Contributions to robotic manipulation under uncertainty : calibration, estimation, motion planning

Nguyen, Huy

2019

Nguyen, H. (2019). Contributions to robotic manipulation under uncertainty : calibration, estimation, motion planning. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/83537>

<https://doi.org/10.32657/10220/48014>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**CONTRIBUTIONS TO ROBOTIC MANIPULATION
UNDER UNCERTAINTY: CALIBRATION, ESTIMATION,
MOTION PLANNING**

HUY NGUYEN

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

2019

**CONTRIBUTIONS TO ROBOTIC MANIPULATION
UNDER UNCERTAINTY: CALIBRATION, ESTIMATION,
MOTION PLANNING**

HUY NGUYEN

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2019

HUY NGUYEN

*Contributions to Robotic Manipulation Under Uncertainty:
Calibration, Estimation, Motion Planning*

Nanyang Technological University

Control Robotics Intelligence Group (CRI)

School of Mechanical and Aerospace Engineering

Robotics Research Centre

50 Nanyang Avenue

N3-01a-01

Singapore 639798

STATEMENT OF ORIGINALITY

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

09-April-2019

.....

Date

.....

NGUYEN DINH HUY

SUPERVISOR DECLARATION STATEMENT

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

09-April-2019

.....

Date

.....

PHAM QUANG CUONG

AUTHORSHIP ATTRIBUTION STATEMENT

This thesis contains material from 2 papers published in the following peer-reviewed journals in which I am listed as an author.

Chapter 4 is published as **Nguyen, Huy**, and Quang-Cuong Pham (2018). “*On the covariance of X in $AX = XB$* ”. In *IEEE Transactions on Robotics* 34.6, pp. 1651–1657.

The contributions of the co-authors are as follows:

- Assoc. Prof. Quang-Cuong Pham provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts.
- The algorithm was proposed and implemented by me.
- I designed the study and performed all the laboratory work at the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. I also analyzed the data.

Appendix A is published as **Nguyen, Huy**, and Quang-Cuong Pham (2016). “*Time-optimal path parameterization of rigid-body motions: Applications to spacecraft reorientation*”. In: *Journal of Guidance, Control, and Dynamics* 39.7, pp: 1667-1671.

The contributions of the co-authors are as follows:

- Assoc. Prof. Quang-Cuong Pham provided the initial project direction.
- I wrote the drafts of the manuscript. The manuscript was revised together with Assoc. Prof. Quang-Cuong Pham.
- The algorithm was proposed and implemented by me.
- I designed the simulations, performed all the data collection and analyzed the data.

09-April-2019

.....

Date

.....

NGUYEN DINH HUY

Abstract

Robotics has played a considerable role in increasing industrial productivity across the globe over the last decades. However, the automation of complex manipulation tasks, such as robotic assembly, is still beyond the capabilities of today’s industrial robots. One of the main problems to achieve this level of automation is associated with the uncertainties in positions and orientations of object poses in real-world environments.

To tackle this problem, one of our key ideas is to explicitly represent and estimate uncertainties by using the calculus of probability theory and assigning probabilities to all potential positions of the object. As a result, the ambiguity and degree of certainty can be described in a mathematical way, which allows us to obtain a deep understanding of these uncertainties and design better solutions to the associated problems. In line with this idea, in this thesis, we present a number of contributions to robotic manipulation under uncertainty, including calibration, estimation and motion planning problems.

First, we propose a probabilistic framework to precisely keep track of uncertainties throughout the entire manipulation process. In agreement with common manipulation pipelines, we decompose the process into two consecutive stages, namely perception and physical interaction. Each stage is associated with different sources and types of uncertainties, requiring different techniques. We discuss which representation of uncertainties is the most appropriate for each stage (e.g. as probability distributions in $SE(3)$ during perception, as weighted particles during physical interactions), how to convert from one representation to another, and how to initialize or update the uncertainties at each step of the process (camera calibration, image processing, pushing, grasping, etc.). Finally, we demonstrate the benefit of this fine-grained knowledge of uncertainties in an actual assembly task.

Second, we also present an approach to estimate the uncertainties of the hand-eye

transformation. Even though the hand-eye calibration has been a long-established fundamental problem in the robot vision for several decades, there currently exists no method to derive the covariance of the hand-eye transformation. Such information is the most generic and relevant quantification of the uncertainties in the hand-eye calibration process. After obtaining the covariance of the hand-eye transformation, we also discuss a propagation method to compute the covariance of the object pose estimation in a real setting.

Next, we present other contributions to decrease the running time and to deal with outlier measurements when performing touch-based localization in cluttered environments. In fact, such outlier measurements often lead to significant loss in precision in existing approaches. Experiments showed that our algorithm could provide, in a timely fashion, accurate and reliable localization in cluttered environments, in the presence of outliers.

Finally, while the main contributions of this thesis lie in handling uncertainties in manipulation tasks, we also make a contribution toward solving the problem of trajectory planning with kinodynamic constraints in the space of rigid-body motion $SE(3)$.

Looking forward, it is our hope that this thesis will serve as a starting point for further development of the robotic automation of manipulation tasks.

Acknowledgements

This thesis is the culmination of my journey of PhD which was just like climbing a high peak step by step accompanied with encouragement, hardship, trust, and frustration. At this moment of accomplishment, I realized though only my name appears on the cover of this dissertation, a great many people including my family members, my friends, colleagues and many more have contributed to accomplishing this journey.

First and foremost, I would like to thank my supervisor Quang-Cuong Pham, who has gone above and beyond anything I could have expected from an advisor. I am thankful for his support, guidance in life and research during these years.

Thanks to the good friends I have here in CRI group—Puttichai Lertkultanon, Francisco Suarez-Ruiz, Ahmad Bin Anwar, Tien Hung Pham and many more. You made my experience in the lab special and unforgettable. We have had a lot of great discussions about both research and non-research. I cherish all the good times that we have spent together.

Also thanks to all Vietnamese friends I have here in NTU for the stimulating discussions, for all the badminton matches and for all the fun we have had in the last four years.

Last but not least, I am eternally grateful to my Mom, Dad, Brother and my beloved in-laws for always believing in me and encouraging me to follow my dreams. Most importantly, a very special thanks to my loving wife, Chau Phan, for your unconditional, persistent love, for your support and understanding during my pursuit of PhD degree that made the completion of thesis possible. You were always around at times I thought that it was impossible to continue, you helped me to keep things in perspective. Without you, I would not have come this far.

Contents

1	Introduction	17
1.1	Motivations	17
1.2	Contributions	21
1.2.1	A Probabilistic Framework for Tracking Uncertainties in Manipulation Tasks	21
1.2.2	Uncertainty in Hand-eye Calibration	22
1.2.3	Improvements to Touch-based Localization	23
1.2.4	Contribution to Motion Planning	23
1.3	Outline of the Thesis	23
2	Literature Review	25
2.1	Uncertainty Representation and Propagation	26
2.1.1	3D Pose Representation	26
2.1.2	Uncertainty Representation	28
2.1.3	Uncertainty Propagation	29
2.2	Uncertainties in Manipulation Tasks	30
2.3	Alternative Approaches to Deal with Uncertainty in Manipulation	36
2.3.1	Compliant Motions	37
2.3.2	Belief Space Planning	37
3	A Probabilistic Framework for Tracking Uncertainties in Robotic Manipulation	39
3.1	A Probabilistic Framework for Tracking Uncertainty	40
3.1.1	Choices of Uncertainty Representations	40
3.1.2	Perception	43

3.1.3	Physical Interactions	46
3.2	Application to Part Assembly	50
3.2.1	Single Pin Insertion	50
3.2.2	Double Pin Insertion	54
3.3	Conclusion	55
4	Uncertainty in Hand-eye Calibration	57
4.1	Related Works	59
4.1.1	Solutions to the Hand-eye Calibration	59
4.1.2	Uncertainties in the Hand-eye Calibration	60
4.2	Background	63
4.2.1	Formulation of the Hand-eye Calibration Problem	63
4.2.2	Representation of Rigid-body Transformations and of their Un- certainties	65
4.2.3	Forward and Backward Propagation of Covariance	67
4.3	Derivation of the Covariance of $\mathbf{X}$	68
4.3.1	Covariance of the Rotation Part of $\mathbf{X}$	68
4.3.2	Covariance of the Translation Part of $\mathbf{X}$	73
4.4	Experiments	76
4.4.1	Synthetic Calibration Data	76
4.4.2	Real Calibration Data	79
4.4.3	Covariance of the Object Pose Estimation	83
4.5	Conclusion	85
5	Improvements to Touch-based Estimation	87
5.1	Related Works and Problem Setting	89
5.1.1	Related Works	89
5.1.2	Bayesian estimation	91
5.1.3	The measurement model	92
5.2	Accelerating Bayesian Updates by Efficient Face Selection	94
5.2.1	Outline of the Algorithm	94
5.2.2	Algorithm Parameters	95
5.3	Outlier Classification Using RANSAC	97

5.3.1	Outline of the Algorithm	97
5.3.2	Algorithm Parameters	99
5.4	Simulations	100
5.4.1	Efficient Face Selection	100
5.4.2	Object Localization in a Cluttered Environment	102
5.5	Conclusion	104
6	Conclusion	105
6.1	Summary	105
6.2	Future Works	108
	Appendix A Contributions to Motion Planning	109
A.1	Approaches to Optimal, Kinodynamic Motion Planning	110
A.1.1	Sampling-based Path Planners	111
A.1.2	Trajectory Planning with Kinodynamic Constraints	113
A.1.3	Trajectory Planning Pipeline	114
A.2	Trajectory Planning with Kinodynamic Constraints in $SO(3)$, $SE(3)$	116
A.2.1	Overview	116
A.2.2	Interpolation in $SO(3)$ and $SE(3)$	117
A.2.3	Time-Optimal Path Parameterization in $SO(3)$	120
A.2.4	Implementation and Evaluation	123
A.2.5	Conclusion	127

Chapter 1

Introduction

1.1 Motivations

Robotics has been playing a considerable role in increasing the industrial productivity across the globe over the last decades. However, the automation is still limited to relatively simple, monotonous and repetitive manipulation tasks, such as picking and placing, welding, or spray painting. Meanwhile, the automation of more complex manipulation tasks, i.e. robotic furniture assembly, is still beyond the capabilities of today industrial robots. In fact, there are a number of challenging problems to achieve this level of automation. Among them is indeed the uncertainties in the positions and orientations of the object poses in the real world environments.

