsymbol{L}^{1,2} \\ \boldsymbol{L}^{2,1} & \boldsymbol{L}^{2,2} \end{bmatrix}.$$

Normally, we compute the log-determinant function $f_{obj}(\mathcal{N}_a \cup \{i\})$ after deleting the i-th row and column. However, by this way, because of the decrease of the matrix dimension, the order vector $\boldsymbol{p}$ is also changed, which makes the re-use operation complex. So we use the following matrix $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))^*$ to replace $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))$:

$$\log\det(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))^*) = \log\det(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))) + \log\det(1),$$

$$\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))^* = \begin{bmatrix} \boldsymbol{L}^{1,1} & \mathbf{0} & \boldsymbol{L}^{1,2} & \\ \mathbf{0} & 1 & \mathbf{0} & \}_{i-th} \\ \boldsymbol{L}^{2,1} & \underbrace{\mathbf{0}}_{i-\text{th}} & \boldsymbol{L}^{2,2} & \end{bmatrix}. \tag{4.11}$$

Using (4.11), the order vector $\boldsymbol{p}$ can be used for multiple times without additional operations, because of the constant matrix dimensions. The computational time of the AMDP will be saved. Meanwhile, benefiting from the order reuse, the rank-1 update can be used in the following sub-section.

Rank-1 update

Based on (4.11), let

$$\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a) = \begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{c}_{21} & c_{22} & \boldsymbol{0} \\ \boldsymbol{C}_{31} & \boldsymbol{c}_{32} & \boldsymbol{C}_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{C}_{11}^\top & \boldsymbol{c}_{21}^\top & \boldsymbol{C}_{31}^\top \\ \boldsymbol{0} & c_{22} & \boldsymbol{c}_{32}^\top \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C}_{33}^\top \end{bmatrix} \tag{4.12}$$

be the Cholesky factorization of the matrix $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)$ and let

$$\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))^* = \begin{bmatrix} \bar{\boldsymbol{C}}_{11} & \boldsymbol{0} & \boldsymbol{0} \\ \bar{\boldsymbol{c}}_{21} & \bar{c}_{22} & \boldsymbol{0} \\ \bar{\boldsymbol{C}}_{31} & \bar{\boldsymbol{c}}_{32} & \bar{\boldsymbol{C}}_{33} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{C}}_{11}^\top & \bar{\boldsymbol{c}}_{21}^\top & \bar{\boldsymbol{C}}_{31}^\top \\ \boldsymbol{0} & \bar{c}_{22} & \bar{\boldsymbol{c}}_{32}^\top \\ \boldsymbol{0} & \boldsymbol{0} & \bar{\boldsymbol{C}}_{33}^\top \end{bmatrix} \tag{4.13}$$

be the Cholesky factorization of the matrix $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash(\mathcal{N}_a\cup\{i\}))^*$, we have: $\boldsymbol{C}_{11} = \bar{\boldsymbol{C}}_{11}$, $\boldsymbol{C}_{31} = \bar{\boldsymbol{C}}_{31}$, $\bar{\boldsymbol{c}}_{21} = \boldsymbol{0}$, $\bar{c}_{22} = 1$; $\bar{\boldsymbol{c}}_{32} = \boldsymbol{0}$,

$$\begin{aligned} \boldsymbol{C}_{31}\boldsymbol{C}_{31}^\top + \boldsymbol{c}_{32}\boldsymbol{c}_{32}^\top + \boldsymbol{C}_{33}\boldsymbol{C}_{33}^\top &= \bar{\boldsymbol{C}}_{31}\bar{\boldsymbol{C}}_{31}^\top + \bar{\boldsymbol{C}}_{33}\bar{\boldsymbol{C}}_{33}^\top \\ \Rightarrow\ \bar{\boldsymbol{C}}_{33}\bar{\boldsymbol{C}}_{33}^\top &= \boldsymbol{c}_{32}\boldsymbol{c}_{32}^\top + \boldsymbol{C}_{33}\boldsymbol{C}_{33}^\top. \end{aligned} \tag{4.14}$$

Because $\boldsymbol{c}_{32}$ is a non-zero vector and $\boldsymbol{C}_{33}\boldsymbol{C}_{33}^\top$ is a full-rank positive matrix, we can find that (4.14) is a rank-1 update. To obtain $\bar{\boldsymbol{C}}_{33}$ belongs to rank-1 update of a sparse Cholesky factorization. So the objective function can be written as:

$$\log\det(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))^*) = \log\det(\underbrace{\boldsymbol{C}_{11}\boldsymbol{C}_{11}^\top}_{\text{Re-use}}) + \log\det(\overbrace{\underbrace{\boldsymbol{c}_{32}\boldsymbol{c}_{32}^\top}_{\text{Re-use}} + \underbrace{\boldsymbol{C}_{33}\boldsymbol{C}_{33}^\top}_{\text{Re-use}}}^{\text{Rank 1 update}}). \tag{4.15}$$

Thank to the order re-use, based on "CSparse" package, the whole rank-1 update can be finished very quickly without considering the row and column order.

4.3.5 Whole improved algorithm and its computational complexity analysis

Based on the above technologies, we can summarize our new improved method by the pseudo-code form in Algorithm 4 and 5.

The computational complexities of the original 1-step greedy algorithm and the random greedy algorithm have been discussed in Section 4.3.1 and 4.3.2. Due to the speed-up technologies in Section 4.3.4, the computational complexity of the improved method is reduced and shown as follows:

At the k-th iteration, because of the sparseness, the computational complexity of the rank1 update of the Cholesky factorization of the n_p-k+1 dimension matrix $\boldsymbol{c}_{32}\boldsymbol{c}_{32}^{\top}+\boldsymbol{C}_{33}\boldsymbol{C}_{33}^{\top}$ is generally smaller than $O((n_p-k+1)^2)$. Without considering the lazy evaluation, in each iteration, the matrix dimensions change from n_p-k+1 to 1. For one iteration, the whole computational complexity is $O((n_p-k+1)^2+\cdots+1^2)$. Assume that the efficient of the lazy evaluation can be simplified a coefficient: $\omega, \omega \in (0,1)$, the computational complexity of the whole method will be:

$$\begin{aligned}
&\sum_{k=2}^{N}\omega O((n_p-k+1)^2+\cdots+1^2)\\
=&\sum_{k=2}^{N}\omega O(\frac{((n_p-k+1)(n_p-k+2)(2n_p-2k+3))}{6})\\
\approx& O(\frac{\omega N}{3}n_p^3).
\end{aligned} \tag{4.16}$$

Compared with 1-step greedy algorithm, our new method is much faster than the original algorithm $O(\frac{\omega N}{3}n_p^3) << O\left(30Nn_p^4\right)$. This conclusion can be also verified by the numerical simulation in Section 4.5.2.

Algorithm 4 Improved lazy Greedy algorithm using sparse Cholesky decompositon, order re-use and rank-1 update

Require: Two weighted Laplacian matrices $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^3}$ and $\widehat{\boldsymbol{L}}_w^{SO(3)}$; the fixed anchored number N

Ensure: The sub-optimal anchored poses set $\mathcal{N}_a$

//Based on Lemma 6

$\mathcal{N}_a \leftarrow \{1\}$;

//Choose a fill-reducing permutation heuristic P

$\boldsymbol{p} \leftarrow COLAMD(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^3}(\mathcal{V}\backslash\mathcal{N}_a))$; ▷ e.g., Column approximate minimum degree

//Sparse Cholesky factor based $\boldsymbol{p}$

$\boldsymbol{C}_1 \leftarrow SparseCholesky\left(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)(\boldsymbol{p},\boldsymbol{p})\right)$;

$\boldsymbol{C}_2 \leftarrow SparseCholesky\left(\widehat{\boldsymbol{L}}_w^{SO(3)}(\mathcal{V}\backslash\mathcal{N}_a)(\boldsymbol{p},\boldsymbol{p})\right)$;

$f_{obj}(\mathcal{N}_a) \leftarrow 2\cdot n\cdot\sum_i \log(\boldsymbol{C}_1)_{i,i} + 2\cdot d\cdot\sum_i \log(\boldsymbol{C}_2)_{i,i}$;

$f_{value} \leftarrow f_{obj}(\mathcal{N}_a)$; ▷ Apply to compute upper bound for lazy evaluation

//1-step improved Greedy algorithm

while $|\mathcal{N}_a| \leq N$ **do** ▷ Cardinality-fixed constraint

  $\boldsymbol{C}_1 \leftarrow SparseCholesky\left(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)^*(\boldsymbol{p},\boldsymbol{p})\right)$;

  $\boldsymbol{C}_2 \leftarrow SparseCholesky\left(\widehat{\boldsymbol{L}}_w^{SO(3)}(\mathcal{V}\backslash\mathcal{N}_a)^*(\boldsymbol{p},\boldsymbol{p})\right)$;

  for $j = |\mathcal{N}_a|+1 : 1 : n_p$ **do**

    //Lazy evaluation

    $\mathcal{S} \leftarrow Order(m),\ m \leftarrow |\mathcal{N}_a|+1$;

    if $\Delta_m > f_{index}$ **then** break;

    else

      $m = Order(j)$; ▷ Lazy evaluation for largest several solutions $f_{obj}(\mathcal{N}_a \cup \{s_m\})$;

      $f_{obj}(\mathcal{N}_a \cup \{s_m\}), s_m \in \mathcal{V}\setminus\mathcal{N}_a \leftarrow Rank1\_reuse(\boldsymbol{C}_1, \boldsymbol{C}_2)$; ▷ Algorithm 4

      $\Delta_m \leftarrow f_{value} - f_{obj}(\mathcal{N}_a \cup \{s_m\})$; ▷ Upper bounds

      if $\Delta_m < f_{index}$ **then** $f_{index} \leftarrow \Delta_m$;

      $\mathcal{S} \leftarrow \mathcal{S} \cup \{s_m\}$;

  end

  $s_k^* \leftarrow \max_{s_m\in\mathcal{S}} f_{obj}(\mathcal{N}_a \cup \{s_m\})$;

  $Order \leftarrow sort(\Delta_m, \text{all } m)$; ▷ Sort in ascending order

  $f_{value} \leftarrow f_{obj}(\mathcal{N}_a \cup \{s_k^*\})$;

  $\mathcal{N}_a \leftarrow \mathcal{N}_a \cup \{s_k^*\}$;

  $k \leftarrow k+1$;

return $\mathcal{N}_a$. ▷ Selected set with good performance

4.4 Potential applications in landmark setting of 2D/3D mapping and trajectory assignment of CSLAM

We present the landmark setting in 2D/3D mapping task and the trajectory

Algorithm 5 Objective computation using reuse and rank-1 update technology ($Rank1\_reuse$)

Require: Cholesky factorization of the matrices $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)^*$ and $\widehat{\boldsymbol{L}}_w^{SO(3)}(\mathcal{V}\backslash\mathcal{N}_a)^*$: $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$; the order of potential anchor $\{s_k\}$: i
Ensure: The objective function $f_{obj}(\mathcal{N}_a \cup \{s_k\})$
1: //Initialization
2: $\boldsymbol{C}_{11} \leftarrow \boldsymbol{C}_1(1:i-1, 1:i-1), \boldsymbol{c}_{32} \leftarrow \boldsymbol{C}_1(i, i+1:end), \boldsymbol{C}_{33} \leftarrow \boldsymbol{C}_1(i+1:end, i+1:end)$;
3: $\boldsymbol{C}_{11}^* \leftarrow \boldsymbol{C}_2(1:i-1, 1:i-1), \boldsymbol{c}_{32}^* \leftarrow \boldsymbol{C}_2(i, i+1:end), \boldsymbol{C}_{33}^* \leftarrow \boldsymbol{C}_2(i+1:end, i+1:end)$;
4: $\boldsymbol{v} \leftarrow cs\_etree(\boldsymbol{C}_{33}, \text{“}col\text{”})$; ▷ Return the elimination tree of $\boldsymbol{C}_{33}^\top \boldsymbol{C}_{33}$.
5: //Rank-1 update to sparse Cholesky factor based on $\boldsymbol{v}$ and old Cholesky factors $\boldsymbol{c}_{32}$ and $\boldsymbol{C}_{33}$
6: $f_{rank1} \leftarrow 2n\sum_j \log(\boldsymbol{c}_{32}\boldsymbol{c}_{32}^\top + \boldsymbol{C}_{33}\boldsymbol{C}_{33}^\top)_{jj} + 2d\sum_j \log(\boldsymbol{c}_{32}^*\boldsymbol{c}_{32}^{*\top} + \boldsymbol{C}_{33}^*\boldsymbol{C}_{33}^{*\top})_{jj}$;
7: $f_{obj}(\mathcal{N}_a \cup \{s_k\}) \leftarrow 2n\sum_j \log(\boldsymbol{C}_{11})_{jj} + 2d\sum_j \log(\boldsymbol{C}_{11}^*)_{jj} + f_{rank1}$;
8: **return** $f_{obj}(\mathcal{N}_a \cup \{s_k\})$.

assignment for CSLAM based on the proposed method, since they are two important applications in robotics.

4.4.1 Landmarks setting in 2D/3D mapping

In some highly-accurate mapping tasks, the pose-graph optimization is widely used as an important tool to improve the accuracy and the loop-closure detecting ability. However, in the large-scale feature-less SLAM problem, like DARPA Subterranean Challenge [1], it is very difficult to obtain a very accurate global pose estimation in pose-graph SLAM using only one anchor by fixing the initial pose. For example, the state-of-the-art technologies, including RTAB-map [91] and Cartographer [55], can only reach 10∼50cm accuracy in 100∼300m trajectory using MIT Stata Center data set.

The landmark setting with ground truth is a simple and efficient way to improve the accuracy of the pose-graph SLAM. A landmark will offer accurate global information to the nearby poses in the whole pose-graph, which is similar to anchoring these poses. Fig. 4.5 presents an illustration on the reduced weighted Laplacian

matrices of the base pose graph, the anchored pose-graph, and the pose-feature graph with anchored landmarks. The rows and columns of the Laplacian matrix corresponding to the anchored poses are deleted, which is equal to the infinite information saving in the diagonal block, as shown in the red diagonal sub-matrix. In (c), the added landmark commonly brings some new information to the connected pose and its corresponding diagonal elements will add a new yellow diagonal submatrix. When the weight of the measurement between landmark and the pose tends to infinity, the information increment of anchoring the landmark will approach the one of anchoring the pose.

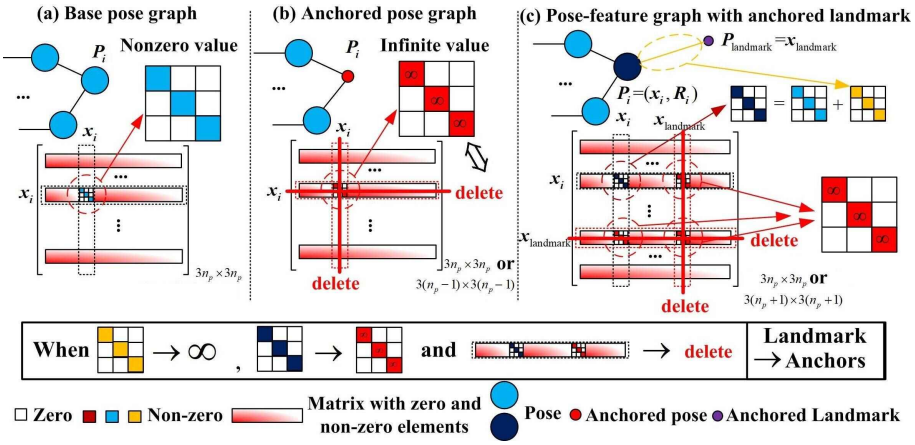


Figure 4.5 : The information involving in the pose P_i corresponding to different graphs: (a) Base pose graph and its corresponding weighted Laplacian matrix; (b) Pose graph with the anchored pose P_i and its weighted Laplacian matrix deleting the corresponding rows and columns; (c) Pose-feature graph with the anchored feature $P_{landmark}$ and its weighted Laplacian matrix deleting the rows and columns corresponding to $P_{landmark}$.

However, the landmark setting usually needs many external calibration operations with some precise instruments, such as ETS, and high human resource cost, so a good landmark setting should use fewer landmarks and yet obtain high quality map. In the following, we will explain how to apply our anchor selection method to perform the landmark setting using twice-trajectory approach and redundant landmarks approach.

Firstly, we propose the twice-trajectory approach to apply our anchor selection method to the landmark setting. As shown in Fig. 4.6, following a designed path, the robot first collects the data based on the on-board sensors, including laser data, camera data, and so on, without using anchors. Based on this collected data, we can output its corresponding pose graph and select the optimal anchors using the anchor selection framework. Then, the landmarks are set based on the selected anchors. The robot moves back to the origin, follows the path, and collects the related data with these calibrated landmarks again. Because of the similar environment and path, the measurements, especially the pose-graph structure, of two data collection processes will be similar[†]. Based on the static landmarks set by the selected anchors, the occupancy grid map is built with better accuracy using the SLAM tool-pack, like Cartographer[‡]. The anchor selection process becomes the preparation to choose the positions of the landmarks and further improves the accuracy of the pose-graph result.

The specific operations of the other way, called as redundant landmarks approach, are shown as follows. First, before the SLAM task, many redundant marks with unknown global location are placed along the trajectory. An ideal setting way is to ensure that each pose in the obtained pose graph can measure only one set mark. In practice, the marks can be placed as uniform as possible without considering whether each pose can detect the marks. Second, the SLAM task is per-

[†]Because of the sensor noise and the slight environment change, even though the robot follows the same path twice, practically, we know that it is impossible to obtain the exactly same pose graphs in two trajectories. The corresponding differences between two pose graphs will cause the potential deterioration of the proposed method's performance. However, because the main structures of two different pose graphs are highly similar, the affect of the performance deterioration is limited. Meanwhile, the twice-trajectory method will introduce the additional cost in the robot motion and the additional calibration operations. Compared with the operations to improve the used SLAM method, these costs are commonly acceptable especially for the industrial users, who do not want to deeply revise the SLAM methods.

[‡]It is noted that our anchor selection framework can be used not only for Cartographer, but also applied in different SLAM methods using pose-graph optimization, including ORB-SLAM2 [106], RTAB-Map [91], and so on.

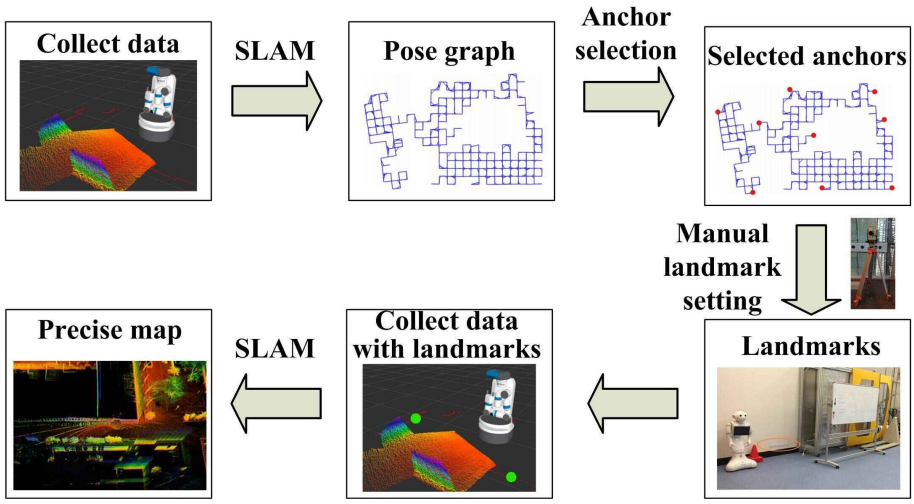


Figure 4.6 : Landmark setting operations using the twice-trajectory approach.

formed by following the designed path. During this process, all detected marks are tracked, mapped, and involved in the SLAM method. Then, based on the obtained pose-graph[§], our proposed anchor selection method is used to pick out the optimal anchors. Following, the marks detected by these selected poses need to be calibrated and the calibrated results will be regarded as the additional measurements and be introduced into the original SLAM measurement network. Finally, with the help of these additional calibrated measurements, the SLAM problem is solved again to get a better solution. Compared with the twice-trajectory approach, this approach only needs to follow the designed path once without repeating. The disadvantage is that many more marks need to be placed because of the large number of the robot poses[¶].

[§]Most well-known mature laser/visual SLAM methods, like cartographer and ORB-SLAM2, use the pose-graph optimization. They also have ability to track some un-calibrated marks, like the 'landmark' interface in Cartographer.

[¶]Because of its disadvantage, the implement of the redundant landmarks approach is relatively difficult, in our latter simulation and experiment, if we do not specifically mention the way to realize the landmark setting, the default way is the twice-trajectory approach.

4.4.2 Trajectory assignment in CSLAM

CSLAM [50] is a framework to use several robots to finish the SLAM task cooperatively in an unknown space, which means that every robot first obtains a sub-map using the classical SLAM framework based on its initial pose and finally a global map is obtained by fusing all the sub-maps. A natural question is, how to assign the trajectories for different robots such that they can obtain a better SLAM result?

In some CSLAM tasks, the initial poses of the robots are known in a common global frame, which means the initial robot poses can be regarded as multiple anchors of the pose-graph SLAM problem. In this situation, the trajectory assignment problem for the robots becomes a similar problem to find the best anchored poses. The anchor selection for CSLAM can be performed using the following steps. Firstly, only one robot[‖] is used to follow the complete designed path including all places which need to be visited by all robots. After the pose graph is built, the optimal anchored poses can be selected using the anchor selection method, given the number of the robots as the number of the anchors. Then, the static initial poses of the robots are calibrated using some additional measurements. The path is divided into multiple parts by the anchored poses and each one between every two adjacent anchors is assigned to the corresponding robot[**]. Finally, the robots move from their own origins to follow the assigned paths and complete the CSLAM tasks.

4.5 Simulations

In this section, some simulations are performed to validate our theoretical developments and evaluate the performance of the presented method based on some

[‖]Because of the resource and scenario constraints, in some tasks, multiple robots need to be used to perform the CSLAM task. We can perform CSLAM task using multiple robots with unknown initial poses to replace the task using only one robot with only one initial pose.

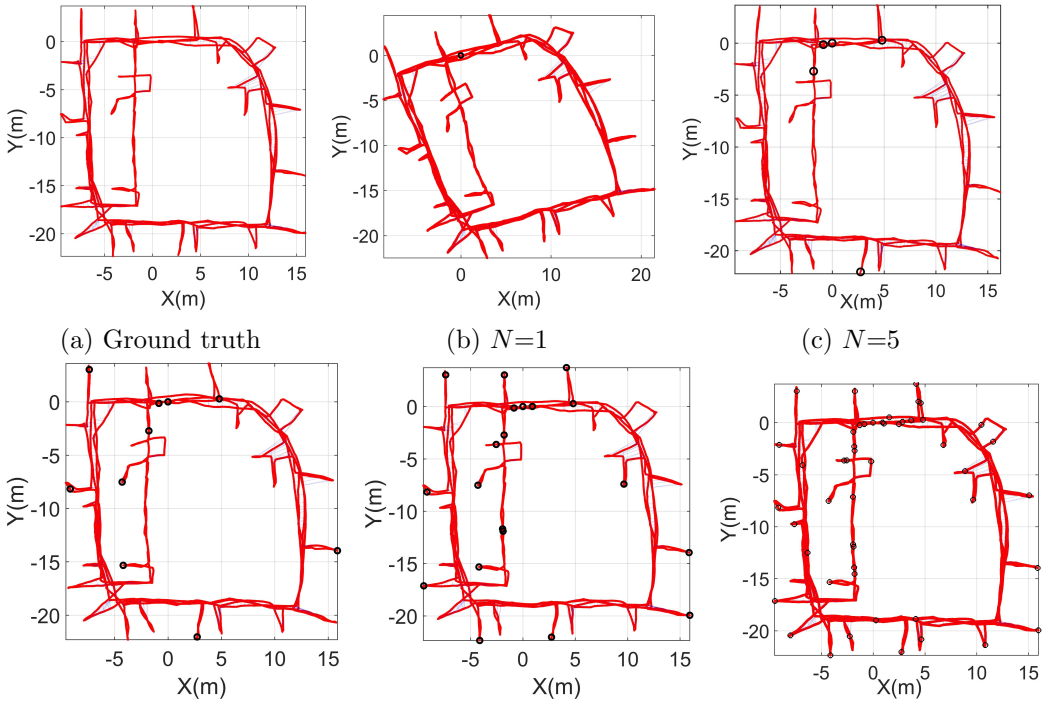
[**]In order to avoid some paths of some robots are much shorter than the others, the potential pose set for anchors can be constrained such that the lengths of different paths are all similar.

well-known datasets.

All of the following experiments are performed on a Dell E5570 laptop with an Intel Core i5-6300U 2.40 GHz processor and 8 GB of RAM running Windows 7 Enterprise. Our experimental implementations are written in MATLAB.

4.5.1 Different number of anchored poses

Based on Intel dataset (1728 poses), we increase the number of the anchored poses, choose them by the improved greedy method, and finally get the estimated results based on the anchored poses, shown in Fig. 4.7.



(a) Ground truth (b) N=1 (c) N=5

(d) N=10 (e) N=20 (f) N=50

Figure 4.7 : Estimated results using the Intel dataset (Red line) based on the best selected anchors (Black circles) obtained from the lazy greedy method, N is the number of anchors

Fig. 4.7a is the ground truth of the Intel dataset. In Fig. 4.7b, we can see that it is really hard to get a good pose-graph SLAM result based on one anchor, when the noises are large (Same noise level as real dataset). When the number of anchored

poses N increases from 5 to 50, based on the anchored poses, the estimated results become better and better as can be seen from Fig. 4.7c to Fig. 4.7f. Their mean coordinate errors are respectively 2.3727 m (Fig. 4.7b), 2.9514×10^{-1} m (Fig. 4.7c), 1.7361×10^{-1} m (Fig. 4.7d), 7.5231×10^{-2} m (Fig. 4.7e), 2.7199×10^{-2} m (Fig. 4.7f). It is easy to see that our sub-modular optimization method can find the efficient anchored poses to improve the accuracy of the SLAM result.

4.5.2 Comparison between improved method and normal greedy method

Because our improved method will not change the result obtained by the 1-step greedy method, in this section, we only compare it with the normal greedy method in the aspect of the computational efficiency to verify the analysis shown in Section 4.3.1 and (4.16). In the following table, we compare them by several real or well-known datasets, including: 2D: CSAIL, intel, manhattan, KITTI, city10000, ais2klinik ; 3D: tinyGrid3D, smallGrid3D, torus3D, cubicle, grid3D and rim.

In Table 4.1, the greedy method represents Algorithm 2 only using the sparse Cholesky decomposition shown in Section 4.3.4. The improved method means that all speed-up technologies shown in Section 4.3.4 are applied. Under the circumstances with the same final results, we can find that our proposed method can be much faster than the presented naive 1-st greedy method. Even for some very large datasets, like city10000, ais2klinik and rim, with more than 10^4 poses, our later presented method can solve them within several minutes. In order to further understand the efficient of different technologies in Section 4.3.4, using Intel and CSAIL datasets, we apply these technologies one by one, and then, show the computational time changing with their applications in Fig. 4.8.

In Fig. 4.8, we can find that, by the help of these speed up technology, the running ability of this method becomes faster and faster.

Table 4.1 : Comparison of Computational time for different SLAM datasets

	Dataset	# Poses	Mean Computational time in 10 runs[s] N=1	N=5	N=10	N=20	N=50
Greedy method (2D)	CSAIL	1045	0.0584	3.6349	7.8533	16.1398	40.4455
	Intel	1728	0.0602	15.4858	34.5448	72.0651	183.0900
	manhattan	3500	0.0585	86.7457	187.9391	401.7045	1110.5978
	KITTI	4541	0.0571	57.2313	139.8398	294.8835	727.6991
	city10000	10000	0.0606	1298.3256	2927.0964	6117.0702	15535.6554
	ais2klinik	15115	0.0773	991.3454	2132.4369	4578.9760	11764.0283
Greedy method (3D)	tinyGrid3D	9	0.0892	0.0209	-*	-	-
	smallGrid3D	125	0.0840	0.2126	0.4268	0.7817	1.4366
	torus3D	5000	0.0606	595.9520	1347.1887	2710.1122	6289.2731
	cubicle	5750	0.0811	537.1241	1196.3851	2211.4586	6077.1242
	grid3D	8000	0.0571	5618.1917	10152.7144	>20000	>20000
	rim	10195	0.0870	2582.7873	5372.1122	11421.1211	>20000
Improved method (2D)	CSAIL	1045	0.0144	0.7732	0.9865	1.2732	1.7657
	Intel	1728	0.0175	3.0832	4.0620	4.9781	5.8554
	manhattan	3500	0.0322	7.5656	13.1060	16.8376	21.3019
	KITTI	4541	0.0215	10.1819	14.0718	18.5387	28.9056
	city10000	10000	0.0948	63.2231	64.1240	142.4568	180.8735
	ais2klinik	15115	0.0508	100.2318	122.2698	169.7197	238.3487
Improved method (3D)	tinyGrid3D	9	0.0930	0.0475	-	-	-
	smallGrid3D	125	0.0705	0.1965	0.2387	0.2762	0.3965
	torus3D	5000	0.0716	70.6634	73.1187	83.2794	94.5362
	cubicle	5750	0.0590	76.8378	78.3145	84.4654	102.0602
	grid3D	8000	0.0871	1216.2893	1804.8911	1953.7712	2107.2340
	rim	10195	0.2131	290.2561	295.7529	308.2754	321.0302

* '-' means the computation is infeasible, because the pose number is smaller than the anchor number.

4.5.3 Comparison with two heuristics

In this part, our method is compared with two heuristics, including maximal node degree way and random way, using the CSAIL, Fr-clinic and FR079 datasets. The maximal node degree way, named 'Max', means to choose the nodes which have the maximal weighted node degrees. Because these nodes are the strongest connected nodes with the whole network, they can help to improve the accuracy of the SLAM result. The random way, named 'Random', is a cheap but not very efficient way. We make full use of its fast running ability. The random way is run for many times (100 times) and the best one is picked out. Their comparisons of the

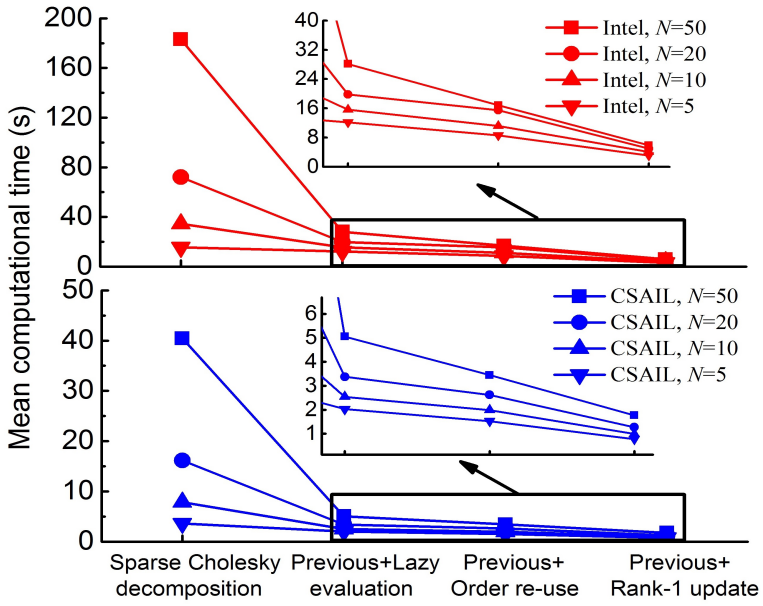


Figure 4.8 : Computational time reduced with the applications of speed-up technologies

final estimated coordinate errors are presented in Fig. 4.9, Fig. 4.10 and Fig. 4.11. In every figure, the top-right corner shows the ground truth of the corresponding dataset.

Based on Fig. 4.9, Fig. 4.10 and Fig. 4.11, we can see that the greedy-based method has the best performance and its performance is stable for different datasets and anchor numbers. In order to further show the superiority of our method, based on the data in Fig. 4.9, Fig. 4.11 and Fig. 4.10, we define $RE = \left(\sum_{N=2}^{N=20} \frac{\text{Others}-\text{ours}}{\text{ours}}\right)$ /19 and $RE1 = \max_{N=2,\cdots,20} \frac{\text{Others}-\text{ours}}{\text{ours}}$ by the estimated coordinate errors, where Others and ours respectively are the estimated coordinate errors of the poses using different number of anchors corresponding to different methods. Their results are shown in Table 4.2.

In Table 4.2, we can find that our method has more than 40% and 176.87% advantage compared with the other methods in the mean and maximal accuracy of the final estimated results. Our method can offer a stable anchor selection result

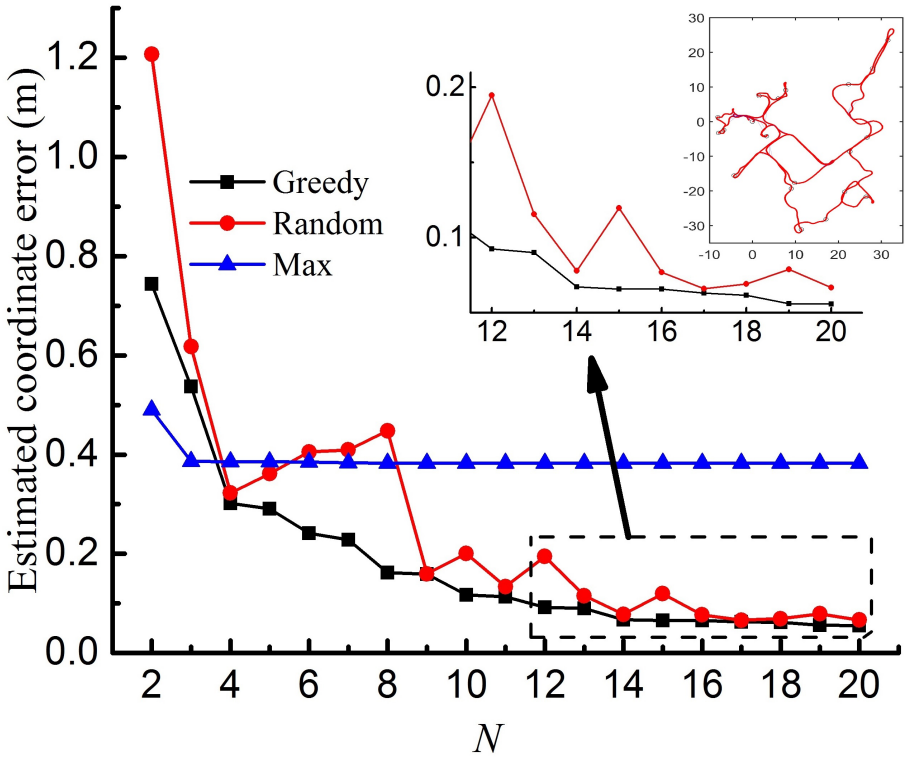


Figure 4.9 : Comparison with two heuristics using CSAIL dataset

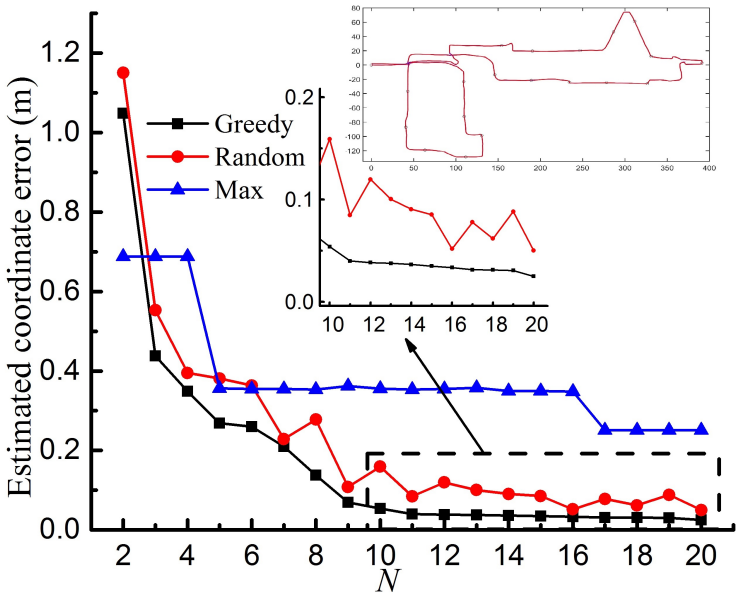


Figure 4.10 : Comparison with two heuristics using FR079 dataset

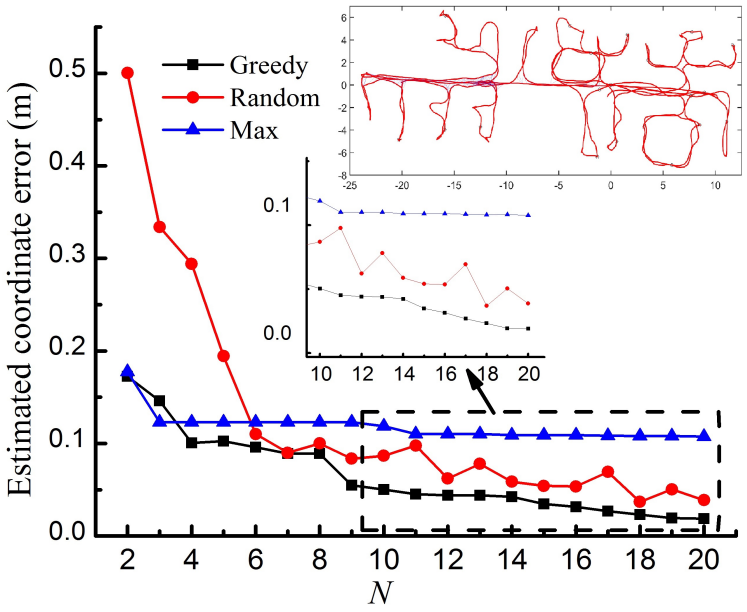


Figure 4.11 : Comparison with two heuristics using Fr-clinic dataset

Table 4.2 : Comparison with two heuristics for different SLAM datasets

Superiority	Average (RE)		Maximal ($RE1$)	
Dataset	Random	Max	Random	Max
CSAIL	44.96%*	270.78%	176.87%	587.47%
Fr-clinic	86.16%	160.93%	192.53%	469.81%
FR079	98.06%	504.59%	212.00%	940.98%

* When the value is bigger than 0, it means that our method is better.

with good performance.

4.5.4 Approximation performance verification and uncertainty level comparison with heuristics

In this chapter, we approximate the original unsolvable problem in (4.1) to the new sub-matrix selection problem based on Theorem 7. In this part, we use CSAIL and FR079 datasets to show the approximated performance and further compare the uncertainty level of the obtained results using different methods. The approximation and comparison results are shown in Fig. 4.23 and Fig. 4.13.

It is easy to find that the obtained objective function values, used in Eq.(4.2),

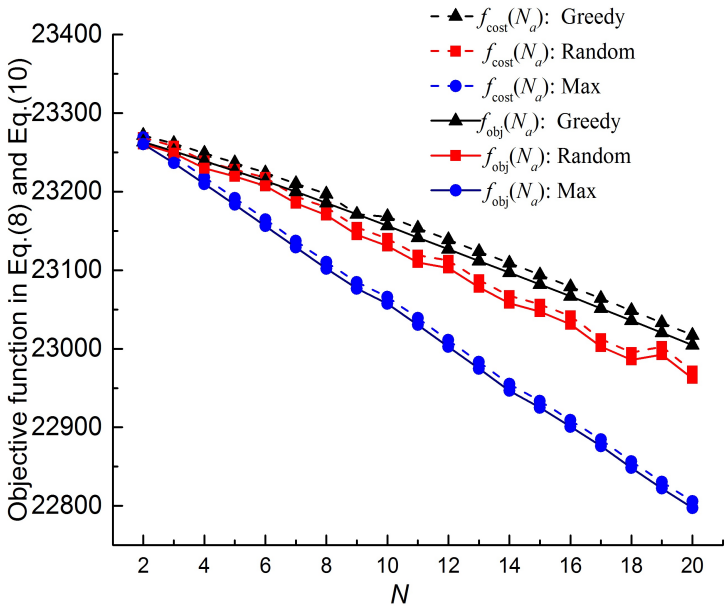


Figure 4.12 : Approximation perfromance and comparison with two heuristics using FR079 dataset

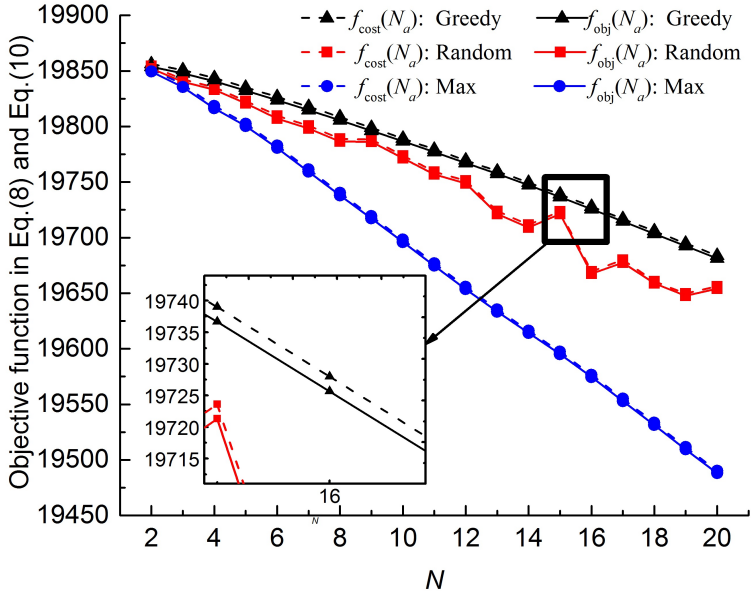


Figure 4.13 : Approximation perfromance and comparison with two heuristics using CSAIL dataset

gets greatly closed to the original log-determinant of the FIM. The approximation performance is very good, which helps us to solve the new problem to get efficient results. Meanwhile, the results also show that the final result uncertainty of our greedy method is smaller than the others.

4.5.5 Performance bounds

In this part, we will verify the tightness of our presented bounds. We use multiple methods to select anchors, and then output the objective function of the CSAIL dataset, shown in Fig. 4.14. Based on Lemma 2, we can get the lower bound and the upper bound of the random greedy and verify that the lazy greedy method locates in the bounds, which means that the optimal solution locates in a smaller area. This area is shown as the dashed area, named "Optimal area".

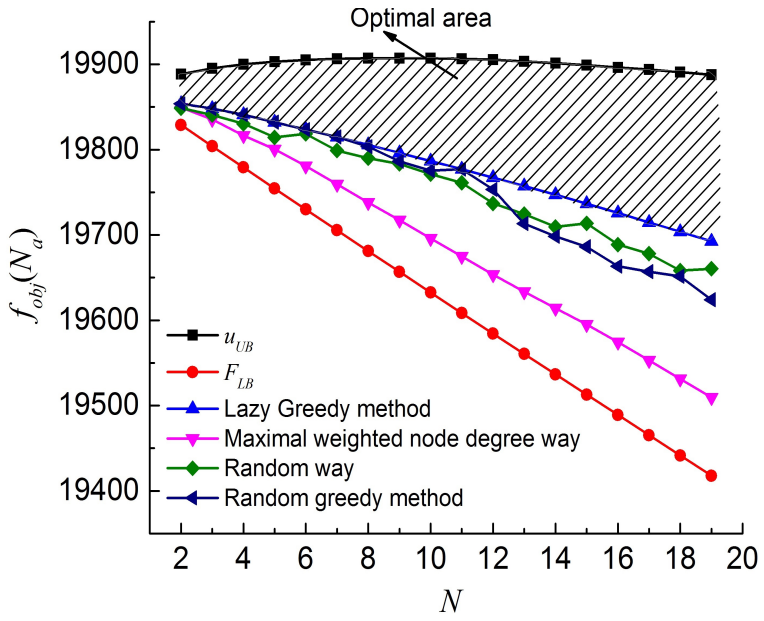


Figure 4.14 : Bounds for Lemma 2 using CSAIL dataset

4.5.6 Simulation for potential applications

In this subsection, we present the applications of anchor selection in the CSLAM trajectory assignment problem and the landmarks setting of the mapping problems.

CSLAM trajectory assignment problem

Based on the anchor selection method, we use 5 robots to share the trajectory of the real CSAIL dataset (1045 poses). Assuming that the initial poses of the five robots are known, the trajectory assignment result is shown in Fig. 4.15. The final mean coordinate error $RE2 = \sum_{i=1}^{n_p} \|\boldsymbol{x}_i^{opt} - \boldsymbol{x}_i^G\|/n_p$ reduces from 2.1658m (only one robot) to 0.2905m (5 robots with known initial poses), where $\boldsymbol{x}_i^{opt}$ and $\boldsymbol{x}_i^G$ are the optimal solution obtained by the GN method using 5 robots and the ground truth of the dataset respectively. The results show that our method can be used in the CSLAM trajectory assignment problem with the known initial poses to improve the result accuracy.