Uncertainties arise from two major sources: imperfect sensing and the physical interactions between the robot and its surrounding. Such uncertainties present major challenges in manipulation tasks in the real world environments since any robotic system would, at least, require the abilities to

- (i) properly describe and reason about the effects of the uncertainties of the manipulated objects as well as the environments;
- (ii) integrate the information about the uncertainties into a task plan to keep track

of the uncertainties before and after each action of the robot;

(iii) plan an appropriate sequence of interactions with the objects to achieve the desired result that is usually specified as the goal object poses.

“A robot that carries a notion of its own uncertainty and that acts accordingly is superior to one that does not.”-[Thrun et al., 2005]

To illustrate, consider the assembly task depicted in Fig. 1.1. For the robot to successfully assemble the two objects, the poses of the objects relative to the robot must be known to the system. However, uncertainties are introduced into the object pose estimation during camera calibration and image processing (e.g. *tool calibration*, *hand-eye calibration* and *object pose estimation*). Next, these uncertainties evolve during the physical interactions between the robot and the objects (i.e. *pushing*, *grasping*, etc.). Thus, the actual object poses before the last step of the assembly might significantly differ from the assumed poses, which in turn may cause assembly failures.

While it is not possible to totally eliminate all uncertainties – because of the inherent noise introduced during the various perception and action stages – we argue in this thesis that precisely keeping track of the uncertainties can significantly improve the speed and success rate of manipulation.

More specifically, we will present a probabilistic framework to precisely keep track of uncertainties throughout the entire manipulation process. In agreement with common manipulation pipelines [Suárez-Ruiz et al., 2018], we decompose the process into two consecutive stages, namely perception and physical interaction. Each stage is associated with different sources and types of uncertainties, requiring different techniques. We discuss which representation of uncertainties is the most appropriate for each stage (e.g. as probability distributions in $SE(3)$ during perception, as weighted particles during physical interactions), how to convert from one representation to another, and how to initiate or update the uncertainties at each step of the process (camera calibration, image processing, pushing, grasping, etc.). Finally, we demonstrate the benefit of this fine-grained

Figure 1.1: A typical assembly task. **A:** Setup, comprising a 6-DOF robot equipped with parallel-jaw gripper, a 3D camera, and objects to be assembled. **B:** Uncertainty in object pose estimation. **C:** The uncertainty on the pose of the brown object has evolved under the grasping action. **D:** The elliptical search pattern for assembly is optimized based on the fine-grained knowledge of the uncertainties.

knowledge of uncertainties in an actual assembly task.

While tracking uncertainties has long been a central theme in mobile robotics [Thrun et al., 2005], it has received comparatively less attention in industrial robotics in general, and robotic manipulation in particular. Yet, precisely tracking uncertainties is crucial for robots to successfully and safely operate in unstructured and dynamic environments which, according to many reports¹, are becoming more common in the industry.

In point of fact, the idea of developing methods for estimating uncertainties in a robot

¹See for instance “The Robotics Revolution: The Next Great Leap in Manufacturing” by the Boston Consulting Group <https://on.bcg.com/2FLfaGW>.

system is by no means new. Some fundamental works on modelling spatial uncertainties were studied in [Smith et al., 1990], [Durrant-Whyte, 1988], [Wang and Chirikjian, 2008], [Barfoot and Furgale, 2014]. The above developments have been utilized to establish a theoretical basis for a popular subfield of the *simultaneous localization and mapping* (SLAM) problem [Thrun et al., 2005, Durrant-Whyte and Bailey, 2006]. However, most existing researches have only focused on *mobile robotics*. Not much attention was paid explicitly to the modelling, estimating and tracking the uncertainties in *manipulation tasks*. In fact, most manipulation tasks are sequences of primitive phases consisting of calibration, perception and manipulating actions. Existing works, however, have only focused on estimating uncertainties in individual phases, not throughout the whole process, where multiple sources of uncertainties interact and compound [Sallinen, 2003].

The works closest to ours are [Su and Lee, 1991, Su and Lee, 1992a], in which authors presented a methodology for manipulating and propagating spatial uncertainties in a robotic assembly system for generating executable actions for accomplishing a desired task. Although the works proposed a novel framework to work with uncertainties, there was no discussion on how these uncertainties are estimated. Based on those works, [Sallinen, 2003] proposed methods for estimating the geometrical relationships between coordinate frames and the spatial uncertainties in estimated model parameters. However, later phases (i.e. manipulating actions) where contacts between robot(s) and the environment occur have not been investigated. On another note, their estimation model were based on an Euler angles parameterization, as opposed to the $SE(3)$ formulation, is well-known to involve singularities.

A alternative approach to deal with uncertainties while performing manipulation tasks is to plan the geometric motions of the objects and then execute these motions on a compliant robot. This simple approach has proved to be a key component for the success of many manipulation tasks [Suárez-Ruiz et al., 2018, Knepper et al., 2013, Wahrburg et al., 2014, Van Wyk et al., 2018]. To compute robust motions under the presence of

motion and sensing uncertainties, many works have incorporated such information into their planners [Phillips-Grafflin and Berenson, 2017, Sieverling et al., 2017, Wirnshofer et al., 2018]. This line of research are generally referred to as belief space planning [Platt Jr et al., 2010, Melchior and Simmons, 2007, Hauser, 2010]. Even though such approaches have enabled many works to reliably accomplish complex manipulation tasks in realistic, uncertain environment, they commonly requires a set of hypotheses of pose for every manipulated object in the environment. In fact, these sets of hypotheses are often assumed to be much larger than the real ones to account for ambiguous estimations. Consequently, robots have to perform many more redundant motions and spend more time to reason the effects of these uncertainties. This lack of access to explicit uncertainty information motivates us to study and proposed a new probabilistic framework for tracking uncertainties in manipulation tasks. In turn, the capabilities to obtain such fine-grain information enabled by our framework can also be utilized in these planning approaches to achieve a higher level of performance when dealing with uncertainties.

With this motivation, the methodology developed in this thesis is designed in such a way that it is applicable to different types of manipulation tasks and can be carried out in either a factory environment or an unstructured environment. Looking forward, we believe the capabilities achieved by our proposed techniques will definitely put a significant milestone towards the robotic automation of manipulation tasks.

1.2 Contributions

1.2.1 A Probabilistic Framework for Tracking Uncertainties in Manipulation Tasks

Many existing works on manipulation tasks commonly assume that all inputs of planning algorithms; such as environment models, robot kinematics, and initial object poses; are precise. However, there are always uncertainties in those knowledge, i.e. the vision

system can only provide data with some limited accuracy. These uncertainties can propagate and even be amplified via various interactions during manipulation. As a result, executing such erroneous plans can eventually cause damages to the manipulated objects or even to the robotic system itself. Therefore, it is crucial to investigate how the knowledge of the uncertainties can be obtained and how such knowledge can be incorporated into the planning stage and also the motion execution stage in order to produce sufficiently robust manipulation motions.

With the stated purpose, **the first contribution of this thesis is a probabilistic framework to precisely keep track of the uncertainties in manipulation tasks.** In fact, we decompose the manipulation task into two consecutive stages, namely perception and physical interactions. Each stage is associated with different sources and types of uncertainties, requiring different techniques. We discuss which representation of uncertainties is the most appropriate for each stage (e.g. as probability distributions in $SE(3)$ during perception, as weighted particles during physical interactions), how to convert from one representation to another, and how to initiate or update the uncertainties at each step of the process (camera calibration, image processing, pushing, grasping, etc.). Finally, we demonstrate the benefit of this fine-grained knowledge of uncertainties in an actual assembly task.

1.2.2 Uncertainty in Hand-eye Calibration

Along the line of estimating the uncertainties in the perception stage, **we also present a novel approach to estimate the uncertainties of the hand-eye transformation.** Even though the hand-eye calibration has been a long-established fundamental problem in the robot vision for several decades, there has yet to be any works that provide a derivation of the covariance of the hand-eye transformation. Such information is ultimately the most generic and relevant quantification of the uncertainties in the hand-eye calibration process. After obtaining the covariance of the hand-eye transformation, we also discuss

the propagation method to compute the covariance of the object pose estimation in a real setting.