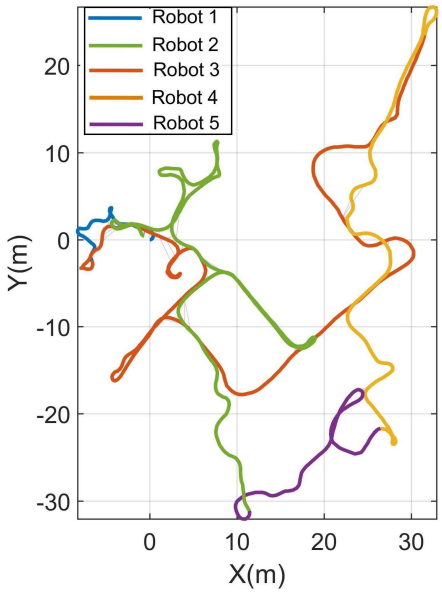


Figure 4.15 : Five robots perform coorperative SLAM using CSAIL dataset.

So as to further validate this application, we present a CSLAM simulation using five quad-rotor unmanned aerial vehicles (UAVs). At first, only one UAV is used to get the initial pose graph. As shown in Fig. 4.14, in a 7m×7m×1m environment with several regular obstacles (cubics, cylinders, and spheres), a UAV moves from the first pentagram $(0.0, 2.0, 0.2)$ with a velocity 0.1m/s, passes several pre-defined way-

points (blue pentagrams), and meanwhile performs the pose-graph SLAM task. In each simulation, the positions of the features are randomly generated. In the moving process of the UAV, the features will be detected when they locate in the sensor range of the UAV (1.5m) and the relative pose measurements are obtained based on the common features detected from two poses. So as to make the measurements follow the noise assumption shown in (B.51), using the singular value decomposition, the noise-free relative rotation and translation measurements are obtained from the 3D features observations, and then, the random noises are sampled by $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{ij})$ and $Lang(\boldsymbol{I}_{n\times n}, \kappa_{ij})$. The parameters $\kappa_{ij} = 10^3 N_v$ and $\delta_{ij} = 50N_v$ for visual odometry are set to be proportional to the feature number N_v, which are visible both from P_i and P_j [106]. The noise parameters δ_{ij}^{-2} and κ_{ij} for the control input are 1.6×10^5 and 800 in every step (The simulation time step Δt is set as 1s.). In Fig. 4.14, the yellow star trajectory is the pose-graph SLAM result using only one UAV with no additional anchors.

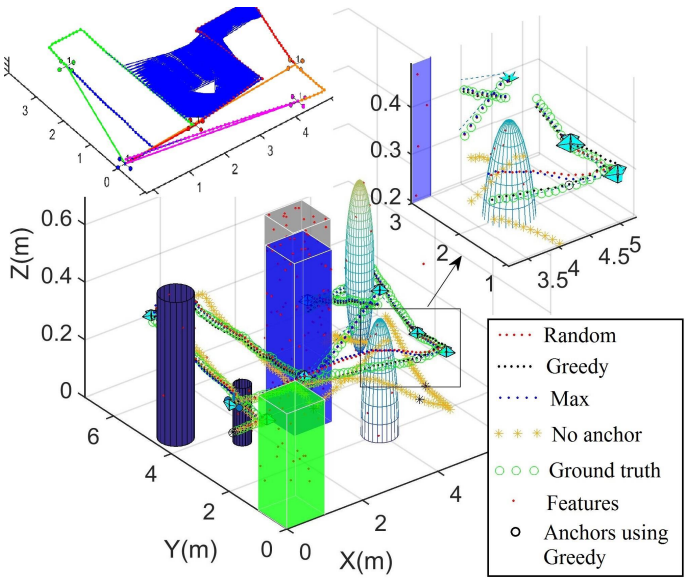


Figure 4.16 : Estimated result for collaborative pose-graph SLAM.

Then, based on the generated pose-graph, using multiple anchor selection methods, the starting points of the UAVs are picked out and the trajectories between

the adjacent starting points are assigned to the corresponding UAVs. Following, using the external global measurement tools to obtain the initial poses of the UAVs, the collaborative pose-graph SLAM tasks for five UAVs are solved. By randomly generating the different features, we run this simulation 20 times using all methods. One of the results are shown in Fig. 4.16. The red, black, and blue point trajectories are the estimated results using the random method, the greedy-based method, and the maximal node degree method, respectively. Using the proposed greedy-based method, the trajectory assignment results with five trajectories (blue: 1-st UAV; green: 2-nd UAV; red: 3-rd UAV; Orange: 4-th UAV, and magenta: 5-th UAV) and the generated pose-graph with relative measurements (blue lines) and poses (colorful points) are shown in the top-left of Fig. 4.16.

The statistical results of the estimated coordinate errors using three methods are presented by the box chart in Fig. 4.17. The RE compared with the random method and the 'Max' method are 128.49% and 302.92%. The $RE1$ compared with the random method and the 'Max' method are respectively 327.23% and 660.54%. In 20 times, our method shows a consistently better performance over the other methods, which verifies the practicality of our framework in the CSLAM task.

Landmarks setting problem in feature-based SLAM

For a feature-based SLAM simulation, we aim to obtain high-quality estimates of 100 randomly-generated features in a 120m $\times$ 120m environment by adding 4 landmarks with known locations (since the first pose is also known, this corresponds to 5 anchors), shown in Fig. 4.18. The sensor range is 15m. Firstly, without adding landmarks (only 1 anchor), we follow the designed path and collect the measurement data. Using the optimization-based feature-based SLAM, the estimated result is relatively poor especially in the boxed areas, which makes the mean error of the mapped features very large (about 0.217m). Based on the pose-graph generated by

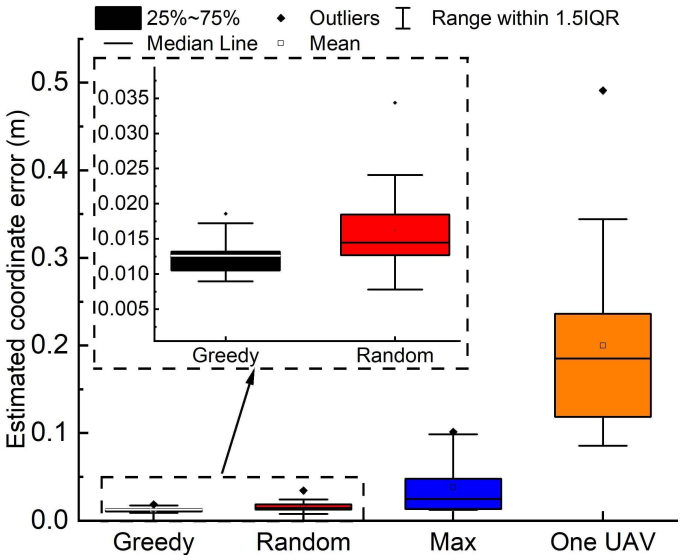


Figure 4.17 : Box chart of the estimated coordinate errors using different methods.

the relative pose measurements, using our presented method, the optimal anchored poses are selected and the corresponding 4 landmarks are set near the selected anchors, as shown in Fig. 4.18. We perform the feature-based SLAM by following the path again. The ATE of the final estimated results with the aid of the fixed landmarks is reduced to 0.0097m.

Compared with the unstable random method, the advantage of our method will become more obvious, if the distribution of the poses and the measurements is relatively uneven. In order to show this phenomenon, we change the size of the sampling area to generate the uniformly distributed features. We can apply a variable ξ to control the size of the feature area to be 120ξm$\times 120\xi$m. After changing the variable $\xi < 1$, we can randomly generate the features in the changed area, and then, we select the anchors by using two methods (the random method and the presented method) and obtain the features using SLAM with landmarks. Finally, the ratios of their estimated results changing with the variable ξ are shown in Fig. 4.19. Because of the randomness of the features, we test 20 different feature datasets and get the average result.

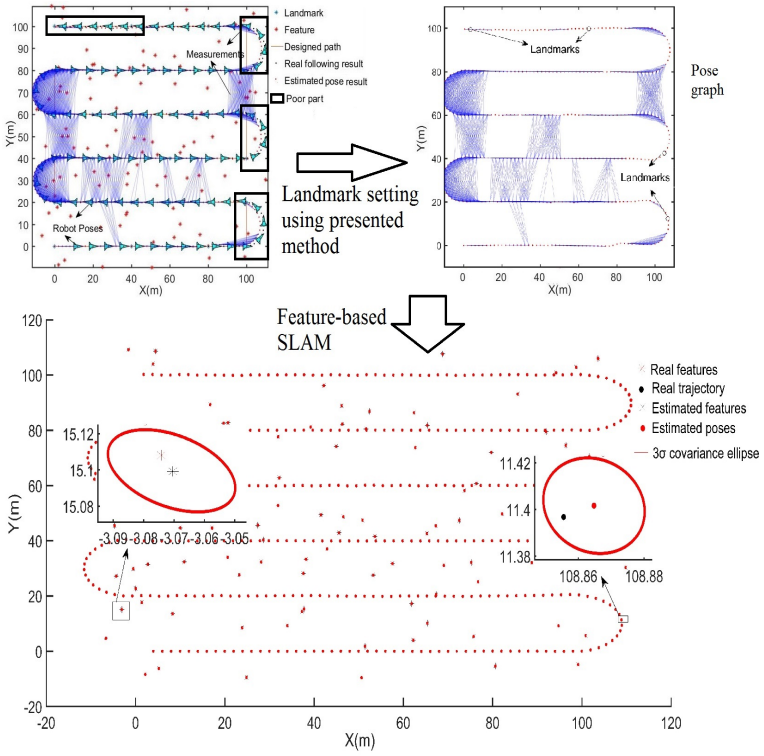


Figure 4.18 : Operations to obtain good SLAM result using presented method

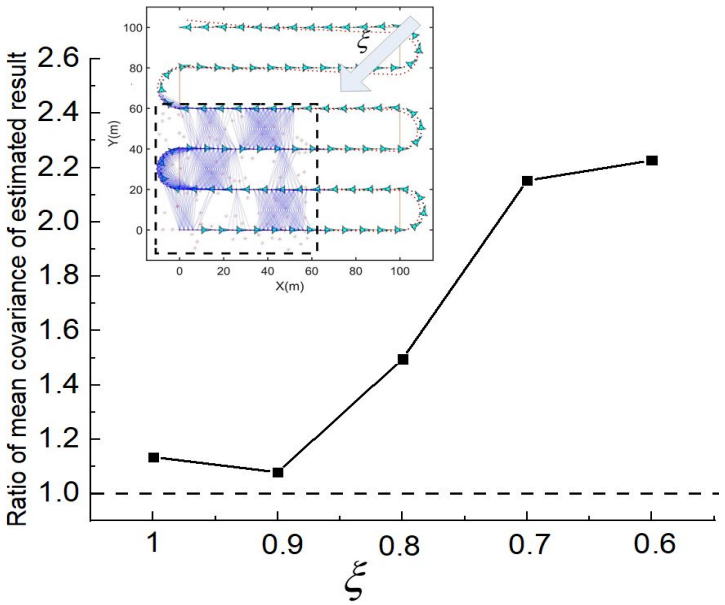


Figure 4.19 : Ratio of covariance errors of SLAM results changes with ξ (whether it is an relatively uniform pose-graph) using random way and presented method. The ratio is defined as $\frac{\text{Random}}{\text{Ours}}$. So when values are larger than 1, it means our method can obtain better result with smaller error.

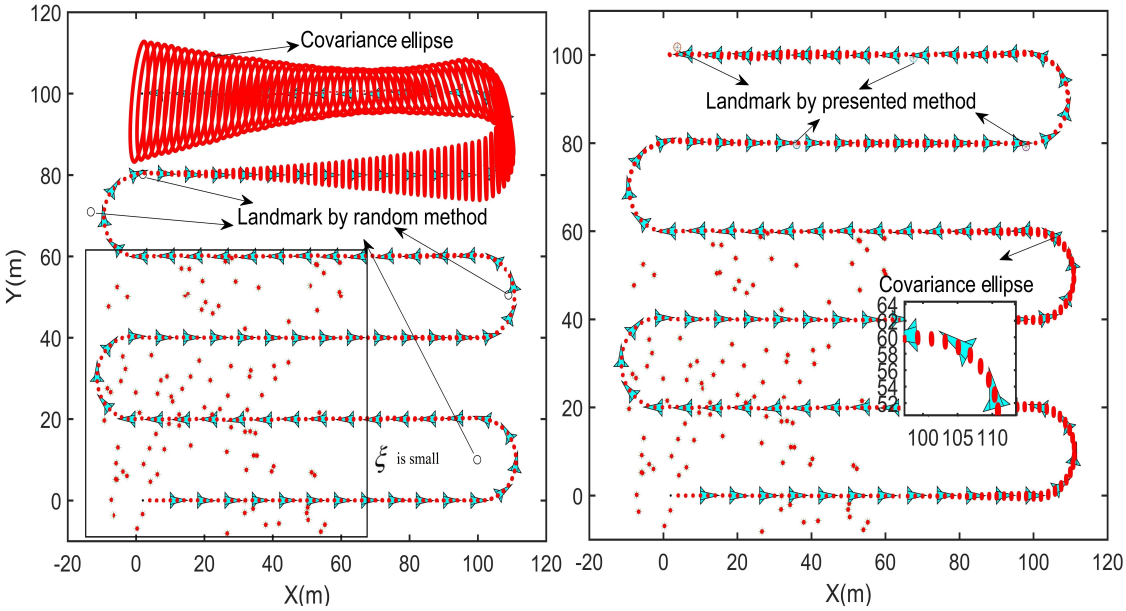


Figure 4.20 : Comparison of feature-based SLAM results using random way and presented method, when ξ is small (inhomogeneous pose-graph)

Our method achieves better performance when ξ is smaller based on Fig. 4.19. Using 5 anchors, we also show the estimated result with red covariance ellipse using different methods when ξ is small ($\xi = 0.6$) in Fig. 4.20. The advantage of our method is obvious as shown from the estimated ellipses of the poses.

Landmarks setting problem using Cartographer and Fetch simulator

An occupancy grid mapping task using the Fetch simulator and Google Cartographer is implemented to further verify the practicality of our framework (Fig. 4.6) in a well-known Garage map (Fig. 4.21). The data are collected from the environment by following a designed path (lower half part) and an initial map is generated using Cartographer, which is shown in Fig. 4.22. Based on the generated pose-graph, the optimal anchored poses are selected and then the simulated landmarks are set in front of the robot poses (1m away, in front of the robot) (Fig. 4.23). After we get these landmarks, our following work is to involve these special measurements in the mapping process. There is an interface in Cartographer to track some landmarks

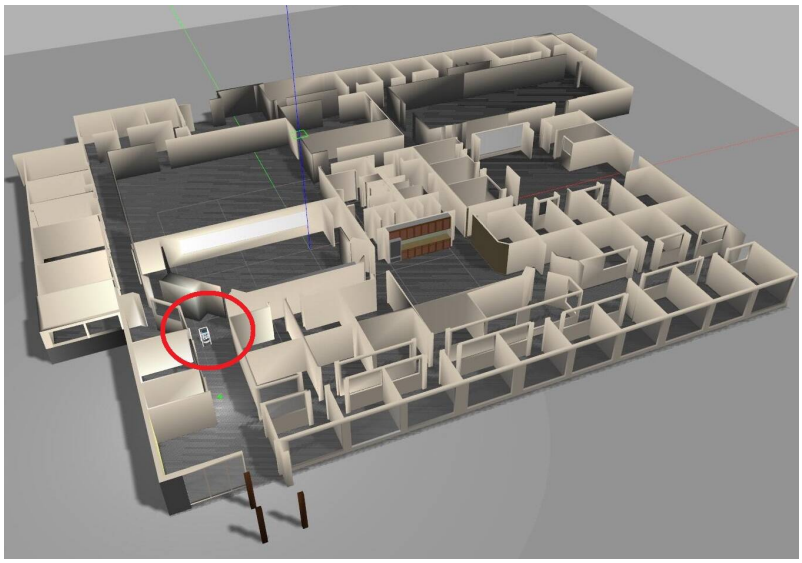


Figure 4.21 : Fetch simulator in the Willow Garage map.

without knowing their global locations. In order to anchor them, at the first pose $\boldsymbol{x}_0$, we publish simulated measurements between all selected landmarks and robot poses with large weights, which means the global locations of these landmarks are almost known. Then, after publishing these landmarks, the landmark measurements will be only obtained when the robot locates near to these landmarks. This process is shown in Fig. 4.24. Finally, we can generate a new map using Cartographer with landmarks. The mean errors of relative distance of some corner points A, B, and C, which are pointed out in Fig. 4.25, in the new map reduce from 0.32m, 0.42m, and 0.78m to 0.21m, 0.31m, and 0.39m, respectively. We also generate a larger map based on this environment. The comparison between the results without using the landmarks, using our method with 5 landmarks, and using the random method with 5 landmarks are shown in Fig. 4.26 and Table 4.3. These results show that our method helps to find the efficient landmarks and generate an accurate occupancy grid map.

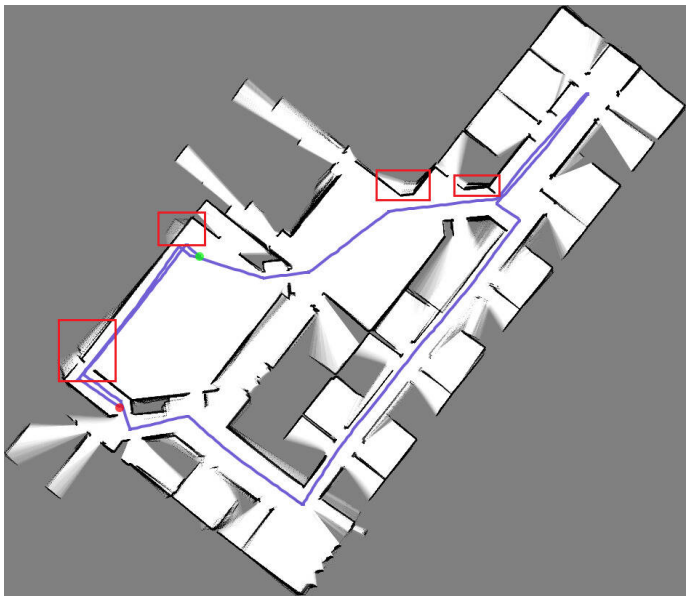


Figure 4.22 : Cartographer result without using anchors (some poor parts in red boxes).

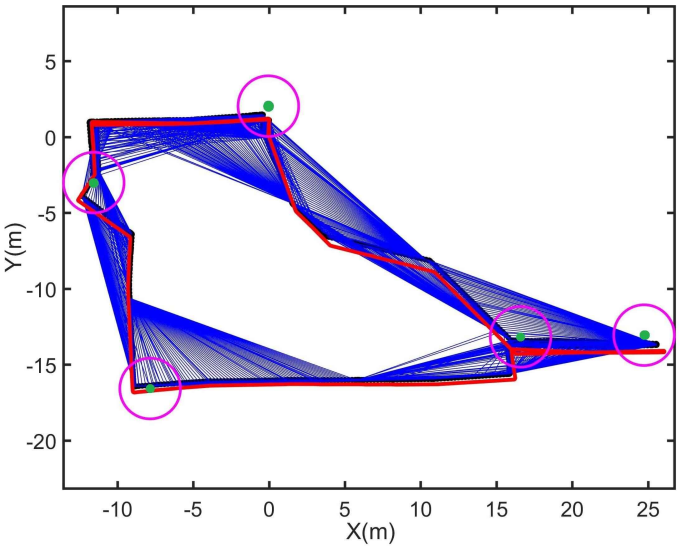
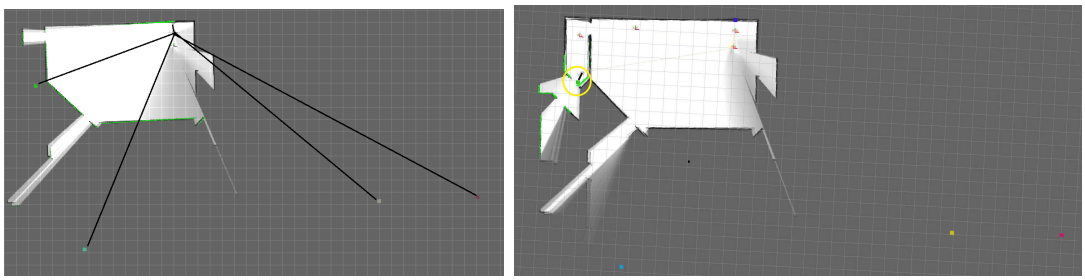


Figure 4.23 : Pose graph corresponding to Cartogaprapher result (black points and blue edges), ground truth (red points), and selected points for landmarks (green points and magenta circles).

4.6 Experiments

In this section, some experimental results are presented to validate the practicality of our theoretical developments and evaluate the performance of the method

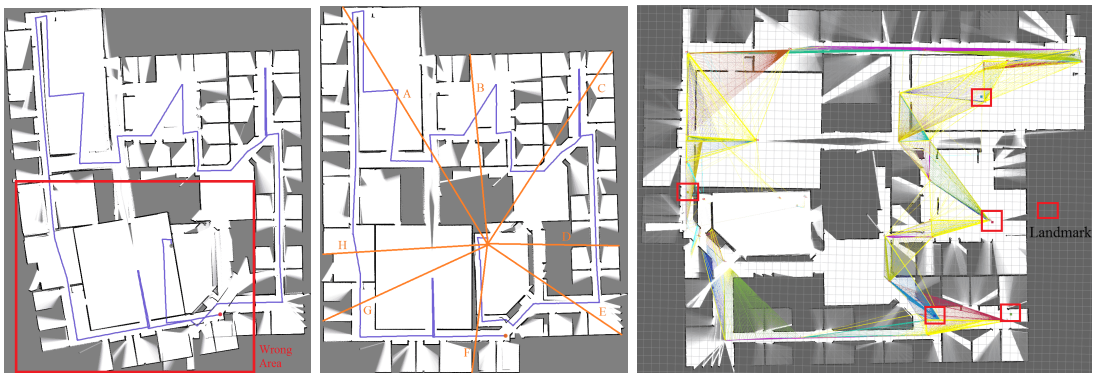


(a) Landmark initialization based on the first pose.

(b) Measurements from 2-nd landmark when robot moves within yellow circle.

Figure 4.24 : Publish 5 landmarks in simulator

Figure 4.25 : New map based on five landmarks.



(a) Result without using landmarks.

(b) Result with 5 landmarks (our method).

(c) 5 selected landmarks using our method.

Figure 4.26 : Comparison results for the large maps.

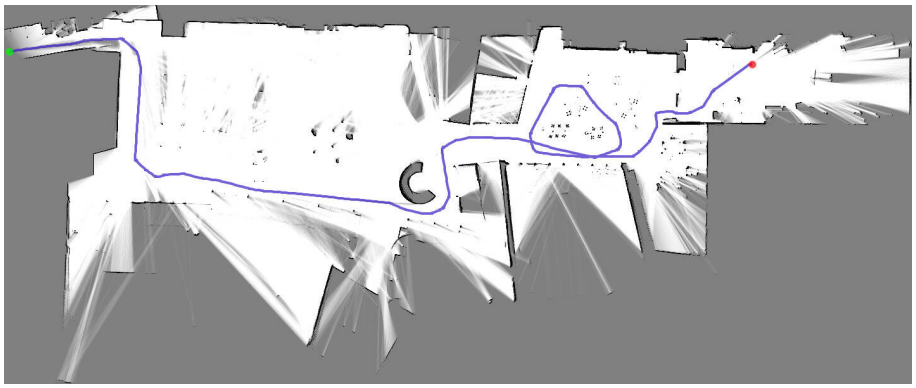
Table 4.3 : Length comparison of some line segments in obtained maps

Line	Ground truth	Our method	Random	No landmarks
A	42.10 m	**41.90 m**	41.88 m	41.06 m
B	28.86 m	28.54 m	**28.64 m**	28.36 m
C	34.91 m	**34.80 m**	35.05 m	35.07 m
D	19.73 m	19.49 m	**19.55 m**	19.47 m
E	24.28 m	**23.99 m**	23.77 m	23.68 m
F	19.16 m	**19.33 m**	18.70 m	16.70 m
G	26.98 m	**26.89 m**	25.66 m	23.66 m
H	24.59 m	**24.46 m**	23.58 m	21.58 m

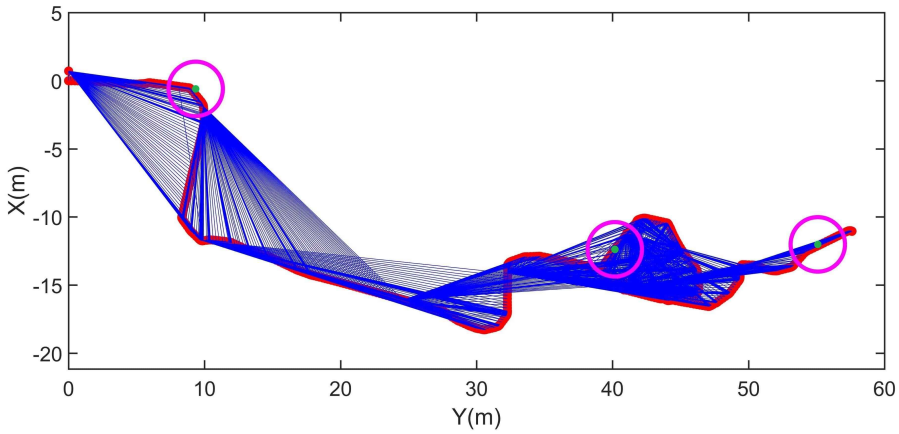
in a real lab environment (UTS Tech Lab). The experiments are performed using C++ on a MSI GL62VR laptop running Ubuntu 16.04.6 LTS as well as on the real Fetch robot. The laser reflectors, which can increase the intensity of the laser scan, are regarded as the identified landmarks for Fetch robot.

We first control the Fetch robot to collect data and implement a mapping process using Cartographer without using the reflectors as landmarks. In a relatively large environment, the SLAM result is not accurate using Cartographer only in Fig. 4.27a. Using the greedy-based anchor selection method, 3 key poses are selected with the largest increment of the tree-connectivity, which is shown in Fig. 4.27b, and then, set the landmarks with the laser reflectors by the theodolite. Based on the similar trajectory and landmarks, we can achieve a better Cartographer result (Fig. 4.28). Using the same collected data, we also output the result without landmarks. The distances of four line segments shown in Fig. 4.28 are measured by a tape measure and compared with the results of pure Cartographer. Table 4.4 shows that the results of our method are closer to the actual measurements. Photographs of the data collection process using the Fetch robot are shown in Fig. 4.29.

With the help of the landmarks, the map error reduces a lot, which shows the efficiency of the proposed method in the real mapping task in the lab environment.



(a) Obtained map for UTS Tech lab without landmarks.



(b) Pose graph (red points and blue edges) and selected points (green points and magenta circles) for landmarks using Fetch robot.

Figure 4.27 : Cartographer result without using landmarks and obtained landmarks

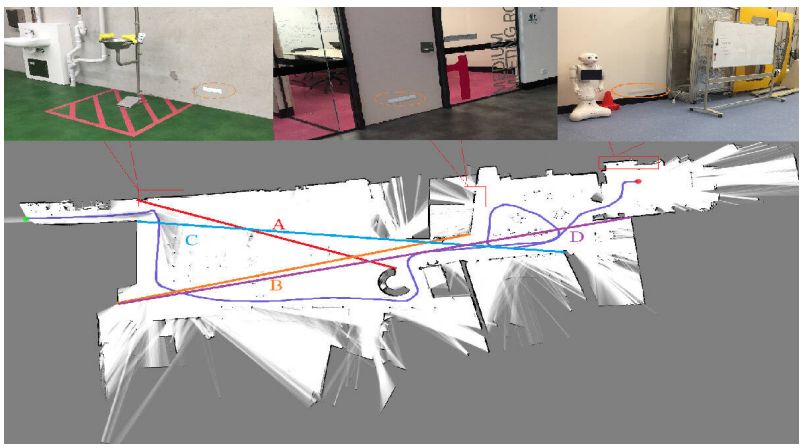


Figure 4.28 : Obtained occupacy grid map based on 3 landmarks.

4.7 Summary

This chapter formulates the anchor selection problem in the 2D/3D pose-graph SLAM problem with multiple anchors and presents a sub-modular optimization

Table 4.4 : Length comparison and error percentage of some line segments in obtained occupacy grid maps

Line	Ground Truth	Using Landmarks	No Landmarks
A	26.345m	**25.92m (1.61%)**	24.48m (7.08%)
B	33.885m	**34.11m (0.66%)**	34.38m (1.46%)
C	40.623m	**40.45m (0.43%)**	40.41m (0.52%)
D	48.823m	**48.09m (1.50%)**	47.59m (2.53%)

(a) Theodolite. (b) Data collection. (c) Fetch robot.

Figure 4.29 : Experiments using Fetch robot.

framework to solve it. Based on the graph topology, the anchor selection problem based on the original D-optimality metric is approximated as the sub-matrix selection problem for the weighted Laplacian matrix, which greatly reduces its computational complexity. Then, this new sub-matrix selection problem is proved to be a non-normalized non-monotone sub-modular optimization with the cardinality-fixed constraint, which is NP-hard. In order to solve it, the natural greedy method and the random greedy method with the lazy evaluation are presented. The performance bounds of the optimal solution are presented and discussed. We also present multiple technologies to improve the running time ability of the presented method. Several applications using this technology are put forward. The final simulation and experimental results based on some well-known datasets and real lab environment

show the outstanding performance of the proposed approximated methods.

Chapter 5

2D Active SLAM based on Graph Topology, Sub-map Joining and Convex Optimization

As shown in Chapter 2, despite we already have very powerful SLAM front-end and robust back-end methods, following along with a challenging trajectory, it is really hard to get an accurate and consistent SLAM result using the best available SLAM. The active SLAM, which involves two aspects: SLAM and trajectory planning, offer us an important option to apply the robot action to get the better measurement and sufficiently use these obtained information. Our purpose in this chapter is to plan a reasonable trajectory for robot to improve the accuracy of the SLAM results and even make the system more robust. In this chapter, we propose an active SLAM algorithm based on graph topology, sub-map joining and convex optimization method for 2D feature-based SLAM problem.

5.1 Problem statement

5.1.1 Vehicle model

Based on the constant altitude and zero pitch/roll angle flight assumption, the kinematic equations of the UAV moving at a constant velocity V in a 2D environment can be written as:

$$\begin{aligned} \widehat{\boldsymbol{v}} &= \boldsymbol{R}_k^\top(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + (\ \delta x_k \quad \delta y_k\ )^\top, \\ \omega_k &= (\theta_{k+1} - \theta_k)/\Delta t + \delta\omega_k\ , \end{aligned} \tag{5.1}$$

where $\widehat{\boldsymbol{v}} = (\ V\Delta t \quad 0\ )^\top$, Δt is a discrete time interval, $(\boldsymbol{x}_k{}^\top, \theta_k) = (x_k, y_k, \theta_k)$ is the pose of the UAV at the k-th step, ω_k is the angular velocity of the UAV at the k-th step, δx_k, δy_k and $\delta\omega_k$ are the discrete time noises of the coordinates and angular velocity and are assumed to be zero-mean Gaussian, $\boldsymbol{R}_k = \begin{pmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{pmatrix} \in SO(2)$ is the rotation matrix of the k-th pose of the UAV.

5.1.2 Sensor model

There are many types of sensors used in active SLAM tasks. We assume there are a number of natural landmarks locating on the ground, and there is a downward looking sensor, such as a camera, installing on the bottom of the UAV. On the 2D environment, this sensor can be regarded as an omnidirectional sensor for the SLAM. The range of the sensor is R_s. In this chapter, a simplified range sensor model is used:

$$\boldsymbol{z}_k^{fi} = \boldsymbol{R}_k^\top(\boldsymbol{x}^{fi} - \boldsymbol{x}_k) + (\ w_x^k \quad w_y^k\ )^\top, \tag{5.2}$$

where $\boldsymbol{z}_k^{fi}$ is the observed value of the i-th feature at k-th step, $\boldsymbol{x}^{fi}$ is the coordinate of the i-th feature, $w_x^k,\ w_y^k$ are the noises of the sensor in the x and y axes and are assumed to be zero-mean Gaussian.

5.1.3 Problem statement

The problem is to select a series of control inputs to generate a safe collision-free trajectory to cover the area of interest as quickly as possible, while performing 2D point feature based SLAM continuously with accurate localization and mapping results. The environment is assumed to contain a number of point features with unknown locations. The UAV can observe features within its sensor range. The boundary of the area of interest to be covered and mapped is known. Some no-fly zones are also known before performing the task. The whole scene is shown in Fig.

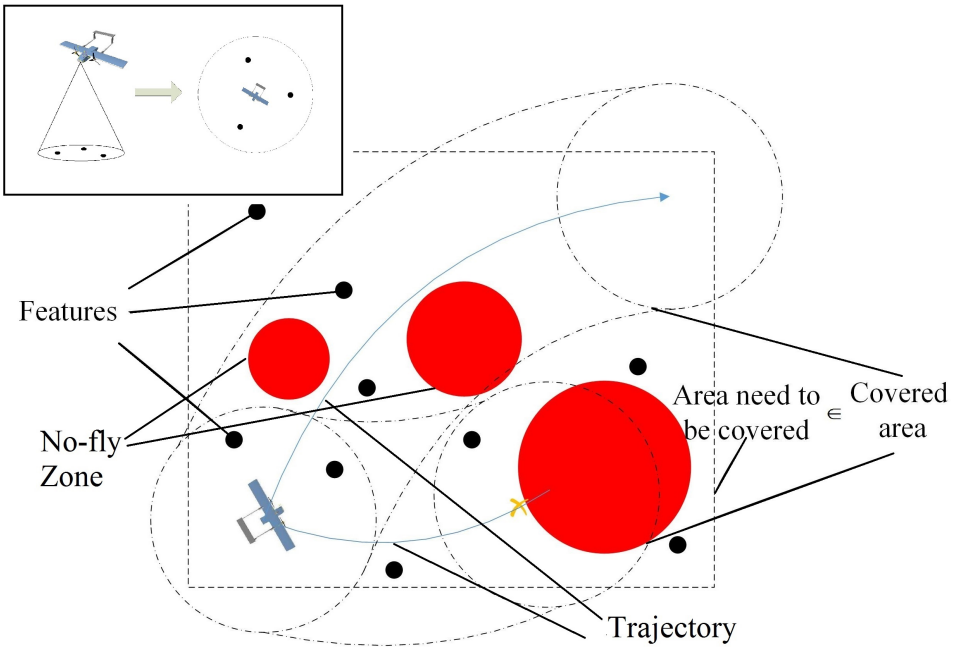


Figure 5.1 : SLAM task and coverage task

5.1.

5.2 MPC framework for uncertainty minimization task and coverage task

In this section, we present the formulation and the solution framework for the two tasks.

5.2.1 MPC framework for uncertainty minimization task

The problem is considered in the framework of MPC by an L-step look-ahead. Similar to [68], the objective function of the uncertainty minimization task is based on the generalized belief at the L-th planning step:

$$\begin{aligned} f_a(u_{k:k+L-1}) &= f_a(gb(\boldsymbol{X}_{k+L})) = -\log\left(\det(\boldsymbol{\mathcal{I}}_{k+L}(\boldsymbol{X}_{k+L}^{opt}))\right) \\ \boldsymbol{X}_{k+L}^{opt} &= \underset{\boldsymbol{X}_{k+L}}{\operatorname{argmin}} \ -\log(p(\boldsymbol{X}_{k+L}|\boldsymbol{Z}_{1:k}, u_{0:k+L-1}, \boldsymbol{Z}_{k+1:k+L}) \end{aligned}, \tag{5.3}$$

where $p(\star|\bullet)$ means the probability to estimate $\star$ by the condition $\bullet$. $u_{0:k+L-1}$ denotes the $k+L$ control inputs $u_0, \ \cdots, u_{k+L-1}$, $\boldsymbol{X}_{k+L}$ is the real state vector including the $k+L-1$ poses and the coordinates of the features, $gb(\boldsymbol{X}_{k+L})$ is the Gaussian belief of $\boldsymbol{X}_{k+L}$, $gb(\boldsymbol{X}_{k+L}) \approx \mathcal{N}(\boldsymbol{X}^{opt}_{k+L}, \boldsymbol{\mathcal{I}}_{k+L}(\boldsymbol{X}^{opt}_{k+L})^{-1})$, $\boldsymbol{\mathcal{I}}_{k+L}(\boldsymbol{X}^{opt}_{k+L})$ means the FIM corresponding to state $\boldsymbol{X}^{opt}_{k+L}$, $\boldsymbol{Z}_{1:k}$ and $\boldsymbol{Z}_{k+1:k+L}$ denote the observed values from step 1 to k and step $k+1$ to $k+L$, respectively. It is noted that the measurements $\boldsymbol{Z}_{1:k}$ are the real measurements from the sensors. However, the the measurements $\boldsymbol{Z}_{k+1:k+L}$ are the predicted measurements based on the prior information.

Using Bayes' theorem, we have:

$$\begin{aligned} &p(\boldsymbol{X}_{k+L}|\boldsymbol{Z}_{1:k}, u_{0:k-1}, \boldsymbol{Z}_{k+1:k+L}, u_{k:k+L-1}) \\ =&\frac{p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{X}_{k+L}, \boldsymbol{Z}_{1:k}, u_{0:k+L-1})p(\boldsymbol{X}_{k+L}|\boldsymbol{Z}_{1:k}, u_{0:k+L-1})}{p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{Z}_{1:k}, u_{0:k+L-1})}. \end{aligned} \tag{5.4}$$

We now take the assumption that the prior $p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{Z}_{1:k}, u_{0:k+L-1})$ is uninformative and is set to 1. This assumption is fairly standard in inference, see [68]. The above equation (5.5) can thus be rewritten as:

$$\begin{aligned} &p(\boldsymbol{X}_{k+L}|\boldsymbol{Z}_{1:k}, u_{0:k-1}, \boldsymbol{Z}_{k+1:k+L}, u_{k:k+L-1}) \\ \propto&p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{X}_{k+L}, \boldsymbol{Z}_{1:k}, u_{0:k+L-1})p(\boldsymbol{X}_{k+L}|\boldsymbol{Z}_{1:k}, u_{0:k+L-1}). \end{aligned} \tag{5.5}$$

The future measurement $\boldsymbol{Z}_{k+1:k+L}$ is a probabilistic event. If the j-th feature is outside the sensor range from the actual position of the robot, it will not be observed. We know nothing about the unmapped features, so, in the MPC predicted framework, we only consider the measurements from the features already mapped by the SLAM method. Even for the mapped features, the exact values of the future measurement $\boldsymbol{Z}_{k+1:k+L}$ are also unknown because of the uncertainty, in order to solve this problem, we assume the measurement $\boldsymbol{Z}_{k+1:k+L}$ is perfect (zero-innovation). In other words, the predicted measurement will not effect the SLAM estimated result

but change the FIM. Using the Markov property, we obtain the predicted state vector $\boldsymbol{X}_{k+L}^{opt}$:

$$\boldsymbol{X}_{k+L}^{opt} = \underset{\boldsymbol{X}_{k+L}}{\operatorname{argmin}}\, p(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k}, u_{0:k-1}) p(\boldsymbol{X}_{k+1:k+L}|\boldsymbol{X}_k, u_{k:k+L-1}). \tag{5.6}$$

Equation (5.6) includes two parts. The first part is a classical SLAM problem, and the second is a prediction process with zero-innovation probabilistic measurement $\boldsymbol{Z}_{k+1:k+L}$. Assuming that the SLAM result at k-th step is $\boldsymbol{X}_k^{opt}$, the predicted future pose $\boldsymbol{X}_{k+L}^{opt}$ will be:

$$\boldsymbol{X}_{k+L}^{opt} = \begin{bmatrix} \boldsymbol{X}_k^{opt} \\ \boldsymbol{x}_{k+1}^{opt} \\ \vdots \\ \boldsymbol{x}_{k+L}^{opt} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_k^{opt} \\ f_v(\boldsymbol{x}_k^{opt}, u_k) \\ \vdots \\ f_v(\boldsymbol{x}_{k+L-1}^{opt}, u_{k+L-1}) \end{bmatrix}, \tag{5.7}$$

where $f_v(\star)$ is the motion equation without uncertainty shown in (5.1) and $\boldsymbol{x}_{k+i}^{opt}$ is the predicted pose at step $k+i$.

Finally, using the *D-opt* optimality criterion, the optimization problem for uncertainty minimization will be:

$$\begin{aligned} \min_{u_{k:k+L-1}} \quad & f_a(u_{k:k+L-1}) = -\log\left(\det(\boldsymbol{\mathcal{I}}_{k+L}(\boldsymbol{X}_{k+L}^{opt}))\right) \\ \text{s.t. } & \boldsymbol{X}_{k+L}^{opt} = \left(\boldsymbol{X}_k^{opt\top}, \cdots, f_v(\boldsymbol{x}_{k+L-1}^{opt}, u_{k+L-1})^\top\right)^\top \\ & |u_{k+i}| \le C_u, \quad i = 0, \cdots, L-1 \\ & \{\boldsymbol{x}_k^{opt}, \cdots, \boldsymbol{x}_{k+L-1}^{opt}\} \notin \mathit{Obstacle}, \end{aligned} \tag{5.8}$$

where C_u is the control limitation, *Obstacle* is the set of no-fly zones. The expectation operator in the objective function is approximated by a single sample considering maximum-likelihood ideal observations.

5.2.2 Coverage task under uncertainty based on MPC

As the robot moves, the covered area will increase based on its actual trajectory and sensing range. However, only an estimated position (with uncertainty) is available from the SLAM result, so firstly we need to formulate an expression for area covered under uncertainty.

At the k-th step, the major and minor axes of the 95% confidence ellipse $S_{95\%}$ of a robot position $\widehat{\boldsymbol{x}}_k^{opt} \sim \mathcal{N}(\bar{\boldsymbol{x}}_k^{opt}, \boldsymbol{\mathcal{I}}_k^{-1})$ is $2\sqrt{5.991}\lambda_1(\boldsymbol{\mathcal{I}}_k^{-1})$ and $2\sqrt{5.991}\lambda_2(\boldsymbol{\mathcal{I}}_k^{-1})$, where $\lambda_1(\boldsymbol{\mathcal{I}}_k^{-1})$ and $\lambda_2(\boldsymbol{\mathcal{I}}_k^{-1})$ are respectively the major and secondary eigenvalues of its covariance matrix. The orientations of the axes of the ellipse are the eigenvectors of the covariance matrix. The range of the sensor is set as R_s. If the discrete coordinates of the points in the confidence ellipse $S_{95\%}$ are $(x_i^s, y_i^s), i = 1, ...,$ the coordinates $(x_i^c, y_i^c), i = 1, ...$ of the bound of the covered area S_i^c at the i-th step in the worst case will be (Fig. 5.2):

$$\begin{aligned} x_i^c &= \frac{l_s - R_s}{l_s}\bar{x}_k^{opt} + \frac{R_s}{l_s}x_i^s \\ y_i^c &= \frac{l_s - R_s}{l_s}\bar{y}_k^{opt} + \frac{R_s}{l_s}y_i^s \quad , \\ l_s &= \sqrt{(\bar{x}_k^{opt} - x_i^s)^2 + (\bar{y}_k^{opt} - y_i^s)^2} \end{aligned} \tag{5.9}$$

where $\bar{\boldsymbol{x}}_k^{opt} = (\bar{x}_k^{opt}, \bar{y}_k^{opt})^T$.

The coverage task is also built based on the MPC framework. The objective

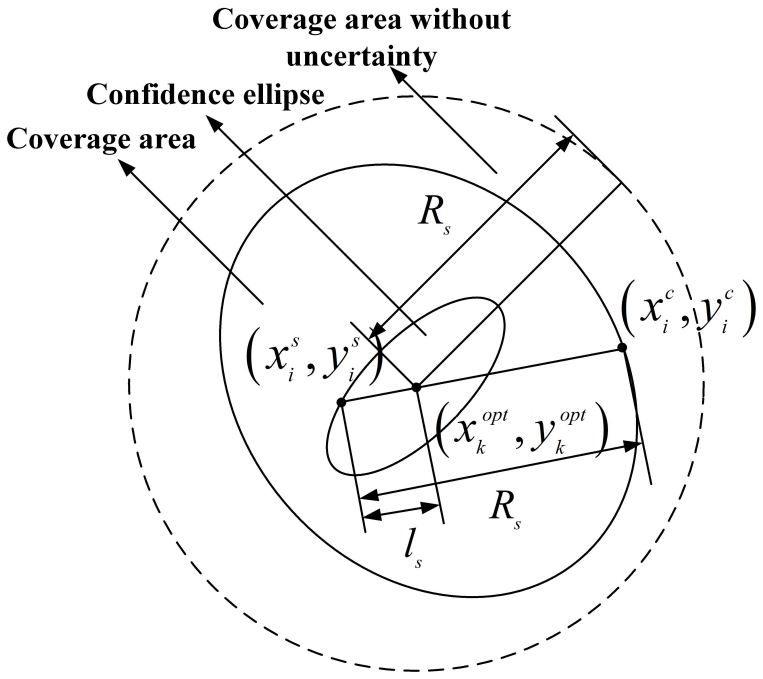


Figure 5.2 : The coverage area under uncertainty

function is defined as:

$$
\begin{aligned}
f_c(u_{k:k+L_c-1}) &= \begin{cases} \frac{1}{f_A(A_{add})} & A_{add} \neq \varnothing \\ \min\limits_{i=1,\ldots,N_r}(l_i^c) & A_{add} = \varnothing \end{cases} \\
A_{add} &= \left(\bigcup_{i=1}^{k+L_c} S_i^c - \bigcup_{i=1}^{k} S_i^c \right) \bigcap Space \quad , \\
l_i^c &= \sum_{j=1}^{L_c} \sqrt{(\bar{x}_{k+j}^{opt} - x_i^r)^2 + (\bar{y}_{k+j}^{opt} - y_i^r)^2}
\end{aligned}
\tag{5.10}
$$

where $f_A(\star)$ is a function that computes the area of $\star$, L_c is the number of the look-ahead steps, (x_i^r, y_i^r) is the centroid of the i-th uncovered area, N_r is the number of remaining uncovered areas, $Space$ means the whole planning space, A_{add} is the new covered space executing L_c look-ahead control inputs, and l_i^c represents the minimum distance between the estimated position of the robot and the centroids of the uncovered areas.