1.2.3 Improvements to Touch-based Localization

Touch-based localization is one of the important actions in the physical interaction stage. One major difficulty is that, when the target object is placed within a cluttered environment, the robot might touch a different object and/or obstacles instead of the target object, generating thereby an outlier measurement. Existing methods are inherently fragile with respect to such outliers. To address this problem, we then present **another contributions to decrease the running time and to deal with outlier measurements when performing touch-based localization in cluttered environments**.

1.2.4 Contribution to Motion Planning

While the main contributions of this thesis lie in handling uncertainties in manipulation task, we also make an improvement in motion planning problem. As execution time is one of the crucial factors in industrial productivity, apart from being safe, the generated motions should be also fast. We first briefly review a number of existing approaches to kinodynamic motion planning and discuss the methods we employ to generate robot trajectories in our system. Then **we present another contribution which is trajectory planning with kinodynamic constraints in the space of rigid-body motion $SE(3)$** .

1.3 Outline of the Thesis

This thesis consists of 6 chapters, including this introductory chapter. The rest of the thesis is organized as follows.

In Chapter 2, we give an overview of the literature regarding methods of modelling the spatial uncertainties in a robotic manipulation, hand-eye calibration, work object

localization. This chapter provides a bird’s-eye view of the existing literature in the fields and describe how our contributions fit in the literature.

The next three chapters discuss our works on handling the uncertainties in different phases of a manipulation task. In Chapter 3, we present a probabilistic framework developed for estimating and handling the uncertainties in manipulation tasks. We also show the results of a hardware experiment in which we exploit the benefits of having explicit uncertainty information to accomplish a part assembly task. Chapter 4 presents our study on estimating the uncertainties of the solution of the hand-eye calibration problem. We also provide a discussion on how we conduct the experiments to estimate the uncertainties of the hand-eye transformation and of the pose estimation in a real setting. Chapter 5 discusses the touch-based estimation problem and present the improvements we made to decrease the running time and to deal with outlier measurements.

Chapter 6 draws the conclusions and sketches directions for future works.

Appendix A is devoted to the motion planning problem. We briefly review a number of existing approaches to kinodynamic motion planning and discuss the methods we employ to generate robot trajectories in our system. We then also present our contribution for the problem of planning fast trajectories in the spaces of three-dimensional rotations $SO(3)$ and three-dimensional rigid body motions $SE(3)$ under kinodynamic constraints in cluttered environments.

Chapter 2

Literature Review

Today, robots are required to operate in increasingly unstructured environments. While robots working in an assembly line carry out identical tasks everyday, those working in “modern” factory where new parts to be assembled/produced are introduced every months need to accommodate numerous challenges that arise in unstructured and dynamic environments. As a result, robots more rely on sensor inputs and softwares to observe the environments. However, these sensors have yet to be perfect, which inherently introduces uncertainties in the knowledge of robots about the surrounding environments. This leads to a fundamental problem that is to equip robots with an ability to cope with uncertainties. In fact, there have been many researchers who tackled this problem over the past decades. In this Chapter, we will provide a structural literature review on those previous works that focus on uncertainties in manipulation tasks.

The Chapter consists of 3 Sections, namely Uncertainty Representation and Propagation, Uncertainties in Manipulation Tasks and Alternative Approaches to Deal with Uncertainty in Manipulation. In each Section, we review related research works and describe how the contributions of this thesis fits in the literature.

2.1 Uncertainty Representation and Propagation

As the core idea of the thesis is to explicitly represent and propagate the uncertainties in the system, we will first provide an overview of existing mathematical tools for modeling and propagating the uncertainties over the past decades.

2.1.1 3D Pose Representation

The pose of a rigid object is defined by six degrees of freedom (6DOF) in which three components describe the object orientation whereas the other three describe the object position. The challenge in choosing a representation of a pose is that representations have either singularities or constraints. This results from the fact that the orientation variables are not vectors, but rather members of a noncommutative group. Furthermore, in many problems in robotics, a pose of an object must be composed, inverted, differentiated and interpolated. Without a coherent and robust framework for representing and working with 3D poses, these tasks would require a great deal of effort.

In point of fact, this problem has attracted an extensive attention from many researchers in the scope of robotics. Some of them include [Murray et al., 1994a] and [Park, 1991] who have presented related theoretic notations and terminologies which have now become standard vocabulary in the robotics community. Their fundamental methodologies were based on the use of 4×4 **homogenous transformation matrices** and exponential mapping to represent 3D poses. In particular, we briefly recall here the standard definitions for the rotation group $\mathbf{SO}(3)$ and the special Euclidean group $\mathbf{SE}(3)$.

The special orthogonal group $\mathbf{SO}(3)$ is a matrix Lie group representing rotations, and its associated Lie algebra, denoted $\mathfrak{so}(3)$ with the matrix commutator as Lie bracket. The Lie algebra of $\mathbf{SO}(3)$ denoted by $\mathfrak{so}(3)$ is given by the set of 3×3 real skew-symmetric

matrices of the form

$$[\boldsymbol{\omega}] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \quad (2.1)$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ is the axis-angle representation for rotation. Elements of $\mathbf{SO}(3)$ are given by the 3×3 real matrices $\mathbf{R}$ satisfying $\mathbf{R}^\top \mathbf{R} = \mathbb{I}$ and $\det \mathbf{R} = 1$.

The group of rigid transformations in 3D space, $\mathbf{SE}(3)$ (the special Euclidean group), is well represented by linear transformations on homogeneous four-vectors:

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}, \text{ with } \mathbf{R} \in \mathbf{SO}(3), \mathbf{t} \in \mathbb{R}^3. \quad (2.2)$$

A minimal representation of this transformation is defined by the corresponding Lie algebra $\mathfrak{se}(3)$ which is the tangent space of $\mathbf{SE}(3)$ at the identity. The algebra elements are 6-vectors $(\mathbf{v}, \boldsymbol{\omega})^\top$ where $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ is the axis-angle representation for rotation, and $\mathbf{v}$ is a rotated version of the translation $\mathbf{t}$.

Elements of the $\mathfrak{se}(3)$ algebra can be mapped to the $\mathbf{SE}(3)$ group using the exponential mapping $\exp()$:

$$\exp([\mathbf{v}, \boldsymbol{\omega}]) \stackrel{\text{def}}{=} \begin{bmatrix} \exp([\boldsymbol{\omega}]) & \mathbf{V}\mathbf{v} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}. \quad (2.3)$$

Here,

$$\exp([\boldsymbol{\omega}]) = \mathbb{I} + \frac{\sin \|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|} [\boldsymbol{\omega}] + \frac{1 - \cos \|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^2} [\boldsymbol{\omega}]^2 \quad (2.4)$$

is the well-known *Rodrigues' formula*,

$$\mathbf{V} = \mathbb{I} + \frac{1 - \cos \|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^2} [\boldsymbol{\omega}] + \frac{\|\boldsymbol{\omega}\| - \cos \|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^3} [\boldsymbol{\omega}]^2, \quad (2.5)$$

and $[\cdot]$ is an operator which maps a 3-vector to its skew-symmetric matrix (we also use

this operator to turn $(\omega, \mathbf{v})^\top$ into a 4×4 member of Lie algebra $\mathfrak{se}(3)$.

The inverse of the exponential, or logarithm, is used to obtain the vector of exponential coordinates ϕ from the group element $\mathbf{R} \in \mathbf{SO}(3)$ as $\phi = (\log \mathbf{R})^\vee$.

There are two main advantages to use Lie group and Lie algebra for representing 3D poses. First, it is free of singularities and avoids the need to enforce constraints when solving optimal estimation problems. For example, the singularities due to the use of local coordinates are inevitable when one chooses to describe rotation using Euler angles. Second, the Lie groups and their associated mathematical machinery provide a rigorous treatment to address many essential operations, and have proven to be the most effective tool used to resolve many principal issues in robotics.

2.1.2 Uncertainty Representation

In early works by [Taylor, 1976] and [Brooks, 1982], uncertainty is simply represented by worst-case bounds which include all possible errors (min-max approach). Although this approach is simple, it usually results in conservative estimates which make it difficult to apply in decision-making process.

To address this limitation, the probabilistic approach which uses the calculus of probability theory and assign probabilities to all potential positions of the object has been proposed. The pose of an object is now represented by a probability distribution over the space [Su and Lee, 1991], [Su and Lee, 1992b]. As a result, the probabilistic representation can make use of probability theory and, thus, provide more uncertainty manipulation methodologies (e.g. propagation, fusion, etc.) [Brooks, 1985], [Smith and Cheeseman, 1986], [Smith et al., 1990], [Durrant-Whyte, 1988]. This approach has then been utilized to establish a theoretical basis for a popular subfield of the *simultaneous localization and mapping* (SLAM) problem [Durrant-Whyte and Bailey, 2006]. Recently, this probabilistic approach has been further investigated in [Wang and Chirikjian, 2008], [Chirikjian, 2009], [Chirikjian, 2011], [Barfoot and Furgale, 2014] where they provided a rigorous treatment

of representing and propagating uncertainty on $SO(3)$ and $SE(3)$. For example, in [Barfoot and Furgale, 2014], by representing a mean as a 4×4 transformation matrix and its associated uncertainty as a 6×1 perturbation (with an associated 6×6 covariance matrix), these method attained a simple but useful and easy-to-implement representation and provided a straightforward, practical approach for associating uncertainty with transformation matrices. Most of the above methods share a basic idea that beliefs are represented by multivariate normal distributions (Gaussian distributions). Although this is by far the most popular family of techniques to date, there are a number of shortcomings. Most importantly, Gaussian distributions are unimodal (probability density exhibits single peak). Such assumption is typical of many applications in which the true distribution centers around the true pose with a small margin of uncertainty. However, in other cases where one must handle multimodal distributions, this would produce a poor match to the real distributions. In that situation, particles and mixture of Gaussians representations are more appropriate. Since each representation possesses different advantages and disadvantages, suitable choices should be made based on the problem nature.