We now formulate the MPC problem for the coverage task:

$$\begin{aligned} &\min_{u_{k:k+L_c-1}} \quad f_c(u_{k:k+L_c-1}) \\ &s.t. \boldsymbol{X}^{opt}_{k+L_c} = (\boldsymbol{X}^{opt}_{k}, \cdots, f_v(x^{opt}_{k+L-1}, u_{k+L_c-1}))^T. \\ &\quad |u_{k+i}| \leq C, \quad i = 0, \cdots, L_c - 1 \end{aligned} \tag{5.11}$$

5.2.3 Solution framework

Considering its highly-nonlinear and non-analytical formula, we use the SQP approach to compute a sub-optimal solution for the coverage MPC problem. Then, for the uncertainty minimization MPC problem, we first approximate it into a least-squares problem through the graph topology and solve it using convex relaxation and optimization method.

Because the physical meanings and the units of the two objective functions are significantly different, we use a switching mechanism instead of the common weight way. The switching mechanism is shown as follows [30]:

$$\begin{aligned} u_r &= \begin{cases} u_c, & Index_1 \leq C_1^{index}, \\ c_c u_c + c_a u_a, & Index_1 \in (C_1^{index}, C_2^{index}), \\ u_a, & Index_1 \geq C_2^{index}, \end{cases} \\ Index_1 &= \sum_{i=1}^{N_f} (\lambda_i^{fx} + \lambda_i^{fy}), \end{aligned} \tag{5.12}$$

where u_c is the first control input of the coverage task, u_a is the first control input of the *D-opt* MPC solution, c_a and c_c are weights. C_1^{index} and C_2^{index} are two switching indexes, λ_i^{fx} and λ_i^{fy} are the eigenvalues of the covariance matrices of the i-th features at the x and y axes based on the SLAM result, N_f is the number of detected features.

5.3 Solution for the uncertainty minimization problem

In this section, the D-opt MPC problem will be solved by the graphic structure of 2D feature-based SLAM, the convex relaxation and the SVD-based rounding method.

5.3.1 Graphic structure and uncertainty bounds of 2D feature-based SLAM

Assumption 1: Isotropic Noise. We assume that the noises in (5.1) and (5.2) satisfy: $\begin{pmatrix} \delta x_k & \delta y_k \end{pmatrix}^\top \sim \mathcal{N}\left(\mathbf{0}, \delta_v(k)^2 \boldsymbol{I}_{2\times 2}\right)$, $\begin{pmatrix} w_x^k & w_y^k \end{pmatrix}^\top \sim \mathcal{N}\left(\mathbf{0}, \delta_f(k)^2 \boldsymbol{I}_{2\times 2}\right)$, $\delta\omega_k \sim \mathcal{N}(0, \delta_\theta(k)^2)$.

The state vector is rewritten as $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{p}_v^\top & \boldsymbol{\theta}_v^\top \end{pmatrix}^\top$ where $\boldsymbol{p}_v$ represents the coordinates of the UAV and the features, and $\boldsymbol{\theta}_v$ represents the orientation angles of the UAV. The stacked vector of translational and rotational measurements is $\boldsymbol{z} = \begin{pmatrix} \boldsymbol{z}_p^\top & \boldsymbol{z}_\theta^\top \end{pmatrix}^\top$.

Based on (5.1) and (5.2), the measurement model, $\boldsymbol{z} = h(\boldsymbol{X}) + \boldsymbol{\delta}$, can be expressed as,

$$\begin{bmatrix} \boldsymbol{z}_p \\ \boldsymbol{z}_\theta \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}^\top (\boldsymbol{A_g} \otimes \boldsymbol{I}_{2\times 2})^\top & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A_p}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\theta} \end{bmatrix} + \boldsymbol{\delta}, \tag{5.13}$$

where $\boldsymbol{A_p}$ and $\boldsymbol{A_g}$ respectively are the reduced incident matrices corresponding to the pose graph and the whole graph, here $\otimes$ is the Kronecker product, $\boldsymbol{R} \triangleq \mathrm{diag}(\boldsymbol{R}_1, \boldsymbol{R}_2, ..., \boldsymbol{R}_m)$ where m is the number of measurements, $\boldsymbol{R}_i$ is the rotation matrix corresponding to the UAV orientation making the i-th observation, $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma}$ can be written as $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\Sigma}_p \otimes \boldsymbol{I}_{2\times 2}; \boldsymbol{\Sigma}_\theta)$, where $\boldsymbol{\Sigma}_p = \mathrm{diag}(\delta_v(1)^2, \cdots, \delta_v(k)^2, \delta_f(1)^2, \cdots, \delta_f(m-k)^2)$, and $\boldsymbol{\Sigma}_\theta = \mathrm{diag}(\ \delta_\theta(1)^2, \cdots, \delta_\theta(k)^2\ )$.

The FIM is given by:

$$\boldsymbol{\mathcal{I}}(\boldsymbol{X}) = \boldsymbol{J}(\boldsymbol{X})^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{J}(\boldsymbol{X}) \tag{5.14}$$

where $\boldsymbol{J}(\boldsymbol{X}) \triangleq \frac{\partial h(\boldsymbol{X})}{\partial \boldsymbol{X}}$ is the Jacobian matrix of the measurement model. Introduce (5.13) into (5.14), we can get the FIM:

$$\boldsymbol{\mathcal{I}}(\boldsymbol{X}) = \begin{bmatrix} \boldsymbol{L}^{\boldsymbol{g}}_{\boldsymbol{w_p}} \otimes \boldsymbol{I}_{2\times 2} & \boldsymbol{A}^{\boldsymbol{g}}_{\boldsymbol{w_p}} \otimes \boldsymbol{I}_{2\times 2}\boldsymbol{\Gamma}\triangle_{\boldsymbol{w_p}} \\ (\boldsymbol{A}^{\boldsymbol{g}}_{\boldsymbol{w_p}} \otimes \boldsymbol{I}_{2\times 2}\boldsymbol{\Gamma}\triangle_{\boldsymbol{w_p}})^{\top} & \triangle^{\top}_{\boldsymbol{w_p}}\triangle_{\boldsymbol{w_p}} + \boldsymbol{L}^{\boldsymbol{p}}_{\boldsymbol{w_\theta}} \end{bmatrix}, \tag{5.15}$$

where $\boldsymbol{L}^{\boldsymbol{g}}_{\boldsymbol{w_p}} = \boldsymbol{A_g}\boldsymbol{\Sigma}^{-\mathbf{1}}_{\boldsymbol{p}}\boldsymbol{A}^{\top}_{\boldsymbol{g}}$ is the reduced weighted Laplacian matrix of the whole graph. $\boldsymbol{A}^{\boldsymbol{g}}_{\boldsymbol{w_p}} = \boldsymbol{A_g}\boldsymbol{\Sigma}^{-\frac{\mathbf{1}}{\mathbf{2}}}_{\boldsymbol{p}}$ is the reduced weighted incidence matrix, $\boldsymbol{\Gamma}$ is given by, $\boldsymbol{\Gamma} \triangleq \boldsymbol{I}_{m\times m} \otimes \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $\boldsymbol{L}^{\boldsymbol{p}}_{\boldsymbol{w_\theta}}$ is the reduced weighted Laplacian matrix for pose graph, $\boldsymbol{A}^{\boldsymbol{p}}_{\boldsymbol{w_\theta}} = \boldsymbol{A_p}\boldsymbol{\Sigma}^{-\frac{\mathbf{1}}{\mathbf{2}}}_{\boldsymbol{\theta}}$, $\triangle_{\boldsymbol{w_p}} = \triangle\boldsymbol{\Sigma}^{-\frac{\mathbf{1}}{\mathbf{2}}}_{\boldsymbol{p}}$, $\triangle \in \mathbb{R}^{2m\times n}$ is a special structure. Suppose in the k-th measurement, the i_kth node has observed the j_k-th node. For a 2×1 block in $\triangle$,

$$\triangle_{2k-1:2k,i_k} = \boldsymbol{p}_{j_k} - \boldsymbol{p}_{i_k}. \tag{5.16}$$

The remaining elements in $\triangle$ are all zero.

Then, similar to [80], we can get the lower bounds of the FIM for 2D feature-based SLAM using the D-optimality criterion as:

$$\log(\det(\boldsymbol{\mathcal{I}}(\boldsymbol{X}))) \geq 2\log(\det(\boldsymbol{L}^{\boldsymbol{g}}_{\boldsymbol{w_p}})) + \log(\det(\boldsymbol{L}^{\boldsymbol{p}}_{\boldsymbol{w_\theta}})), \tag{5.17}$$

where $\boldsymbol{L}^{\boldsymbol{g}}_{\boldsymbol{w_p}} = \boldsymbol{A_g}\boldsymbol{\Sigma}^{-\mathbf{1}}_{\boldsymbol{p}}\boldsymbol{A}^{\top}_{\boldsymbol{g}}$ is the reduced weighted Laplacian matrix of the whole graph, $\boldsymbol{L}^{\boldsymbol{p}}_{\boldsymbol{w_\theta}} = \boldsymbol{A_p}\boldsymbol{\Sigma}^{-\mathbf{1}}_{\boldsymbol{\theta}}\boldsymbol{A}^{\top}_{\boldsymbol{p}}$ is the reduced weighted Laplacian matrix for pose graph, the simple proof process to get the FIM and the lower bound conclusion is given in the Appendix B.3.1.

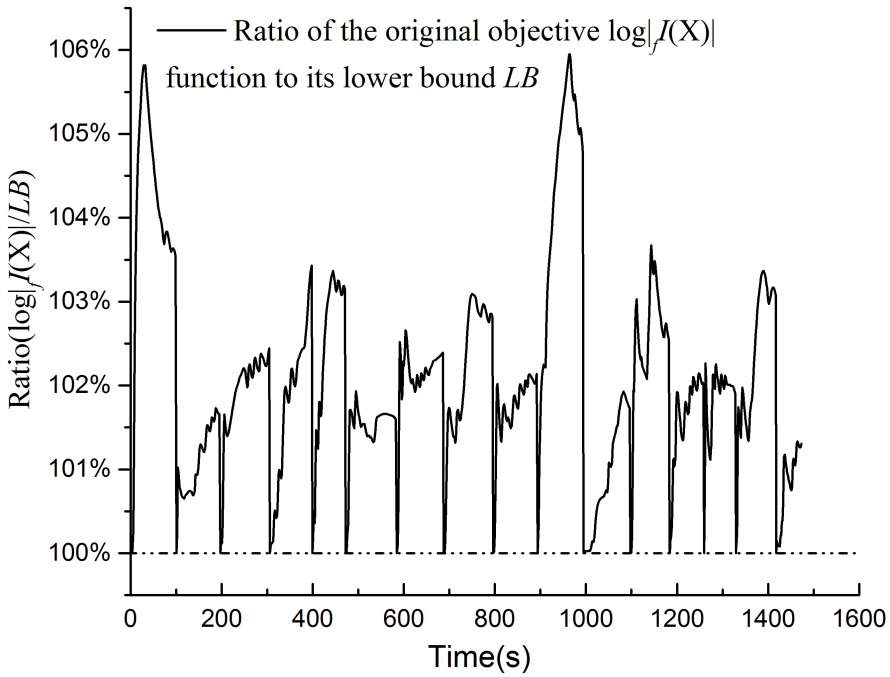


Figure 5.3 : The objective function values of the uncertainty mininization problem are very close to the lower bounds

5.3.2 A discussion on the D-opt objective function

We try to use the lower bound in (5.17) to replace the original objective function. We know that if the lower bound of the uncertainty can meet the requirements of the task, the original one will meet it unquestionably. Further, in most situations the value $\log(\det(\boldsymbol{\mathcal{I}}(\boldsymbol{X})))$ approaches the lower bound. To validate this result numerically, we performed an experiment using one of our simulation datasets. Fig. 5.3 shows that the real objective function value is very close to its lower bound.

Based on [80], $\det(\boldsymbol{L}^{\boldsymbol{g}}_{\boldsymbol{w_p}})$ and $\det(\boldsymbol{L}^{\boldsymbol{p}}_{\boldsymbol{w_\theta}})$ are equal to the weighted numbers of the spanning trees of the whole graph and the pose graph, respectively. However, the weighted number of the spanning trees of a graph is directly based on the number of the cycles in this graph. The whole graph including pose graph and measurements is much more important than the single pose graph, because there are no cycles in the pose graph for the feature-based SLAM usually. So we do not need to consider

the part $\det(\boldsymbol{L}^{\boldsymbol{p}}_{\boldsymbol{w_\theta}})$.

Because we know nothing about the non-detected features, the only way to improve the measurements probability is to make the future poses of the UAV get close to the detected features and make their distance smaller than the range of the sensor. So this distance needs to be minimized as an objective function. Even though every new measurement for the observed feature will result in at least one new cycle in the future graph, these cycles obtained by different old features are greatly different. In general, for the features which have been observed many times, adding a new measurement will not make a big contribution to decrease the uncertainty. So we will use a weight c_j to identify the different importance of the observed features.

Hence, for the omni-directional range sensors, we use a new least-squares function to replace the original objective function $-\log(\det(\boldsymbol{\mathcal{I}}_{k+L}(\boldsymbol{X}^{opt}_{k+L})))$:

$$\begin{aligned} &\sum_{i=1}^{L}\sum_{j=1}^{N_f} c_j \|\boldsymbol{x}^{opt}_{k+i} - \boldsymbol{x}^{opt}_{fj}\|^2, \\ c_j = &\frac{\log(\det(\widehat{\boldsymbol{L}}^{\boldsymbol{g}}_{\boldsymbol{w_p}}(j))) - \log(\det(\boldsymbol{L}^{\boldsymbol{g}}_{\boldsymbol{w_p}}))}{\sum_{j=1}^{N_f}(\log(\det(\widehat{\boldsymbol{L}}^{\boldsymbol{g}}_{\boldsymbol{w_p}}(j))) - \log(\det(\boldsymbol{L}^{\boldsymbol{g}}_{\boldsymbol{w_p}})))}, \end{aligned} \tag{5.18}$$

where $\boldsymbol{x}^{opt}_{fj}$ is the estimated coordinate of the observed features. $\widehat{\boldsymbol{L}}^{\boldsymbol{g}}_{\boldsymbol{w_p}}(j)$ is the new weighted Laplacian matrix of the whole graph of $\boldsymbol{X}^{opt}_{k+L}$ by adding a virtual future measurement between the j-th observed feature and the predicted future pose $\boldsymbol{x}^{opt}_{k+1}$, where

$$\begin{aligned} \widehat{\boldsymbol{L}}^{\boldsymbol{g}}_{\boldsymbol{w_p}}(j) &= \boldsymbol{A_g}(j)\boldsymbol{\Sigma}^{*}_{\boldsymbol{p}}(j)^{-1}\boldsymbol{A_g}(j)^{\top}, \\ \boldsymbol{A_g}(j) &= (\ \boldsymbol{A_g} \quad \boldsymbol{a}_j\ ), \end{aligned} \tag{5.19}$$

where $\boldsymbol{a}_j$ is a dimensional column vector that has only two non-zero elements, respectively -1 and 1. -1 locates at the k-th element and 1 locates at $k+j$-th element, $\boldsymbol{\Sigma}^{*}_{\boldsymbol{p}}(j) = \mathrm{diag}(\delta_v(1)^2, \cdots, \delta_v(k)^2, \delta_f(1)^2, \cdots, \delta_f(m-k)^2, {\delta_f^j}^2) \otimes \boldsymbol{I}_{2\times 2}$.

The main purpose of the new formulation is to get closed to the most important

features to add the measurements based on the larger information increment.

The coefficient c_j, which is computed before the optimization and used as a constant value in the least-square problem, is defined to simulate the future one-step measurement and embody the information increment. It is easy to known that, for the large-scale SLAM problem, even though the weighted Laplacian matrix $\widehat{\boldsymbol{L}}^{\boldsymbol{g}}_{\boldsymbol{w_p}}(j)$ has the smaller dimensions and the sparser structure than the FIM, its determinant is still a time-consuming computation process. In Section 5.4, the submap joining technology is used to limit the dimension of $\widehat{\boldsymbol{L}}^{\boldsymbol{g}}_{\boldsymbol{w_p}}(j)$. And we also presents a fast algorithm using the order re-use technology and the sparse Cholesky decomposition with a good fill-reducing permutation to compute this coefficient c_j shown in Algorithm 6.

Algorithm 6 Compute Coefficient c_j

Require: Two weighted Laplacian matrices $\widehat{\boldsymbol{L}}^{\boldsymbol{g}}_{\boldsymbol{w_p}}(j)$ and $\boldsymbol{L}^{\boldsymbol{g}}_{\boldsymbol{w_p}}$.
Ensure: Coefficient c_j

```
//Choose a fill-reducing permutation heuristic p
p ← COLAMD(L^g_{w_p})                    ▷ e.g., Column approximate minimum degree
p' ← [p; size(p) + 1]                    ▷ e.g., Reuse order for L̂^g_{w_p}(j)
//Sparse Cholesky factor based P and P'
C_1 ← SparseCholesky(L^g_{w_p}(p, p))
C_2 ← SparseCholesky(L̂^g_{w_p}(j)(p', p'))
//Compute coefficient c_j by eq. (5.18)
log(det(L^g_{w_p})) ← 2 · 3 · Σ_i log(C_1)_{i,i}
log(det(L̂^g_{w_p}(j))) ← 2 · 3 · Σ_i log(C_2)_{i,i}
return c_j ← (log(det(L̂^g_{w_p}(j))) − log(det(L^g_{w_p}))) / Σ_{j=1}^{m} (log(det(L̂^g_{w_p}(j))) − log(det(L^g_{w_p})))
```

Algorithm 6 can be computed much faster than the log-determinant function of the dense matrix, whose computation complexity is $O(n^3)$.

5.3.3 Transformation into a convex optimization problem

In this section, we will discuss a spectrahedral description method and a no-fly zone half-space representation to produce a convex problem. The new MPC problem can be rewritten as:

$$\begin{aligned} \min_{u_{k:k+L-1}} \quad & \sum_{i=1}^{L}\sum_{j=1}^{N_f} c_j \left\| \boldsymbol{x}_{k+i}^{opt} - \boldsymbol{x}_{fj}^{opt} \right\|^2 \\ \text{s.t.} \quad & \boldsymbol{x}_{k+i+1}^{opt} = \boldsymbol{x}_{k+i}^{opt} + \boldsymbol{R_{k+i}}\widehat{\boldsymbol{v}} \\ & \theta_{k+i+1}^{opt} = \theta_{k+i}^{opt} + u_{k+i}\Delta T \\ & |u_{k+i}| \leq C_u, i = 0, ..., L-1 \\ & \{\boldsymbol{x}_k^{opt}, \cdots, \boldsymbol{x}_{k+L-1}^{opt}\} \notin Obstacle, \end{aligned} \tag{5.20}$$

where $\boldsymbol{x}_{k+i}^{opt} = (x_{k+i}^{opt}, y_{k+i}^{opt})^\top$, $\boldsymbol{x}_{fj}^{opt} = (x_{fj}^{opt}, y_{fj}^{opt})^\top$.

This problem is not convex because of the rotation matrix $R_{k+i} \in SO(2)$ and the collision-free constraints. In order to use the convex optimization mehod, we need a spectrahedral description of conv $SO(2)$ and intoduce a no-fly zone half-space representation method.

Convex relaxation of rotation matrix

Theorem 8 (Spectrahedral description of conv $SO(2)$)**.** *[120] The convex hull of $SO(n)$ is a spectrahedron for all $n \in \mathbb{N}$. Explicitly*

$$\begin{aligned} conv\ SO(n) = \Big\{ \boldsymbol{X} \in \mathbb{R}^{n\times n} : & \begin{bmatrix} \boldsymbol{0} & \boldsymbol{X} \\ \boldsymbol{X}^\top & \boldsymbol{0} \end{bmatrix} \\ \preceq \boldsymbol{I}_{2n\times 2n}, & \sum_{i,j=1}^{n} \boldsymbol{A_{ij}}[\boldsymbol{RX}]_{ij} \preceq (n-2)\boldsymbol{I}_{2^{n-1}\times 2^{n-1}} \Big\}, \end{aligned} \tag{5.21}$$

where $\boldsymbol{A_{ij}}$ *is a collection of symmetric* $2^{n-1}\times 2^{n-1}$ *matrices,* $\boldsymbol{R} = diag(1, 1, \ldots, 1, -1)^{n\times n}$. *In the special cases* $n = 2$, *its representation can be simplified to:*

$$conv\ SO(2) = \left\{ \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \in \mathbb{R}^{2\times 2} : \begin{bmatrix} 1+c & s \\ s & 1-c \end{bmatrix} \succeq 0 \right\}. \tag{5.22}$$

Based on Theorem 1, let $c = u^*_{i,1}$ and $s = u^*_{i,2}$, the first constraint can be revised as:

$$\begin{aligned} x^{opt}_{k+i+1} &= x^{opt}_{k+i} + v u^*_{i,1}\Delta t, \\ y^{opt}_{k+i+1} &= y^{opt}_{k+i} + v u^*_{i,2}\Delta t, \\ &\begin{bmatrix} 1+u^*_{i,1} & u^*_{i,2} \\ u^*_{i,2} & 1-u^*_{i,1} \end{bmatrix} \succeq 0. \end{aligned} \tag{5.23}$$

The constraints $\theta^{opt}_{k+i+1} = \theta^{opt}_{k+i} + u_{k+i}\Delta t$ and $|u_{k+i}| \le C_u$ can be written as a unified form:

$$|\theta^{opt}_{k+i+1} - \theta^{opt}_{k+i}| \le C_u\Delta t,\ i = 0, ..., L-1. \tag{5.24}$$

Using the mean value theorem, we introduce new constraints to relax and replace the constraint (5.24):

$$\begin{aligned} |u^*_{i+1,1} - u^*_{i,1}| \le& |\sin(\theta^*)| C_u\Delta t \le C_u\Delta t, \\ |u^*_{i+1,2} - u^*_{i,2}| \le& |\cos(\theta^*)| C_u\Delta t \le C_u\Delta t, \end{aligned} \tag{5.25}$$

where $\theta^* \in (\theta^{opt}_{k+i+1}, \theta^{opt}_{k+i})$ if $\theta^{opt}_{k+i} > \theta^{opt}_{k+i+1}$ or $\theta^* \in (\theta^{opt}_{k+i}, \theta^{opt}_{k+i+1})$ if $\theta^{opt}_{k+i} < \theta^{opt}_{k+i+1}$.

This convex relaxation method based on the spectrahedral description can be used not only in the 2D situation. It can be easily applied in the 3D trajectory planning.

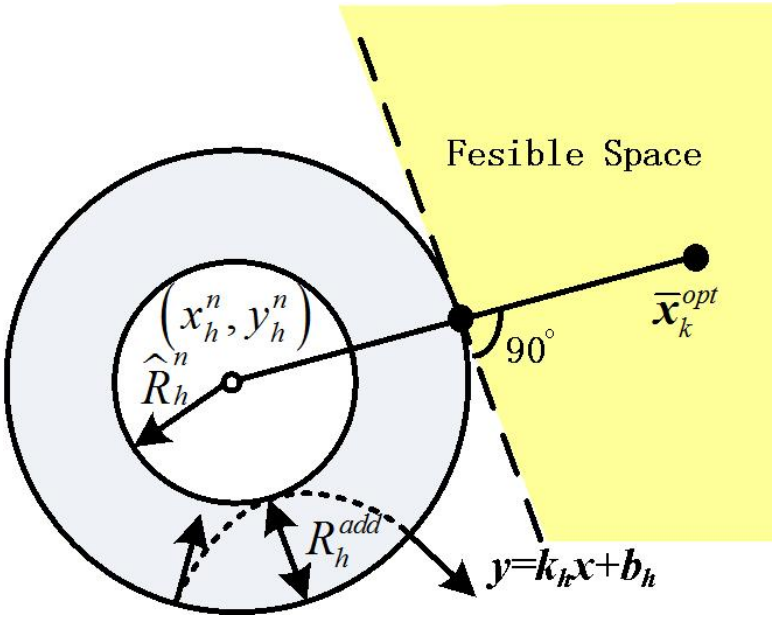


Figure 5.4 : Illustration of a convex half-space representation method

Non-convex no-fly zone constraints issue

The no-fly zone is a circular area which is a convex region. Its complementary set in the whole space is a non-convex region. The half-space representation method presents a good way to approximate the non-convex feasible region. The half-space representation method is firstly presented by [104]. We use a similar idea of this representation and add an emergency mode to improve its robustness in the MPC framework.

As shown in Fig. 5.4, the no-fly zone is firstly enlarged by an emergency safe range R_h^{add}, and then a tangent line $y = k_h x + b_h$ is obtained by the geometrical relationship between the coordinate of the center point of the no-fly zone (x_h^n, y_h^n), the UAV position $\boldsymbol{x}_k^{opt}$ and the radius $R_h^{all} = R_h^{add} + \widehat{R}_h^n$. We can use the following inequalities to limit the feasible space of the predicted future poses.

$$\begin{aligned} &(k_h x_{k+i}^{opt} + b_h - y_{k+i}^{opt})(k_h x_h^n + b_h - y_h^n) < 0, \\ &h = 1, 2, \cdots, N_o;\ i = 1, 2, \cdots, L, \end{aligned} \tag{5.26}$$

where N_o is the number of the no-fly zones in the area.

The emergency safe range R_h^{add} is used in some special situations. Because of the control limitation and the rounding operation shown in Section 5.3.4, even though the UAV $\boldsymbol{x}_k^{opt}$ locates in the feasible space, the future poses $\{\boldsymbol{x}_{k+1}^{opt}, \cdots, \boldsymbol{x}_{k+L}^{opt}\}$ may all or partly locate in the other side of the tangent line $y = k_h x + b_h$. We introduce the emergency safe range R_h^{add} to avoid these special situations. When the predicted future poses $\boldsymbol{x}_{k+i}^{opt}, i = 1, 2, \cdots, L$ are forced to locate in the emergency safe annulus, which means the convex method is insoluble, the optimization method needs to be stopped and replaced by a sharp curve at the maximum control input C_u and the best direction. The best direction means that the UAV can go outside the emergency safe annulus as soon as possible. The extreme case is when the orientation of the UAV faces to the center point of the no-fly zone (x_h^n, y_h^n) and the UAV locates on the boundary of the annulus. It needs the biggest R_h^{add}, which can be computed by:

$$R_h^{add} = \sqrt{\frac{2\widehat{R}_h^n V}{C_u} + \widehat{R}_h^{n2}} - \widehat{R}_h^n. \tag{5.27}$$

New convex problem

The new convex problem is:

$$\begin{aligned} \min_{u_{k:k+L-1}} \quad & \sum_{i=1}^{L} \sum_{j=1}^{N_f} c_j \left\| \boldsymbol{x}_{k+i}^{opt} - \boldsymbol{x}_{fj}^{opt} \right\|^2 \\ \text{s.t.} \quad & \text{Eq.}(5.23), \text{Eq.}(5.25), \text{Eq.}(5.26) \end{aligned} \tag{5.28}$$

This convex problem can be easily solved using the interior-point methods. This algorithm has been integrated in many convex toolboxes such as lsqlin function in MATLAB, CVX and CVXGEN. Because of its convexity, its solution will be globally optimal.

5.3.4 Candidate solution to original least-squares problem

Because of the relaxation in the above section, the solution to (5.28) may not be a feasible solution to the original non-convex least-squares problem. In this section, the way to generate a candidate solution is addressed. Suppose the solution of the new convex problem (5.28) is:

$$\begin{aligned} \boldsymbol{v}_1^* &= (\widehat{u}_{0,1}, \widehat{u}_{1,1}, ..., \widehat{u}_{L-1,1})^\top, \\ \boldsymbol{v}_2^* &= (\widehat{u}_{0,2}, \widehat{u}_{1,2}, ..., \widehat{u}_{L-1,2})^\top. \end{aligned} \tag{5.29}$$

Its corresponding solutions in conv $SO(2)$ are:

$$\boldsymbol{R}_{\boldsymbol{1}}^*, \cdots, \boldsymbol{R}_{\boldsymbol{i}}^* = \begin{bmatrix} \widehat{u}_{i-1,1} & -\widehat{u}_{i-1,2} \\ \widehat{u}_{i-1,2} & \widehat{u}_{i-1,1} \end{bmatrix}, \cdots, \boldsymbol{R}_{\boldsymbol{L}}^*. \tag{5.30}$$

The rounding procedure is to find the nearest valid rotation in $SO(2)$ to replace the solution shown in (5.30). The problem for conv $SO(n)$ has been defined and studied previously [117]:

$$\begin{aligned} &\pi_R : \text{conv } SO(n) \rightarrow SO(n), \\ &\pi_R(\boldsymbol{R}_{\boldsymbol{i}}^*) \in \min_{\widehat{\boldsymbol{R}}_{\boldsymbol{i}} \in SO(n)} \|\widehat{\boldsymbol{R}}_{\boldsymbol{i}} - \boldsymbol{R}_{\boldsymbol{i}}^*\|_F, \end{aligned} \tag{5.31}$$

where $\| \bullet \|_F$ is the Frobenius norm. The solution to optimization problem (5.31) is based upon the singular value decomposition of $\boldsymbol{R}_{\boldsymbol{i}}^*$: Let a diagonal matrix $\boldsymbol{S}$, unitary matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are the singular value decomposition of $\boldsymbol{R}_{\boldsymbol{i}}^*$ meets $\boldsymbol{R}_{\boldsymbol{i}}^* =$

$\boldsymbol{U} * \boldsymbol{S} * \boldsymbol{V}^\top$, the solution $\widehat{\boldsymbol{R}}_{\boldsymbol{i}}$ is [129]:

$$\begin{aligned} \widehat{\boldsymbol{R}}_{\boldsymbol{i}} &= \boldsymbol{U} * \boldsymbol{S}' * \boldsymbol{V}^\top, \\ \boldsymbol{S}' &= \begin{cases} \boldsymbol{I}_{n\times n} & if \ \det(\boldsymbol{U})\det(\boldsymbol{V}) = 1 \\ \text{diag}\{1, \cdots, 1, -1\}^{n\times n} & if \ \det(\boldsymbol{U})\det(\boldsymbol{V}) = -1. \end{cases} \end{aligned} \tag{5.32}$$

For the $SO(2)$ case, there is an equivalent simple form for the solution $\widehat{\boldsymbol{R}}_{\boldsymbol{i}}$:

$$\widehat{\boldsymbol{R}}_{\boldsymbol{i}} = \begin{bmatrix} \frac{\widehat{u}_{i-1,1}}{\sqrt{\widehat{u}_{i-1,1}^2+\widehat{u}_{i-1,2}^2}} & -\frac{\widehat{u}_{i-1,2}}{\sqrt{\widehat{u}_{i-1,1}^2+\widehat{u}_{i-1,2}^2}} \\ \frac{\widehat{u}_{i-1,2}}{\sqrt{\widehat{u}_{i-1,1}^2+\widehat{u}_{i-1,2}^2}} & \frac{\widehat{u}_{i-1,1}}{\sqrt{\widehat{u}_{i-1,1}^2+\widehat{u}_{i-1,2}^2}} \end{bmatrix}. \tag{5.33}$$

We can also get the control inputs of the original non-convex non-linear constrainted least-squares problem by:

$$\begin{aligned} u_{k+i} =& f_{round}\left(\theta_{k+i+1}^{opt} - \theta_{k+i}^{opt}\right) \\ =& f_{round}\left(\text{atan2}(\widehat{u}_{i+1,2}, \widehat{u}_{i+1,1}) - \text{atan2}(\widehat{u}_{i,2}, \widehat{u}_{i,1})\right), \end{aligned} \tag{5.34}$$

where atan2$(\bullet, \star)$ is the functionality of the multi-valued inverse tangent, $f_{round}(\star)$ is a piecewise function:

$$f_{round}(\star) = \begin{cases} -C_u, & \star/\Delta t < -C_u, \\ \star/\Delta t, & -C_u \leq \star/\Delta t \leq C_u, \\ C_u, & \star/\Delta t > C_u. \end{cases} \tag{5.35}$$

Then, we get the solution $(u_k, u_{k+1}, .., u_{k+L-1})^\top$.

5.4 Planing based on submap joining

Based on efficient nonlinear optimization, there are several state-of-the-art techniques to get SLAM solution for medium-large scenarios quickly: g2o [90] and iSAM2

[74]. Submap joning [146] is not an essential tool for SLAM. But it is very useful for the planning process in the active SLAM. We will firstly review Linear SLAM. Then we will show the benefit of planning in the submap and put forward some conclusions about Linear SLAM. Following, we will show how to use these conclusions in active SLAM. Finally, we will show some new problems caused by submap.

5.4.1 Linear SLAM

If we don't want to get the global map, we can finish the active SLAM in every submap without joining. However, having a global map from time to time is useful for improving the obstacle avoidance and area coverage. We can use Linear SLAM method to obtain the global map.

Assuming there are two submaps 1 and 2 with state vectors $\{\boldsymbol{x^{L1}}, \boldsymbol{x^{L2}}\}$ and their corresponding information matrices $\boldsymbol{\mathcal{I}}(\boldsymbol{x^{L1}})$ and $\boldsymbol{\mathcal{I}}(\boldsymbol{x^{L2}})$, we need to compute the state vectors and the information matrices of the joined global map. It is noted that the last pose of the sub-map 1 and the first pose of the sub-map 2 are the same pose.

Using Linear SLAM algorithm, the submap 1 is transformed based on the last pose. The new state vector based on last pose is $\boldsymbol{x^G}$. The function between $\boldsymbol{x^{L1}}$ and $\boldsymbol{x^G}$ is defined as: $\boldsymbol{x^G} = f_t(\boldsymbol{x^{L1}})$. Then, the new information matrix of new global map $\boldsymbol{x^G}$ can be written as:

$$\boldsymbol{\mathcal{I}}_{L1} = \boldsymbol{J}_{f_t}(\boldsymbol{x^{L1}})^\top \boldsymbol{\mathcal{I}}(\boldsymbol{x^{L1}}) \boldsymbol{J}_{f_t}(\boldsymbol{x^{L1}}), \tag{5.36}$$

where $\boldsymbol{J}_{f_t}(\star)$ are the Jacobi matrices of the function f_t.

Based on Linear SLAM and the last pose, the submap joining problem becomes

a linear least squares problem to minimize the following objective function:

$$\min \ f_L(\boldsymbol{X}) = \left\| \boldsymbol{x^G} - \boldsymbol{X_1} \right\|^2_{\boldsymbol{\mathcal{I}_{L1}}} + \left\| \boldsymbol{x^{L2}} - \boldsymbol{X_2} \right\|^2_{\boldsymbol{\mathcal{I}_{L2}}}. \tag{5.37}$$

where $\boldsymbol{\mathcal{I}_{L2}} = \boldsymbol{\mathcal{I}}(\boldsymbol{x^{L2}})$, $\boldsymbol{X}$ is a subset of $[\boldsymbol{X_1}^\top, \boldsymbol{X_2}^\top]^\top$ based on the date association. This linear least squares problem can be written in a compact form as:

$$\min \ \|\boldsymbol{A_Z X} - \boldsymbol{Z}\|^2_{\boldsymbol{I_Z}}, \tag{5.38}$$

where $\boldsymbol{A_Z}$ is a sparse coefficient matrix shown the date association relationship between $\boldsymbol{X}$ and $[\boldsymbol{X_1}^\top, \boldsymbol{X_2}^\top]$. $\boldsymbol{\mathcal{I}_Z}$ is the corresponding information matrix given by

$$\boldsymbol{\mathcal{I}_Z} = \begin{bmatrix} \boldsymbol{\mathcal{I}}_{L1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{I}}_{L2} \end{bmatrix}. \tag{5.39}$$

The corresponding global FIM is:

$$\boldsymbol{\mathcal{I}_{all}} = \boldsymbol{A_Z}^\top \boldsymbol{\mathcal{I}_Z} \boldsymbol{A_Z}. \tag{5.40}$$

5.4.2 Importance of using submap planning idea

In the active SLAM process, we only perform planning and estimating in every submap. The benefit of using submap planning is discussed as follows:

Control switching problem

This chapter presents a simple and efficient control switching strategy (5.12) to coordinate the two problems both. This mechanism will be efficient based on the suitable switching indexes λ_i^{fx} and λ_i^{fy}. However, if local submap is not used, increased with the growing of the length of the robot trajectory, the uncertainties of the new features will become large because of the poor accuracy of the later poses.

It is very hard to find the constant switching indexes to assign the control inputs. The whole active SLAM task will fail. We can only try to find increasing functions to replace these indexes, which is difficult and ill-suited. Planning in the submap is a good way to limit the range of the eigenvalues of the covariance matrices, because the new covariance matrix is built based on the zero uncertainty of the first pose of each submap. Then, the switching indexes λ_i^{fx} and λ_i^{fy} can be set very easily and suitably based on the size of the submap.

Local minimum problem

From (5.28), we can learn that the objective function will encourage the robot to move closer to a changing center point $\boldsymbol{p}_c$ decided by the features: $\boldsymbol{x}_{f1}^{opt}, \cdots, \boldsymbol{x}_{fm}^{opt}$ and the weighted $c_1, \cdots, c_m$. If there is no submap, with the large number of old features, the moving range of the point $\boldsymbol{p}_c$ will become smaller and smaller, which leads to a local minimum point. The UAV may fly as a circle around this point. This problem will greatly slow down the active SLAM process, reduce the efficiency of the algorithm, and even cause the failure of the whole task. The submap is a good way to get out of the local minimum problem by producing the new center point $\boldsymbol{p}_c$ in each submap, and meanwhile, it will enlarge its moving range.

Planning and estimating complexity

Even using some powerful SLAM tools, the submap can help to further improve the running-time ability of the SLAM with an acceptable information loss. For the planning, the submap can help to limit the size of the objective function and the number of the no-fly zone avoidance constraints. Especially in the large-scale scene, the useless distant features and no-fly zones will greatly increase the planning complexity. So the submap planning is a good way to reduce the optimization complexity.

5.4.3 Bound of the eigenvalues of the FIM of the global map

In this section, we will discuss the relationship between the eigenvalues of the FIM of the local submaps and global joining map. This conclusion can be used to quickly check the accuracy of the SLAM result without using joining process.

We can get the following conclusions:

Conclusion 1: The matrix $\boldsymbol{A_Z}^\top \boldsymbol{A_Z}$ is a diagonal matrix.

Conclusion 2: The eigenvalues $\lambda_i(\boldsymbol{\mathcal{I}_Z})$ of the submap1 and submap 2 and the eigenvalues $\lambda_i(\boldsymbol{\mathcal{I}_{all}})$ of the joining global map satisfy:

For $i = 1, 2, \cdots, k$

$$\lambda_i(\boldsymbol{\mathcal{I}_Z})\widehat{\lambda}(\boldsymbol{A_Z}) \leq \lambda_i(\boldsymbol{\mathcal{I}_{all}}) \leq \lambda_{n-k+i}(\boldsymbol{\mathcal{I}_Z})\widetilde{\lambda}(\boldsymbol{A_Z}), \tag{5.41}$$

where $\widehat{\lambda}(\boldsymbol{A_Z})$ and $\widetilde{\lambda}(\boldsymbol{A_Z})$ are the minimal and maximum eigenvalues of matrix $\boldsymbol{A_Z^\top A_Z}$.

The simple proof process to get conclusion 1 and 2 is given in Appendix B.3.2.

5.4.4 Active SLAM with submap planning technology

There is a key question in the application of the submap planning technology: After building several submaps, when is the best time to generate the global map using Linear SLAM? The joining process is time-consuming, so we need to avoid the frequent call of the submap joining algorithm. However, because we need to estimate the global positions of the obstacles and the coverage area by the coordinate transform without joining process, the global uncertainty is generated by the linear superposition, which leads the approximate estimated uncertainty is bigger than the actual situation after using linear SLAM. This bigger approximate uncertainty, which needs the redundant safe ranges in Eq. (5.27), may affect the performance

of the coverage task and the obstacle avoidance task. So it is necessary to use the joining process at the suitable time in the planning process. The conclusion 2 is a good way to check the accuracy of the global SLAM result and identify when to finish the joining in the planning process.

When the bounds in Eq. (5.42) meets:

$$\frac{\sum_{i=1}^{k} \lambda_{n-k+i}(\boldsymbol{\mathcal{I}_Z})\widetilde{\lambda}(\boldsymbol{A_Z})}{\sum_{i=1}^{k} \lambda_i(\boldsymbol{\mathcal{I}_Z})\widehat{\lambda}(\boldsymbol{A_Z})} \leq D \tag{5.42}$$

where D is a coefficient to change the gap between the lower bound and the upper bound.

It means that the global map is well-bounded by these two bounds in a small range. We do not need to finish the submap joining algorithm. Otherwise, we can use Linear SLAM to generate the global map. This process is finished in the independent CPU core, which is parallel to the main active SLAM program. It will not affect the run-time ability of the active SLAM.

5.4.5 New problems caused by submap joining

Because of the submap, several new issues are introduced in the problem solving process.

The first problem is how to finish the no-fly zone avoidance problem in the submap. It is easy to know that the avoidance is based on the global coordinate relationship between the UAV and the no-fly zones. However, the joining process is time-consuming if this process is used at every step. So, without using the joining process, we need to estimate the coordinates of the no-fly zones in every submap by:

$$\begin{aligned}
\boldsymbol{x}_{\boldsymbol{N_i^{sub}}}^{\boldsymbol{o_h}} &= \boldsymbol{R}_{N_i^{sub}-1}{}^{\top}(\boldsymbol{x}_{\boldsymbol{N_i^{sub}-1}}^{\boldsymbol{o_h}} - \boldsymbol{x}_{N_i^{sub}-1}), \\
&\dots \\
\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{o_h}} &= \boldsymbol{R}_1{}^{\top}(\boldsymbol{x}_{\boldsymbol{h}}^{\boldsymbol{n}} - \boldsymbol{x}_1),
\end{aligned} \tag{5.43}$$

where $\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{o_h}}$ is the xy coordinate of the h-th no-fly zone in the i-th submap, $\boldsymbol{x}_{\boldsymbol{h}}^{\boldsymbol{n}}$ means the coordinate of the h-th no-fly zone in the global map.

Similar to this problem, the next problem is how to finish the coverage task based on the submap. It is easy to know that the covered area is obtained based on the global coordinates of the robot. Because of the submap joining method, we cannot obtain the accurate global coordinates of the robot without the joining process. However, the joining process is time-consuming if this process is used at every step. So we need to estimate the pose and its uncertainty without performing the joining. We assume that at step k the robot locates at N_i^{sub}-th submap and sub_j-th step of the submap. The estimated coordinate $\boldsymbol{x}_i^{\boldsymbol{sub_j}},\ i \in \{1, 2, ..., N_i^{sub}\}$ and its corresponding 2×2 covariance matrix $\boldsymbol{Cov}_{\boldsymbol{i}}^{\boldsymbol{sub_j}}$ in i-th submap coordinate system can be calculated by:

$$\begin{aligned}
\boldsymbol{x}_{\boldsymbol{N_i^{sub}-1}}^{\boldsymbol{sub_j}} &= \bar{\boldsymbol{R}}_{N_i^{sub}-1}\boldsymbol{x}_{\boldsymbol{N_i^{sub}}}^{\boldsymbol{sub_j}} + \bar{\boldsymbol{x}}_{N_i^{sub}-1} \\
&\vdots \\
\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{sub_j}} &= \bar{\boldsymbol{R}}_1\boldsymbol{x}_{\boldsymbol{2}}^{\boldsymbol{sub_j}} + \bar{\boldsymbol{x}}_1 \\
\boldsymbol{x}_k^{opt} &= \boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{sub_j}} \\
\boldsymbol{Cov}_{\boldsymbol{N_i^{sub}-1}}^{\boldsymbol{sub_j}} &= \bar{\boldsymbol{R}}_{N_i^{sub}-1}\boldsymbol{Cov}_{\boldsymbol{N_i^{sub}}}^{\boldsymbol{sub_j}}\bar{\boldsymbol{R}}_{N_i^{sub}-1}^{-1} + \overline{\boldsymbol{Cov}}_{\boldsymbol{N_i^{sub}-1}}^{\boldsymbol{sub_j}} \\
&\vdots \\
\boldsymbol{Cov}_{\boldsymbol{1}}^{\boldsymbol{sub_j}} &= \bar{\boldsymbol{R}}_1\boldsymbol{Cov}_{\boldsymbol{2}}^{\boldsymbol{sub_j}}\bar{\boldsymbol{R}}_1^{-1} + \overline{\boldsymbol{Cov}}_{\boldsymbol{1}}^{\boldsymbol{sub_j}} \\
\boldsymbol{Cov}_{\boldsymbol{k}}^{\boldsymbol{opt}} &= \boldsymbol{Cov}_{\boldsymbol{1}}^{\boldsymbol{sub_j}}
\end{aligned}, \tag{5.44}$$

where $\bar{\boldsymbol{R}}_i$ are respectively the rotation matrices of the last pose of the i-th submap, $\bar{\boldsymbol{x}}_i$

is the xy coordinate of the last pose of the i-th submap, $\overline{\boldsymbol{Cov}}_{\boldsymbol{i}}^{\boldsymbol{sub_j}}$ is the corresponding 2×2 covariance matrix. So with the current pose in the first local frame $\boldsymbol{x}_k^{opt}$ and its covariance matrix $\boldsymbol{Cov}_{\boldsymbol{k}}^{\boldsymbol{opt}}$, we can get the covered area of the robots based on submap without joining at every step.

The other question is that the features detected and mapped in the other submaps will be regarded as a new feature in the new planning problem. It may lead to the UAV continuously visiting these old features as the unknown features. In order to solve this problem, after opening a new submap, we need to judge whether the feature has been detected and whether its uncertainty has been reduced to an acceptable level in the other submaps or not. If a feature meets this condition, it will not be used in the objective function of the uncertainty minimization task.

5.5 Whole algorithm

The pseudocode of whole active SLAM method is shown as follows:

Algorithm 7 Active SLAM based on submap joining, convex optimization and graph toplogy

Require: Area to be covered *Space*; Vehicle parameters: Velocity v, Sensor range R_s, Control limitation C_u; No-fly zone size R_h^n and position $\boldsymbol{x_h^n}$; Other setting parameters.

Ensure: Estimated poses and mapped features (SLAM results).

1: **repeat**
2: //Get model and measurement
3: Move the robot based on dynamic model with noises
4: Get the measurements from sensor model
5: //SLAM
6: Date association
7: Solve SLAM problem by GN by MATLAB-Graph-Optimization package
8: //Switching mechanism
9: **if** $Index_1 \geq C_2^{index}$ **then**
10: //Active SLAM task
11: Build weighted Laplacian matrix $\widehat{\boldsymbol{L}}_{\boldsymbol{w_p}}^{\boldsymbol{g}}$
12: Identify the mapped feature with a good accuracy in other submaps and mark them
13: Compute coefficient c_j by Algorithm 1 in [30]
14: Built convex optimization problem by Eq. (25) in [30], solve and round it
15: **elseif** $Index_1 \geq C_2^{index}$ **then**
16: //Coverage task
17: Built coverage optimization by Eq. (16) in [30] and solve it by SQP
18: **else**
19: //Both tasks
20: Solve active SLAM by line 10-14 and coverage task by line 15-16 both
21: //Submap joining
22: Whether to open a new submap by current pose
23: Whether to use Linear SLAM to finish map joining
24: $t \leftarrow t + \triangle t$
25: **until** Believable covered area is large enough

5.6 Simulations and experiments

Some simulation and experimental results are shown to verify the effectiveness of the presented algorithm.

5.6.1 Active SLAM using proposed method

All simulations and experimental implementations are performed on a HP EliteDesk 800 G2 desktop with an Intel Core i5-6500 3.20 GHz processor and 8 GB of RAM running Windows 10 Enterprise. The UAV, using a 20 m limited range omnidirectional sensor, moves in a 100×100 m square-shaped space with the known bounds, 30 unknown static features and 10 known static no-fly zones. Specifically, it moves at 1 m/s and its control input ω_k is limited in $[-0.3, 0.3]$ rad/s. Synthetic errors, with a Gaussian distribution, are generated for the odometry model of the UAV ($\delta_v(k) = 0.1$ m and $\delta_\omega(k) = 0.1$ rad) and the sensor measurements ($\delta_f(k) = 0.3$ m). 10 different random features and no-fly zones set names as dataset 1-10. The other parameters are set as: $c_a = 0.85$, $c_c = 0.15$, $C_1^{index} = 0.06N_f$ m, $C_2^{index} = 0.1N_f$ m, the number of the pose of the submap is 50. When the whole area is covered more than 98% percent and all features are detected, the simulation will stop. The final result of the proposed method using dataset 1 is shown in Fig. 5.5.

In Fig. 5.5, the blue circles are the 3σ covariance ellipses of the features. It shows that the UAV finishes the SLAM task with good accuracy. The area of interest is also covered by the sensor safely.

5.6.2 Comparisions

The effect of the submap joining method

Linear SLAM method can limit the size of the SLAM problem with an acceptable information loss. Here we compare the performance when different submap sizes are used. Except the submap size and the features, the other settings are the same as the settings of the simulation described in Section 5.6.1. Because without using the submap the estimation process will become slow, its stop condition is that the simulation time reaches 2000s. Then, the comparison results are presented in Fig. 5.6.

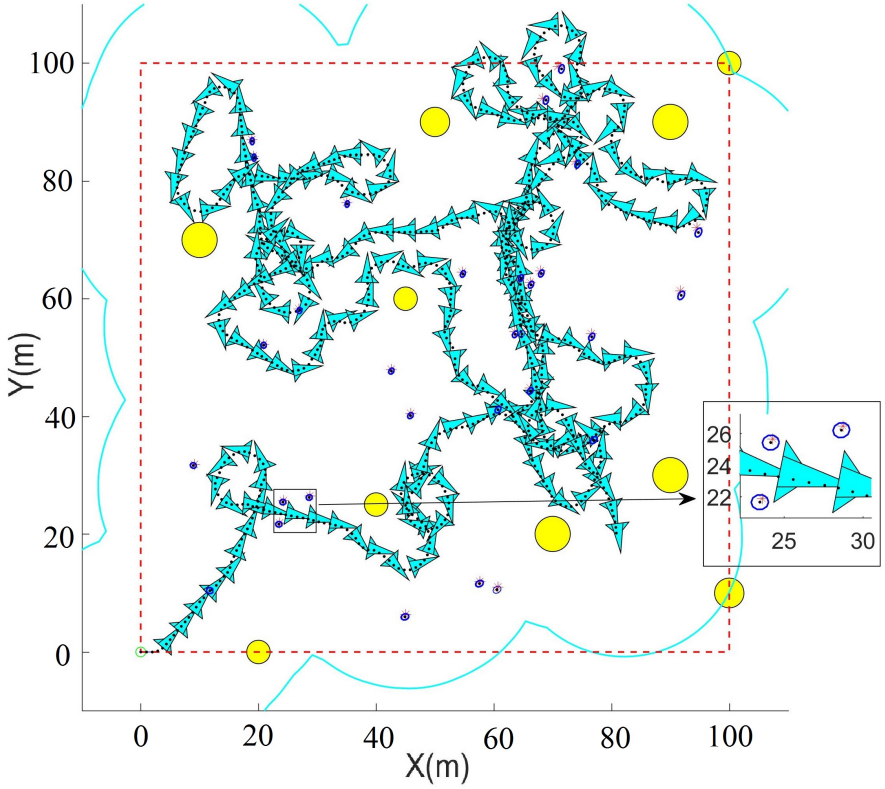


Figure 5.5 : Final results for dataset 1 (The blue triangles show the real trajectory of the UAV at each 5 steps. The black points show the estimated trajectory and features obtained by SLAM. The red star points are the real positions of the features. The yellow circles are the no-fly zones.)

Because of the different number of poses involved, we only compare the information matrix of the features by the *D-opt* optimality criteria $\log(\det(\boldsymbol{\mathcal{I}}_{\boldsymbol{f}}(\boldsymbol{X})))$. All simulations using submaps finish the coverage task and detect all features. In Fig. 5.6, we can also find that, with the increase of the submap size, the SLAM process will become a time-consuming process because of the larger numbers of estimated variables.

For the non-submap simulation, we present a trajectory result in Fig. 5.7. Let's analyze its reason. At the beginning, the switching indexes are suitable. With the growing of the moving distance, because of the lack of submap, the large uncertainty of the new features, such as A, B and C features, cannot trigger the control switching,

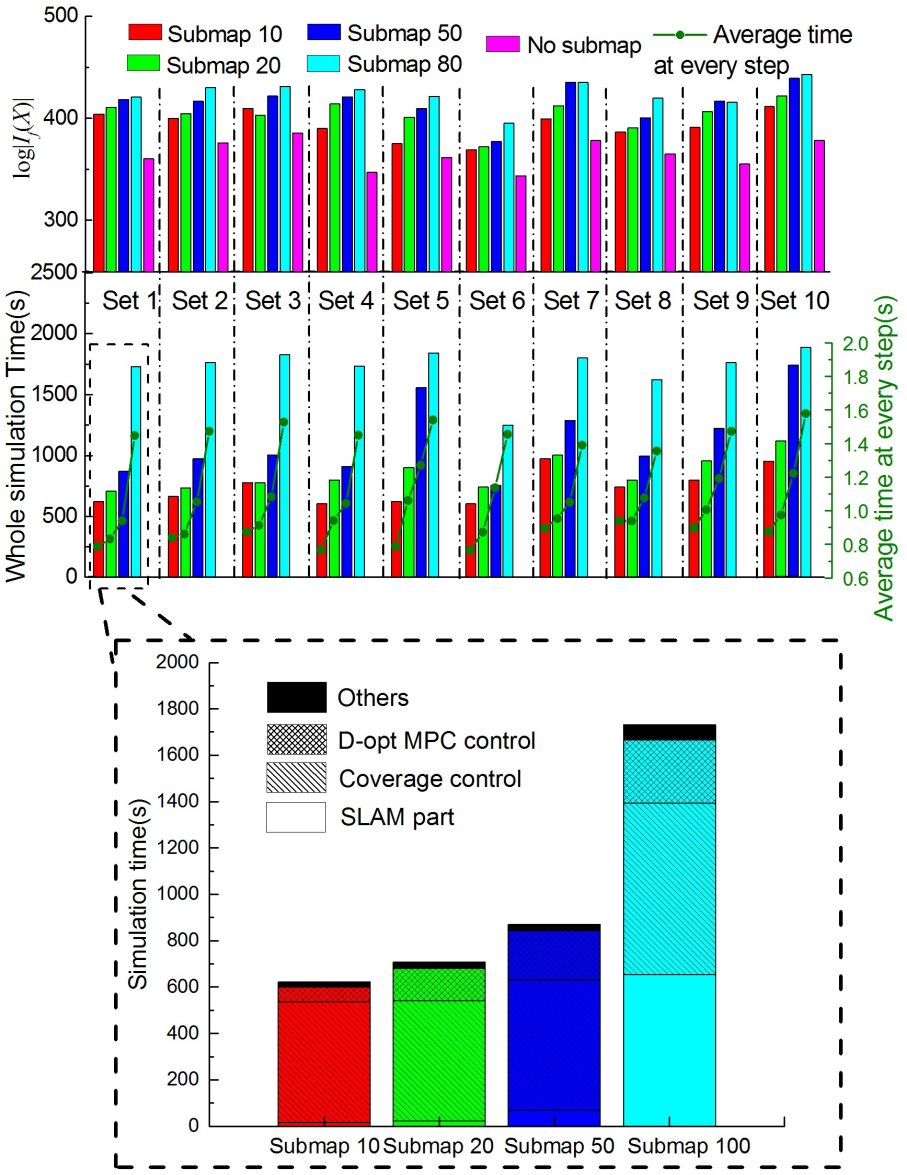


Figure 5.6 : Comparision results using different sizes of submaps (In 3000s, the simulation without using submap does not finish the coverage task. the covered percents from set 1 to 10 are respectively 30.8%, 33.1%, 39.1%, 34.6%, 26.1%, 39.1%, 33.2%, 41.2%, 31.3% and 37.3%.)

based on (5.12). The UAV keeps finishing the uncertainty minimization task among the known features, which leads no new features are detected, meanwhile. The UAV drops into a local minimum position (in grid-lines). Because the distances between the local minimum position and the marginal features, such as A, B, C and D features, are larger than the sensor range. So the UAV moving process cannot

add any measurement for these features. Meanwhile, some features locate in the bule box have been observed too many times, which cannot make a big contribution to improve the objective function. So many measurements are redundant and the measurements of some features (A, B and C features) are insufficient, which leads to the bad objective function value. So in short, the submap is indispensable for this solution framework.

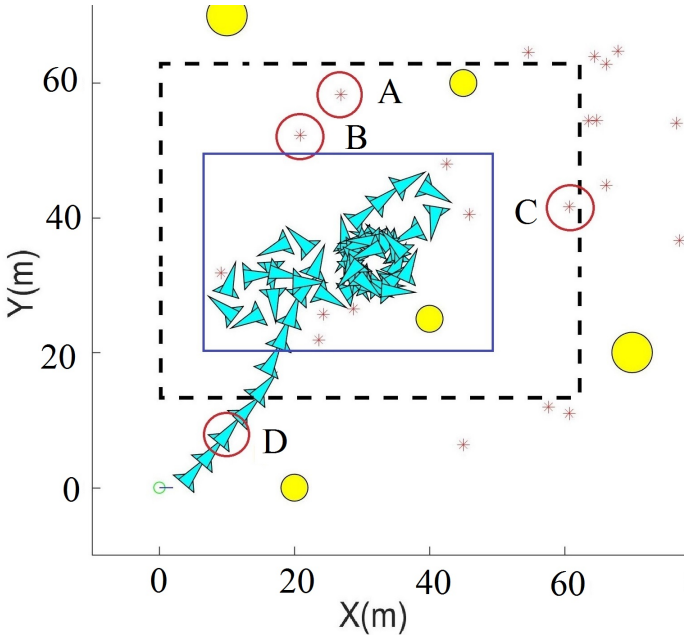


Figure 5.7 : Result without using submaps (Without the submap, the algorithm cannot finish the coverage task.)

Effect of graphical approximation and convex method

We use the graphical method to approximate the original *D-opt* objective function and solve the problem by the convex method. In this part, we solve the uncertainty minimization by the SQP method with the original objective function [30] and the greedy algorithm with 60 uniform candidate actions. The initial inputs of the SQP method is the solutions of the greedy method. The results based on dataset 1 with different look-ahead steps are shown in Fig. 5.8.

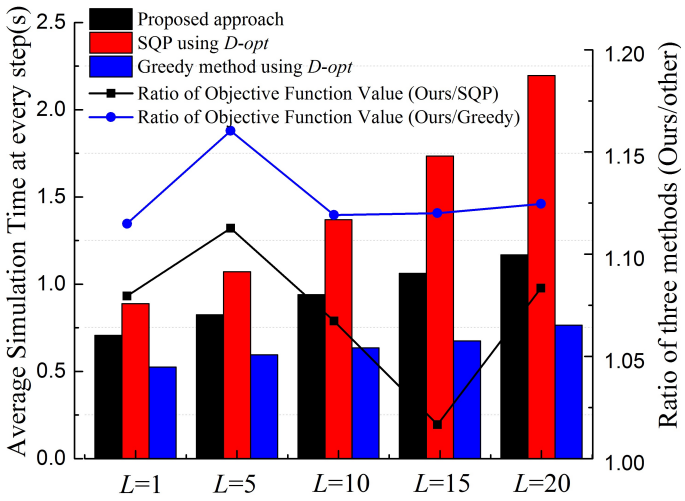


Figure 5.8 : Comparision results using graphical approximation and convex method with SQP method. greedy method and original objective function(Histogram shows the time and line graph shows the ratio)

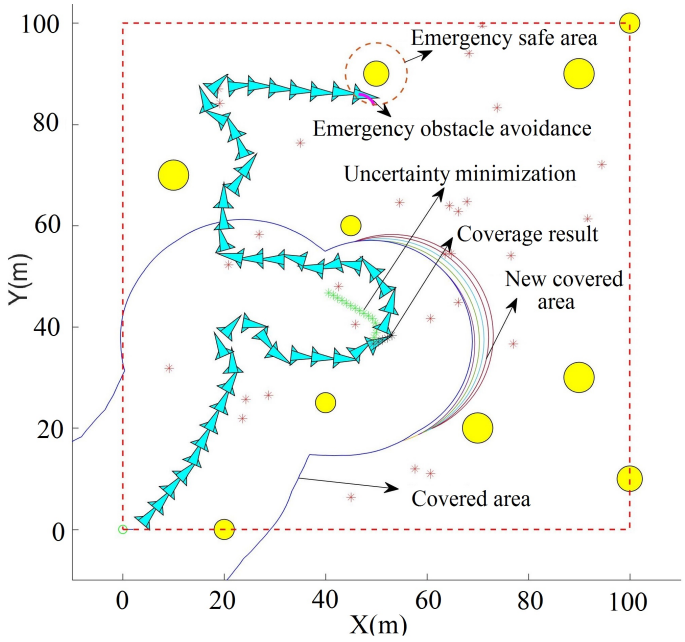


Figure 5.9 : Trajectory and optimization outputs (The green and black dotted line respectively indicate the optimization results of the rounding convex method in 15 look-ahead steps and the coverage task in 5 look-ahead steps.)

Because the original objective function is a non-convex complex function whose solving difficulty grows with the look-ahead step L, we can see the estimation and planning time grow quickly. Because of the convexity, the new problem can be

solved in polynomial time. Meanwhile, many local minimums were obtained when using the SQP method [29]. So its final solution is worse than the one using our proposed method.

5.6.3 Discussion about control inputs

In this part, we will discuss the real active control input u_r obtained by two different MPC problems and the emergency operations.

Control switching

Fig. 5.9 shows the result using dataset 1 at the 80-th step and the 144-th step, including the real UAV trajectory and covered area. We can see that the uncertainty minimization results lead the UAV to loop-closure and the coverage results make the UAV visit the unknown space. At the 144-th step, the UAV needs to finish an emergency obstacle avoidance operation, shown as the peach trajectory. Based on the suitable switching indexes C_1^{index} and C_2^{index}, we can switch the control inputs easily. In the simulation of Fig. 5.5 (dataset 1), the ratios of using the emergency avoidance control, the coverage control, the uncertainty minimization or the combination control as the final acting control u_r in are 5.36%, 26.08%, 21.86% and 46.70%. Four different control inputs are used rather than the single input.

Effect of no-fly zones

Compared with the collision-free environment, the emergency operations, which are not obtained by an optimization process, will reduce the optimality of the solution. So the proportion of the emergency operations cannot be too large. The proportions of the control inputs using the emergency operations based on different dateset are 5.36%, 3.35%, 3.85%, 2.11%, 4.127%, 1.61%, 5.61%, 4.99%, 4.41% and 6.84%. They are all small ($< 8\%$). So the solution optimality is only slightly affected by the emergency avoidance operations.

5.6.4 Off-line experimental verification

In this part, we try to use a real quad-rotor UAV in a small indoor environment to verify the flyable of the final trajectory and also show the good performance of the SLAM result. Because our coverage task is an off-line algorithm and our current code is based on MATLAB, we can only finish an experiment based on the off-line active SLAM result right now.

Based on the global vision localization system, we can get the ground truth of the UAV poses and the coordinate of the features. Then, we can compute the local measurement using the models Eq. (5.1) and (5.2) based on the poses and the features adding some Gaussian noises. These noises, which are equal to the simulation data in the active SLAM, are generated previously. After following the whole trajectory obtained by the off-line active SLAM results, we can get all measurements similar to the simulation but including the real dynamic model. Using these measurements, we can build and solve the SLAM problem. The ground truth and estimated result are plotted in Fig. 5.10. It also shows the composite photograph during the flying progress. Compared with the result only using coverage task, the objective function of our method has 9.3% advantage in this small area.

5.7 Summary

In this chapter, we have formulated the *D-opt* uncertainty minimization problem and the coverage problem with pose uncertainty in the submaps by the MPC framework. Linear SLAM submap joining method is used to ensure the effectiveness of the proposed method and reduce the estimation and planning time. We have presented a new non-convex constrained least squares problem to replace the traditional *D-opt* uncertainty minimization problem by using the distance function and the lower bound of the D-optimality criterion. We also have proposed a convex planning approach for solving this new problem. A theorem is presented and proved

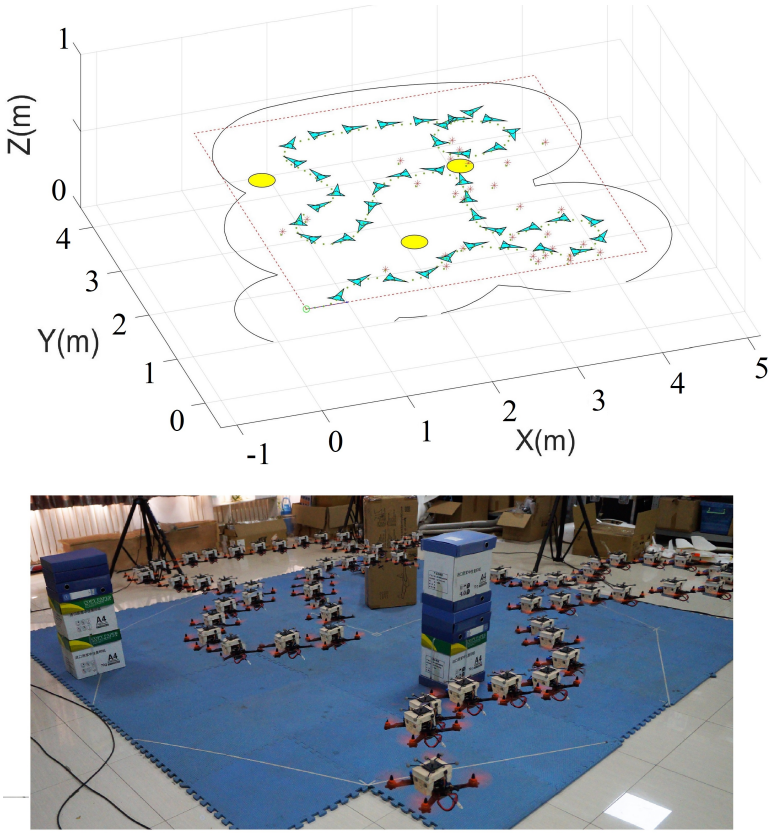


Figure 5.10 : Ground truth (Blue vehicles), estimated result (Green way-points) and flying photograph

to show the bound of the eigenvalues of the information matrix of the joined global map. We test our approach in realistic simulation scenarios; experimental results show that the approach is able to deal with the coverage tasks in an unknown environment with a good SLAM result. Compared with the SQP-based method, the approach produces better results with higher efficiency.

Chapter 6

On-line 3D Active Pose-graph SLAM based on Key Poses using Graph Topology and Sub-maps

As shown in Chapter 2, in 3D case, the computational complexity is one of the key problems in current active SLAM research area. The traditional optimal design metrics are really expensive for large-scale 3D problem, which make the on-line planning is really hard to achieve. Our goal in this chapter aims to investigate the graph structure of SLAM and introduce two novel and efficient planning metrics into active SLAM to improve the accuracy of the SLAM result and finish previously defined task in 3D environment. In this chapter, we propose an on-line pose-graph active SLAM algorithm based on graph topology, key poses and sub-maps.

6.1 Problem description

Assume that the robot is performing some tasks in an unknown environment such as coverage, exploration or search. Our active pose-graph SLAM framework is used to reduce pose uncertainty and further helps to finish the original tasks. The active SLAM problem considered in this chapter is to mostly retain the main originally designed path while finding good additional loop-closure trajectories between the current last pose and the previous poses to reduce the uncertainty of pose-graph SLAM subject to resource limitations.

For example, suppose the robot is performing a coverage task in an unknown space. When the pose uncertainty in pose-graph SLAM is large, in moving from the current last pose, the robot needs to revisit a previous pose and then return back to

perform the original task. Fig. 6.1 shows a system using our method. Compared to the MPC framework [30], this method can be easily applied in different tasks, and does not need to run in every step.

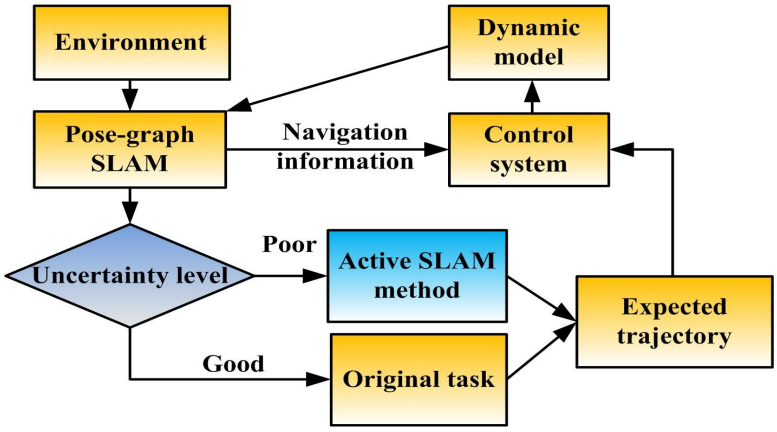


Figure 6.1 : Role of our active SLAM method in robot system

6.2 Pose-graph SLAM

6.2.1 Synchronization on $\mathbb{R}^n \times SO(n)$

Considering the pose-graph SLAM problem in (3.6), the SE-sync method [116] is applied to solve it. This method is one of the state-of-the-art algorithms, which shows the outstanding computational efficiency and certifiably globally optimal property. After the computational enhancements, it is as fast as some highly optimized libraries, like g2o [90].

6.2.2 FIM and optimality design metrics

The FIM $\boldsymbol{\mathcal{I}}$ of the pose-graph SLAM result is a tool to assess the quality of the measurement network. Based on $\boldsymbol{\mathcal{I}} = \boldsymbol{J}^\top \boldsymbol{\Sigma} \boldsymbol{J}$, in which $\boldsymbol{J}$ is the Jacobian matrix for (B.51), the FIM $\boldsymbol{\mathcal{I}}_{2D}$ for 2D SLAM with block-isotropic Gaussian noise is shown in

previous chapter:

$$\boldsymbol{\mathcal{I}}_{2D} = \left[\begin{array}{c:c} \boldsymbol{L}_w^{\mathbb{R}^2} & \triangle_w^\top \\ \hdashline \triangle_w & \boldsymbol{L}_w^{SO(2)} + \mathrm{diag}\{\psi_1, \cdots, \psi_{n_p}\} \end{array}\right], \tag{6.1}$$

where $\boldsymbol{L}_w^{\mathbb{R}^2}$ and $\boldsymbol{L}_w^{SO(2)}$ are the reduced weighted Laplacian matrices after using the Kronecker product operation, $\psi_i = \sum_{j \in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2, i = 1, \cdots, n_p$.

The D-optimality and T-optimality design metrics are two important metrics directly using the FIM without the inverse operation. The D-optimality and T-optimality metrics are, respectively, to maximize the log-determinant function and the trace function $\log(\det(\boldsymbol{\mathcal{I}}_{\boldsymbol{nD}})), \mathrm{trace}(\boldsymbol{\mathcal{I}}_{nD})$.

6.3 Relation between design metrics and graph topology

In this chapter, the D- and T-optimality design metrics are shown to have a strong relationship with the graph structures, including weighted node degree and weighted tree-connectivity, of the SLAM measurement network.

6.3.1 T-optimality metric and weighted node degree

Based on (6.1), both 2D and 3D situation, we can get the T-optimality design metric of the FIM:

$$\mathrm{trace}(\boldsymbol{\mathcal{I}}_{nD}) = \mathrm{trace}(\boldsymbol{L}_w^{\mathbb{R}^n}) + \mathrm{trace}(\boldsymbol{L}_w^{SO(n)}) + \sum_{i=1}^{n_p} \sum_{j \in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2. \tag{6.2}$$

For many real-world pose-graph SLAM problems, compared with $\mathrm{trace}(\boldsymbol{L}_w^{\mathbb{R}^n})$ and $\mathrm{trace}(\boldsymbol{L}_w^{SO(n)})$, the last term $\sum_{i=1}^{n_p} \sum_{j \in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2$ is relatively small, so

we have:

$$\begin{aligned}\text{trace}(\boldsymbol{\mathcal{I}}_{nD}) \approx &\text{trace}(\boldsymbol{L}_w^{\mathbb{R}^n}) + \text{trace}(\boldsymbol{L}_w^{SO(n)}) \\ = &\sum_{i=1}^{n_p} \sum_{j \in V_i} \left(\omega_{ij} + n\delta_{ij}^{-2}\right),\end{aligned} \tag{6.3}$$

where $\sum_{j \in V_i} \left(\omega_{ij} + n\delta_{ij}^{-2}\right)$ is the weighted value of the graph edge connected with the i-th node, named weighted node degree. It is easy to find that maximizing the T-optimality metric is almost equivalent to maximizing the weighted node degree. The weighted node degree is a computationally efficient metric with a similar effect for the T-optimality-based active SLAM problem.

6.3.2 D-optimality metric and weighted tree-connectivity

In (3.34), the lower and upper bounds of the D-optimality metric of the FIM are shown as:

$$\begin{aligned}&\mathcal{L} \leq \log(\det(\boldsymbol{\mathcal{I}}_{2D})) \leq \mathcal{U} \\ &\mathcal{L} = \log(\det(\boldsymbol{L}_w^{\mathbb{R}^2})) + \log(\det(\boldsymbol{L}_w^{SO(2)})) \\ &\mathcal{U} = \log(\det(\boldsymbol{L}_w^{\mathbb{R}^2})) + \sum_{i=1}^{n} \log(\lambda_i(\boldsymbol{L}_w^{SO(2)}) + \lambda_\infty),\end{aligned} \tag{6.4}$$

We assume that the SLAM result is 'near-optimal'. For many real-world datasets, compared with the eigenvalues of the rotation group $\lambda_i(\boldsymbol{L}_w^{SO(2)})$, the term λ_∞ is relatively small, so we have:

$$\mathcal{U} \approx \mathcal{L} \Rightarrow \log(\det(\boldsymbol{\mathcal{I}}_{nD})) \approx \mathcal{L}. \tag{6.5}$$

It is easy to find that optimizing the D-optimality metric is almost equivalent to optimizing the lower bound of the FIM. The lower bound $\mathcal{L}$ is known as the weighted tree-connectivity of the SLAM measurement network.

The computational complexity of the weighted tree-connectivity is much smaller than the original metric due to its lower dimension and sparser structure. It can be

computed much faster than the log-determinant function of the dense matrix $O(n_p^3)$.

6.3.3 Comparison among the four metrics

The weighted node degree and the weighted tree-connectivity metrics have similar performance and better computational efficiency than T- and D-optimality, respectively. This conclusion is verified in Section 6.6.1. For two new metrics, from the computational complexity point of view, the weighted node degree is cheaper than the weighted tree-connectivity. However, from the effectiveness point of view, we have found that the weighted tree-connectivity can identify the uncertainty levels of two graphs with similar weighted node degree. This conclusion is verified in Section 6.6.1. Thus, weighted tree-connectivity is a relatively expensive metric with good-performance.

Based on the above discussion, we have a basic idea for determining the loop-closure trajectory. Weighted node degree is a good metric to be used in large-scale search for rough candidate actions, and weighted tree-connectivity is suitable for sophisticated search within a small elite group.

6.4 On-line active SLAM framework

6.4.1 Fast covariance recovery to trigger active SLAM method

In this chapter, we use the covariance $\widehat{\boldsymbol{C}}_{i'i'}$, which is the block diagonal-elements of the covariance matrix $\boldsymbol{C}$, of the last pose (i'-th) as the index of uncertainty. Meanwhile, to ensure that the robot continues to make progress towards its task, we need to limit the additional time spent by considering how often active SLAM is triggered, the maximal length of the re-visiting path, and so on. When $\text{trace}(\widehat{\boldsymbol{C}}_{i'i'}) \geq Index_1$ and the robot has enough resources (e.g., time or energy), we begin to use active SLAM to find the best loop-closure trajectory. It is well-known that the FIM $\boldsymbol{F} = \boldsymbol{\mathcal{I}}_{nD}$ and the covariance matrix $\boldsymbol{C}$ follows $\boldsymbol{C} = \boldsymbol{F}^{-1}$. However, computing the

inverse of a large matrix is very time-consuming. Luckily, We only need the block diagonal elements corresponding to the last pose. Using the recursive formula [65], the specific elements from the covariance matrix can be efficiently calculated by the square root information matrix $\bar{\boldsymbol{H}}$ ($\boldsymbol{F} = \bar{\boldsymbol{H}}^\top \bar{\boldsymbol{H}}$):

$$\begin{aligned} \boldsymbol{C}_{ii} =& \frac{1}{\bar{\boldsymbol{H}}_{ii}}(\frac{1}{\bar{\boldsymbol{H}}_{ii}} - \sum_{k=i+1, \bar{\boldsymbol{H}}_{ik}\neq 0}^{n} \bar{\boldsymbol{H}}_{ik}\boldsymbol{C}_{ki}) \\ \boldsymbol{C}_{ij} =& \frac{-1}{\bar{\boldsymbol{H}}_{ii}}(\sum_{k=i+1, \bar{\boldsymbol{H}}_{ik}\neq 0}^{j} \bar{\boldsymbol{H}}_{ik}\boldsymbol{C}_{kj} + \sum_{k=j+1, \bar{\boldsymbol{H}}_{ik}\neq 0}^{n} \bar{\boldsymbol{H}}_{ik}\boldsymbol{C}_{jk}), \end{aligned} \tag{6.6}$$

where $\star_{ij}$ means the (i, j)-th element of the matrix $\star$. The specific operations, which are similar to the fast recovery method in SLAM++ [66], using the AMDP and a hash table [73] is shown in Algorithm 8.

Algorithm 8 Fast covariance recovery for last pose

Require: Fisher information matrix $\boldsymbol{F}$
Ensure: The covariance block corresponding to the last pose
1: **Step 1**: Create $\boldsymbol{F}' = \boldsymbol{F}$, delete its column and row corresponding to the last pose, and get $\boldsymbol{F}'_{reduce}$;
2: **Step 2**: $\boldsymbol{p}_{order} = AMDP(\boldsymbol{F}'_{reduce})$;
3: **Step 3**: $\boldsymbol{p}_{order} = [\boldsymbol{p}_{order}; \boldsymbol{p}^{last}_{order}]$ % $\boldsymbol{p}^{last}_{order}$ is the column/row order for the last pose
4: **Step 4**: $\boldsymbol{H} = Cholesky(\boldsymbol{F}(\boldsymbol{p}_{order}, \boldsymbol{p}_{order}))$ % Sparse Cholesky decomposition;
5: **Step 5**: Covariance recovery by (6.6) using the hash table;

6.4.2 RRT-connect and weighted node degree for initial search

First, to get $N_c + 1$ potential old loop-closure poses by dividing the path into several small segments uniformly: $S_p = \{P_1, P_{r(i'/N_c)}, P_{2r(i'/N_c)}, ..., P_{N_c r(i'/N_c)}\}$, where $r(\star)$ means the maximal integer smaller than $\star$. For every potential old loop-closure pose in S_p, seeing it as the target and the last pose $P_{i'}$ as start point, we can generate N_{RRT} candidate paths based on the RRT-connect method [89]. So we have $N_{RRT}(N_c + 1)$ candidate paths. There are two advantages using the RRT-connect method:

- Randomness: As a random sampling path planning method, the RRT-connect method provides us the possibility to find the path with good loop-closure measurements instead of only the shortest paths.

- Real-time ability: The RRT-connect method is an on-line path planning method for small/middle-scale space, which can quickly generate many candidate paths.

Then, for every candidate path, because the noise level is in general proportional to the number of points which are visible in both two poses, we can count its weighted node degree by the weighted number of the common features.

Finally, the best paths are chosen by sorting all the virtual weighted node degrees of the candidate paths. They make up the best elite group with N_e paths.

6.4.3 Tree-connectivity for elite search and key poses selection

For this small elite group, every path is evaluated by the weighted tree-connectivity metric. In order to obtain weighted tree-connectivity, the weighted Laplacian matrix, including the old poses and the future poses in the candidate paths, needs to be constructed incrementally:

$$\boldsymbol{L}_{w,new}^{SO(n)} = \begin{bmatrix} \boldsymbol{L}_{w}^{SO(n)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \boldsymbol{A}_{add}^{\top} \boldsymbol{\Sigma}_{add} \boldsymbol{A}_{add}, \tag{6.7}$$

where $\boldsymbol{A}_{add}$ are the new incidence matrix corresponding to the future measurements. $\boldsymbol{\Sigma}_{add}$ is a diagonal matrix whole diagonal elements are the weighted value of the new graph edge, and they satisfy the following structure:

$$\boldsymbol{A}_{add} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{A}_{add}^{C} & \boldsymbol{A}_{add}^{D} \end{bmatrix}, \ \boldsymbol{\Sigma}_{add} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{add}^{D} \end{bmatrix}. \tag{6.8}$$

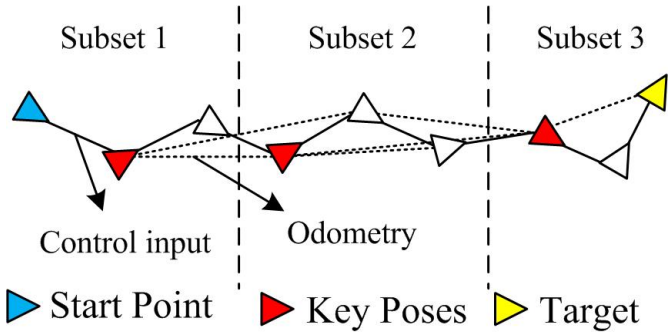


Figure 6.2 : An example showing key poses in RRT-connect result. The key pose has the largest weighted node degree in its corresponding subset.

It is easy to find that $\boldsymbol{A}^C_{add}$, $\boldsymbol{A}^D_{add}$ and $\boldsymbol{\Sigma}^D_{add}$ are all sparse matrices, which can be updated quickly.

Then, the best path is obtained by choosing the one with the largest weighted tree-connectivity. In this best path, all the poses, which are the nodes in RRT-connect, are divided into N_k sub-sets uniquely. In the k-th sub-set, the poses are sorted by their weighted node degrees and the best pose is defined as the k-th key pose $\widehat{P}_k$. The benefit of the application of the sub-set is to avoid the situation where key poses are located at adjoining positions, which reduces the effectiveness of the key poses in a smooth path. An example with 9 poses and 3 subsets is shown Fig. 6.2.

6.4.4 Fast trajectory planning

Based on the above method, N_k key poses are found to guarantee the performance of the selected result. In order to reduce the length of the random sampling path and assign the smooth continuous velocity along the path, we apply a continuous-time trajectory planning method [109] to pass the key poses and connect the start point and the target.

In every dimension (x, y and z), the robot trajectory is represented as a high-

degree polynomial spline with N-th order, whose variable is time t:

$$f_\mu(t) = \boldsymbol{A}^\top \boldsymbol{T}(t), \mu = x, y, z, \tag{6.9}$$

where $\boldsymbol{A} = [a_0, a_1, ..., a_N]^\top$ and $\boldsymbol{T}(t) = \left[t^0, t^1, ..., t^N\right]^\top$.

The coordinates of the key poses and the end constraints are set as the fixed derivatives $\boldsymbol{d_F}$ [109]. The remaining free derivatives $\boldsymbol{d_P}$, which are the optimized variables, are the free waypoints in the spline. The mapping between the polynomial coefficients $\boldsymbol{A}$ and the derivatives meets:

$$\boldsymbol{A} = \widehat{\boldsymbol{T}}^{-1} \boldsymbol{M} \left[\boldsymbol{d_F}^\top \ \boldsymbol{d_P}^\top\right]^\top, \tag{6.10}$$

where $\widehat{\boldsymbol{T}} = [\boldsymbol{T}(t_0) \ \boldsymbol{T}(t_1) \ \cdots \ \boldsymbol{T}(t_N)]^\top$ is a mapping matrix from polynomial coefficients to the end-derivatives, $t_0, \cdots, t_N$ are the reach time corresponding to the derivatives, $\boldsymbol{M}$ is the re-ordering matrix.

The objective function of the trajectory planning problem is to avoid obstacles and to minimize the snap:

$$\begin{gathered} J = \omega_1 J_1 + \omega_2 J_2 \\ J_1 = \int_{t_0}^{t_N} \sum_{\mu=x,y,z} \frac{d^k f_\mu(t)}{dt^k} dt, J_2 = \int_{t_0}^{t_N} c(\boldsymbol{f}) \left\| \frac{d\boldsymbol{f}}{dt} \right\| dt, \end{gathered} \tag{6.11}$$

where $\boldsymbol{f} = [f_x(t), f_y(t), f_z(t)]$, $c(\boldsymbol{f})$ is the potential cost to avoid an obstacle. Its computation is introduced in Section 6.4.6. In our simulation, the coefficient k is set as 4. Using (12), the objective function in (13) can be derived as:

$$J_1 = [\boldsymbol{d_F} \ \boldsymbol{d_P}] \boldsymbol{M}^\top \widehat{\boldsymbol{T}}^{-\top} \boldsymbol{Q}_J \widehat{\boldsymbol{T}}^{-1} \boldsymbol{M} \left[\boldsymbol{d_F}^\top \ \boldsymbol{d_P}^\top\right]^\top \quad , \tag{6.12}$$

where $\boldsymbol{Q}_J$ is the Hessian matrix of J_1.

Then this trajectory optimization problem is solved based on the quasi-Newton method with known gradient. The specific gradient equation is shown in [109].

6.4.5 Special amendment for directional sensor

The above method is suitable only for omni-directional sensors, because the trajectory passes through key poses without a direction constraint. In order to apply this framework with a directional sensor, like a camera, a velocity constraint is introduced in the trajectory planning problem.

For every key pose $\widehat{P}_k$, we need to set its speed $\boldsymbol{V}(\widehat{P}_k)$ based on the RRT-connect result and set its reaching time $T^*(t_k)$. Except the fixed derivative corresponding to the key pose, an additional fixed derivative $\widehat{P}_k^{add}$ needs to be introduced into the continuous-time trajectory planning problem and its corresponding time is $T^*(t_k) + \Delta t_1$:

$$\widehat{P}_k^{add} = \widehat{P}_k + \boldsymbol{V}(\widehat{P}_k)\Delta t_1, \tag{6.13}$$

where Δt_1 is very small to ensure the velocity direction. This simple operation can be used in any pose with the velocity direction constraint. A example of the continuous-time trajectory planning problem with the velocity constraint is shown in Fig. 6.3.

6.4.6 Map representation

Focusing on the pose-graph SLAM, we assume that the environment has been constructed on-line by some tools such as Octomap [109] or SDF tools [49]. In order to avoid obstacles in the following planning process, for a waypoint $\boldsymbol{s}$ in the planning space, we can first identify whether it is in collision with obstacles, and then get the closest point and the distance $d(\boldsymbol{s})$ between this waypoint and the obstacles. In this way, a potential field $c(\boldsymbol{s})$ using the Euclidean Singed Distance Field is built as described in [109]. In this process, in order to avoid the frequent computations

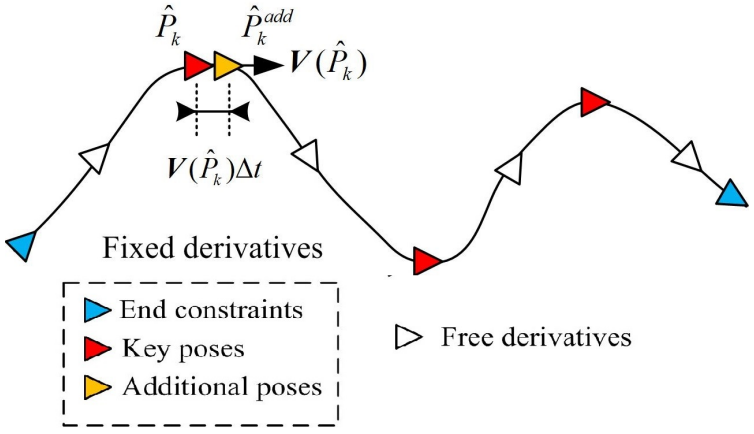


Figure 6.3 : An example of the continuous-time trajectory planning result with velocity constraint based on the fixed derivatives and free derivatives.

of $c(\boldsymbol{s})$ and its gradient in the continuous-time trajectory optimization, they are computed in the mapping process by GPU or individual processor cores quickly before the continuous-time trajectory planning process and then be called using the hash table by the help of the grid representation of the planning space.

6.4.7 Whole framework summary

In this chapter, we summarize all the steps used in our framework and show how to apply it in performing the original non-active SLAM task (Algorithm 9).

Algorithm 9 On-line active pose-graph SLAM method

Require: Robot parameters, original task
Ensure: Best active SLAM trajectory
while Original task hasn't been finished **do**
  if Pose Uncertainty is accepted ($\widehat{\boldsymbol{C}}_{i'i'} < Index_1$) **then**
    Keep performing the original task;
  else
    if Robot resource is enough **then**
      Planning 1:
      Step 1: Output all potential loop-closure poses with equal interval: S_p;
      Step 2: Generate multiple RRT-connect paths based on S_p and last pose $P_{i'}$, evaluate their weighted node degrees and choose elite group;
      Step 3: Evaluate weighted tree-connectivity, choose best path and select key poses;
      Step 4: Build continuous-time trajectory planning problem based on fixed key poses.
      end Planning 1
      repeat
        Control robot to follow continuous-time result;
        if Reach target of Planning 1 result **then**
          Planning 2:
          Generate multiple RRT-connect paths based on current pose and $P_{i'}$, then run rest operations in Step 2-4 again;
          end Planning 2
        else
          Keep performing original task;
      until Believable covered area is large enough

6.4.8 Computational complexity

In this part, we discuss the computational complexity of the proposed method. There are two cases when applying our method: the covariance recovery (Section 6.4.1) and whole active SLAM method (Section 6.4.1 to 6.4.5).