2.1.3 Uncertainty Propagation

The problem of uncertainty propagation on the Euclidean group also appears in a large number of applications in robotics. For example, in manipulator calibration, the uncertainties along different joints can add up and produce the pose errors at the end-effector. If we describe the errors at each joint as a distribution around a nominal one, what “probabilistic methodology” should be used to propagate these errors? In addition, in case many uncertain measurements from multiple sensors are obtained (e.g. the poses of an object are obtained from different cameras), how can these measurements be optimally combined or fused into a single estimate? In fact, due to the non-trivial Lie group structure [Murray et al., 1994a], the usual approach of additive uncertainty in vector spaces cannot be directly applied to members of $SO(3)$ and $SE(3)$.

Over the past decades, there have been a number of works devoted to this problem. Some of the earliest ones include [Brooks, 1985], [Smith and Cheeseman, 1986] and [Su and Lee, 1992b]. In these works, uncertainties are assumed to be small enough that Jacobian-based methods or first-order error propagation theories can be applied to propagate uncertainties. To extend the use of these methods to uncertainties that are not necessarily small, [Wang and Chirikjian, 2008] and [Barfoot and Furgale, 2014] spread the approximation to second-order (and fourth-order) in the expansion of the matrix exponential parameterization of $SE(3)$. This extension brings a significant improvements in term of correctly keeping track of uncertainties on $SE(3)$; and therefore, is more robust over a wide range of applications such as kinematic chain in manipulator arms, mobile robot localization and mapping. Besides, it is worth-noting that the ability of these methods to correctly keep track of uncertainties on $SE(3)$ falls off with increasing uncertainty. This implies that to it may be wise to track uncertainties locally on $SE(3)$ to retain a good approximation. In the cases where uncertainties must be kept globally, approach by [Lee et al., 2008] is preferable.

With regard to the fusion problem in which several uncertain pose estimates are combined into a single estimate, the common approach is to consider this as uncertain measurements of the true pose and perform an iterative scheme to find the optimal pose. Some recent works on this problem include [Smith et al., 2003, Long et al., 2013, Barfoot and Furgale, 2014].

2.2 Uncertainties in Manipulation Tasks

In the last Section, we discussed the mathematical theories for representing and manipulating uncertainties that have been developed over the past decades. As a result, these algorithms have provided a sound methodology and brought significant progress in many difficult areas. Particularly in *mobile robotics*, these developments have been utilized to

establish a theoretical basis for the *simultaneous localization and mapping* (SLAM) problem and have emerged as the sole winner for this difficult problem. In fact, many mathematical and theoretical frameworks mentioned in last Section have been introduced from this community. However, the topic of mobile’s localization and mapping is beyond the scope of this thesis, hence will not be addressed any further. Interested readers can refer to comprehensive reviews on historical developments and specific algorithms at [Durrant-Whyte and Bailey, 2006, Thrun et al., 2005].

On the other side, not much attention was paid explicitly to the modelling, estimating and tracking the uncertainties in *manipulation tasks*. In fact, most of the manipulation tasks are a sequence of primitive phases consisting of calibration, perception and manipulating actions. Existing works, however, have only focused on estimating uncertainties in *individual* phases, not throughout the whole process, where multiple sources of uncertainties interact and compound [Sallinen, 2003].

The works closest to ours are [Su and Lee, 1991, Su and Lee, 1992a], in which authors presented a methodology for manipulating and propagating spatial uncertainties in a robotic assembly system for generating executable actions for accomplishing a desired task. The paper explored the probabilistic representation to handle the spatial uncertainties in assembly tasks. The spatial uncertainty of the location of an object was modeled by motion parameters and characterized by a mean vector and a covariance matrix. Manipulation of uncertainties was focused on compounding the uncertainties and fusing the estimates of the same locations of an object in terms of homogeneous transforms. These manipulation methods of uncertainties were further employed to propagate uncertainties for moving actions, perception actions, and contact actions. Although the work proposed a novel framework to work with uncertainties, there was no discussion on how these uncertainties are estimated. Based on those works, [Sallinen, 2003] proposed methods for estimating the geometrical relationships between coordinate frames and the spatial uncertainties in estimated model parameters. Synthesis method of sensing

planning that used the same form of modelling spatial uncertainties was also presented. However, later phases (i.e. manipulating actions) where contacts between robot(s) and the environment occur have not been investigated. On another note, their estimation model were based on an Euler angles parameterization, as opposed to the $SE(3)$ formulation, is well-known to involve singularities.

To address this problem, we present in Chapter 3 a probabilistic framework to precisely keep track of uncertainties in manipulation tasks. We then demonstrated the benefits of having access to such valuable information by conducting a part assembly experiment on a real setting. In agreement with common manipulation pipelines, we decompose the process into two consecutive stages, namely perception and physical interaction. Each stage is associated with different sources and types of uncertainties, requiring different techniques. The rest of this Section, hence, will give a brief review on these sources of uncertainties and their related problems.

Manipulator Calibration

Along the line of estimating uncertainties in the manipulation system, several works have been proposed to assess the goodness of the **manipulator calibration**. As summarized in [Liou et al., 1993, Karan and Vukobratovic, 1994], the position accuracy of the robot end-effector has been known to originate from many factors such as kinematic parameter variation due to manufacturing and assembly errors, influence of dynamic parameters, friction, backlash, resolution and non-linearity of joint position sensors, etc.

In [Chen and Chao, 1987], authors identified and parameterized the sources that contribute to the positioning error of a manipulator and estimated the values of the parameters. [Veitschegger and Wu, 1987] presented a calibration algorithm for finding the values of the independent kinematic errors by measuring the end-effector Cartesian positions and proposed two compensation algorithms: a differential error transform compensation algorithm and a Newton–Raphson compensation algorithm. [Menq and Borm, 1988] pre-

sented a framework to characterize the distribution of the position errors of robot manipulators over the work space and to study their statistical properties. [Renders et al., 1991] presented a calibration technique of robots based on a maximum likelihood approach for the identification of errors of position and orientation. Besides, together with the developments of many advanced methods for robot calibration, robots are now better built (e.g. have better repeatability), measurement equipments are also better (i.e. more accurate, though certainly not much more affordable, for example a laser tracker still costs more than \$100,000). The mathematical models for robot description have also been further optimized. Consequently, this has brought significant progress on improving the robot accuracy [Khalil and Dombre, 2004, Nubiola and Bonev, 2013, Siciliano and Khatib, 2016].

Hand-eye Calibration

Another closely related line of research is the **hand-eye calibration problem** in which the rigid-body transformation between a camera mounted on the robot end-effector and the end-effector itself is estimated. Mathematically, this problem usually is formulated as $\mathbf{AX} = \mathbf{XB}$, where $\mathbf{X}$ is the unknown 4×4 hand-eye transformation matrix and $\mathbf{A}$ and $\mathbf{B}$ are known 4×4 transformation matrices. As mentioned earlier, this transformation and its associated uncertainty play a crucial role in studying the uncertainties in manipulation tasks. Starting from the late 1980's, a large amount of literature has been devoted to this problem, and a number of efficient methods have been developed. However, there are a limited number of works studies on the uncertainty of hand-eye calibration. In order to have a closer look at this problem and understand why finding the covariance of $\mathbf{X}$ is challenging, we review more thoroughly the related literature in the Section [4.1](#).

Physical Interactions

As mentioned earlier, during the later phases when a contact occurs, the uncertainties of the contacted objects are often constrained and changed. In fact, all manipulating actions involves **physical interactions** with the environment/objects, hence can be also considered as a perceiving action in the sense that sometimes a contact action can reduce the uncertainty of the contacted objects. This idea originates from the truth that humans rely heavily on the sense of touch for manipulation tasks [Flanagan et al., 2006]. Similar to many biological systems, this creates rich and more informative sensory signals that would otherwise not be present. Such fruitful information in the combined space of sensory data and action parameters facilitates the prediction and interpretation of the sensory signal. Working towards the ability to exploit this knowledge in robots, there are a wide spectrum of works on estimating object poses using physical interactions considering different types of objects, sensors, different kinds of contact actions, etc. Recently, [Bohg et al., 2017] provided a comprehensive survey on this field. In the survey, these approaches are subsumed under the term *Interactive Perception* or *Perceptive Manipulation*. Through out the survey, the authors also emphasized the blurred boundary which is traditionally drawn between manipulation and perception.

In this work, we focus on manipulating tasks that usually perform on industrial assembly where common elements, i.e. force/torque sensors, 3D cameras and parallel-jaw grippers, are ubiquitous due to their strength, robustness, cost-effectiveness, ease of integration, and many more. The work in this thesis will only cover two manipulating actions commonly performed in manipulation, which are planar grasping actions and touch-based localization. For other kinds of interactions using different types of sensors, grippers, we refer interested readers to the above survey.