When the map is acceptable or the robot resource is not enough, only Section 6.4.1 is performed. The computational complexity of the fast covariance recovery for last pose is generally much smaller than $O(((n+d)n_p)^3),\ d = n(n-1)/2$, because of the sparsity of the FIM. For a pose-graph SLAM with 200 poses, this method can be run on-line very easily (up to 50Hz) using common commodity computer.

For the second case, which is only triggered several times in the whole task, the most time-consuming parts will be the performance evaluation of the candidate paths and the continuous-time trajectory planning. The computational complexity of performance evaluation of the candidate paths is $O((N_c+1)n_p)+O(N_e n_p^3)$. Based on suitable parameters N_c and N_e, the computational time is limited and will be presented in Section 6.6.1. The computational complexity of the trajectory planning part depends on the obstacles in the environment, the number of the free waypoints and the grid density of the planning space. In general, in a submap, this part can be run in several Hertz.

It is obviously safe if the planning time is smaller than the pose interval in pose-graph network (Δt). Even when the platform has low computational ability and cannot finish the planning within Δt, we can safely use this method on the old graph corresponding to the previous situation several seconds ago. This strategy is feasible since the pose-graph will not change much in only few seconds.

6.5 Sub-map planning and estimating

Even though the proposed method only needs to evaluate the weighted tree-connectivity of the small elite group, it is still expensive for a large-scale problem. So this chapter uses the sub-map estimation and planning idea to improve accuracy of the local map and its real-time ability [146].

When the size of the problem is larger than a threshold ($n_p > Index_2$), a new sub-map is built. All the pose-graph structure before this pose is saved and a new pose-graph SLAM problem is built. The target of the global task is transformed into local coordinates based on the coordinate frames of the sub-maps. The active SLAM method is applied based on the new pose graph without considering other submaps. It is noted that we divide the submaps by the measurements. There are some common poses in different submaps, which serve as loop-closures.

6.6 Simulations and experiments

6.6.1 Simulation

For the simulation, the whole system including the dynamic model, control method, constructed map and active SLAM method, is implemented in MATLAB on a laptop PC with Intel Core i7-7700HQ @ 3.5 GHz and 8GB RAM.

In order to get the measurements following the noisy assumption shown in (B.51), based on the singular value decomposition, the noise-less relative rotation and translation are obtained from the 3D points observations, and then, the random noises are sampled by $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{ij})$ and $Lang(\boldsymbol{I}_{n\times n}, \kappa_{ij})$. The indexes κ_{ij} and δ_{ij} are set to be proportional to the feature number N_v which are visible both in P_i and P_j [106].

Table 6.1 : Comparison of trajectory length and computational time

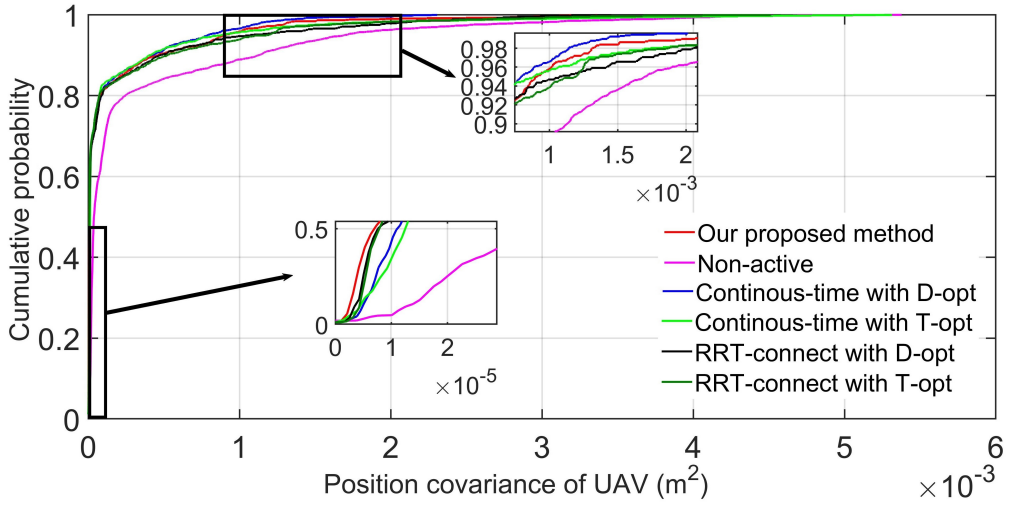
		Average planning time		Trajectory length	
Algorithm		Mean (s)	STD (s^2)	Mean (m)	STD (m^2)
Task 1	Non	-	-	25.44	-
	Ours	2.52	0.37	41.63	1.46
	Con-D	7.65	0.38	41.02	1.72
	Con-T	6.29	0.63	43.92	3.71
	RRT-D	8.45	0.53	41.16	0.78
	RRT-T	6.90	0.55	42.88	0.87
Task 2	Non	-	-	15.20	-
	Ours	1.45	0.56	28.35	4.21
	Con-D	6.00	0.79	27.09	3.29
	Con-T	5.18	0.89	27.17	1.94
	RRT-D	6.45	1.16	30.41	0.44
	RRT-T	5.81	0.53	29.69	0.70

A small-scale simulation

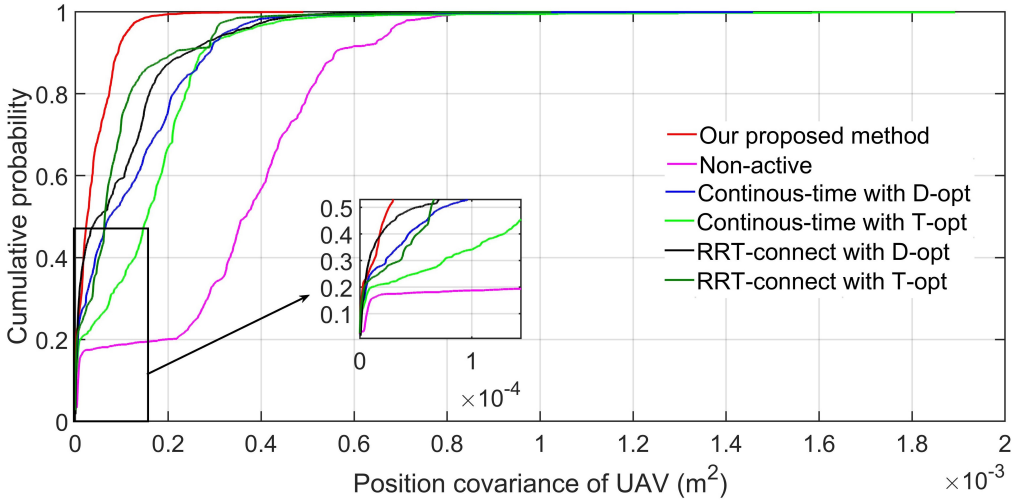
In this part, we present a simulation environment in a 7m $\times$ 7m $\times$ 1m space with several regular obstacles as shown in Fig. 6.5. The original task of the robot is to

pass several way-points (Blue pentagrams) with a velocity 0.1m/s. The simulation time step Δt is set as 1s. The noise parameter δ_{ij}^{-2} and κ_{ij} for the control input are 10^5 and 10^4 in every step. The noise parameter δ_{ij}^{-2} and κ_{ij} for the odometry are $20N_v$ and $10^4 N_v$. $Index_1$, N_c and N_{RRT} are respectively set as 1×10^{-3}, 10 and 5. The submap idea is not used in this small-scale simulation. In other words, one simulation is one sub-map. The maximal one-way loop-closure trajectory is limited below 4m. In order to avoid frequent loop-closure, in one sub-map, the active SLAM method is only allowed to be triggered twice ($N_s = 2$). The limitations of the triggered times and the loop-closure trajectory length are just two examples of the limitations of the robot resources. It is possible to use other ways to automatically limit them.

We compare our method with some other methods. The simulation of following the way points without active loop closure is named as non-active method (Non). The simulations, which use continuous-time trajectory planning method to directly re-visit the potential loop-closure poses without using RRT-connect method, based on the T- and D-optimality metrics are named as continuous-time with D/T-opt (Con-D/T). The simulations, using RRT-connect method, are named as RRT-connect with D/T-opt (RRT-D/T). Because of the randomness, for every task, based on different methods, we run the simulations 10 times, and then apply the cumulative distribution function (CDF) to evaluate the covariance of the robot position at every step [25]. The uncertainty comparison results are shown in Fig. 6.4. We also show the statistical results of the trajectory length and the average simulation time of Planning 1 (usually twice) and Planning 2 (usually twice) of one simulation in Table 6.1 (The data association of the common features for the predicted poses is included in the planning time). One of the trajectories and final estimated results for tasks 1 (398 Poses) and 2 (208 Poses) using our method is shown in Fig. 6.5.



(a) CDF of the covariance in the robot position at every time step of task 1



(b) CDF of the covariance in the robot position at every time step of task 2

Figure 6.4 : CDF for task 1 and 2 based on 10 simulations (The high position covariance means worse performance. We would like a CDF that gets closed to y-axis and reaches 1 as quickly as possible. The red line (Our method) shows good performance (gets closed to y-axis with high ratio) in all lines)

Fig. 6.4 and Table 6.1 show that our method has the best performance in uncertainty reduction and planning speed. In other 10 simulations with 10 different paths, we find that the mean position covariances improve 37% on average by the additional 71% path compared with non-active method. In Fig. 6.5, we see that the active SLAM is triggered twice because of the weak-connectivity of the measurement network. Relative measurements are shown by the blue lines, and the red circles are

the estimated robot poses at last step.

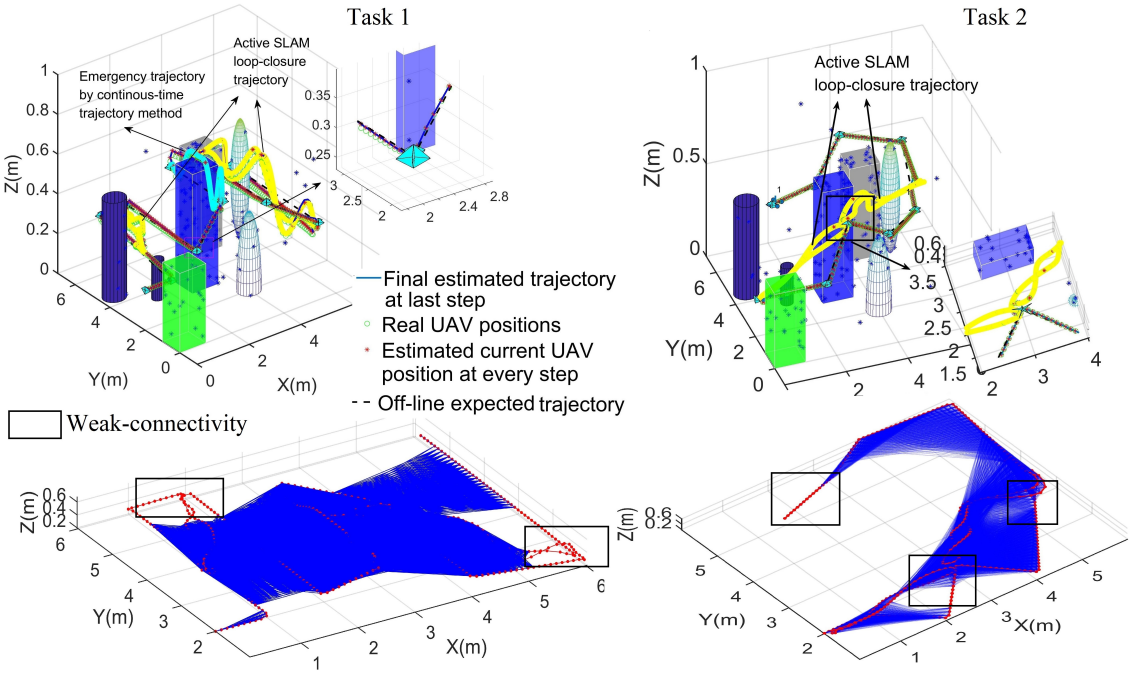


Figure 6.5 : Active SLAM trajectory, estimated pose graph results and relative measurements for task 1 and 2 in a small environment simulation

Large-scale simulation with multiple sub-maps

Even though this space is small, because of the low velocity and the large number of features, there are more than 10^4 relative pose measurements in tasks 1 and 2. It is hard to solve the active SLAM problem in real-time. In order to reduce the computational complexity, the sub-map idea is used. $Index_2$ is set as 150. Simulation results with 9 sub-maps * and more than 10^5 relative measurements are shown in Fig. 6.7.

With the help of sub-maps, the planning and SLAM algorithms are limited to 2s and 1.5s, respectively, in every sub-map for the above simulations.

*It is noted that, in the previous section, no submap joining method, like Linear SLAM [146], is used. It is only used in the large-scale simulation.

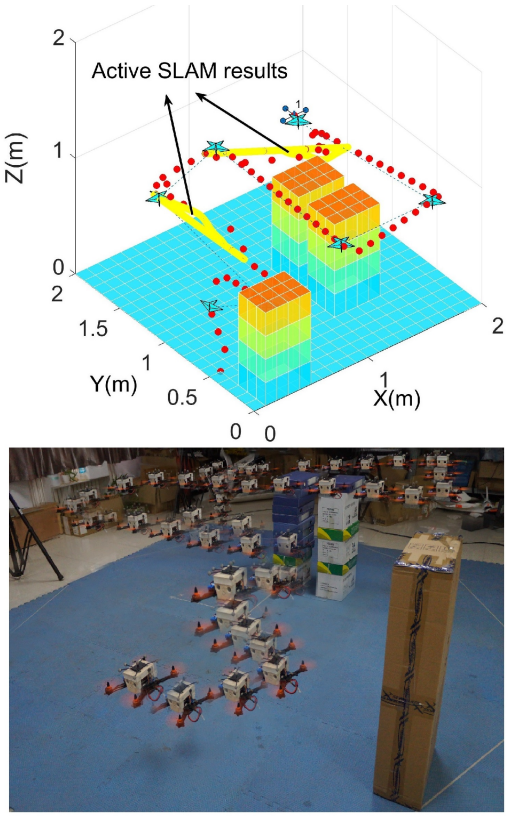


Figure 6.6 : On-line active SLAM (Composite trajectory)

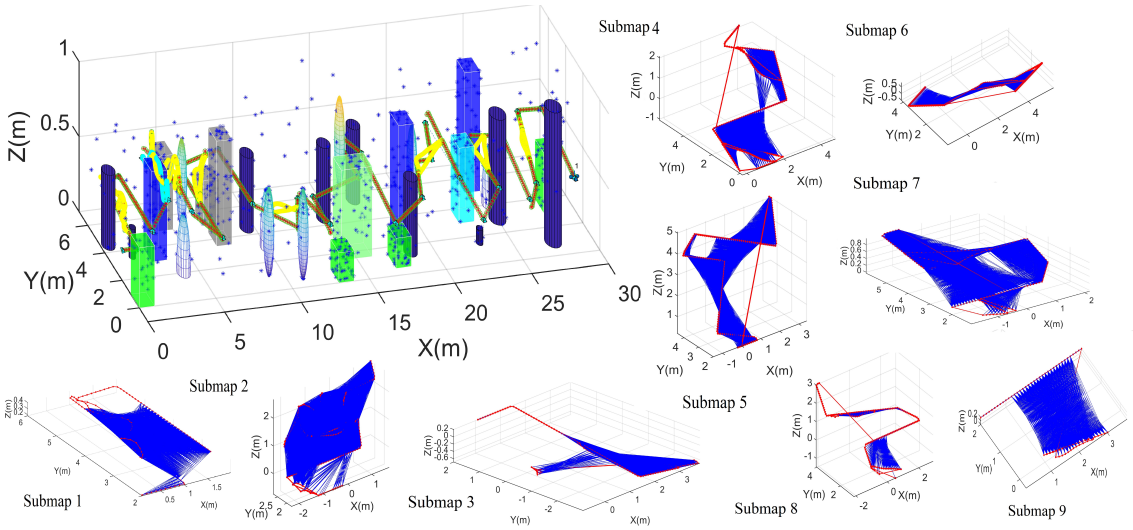


Figure 6.7 : Active SLAM, real and estimated trajectory results in 9 submaps

Comparisons

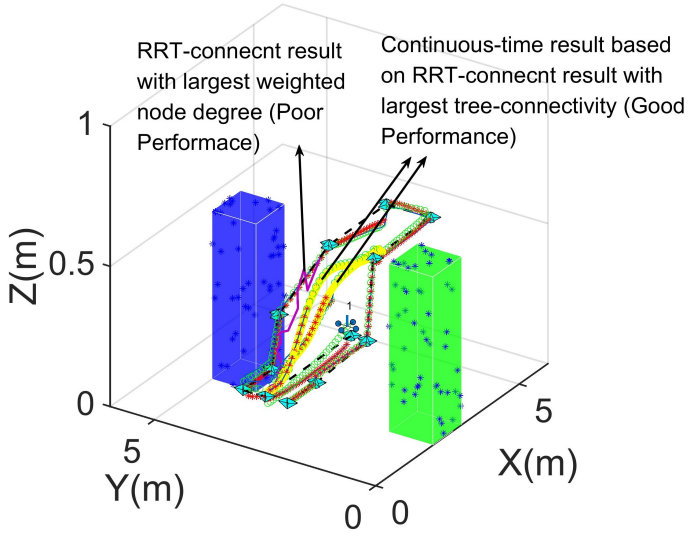


Figure 6.8 : The results based largest weighted tree-connectivity and previous weighted node degree evaluation is better than the ones only based largest weighted node degree in one simulation

Comparison between graphic metrics and optimality design metrics We compare the presented metric, combining the weighted node degree and graph connectivity, with the T- and D-optimality metrics in Fig. 6.9. The following simulation is to perform pose-graph SLAM using a single map. In every step, 50 randomly generated candidate trajectories are evaluated by different metrics, and then record the computation time of these metrics. It is noted that the incremental update technology in Eq.(6.7) and Eq.(6.8) is not used in this simulation. The FIM and weighted Laplacian matrices are generated for every candidate trajectory and their generation time is also included in the computational time shown in Fig. 6.9. Meanwhile, the weighted tree-connectivity and the weighted node degree get close to the D- and T-optimality metrics of the FIM. Define the ratio between these two new metrics with the original ones, where $RE_2 = \frac{\text{trace}(\boldsymbol{\mathcal{I}}_{nD})}{\sum_{i=1}^{n_p}\sum_{j\in V_i}\left(\omega_{ij}+n\delta_{ij}^{-2}\right)}$ and $RE_3 = \frac{\log(\det(\boldsymbol{\mathcal{I}}_{nD}))}{\mathcal{L}}$. Their values are limited in $RE_2 \in (100\%, 108\%)$ and $RE_3 \in (100\%, 101\%)$, respectively.

Comparison between two new metrics In this chapter, we compare two new graphic metrics to indicate that the weighted tree-connectivity have a better perfor-

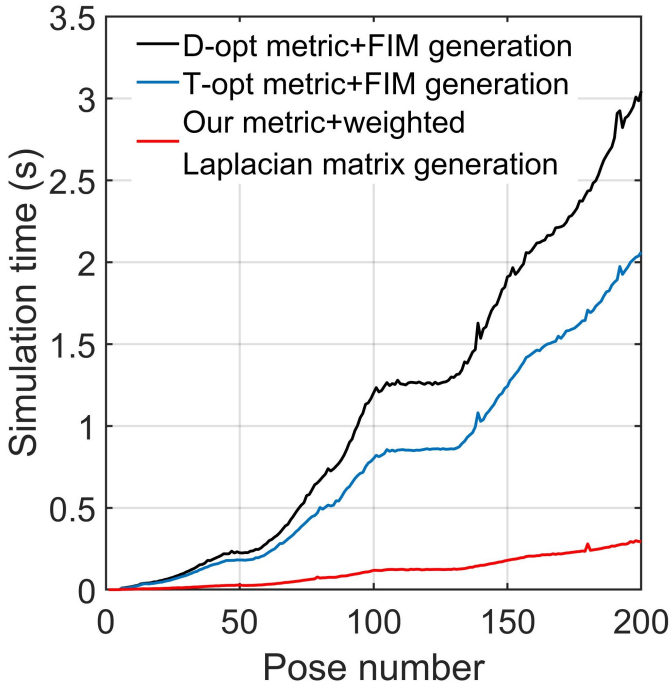


Figure 6.9 : Computational time of three metrics and matrix generations

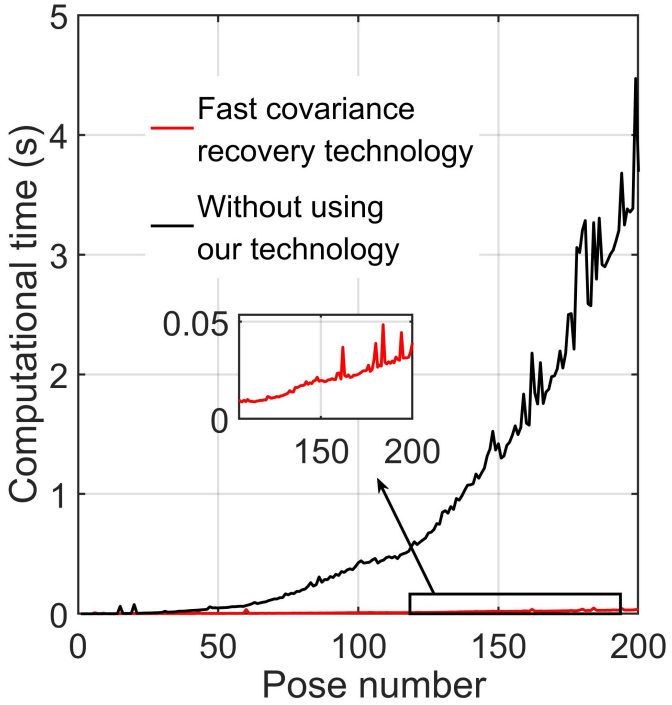


Figure 6.10 : Computational time of using/without using fast covariance recovery

mance than the easily computed metric, weighted node degree.

In Fig. 6.8, the purple path is the best one in the initial RRT-connect results only using the weighted node degree. The small elite group with large weighted node degrees is further evaluated by the weighted tree-connectivity. The final best continuous-time result with the largest weighted tree-connectivity is shown in the yellow trajectory. The yellow trajectory has a 7.3% advantage compared with the purple one in the mean covariance of the poses.

Comparison between using/without using covariance recovery In this chapter, we compare the computational time of our fast covariance recovery with the ones of the fully inverse operation of the FIM.

Because of the high efficiency of the fast covariance recovery, compared with the inverse of the whole dense covariance matrix, Fig. 6.10 shows the outstanding computational performance of our method.

6.6.2 On-line experiments

For the on-line experiments, based on a real quad-rotor UAV platform, the planning method shown in Algorithm 9 is implemented on-line in C++ on a desktop PC with Intel(R) Core(TM) i7-4790K CPU @ 4.0GHz with 32 GB of DDR3 1600MHz RAM. The map is constructed off-line. The virtual relative measurements are obtained by the global localization system using the relative poses and adding expected noise. Pose-graph SLAM is performed on-line but not at every step. The final trajectory is shown in Fig. 6.6. The planning parts of the active SLAM are triggered twice, shown by the yellow lines. The experiments show that the planning part can be finished on-line in 0.2-0.4s (2-5Hz) with about 80 poses.

6.7 Summary

This chapter presents an on-line active pose-graph SLAM method for a robot operating in a 3D unknown environment. This method performs active SLAM by adding loop-closure trajectories to reduce pose uncertainty. It uses the RRT-connect sampling-based path planning method and nonlinear continuous-time trajectory planning method based on key poses. In this process, the cheap metrics, derived in chapter 3, combining the weighted node degree and weighted tree-connectivity are used to choose the key poses and its corresponding best loop-closure trajectory. Simulations and experiments show that our method has good performance and can be run in real-time.

Chapter 7

Conclusions and Future Work

In this thesis, based on the previous references [77, 78], we further explore a overlooked aspect of SLAM, namely the impact of graph topology on the estimation error covariance of the SLAM problem, and apply it in multiple derivate planning problems, including anchor selection and active SLAM problems. In this chapter, we briefly review our achievements and discuss future work.

7.1 Cramér-Rao bounds and optimal design metrics for pose-graph SLAM

2D/3D Pose-graph SLAM is a problem of estimating a set of poses based on noisy measurements of relative rotations and translations. Chapter 3 focuses on the relation between the graphical structure of pose-graph SLAM and FIM, CRLB and its optimal design metrics (T-optimality and D-optimality). As a main contribution, based on the assumption of isotropic Langevin noise for rotation and block-isotropic Gaussian noise for translation, the FIM and CRLB are derived and shown to be closely related to the graph structure, in particular, the weighted Laplacian matrix. We also prove that total node degree and weighted number of spanning trees, as two graph connectivity metrics, are respectively closely related to the trace and determinant of FIM. The discussions show that, compared with the D-optimality, the T-optimality metric is easy-computed but with a general effectiveness. We also present upper and lower bounds for the D-optimality metric, which can be efficiently computed and are almost independent of the estimation results. The

proposed conclusions are verified with several well-known datasets, such as Intel, KITTI, sphere and so on.

Future Work

Theorem 7 can be straightforwardly generalized to 2D/3D feature-based SLAM problems. We formulate the SLAM problem as the synchronization on a direct product group $\mathbb{R}^n \times SO(n)$. Some references, like [10], suggest that our analysis can also be formulated by a group $SE(n)$. We will consider this extension in our future work.

7.2 Anchor selection problem

Chapter 4 considers SLAM problem for robots in situations where accurate estimates for some of the robot poses, termed anchors, are available. These may be acquired through external means, for example, by either stopping the robot at some previously defined known locations or pausing for a sufficient period of time to measure the robot pose with an external measurement system. The main contribution is an efficient algorithm for selecting a set of anchors, from a set of potential poses, that minimizes estimated error in the SLAM solution. It is shown that, based on graph topology, D-optimality design metric of FIM of the pose-graph SLAM problem is closely connected with weighted tree-connectivity of graph structure of measurement network. Based on weighted tree-connectivity, it is shown that the anchor selection problem can be formulated approximately as a sub-matrix selection problem for reduced weighted Laplacian matrix of the pose graph, leading to a maximization problem of a sub-modular function with a cardinality-fixed constraint. Two greedy methods are presented to solve this sub-modular optimization problem with a performance guarantee. Sparseness of the weighted Laplacian matrix is exploited using Cholesky decomposition, AMDP, order re-use and rank-1 update to reduce the cost

of computing the solution. Simulations with public-domain datasets are presented to demonstrate the efficiency and effectiveness of the proposed techniques.

Future Work

The greedy-based anchor selection problem is just an initial available algorithm with a low performance-guarantee. A faster and better-performance method needs to be explored. One of the possible direction is the convex-relaxation algorithm.

At the same time, the applications of the anchor selection problem need to be better designed and could be extended to active perception area and deformable SLAM area.

7.3 Active SLAM

The active SLAM problem considered in Chapter 5 aims to plan a collision-free mobile robot trajectory for SLAM as well as for an area coverage task in 2D case. Based on a MPC framework, the SLAM uncertainty minimization problem and the area coverage problem considering robot pose uncertainty are solved respectively by different methods. For the uncertainty minimization problem, based on the graphical structure of the 2D feature-based SLAM, a non-convex constrained least-squares problem is presented to approximate the original problem. A convex half-space representation is applied to relax the non-convex no-fly zone avoidance constraints. Using convex relaxation, it is further transformed into a convex problem, and then solved by a convex optimization method and a rounding procedure based on the SVD. For the area coverage problem, it is solved by the SQP method. In the whole process, in order to deal with the local minimum problem, control switching problem and computational complexity, we use Linear SLAM, which is a sub-map joining approach. Meanwhile, a theorem about the bound of the eigenvalues of the FIM of the joined global map is presented and proved. Finally, various simulations

are presented to validate the effectiveness of the proposed approach.

In Chapter 6, we present an on-line active pose-graph SLAM framework for robots in 3D environments using graph topology and sub-maps. This framework aims to find the best trajectory for loop-closure by re-visiting old poses based on the T-optimality and D-optimality metrics of the FIM in pose-graph SLAM. In order to reduce computational complexity, graph topologies are introduced, including weighted node degree (T-optimality metric) and weighted tree-connectivity (D-optimality metric), to choose a candidate trajectory and several key poses. With the help of the key poses, a sampling-based path planning method and a continuous-time trajectory optimization method are combined hierarchically and applied in the whole framework. So as to further improve the real-time capability of the method, the sub-map joining method is used in the estimation and planning process for large-scale active SLAM problems. In simulations and experiments, we validate our approach by comparing against existing methods, and we demonstrate the on-line planning part using a quad-rotor UAV.

Future Work

The applications presented in Chapters 5 and 6 are only the first steps towards the use of graph topology in active SLAM areas. We believe that designing reliable graph topologies has the potential to be applied into the multiple robot active SLAM area. We aim to thoroughly investigate this idea in our future work.

Appendices

Appendix A

Preliminaries

A.1 Graph theory for SLAM

A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which is weakly connected, is used to represent pose-graph SLAM problem naturally. In this paper, $\mathcal{V} = \{1, 2, \cdots, n_p\} \bigcup \{0\}$, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and $|\mathcal{E}| = m$. Each node denotes a robot pose, and each edge $(i, j) \in \mathcal{E}$ represents a relative measurement between two robot poses. An undirected graph $\mathcal{G}_1 = (\mathcal{V}, \mathcal{F})$, corresponding to $\mathcal{G}$, has the same nodes and the same but undirected edges with the directed graph $\mathcal{G}$.

We denote the i-th node degree by d_i. For the i-th node, we can define three node sets V_i^+, V_i^- and V_i satisfying $(i, j) \in \mathcal{E} \Leftrightarrow j \in V_i^+$, $(j, i) \in \mathcal{E} \Leftrightarrow j \in V_i^-$ and $V_i = V_i^- \bigcup V_i^+$, so we have $|V_i^+| + |V_i^-| = |V_i| = d_i$. Without loss of generality, the first pose is assumed to be the origin of our global coordinates system.

The incidence matrix of $\mathcal{G}$ is denoted by $\boldsymbol{A}_0 \in \{-1, 0, 1\}^{(n_p+1)\times m}$. $a_{ik} = -1$ and $a_{jk} = 1$ (the (i, k)-th and (j, k)-th element of $\boldsymbol{A}_0$) are non-zero, if there is an edge $e = (i, j) \in \mathcal{E}$. The incidence matrix after anchoring $\boldsymbol{x}_0$ to the origin, $\boldsymbol{A} \in \{-1, 0, 1\}^{n_p \times m}$, is obtained simply by removing the row corresponding to the first node in $\boldsymbol{A}_0$. The Laplacian matrix and the reduced Laplacian matrix of $\mathcal{G}$ are respectively defined as $\boldsymbol{L}_0 \triangleq \boldsymbol{A}_0 \boldsymbol{A}_0^\top$ and $\boldsymbol{L} \triangleq \boldsymbol{A}\boldsymbol{A}^\top$. It can be shown that $\boldsymbol{L}_0$ and $\boldsymbol{L}$ are respectively positive semi-definite and positive definite, if $\mathcal{G}$ is (weakly) connected. The Laplacian matrix and the reduced Laplacian matrix of $\mathcal{G}$ can be written as $\boldsymbol{L}_0 = \boldsymbol{D}_0 - \boldsymbol{W}_0$ and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}_0 \triangleq \mathrm{diag}(d_0, d_1, \ldots, d_n)$, $\boldsymbol{D} \triangleq \mathrm{diag}(d_1, \ldots, d_n)$, $\boldsymbol{W}_0$ and $\boldsymbol{W}$ are respectively the

original adjacency matrix and the adjacency matrix of the graph after removing the row and column corresponding to $\boldsymbol{x}_0$. The weighted Laplacian matrix and the weighted reduced Laplacian matrix are defined as $\boldsymbol{L}_\omega^0 \triangleq \boldsymbol{A}_0\boldsymbol{\Sigma}_0\boldsymbol{A}_0^\top$ and $\boldsymbol{L}_\omega \triangleq \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top$, where $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_0$ are diagonal matrices whose diagonal elements are the weighted value of the graph edge.

A.2 Preliminaries for Chapter 4

In this part, we give some preliminary knowledge about the sub-modularity. These concepts are defined in set function, i.e. functions $f : 2^V \to \mathbb{R}$ that assign each subset $S \subseteq V$ a value $f(S)$, where V denote a finite set, named the ground set.

Definition 2. *(Normalized [69]) The function f is normalized if $f(\emptyset) = 0$.*

Definition 3. *(Monotone [69]) The function f is monotone decreasing if for all $A \subseteq B \subseteq V$, $f(A) \geq f(B)$.*

Definition 4. *(Sub-modularity [69]) The function f is sub-modular if $f(A)+f(B) \geq f(A \cup B) + f(A \cap B),\ \forall A,\ B \subseteq V$. It has an equivalent definition: The function f is sub-modular if $f(A \cup \{u\}) - f(A) \geq f(B \cup \{u\}) - f(B),\ \forall A \subseteq B \subseteq V, u \in V \setminus B$.*

It is easy to find that the sub-modularity exhibits a natural diminishing returns property.

Definition 5. *(Cardinality-fixed submodular maximization [51]) For a submodular function $f : 2^V \to \mathbb{R}$, find: $max_{S \subseteq V, |S|=k} f(S)$.*

The cardinality-fixed constraint comes from restricting S to the bases of the matroid.

Lemma 4. *(Cauchy Interlacing Theorem, [52], p. 411) Let $\boldsymbol{\mathcal{A}}$ be an $n \times n$ symmetric matrix and the set of eigenvalues of $\boldsymbol{\mathcal{A}}$ be denoted as $\lambda_1(\boldsymbol{\mathcal{A}}), \lambda_2(\boldsymbol{\mathcal{A}}), \cdots, \lambda_n(\boldsymbol{\mathcal{A}})$, where it is assumed that $\lambda_1(\boldsymbol{\mathcal{A}}) \leq \lambda_2(\boldsymbol{\mathcal{A}}) \leq \cdots \leq \lambda_n(A)$. Let $\boldsymbol{\mathcal{A}}_i = \boldsymbol{\mathcal{A}}(\{1, ..., n\} \backslash \{i\})$*

is the sub-matrix of $\boldsymbol{\mathcal{A}}$ *by removing the* i*-th column and row for some* $i \in \{1, \cdots, n\}$*, then:*

$$\lambda_1(\boldsymbol{\mathcal{A}}) \leq \lambda_1(\boldsymbol{\mathcal{A}}_i) \leq \lambda_2(\boldsymbol{\mathcal{A}}) \leq \cdots \leq \lambda_{n-1}(\boldsymbol{\mathcal{A}}) \leq \lambda_{n-1}(\boldsymbol{\mathcal{A}}_i) \leq \lambda_n(\boldsymbol{\mathcal{A}}). \tag{A.1}$$

A.3 Preliminaries for Chapter 5

In this part, we give some preliminary knowledge about the variational description of eigenvalues and the Fischers min-max theorem.

Theorem 9. *All eignvalues of a block diagonal matrix are the eignvalues of all block matrixes on the diagonal line.*

Theorem 10 (Variational description of eignvalues)**.** *([99], p. 232) Let* $\boldsymbol{\mathcal{A}}$ *be a real symmetric* $n \times n$ *matrix with eigenvalues* $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$*. Let* $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_n)$ *be an orthogonal* $n \times n$ *matrix which diagonalizes* $\boldsymbol{\mathcal{A}}$*, so that:*

$$\boldsymbol{S}^T \boldsymbol{\mathcal{A}} \boldsymbol{S} = diag(\lambda_1, \lambda_2, \cdots, \lambda_n) \tag{A.2}$$

Then, for $k = 1, 2, \cdots, n$*, we have:*

$$\lambda_k = \min_{\boldsymbol{\mathcal{R}}_{k-1}^T \boldsymbol{x} = 0} \frac{\boldsymbol{x}^\top \boldsymbol{\mathcal{A}} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \max_{\boldsymbol{\mathcal{T}}_{k+1}^\top \boldsymbol{x} = 0} \frac{\boldsymbol{x}^T \boldsymbol{\mathcal{A}} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \tag{A.3}$$

where,

$$\begin{aligned} \boldsymbol{\mathcal{R}}_k &= (\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_k), \\ \boldsymbol{\mathcal{T}}_k &= (\boldsymbol{s}_k, \boldsymbol{s}_{k+1}, \cdots, \boldsymbol{s}_n) \end{aligned} \tag{A.4}$$

Theorem 11 (The Fischer's min-max theorem)**.** *([99], p. 233) Let* $\boldsymbol{\mathcal{A}}$ *be a real symmetric* $n \times n$ *matrix with eigenvalues* $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$*. Let* $1 \leq k \leq n$*. Then,*

for every $n \times (k-1)$ *matrix* $\boldsymbol{\mathcal{B}}$,

$$\min_{\boldsymbol{\mathcal{B}}^T \boldsymbol{x}=0} \frac{\boldsymbol{x}^T \boldsymbol{\mathcal{A}} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \leq \lambda_k \tag{A.5}$$

for every $n \times (n-k)$ *matrix* $\boldsymbol{\mathcal{C}}$,

$$\max_{\boldsymbol{\mathcal{C}}^T \boldsymbol{x}=0} \frac{\boldsymbol{x}^T \boldsymbol{\mathcal{A}} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \geq \lambda_k \tag{A.6}$$

Appendix B

Proofs

B.1 Proofs in Chapter 3

B.1.1 Proof of Theorem 1

In this appendix, we will present the detailed derivations of the FIM for 2D pose-graph SLAM (prove Theorem 1)*. The following lemma will be used in the derivation of the results.

Lemma 5. *[119] X and Y are random variables, when they are independent, we have* $\mathbb{E}\{XY\} = \mathbb{E}\{X\}\mathbb{E}\{Y\}$.

In the log-likelihood function (3.5), based on measurement function (3.1) and (3.3), $\boldsymbol{y}_{ij}$ and $\boldsymbol{Z}_{ij}$ can be seen as the function of the estimand $\boldsymbol{x} \times \boldsymbol{R}$, satisfying: $\boldsymbol{y}_{ij} = \boldsymbol{y}_{ij}(\boldsymbol{R}_i,\ \boldsymbol{x}_i,\ \boldsymbol{x}_j) = \boldsymbol{p}_{ij} - \boldsymbol{R}_i^\top(\boldsymbol{x}_j - \boldsymbol{x}_i)$ and $\boldsymbol{Z}_{ij} = \boldsymbol{Z}_{ij}(\boldsymbol{R}_i,\ \boldsymbol{R}_j) = \boldsymbol{H}_{ij}\boldsymbol{R}_i\boldsymbol{R}_j^\top$. So the function (3.5) can be written as $L(\boldsymbol{y};\boldsymbol{\theta}) = \sum_{(i,j)\in\mathcal{E}} \log f_{ij}(\boldsymbol{y}_{ij}) + \frac{1}{2}\sum_{(i,j)\in\mathcal{F}} \log \widehat{f_{ij}}(\boldsymbol{Z}_{ij})$.

The detailed derivation of the FIM is shown as follows:

$\mathbb{R}^2$ *sub-matrix*

We deal with the inner product corresponding to the bases of the coordinate of the i-th pose as a $\mathbb{R}^{n\times 1},\ n = 2, 3$ vector, of which 2D case satisfying:

$$\langle \mathrm{grad} L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_i^{\boldsymbol{x}}\rangle_{\boldsymbol{X}}^* = \left(\langle \mathrm{grad} L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_{i,1}^{\boldsymbol{x}}\rangle_{\boldsymbol{X}}, \langle \mathrm{grad} L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_{i,2}^{\boldsymbol{x}}\rangle_{\boldsymbol{X}}\right)^\top, \tag{B.1}$$

*In this paper, we only present the simplified version of the derivation process. A longer detailed version is shown in the technical report [32].

Then, based on the definition (3.12), we have:

$$\begin{aligned}\boldsymbol{F}_{i,i_1} =&\mathbb{E}\{\langle \mathrm{grad} L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_i^{\boldsymbol{x}}\rangle_{\boldsymbol{X}}^* \cdot \langle \mathrm{grad} L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_{i_1}^{\boldsymbol{x}}\rangle_{\boldsymbol{X}}^{*\top}\} \\ =&\mathbb{E}\{\mathrm{grad}_{\boldsymbol{x}_i} L(\boldsymbol{y};\boldsymbol{\theta})^\top \cdot \mathrm{grad}_{\boldsymbol{x}_{i_1}} L(\boldsymbol{y};\boldsymbol{\theta})\},\end{aligned} \tag{B.2}$$

where $\boldsymbol{F}_{i,i_1} \in \mathbb{R}^{2\times 2}$, $\mathrm{grad}_{\boldsymbol{x}_i} L(\boldsymbol{y};\boldsymbol{\theta})$ means the gradient of $L(\boldsymbol{y};\ \boldsymbol{\theta})$ with respect to parameter $\boldsymbol{x}_i$.

Based on the log-likelihood function (3.5) and the definition (B.2), we have:

$$\begin{aligned}\mathrm{grad}_{\boldsymbol{x}_i} L(\boldsymbol{y};\boldsymbol{\theta}) =&\mathrm{grad}_{\boldsymbol{x}_i} \sum_{j\in V_i^+} \log f_{ij}(\boldsymbol{y}_{ij}) + \mathrm{grad}_{\boldsymbol{x}_i} \sum_{j\in V_i^-} \log f_{ji}(\boldsymbol{y}_{ji}) \\ =& \sum_{j\in V_i^+} \nabla_{\boldsymbol{y}_{ij}} \log f_{ij}\left(\boldsymbol{y}_{ij}\right) \nabla_{\boldsymbol{x}_i} \boldsymbol{y}_{ij} + \sum_{j\in V_i^-} \nabla_{\boldsymbol{y}_{ji}} \log f_{ji}\left(\boldsymbol{y}_{ji}\right) \nabla_{\boldsymbol{x}_i} \boldsymbol{y}_{ji} \\ =& -\sum_{j\in V_i^+} \delta_{ij}^{-2} \boldsymbol{y}_{ij}^\top \boldsymbol{R}_i^\top + \sum_{j\in V_i^-} \delta_{ji}^{-2} \boldsymbol{y}_{ji}^\top \boldsymbol{R}_j^\top .\end{aligned} \tag{B.3}$$

Situation 1 When $i = i_1$, based on Lemma 5 and $\mathbb{E}\{\boldsymbol{y}_{ij}\} = \mathbb{E}\{\boldsymbol{y}_{ji}\} = \boldsymbol{0}$, substitute the gradient (B.3) into the definition (B.2), we have:

$$\boldsymbol{F}_{i,i_1} =\mathbb{E}\{ \sum_{j\in V_i^+} \delta_{ij}^{-4} \boldsymbol{R}_i \boldsymbol{y}_{ij} \boldsymbol{y}_{ij}^\top \boldsymbol{R}_i^\top + \sum_{j\in V_i^-} \delta_{ji}^{-4} \boldsymbol{R}_j \boldsymbol{y}_{ji} \boldsymbol{y}_{ji}^\top \boldsymbol{R}_j^\top \}. \tag{B.4}$$

Because $\mathbb{E}\{\boldsymbol{y}_{ij}\boldsymbol{y}_{ij}^\top\} = \delta_{ij}^2 \boldsymbol{I}_{2\times 2}$ and $\mathbb{E}\{\boldsymbol{y}_{ji}\boldsymbol{y}_{ji}^\top\} = \delta_{ji}^2 \boldsymbol{I}_{2\times 2}$, we have:

$$\boldsymbol{F}_{i,i_1} = \sum_{j\in V_i^+} \delta_{ij}^{-2} \boldsymbol{I}_{2\times 2} + \sum_{j\in V_i^-} \delta_{ji}^{-2} \boldsymbol{I}_{2\times 2}. \tag{B.5}$$

Situation 2 When $(i, i_1) \in \mathcal{E}$, based on Lemma 5 and $\mathbb{E}\{\boldsymbol{y}_{ij}\} = \mathbb{E}\{\boldsymbol{y}_{ji}\} = \mathbf{0}$, substitute the gradient (B.3) into the definition (B.2), we have:

$$\boldsymbol{F}_{i,i_1} = \mathbb{E}\{\left(-\delta_{ii_1}^{-2}\boldsymbol{y}_{ii_1}^{\top}\boldsymbol{R}_i^{\top}\right)^{\top} \cdot \left(\delta_{ii_1}^{-2}\boldsymbol{y}_{ii_1}^{\top}\boldsymbol{R}_i^{\top}\right)\} = -\delta_{ii_1}^{-2}\boldsymbol{I}_{2\times 2}. \tag{B.6}$$

Based on the sub-matrix (B.6), we have

$$\boldsymbol{F}_{i_1,i} = \boldsymbol{F}_{i,i_1}^{\top} = -\delta_{i_1 i}^{-2}\boldsymbol{I}_{2\times 2}. \tag{B.7}$$

Situation 3 When $(i, i_1) \notin \mathcal{E}$ and $i \neq i_1$, based on $\mathbb{E}\{\boldsymbol{y}_{ij}\boldsymbol{y}_{i_1 j_1}^{\top}\} = \mathbb{E}\{\boldsymbol{y}_{ij}\}\mathbb{E}\{\boldsymbol{y}_{i_1 j_1}^{\top}\} = \mathbf{0}$, we have $\boldsymbol{F}_{i,i_1} = \mathbf{0}_{2\times 2}$.