We stress again that although there are a vast number of works has been proposed in the literature to deal with uncertainties in manipulating tasks, existing works have only focused on several *single* tasks, but not in whole systems with accumulations of spatial

uncertainties including several phases: robot calibration, hand-eye calibration, camera-based object localization, physical interactions, etc. Moreover, the benefits of having access to explicit uncertainty information will be studied and demonstrated in designing better solution to manipulation tasks,i.e. assembling tasks.

In Section 5.1, we will give a in-depth review the *tactile (or touch-based) localization problem*. The rest of this section will focus on the other type of physical interaction,i.e. planar grasping action.

Many Bayesian estimation frameworks for touch-based localization have been shown to be effectively applicable and well suit to the planar grasping action, and vice versa. However, the planar grasping action produces relatively information-poor measurements, i.e. post-grasp gripper distances. Apart from that, the planar grasping action is closely related to *pushing* which involves finite motions on a supporting surface with ambiguous pressure and friction distribution. As a result, analytical measurement models are also much more complex and hard to obtain.

To perform object localization during the planar grasping action, a common approach is using physical models to build state estimators and track the pose of the object from the post-grasp gripper distances [Koval et al., 2016, Zhou et al., 2017]. In many preliminary works, these models were usually built based on the assumption of quasi-static rigid body mechanics with Coulomb friction [Mason, 1986, Goyal et al., 1991]. It is because this assumption allows the problem to be more approachable and attains simpler models. Besides, this is also considered as a suitable assumption for the scales and speeds in various robotic manipulation applications. More recently, many follow-up works focus on proposing data-driven models as an alternative approach so that the objects' motions can be captured more accurately in real environments [Walker and Salisbury, 2008, Lau et al., 2011, Meriçli et al., 2015, Zhou et al., 2016]. However, one of the challenges when applying this approach is the large amount of experimental data required to identify the model.

In Chapter 3, Section 3.1.3, we will present a more in-depth discussion on how to perform the object localization during the planar grasping action.

2.3 Alternative Approaches to Deal with Uncertainty in Manipulation

A common approach to deal with uncertainties while performing manipulation tasks is to plan the geometric motions of the objects and then execute these motions on a compliant robot. This simple approach has proved to be a key component for the success of many manipulation tasks [Suárez-Ruiz et al., 2018, Knepper et al., 2013, Wahrburg et al., 2014, Van Wyk et al., 2018]. To compute robust motions under the presence of motion and sensing uncertainties, many works have incorporated such information into their planners [Phillips-Grafflin and Berenson, 2017, Sieverling et al., 2017, Wirnshofer et al., 2018]. This line of research are generally referred to as belief space planning [Platt Jr et al., 2010, Melchior and Simmons, 2007, Hauser, 2010]. Even though such approaches have enabled many works to reliably accomplish complex manipulation tasks in realistic, uncertain environment, they commonly requires a set of hypotheses of pose for every manipulated object in the environment. In fact, these sets of hypotheses are often assumed to be much larger than the real ones to account for ambiguous estimations. Consequently, robots have to perform many more redundant motions and spend more time to reason the effects of these uncertainties. This lack of access to explicit uncertainty information motivates us to study and proposed a new probabilistic framework for tracking uncertainties in manipulation tasks. In turn, the capabilities to obtain such fine-gain information enabled by our framework can also be utilized in these planning approaches to achieve a higher level of performance when dealing with uncertainties.

2.3.1 Compliant Motions

Compliance is a key component for many manipulation tasks under uncertainty. Nowadays, most robots used in the industry are stiff, position-controlled. This means these robots are able to obtain very precise control in position and velocity, but have a limited or even no control in force and torque. As a consequence, during manipulation tasks, even a small uncertainty in the system can cause contact forces that easily exceed the physical capabilities of the robots. Therefore compliant motions play an important role for a safe and successful interaction between robots in uncertain environment. Compliant motions have enabled many works to reliably perform part insertions, for example [Van Wyk et al., 2018, Suárez-Ruiz et al., 2018]. On the other side, one can also use compliant motions to maintain contact with an object to reduce sensing time during the data collection of touch-based localization problem. [Petrovskaya and Khatib, 2011a].

Compliant motions can be achieved by either software or hardware. Often, the former is known as active compliance as it is realized by force-based feedback control. The latter is passive compliance as it uses a compliant mechanism to handle contacts between the robot and the environment. Compared to passive compliant, active compliant motion gives a robot the flexibility to handle a wide range of tasks where compliance is required. An early survey of active compliance is provided at [Lefebvre et al., 2005]. General techniques for achieving compliant motion include model-based [Bruyninckx et al., 1995, De Schutter et al., 2007] and reinforcement learning approaches [Gullapalli et al., 1994, Levine et al., 2015, Levine et al., 2016].

2.3.2 Belief Space Planning

Another related problem is planning under motion and sensing uncertainties which is generally known as belief space planning. Exact solutions to the Partially-Observable Markov Decision Process (POMDP) only exist for a very limited problem class. As a

result, research has focused on various approaches to approximating the solution. One approach is the “most-likely state” approximation. This method assumes that the true state of the MDP that underlies the POMDP is in fact the mode of the current belief state. Actions are taken according to the optimal policy in the underlying MDP. The approach considers stochasticity in the underlying process dynamics, but assumes no state uncertainty exists in the future. More sophisticated versions of this approximation include Q-MDP [Littman et al., 1995] and FIB [Hauskrecht, 2000].

Another approach is the belief roadmap approach [Prentice and Roy, 2010] which is applicable to POMDPs with continuous state, action, and observation spaces. The work ranks paths through a probabilistic roadmap defined in the underlying state space in terms of the change in covariance over the path. Based on the assumption of Gaussian process and measurement noise, many other efficient algorithms have been proposed [Platt Jr et al., 2010, Bry and Roy, 2011, Van Den Berg et al., 2011] to tackle the computational complexity. Van der Berg *et al.* propose a related approach where a set of potential trajectories are evaluated by tracking the belief state along each trajectory and selecting the one that minimizes the likelihood of failure [Van Den Berg et al., 2011]. Other approaches apply RRTs to belief space planning problems [Melchior and Simmons, 2007, Hauser, 2010]. Recently, many works employ sampling-based belief approximations where robot and environment uncertainty are represented using particles [Kim and Likhachev, 2017, Phillips-Grafflin and Berenson, 2017, Sieverling et al., 2017]. In these works, a physics simulator is commonly used in order to integrate each particle’s system dynamics. Though these approaches are designed to deal with non-linearities and/or hybrid dynamics (which are usually encountered in manipulation tasks), they are known to have very high computational complexity.

Chapter 3

A Probabilistic Framework for Tracking Uncertainties in Robotic Manipulation

While tracking uncertainties has long been a central theme in mobile robotics [Thrun et al., 2005], it has received comparatively less attention in industrial robotics in general, and robotic manipulation in particular. Yet, precisely tracking uncertainties is crucial for robots to successfully and safely operate in unstructured and dynamic environments which, according to many reports¹, are becoming more common in the industry.

While it is not possible to totally eliminate all uncertainties – because of the inherent noise introduced during the various perception and action stages – we argue in this chapter that precisely keeping track of the uncertainties can significantly improve the speed and success rate of manipulation.

More specifically, we present a probabilistic framework to precisely keep track of uncertainties throughout the entire manipulation process. In agreement with common manipulation pipelines [Suárez-Ruiz et al., 2018], we decompose the process into two

¹See for instance “The Robotics Revolution: The Next Great Leap in Manufacturing” by the Boston Consulting Group <https://on.bcg.com/2FLfaGW>.

consecutive stages, namely perception and physical interaction. Each stage is associated with different sources and types of uncertainties, requiring different techniques. We discuss which representation of uncertainties is the most appropriate for each stage (e.g. as probability distributions in $SE(3)$ during perception, as weighted particles during physical interactions), how to convert from one representation to another, and how to initiate or update the uncertainties at each step of the process (camera calibration, image processing, pushing, grasping, etc.). Finally, we demonstrate the benefit of this fine-grained knowledge of uncertainties in an actual assembly task.

3.1 A Probabilistic Framework for Tracking Uncertainty

In general, most typical manipulation tasks can be decomposed into two consecutive stages, namely perception and physical interactions, see Figure 3.1. The rest of this section will discuss which representation of uncertainties is the most appropriate for each stage, and how to initialize or update the uncertainties at each step of both stages.

3.1.1 Choices of Uncertainty Representations

During Perception Stage

During this stage, we choose to represent uncertainties of 3D poses as probability distribution in $SE(3)$. This is accomplished by storing the means as a (singularity-free) transformation matrix and using a (constraint-sensitive) perturbation of the pose (with associated covariance matrices). In particular, we model the uncertainties of the rotation and the translation parts separately as follows:

We assume that the rotation part of a pose is corrupted as below

$$\mathbf{R} = \exp([\xi_R])\bar{\mathbf{R}}, \quad (3.1)$$

Figure 3.1: A typical assembly task. **A**: Setup, comprising a 6-DOF robot equipped with parallel-jaw gripper, a 3D camera, and objects to be assembled. **B**: Uncertainty in object pose estimation. **C**: The uncertainty on the pose of the brown object has evolved under the grasping action. **D**: The elliptical search pattern for assembly is optimized based on the fine-grained knowledge of the uncertainties.

where $\bar{\mathbf{R}} \in SO(3)$ is the mean of $\mathbf{R}$, and the small perturbation variable $\xi_{\mathbf{R}} \in \mathbb{R}^3$ is zero-mean Gaussian with covariance matrix $\Sigma_{\mathbf{R}}$.