Combine the sub-matrices (B.5), (B.6) and (B.7), let $(\boldsymbol{L}_w^{\mathbb{R}^2})_{i,i_1} = \boldsymbol{F}_{i,i_1}$, we have the (i, i_1)-th block of the sub-matrix $\boldsymbol{L}_w^{\mathbb{R}^2}$ corresponding to the (i, i_1)-th block of the FIM:

$$(\boldsymbol{L}_w^{\mathbb{R}^2})_{i,i_1} = \begin{cases} \sum_{j_1 \in V_i^+} \delta_{ij_1}^{-2}\boldsymbol{I}_{2\times 2} + \sum_{j_2 \in V_i^-} \delta_{j_2 i}^{-2}\boldsymbol{I}_{2\times 2} & i = i_1 \\ -\delta_{ii_1}^{-2}\boldsymbol{I}_{2\times 2} & (i, i_1) \in \mathcal{E} \\ -\delta_{i_1 i}^{-2}\boldsymbol{I}_{2\times 2} & (i_1, i) \in \mathcal{E} \\ \mathbf{0}_{2\times 2} & \text{else,} \end{cases} \tag{B.8}$$

It can be seen that $\boldsymbol{L}_w^{\mathbb{R}^2} = \boldsymbol{L}_{w_{\mathbb{R}}} \otimes \boldsymbol{I}_{2\times 2}$, where:

$$(\boldsymbol{L}_{w_{\mathbb{R}}})_{i,i_1} = \begin{cases} \sum_{j \in V_i} w_{ij}^{\mathbb{R}} & i = i_1 \\ -w_{ii_1}^{\mathbb{R}} & (i, i_1) \in \mathcal{E} \\ -w_{i_1 i}^{\mathbb{R}} & (i_1, i) \in \mathcal{E} \\ 0 & \text{else,} \end{cases} \tag{B.9}$$

where $w_{ij}^{\mathbb{R}} = \delta_{ij}^{-2}$. Thus $\boldsymbol{L}_{w_{\mathbb{R}}}$ is a weighted Laplacian matrix.

*$SO(2)$ **by** $\mathbb{R}^2$ **coupling sub-matrix***

Based on the geometry of the parameter space and the definition in (3.12), we can get the block of the $SO(2)$ by $\mathbb{R}^2$ coupling sub-matrix:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,i_1} =&\mathbb{E}\{\langle \text{grad}L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_i^{\boldsymbol{R}}\rangle_{\boldsymbol{R}} \cdot \langle \text{grad}L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_{i_1}^{\boldsymbol{x}}\rangle_{\boldsymbol{X}}^{*\top}\} \\ =&\mathbb{E}\{\text{trace}(\text{grad}_{\boldsymbol{R}_i} L(\boldsymbol{y};\boldsymbol{\theta})^\top \cdot \boldsymbol{R}_i\boldsymbol{E}) \cdot \text{grad}_{\boldsymbol{x}_{i_1}} L(\boldsymbol{y};\boldsymbol{\theta})\},\end{aligned} \tag{B.10}$$

where $\boldsymbol{F}_{n_p+1+i,i_1} \in \mathbb{R}^{1\times 2}$, $\text{grad}_{\boldsymbol{R}_i} L(\boldsymbol{y};\boldsymbol{\theta})$ means the gradient of $L(\boldsymbol{y};\boldsymbol{\theta})$ with respect to parameter $\boldsymbol{R}_i$.

Based on the definition (B.10), we have:

$$\text{grad}_{\boldsymbol{R}_i} L(\boldsymbol{y};\boldsymbol{\theta}) = \text{grad}_{\boldsymbol{R}_i} \frac{1}{2} \sum_{(i,j)\in\mathcal{F}} \log \widehat{f}_{ij}(\boldsymbol{Z}_{ij}) + \text{grad}_{\boldsymbol{R}_i} \sum_{j\in V_i^+} \log f_{ij}(\boldsymbol{y}_{ij}). \tag{B.11}$$

For the first part $\text{grad}_{\boldsymbol{R}_i} \frac{1}{2} \sum_{(i,j)\in\mathcal{F}} \log \widehat{f}_{ij}(\boldsymbol{Z}_{ij})$, based on the PDF (3.4), we have:

$$\text{grad}_{\boldsymbol{R}_i} \frac{1}{2} \sum_{(i,j)\in\mathcal{F}} \log \widehat{f}_{ij}(\boldsymbol{Z}_{ij}) =\text{grad}_{\boldsymbol{R}_i} \sum_{j\in V_i} \log(\frac{1}{c_2(\kappa_{ij})} \exp(\kappa_{ij}\text{trace}(\boldsymbol{Z}_{ij}))). \tag{B.12}$$

It is noted that, because of the un-directed graph $\mathcal{F}$ and the coefficient $\frac{1}{2}$, an undirected edge can be seen as the combination of two directed edges, $(i,j) \in \mathcal{E}$ and $(j,i) \in \mathcal{E} \Leftrightarrow j \in V_i$.

Based on the PDF (3.4) for the measurement error $\boldsymbol{Z}_{ij}$, solving (3.3) for $\boldsymbol{Z}_{ij}$ and substituting into the PDF (3.4), we have:

$$\log \widehat{f}_{ij}(\boldsymbol{Z}_{ij}) = -\log(c_n(\kappa_{ij})) + \kappa_{ij}\text{trace}(\boldsymbol{H}_{ij}\boldsymbol{R}_i\boldsymbol{R}_j^\top). \tag{B.13}$$

Since the inner product on $\mathbb{R}^{n\times n}$ is $\langle \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2\rangle = \text{trace}(\boldsymbol{\Omega}_1^\top \boldsymbol{\Omega}_2)$, the gradient (B.13)

implies that the Euclidean gradients of the log-likelihood function are:

$$\nabla_{\boldsymbol{R}_i} \log \widehat{f}_{ij}(\boldsymbol{Z}_{ij}) = \kappa_{ij} \boldsymbol{H}_{ij}^\top \boldsymbol{R}_j. \tag{B.14}$$

Finally, based on the gradient on $SO(n)$ (3.10), we have the Riemannian gradient of the log-likelihood function:

$$\begin{aligned} \operatorname{grad}_{\boldsymbol{R}_i} \sum_{j \in V_i} \log \widehat{f}_{ij}(\boldsymbol{Z}_{ij}) &= \sum_{j \in V_i} \kappa_{ij} \boldsymbol{R}_i \operatorname{skew}\left(\boldsymbol{R}_i^\top \boldsymbol{H}_{ij}^\top \boldsymbol{R}_j\right) \\ &= \sum_{j \in V_i} \kappa_{ij} \boldsymbol{R}_i \operatorname{skew}(\boldsymbol{R}_j^\top \boldsymbol{Z}_{ij}^\top \boldsymbol{R}_j). \end{aligned} \tag{B.15}$$

For 2D case, because of $\boldsymbol{R}_j^\top \boldsymbol{Z}_{ij}^\top \boldsymbol{R}_j = \boldsymbol{Z}_{ij}^\top$, so we have:

$$\operatorname{grad}_{\boldsymbol{R}_i} \sum_{j \in V_i} \log \widehat{f}_{ij}(\boldsymbol{Z}_{ij}) = \sum_{j \in V_i} -\kappa_{ij} \boldsymbol{R}_i \operatorname{skew}(\boldsymbol{Z}_{ij}). \tag{B.16}$$

Let's consider the second part in the definition (B.11), we can first get the usual gradient of $f_{ij}(\boldsymbol{y}_{ij})$ in $\mathbb{R}^{n \times n}$:

$$\begin{aligned} \nabla_{\boldsymbol{R}_i \in \mathbb{R}^{n \times n}} \ \log \ f_{ij}(\boldsymbol{y}_{ij}) =& \nabla_{\boldsymbol{R}_i \in \mathbb{R}^{n \times n}} \log \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}|^{1/2}} \exp(-\frac{1}{2} \boldsymbol{y}_{ij}^\top \boldsymbol{\Sigma}_{ij}^{-1} \boldsymbol{y}_{ij})) \\ =& \nabla_{\boldsymbol{R}_i \in \mathbb{R}^{n \times n}} - \delta_{ij}^{-2} \frac{1}{2} \boldsymbol{y}_{ij}^\top \boldsymbol{y}_{ij} = \delta_{ij}^{-2} (\boldsymbol{x}_j - \boldsymbol{x}_i) \boldsymbol{y}_{ij}^\top. \end{aligned} \tag{B.17}$$

Based on the general definition of the gradient on $SO(2)$ shown in the gradient (3.10), let $\boldsymbol{Q} = \boldsymbol{R}_i$, we have:

$$\begin{aligned} \operatorname{grad}_{\boldsymbol{R}_i} \ \log f_{ij}(\boldsymbol{y}_{ij}) =& \boldsymbol{R}_i \operatorname{skew}(\boldsymbol{R}_i^\top \cdot \nabla_{\boldsymbol{R}_i \in \mathbb{R}^{n \times n}} \log \ f_{ij}(\boldsymbol{y}_{ij})) \\ =& \delta_{ij}^{-2} ((\boldsymbol{x}_j - \boldsymbol{x}_i) \boldsymbol{y}_{ij}^\top - \boldsymbol{R}_i \boldsymbol{y}_{ij} (\boldsymbol{x}_j - \boldsymbol{x}_i)^\top \boldsymbol{R}_i)/2. \end{aligned} \tag{B.18}$$

Re-write the above equation (B.18) based on the basis $\boldsymbol{R}_i\boldsymbol{E}$, we have [44]:

$$\begin{aligned}\mathrm{grad}_{\boldsymbol{R}_i}\sum_{j\in V_i^+}\log f_{ij}(\boldsymbol{y}_{ij}) &= \sum_{j\in V_i^+}\frac{1}{2}\boldsymbol{R}_i\boldsymbol{E}\nabla_{\boldsymbol{R}_i}\log f_{ij}(\boldsymbol{y}_{ij}) \\ &= \sum_{j\in V_i^+}\frac{1}{2}\boldsymbol{R}_i\boldsymbol{E}\left(\delta_{ij}^{-2}\boldsymbol{y}_{ij}^\top\nabla_{\boldsymbol{R}_i}\boldsymbol{y}_{ij}\right) \\ &= \sum_{j\in V_i^+}\frac{1}{2}\delta_{ij}^{-2}\boldsymbol{R}_i\boldsymbol{E}\left(\boldsymbol{y}_{ij}^\top\boldsymbol{E}^\top\boldsymbol{R}_i^\top(\boldsymbol{x}_j-\boldsymbol{x}_i)\right).\end{aligned} \tag{B.19}$$

For $\mathrm{grad}_{\boldsymbol{x}_{i_1}}L(\boldsymbol{y};\boldsymbol{\theta})$, based on the gradient (B.3), we have:

$$\mathrm{grad}_{\boldsymbol{x}_{i_1}}L(\boldsymbol{y};\boldsymbol{\theta}) = -\sum_{j_1\in V_{i_1}^+}\delta_{i_1j_1}^{-2}\boldsymbol{y}_{i_1j_1}^\top\boldsymbol{R}_{i_1}^\top + \sum_{j_1\in V_{i_1}^-}\delta_{j_1i_1}^{-2}\boldsymbol{y}_{j_1i_1}^\top\boldsymbol{R}_{j_1}^\top. \tag{B.20}$$

Substitute the gradient functions (B.16) (B.19) and (B.20) into the definition (B.10), let:

$$g_{ij} = (\boldsymbol{x}_i-\boldsymbol{x}_j)^\top\boldsymbol{R}_i\boldsymbol{E}\boldsymbol{y}_{ij}\in\mathbb{R}, \tag{B.21}$$

based on Lemma 5 and $\mathbb{E}\{\boldsymbol{y}_{ij}\}=\boldsymbol{0}$, we have:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,i_1} =&\mathbb{E}\{\mathrm{trace}(\mathrm{grad}_{\boldsymbol{R}_i}L(\boldsymbol{y};\boldsymbol{\theta})^\top\cdot\boldsymbol{R}_i\boldsymbol{E})\cdot\boldsymbol{I}_{2\times2}^\top\cdot\mathrm{grad}_{\boldsymbol{x}_{i_1}}L(\boldsymbol{y};\boldsymbol{\theta})\} \\ =&\boldsymbol{0}+\sum_{j\in V_i^+}\sum_{j_1\in V_{i_1}^+}\mathbb{E}\{\delta_{ij}^{-2}\delta_{i_1j_1}^{-2}\mathrm{trace}\,(g_{ij})\,\boldsymbol{y}_{i_1j_1}^\top\boldsymbol{R}_{i_1}^\top\} \\ &-\sum_{j\in V_i^+}\sum_{j_1\in V_{i_1}^-}\mathbb{E}\{\delta_{ij}^{-2}\delta_{j_1i_1}^{-2}\mathrm{trace}\,(g_{ij})\,\boldsymbol{y}_{j_1i_1}^\top\boldsymbol{R}_{j_1}^\top\}.\end{aligned} \tag{B.22}$$

Situation 1

Using $\boldsymbol{E}\boldsymbol{R}_i = \boldsymbol{R}_i\boldsymbol{E}$, if $i = i_1$, we have:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,i_1} &= \sum_{j\in V_i^+}\sum_{j_1\in V_{i_1}^+}\mathbb{E}\{\delta_{ij}^{-2}\delta_{i_1j_1}^{-2}\operatorname{trace}(g_{ij})\,\boldsymbol{y}_{i_1j_1}^\top\boldsymbol{R}_{i_1}^\top\} + \mathbf{0}\\ &= \sum_{j\in V_i^+}\delta_{ij}^{-4}(\boldsymbol{x}_i-\boldsymbol{x}_j)^\top\boldsymbol{E}\boldsymbol{R}_i\mathbb{E}\{\boldsymbol{y}_{ij}\boldsymbol{y}_{ij}^\top\}\boldsymbol{R}_i^\top\\ &= \sum_{j\in V_i^+}\delta_{ij}^{-2}(\boldsymbol{x}_i-\boldsymbol{x}_j)^\top\boldsymbol{E}.\end{aligned} \tag{B.23}$$

Situation 2

If $(i,i_1)\in\mathcal{E}$, we have:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,i_1} =&\mathbf{0} - \sum_{j\in V_i^+}\sum_{j_1\in V_{i_1}^-}\mathbb{E}\{\delta_{ij}^{-2}\delta_{j_1i_1}^{-2}\operatorname{trace}(g_{ij})\,\boldsymbol{y}_{j_1i_1}^\top\boldsymbol{R}_{j_1}^\top\}\\ =&\delta_{ii_1}^{-4}(\boldsymbol{x}_{i_1}-\boldsymbol{x}_i)^\top\boldsymbol{E}\boldsymbol{R}_i\mathbb{E}\{\boldsymbol{y}_{ij}\boldsymbol{y}_{j_1i_1}^\top\}\boldsymbol{R}_{j_1}^\top\\ =&\delta_{ii_1}^{-2}(\boldsymbol{x}_{i_1}-\boldsymbol{x}_i)^\top\boldsymbol{E}.\end{aligned} \tag{B.24}$$

Situation 3 When $(i,i_1)\notin\mathcal{E}$ and $i\neq i_1$, based on $\mathbb{E}\{\boldsymbol{y}_{ij}\boldsymbol{y}_{i_1j_1}^\top\} = \mathbb{E}\{\boldsymbol{y}_{ij}\}\mathbb{E}\{\boldsymbol{y}_{i_1j_1}^\top\} = \mathbf{0}_{2\times2}$, we have $\boldsymbol{F}_{n_p+1+i,i_1} = \mathbf{0}_{1\times2}$.

Combining the above situations, let $(\triangle_w)_{i,i_1} = \boldsymbol{F}_{n_p+1+i,i_1}$, we have the $SO(2)$ by $\mathbb{R}^2$ coupling sub-matrix (3.16).

$\mathbb{R}^2$ ***by*** $SO(2)$ ***coupling sub-matrix***

Because the FIM is symmetrical, we can get the block of the $\mathbb{R}^2$ by $SO(2)$ coupling sub-matrix: $\boldsymbol{F}_{i_1,n_p+1+i} = \boldsymbol{F}_{n_p+1+i,i_1}^\top$.

$SO(2)$ ***sub-matrix***

Using the definition in (3.12), we can get the block of the $SO(2)$ sub-matrix:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,n_p+1+i_1} =&\mathbb{E}\{\langle \text{grad}L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_i^{\boldsymbol{R}}\rangle_{\boldsymbol{R}} \cdot \langle \text{grad}L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_{i_1}^{\boldsymbol{R}}\rangle_{\boldsymbol{R}}^{\top}\} \\ =&\mathbb{E}\{\text{trace}(\text{grad}_{\boldsymbol{R}_i} L(\boldsymbol{y};\boldsymbol{\theta})^{\top}\boldsymbol{R}_i\boldsymbol{E}) \cdot \text{trace}(\text{grad}_{\boldsymbol{R}_{i_1}} L(\boldsymbol{y};\boldsymbol{\theta})^{\top}\boldsymbol{R}_{i_1}\boldsymbol{E})^{\top}\},\end{aligned} \tag{B.25}$$

where $\boldsymbol{F}_{n_p+1+i,n_p+1+i_1} \in \mathbb{R}^{1\times 1}$.

Based on the definition (B.25), the gradient (B.16), (B.19), Lemma 5 and $\mathbb{E}\{g_{ij}\} = 0$, we have:

$$\begin{aligned}&\boldsymbol{F}_{n_p+1+i,n_p+1+i_1} = \sum_{j\in V_i^+}\sum_{j_1\in V_{i_1}^+} \mathbb{E}\{\delta_{ij}^{-2}\delta_{i_1j_1}^{-2} g_{ij}g_{i_1j_1}^{\top}\}+ \\ &\mathbb{E}\{\sum_{j\in V_i} \kappa_{ij}\text{trace}(\text{skew}(\boldsymbol{Z}_{ij})\boldsymbol{E}) \sum_{j_1\in V_{i_1}} \kappa_{i_1j_1}\text{trace}(\text{skew}(\boldsymbol{Z}_{i_1j_1})\boldsymbol{E})\}.\end{aligned} \tag{B.26}$$

Set $\boldsymbol{Z}_{ij} = \begin{bmatrix} c_{ij} & -s_{ij} \\ s_{ij} & c_{ij} \end{bmatrix}$, where $c_{ij} = \cos(\alpha_{ij})$ and $s_{ij} = \sin(\alpha_{ij})$, α_{ij} is the minimal representation [44], we have:

$$\text{trace}(\text{skew}(\boldsymbol{Z}_{ij})\boldsymbol{E}) = -2\kappa_{ij}s_{ij}. \tag{B.27}$$

Substitute (B.27) into the sub-matrix (B.26), we have:

$$\boldsymbol{F}_{n_p+1+i,n_p+1+i_1} = \sum_{j\in V_i^+}\sum_{j_1\in V_{i_1}^+} \delta_{ij}^{-2}\delta_{i_1j_1}^{-2}\mathbb{E}\{g_{ij}g_{i_1j_1}^{\top}\} + \sum_{j\in V_i}\sum_{j\in V_{i_1}} 4\kappa_{ij}\kappa_{i_1j_1}\mathbb{E}\{s_{ij}s_{i_1j_1}\}. \tag{B.28}$$

Because when the minimal representation α_{ij} gets close to 0, it satisfies $\alpha_{ij} \approx s_{ij}$. In the 2D plane, the isotropic Langevin distribution is similar to the Gaussian distribution, we can regard it as: $s_{ij} \sim \mathcal{N}(0, \frac{1}{2\kappa_{ij}})$ [116]. In fact, the accurate value

of $\mathbb{E}\{s_{ij}s_{i_1j_1}\}$ is $\frac{I_1(2\kappa_{ij})}{2\kappa_{ij}I_0(2\kappa_{ij})}$ refereed to Appendix B.1.3. When κ_{ij} becomes large, we will have: $\frac{I_1(2\kappa_{ij})}{2\kappa_{ij}I_0(2\kappa_{ij})} \to \frac{1}{2\kappa_{ij}}$.

Situation 1 When $i = i_1$, we have:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,n_p+1+i_1} &= \sum_{j\in V_i^+} \delta_{ij}^{-4}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \mathbb{E}\{\boldsymbol{y}_{ij}\boldsymbol{y}_{ij}^\top\}(\boldsymbol{x}_i - \boldsymbol{x}_j) + \sum_{j\in V_i} 4\kappa_{ij}^2 \frac{I_1(2\kappa_{ij})}{2\kappa_{ij}I_0(2\kappa_{ij})} \\ &= \sum_{j\in V_i} 2\kappa_{ij}\frac{I_1(2\kappa_{ij})}{I_0(2\kappa_{ij})} + \sum_{j\in V_i^+} \delta_{ij}^{-2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2.\end{aligned} \tag{B.29}$$

Situation 2 When $i \neq i_1$ and $(i, i_1) \in \mathcal{E}$, use $s_{ii_1} = -s_{i_1i}$, we have:

$$\boldsymbol{F}_{n_p+1+i,n_p+1+i_1} = \begin{cases} -2\kappa_{ii_1}\frac{I_1(2\kappa_{ii_1})}{I_0(2\kappa_{ii_1})} & (i, i_1) \in \mathcal{E} \\ -2\kappa_{i_1i}\frac{I_1(2\kappa_{i_1i})}{I_0(2\kappa_{i_1i})} & (i_1, i) \in \mathcal{E}. \end{cases} \tag{B.30}$$

Situation 3 When $(i, i_1) \notin \mathcal{E}$ and $i \neq i_1$, based on $\mathbb{E}\{\boldsymbol{s}_{ij}\boldsymbol{s}_{i_1j_1}\} = \mathbb{E}\{\boldsymbol{s}_{ij}\}\mathbb{E}\{\boldsymbol{s}_{i_1j_1}\} = 0$, we have $\boldsymbol{F}_{n_p+1+i,n_p+1+i_1} = 0$.

Finally, we have the $(n_p + 1 + i, n_p + 1 + i_1)$-th block of the FIM corresponding to the $SO(2)$ sub-matrix written as:

$$\boldsymbol{F}_{n_p+1+i,n_p+1+i_1} = \begin{cases} \sum_{j\in V_i} 2\kappa_{ij}\frac{I_1(2\kappa_{ij})}{I_0(2\kappa_{ij})} + \psi_i & i = i_1 \\ -2\kappa_{ii_1}\frac{I_1(2\kappa_{ii_1})}{I_0(2\kappa_{ii_1})} & (i, i_1) \in \mathcal{E} \\ -2\kappa_{i_1i}\frac{I_1(2\kappa_{i_1i})}{I_0(2\kappa_{i_1i})} & (i_1, i) \in \mathcal{E} \\ 0 & \text{else}, \end{cases} \tag{B.31}$$

$$\psi_i = \sum_{j\in V_i^+} \delta_{ij}^{-2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2.$$

When ψ_i is ignored, it is easy to find that the $SO(2)$ sub-matrix is a weighted

Laplacian matrix. The (i, i_1)-th block of the weighted Laplacian matrix $\boldsymbol{L}_{w_{SO(2)}}$ is written as:

$$(\boldsymbol{L}_{w_{SO(2)}})_{i,i_1} = \begin{cases} \sum_{j \in V_i} w_{ij}^{SO(2)} & i = i_1 \\ -w_{ii_1}^{SO(2)} & (i, i_1) \in \mathcal{E} \\ -w_{i_1 i}^{SO(2)} & (i_1, i) \in \mathcal{E} \\ 0 & \text{else,} \end{cases} \tag{B.32}$$

where $w_{ij}^{SO(2)} = 2\kappa_{ij} \frac{I_1(2\kappa_{ij})}{I_0(2\kappa_{ij})}$.

The whole FIM

Combine the sub-matrices (3.16), (B.8), and (B.31), we can get the complete FIM (3.15). So Theorem 1 is proved.

B.1.2 Proof of Theorem 2

In this appendix, we will present the derivations of the FIM for 3D pose-graph SLAM (prove Theorem 2). The detailed process to get four sub-matrices is shown as follows:

$\mathbb{R}^3$ ***sub-matrix***

The derivation of this sub-matrix is similar to Appendix B.1.1. Based on the $\mathbb{R}^2$ sub-matrix (B.8), we can directly obtain the sub-matrix $\boldsymbol{L}_w^{\mathbb{R}^3}$ corresponding to Euclidean space $\mathbb{R}^3$:

$$\boldsymbol{L}_w^{\mathbb{R}^3} = \boldsymbol{L}_{w_{\mathbb{R}}} \otimes \boldsymbol{I}_{3\times 3}. \tag{B.33}$$

*$SO(3)$ **by** $\mathbb{R}^3$ **coupling sub-matrix***

Based on the geometry of the parameter space and the definition in (3.12), we can get the block of the $SO(3)$ by $\mathbb{R}^3$ coupling sub-matrix:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,i_1}(k) =&\mathbb{E}\{\langle \text{grad}L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_{i,k}^{\boldsymbol{R}}\rangle_{\boldsymbol{R}} \cdot \langle \text{grad}L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_{i_1}^{\boldsymbol{X}}\rangle_{\boldsymbol{X}}^{*\top}\} \\ =&\mathbb{E}\{\text{trace}(\text{grad}_{\boldsymbol{R}_i} L(\boldsymbol{y};\boldsymbol{\theta})^\top \cdot \boldsymbol{R}_i\boldsymbol{E}_k) \cdot \text{grad}_{\boldsymbol{x}_{i_1}} L(\boldsymbol{y};\boldsymbol{\theta})\},\end{aligned} \tag{B.34}$$

where $\boldsymbol{F}_{n_p+1+i,i_1}(k) \in \mathbb{R}^{1\times 3}$ means the k-th row of the (n_p+1+i, i_1)-th block of the FIM.

Based on the definition (B.34), we have the gradient function (B.11). Based on the gradient (B.16), we can obtain the first part of the gradient function (B.11) as:

$$\text{grad}_{\boldsymbol{R}_i} \frac{1}{2} \sum_{(i,j)\in\mathcal{F}} \log \widehat{f_{ij}}(\boldsymbol{Z}_{ij}) = \sum_{j\in V_i} -\kappa_{ij}\boldsymbol{R}_i \text{skew}(\boldsymbol{R}_j^\top \boldsymbol{Z}_{ij}\boldsymbol{R}_j). \tag{B.35}$$

Let's consider the second part, based on the basis $\boldsymbol{R}_i\boldsymbol{E}_k$ [44]:

$$\begin{aligned}\text{grad}_{\boldsymbol{R}_i} \sum_{j\in V_i^+} \log f_{ij}(\boldsymbol{y}_{ij}) &= \sum_{j\in V_i^+} \sum_{k=1}^{3} \boldsymbol{R}_i\boldsymbol{E}_k \nabla_{\boldsymbol{R}_i\boldsymbol{E}_k} \log f_{ij}(\boldsymbol{y}_{ij}) \\ &= \sum_{j\in V_i^+} \delta_{ij}^{-2} \sum_{k=1}^{3} \boldsymbol{R}_i\boldsymbol{E}_k(\boldsymbol{y}_{ij}^\top \boldsymbol{E}_k^\top \boldsymbol{R}_i^\top(\boldsymbol{x}_j - \boldsymbol{x}_i)).\end{aligned} \tag{B.36}$$

Substitute the gradient (B.20), (B.35), and (B.36) into the definition (B.34), let $\boldsymbol{K}_{ij} = (\sum_{k=1}^{3} \boldsymbol{R}_i\boldsymbol{E}_k\boldsymbol{y}_{ij}^\top \boldsymbol{E}_k^\top \boldsymbol{R}_i^\top(\boldsymbol{x}_i - \boldsymbol{x}_j))^\top \boldsymbol{R}_i\boldsymbol{E}_k \in \mathbb{R}^{3\times 3}$, based on Lemma 5 and

$\mathbb{E}\{\boldsymbol{y}_{ij}\} = \mathbf{0}$, we have:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,i_1}(k) =&\mathbb{E}\{\text{trace}(\text{grad}_{\boldsymbol{R}_i} L(\boldsymbol{y};\boldsymbol{\theta})^\top \cdot \boldsymbol{R}_i \boldsymbol{E}_k) \cdot \text{grad}_{\boldsymbol{x}_{i_1}} L(\boldsymbol{y};\boldsymbol{\theta})\} \\ =&\mathbf{0} + \sum_{j\in V_i^+}\sum_{j_1\in V_{i_1}^+} \mathbb{E}\{\delta_{ij}^{-2}\delta_{i_1j_1}^{-2}\text{trace}\left(\boldsymbol{K}_{ij}\right)\boldsymbol{y}_{i_1j_1}^\top \boldsymbol{R}_{i_1}^\top\} \\ &- \sum_{j\in V_i^+}\sum_{j_1\in V_{i_1}^-} \mathbb{E}\{\delta_{ij}^{-2}\delta_{j_1i_1}^{-2}\text{trace}\left(\boldsymbol{K}_{ij}\right)\boldsymbol{y}_{j_1i_1}^\top \boldsymbol{R}_{j_1}^\top\}.\end{aligned} \tag{B.37}$$

Situation 1 If $i = i_1$, we have:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,i_1}(k) &= \sum_{j\in V_i^+}\sum_{j_1\in V_{i_1}^+} \mathbb{E}\{\delta_{ij}^{-2}\delta_{i_1j_1}^{-2}\text{trace}\left(\boldsymbol{K}_{ij}\right)\boldsymbol{y}_{i_1j_1}^\top \boldsymbol{R}_{i_1}^\top\} + \mathbf{0} \\ &= \sum_{j\in V_i^+} \delta_{ij}^{-4}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{E}_k \mathbb{E}\{\boldsymbol{y}_{ij}\boldsymbol{y}_{ij}^\top\}\boldsymbol{R}_i^\top \\ &= \sum_{j\in V_i^+} \delta_{ij}^{-2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{E}_k \boldsymbol{R}_i^\top.\end{aligned} \tag{B.38}$$

Situation 2 If $(i, i_1) \in \mathcal{E}$, we have:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,i_1}(k) =&\mathbf{0} - \sum_{j\in V_i^+}\sum_{j_1\in V_{i_1}^-} \mathbb{E}\{\delta_{ij}^{-2}\delta_{j_1i_1}^{-2}\text{trace}\left(\boldsymbol{K}_{ij}\right)\boldsymbol{y}_{j_1i_1}^\top \boldsymbol{R}_{j_1}^\top\} \\ =&\delta_{ij}^{-4}(\boldsymbol{x}_j - \boldsymbol{x}_i)^\top \boldsymbol{R}_i \boldsymbol{E}_k \mathbb{E}\{\boldsymbol{y}_{ij}\boldsymbol{y}_{j_1i_1}^\top\}\boldsymbol{R}_{j_1}^\top \\ =&\delta_{ij}^{-2}(\boldsymbol{x}_j - \boldsymbol{x}_i)^\top \boldsymbol{R}_i \boldsymbol{E}_k \boldsymbol{R}_i^\top.\end{aligned} \tag{B.39}$$

So, let $(\triangle_w^{3D})_{i,i_1} = \boldsymbol{F}_{n_p+1+i,i_1}$, we have the (i, i_1)-th block of the $SO(3)$ by $\mathbb{R}^3$

coupling sub-matrix $\triangle_w$ corresponding to the (n_p+1+i, i_1)-th block of the FIM:

$$(\triangle_w^{3D})_{i,i_1} = \begin{cases} \begin{bmatrix} \sum_{j\in V_i^+} \delta_{ij}^{-2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{E}_1 \boldsymbol{R}_i^\top \\ \sum_{j\in V_i^+} \delta_{ij}^{-2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{E}_2 \boldsymbol{R}_i^\top \\ \sum_{j\in V_i^+} \delta_{ij}^{-2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{E}_3 \boldsymbol{R}_i^\top \end{bmatrix} & i = i_1 \\ \begin{bmatrix} \delta_{ii_1}^{-2}(\boldsymbol{x}_{i_1} - \boldsymbol{x}_i)^\top \boldsymbol{R}_i \boldsymbol{E}_1 \boldsymbol{R}_i^\top \\ \delta_{ii_1}^{-2}(\boldsymbol{x}_{i_1} - \boldsymbol{x}_i)^\top \boldsymbol{R}_i \boldsymbol{E}_2 \boldsymbol{R}_i^\top \\ \delta_{ii_1}^{-2}(\boldsymbol{x}_{i_1} - \boldsymbol{x}_i)^\top \boldsymbol{R}_i \boldsymbol{E}_3 \boldsymbol{R}_i^\top \end{bmatrix} & (i, i_1) \in \mathcal{E} \\ \boldsymbol{0}_{3\times 3} & \text{else,} \end{cases}. \tag{B.40}$$

$\mathbb{R}^3$ *by* $SO(3)$ *coupling sub-matrix*

Because the FIM is symmetrical, we can get the block of the $\mathbb{R}^3$ by $SO(3)$ coupling sub-matrix: $\boldsymbol{F}_{i_1,n_p+1+i} = \boldsymbol{F}_{n_p+1+i,i_1}^\top$.

$SO(3)$ *sub-matrix*

Using the definition in (3.12), we can get the block of the $SO(3)$ sub-matrix:

$$\begin{aligned} \boldsymbol{F}_{n_p+1+i,n_p+1+i_1}(k,l) =& \mathbb{E}\{\langle \mathrm{grad} L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_{i,k}^{\boldsymbol{R}} \rangle_{\boldsymbol{R}} \cdot \langle \mathrm{grad} L(\boldsymbol{y};\boldsymbol{\theta}), \boldsymbol{E}_{i_1,l}^{\boldsymbol{R}} \rangle_{\boldsymbol{R}}^\top \} \\ =& \mathbb{E}\{\langle \mathrm{grad}_{\boldsymbol{R}_i} L(\boldsymbol{y};\boldsymbol{\theta})^\top, \boldsymbol{R}_i \boldsymbol{E}_k \rangle_{\boldsymbol{R}} \cdot \langle \mathrm{grad}_{\boldsymbol{R}_{i_1}} L(\boldsymbol{y};\boldsymbol{\theta})^\top, \boldsymbol{R}_{i_1} \boldsymbol{E}_l \rangle_{\boldsymbol{R}} \}, \end{aligned} \tag{B.41}$$

where $\boldsymbol{F}_{n_p+1+i,n_p+1+i_1}(k,l) \in \mathbb{R}$ means the (k,l)-th element of the (n_p+1+i, n_p+1+i_1)-th block of the FIM.

Based on the definition (B.41), the gradient (B.35) and (B.36), Lemma 5 and

$\mathbb{E}\{\widehat{g_{ij,k}}\} = 0$, let $\widehat{g_{ij,k}} = (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{E}_k \boldsymbol{y}_{ij}$, we have:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,n_p+1+i_1}(k,l) &= \sum_{j\in V_i^+}\sum_{j_1\in V_{i_1}^+} \mathbb{E}\{\delta_{ij}^{-2}\delta_{i_1j_1}^{-2}\widehat{g_{ij,k}}\widehat{g_{i_1j_1,l}}^\top\} \\ &+ \mathbb{E}\{\sum_{j\in V_i}\kappa_{ij}\langle \boldsymbol{R}_i\text{skew}(\boldsymbol{R}_j^\top \boldsymbol{Z}_{ij}^\top \boldsymbol{R}_j), \boldsymbol{R}_i\boldsymbol{E}_k\rangle_{\boldsymbol{R}}\cdot \\ &\sum_{j_1\in V_{i_1}}\kappa_{i_1j_1}\langle \boldsymbol{R}_{i_1}\text{skew}(\boldsymbol{R}_{j_1}^\top \boldsymbol{Z}_{i_1j_1}^\top \boldsymbol{R}_{j_1}), \boldsymbol{R}_{i_1}\boldsymbol{E}_l\rangle_{\boldsymbol{R}}\}.\end{aligned} \tag{B.42}$$

Let's consider the following mapping $G_{ij}(\boldsymbol{Z}_{ij}) = \boldsymbol{R}_i\text{skew}(\boldsymbol{R}_j^\top \boldsymbol{Z}_{ij}^\top \boldsymbol{R}_j)\boldsymbol{R}_i^\top$ and its three bases $\boldsymbol{R}_i\boldsymbol{E}_1\boldsymbol{R}_i^\top$, $\boldsymbol{R}_i\boldsymbol{E}_2\boldsymbol{R}_i^\top$, and $\boldsymbol{R}_i\boldsymbol{E}_3\boldsymbol{R}_i^\top$. We have:

$$G_{ij}(\boldsymbol{Z}_{ij}) = \sum_{k=1}^{3} \Phi_k^{ij}\cdot \boldsymbol{R}_i\boldsymbol{E}_k\boldsymbol{R}_i^\top, \tag{B.43}$$

where $\Phi_k^{ij} = \langle \boldsymbol{R}_i\text{skew}(\boldsymbol{R}_j^\top \boldsymbol{Z}_{ij}^\top \boldsymbol{R}_j)\boldsymbol{R}_i^\top, \boldsymbol{R}_i\boldsymbol{E}_k\boldsymbol{R}_i^\top\rangle_{\boldsymbol{R}}$.

Lemma 6. *If* $\boldsymbol{Z}_{ij} \sim Lang(\boldsymbol{I}_{n\times n}, \kappa_{ij}), \boldsymbol{\Phi}_{ij} = [\Phi_1^{ij}, \Phi_2^{ij}, \Phi_3^{ij}]^\top$, *we have:* $\mathbb{E}\{\Phi_k^{ij}\} = \mathbb{E}\{\Phi_l^{ij}\} = 0,\ \mathbb{E}\{\Phi_k^{ij}\Phi_l^{ij}\} = 0,\ \mathbb{E}\{{\Phi_k^{ij}}^2\} = \mathbb{E}\{{\Phi_l^{ij}}^2\} = \frac{\mathbb{E}\{\|skew(\boldsymbol{Z}_{ij})\|_F^2\}}{3}$.

Proof. See Appendix B.1.4. □

Situation 1 If $i = i_1$ and $k = l$, based on the sub-matrix (B.42), and Lemma 6, we have:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,n_p+1+i_1}(k,k) &= \sum_{j\in V_i}\kappa_{ij}^2\mathbb{E}\{{\Phi_k^{ij}}^2\} \\ &+ \sum_{j\in V_i^+}\delta_{ij}^{-4}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i\boldsymbol{E}_k\mathbb{E}\{\boldsymbol{y}_{ij}\boldsymbol{y}_{ij}^\top\}\boldsymbol{E}_k^\top\boldsymbol{R}_i^\top(\boldsymbol{x}_i - \boldsymbol{x}_j) \\ &= \sum_{j\in V_i}\frac{\omega_{ij}}{3} + \sum_{j\in V_i^+}\delta_{ij}^{-2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i\boldsymbol{I}_{3\times3}^{k,k}\boldsymbol{R}_i^\top(\boldsymbol{x}_i - \boldsymbol{x}_j).\end{aligned} \tag{B.44}$$

where $\boldsymbol{I}_{3\times3}^{k,k} = \boldsymbol{E}_k\boldsymbol{E}_k^\top$, $\boldsymbol{I}_{3\times3}^{k,l} = \boldsymbol{E}_k\boldsymbol{E}_l^\top$.

$$\omega_{ij} = \frac{\kappa_{ij}^2\left(2I_0(2\kappa_{ij}) - I_1(2\kappa_{ij}) - 2I_2(2\kappa_{ij}) + I_3(2\kappa_{ij})\right)}{2I_0(2\kappa_{ij}) - 2I_1(2\kappa_{ij})}, \tag{B.45}$$

where the specific derivation is shown in Appendix B.1.5.

Situation 2 If $i = i_1$ and $k \neq l$, based on the sub-matrix (B.42) and Lemma 6, $\mathbb{E}\{\Phi_k^{ij}\} = \mathbb{E}\{\Phi_k^{ij}\}\mathbb{E}\{\Phi_l^{ij}\} = 0$, we have:

$$\begin{aligned}\boldsymbol{F}_{n_p+1+i,n_p+1+i_1}(k,l) &= 0 + \sum_{j\in V_i^+} \delta_{ij}^{-4}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{E}_k \mathbb{E}\{\boldsymbol{y}_{ij}\boldsymbol{y}_{ij}^\top\}\boldsymbol{E}_l^\top \boldsymbol{R}_i^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) \\ &= \sum_{j\in V_i^+} \delta_{ij}^{-2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{I}_{3\times 3}^{k,l} \boldsymbol{R}_i^\top (\boldsymbol{x}_i - \boldsymbol{x}_j).\end{aligned} \tag{B.46}$$

Situation 3 If $(i, i_1) \in \mathcal{E}$ and $k = l$, based on the sub-matrix (B.42) and Lemma 6, we have:

$$\boldsymbol{F}_{n_p+1+i,n_p+1+i_1}(k,l) = -\kappa_{ii_1}^2 \mathbb{E}\{{\Phi_k^{ii_1}}^2\} = -\frac{\omega_{ii_1}}{3}. \tag{B.47}$$

Situation 4 If $(i_1, i) \in \mathcal{E}$ and $k = l$, based on the sub-matrix (B.42), we have:

$$\boldsymbol{F}_{n_p+1+i,n_p+1+i_1}(k,l) = \boldsymbol{F}_{n_p+1+i_1,n_p+1+i}(k,l)^\top \tag{B.48}$$

Finally, the $(n_p + 1 + i, n_p + 1 + i_1)$-th block of the FIM corresponding to the $SO(3)$ sub-matrix are written as:

$$\boldsymbol{F}_{n_p+1+i,n_p+1+i_1} = \begin{cases} \sum_{j\in V_i} \frac{\omega_{ij}}{3}\boldsymbol{I}_{3\times 3} + \boldsymbol{\Psi}_i & i = i_1 \\ -\frac{\omega_{ii_1}}{3}\boldsymbol{I}_{3\times 3} & (i,i_1) \in \mathcal{E} \\ -\frac{\omega_{i_1 i}}{3}\boldsymbol{I}_{3\times 3} & (i_1,i) \in \mathcal{E} \\ \boldsymbol{0}_{3\times 3} & \text{else,} \end{cases} \tag{B.49}$$

where $\boldsymbol{F}_{n_p+1+i,n_p+1+i_1} \in \mathbb{R}^{3\times 3}$, $\boldsymbol{\Psi}_i$ is shown in the equation (3.20). When $\boldsymbol{\Psi}_i$ is ignored, it is easy to find that the $SO(3)$ sub-matrix can be written as $\boldsymbol{L}_w^{SO(3)} =$

$\boldsymbol{L}_{w_{SO(3)}} \otimes \boldsymbol{I}_{3\times 3}$. The (i, i_1)-th block of the weighted Laplacian matrix $\boldsymbol{L}_{w_{SO(3)}}$ is:

$$(\boldsymbol{L}_{w_{SO(3)}})_{i,i_1} = \begin{cases} \sum_{j\in V_i} w_{ij}^{SO(3)} & i = i_1 \\ -w_{ii_1}^{SO(3)} & (i, i_1) \in \mathcal{E} \\ -w_{i_1 i}^{SO(3)} & (i_1, i) \in \mathcal{E} \\ 0 & \text{else,} \end{cases} \tag{B.50}$$

where $w_{ij}^{SO(3)} = \frac{\omega_{ij}}{3}$, thus $\boldsymbol{L}_{w_{SO(3)}}$ is a weighted Laplacian matrix.

The whole FIM

Combine the sub-matrices (B.33), (B.40), and (B.49), we can get the complete FIM (3.19). So Theorem 2 is proved.