The translation part of a pose is corrupted as below

$$\mathbf{t} = \xi_{\mathbf{t}} + \bar{\mathbf{t}}, \quad (3.2)$$

where $\bar{\mathbf{t}} \in \mathbb{R}^3$ is the mean of $\mathbf{t}$, and $\xi_{\mathbf{t}} \in \mathbb{R}^3$ is the zero-mean Gaussian perturbation with covariance matrix $\Sigma_{\mathbf{t}}$.

This representation is free of singularities and avoids the need to enforce constraints

when solving optimal estimation problems. Moreover, such representation also allows us to utilize the Lie groups and their associated mathematical machinery to provide a rigorous treatment to address many essential operations, i.e. propagation and fusion of uncertainties (see later in 3.1.2). Thanks to these advancement, we are able to store uncertainty information as analytical form and evaluate the uncertainties of object pose estimation in timely fashion.

Note that, the representation also implies that rotation and translation noises are independent², which on some occasions (e.g. manipulator with large deflections on the robot’s links) might require further considerations. Because of this assumption, the space we consider is, strictly speaking, $SO(3) \times \mathbb{R}^3$, rather than $SE(3)$.

Under this representation, belief states are represented by multivariate normal distributions (Gaussian distributions). Although it can bring us a fast, analytical method to estimate uncertainties in the perception stage, there are a number of shortcomings. Most importantly, Gaussian distributions are unimodal (probability density exhibits single peak). However, in other cases where one must handle multimodal distributions, i.e. during the physical interaction stage, particles are more preferable.

During Physical Interaction Stage

Unlike the perception stage where camera measurements are easily used to infer the poses of the objects, physical interactions only provide local information about those poses. Consequently, motion and observation models during this stage are often highly non-linear and lack of analytic derivatives. Moreover, the resulting belief state is usually non-Gaussian and may be multi-modal. This, therefore, naturally leads to the use of particle filter and its variants to track the belief state during the physical interaction stage.

²Since there is in general no bi-invariant distance on $SE(3)$ [Park and Ravani, 1997], when solving optimal estimation problems, finding the rotation and translation components simultaneously would require a non-trivial rotation/translation weighting in any cases. As a result, we choose to model the uncertainties of the rotation and the translation parts separately instead of working directly on $SE(3)$.

3.1.2 Perception

Regarding the perception stage, we are interested in estimating the poses of the objects and their associated uncertainties with respect to the robot base frame. As illustrated in Figure 3.1, such uncertainties are definitely originated from three main sources which include: (i) the uncertainty of the object pose estimation in the camera frame, (ii) the uncertainty of the camera position (commonly known as the hand-eye transformation), and (iii) the uncertainty of the robot end-effector positioning.

$${}_b\mathbf{T}^{o1} = {}_b\mathbf{T}^c {}_c\mathbf{T}^{o1} \quad (3.3)$$

We note that in this setting the camera is mounted at a fixed position in the environment (not on the robot end-effector). Hence, the uncertainty of the robot end-effector positioning does not contribute directly to the uncertainties of the object poses. Instead, it affects the uncertainty of the camera position via the hand-eye calibration process [Nguyen and Pham, 2018].

In this section, we shall provide a methodology to estimate explicitly these sources of uncertainties. We will also give a discussion on how the uncertainties can be propagated so that the poses of the objects and their associated uncertainties with respect to the robot base frame can be eventually obtained.

Estimating Uncertainties

- *Uncertainty on the robot positioning ${}_b\mathbf{T}^e$* : Thanks to the advancement of calibration methods and measurement equipment, robots are now better built with higher accuracy. After calibration, a common positioned control industrial robot can achieve sub-millimeter in the mean position errors of the end-effector. For example, our experiments show that the mean of the position errors of our calibrated manipulator is about 0.3 mm. This can be considered negligible compared to other sources of

uncertainties in our system, e.g. the uncertainty of the pose estimation using our camera system. Because of this result, we find it usually safe to assume accurate joint measurements and robot model.

- *Uncertainty on the hand-eye calibration ${}_bT^c$* : Commonly, this problem can be formulated as: solve for X in $AX = XB$, where X is the unknown 4×4 hand-eye transformation matrix and A and B are known 4×4 transformation matrices.

Here, we are interested, not merely in solving for the hand-eye transformation, but more comprehensively, in evaluating its covariance. To address this problem, we follow [Nguyen and Pham, 2018] where they propose a novel algorithm to rigorously work out such a derivation. In particular, the technique exploits the benefits of applying optimization techniques directly on $SE(3)$ to obtain the derivation of the covariance of the hand-eye transformation.

- *Uncertainty on the camera-based pose estimation ${}_cT^o$* : Regarding the camera-based pose estimation problem, its uncertainty mainly depends on the intrinsic calibration of the camera. However, a model to capture the uncertainties of the camera parameters is highly non-linear and difficult to be evaluated. Hence, we decide to estimate such uncertainty by drawing a number of samples and performing Monte Carlo estimation. Despite the fact that this approach is slow, it is proven to bring out a more accurate estimation. A more detailed clarification of how this Monte Carlo estimation performed in the real systems can be found in the Experimental Section of [Nguyen and Pham, 2018].

Propagating Uncertainties

After obtaining all sources of uncertainties in the perception stage, we are now in a position to estimate the uncertainty of the object pose with respect to the robot base frame. In this work, we adopt the covariance propagation method proposed in [Barfoot

and Furgale, 2014] to the case where rotation and translation are decoupled.

Consider two noisy poses $\mathbf{T}_1$ and $\mathbf{T}_2$, whose nominal values and associated uncertainties are $\{\bar{\mathbf{R}}_1, \Sigma_{\mathbf{R}_1}\}$, $\{\bar{\mathbf{t}}_1, \Sigma_{\mathbf{t}_1}\}$ and $\{\bar{\mathbf{R}}_2, \Sigma_{\mathbf{R}_2}\}$, $\{\bar{\mathbf{t}}_2, \Sigma_{\mathbf{t}_2}\}$ respectively.

Let $\mathbf{T}_{12} = \mathbf{T}_1 \mathbf{T}_2$ be the compounded pose, we have

$$\bar{\mathbf{R}}_{12} = \bar{\mathbf{R}}_1 \bar{\mathbf{R}}_2, \quad \bar{\mathbf{t}}_{12} = \bar{\mathbf{R}}_1 \bar{\mathbf{t}}_2 + \bar{\mathbf{t}}_1. \quad (3.4)$$

Similar to [Barfoot and Furgale, 2014] (Section III), the covariance matrix of the rotation can be estimated by:

$$\begin{aligned} \Sigma_{\mathbf{R}_{12}} &\approx \Sigma_{\mathbf{R}_1} + \Sigma'_{\mathbf{R}_2} \\ &+ \frac{1}{12}(\mathcal{A}_1 \Sigma'_{\mathbf{R}_2} + \Sigma'_{\mathbf{R}_2} \mathcal{A}_1^\top + \Sigma_{\mathbf{R}_1} \mathcal{A}_2 + \Sigma_{\mathbf{R}_1} \mathcal{A}_2^\top) + \frac{\mathcal{B}}{4}, \end{aligned} \quad (3.5)$$

$$\text{where} \quad \Sigma'_{\mathbf{R}_2} := \bar{\mathbf{R}}_1 \Sigma_{\mathbf{R}_2} \bar{\mathbf{R}}_1^\top, \quad (3.6)$$

$$\mathcal{A}_1 := \langle \langle \Sigma_{\mathbf{R}_1} \rangle \rangle, \mathcal{A}_2 := \langle \langle \Sigma'_{\mathbf{R}_2} \rangle \rangle, \mathcal{B} := \langle \langle \Sigma_{\mathbf{R}_1}, \Sigma'_{\mathbf{R}_2} \rangle \rangle, \quad (3.7)$$

$$\text{and} \quad \langle \langle \mathbf{M} \rangle \rangle := -\text{tr}(\mathbf{M})\mathbb{I} + \mathbf{M}, \quad (3.8)$$

$$\langle \langle \mathbf{M}, \mathbf{N} \rangle \rangle := \langle \langle \mathbf{M} \rangle \rangle \langle \langle \mathbf{N} \rangle \rangle + \langle \langle \mathbf{NM} \rangle \rangle, \quad (3.9)$$

with $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n \times n}$. Regarding the translation vector, its covariance matrix can be estimated simply by using the forward propagation method:

$$\Sigma_{\mathbf{t}_{12}} \approx \Sigma_{\mathbf{t}_1} + \bar{\mathbf{R}}_1 \Sigma_{\mathbf{t}_2} \bar{\mathbf{R}}_1^\top + [\bar{\mathbf{R}}_1 \bar{\mathbf{t}}_2] \Sigma_{\mathbf{R}_1} [\bar{\mathbf{R}}_1 \bar{\mathbf{t}}_2]^\top. \quad (3.10)$$

In summary, to compound two poses, we propagate the means using (3.4) and the covariances using (3.5, 3.10).

It is also worth noting that the ability to correctly keep track of uncertainties on $\mathbf{SE}(3)$ decreases with increasing uncertainty, as error increases with increasing uncertainty. This suggests that it may be wise to use only relative pose variables in order to retain good approximations.

3.1.3 Physical Interactions

In this work, we focus on manipulating tasks that usually perform on industrial assembly where common elements, i.e. force/torque sensors and parallel-jaw grippers, are ubiquitous due to their strength, robustness, cost-effectiveness, ease of integration, and many more. The work in this Chapter will only cover two manipulating actions commonly performed in manipulation, which are planar grasping actions and touch-based localization.

Problem Formulation

Given the initial distribution of the objects obtained from the previous perception stage, the goal is to update the object distributions after the planar grasping action or/and the touch-based localization. As commonly known, the Bayesian filter is considered as the most general algorithm for filtering a belief state given initial knowledge and a sequence of actions and observations. Hence, the problem will be cast into the Bayesian framework and be addressed as a nonlinear filtering problem as shown below.

Let $\mathbf{X}$ be the state of a dynamical system which evolves under actions $\mathbf{u}$ and provides observations $\mathbf{y}$. Starting with $P(\mathbf{X}_0)$ – the *prior* distribution over the state $\mathbf{X}$ – the goal is to recursively update the following conditional probability

$$P(\mathbf{X}_{t+1}|\mathbf{y}) = \eta P(\mathbf{y}|\mathbf{X}_t)P(\mathbf{X}_t). \quad (3.11)$$

Here $P(\mathbf{X}_{t+1}|\mathbf{y})$ is known as the *posterior*, which represent our uncertain belief about the state $\mathbf{X}$ after having incorporated the measurement $\mathbf{y}$. On the right-hand side, the first factor $P(\mathbf{y}|\mathbf{X}_t)$ is the *measurement probability*, which encodes the likelihood of the measurement given the state (*measurement model*). The second factor $P(\mathbf{X}_t)$ is the *prior*, which represents our belief about $\mathbf{X}$ before obtaining the measurements $\mathbf{y}$. The factor η is a normalizing factor independent of the state $\mathbf{X}_t$ and needs not be computed.

Over the next sections, we will discuss how to apply this filter into two common contact actions, which are planar grasping actions and touch-based localization.

Planar Grasping

Regarding planar grasping action, the state is the pose $X \in SE(2)$ of the manipulated object. Actions are motions of the hand, given by the velocity u . During contact, the object moves with a velocity $f_\phi(X, u)$ where the function f encodes the physics of the object motion in response to the intended motion of the gripper. The parameter ϕ includes environmental properties. In this work, we particularly build analytical state estimators to track the poses of the objects from the post-grasp gripper distances based on the works from [Mason, 1986, Goyal et al., 1991]. This motion model is constructed following the assumption of quasi-static rigid body mechanics with Coulomb friction. Such assumption not only allows us to attain more approachable and simpler models, but also well suit the scale and speed of our application.

Relating to the Bayesian updates, the particle filter³ first samples the particles X^i from the prior distribution, then uses the motion model in order to simulate and obtain X^i_{sim} . Once the final object poses are acquired, their associated finger widths $d_{X^i_{\text{sim}}}$ can be estimated. This information will be used to compute an important weight for each forward-simulated particle. During the weighting step, the particles which are consistent with our measurement y will be assigned with higher probability. In particular, the post grasp distance measurements are assumed to be corrupted by Gaussian noise with the

³Particle filters represent the prior distribution by a set of particles/samples. Each particle has a likelihood weight assigned to it that represents the probability of that particle being sampled from the distribution. These weights are then updated following the conditional probability (see Equation 3.11). Next, in the resampling step, the particles with negligible weights are replaced by new particles in the proximity of the particles with higher weights. These steps are repeated until termination criteria met. Many different versions of particle filters exist, interested readers might refer to [Doucet et al., 2001] for a survey.

Figure 3.2: Once a planar grasping action is successfully performed, it sometimes can significantly improve the estimation of the objects in the closing direction of the gripper.

variance σ_d . The measurement probability is computed as follows

$$P(\mathbf{y}|X_t^i) = \eta_y \exp \left(-\frac{1}{2} \frac{(\mathbf{y} - d_{X_{\text{sim}}^i})^2}{\sigma_d^2} \right), \quad (3.12)$$

where η_y is a constant and will be taken into account during the normalization.

It is also worth-noting that once a planar grasping action is successfully performed, it can significantly improve the estimation of the objects in the closing direction of the gripper. Especially, in the cases where the geometries of the grasped objects are simple, e.g. rectangular boxes, the uncertainties of the objects could shrink to an one-dimensional distribution, see Figure 3.2. Besides, in this work, we also assume that the objects do not end up slipping out or being jammed at an undesired position. This assumption is considered reasonable in this case because the object geometries are simple and the slow

Figure 3.3: The object localization problem via touch. The photo shows the robot interacting with a wooden stick using a pin and its force/torque sensor.

movement of the fingers is always under control.

Touch-based Localization

With regards to touch-based localization, this type of interaction is often employed to further improve the pose estimation of the objects. As mentioned earlier, the objects to be located are, in general, assumed to be static during the measurement collection. This commonly-chosen assumption is realistic in such cases of handling very slight contact and the objects that are heavy or mounted on a support fixture to prevent their possible movements.

In this problem, the state is the pose $X \in SE(3)$ of an object O with a known shape. The measurements $\mathbf{y} = \mathbf{y}_0, \dots, \mathbf{y}_n$ are obtained by touching the object with the end-effector of the robot (as shown in Figure 3.3). Each measurement $\mathbf{y}_k := (\mathbf{y}_k^{\text{pos}}, \mathbf{y}_k^{\text{nor}})$ consists of a contact position $\mathbf{y}_k^{\text{pos}}$ and a contact normal $\mathbf{y}_k^{\text{nor}}$.

In fact, many works have been proposed and are able to solve the 6-DOF localization problem efficiently and reliably. In this work, we use the *proximity measurement model* and *Scaling Series* method in [Petrovskaya and Khatib, 2011a, Nguyen and Pham, 2017] to perform the touch-based localization owing to its computational efficiency.

3.2 Application to Part Assembly

In this section, we discuss an application of our framework to parts assembly tasks. The robotic platform used in this work is characterized by cost-efficient, off-the-shelf components combined with a classical position-control industrial manipulator. In particular, the main components of our platform are:

- 1 Denso VS060: Six-axis industrial manipulator.
- 1 Robotiq Gripper 2-Finger 85: Parallel adaptive gripper designed for industrial applications. Closure position, velocity and force can be controlled. The gripper opening goes from 0 to 85 mm. The grip force ranges from 30 to 100 N.
- 1 ATI Gamma Force-Torque (F/T) Sensor: It measures all six components of force and torque. They are calibrated with the following sensing ranges: $\mathbf{f} = [32, 32, 100]$ N and $\boldsymbol{\tau} = [2.5, 2.5, 2.5]$ Nm.
- 1 Ensenso 3D camera N35-802-16-BL

3.2.1 Single Pin Insertion

To illustrate the effectiveness of the framework, we first consider a dexterous task: a cylindrical peg-in-hole task. In this task, two rectangular boxes ($20 \times 50 \times 110$) mm are placed randomly on a known table. The first box *Obj 1*, also the static one, is clamped down on the table with a cylindrical pin ($r = 4$ mm, $l = 30$ mm) pre-inserted. The second box-*Obj 2*, considered as the movable one, will be grasped by the robot. Besides, the camera mounted on a fixed tri-pod will be used to localize the two above boxes (see Figure 3.1). As shown in the Figure, to perform its task, the robot arm needs to grasp *Obj 2* and insert it into *Obj 1*. After it completes the insertion, it will then release *Obj 2*. Note that the geometries of all objects are assumed to be known.

This is a good example for our framework illustration since most of the capabilities we have developed are required to perform this task. The task also constitutes one of the key steps in many assembly processes. In addition, it yields a fully quantifiable way to measure the performance of a robotic manipulation system. For these reasons, it is considered as a good test for the generalizability and simplicity of our approach.

Regarding the execution of the task, the process is naturally divided into three sequent sub-tasks as below:

- locate the two objects by using the camera;
- pick Obj 2;
- compliant insertion of the two objects.

Perception

A camera mounted on a fixed tri-pod is used to localize the two boxes. In this setting, we note that the transformation of an object relative to the robot base is given by

$${}_bT^o = {}_bT^c {}_cT^o. \quad (3.13)$$

This implies that the uncertainty of the robot positioning does not directly affect the uncertainty of the object pose with respect to the robot base. However, as mentioned in Section 3.1.2, the uncertainty of the robot positioning (if existing) also affects the estimation of the ${}_bT^c$ via the hand-eye calibration. Again, it is noted that we assume the uncertainty of our manipulator is negligible compared to the uncertainty of the pose estimation using our camera system. We follow the proposed framework in Section 3.1.2 to estimate the poses of the two objects and their associated uncertainties.

Pick Obj 2

Once the two objects are located, the robot arm needs to perform the picking task. In fact, this task resembles the planar grasping action which has been discussed earlier in Section 3.1.3.

To be specific, before grasping Obj 2, the robot arm moves its gripper to a pre-grasp position which is above the object. The gripper, then, gradually close its fingers until it exceeds the force limit. After the execution of the grasping action, we read the distance between the fingers from the Robotiq gripper encoder, and update the distribution of the object. In this step, the previous information obtained from the perception sub-task is used as the initial uncertainty region of Obj 2. Note that since we assume the table plane to be known, we then project Obj 2 uncertainty region onto this surface. Moreover, the pose of Obj 2 and its associated uncertainty will be presented in the gripper local frame as Obj 2 is now rigidly grasped by the gripper.