B.1.3 The derivation of $\mathbb{E}\{s_{ij}s_{ij}\}$

This appendix is used to get the covariance $\mathbb{E}\{s_{ij}s_{ij}\}$ of the isotropic Langevin distribution in the 2D case, which is used in the sub-matrix (B.28). Because of $\text{skew}(\boldsymbol{Z}_{ij}) = s_{ij}\boldsymbol{E}$ and $s_{ij} = \frac{1}{2}\langle \text{skew}(\boldsymbol{Z}_{ij}), \boldsymbol{E}\rangle_{\boldsymbol{R}}$, we have:

$$\text{skew}(\boldsymbol{Z}_{ij}) = \frac{1}{2}\langle \text{skew}(\boldsymbol{Z}_{ij}), \boldsymbol{E}\rangle_{\boldsymbol{R}}\boldsymbol{E}. \tag{B.51}$$

Combine above equations, we have:

$$\begin{aligned} \mathbb{E}\{\|\text{skew}(\boldsymbol{Z}_{ij})\|_F^2\} =& \mathbb{E}\{\|\frac{1}{2}\langle \text{skew}(\boldsymbol{Z}_{ij}), \boldsymbol{E}\rangle_{\boldsymbol{R}}\boldsymbol{E}\|_F^2\} \\ =& \mathbb{E}\{\frac{1}{4}\cdot 2\left(\langle \text{skew}(\boldsymbol{Z}_{ij}), \boldsymbol{E}\rangle_{\boldsymbol{R}}^{\top}\cdot\langle \text{skew}(\boldsymbol{Z}_{ij}), \boldsymbol{E}\rangle_{\boldsymbol{R}}\right)\} \\ =& \mathbb{E}\{2s_{ij}s_{ij}\}. \end{aligned} \tag{B.52}$$

So we have: $\mathbb{E}\{{s_{ij}}^2\} = \frac{1}{2}\mathbb{E}\{\|\text{skew}(\boldsymbol{Z}_{ij})\|_F^2\}$. Based on [10], the PDF of the isotropic Langevin distribution on $SO(n)$ satisfies the following property, which is

translated by the extended bi-invariance property:

$$\forall\ \boldsymbol{U}_1, \boldsymbol{U}_2 \in \bar{O}(n)\ s.t.\ \det(\boldsymbol{U}_1\boldsymbol{U}_2) = 1, \int_{SO(n)} \widehat{f_{ij}}(\boldsymbol{Z}) d\mu(\boldsymbol{Z}) = \int_{SO(n)} \widehat{f_{ij}}(\boldsymbol{Z}_{ij}) d\mu(\boldsymbol{Z}_{ij}). \tag{B.53}$$

where $\boldsymbol{Z} = \boldsymbol{U}_1\boldsymbol{Z}_{ij}\boldsymbol{U}_2$. So we can use the Weyl's integration formula specialized to $SO(n), n = 2, 3$ [16]. For 2D case, let $\boldsymbol{U}_1\boldsymbol{Z}_{ij}\boldsymbol{U}_2 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$, we have:

$$\begin{aligned}
\mathbb{E}\{{s_{ij}}^2\} =& \frac{1}{2}\mathbb{E}\{\|\text{skew}(\boldsymbol{Z}_{ij})\|_F^2\} \\
=& \frac{1}{8}\int_{SO(2)} \|\boldsymbol{Z} - \boldsymbol{Z}^\top\|_F^2 \widehat{f_{ij}}(\boldsymbol{Z}) d\mu(\boldsymbol{Z}) \\
=& \frac{1}{8}\frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{8\sin^2(\theta)}{c_2(\kappa_{ij})} \exp(2\kappa_{ij}\cos(\theta)) d\theta \\
=& \frac{\int_{-\pi}^{\pi}(1-\cos^2(\theta))\exp(2\kappa_{ij}\cos(\theta))d\theta}{2\pi I_0(2\kappa_{ij})}. \\
& \cos(0\theta) = 1,\ \ \cos(2\theta) = 2\cos^2(\theta) - 1 \Downarrow \\
=& \frac{1}{I_0(2\kappa_{ij})}\frac{1}{2\pi}\int_{-\pi}^{\pi}(\frac{cos(0\theta)}{2} - \frac{\cos(2\theta)}{2})\exp(2\kappa_{ij}\cos(\theta))d\theta.
\end{aligned} \tag{B.54}$$

Based on $I_1(2\kappa_{ij}) = \kappa_{ij}(I_0(2\kappa_{ij}) - I_2(2\kappa_{ij}))$ and the PDF (3.4), we have:

$$\mathbb{E}\{{s_{ij}}^2\} = \frac{1}{I_0(2\kappa_{ij})}\left(\frac{I_0(2\kappa_{ij})}{2} - \frac{I_2(2\kappa_{ij})}{2}\right) = \frac{I_1(2\kappa_{ij})}{2\kappa_{ij}I_0(2\kappa_{ij})}. \tag{B.55}$$

B.1.4 Proof of Lemma 6

In this appendix, we will prove Lemma 6, which is used in the sub-matrix (B.44). The following lemmas will be used in the derivation process.

Lemma 7. *[119] $\boldsymbol{X}$ is a random matrix as well as $\boldsymbol{A}$ and $\boldsymbol{B}$ are two constant matrices, we have $\mathbb{E}\{\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}\} = \boldsymbol{A}\mathbb{E}\{\boldsymbol{X}\}\boldsymbol{B}$.*

Lemma 8. *[119] $\boldsymbol{X}$ and $\boldsymbol{Y}$ are two random matrices, we have $\mathbb{E}\{\boldsymbol{X} \pm \boldsymbol{Y}\} = \mathbb{E}\{\boldsymbol{X}\} \pm \mathbb{E}\{\boldsymbol{Y}\}$.*

Lemma 9. *[10] There exists a permutation matrix $\boldsymbol{P}_{kl} \in \bar{O}(n)$ such that $\boldsymbol{P}_{kl}^\top \boldsymbol{E}_k \boldsymbol{P}_{kl} = \boldsymbol{E}_l$ and $\boldsymbol{P}_{kl}^\top \boldsymbol{E}_l \boldsymbol{P}_{kl} = -\boldsymbol{E}_k$, where $\boldsymbol{E}_k$ and $\boldsymbol{E}_l$ are defined in the definition* (3.18).

1) Based on Lemmas 7 and 8, it is easy to know that $\mathbb{E}\{\Phi_k^{ij}\} = \mathbb{E}\{\langle \boldsymbol{R}_i \boldsymbol{R}_j^\top \text{skew}(\boldsymbol{Z}_{ij}^\top) \boldsymbol{R}_j \boldsymbol{R}_i^\top, \boldsymbol{R}_i \boldsymbol{E}_k \boldsymbol{R}_i^\top \rangle_{\boldsymbol{R}}\} = \langle \boldsymbol{R}_i \boldsymbol{R}_j^\top \text{skew}(\boldsymbol{I}_{2\times 2}) \boldsymbol{R}_j \boldsymbol{R}_i^\top, \boldsymbol{R}_i \boldsymbol{E}_k \boldsymbol{R}_i^\top \rangle_{\boldsymbol{R}} = 0$, so we have $\mathbb{E}\{\Phi_k^{ij}\} = \mathbb{E}\{\Phi_l^{ij}\} = 0$.

2) Based on Lemma 9 and $\boldsymbol{H}_{ij} = \boldsymbol{H}_{ji}^\top$, we have:

$$
\begin{aligned}
\mathbb{E}\{\Phi_k^{ij}\Phi_l^{ij}\} =& \mathbb{E}\{\langle G_{ij}(\boldsymbol{Z}_{ij}), \boldsymbol{R}_i \boldsymbol{E}_k \boldsymbol{R}_i^\top \rangle_{\boldsymbol{R}} \langle G_{ij}(\boldsymbol{Z}_{ij}), \boldsymbol{R}_i \boldsymbol{E}_l \boldsymbol{R}_i^\top \rangle_{\boldsymbol{R}}\} \\
=& \mathbb{E}\{-\langle G_{ij}(\boldsymbol{Z}_{ij}), \boldsymbol{R}_i \boldsymbol{P}_{kl}^\top \boldsymbol{E}_l \boldsymbol{P}_{kl} \boldsymbol{R}_i^\top \rangle_{\boldsymbol{R}} \\
& \langle G_{ij}(\boldsymbol{Z}_{ij}), \boldsymbol{R}_i \boldsymbol{P}_{kl}^\top \boldsymbol{E}_k \boldsymbol{P}_{kl} \boldsymbol{R}_i^\top \rangle_{\boldsymbol{R}}\} \\
=& \mathbb{E}\{-\langle G_{ij}(\boldsymbol{Z}_{ij}), \boldsymbol{R}_i \boldsymbol{P}_{kl}^\top \boldsymbol{R}_i^\top \boldsymbol{R}_i \boldsymbol{E}_l \boldsymbol{R}_i^\top \boldsymbol{R}_i \boldsymbol{P}_{kl} \boldsymbol{R}_i^\top \rangle_{\boldsymbol{R}} \\
& \langle G_{ij}(\boldsymbol{Z}_{ij}), \boldsymbol{R}_i \boldsymbol{P}_{kl}^\top \boldsymbol{R}_i^\top \boldsymbol{R}_i \boldsymbol{E}_k \boldsymbol{R}_i \boldsymbol{R}_i^\top \boldsymbol{P}_{kl} \boldsymbol{R}_i^\top \rangle_{\boldsymbol{R}}\} \\
=& \int_{SO(3)} -\langle G_{ij}(\boldsymbol{R}_j \boldsymbol{P}_{kl} \boldsymbol{R}_j^\top \boldsymbol{Z}_{ij}^\top \boldsymbol{R}_j \boldsymbol{P}_{kl}^\top \boldsymbol{R}_j^\top), \boldsymbol{R}_i \boldsymbol{E}_l \boldsymbol{R}_i^\top \rangle_{\boldsymbol{R}} \\
& \langle G_{ij}(\boldsymbol{R}_j \boldsymbol{P}_{kl} \boldsymbol{R}_j^\top \boldsymbol{Z}_{ij}^\top \boldsymbol{R}_j \boldsymbol{P}_{kl}^\top \boldsymbol{R}_j^\top), \boldsymbol{R}_i \boldsymbol{E}_k \boldsymbol{R}_i^\top \rangle_{\boldsymbol{R}} \widehat{f_{ij}}(\boldsymbol{Z}_{ij}) d\mu(\boldsymbol{Z}_{ij}).
\end{aligned}
\tag{B.56}
$$

Because the isotropic Lagevin distribution follows the bi-invariance property, satisfying $\mu(\boldsymbol{R}_j \boldsymbol{P}_{kl} \boldsymbol{R}_j^\top \boldsymbol{Z}_{ij}^\top \boldsymbol{R}_j \boldsymbol{P}_{kl}^\top \boldsymbol{R}_j^\top) = \mu(\boldsymbol{Z}_{ij})$, and $\widehat{f_{ij}}$ is a spectral function, meaning $\widehat{f_{ij}}(\boldsymbol{R}_j \boldsymbol{P}_{kl} \boldsymbol{R}_j^\top \boldsymbol{Z}_{ij}^\top \boldsymbol{R}_j \boldsymbol{P}_{kl}^\top \boldsymbol{R}_j^\top) = \widehat{f_{ij}}(\boldsymbol{Z}_{ij}^\top) = \widehat{f_{ij}}(\boldsymbol{Z}_{ij})$, we have: $\mathbb{E}\{\Phi_k^{ij}\Phi_l^{ij}\} = -\mathbb{E}\{\Phi_l^{ij}\Phi_k^{ij}\} = 0$, which means that Φ_l^{ij} and Φ_k^{ij} are independent. Similarly, we can get: $\mathbb{E}\{{\Phi_l^{ij}}^2\} = \mathbb{E}\{{\Phi_k^{ij}}^2\}$. We can also get:

Lemma 10. *If $\boldsymbol{R}_i$, $\boldsymbol{R}_{i_1}$, and $\boldsymbol{R}_{i_2} \in SO(n)$ and $\boldsymbol{Z}_{ij} \sim Lang(\boldsymbol{I}_{n\times n}, \kappa_{ij})$, we have: (a) $\mathbb{E}\{\bar{\Phi}\} = \mathbb{E}\{\hat{\Phi}\} = 0$ and (b) $\mathbb{E}\{\bar{\Phi}^2\} = \mathbb{E}\{\hat{\Phi}^2\}$, where $\bar{\Phi} = \langle skew(\boldsymbol{Z}_{ij}^\top), \boldsymbol{E}_k \rangle_{\boldsymbol{R}}$, $\hat{\Phi} = \langle \boldsymbol{R}_i skew(\boldsymbol{R}_{i_1} \boldsymbol{Z}_{ij}^\top \boldsymbol{R}_{i_1}^\top) \boldsymbol{R}_i^\top, \boldsymbol{R}_{i_2} \boldsymbol{E}_k \boldsymbol{R}_{i_2}^\top \rangle_{\boldsymbol{R}}$.*

Proof. a. $\mathbb{E}\{\hat{\Phi}\} = \mathbb{E}\{\langle \boldsymbol{R}_i \text{skew}(\boldsymbol{R}_{i_1} \boldsymbol{Z}_{ij}^\top \boldsymbol{R}_{i_1}^\top) \boldsymbol{R}_i^\top, \boldsymbol{R}_{i_2} \boldsymbol{E}_k \boldsymbol{R}_{i_2}^\top \rangle_{\boldsymbol{R}}\} = \mathbb{E}\{\langle \boldsymbol{R}_i \boldsymbol{R}_{i_1} \text{skew}(\boldsymbol{Z}_{ij}^\top) \boldsymbol{R}_{i_1}^\top \boldsymbol{R}_i^\top, \boldsymbol{R}_{i_2} \boldsymbol{E}_k \boldsymbol{R}_{i_2}^\top \rangle_{\boldsymbol{R}}\} = \langle \boldsymbol{R}_i \boldsymbol{R}_{i_1} \text{skew}(\boldsymbol{I}_{2\times 2}^\top) \boldsymbol{R}_{i_1}^\top \boldsymbol{R}_i^\top, \boldsymbol{R}_{i_2} \boldsymbol{E}_k \boldsymbol{R}_{i_2}^\top \rangle_{\boldsymbol{R}} = 0$ and $\mathbb{E}\{\bar{\Phi}\} =$

$\mathbb{E}\{\langle \text{skew}(\boldsymbol{Z}_{ij}^{\top}), \boldsymbol{E}_k\rangle_{\boldsymbol{R}}\} = \langle \text{skew}(\boldsymbol{I}_{2\times 2}^{\top}), \boldsymbol{E}_k\rangle_{\boldsymbol{R}} = 0.$

b. $\mathbb{E}\{\hat{\Phi}^2\} = \mathbb{E}\{\langle \boldsymbol{R}_i \text{skew}(\boldsymbol{R}_{i_1}\boldsymbol{Z}_{ij}^{\top}\boldsymbol{R}_{i_1}^{\top})\boldsymbol{R}_i^{\top}, \boldsymbol{R}_{i_2}\boldsymbol{E}_k\boldsymbol{R}_{i_2}^{\top}\rangle_{\boldsymbol{R}}^2\} = \mathbb{E}\{\langle \boldsymbol{R}_{i_2}\boldsymbol{R}_i \text{skew}(\boldsymbol{R}_{i_1}\boldsymbol{Z}_{ij}^{\top}\boldsymbol{R}_{i_1}^{\top})\boldsymbol{R}_i^{\top}\boldsymbol{R}_{i_2}^{\top}, \boldsymbol{E}_k\rangle_{\boldsymbol{R}}^2\} = \mathbb{E}\{\langle \text{skew}(\boldsymbol{R}_{i_2}\boldsymbol{R}_i\boldsymbol{R}_{i_1}\boldsymbol{Z}_{ij}^{\top}\boldsymbol{R}_{i_1}^{\top}\boldsymbol{R}_i^{\top}\boldsymbol{R}_{i_2}^{\top}), \boldsymbol{E}_k\rangle_{\boldsymbol{R}}^2\}$. Applying the bi-invariance property ($\mu(\boldsymbol{R}_{i_2}\boldsymbol{R}_i\boldsymbol{R}_{i_1}\boldsymbol{Z}_{ij}^{\top}\boldsymbol{R}_{i_1}^{\top}\boldsymbol{R}_i^{\top}\boldsymbol{R}_{i_2}^{\top}) = \mu(\boldsymbol{Z}_{ij}^{\top})$ and $\widehat{f_{ij}}(\boldsymbol{R}_{i_2}\boldsymbol{R}_i\boldsymbol{R}_{i_1}\boldsymbol{Z}_{ij}^{\top}\boldsymbol{R}_{i_1}^{\top}\boldsymbol{R}_i^{\top}\boldsymbol{R}_{i_2}^{\top}) = \widehat{f_{ij}}(\boldsymbol{Z}_{ij}^{\top})$), we have: $\mathbb{E}\{\bar{\Phi}^2\} = \mathbb{E}\{\hat{\Phi}^2\}$. □

3) Based on $\mathbb{E}\{{\Phi_l^{ij}}^2\} = \mathbb{E}\{{\Phi_k^{ij}}^2\}$, based on Lemma 10, we have:

$$\begin{aligned}\mathbb{E}\{\|\text{skew}(\boldsymbol{Z}_{ij})\|_F^2\} &= \mathbb{E}\{\sum_{k=1}^{3}\langle \text{skew}(\boldsymbol{Z}_{ij}), \boldsymbol{R}_i\boldsymbol{E}_k\boldsymbol{R}_i^{\top}\rangle_{\boldsymbol{R}}^2\} \\ &= \mathbb{E}\{\sum_{k=1}^{3}\langle G_{ij}(\boldsymbol{Z}_{ij}), \boldsymbol{R}_i\boldsymbol{E}_k\boldsymbol{R}_i^{\top}\rangle_{\boldsymbol{R}}^2\} = \mathbb{E}\{3{\Phi_k^{ij}}^2\}.\end{aligned} \tag{B.57}$$

So we have: $\mathbb{E}\{{\Phi_k^{ij}}^2\} = \mathbb{E}\{{\Phi_l^{ij}}^2\} = \frac{1}{3}\mathbb{E}\{\|\text{skew}(\boldsymbol{Z}_{ij})\|_F^2\}$.

B.1.5 The computation of the weight ω_{ij}

This appendix is presented to compute the weight ω_{ij} shown in the sub-matrix (B.44). Based on the extended bi-invariance property (B.53), for 3D case, let $\boldsymbol{Z} = \boldsymbol{U}_1\boldsymbol{Z}_{ij}\boldsymbol{U}_2 =$

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$, we have:

$$\begin{aligned}
\omega_{ij} =& \kappa_{ij}^2 \mathbb{E}\{\|\text{skew}(\boldsymbol{Z}_{ij})\|_F^2\} \\
& \text{Use property (B.53) and definition of } \mathbb{E}\{\} \Downarrow \\
=& \frac{\kappa_{ij}^2}{4} \int_{SO(3)} \|\boldsymbol{Z} - \boldsymbol{Z}^T\|_F^2 \widehat{f}_{ij}(\boldsymbol{Z}) d\mu(\boldsymbol{Z}) \\
& \text{Consider integral on } \theta \Downarrow \\
=& \frac{\kappa_{ij}^2}{4} \frac{1}{2\pi} \int_{-\pi}^{\pi} 8 \sin^2(\theta) \widehat{f}_{ij}(\boldsymbol{Z}) (1 - \cos(\theta)) d\theta \\
& \text{Introduce the noise model (3.4)} \Downarrow \\
=& \frac{2\kappa_{ij}^2 \int_{-\pi}^{\pi} \sin^2(\theta)(1 - \cos(\theta)) \exp(2\kappa_{ij} \cos(\theta)) d\theta}{2\pi(I_0(2\kappa_{ij}) - I_1(2\kappa_{ij}))} \\
& \sin^2(\theta) = 1 - \cos^2(\theta) \Downarrow \\
=& \frac{2\kappa_{ij}^2}{I_0(2\kappa_{ij}) - I_1(2\kappa_{ij})} \frac{1}{2\pi} \int_{-\pi}^{\pi} \rho(\theta) \exp(2\kappa_{ij} \cos(\theta)) d\theta,
\end{aligned} \tag{B.58}$$

where $\rho(\theta) = (1 - \cos(\theta) - \cos^2(\theta) + \cos^3(\theta))$.

We know that:

$$\cos(1\theta) = \cos(\theta), \;\; \cos(3\theta) = 4\cos^3(\theta) - 3\cos(\theta). \tag{B.59}$$

Based on the PDF (3.4), introduce the trigonometric functions of (B.54) and (B.59) into $\rho(\theta)$, we have:

$$\begin{aligned}
\omega_{ij} &= \frac{2\kappa_{ij}^2 \left(\frac{I_0(2\kappa_{ij})}{2} - \frac{I_1(2\kappa_{ij})}{4} - \frac{I_2(2\kappa_{ij})}{2} + \frac{I_3(2\kappa_{ij})}{4} \right)}{I_0(2\kappa_{ij}) - I_1(2\kappa_{ij})} \\
&= \frac{\kappa_{ij}^2 \left(2I_0(2\kappa_{ij}) - I_1(2\kappa_{ij}) - 2I_2(2\kappa_{ij}) + I_3(2\kappa_{ij})\right)}{2I_0(2\kappa_{ij}) - 2I_1(2\kappa_{ij})}.
\end{aligned} \tag{B.60}$$

B.1.6 Proof of the eigenvalues of $\boldsymbol{\Psi}_i$

This appendix is used to obtain the eigenvalues of the matrix $\boldsymbol{\Psi}_i$, and then the eigenvalues can be used in the computational of trace function $\sum_{i=1}^{n_p} \text{trace}(\boldsymbol{\Psi}_i)$ and the proof of Corollary 1. Based on the definition (B.49), $\boldsymbol{\Psi}_i$ can be written as:

$$\begin{aligned} \boldsymbol{\Psi}_i &= \sum_{j \in V_i^+} \boldsymbol{\Psi}_{(i,j)}, \ \boldsymbol{\Psi}_{(i,j)} = \begin{bmatrix} \psi_{(i,j)}^{11} & \psi_{(i,j)}^{12} & \psi_{(i,j)}^{13} \\ \psi_{(i,j)}^{12} & \psi_{(i,j)}^{22} & \psi_{(i,j)}^{21} \\ \psi_{(i,j)}^{13} & \psi_{(i,j)}^{21} & \psi_{(i,j)}^{33} \end{bmatrix}, \\ \psi_{(i,j)}^{kl} &= \delta_{ij}^{-2} (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{I}_{3\times 3}^{k,l} \boldsymbol{R}_i^\top (\boldsymbol{x}_i - \boldsymbol{x}_j). \end{aligned} \tag{B.61}$$

Let $\frac{1}{\sqrt{2}} \boldsymbol{R}_i^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) = (p_1, \ p_2, \ p_3)^\top$, we can get:

$$\boldsymbol{\Psi}_{(i,j)} = \delta_{ij}^{-2} \begin{bmatrix} p_2^2 + p_3^2 & -p_1 p_2 & -p_1 p_3 \\ -p_1 p_2 & p_1^2 + p_3^2 & -p_2 p_3 \\ -p_1 p_3 & -p_2 p_3 & p_1^2 + p_2^2 \end{bmatrix}. \tag{B.62}$$

For a 3×3 matrix $\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$, $\det(\lambda \boldsymbol{I}_{3\times 3} - \boldsymbol{B}) = 0$ can be written as:

$$\begin{aligned} &\lambda^3 + k_1 \lambda^2 + k_2 \lambda + k_3 = 0, \\ &k_1 = -b_{11} - b_{22} - b_{33}, \\ &k_2 = b_{11} b_{22} + b_{11} b_{33} + b_{22} b_{33} - b_{12}^2 - b_{23}^2 - b_{33}^2, \\ &k_3 = -\det(\boldsymbol{B}). \end{aligned} \tag{B.63}$$

We can easily find that $\det(\boldsymbol{\Psi}_{(i,j)}) = 0$. So $k_3 = -\det(\boldsymbol{\Psi}_{(i,j)}) = -\det(\boldsymbol{B}) = 0$.

Substitute the matrix (B.62) into the equations (B.63), we have:

$$\begin{aligned} &\lambda^2 + k_1\lambda + k_2 = 0, \\ &k_1 = -2(p_1^2 + p_2^2 + p_3^2), \Rightarrow \left(\lambda - (p_1^2 + p_2^2 + p_3^2)\right)^2 = 0. \\ &k_2 = (p_1^2 + p_2^2 + p_3^2)^2, \end{aligned} \tag{B.64}$$

Based on $p_1^2 + p_2^2 + p_3^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} \boldsymbol{R}_i^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) = \frac{1}{2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$, we can known that the eigenvalues of $\boldsymbol{\Psi}_{(i,j)}$ are $\lambda_1^{ij} = \lambda_2^{ij} = \frac{1}{2}\delta_{ij}^{-2}\|\boldsymbol{x}_j - \boldsymbol{x}_i\|_2^2$ and $\lambda_3^{ij} = 0$.

B.2 Proofs in Chapter 4

B.2.1 Proof of Theorem 7

In this appendix, we prove Theorem 7 based on the following Cauchy Interlacing Theorem.

Theorem 12. *(Cauchy Interlacing Theorem, [52], p. 411) Let $\boldsymbol{A}$ be an $n \times n$ Hermitian matrix and the eigenvalues of $\boldsymbol{A}$ be denoted as $\lambda_1(\boldsymbol{A}), \lambda_2(\boldsymbol{A}), \cdots, \lambda_n(\boldsymbol{A})$, where it is assumed that $\lambda_1(\boldsymbol{A}) \leq \lambda_2(\boldsymbol{A}) \leq \cdots \leq \lambda_n(\boldsymbol{A})$. Let $\boldsymbol{A}_i = \boldsymbol{A}(\{1, ..., n\} \backslash \{i\})$ be the sub-matrix of $\boldsymbol{A}$ by removing the i-th column and row for some $i \in \{1, \cdots, n\}$, then:*

$$\lambda_1(\boldsymbol{A}) \leq \lambda_1(\boldsymbol{A}_i) \leq \lambda_2(\boldsymbol{A}) \leq \cdots \leq \lambda_{n-1}(\boldsymbol{A}) \leq \lambda_{n-1}(\boldsymbol{A}_i) \leq \lambda_n(\boldsymbol{A}). \tag{B.65}$$

a) Non-normalized It is easy to find that $f_{obj}(\emptyset) \neq 0$, so the objective function is a non-normalized function.

b) Non-monotone We know that the eigenvalues of the reduced weighted Laplacian matrix $\lambda_i(\boldsymbol{\mathcal{L}}(\mathcal{V} \backslash \mathcal{N}_a))$, $i = 1, 2, \cdots$ are possible to be smaller than 1, which means their log function $\log(\lambda_i(\boldsymbol{\mathcal{L}}(\mathcal{V} \backslash \mathcal{N}_a)))$ are possible to be smaller than 0. For

the objective function, we also have: $\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a))) = \sum_i \log(\lambda_i(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a)))$. With the growth of $\mathcal{N}_a$, the deleting of the row and column, caused by the anchored poses, may delete the eigenvalue of which value is bigger than 1 or smaller than 1. As an example, we assume the minimal eigenvalue $\lambda_1(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a))$ to be smaller than 1 ($\log(\lambda_1(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a))) < 0$). Then, after anchoring a pose, based on the Cauchy Interlacing Theorem, the minimal eigenvalue will increase and we assume that the new minimal eigenvalue $\lambda_1(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a\backslash\{i'\}))$ to be larger than 1 ($\log(\lambda_1(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a\backslash\{i'\}))) > 0$). In this case, we may get $\sum_i \log(\lambda_i(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a))) < \sum_i \log(\lambda_i(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a\backslash\{i'\})))$. After this iteration, all eigenvalues will be larger than 1. Based on the Cauchy Interlacing Theorem, we have: $\sum_i \log(\lambda_i(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a\backslash\{i'\}))) > \sum_i \log(\lambda_i(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a\backslash\{i', j'\})))$. So, in short, this objective function is non-monotone. The following is a numerical example[†]:

The incident matrix of this graph is:

$$\boldsymbol{A} = \begin{bmatrix} -1 & 0 & 0 & 0 & -1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{B.66}$$

[†]As an example, we only consider the sub-matrix corresponding to the translation part. It is easy to extend the conclusion to the whole objective function $\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a)))$.

The diagonal weighted matrix is:

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1.7 & 0 & 0 & 0 & 0 \\ 0 & 1.8 & 0 & 0 & 0 \\ 0 & 0 & 1.4 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 1.86 \end{bmatrix} \tag{B.67}$$

So the weighted Laplacian matrix is:

$$\boldsymbol{L}_{w_{\mathbb{R}}} = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\top} = \begin{bmatrix} 3.56 & -1.7 & 0 & -1.86 & 0 \\ -1.7 & 3.5 & -1.8 & 0 & 0 \\ 0 & -1.8 & 3.2 & -1.4 & 0 \\ -1.86 & 0 & -1.4 & 4.76 & -1.5 \\ 0 & 0 & 0 & -1.5 & 1.5 \end{bmatrix} \tag{B.68}$$

When only one anchor (first row and column) is deleted, we have:

$$\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(2:5, 2:5))) = 3.3546; \tag{B.69}$$

When the first and last poses are deleted, we have:

$$\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(2:4, 2:4))) = 3.4349; \tag{B.70}$$

When the first, second, and last poses are deleted, we have:

$$\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(3:4, 3:4))) = 2.5857; \tag{B.71}$$

From this numerical example, we can see that the objective function built based

on the weighted Laplacian matrix is not a monotone function.

c) Non-negative As shown in footnote 4, without loss of generality, we assume that the weights of the edges in the pose graph are all bigger than 1. For any $\mathcal{N}_a$, $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\mathcal{N}_a)$ and $\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\mathcal{N}_a)$ can be respectively written as:

$$\begin{aligned}\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\mathcal{N}_a) &= \boldsymbol{L}_{w_{\mathbb{R}}}^{\mathcal{V}\backslash\mathcal{N}_a} + \widehat{\boldsymbol{L}}_{w_{\mathbb{R}}}(\mathcal{N}_a),\\ \boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\mathcal{N}_a) &= \boldsymbol{L}_{w_{SO(n)}}^{\mathcal{V}\backslash\mathcal{N}_a} + \widehat{\boldsymbol{L}}_{w_{SO(n)}}(\mathcal{N}_a),\end{aligned} \tag{B.72}$$

where $\boldsymbol{L}_{w_{\mathbb{R}}}^{\mathcal{V}\backslash\mathcal{N}_a}$ and $\boldsymbol{L}_{w_{SO(n)}}^{\mathcal{V}\backslash\mathcal{N}_a}$ are the reduced weighted Laplacian matrices with only one anchor corresponding to the graphs $\mathcal{G}_{w_{\mathbb{R}}}^{\mathcal{V}\backslash\mathcal{N}_a}$ and $\mathcal{G}_{w_{SO(n)}}^{\mathcal{V}\backslash\mathcal{N}_a}$ with $|\mathcal{V}\backslash\mathcal{N}_a|$ un-anchored nodes and 1 anchored node; $\widehat{\boldsymbol{L}}_{w_{\mathbb{R}}}(\mathcal{N}_a)$ and $\widehat{\boldsymbol{L}}_{w_{SO(n)}}(\mathcal{N}_a)$ are two diagonal positive semidefinite $|\mathcal{V}\backslash\mathcal{N}_a| \times |\mathcal{V}\backslash\mathcal{N}_a|$ matrices, whose non-zero diagonal elements are corresponding to the nodes connected with the deleted nodes $\mathcal{N}_a$. They satisfy: $\boldsymbol{L}_{w_{\mathbb{R}}}^{\mathcal{V}\backslash\mathcal{N}_a} \succeq \boldsymbol{0}$, $\widehat{\boldsymbol{L}}_{w_{\mathbb{R}}}(\mathcal{N}_a) \succeq \boldsymbol{0}$, $\boldsymbol{L}_{w_{SO(n)}}^{\mathcal{V}\backslash\mathcal{N}_a} \succeq \boldsymbol{0}$, and $\widehat{\boldsymbol{L}}_{w_{SO(n)}}(\mathcal{N}_a) \succeq \boldsymbol{0}$.

Lemma 11. *[?] For any two matrices $\boldsymbol{N} \succeq \boldsymbol{0}$ and $\boldsymbol{M} \succeq \boldsymbol{0}$, we have:*

$$\det(\boldsymbol{M}+\boldsymbol{N}) \geq \det(\boldsymbol{M}) \tag{B.73}$$

Based on Lemma 11, we have:

$$\begin{aligned}\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\mathcal{N}_a))) &\geq \log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}^{\mathcal{V}\backslash\mathcal{N}_a})),\\ \log(\det(\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\mathcal{N}_a))) &\geq \log(\det(\boldsymbol{L}_{w_{SO(n)}}^{\mathcal{V}\backslash\mathcal{N}_a})).\end{aligned} \tag{B.74}$$

Based on Theorem 1, the determinant of the reduced weighted Laplacian matrix with one anchor is equal to the tree-connectivity of its corresponding graph, so we

have:

$$\begin{aligned}\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}^{\mathcal{V}\backslash\mathcal{N}_a})) &= \log(\sum_{\mathcal{T}_{\mathbb{R}}\in\mathcal{T}_{\mathcal{G}_{w_{\mathbb{R}}}^{\mathcal{V}\backslash\mathcal{N}_a}}}\prod_{e_1\in\mathcal{E}(\mathcal{T}_{\mathbb{R}})} w(e_1)),\\ \log(\det(\boldsymbol{L}_{w_{SO(n)}}^{\mathcal{V}\backslash\mathcal{N}_a})) &= \log(\sum_{\mathcal{T}_{SO(n)}\in\mathcal{T}_{\mathcal{G}_{SO(n)}^{\mathcal{V}\backslash\mathcal{N}_a}}}\prod_{e_2\in\mathcal{E}(\mathcal{T}_{SO(n)})} w(e_2)).\end{aligned} \tag{B.75}$$

where $\mathcal{T}_{\mathcal{G}_{w_{\mathbb{R}}}^{\mathcal{V}\backslash\mathcal{N}_a}}$ and $\mathcal{T}_{\mathcal{G}_{SO(n)}^{\mathcal{V}\backslash\mathcal{N}_a}}$ are respectively the sets of all spanning trees of $\mathcal{G}_{w_{\mathbb{R}}}^{\mathcal{V}\backslash\mathcal{N}_a}$ and $\mathcal{G}_{SO(n)}^{\mathcal{V}\backslash\mathcal{N}_a}$, $\mathcal{T}_{\mathbb{R}}$ and $\mathcal{T}_{SO(n)}$ are two spanning trees, $\mathcal{E}(\mathcal{T}_{\mathbb{R}})$ and $\mathcal{E}(\mathcal{T}_{SO(n)})$ respectively represent the sets of edges in $\mathcal{T}_{\mathbb{R}}$ and $\mathcal{T}_{SO(n)}$.

Because of $w(e_1) >> 1$ and $w(e_2) >> 1$, we have:

$$\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}^{\mathcal{V}\backslash\mathcal{N}_a})) \geq \log(\prod_{e_1\in\mathcal{E}(\mathcal{T}_{\mathbb{R}})} w(e_1)) = \sum_{e_1\in\mathcal{E}(\mathcal{T}_{\mathbb{R}})}\log(w(e_1)) >> \sum_{e_1\in\mathcal{E}(\mathcal{T}_{\mathbb{R}})}\log(1) = 0. \tag{B.76}$$

Similarly, $\log(\det(\boldsymbol{L}_{w_{SO(n)}}^{\mathcal{V}\backslash\mathcal{N}_a})) >> 0$, then we have:

$$\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a))) = n\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\mathcal{N}_a))) + d\log(\det(\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\mathcal{N}_a))) >> 0 \tag{B.77}$$

Hence, this objective function is non-negative.

d) Submodular Because the matrix $\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a)$ is a Hermitian positive-definite matrix, we can get its classical Cholesky decomposition by $\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a) = \boldsymbol{V}^{\top}\boldsymbol{V}$ and denote the columns of $\boldsymbol{V}$ corresponding to i-th pose by v_i. Then, based on the geometrical interpretation of the determinant, we know that the determinant is the volume of the parallelepiped in any dimensions, so we have:

$$\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a)) = vol^2(\{v_1, \cdots, v_{n_p-N}\}), \tag{B.78}$$

where $vol^2(\bullet)$ denotes the squared volume of the parallelepiped spanned by the vectors $\bullet$ [133].

We now add an element $n_p - N + 2 \in \mathcal{N}_a \backslash \{n_p - N + 1\}$ to the sets $\mathcal{V} \backslash \mathcal{N}_a = \{1, \cdots, n_p - N\}$ and $\mathcal{V} \backslash \mathcal{N}_a \cup \{n_p - N + 1\}$, which means the rows and columns corresponding to the $n_p - N + 2$-th pose are added to the reduced weighted Laplacian matrix. For 2D case, it is noted that an element v_{n_p-N+2} includes three columns: two belong to $\boldsymbol{L}_w^{\mathbb{R}^n}(\mathcal{V})$ and the other one belongs to $\boldsymbol{L}_w^{SO(n)}(\mathcal{V})$. In the following proof, we will only consider one column of them first. Because of the full rank of the reduced weighted Laplacian matrix, we have:

$$v_{n_p-N+2} = v^{\perp}_{n_p-N+2} + v^{\parallel}_{n_p-N+2}, \tag{B.79}$$

where $v^{\perp}_{n_p-N+2} \perp span\{v_1, \cdots, v_{n_p-N+1}\}$ and $v^{\parallel}_{n_p-N+2} \parallel span\{v_1, \cdots, v_{n_p-N+1}\}$, $v^{\perp}_{n_p-N+2} \neq \mathbf{0}$.

Since $vol^2(\{v_1, \cdots, v_{n_p-N+1}, v^{\parallel}_{n_p-N+2}\}) = 0$ (the determinant function of the square matrix with non-full rank is equal to 0); $vol^2(v_{\mathcal{V}\backslash\mathcal{N}_a} \cup \{\frac{v^{\parallel}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}\}) > 0$, $v_{\mathcal{V}\backslash\mathcal{N}_a} = \{v_1, \cdots, v_{n_p-N}\}$ and $\frac{v^{\perp}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}$ is a unit vector satisfying $\frac{v^{\perp}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|} \perp span\{v_1, \cdots, v_{n_p-N+1}\}$ and $\frac{v^{\perp}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|} \perp span\{v_1, \cdots, v_{n_p-N}\}$ (because of $v_{\mathcal{V}\backslash\mathcal{N}_a} = \{v_1, \cdots, v_{n_p-N}\} \in \{v_1, \cdots, v_{n_p-N+1}\}$), we have:

$$
\begin{aligned}
&\frac{\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a\cup\{n_p-N+1,n_p-N+2\}))}{\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a\cup\{n_p-N+2\}))}\\
&\quad\textbf{Based on (B.78)}\Downarrow\\
=&\frac{vol^2(\{v_1,\cdots,v_{n_p-N+2}\})}{vol^2(\{v_1,\cdots,v_{n_p-N},v_{n_p-N+2}\})}\\
&\quad\textbf{Based on (B.79) and divide } \|v^{\perp}_{n_p-N+2}\|\Downarrow\\
=&vol^2(\{v_1,\cdots,v_{n_p-N+1},\frac{v^{\perp}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}\})/(vol^2(v_{\mathcal{V}\backslash\mathcal{N}_a}\cup\\
&\{\frac{v^{\perp}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}\})+vol^2(v_{\mathcal{V}\backslash\mathcal{N}_a}\cup\{\frac{v^{\|}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}\}))\\
&\quad\frac{v^{\perp}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}\textbf{ is a unit vector }^{\ddagger}\Downarrow\\
=&\frac{vol^2(v_{\mathcal{V}\backslash\mathcal{N}_a}\cup\{v_{n_p-N+1}\})}{vol^2(v_{\mathcal{V}\backslash\mathcal{N}_a})+vol^2(v_{\mathcal{V}\backslash\mathcal{N}_a}\cup\{\frac{v^{\|}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}\})}\\
&\quad vol^2(v_{\mathcal{V}\backslash\mathcal{N}_a}\cup\{\frac{v^{\|}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}\})>0\Downarrow\\
\leq&\frac{vol^2(v_{\mathcal{V}\backslash\mathcal{N}_a}\cup\{v_{n_p-N+1}\})}{vol^2(v_{\mathcal{V}\backslash\mathcal{N}_a})}\\
=&\frac{\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a\cup\{n_p-N+1\}))}{\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a))}.
\end{aligned}
\tag{B.80}
$$

,

Because of the monotone increasing of the log function, we can get $\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a\cup\{n_p-N+1,n_p-N+2\})))-\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a\cup\{n_p-N+1\})))\leq\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a\cup$

---

‡For the elementary transformation, the determinant and its corresponding volume remains unchanged. By a finite sequence of the elementary transformation, we can transform the matrix to the diagonal form. In other words, we have $vol^2(\{v_1,\cdots,v_{n_p-N+1},\frac{v^{\perp}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}\})=vol^2(\{v^*_1,\cdots,v^*_{n_p-N+1},\frac{v^{\perp}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}\})$, where $\{v^*_1,\cdots,v^*_{n_p-N+1}\}$ can lead to the diagonal matrix $\mathrm{diag}(\lambda_1,\ \lambda_2,\ \cdots,\lambda_{n_p-N+1})$ with n_p-N+1 dimension, satisfying $v^*_1=(\lambda_1,0,\cdots,0)^{\top}$, $v^*_2=(0,\lambda_2,0,\cdots,0)^{\top},\cdots,v^*_{n_p-N+1}=(0,\cdots,0,\lambda_{n_p-N+1})^{\top}$. For this new formulation , we can easily know that $\frac{v^{\perp}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}$ will be $(0,\cdots,0,1)^{\top}$ with one additonal dimension and re-ordering, so we have $vol^2(\{v_1,\cdots,v_{n_p-N},\frac{v^{\perp}_{n_p-N+2}}{\|v^{\perp}_{n_p-N+2}\|}\})=\det(\mathrm{diag}((v_{\mathcal{V}\backslash\mathcal{N}_a}),1))=\det((v_{\mathcal{V}\backslash\mathcal{N}_a}))+\det(1)=vol^2(\{v_1,\cdots,v_{n_p-N}\})$.

$\{n_p - N + 2\}))) - \log(\det\ (\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a)))$. Let $\mathcal{A} = \mathcal{N}_a\backslash\{n_p - N + 1,\ n_p - N + 2\}$, $\mathcal{B} = \mathcal{N}_a\backslash\{n_p - N + 2\}$, and $\mathcal{A} \subset \mathcal{B}$, we have:

$$\begin{aligned}
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{A})) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{A} \cup \{n_p - N + 2\}))) \leq \\
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{B}))) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{B} \cup \{n_p - N + 2\})))) \\
&\qquad\qquad \Longrightarrow \qquad\qquad (B.81) \\
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{A} \cup \{n_p - N + 2\}))) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{A})) \geq \\
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{B} \cup \{n_p - N + 2\})))) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{B}))).
\end{aligned}$$

Let's consider a set $\widehat{\mathcal{A}}$, satisfying $\{\widehat{\mathcal{A}} \cup \{n_p - N + 2\}\} \subset \mathcal{V}$, which is the subset of the set $\widehat{B} = \widehat{\mathcal{A}} \cup \{\widehat{i}_1,\ \widehat{i}_2, \cdots,\ \widehat{i}_{|\widehat{\mathcal{B}}|-|\widehat{\mathcal{A}}|}\} \subset \mathcal{V}$. Based on (B.81), we have:

$$\begin{aligned}
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\widehat{\mathcal{A}} \cup \{n_p - N + 2\}))) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\widehat{\mathcal{A}})) \geq \\
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash((\widehat{\mathcal{A}} \cup \{\widehat{i}_1\}) \cup \{n_p - N + 2\}))) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\widehat{\mathcal{A}} \cup \{\widehat{i}_1\}))) \\
&\geq \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash((\widehat{\mathcal{A}} \cup \{\widehat{i}_1,\ \widehat{i}_2\}) \cup \{n_p - N + 2\}))) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\widehat{\mathcal{A}} \cup \{\widehat{i}_1,\ \widehat{i}_2\}))) \\
&\geq \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash((\widehat{\mathcal{A}} \cup \{\widehat{i}_1,\ \widehat{i}_2, \cdots,\ \widehat{i}_{|\widehat{\mathcal{B}}|-|\widehat{\mathcal{A}}|}\}) \cup \{n_p - N + 2\}))) \\
&- \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\widehat{\mathcal{A}} \cup \{\widehat{i}_1,\ \widehat{i}_2, \cdots,\ \widehat{i}_{|\widehat{\mathcal{B}}|-|\widehat{\mathcal{A}}|}\})))) \\
&= \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\widehat{\mathcal{B}} \cup \{n_p - N + 2\})))) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\widehat{\mathcal{B}}))).
\end{aligned} \tag{B.82}$$

So we can know that the sub-matrix selection of the log-determinant function of the weighted Laplacian matrix is a sub-modular optimization problem[§].

Then, let's consider the problem stated in (12), at every step, the anchor selection problem needs to operate three (2D) or six (3D) columns and rows corresponding to one pose. We only prove the 2D case here and the 3D case is totally similar.

[§]It is noted that this conclusion does not limit that the deleted state vectors belong to the locations or orientations, as shown in footnote 2, which means that our framework can be extended into the cases only anchoring the locations or orientation.

Assuming that we have a deleted rows and columns set $\mathcal{S}_a$ corresponding to the deleted poses $\mathcal{N}_a$ before deleting i-th and j-th poses, based on the sub-modularity of the sub-matrix selection of the log-determinant function of the weighted Laplacian matrix, we have:

$$\begin{aligned}
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2i-1, 2i, 2n_p+i\})))) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2i-1, 2i\}))))\\
&\geq \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2j-1, 2j, 2n_p+j\} \cup \{2i-1, 2i, 2n_p+i\}))))-\\
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2j-1, 2j, 2n_p+j\} \cup \{2i-1, 2i\}))))
\end{aligned} \tag{B.83}$$

$$\begin{aligned}
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2i-1, 2i\})))) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{S}_a \cup \{2i-1\}))) \geq\\
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2j-1, 2j, 2n_p+j\} \cup \{2i-1, 2i\}))))-\\
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2j-1, 2j, 2n_p+j\} \cup \{2i-1\}))))
\end{aligned} \tag{B.84}$$

$$\begin{aligned}
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2i-1\})))) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{S}_a)))\\
&\geq \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2j-1, 2j, 2n_p+j\} \cup \{2i-1\}))))\\
&- \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2j-1, 2j, 2n_p+j\}))))
\end{aligned} \tag{B.85}$$

Based on the sum of the inequations (B.83), (B.84), and (B.85), we have:

$$\begin{aligned}
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2i-1, 2i, 2n_p+i\})))) - \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{S}_a))) \geq\\
&\log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2j-1, 2j, 2n_p+j\} \cup \{2i-1, 2i, 2n_p+i\}))))\\
&- \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash(\mathcal{S}_a \cup \{2j-1, 2j, 2n_p+j\}))))
\end{aligned} \tag{B.86}$$

Rewrite (B.86) based on the pose and the objective function of the problem (12) and $\mathcal{N}_a \subset \mathcal{N}_a \cup \{j\}$, then we have:

$$f_{obj}(\mathcal{N}_a \cup \{i\}) - f_{obj}(\mathcal{N}_a) \geq f_{obj}(\mathcal{N}_a \cup \{i, j\}) - f_{obj}(\mathcal{N}_a \cup \{j\}) \tag{B.87}$$

Using the similar recursive idea in (B.82), we can get that $f_{obj}(\bullet)$ is a sub-modular

function.

e) Cardinality-fixed constraint At the same time, the constraint $|\mathcal{N}_a| = N$ is the cardinality-fixed constraint.

In conclusion, the problem (12) is a cardinality-fixed non-negative non-normalized non-monotone sub-modular optimization problem. So Theorem 3 is proved.

B.2.2 Proof of Lemma 2

Without loss of generality, we denote the solution of the problem (4.2) as $\mathcal{N}_a = \{\widehat{v}_1, \cdots, \widehat{v}_N\}$. Now we delete the corresponding row and column of $\boldsymbol{\mathcal{L}}$ from $\widehat{v}_1$ to $\widehat{v}_N$ one by one. For sub-matrices $\widehat{\boldsymbol{L}}_w^{\mathbb{R}^2}$ and $\widehat{\boldsymbol{L}}_w^{SO(n)}$, based on the Cauchy Interlacing Theorem, we have:

$$\begin{aligned}
&\lambda_1(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V})) \leq \lambda_1(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\{\widehat{v}_1\})) \leq \\
&\lambda_1(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\{\widehat{v}_1, \widehat{v}_2\})) \leq \cdots \leq \lambda_1(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)) \\
&\qquad\qquad\qquad\qquad \vdots \\
&\lambda_{n_p-N}(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V})) \leq \lambda_{n_p-N}(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\{\widehat{v}_1\})) \leq \\
&\lambda_{n_p-N}(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\{\widehat{v}_1, \widehat{v}_n\})) \leq \cdots \leq \lambda_{n_p-N}(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)),
\end{aligned} \tag{B.88}$$

$$\begin{aligned}
&\lambda_1(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)) \leq \cdots \leq \lambda_{N-2}(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\{\widehat{v}_1, \widehat{v}_2\})) \\
&\quad \leq \lambda_{N-1}(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\{\widehat{v}_1\})) \leq \lambda_{N+1}\left(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V})\right) \\
&\qquad\qquad\qquad \vdots \\
&\lambda_{n_p-N}(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)) \leq \cdots \leq \lambda_{n_p-2}(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\{\widehat{v}_1, \widehat{v}_2\})) \\
&\quad \leq \lambda_{n_p-1}(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\{\widehat{v}_1\})) \leq \lambda_{n_p}(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V})).
\end{aligned} \tag{B.89}$$

Because $f_{obj}\left(\mathcal{N}_a\right) = n\sum_1^{n_p-N}\log\lambda_i(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V}\backslash\mathcal{N}_a)) + d\sum_1^{n_p-N}\log\lambda_i(\widehat{\boldsymbol{L}}_w^{SO(n)}(\mathcal{V}\backslash\mathcal{N}_a))$,

we have:

$$
\begin{aligned}
&n \sum_{1}^{n_p-N} \log \lambda_i(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V})) + d \sum_{1}^{n_p-N} \log \lambda_i(\widehat{\boldsymbol{L}}_w^{SO(n)}(\mathcal{V})) \leq \\
&f_{obj}(\mathcal{N}_a) \leq n \sum_{N+1}^{n_p} \log \lambda_i(\widehat{\boldsymbol{L}}_w^{\mathbb{R}^n}(\mathcal{V})) + d \sum_{N+1}^{n_p} \log \lambda_i(\widehat{\boldsymbol{L}}_w^{SO(n)}(\mathcal{V})).
\end{aligned} \tag{B.90}
$$

The proof is completed.

B.2.3 Proof of Lemma 3

The value $\mathcal{U}_{greedy}$ obtained from the lazy greedy algorithm is naturally the lower bound of the optimal solution $\mathcal{U}_{opt}$. Meanwhile, because of the approximation factor of the random greedy algorithm, we have that $\left(\frac{1-\frac{N}{en_p}}{e} - \varepsilon\right)^{-1} \mathcal{U}^*_{greedy}$ is the upper bound of the optimal solution $\mathcal{U}_{opt}$. Combining Lemma 2, we can finally get Lemma 3.

B.3 Proofs in Chapter 5

B.3.1 Proof of the lower bound of the D-optimality metric in feature-based SLAM

Proof: Introduce the Jacobian matrix, we have:

$$
\begin{aligned}
&\boldsymbol{\mathcal{I}}(\boldsymbol{X}) = \boldsymbol{J}(\boldsymbol{X})^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{J}(\boldsymbol{X}) \\
&= \begin{bmatrix} \frac{\partial \boldsymbol{h_p}}{\partial \boldsymbol{p}}^\top & \frac{\partial \boldsymbol{h_\theta}}{\partial \boldsymbol{p}}^\top \\ \frac{\partial \boldsymbol{h_p}}{\partial \boldsymbol{\theta}}^\top & \frac{\partial \boldsymbol{h_\theta}}{\partial \boldsymbol{\theta}}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} \otimes \boldsymbol{I}_{2\times 2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{h_p}}{\partial \boldsymbol{p}} & \frac{\partial \boldsymbol{h_p}}{\partial \boldsymbol{\theta}} \\ \frac{\partial \boldsymbol{h_\theta}}{\partial \boldsymbol{p}} & \frac{\partial \boldsymbol{h_\theta}}{\partial \boldsymbol{\theta}} \end{bmatrix}
\end{aligned} \tag{B.91}
$$

For every part, we have:

$$\begin{aligned}
&\frac{\partial \boldsymbol{h_p}}{\partial \boldsymbol{p}}^{\top} \boldsymbol{\Sigma}_p^{-1} \otimes \boldsymbol{I}_{2\times 2} \frac{\partial \boldsymbol{h_p}}{\partial \boldsymbol{p}} = \left(\boldsymbol{A_g} \boldsymbol{R} \boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} \boldsymbol{R}^{\top} \boldsymbol{A}_{\boldsymbol{g}}^{\top}\right) \otimes \boldsymbol{I}_{2\times 2} \\
&\frac{\partial \boldsymbol{h_p}}{\partial \boldsymbol{p}}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} \otimes \boldsymbol{I}_{2\times 2} \frac{\partial \boldsymbol{h_p}}{\partial \boldsymbol{\theta}} = \left(\boldsymbol{A_g} \otimes \boldsymbol{I}_{2\times 2}\right) \boldsymbol{R} \boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} \otimes \boldsymbol{I}_{2\times 2} \boldsymbol{\Gamma} \boldsymbol{R}^{\top} \triangle_{\boldsymbol{w_p}} \\
&\frac{\partial \boldsymbol{h_p}}{\partial \boldsymbol{\theta}}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} \otimes \boldsymbol{I}_{2\times 2} \frac{\partial \boldsymbol{h_p}}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{h_\theta}}{\partial \boldsymbol{\theta}}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} \frac{\partial \boldsymbol{h_\theta}}{\partial \boldsymbol{\theta}} \\
&= \triangle^{\top} \boldsymbol{R} \boldsymbol{\Gamma}^{\top} (\boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} \otimes \boldsymbol{I}_{2\times 2}) \boldsymbol{\Gamma} \boldsymbol{R}^{\top} \triangle + \boldsymbol{A_p} \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} \boldsymbol{A}_p^{\top}
\end{aligned} \tag{B.92}$$

Based on $\boldsymbol{R}\boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1}\boldsymbol{R}^{\top} = \boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1}$, $\boldsymbol{\Gamma}\boldsymbol{R}^{\top} = \boldsymbol{R}^{\top}\boldsymbol{\Gamma}$, $\boldsymbol{\Gamma}^{\top}\boldsymbol{\Gamma} = \boldsymbol{I}_{2m\times 2m}$ and the FIM is a symmetrical matrix, we can get the FIM $\boldsymbol{\mathcal{I}}(\boldsymbol{X})$.

Based on the D-optimality criterion, we have:

$$\log(\det \begin{bmatrix} \boldsymbol{L}_{\boldsymbol{w_p}}^{\boldsymbol{g}} \otimes \boldsymbol{I}_{2\times 2} & \boldsymbol{A}_{\boldsymbol{w_p}}^{\boldsymbol{g}} \otimes \boldsymbol{I}_{2\times 2} \boldsymbol{\Gamma} \triangle_{\boldsymbol{w_p}} \\ (\boldsymbol{A}_{\boldsymbol{w_p}}^{\boldsymbol{g}} \otimes \boldsymbol{I}_{2\times 2} \boldsymbol{\Gamma} \triangle_{\boldsymbol{w_p}})^{\top} & \triangle_{\boldsymbol{w_p}}^{\top} \triangle_{\boldsymbol{w_p}} + \boldsymbol{L}_{\boldsymbol{w_\theta}}^{\boldsymbol{p}} \end{bmatrix}), \tag{B.93}$$

Schurs Determinant Formula: if $\widetilde{\boldsymbol{A}}^{-1}$ exists,

$$\det \begin{bmatrix} \widetilde{\boldsymbol{A}} & \widetilde{\boldsymbol{B}} \\ \widetilde{\boldsymbol{C}} & \widetilde{\boldsymbol{D}} \end{bmatrix} = \det(\widetilde{\boldsymbol{A}}) \det(\widetilde{\boldsymbol{D}} - \widetilde{\boldsymbol{C}} \widetilde{\boldsymbol{A}}^{-1} \widetilde{\boldsymbol{B}}) \tag{B.94}$$

Using the Schurs Determinant Formula, we can re-write the Eq.(46) into:

$$\det(\widetilde{\boldsymbol{A}}) = \det(\boldsymbol{L}_{\boldsymbol{w_p}}^{\boldsymbol{g}} \otimes \boldsymbol{I}^{2\times 2}) = \det(\boldsymbol{L}_{\boldsymbol{w_p}}^{\boldsymbol{g}})^2 \tag{B.95}$$

$$\begin{aligned}
&\det(\widetilde{\boldsymbol{D}} - \widetilde{\boldsymbol{C}} \widetilde{\boldsymbol{A}}^{-1} \widetilde{\boldsymbol{B}}) = \\
&\det(\triangle_{\boldsymbol{w_p}}^{\top} \triangle_{\boldsymbol{w_p}} + \boldsymbol{L}_{\boldsymbol{w_\theta}}^{\boldsymbol{p}} - \triangle_{\boldsymbol{w_p}}^{\top} \boldsymbol{\Gamma}^{\top} (\boldsymbol{A}_{\boldsymbol{w_p}}^{\boldsymbol{g}}{}^{\top} \boldsymbol{L}_{\boldsymbol{w_p}}^{\boldsymbol{g}}{}^{-1} \boldsymbol{A}_{\boldsymbol{w_p}}^{\boldsymbol{g}}) \otimes \boldsymbol{I}_{2\times 2} \boldsymbol{\Gamma} \triangle_{\boldsymbol{w_p}})
\end{aligned} \tag{B.96}$$

Eq.(B.96) can be writed as:

$$\begin{aligned} \text{Eq.(B.96)} =& 2\log\det(\boldsymbol{L}^{\boldsymbol{g}}_{\boldsymbol{w_p}}) + \log\det(\triangle^{\top}_{\boldsymbol{w_p}}\boldsymbol{P}^{\boldsymbol{g}}_{\boldsymbol{w_p}}\triangle_{\boldsymbol{w_p}} + \boldsymbol{L}^{\boldsymbol{p}}_{\boldsymbol{w_\theta}}) \\ \boldsymbol{P}^{\boldsymbol{g}}_{\boldsymbol{w_p}} =& \boldsymbol{I}_{2m\times 2m} - \boldsymbol{\Gamma}^{\top}(\boldsymbol{A}^{\boldsymbol{g}}_{\boldsymbol{w_p}}{}^{\top}\boldsymbol{L}^{\boldsymbol{g}}_{\boldsymbol{w_p}}{}^{-1}\boldsymbol{A}^{\boldsymbol{g}}_{\boldsymbol{w_p}}) \otimes \boldsymbol{I}_{2\times 2}\boldsymbol{\Gamma} \end{aligned} \tag{B.97}$$

For any two matrixes $\boldsymbol{N}$ and $\boldsymbol{M}$, meeting $\boldsymbol{M} \succeq \boldsymbol{0}$ and $\boldsymbol{N} \succeq \boldsymbol{0}$, we have:

$$\det(\boldsymbol{M}+\boldsymbol{N}) = \det(\boldsymbol{M}+\boldsymbol{N}) \geq \det(\boldsymbol{M}) \tag{B.98}$$

Because $\boldsymbol{P}^{g}_{w_p}$ is the orthogonal projection matrices, So we have:

$$\log(\det(\boldsymbol{\mathcal{I}}(\boldsymbol{X}))) \geq 2\log(\det(\boldsymbol{L}^{\boldsymbol{g}}_{\boldsymbol{w_p}})) + \log\left(\det\left(\boldsymbol{L}^{\boldsymbol{p}}_{\boldsymbol{w_\theta}}\right)\right) \tag{B.99}$$

It is proved.

B.3.2 Proof of Conclusion 1

Proof: The coefficient matrix $\boldsymbol{A_Z}$ has a speical struction. In fact, it shows the corresponding relationship between the state vectors of the submap and the joining global map. So it is a block matrix with multiple identity matrix and column full rank. At the same time, in every row, there is only one non-zero element, like:

$$\boldsymbol{A_Z} = \begin{bmatrix} & & & \vdots & & & \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ & & & \vdots & & & \end{bmatrix}^{r_1\times r_2}, \tag{B.100}$$

where $r_1 \geq r_2$ So we can get:

$$(\boldsymbol{A}^{\top}_{\boldsymbol{Z}} * \boldsymbol{A_Z})_{i,j} = \begin{cases} zero, & i \neq j, \\ non-zero, & i = j. \end{cases} \tag{B.101}$$

So it is a diagonal matrix.

B.3.3 Proof of Conclusion 2

Proof: (1) Based on (A.3), we have:

$$\lambda_i(\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}}) = \min_{\boldsymbol{\mathcal{R}}_{i-1}^{\top}\boldsymbol{x}=0} \frac{\boldsymbol{x}^{\top}\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}}\boldsymbol{x}}{\boldsymbol{x}^{\top}\boldsymbol{x}}. \tag{B.102}$$

When $\boldsymbol{x} = \boldsymbol{A}_{\boldsymbol{Z}}\boldsymbol{y}$, which is a special solution for (B.102), we have:

$$\min_{\boldsymbol{\mathcal{R}}_{k-1}^{\top}\boldsymbol{x}=0} \frac{\boldsymbol{x}^{\top}\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}}\boldsymbol{x}}{\boldsymbol{x}^{\top}\boldsymbol{x}} \leq \min_{\substack{\boldsymbol{\mathcal{R}}_{k-1}^{\top}\boldsymbol{x}=0 \\ \boldsymbol{x}=\boldsymbol{A}_{\boldsymbol{Z}}\boldsymbol{y}}} \frac{\boldsymbol{x}^{\top}\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}}\boldsymbol{x}}{\boldsymbol{x}^{\top}\boldsymbol{x}}. \tag{B.103}$$

Let $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{R}}_{k-1}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}$, we have:

$$Eq.(\text{B.103}) = \min_{\boldsymbol{\mathcal{B}}\boldsymbol{y}=0} \frac{\boldsymbol{y}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}^{\top}\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}}\boldsymbol{A}_{\boldsymbol{Z}}\boldsymbol{y}}{\boldsymbol{y}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}\boldsymbol{y}}. \tag{B.104}$$

Because $\boldsymbol{A}_{\boldsymbol{Z}}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}$ are the diagonal matrix, defined as:

$$\boldsymbol{A}_{\boldsymbol{Z}}^{\top}\boldsymbol{A}_{\boldsymbol{Z}} = \text{diag}(\lambda_1(\boldsymbol{A}_{\boldsymbol{Z}}), \lambda_2(\boldsymbol{A}_{\boldsymbol{Z}}), \cdots, \lambda_k(\boldsymbol{A}_{\boldsymbol{Z}})). \tag{B.105}$$

We can re-written the denominator $\boldsymbol{y}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}\boldsymbol{y}$ as: $\lambda_1(\boldsymbol{A}_{\boldsymbol{Z}})y_1^2 + \lambda_2(\boldsymbol{A}_{\boldsymbol{Z}})y_2^2 + \cdots \lambda_k(\boldsymbol{A}_{\boldsymbol{Z}})y_k^2$. Then, we have:

$$\boldsymbol{y}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}\boldsymbol{y} = \lambda_1(\boldsymbol{A}_{\boldsymbol{Z}})y_1^2 + \cdots \lambda_k(\boldsymbol{A}_{\boldsymbol{Z}})y_k^2 \geq \widehat{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}})(\boldsymbol{y}^{\top}\boldsymbol{y}). \tag{B.106}$$

So, we obtain:

$$\min_{\boldsymbol{\mathcal{B}}\boldsymbol{y}=0} \frac{\boldsymbol{y}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}^{\top}\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}}\boldsymbol{A}_{\boldsymbol{Z}}\boldsymbol{y}}{\boldsymbol{y}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}\boldsymbol{y}} \leq \min_{\boldsymbol{\mathcal{B}}\boldsymbol{y}=0} \frac{\boldsymbol{y}^{\top}\boldsymbol{A}_{\boldsymbol{Z}}^{\top}\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}}\boldsymbol{A}_{\boldsymbol{Z}}\boldsymbol{y}}{\widehat{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}})\boldsymbol{y}^{\top}\boldsymbol{y}}. \tag{B.107}$$

Based on **The Fischer's min-max theorem** ([99], p. 233) and $\widehat{\lambda}(A_Z) > 0$, we can get:

$$\lambda_i(\boldsymbol{\mathcal{I}_Z}) \leq \min_{\boldsymbol{\mathcal{B}y}=0} \frac{\boldsymbol{y}^\top \boldsymbol{A}_{\boldsymbol{Z}}^\top \boldsymbol{\mathcal{I}_Z} \boldsymbol{A_Z y}}{\widehat{\lambda}(\boldsymbol{A_Z}) \boldsymbol{y}^\top \boldsymbol{y}} \leq \frac{\lambda_i(\boldsymbol{\mathcal{I}_{all}})}{\widehat{\lambda}(\boldsymbol{A_Z})}. \tag{B.108}$$

$$\Longrightarrow \widehat{\lambda}(\boldsymbol{A_Z}) \lambda_i(\boldsymbol{\mathcal{I}_Z}) \leq \lambda_i(\boldsymbol{\mathcal{I}_{all}}). \tag{B.109}$$

In short, we has proved the left part of **Conclusion 3**.

(2) Then, let's prove the other one:

For $n - k + 1 \leq j \leq n$

$$\lambda_j(\boldsymbol{\mathcal{I}_Z}) = \max_{\mathcal{T}_{k+1}^\top \boldsymbol{x}=0} \frac{\boldsymbol{x}^\top \boldsymbol{\mathcal{I}_Z} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}. \tag{B.110}$$

When $\boldsymbol{x} = \boldsymbol{A_Z y}$, which is a special solution for (B.110), we have:

$$\max_{\boldsymbol{\mathcal{T}}_{k+1}^\top \boldsymbol{x}=0} \frac{\boldsymbol{x}^\top \boldsymbol{\mathcal{I}_Z} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \geq \max_{\substack{\boldsymbol{\mathcal{T}}_{k+1}^\top \boldsymbol{x}=0 \\ \boldsymbol{x}=\boldsymbol{A_Z y}}} \frac{\boldsymbol{x}^\top \boldsymbol{\mathcal{I}_Z} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}. \tag{B.111}$$

Let $\boldsymbol{\mathcal{B}_1} = \mathcal{T}_{k+1}^\top \boldsymbol{A_Z}$, we have:

$$\max_{\substack{\boldsymbol{\mathcal{T}}_{k+1}^\top \boldsymbol{x}=0 \\ \boldsymbol{x}=\boldsymbol{A_Z y}}} \frac{\boldsymbol{x}^\top \boldsymbol{\mathcal{I}_Z} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \max_{\boldsymbol{\mathcal{B}_1 y}=0} \frac{\boldsymbol{y}^\top \boldsymbol{A}_{\boldsymbol{Z}}^\top \boldsymbol{\mathcal{I}_Z} \boldsymbol{A_Z y}}{\boldsymbol{y}^\top \boldsymbol{A}_{\boldsymbol{Z}}^\top \boldsymbol{A_Z y}}. \tag{B.112}$$

Because $\boldsymbol{A}_{\boldsymbol{Z}}^\top \boldsymbol{A_Z}$ are the diagonal matrix, we have:

$$\boldsymbol{y}^\top \boldsymbol{A}_{\boldsymbol{Z}}^\top \boldsymbol{A_Z y} = \lambda_1(\boldsymbol{A_Z}) y_1^2 + \cdots \lambda_k(\boldsymbol{A_Z}) y_k^2 \leq \widetilde{\lambda}(\boldsymbol{A_Z})(\boldsymbol{y}^\top \boldsymbol{y}). \tag{B.113}$$

So we can get:

$$\max_{\boldsymbol{\mathcal{B}_1 y}=0} \frac{\boldsymbol{y}^\top \boldsymbol{A}_{\boldsymbol{Z}}^\top \boldsymbol{\mathcal{I}}_{\boldsymbol{Z}} \boldsymbol{A}_{\boldsymbol{Z}} \boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{A}_{\boldsymbol{Z}}^\top \boldsymbol{A}_{\boldsymbol{Z}} \boldsymbol{y}} \geq \max_{\boldsymbol{\mathcal{B}_1 y}=0} \frac{\boldsymbol{y}^\top \boldsymbol{A}_{\boldsymbol{Z}}^\top \boldsymbol{\mathcal{I}}_{\boldsymbol{Z}} \boldsymbol{A}_{\boldsymbol{Z}} \boldsymbol{y}}{\widetilde{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}})(\boldsymbol{y}^\top \boldsymbol{y})}. \tag{B.114}$$

Based on **The Fischer's min-max theorem** ([99], p. 233) and $\widetilde{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}}) > 0$, we can get:

$$\lambda_j(\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}}) \geq \max_{\boldsymbol{\mathcal{B}_1 y}=0} \frac{\boldsymbol{y}^\top \boldsymbol{A}_{\boldsymbol{Z}}^\top \boldsymbol{\mathcal{I}}_{\boldsymbol{Z}} \boldsymbol{A}_{\boldsymbol{Z}} \boldsymbol{y}}{\widetilde{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}})(\boldsymbol{y}^\top \boldsymbol{y})} \geq \frac{\lambda_{k-n+j}(\boldsymbol{\mathcal{I}}_{\boldsymbol{all}})}{\widetilde{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}})}. \tag{B.115}$$

$$\Longrightarrow \widehat{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}})\lambda_j(\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}}) \geq \lambda_{k-n+j}(\boldsymbol{\mathcal{I}}_{\boldsymbol{all}}). \tag{B.116}$$

Let's choose $j = n - k + i (1 \leq i \leq k)$:

$$\Longrightarrow \widehat{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}})\lambda_{n-k+i}(\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}}) \geq \lambda_i(\boldsymbol{\mathcal{I}}_{\boldsymbol{all}}). \tag{B.117}$$

Finally, based on (B.109) and (B.117), we have: For $i = 1, 2, \cdots, k$

$$\lambda_i(\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}})\widehat{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}}) \leq \lambda_i(\boldsymbol{\mathcal{I}}_{\boldsymbol{all}}) \leq \lambda_{n-k+i}(\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}})\widetilde{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}}). \tag{B.118}$$

It is proved.

In order to numerically verify Conclusion 2, we finish a small feature-based Linear SLAM with 6 poses and 5 features based on two small submaps, and then compute the log function of the eigenvalues $\lambda_i(\boldsymbol{\mathcal{I}}_{\boldsymbol{all}})$ of the global joining matrix and their lower $\lambda_i(\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}})\widehat{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}})$ and upper bounds $\lambda_{n-k+i}(\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}})\widetilde{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}})$, shown in Fig. B.1 .

We can see the log function (Green line) of the eigenvalues of joining FIM is bigger than their corresponding lower bound $\lambda_i(\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}})\widehat{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}})$ and smaller than their corresponding upper bound $\lambda_{n-k+i}(\boldsymbol{\mathcal{I}}_{\boldsymbol{Z}})\widetilde{\lambda}(\boldsymbol{A}_{\boldsymbol{Z}})$ (Red line).

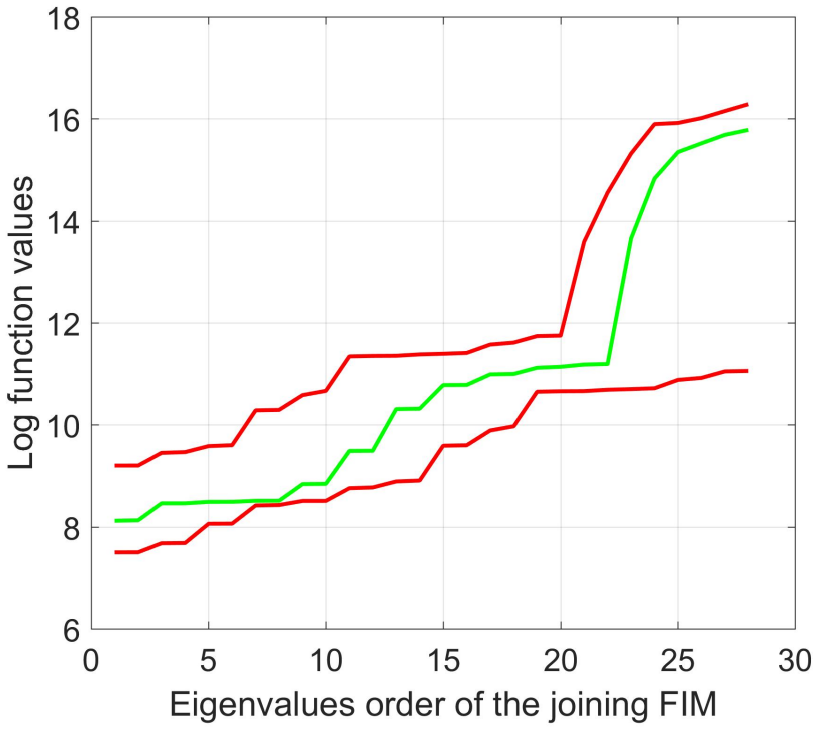


Figure B.1 : Log function of eignvalues and their bounds

Bibliography

[1] "Darpa subterranean (subt) challenge," https://www.subtchallenge.com, accessed: 2021-04-29.

[2] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[3] F. Bai, S. Huang, T. Vidal-Calleja, and Q. Zhang, "Incremental sqp method for constrained optimization formulation in slam," in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2016, pp. 1–6.

[4] F. Bai, T. Vidal-Calleja, and S. Huang, "Robust incremental slam under constrained optimization formulation," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1207–1214, 2018.

[5] R. A. Bailey and P. J. Cameron, "Combinatorics of optimal designs," *Surveys in combinatorics*, vol. 365, no. 19-73, p. 3, 2009.

[6] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[7] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[8] A. Barrau and S. Bonnabel, "An ekf-slam algorithm with consistency properties," *arXiv preprint arXiv:1510.06263*, 2015.

[9] N. Boumal, "On intrinsic cramér-rao bounds for riemannian submanifolds and quotient manifolds," *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1809–1821, 2013.

[10] N. Boumal, A. Singer, P.-A. Absil, and V. D. Blondel, "Cramér–rao bounds for synchronization of rotations," *Information and Inference: A Journal of the IMA*, vol. 3, no. 1, pp. 1–39, 2014.

[11] J. Briales and J. Gonzalez-Jimenez, "Cartan-sync: Fast and global se (d)-synchronization," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2127–2134, 2017.

[12] ——, "Initialization of 3d pose graph optimization using lagrangian duality," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5134–5139.

[13] O. Bruls, M. Arnold, and A. Cardona, "Two lie group formulations for dynamic multibody systems with large rotations," in *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2011, pp. 85–94.

[14] N. Buchbinder and M. Feldman, "Submodular functions maximization problems," 2017.

[15] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz, "Submodular maximization with cardinality constraints," in *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, 2014, pp. 1433–1452.

[16] D. Bump, *Lie groups*. Springer, 2004.

[17] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and

mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[18] L. Carlone, "A convergence analysis for pose graph optimization via gauss-newton methods," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 965–972.

[19] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 545–565, 2016.

[20] L. Carlone and F. Dellaert, "Duality-based verification techniques for 2d slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4589–4596.

[21] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 1–20, 2018.

[22] L. Carlone, D. M. Rosen, G. Calafiore, J. J. Leonard, and F. Dellaert, "Lagrangian duality in 3d slam: Verification techniques and optimal solutions," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 125–132.

[23] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3d slam: a survey on rotation estimation and its use in pose graph optimization," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4597–4604.

[24] M. P. d. Carmo, *Riemannian geometry*. Birkhäuser, 1992.

[25] H. Carrillo, P. Dames, V. Kumar, and J. A. Castellanos, "Autonomous robotic exploration using a utility function based on rényis general theory of entropy," *Autonomous Robots*, vol. 42, no. 2, pp. 235–256, 2018.

[26] H. Carrillo, Y. Latif, M. L. Rodriguez-Arevalo, J. Neira, and J. A. Castellanos, "On the monotonicity of optimality criteria during exploration in active slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1476–1483.

[27] H. Carrillo, I. Reid, and J. A. Castellanos, "On the comparison of uncertainty criteria for active slam," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2080–2087.

[28] S. Chaiken and D. J. Kleitman, "Matrix tree theorems," *Journal of combinatorial theory, Series A*, vol. 24, no. 3, pp. 377–381, 1978.

[29] Y. Chen, S. Huang, R. Fitch, and J. Yu, "Efficient active slam based on submap joining," in *Australasian Conference on Robotics and Automation, ACRA*, 2017.

[30] ——, "Efficient active slam based on submap joining, graph topology and convex optimization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[31] Y. Chen, S. Huang, R. Fitch, L. Zhao, H. Yu, and D. Yang, "On-line 3d active pose-graph slam based on key poses using graph topology and sub-maps," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 169–175.

[32] Y. Chen, S. Huang, L. Zhao, and G. Dissanayake, "Cramér-rao bounds and optimal design metrics for pose-graph slam: Technical report," Tech. Rep., 2020.

[33] ——, "Cramér–rao bounds and optimal design metrics for pose-graph slam," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 627–641, 2021.

[34] Y. Chen, G. Luo, Y. Mei, J. Yu, and X. Su, "Uav path planning using artificial potential field method updated by optimal control theory," *International Journal of Systems Science*, vol. 47, no. 6, pp. 1407–1420, 2016.

[35] Y. Chen, L. Zhao, K. M. B. Lee, C. Yoo, S. Huang, and R. Fitch, "Broadcast your weaknesses: cooperative active pose-graph slam for multiple robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2200–2207, 2020.

[36] E. S. Coakley and V. Rokhlin, "A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices," *Applied and Computational Harmonic Analysis*, vol. 34, no. 3, pp. 379–414, 2013.

[37] H. Cramér, *Mathematical methods of statistics (PMS-9)*. Princeton university press, 2016, vol. 9.

[38] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[39] F. Dellaert and M. Kaess, "Square root sam: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[40] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.

[41] M. P. Do Carmo, G. Fischer, U. Pinkall, and H. Reckziegel, "Differential geometry," in *Mathematical Models*. Springer, 2017, pp. 155–180.

[42] H. Durrant Whyte and T. Bailey, "Simultaneous localization and mapping:

part i," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[43] H. F. Durrant-Whyte, "Uncertain geometry in robotics," *IEEE Journal on Robotics and Automation*, vol. 4, no. 1, pp. 23–31, 1988.

[44] E. Eade, "Lie groups for 2d and 3d transformations."

[45] C. Estrada, J. Neira, and J. D. Tardós, "Hierarchical slam: Real-time accurate mapping of large environments," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 588–596, 2005.

[46] K. Fathian, K. Khosoussi, P. Lusk, Y. Tian, and J. P. How, "Clear: A consistent lifting, embedding, and alignment rectification algorithm for multi-agent data association," *arXiv preprint arXiv:1902.02256*, 2019.

[47] B. D. Flury, "Acceptance–rejection sampling made easy," *SIAM Review*, vol. 32, no. 3, pp. 474–476, 1990.

[48] J. Folkesson and H. Christensen, "Graphical slam-a self-correcting map," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1. IEEE, 2004, pp. 383–390.

[49] F. Gao, Y. Lin, and S. Shen, "Gradient-based online safe trajectory generation for quadrotor flight in complex environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3681–3688.

[50] M. Giamou, K. Khosoussi, and J. P. How, "Talk resource-efficiently to me: Optimal communication planning for distributed loop closure detection," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.

[51] J. Gillenwater, "Maximization of non-monotone submodular functions," 2014.

[52] G. H. Golub and L. C. F. Van, *Matrix computations*. JHU Press, 2012, vol. 3.

[53] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments." in *CIRA*, vol. 99. Citeseer, 1999, pp. 318–325.

[54] D. A. Harville, "Matrix algebra from a statistician's perspective," 1998.

[55] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[56] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based rules for designing consistent ekf slam estimators," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, 2010.

[57] S. Huang, "A review of optimisation strategies used in simultaneous localisation and mapping," *Journal of Control and Decision*, vol. 6, no. 1, pp. 61–74, 2019.

[58] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended kalman filter based slam," *IEEE Transactions on robotics*, vol. 23, no. 5, pp. 1036–1049, 2007.

[59] S. Huang, N. Kwok, G. Dissanayake, Q. Ha, and G. Fang, "Multi-step look-ahead trajectory planning in slam: Possibility and necessity," in *2005 IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 1091–1096.

[60] S. Huang, Y. Lai, U. Frese, and G. Dissanayake, "How far is slam from a linear least squares problem?" in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 3011–3016.

[61] S. Huang, H. Wang, U. Frese, and G. Dissanayake, "On the number of local minima to the point feature based slam problem," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2074–2079.

[62] S. Huang, Z. Wang, and G. Dissanayake, "Sparse local submap joining filter for building large-scale maps," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1121–1130, 2008.

[63] S. Huang, Z. Wang, G. Dissanayake, and U. Frese, "Iterated d-slam map joining: Evaluating its performance in terms of consistency, accuracy and efficiency," *Autonomous Robots*, vol. 27, no. 4, p. 409, 2009.

[64] S. Huang, Z. Wang, and G. Dissanayake, "Sparse local submap joining filter for building large scale maps," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1121–1130, 2008.

[65] V. Ila, L. Polok, M. Solony, and K. Istenic, "Fast incremental bundle adjustment with covariance recovery," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 175–184.

[66] V. Ila, L. Polok, M. Solony, and P. Svoboda, "Slam++ a highly efficient and temporally scalable incremental slam framework," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 210–230, 2017.

[67] V. Indelman, "No correlations involved: Decision making under uncertainty in a conservative sparse information space," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 407–414, 2016.

[68] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 849–882, 2015.

[69] S. T. Jawaid and S. L. Smith, "On the submodularity of sensor scheduling for estimation of linear dynamical systems," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 4139–4144.

[70] S. J. Julier and J. K. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 4. IEEE, 2001, pp. 4238–4243.

[71] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[72] L.-P. Kaelbling, M.-L. Littman, and A.-R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[73] M. Kaess and F. Dellaert, "Covariance recovery from a square root information matrix for data association," *Robotics and autonomous systems*, vol. 57, no. 12, pp. 1198–1210, 2009.

[74] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3281–3288.

[75] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[76] A. K. Kelmans and B. N. Kimelfeld, "Multiplicative submodularity of a matrix's principal minor as a function of the set of its rows and some combinatorial

applications," *Discrete Mathematics*, vol. 44, no. 1, pp. 113–116, 1983.

[77] K. Khosoussi, "Exploiting the intrinsic structures of simultaneous localization and mapping," Ph.D. dissertation, 2017.

[78] K. Khosoussi, M. Giamou, G. S. Sukhatme, S. Huang, G. Dissanayake, and J. P. How, "Reliable graph topologies for slam."

[79] K. Khosoussi, S. Huang, and G. Dissanayake, "Tree-connectivity: Evaluating the graphical structure of slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1316–1322.

[80] ——, "Novel insights into the impact of graph structure on slam," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2707–2714.

[81] A. Kim and R. M. Eustice, "Active visual slam for robotic area coverage: Theory and experiment," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 457–475, 2015.

[82] Y. Kim and A. Kim, "On the uncertainty propagation: Why uncertainty on lie groups preserves monotonicity?" in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3425–3432.

[83] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007, pp. 225–234.

[84] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2d mapping," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 22–29.

[85] D. Kopitkov and V. Indelman, "Computationally efficient belief space planning via augmented matrix determinant lemma and reuse of calculations," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 506–513, 2017.

[86] ——, "No belief propagation required: Belief space planning in high dimensional state spaces via factor graphs, the matrix determinant lemma, and re use of calculation," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1088–1130, 2017.

[87] A. Krause and D. Golovin, "Submodular function maximization." 2014.

[88] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies." *Journal of Machine Learning Research*, vol. 9, no. 2, 2008.

[89] J. J. Kuffner Jr and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning."

[90] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3607–3613.

[91] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[92] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, 1950.

[93] C. Leung, S. Huang, and G. Dissanayake, "Active slam using model predictive control and attractor based exploration," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 5026–5031.

[94] C. Leung, S. Huang, N. Kwok, and G. Dissanayake, "Planning under uncertainty using model predictive control for information gathering," *Robotics and Autonomous Systems*, vol. 54, no. 11, pp. 898–910, 2006.

[95] M. Liu, S. Huang, G. Dissanayake, and H. Wang, "A convex optimization based approach for pose slam problems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1898–1903.

[96] S. Liu, S. P. Chepuri, M. Fardad, E. Maşazade, G. Leus, and P. K. Varshney, "Sensor selection for estimation with correlated measurement noise," *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3509–3522, 2016.

[97] M. Lourakis and A. A. Argyros, "Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment?" in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1526–1531.

[98] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous robots*, vol. 4, no. 4, pp. 333–349, 1997.

[99] J. R. Magnus and H. Neudecker, *Matrix differential calculus with applications in statistics and econometrics*. Wiley, 2019.

[100] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet, "A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot," *Autonomous Robots*, vol. 27, no. 2, pp. 93–103, 2009.

[101] I. Maurović, M. Seder, K. Lenac, and I. Petrović, "Path planning for active slam based on the d* algorithm with negative edge weights," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 8, pp. 1321–1331, 2018.

[102] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using fastslam," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1985–1991.

[103] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," 2002.

[104] J. B. Mueller and R. Larsson, "Collision avoidance maneuver planning with robust optimization," in *International ESA Conference on Guidance, Navigation and Control Systems, Tralee, County Kerry, Ireland*, 2008.

[105] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[106] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[107] K. P. Murphy, "Bayesian map learning in dynamic environments," in *Advances in Neural Information Processing Systems*, 2000, pp. 1015–1021.

[108] S. Nasiri, R. Hosseini, and H. Moradi, "A recursive least square method for 3d pose graph optimization problem," *arXiv preprint arXiv:1806.00281*, 2018.

[109] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran, "Continuous-time trajectory optimization for online uav replanning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 5332–5339.

[110] S. Patil, G. Kahn, M. Laskey, J. Schulman, K. Goldberg, and P. Abbeel, "Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation," in *Algorithmic foundations of robotics XI*. Springer, 2015, pp. 515–533.

[111] L. M. Paz, J. D. Tardós, and J. Neira, "Divide and conquer: Ekf slam in $o(n)$," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1107–1120, 2008.

[112] A. Pázman, *Foundations of optimum experimental design*. Springer, 1986, vol. 14.

[113] C. R. Rao, "Information and the accuracy attainable in the estimation of statistical parameters," in *Breakthroughs in statistics*. Springer, 1992, pp. 235–247.

[114] M. L. Rodríguez-Arévalo, J. Neira, and J. A. Castellanos, "On the importance of uncertainty representation in active slam," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 829–834, 2018.

[115] D. Rosen and L. Carlone, "Computational enhancements for certifiably correct slam," in *International Conference on Intelligent Robots and Systems (IROS) in the workshop Introspective Methods for Reliable Autonomy*, 2017.

[116] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group," *The International Journal of Robotics Research*, p. 0278364918784361, 2016.

[117] D. M. Rosen, C. DuHadway, and J. J. Leonard, "A convex relaxation for approximate global optimization in simultaneous localization and mapping," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5822–5829.

[118] D. M. Rosen, M. Kaess, and J. J. Leonard, "An incremental trust-region method for robust online sparse least-squares estimation," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1262–1269.

[119] S. M. Ross, *Introduction to probability models*. Academic press, 2014.

[120] J. Saunderson, P. A. Parrilo, and A. S. Willsky, "Semidefinite descriptions of the convex hull of rotation matrices," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1314–1343, 2015.

[121] X. Shen and P. K. Varshney, "Sensor selection based on generalized information gain for target tracking in large sensor networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 363–375, 2014.

[122] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous robot vehicles*. Springer, 1990, pp. 167–193.

[123] S. T. Smith, "Covariance, subspace, and intrinsic crame/spl acute/r-rao bounds," *IEEE Transactions on Signal Processing*, vol. 53, no. 5, pp. 1610–1630, 2005.

[124] S. Thrun, W. Burgard, and D. Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Autonomous Robots*, vol. 5, no. 3-4, pp. 253–271, 1998.

[125] ——, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping," in *ICRA*, vol. 1, 2000, pp. 321–328.

[126] ——, *Probabilistic robotics*. MIT press, 2005.

[127] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Y. Ng, "Simultaneous mapping and localization with sparse extended information fil-

ters: Theory and initial results," in *Algorithmic Foundations of Robotics V*. Springer, 2004, pp. 363–380.

[128] Y. Tian, K. Khosoussi, M. Giamou, J. P. How, and J. Kelly, "Near-optimal budgeted data exchange for distributed loop closure detection," *arXiv preprint arXiv:1806.00188*, 2018.

[129] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 376–380, 1991.

[130] P. Vakili, H. Mirzaei, S. Zarbafian, I. C. Paschalidis, D. Kozakov, and S. Vajda, "Optimization on the space of rigid and flexible motions: an alternative manifold optimization approach," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 5825–5830.

[131] R. Valencia, M. Morta, J. Andrade-Cetto, and J. M. Porta, "Planning reliable paths with pose slam," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 1050–1059, 2013.

[132] J. Vallvé and J. Andrade-Cetto, "Active pose slam with rrt*," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2167–2173.

[133] P. l. Vernaza, J. Zhou, K. Kitani, J. A. Bagnell, and R. O. Dpps, "Clustering via determinantal point processes," 2014.

[134] H. Wang, G. Hu, S. Huang, and G. Dissanayake, "On the structure of nonlinearities in pose graph slam," *Robotics: Science and Systems VIII*, p. 425, 2013.

[135] H. Wang, S. Huang, U. Frese, and G. Dissanayake, "The nonlinearity structure

of point feature slam problems with spherical covariance matrices," *Automatica*, vol. 49, no. 10, pp. 3112–3119, 2013.

[136] H. Wang, S. Huang, K. Khosoussi, U. Frese, G. Dissanayake, and B. Liu, "Dimensionality reduction for point feature slam problems with spherical covariance matrices," *Automatica*, vol. 51, pp. 149–157, 2015.

[137] H. Wang, S. Huang, G. Yang, and G. Dissanayake, "Comparison of two different objective functions in 2d point feature slam," *Automatica*, vol. 97, pp. 172–181, 2018.

[138] Z. Wang, S. Huang, and G. Dissanayake, "D-slam: A decoupled solution to simultaneous localization and mapping," *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 187–204, 2007.

[139] E. W. Weisstein, "Modified bessel function of the first kind," 2002.

[140] S. Wenhardt, B. Deutsch, E. Angelopoulou, and H. Niemann, "Active visual object reconstruction using d-, e-, and t-optimal next best views," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–7.

[141] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte, "An efficient approach to the simultaneous localisation and mapping problem," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 406–411.

[142] J. Xavier and V. Barroso, "Intrinsic variance lower bound (ivlb): an extension of the cramér-rao bound to riemannian manifolds," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 5. IEEE, 2005, pp. v–1033.

[143] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake, "Convergence and consistency analysis for a 3-d invariant-ekf slam," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 733–740, 2017.

[144] Y. Zhang, T. Zhang, and S. Huang, "Comparison of ekf based slam and optimization based slam algorithms," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2018, pp. 1308–1313.

[145] L. Zhao, S. Huang, and G. Dissanayake, "Linear slam: A linear solution to the feature-based and pose graph slam based on submap joining," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 24–30.

[146] ——, "Linear slam: Linearising the slam problems using submap joining," *Automatica*, vol. 100, pp. 231–246, 2019.