Compliant Pin Insertion

Next, pin insertion is the final sub-task to be executed. Due to the uncertainties on the position of the objects (Obj 1 and Obj 2), the exact position of the hole/pin is unknown. Moreover, given the parameters of the peg-in-hole set-up, we observe that the insertion will fail in case the position errors are more than 0.5 mm. To cope with such uncertainty, many researchers have proposed to apply force-controlled exploration. For example, [Suárez-Ruiz et al., 2018] used a super-imposed spiral search pattern. In their setting, the hole position is uncertain. The robot is controlled to follow the spiral pattern while maintaining the contact between the pin and the hole. Once the hole is found, the exploration will be immediately terminated. Even though this strategy is able to precisely and consistently detect holes and perform tight pin insertions (100% success across the 28 insertions), it is indeed a “blind searching” algorithm since there is no information about the uncertainties of the objects to be used. This limitation motivates us to study

how such uncertainties can be employed to design a better compliant insertion strategy as below.

After the gripper grasps Obj 2, the robot arm moves the Obj 2 to the pre-inserting position. It, then, moves down until it touches the surface of Obj 1. Once they are in contact, the relative transformation between the two objects is as follows:

$${}_{o1}\mathbf{T}^{o2} = {}_{o1}\mathbf{T}^b {}_b\mathbf{T}^e \mathbf{T}^{o2}, \quad (3.14)$$

where ${}_{o1}\mathbf{T}^b$ is the transformation of the base with respect to Obj 1 frame, ${}_e\mathbf{T}^{o2}$ is the relative transformation between Obj 2 and the end-effector (gripper). As the uncertainty of ${}_e\mathbf{T}^{o2}$ is represented by particles, we assume it to be a single peak Gaussian distribution and perform an empirical estimation to obtain its parameters⁴. The uncertainty of ${}_{o1}\mathbf{T}^{o2}$ is, then, obtained by the forward propagation method. The search pattern is now optimized based on such fine-gained knowledge of the uncertainty. Instead of using the “circular” spiral trajectory, we change the spiral pattern which follows the elliptical shape of the one-standard-deviation covariance ellipsoid of the ${}_{o1}\mathbf{T}^{o2}$ distribution (see Figure 3.1).

Compared to the traditional “circular” spiral search strategy, our new proposed spiral pattern can successfully perform the insertion at a faster speed. The Table 3.1 shows the experimental results of over 50 insertions for each method. As expected, we achieve 100% success across all the insertions. In addition, our method requires only 11.2 ± 4.5 seconds to insert successfully, which is about two times faster compared to the traditional search strategy.

⁴Such assumption is suitable for our experiment where object geometry is simple. For better capture of the multi-model of the distribution, the Gaussian mixture model is recommended.

	Circular spiral search	Elliptical spiral search
Time(s)	22.8 ± 5.7	11.2 ± 4.5

Table 3.1: We compare our elliptical spiral search with the traditional “circular” spiral search strategy.

Figure 3.4: We exploit the benefits of having access to uncertainties information in a double pins insertion task.

3.2.2 Double Pin Insertion

In previous experiment, we have studied *single* pin insertion task in which only the uncertainties in the translation parts of the object poses were employed to derive the new search strategy. We now study another example where both uncertainties of rotation and translation parts are used in planning the insertion strategy. In particular, we keep the experiment setting similar to last task except that both objects now have two pins/holes (see Figure 3.4).

In this experiment, we assume *Obj 2* is fully constrained and its pose is determined before the execution of the compliant double pin insertion. Note that this can be done by simply bringing the object to contact with a known surface in order to further reduce the uncertainty after the grasping action.

To perform the double pin insertion, we decouple the task into two force-controlled explorations. First, the robot arm moves *Obj 2* to the pre-inserting position, slightly tilts *Obj 2*, then starts moving down until contact is detected. We then perform the elliptical spiral search to find the precise position of the first pin/hole using the information of the uncertainty of the translation part. Second, we find the next pin/hole by simply rotating

Obj 2 around the first pin/hole axis while maintaining contact with the surface. In this case, the information of the uncertainty of the rotation part are employed to determine the bound angles of the exploration. As expected, we also achieve 100% success across 25 attempts. The average running time of the force-controlled exploration is 28.3 ± 7.1 seconds.

The video of the two experiments can be found at <https://youtu.be/rbo4QL40fqU>.

3.3 Conclusion

In this work, we have presented a probabilistic framework to precisely keep track of the uncertainties throughout the entire manipulation process. In order to do so, we decompose the manipulation task into two consecutive stages, namely perception and physical interactions. Each stage is associated with different sources and types of uncertainties, requiring different techniques. We discuss which representation of uncertainties is the most appropriate for each stage (e.g. as probability distributions in $SE(3)$ during perception, as weighted particles during physical interactions), how to convert from one representation to another, and how to initialize or update the uncertainties at each step of the process (camera calibration, image processing, pushing, grasping, etc.).

We have shown that precisely keeping track of the uncertainties in the system can significantly improve the speed and success rate of the manipulation. For example, as shown in our experiment on the single cylindrical peg-in-hole task, the spiral search operation was accelerated by two times. We also demonstrated how the information of uncertainties in the system can be applied into more complex task, i.e. double pin insertion. In fact, without such information, it would be extremely challenging to perform the mentioned tasks successfully.

In addition to the mentioned complex insertion tasks, we believe these techniques could find application in various manipulation tasks that require the knowledge of the

uncertainties of the object poses. One application is to estimate the success probability of a particular action. This estimation plays an important role as it enables the robot to improve the overall success rate of the task by not performing unnecessary motions. Another application is to plan the action that can actively reduce the uncertainties in the assembly process and simultaneously decrease the number of actions required. Moreover, future work can also inherit the proposed techniques to employ in bimanual manipulation tasks with higher complexity.

It is also worth-noting that the benefits of the precise information about the uncertainties inevitably comes with an expense. The reason is that it takes more time for the robot to reason and obtain such information from the measurements after each physical interaction. In fact, both planar grasping action and touch-based localization need to be taken into account when dealing with this challenge. Regarding the planar grasping action, [Zhou et al., 2017] have proposed a solution in which the forward motion model is approximated by kernel conditional density estimation (KCDE). Nevertheless, in order to properly address this problem, further investigation to improve the overall speed of the estimation process is required in future works.

Chapter 4

Uncertainty in Hand-eye Calibration

Hand-eye calibration, which consists in identifying the rigid-body transformation between a camera (eye) mounted on the robot end-effector and the end-effector (hand) itself, is a fundamental problem in robot vision. Mathematically, this problem can be formulated as: solve for $\mathbf{X}$ in $\mathbf{AX} = \mathbf{XB}$, where $\mathbf{X}$ is the unknown 4×4 hand-eye transformation matrix and $\mathbf{A}$ and $\mathbf{B}$ are known 4×4 transformation matrices (see details in Section 4.2. Starting from the late 1980's, a large amount of literature has been devoted to this problem, and a number of efficient methods have been developed, see e.g. [Shiu and Ahmad, 1989, Wang, 1992, Park and Martin, 1994, Horaud and Dornaika, 1995, Strobl and Hirzinger, 2006, Ackerman and Chirikjian, 2013].

In this Chapter, we are interested, not merely in solving for $\mathbf{X}$, but more comprehensively, in evaluating the *covariance* of $\mathbf{X}$ from those of $\mathbf{A}$ and $\mathbf{B}$, where $\mathbf{A}$ and $\mathbf{B}$ are now *randomly perturbed* transformation matrices. This fine-grained information is critical in high-precision robotics applications for several reasons.

The uncertainty of the object pose estimation comes from three main sources: (i) the uncertainty of the object pose estimation in the camera frame, (ii) the uncertainty of the hand-eye calibration, and (iii) the uncertainty of the robot end-effector positioning. In practice, source (ii) arguably contributes the most: for instance, a tiny orientation error of 0.05 degrees in the hand-eye calibration already implies an error of 0.6 mm in

object position if the latter is 700 mm away from the camera (typical viewing distance for commodity 3D cameras). In turn, having a precise knowledge of the uncertainty of the object pose estimation is critical:

- In high-precision manufacturing, it is important, not only to know the pose of an object, but also to *guarantee* that the pose estimation error is within some *tolerance*. For instance, when drilling holes in the fuselage of an aircraft, the hole position tolerance is 0.5 mm – which would be violated by an error of 0.05 degrees in the hand-eye calibration, even when assuming that the object pose estimation in the camera frame is perfect (see above);
- The precise knowledge of the object pose covariance *matrix* allows one to intelligently refine the object pose estimation by other perception modes. For instance, in visuo-tactile sensor fusion [Petrovskaya and Khatib, 2011b], knowing that the covariance of the object pose is comparatively large in the translation along, say, the X-axis will prompt us to touch the object along that axis in order to best reduce the uncertainty.

In addition, knowing the covariance of $\mathbf{X}$ allows improving the calibration process itself, by e.g. choosing the appropriate number of measurements to achieve a desired level of precision, or choosing the appropriate matrices $\mathbf{A}$ and $\mathbf{B}$ that minimize the covariance of $\mathbf{X}$.

The goal of this Chapter is to rigorously work out a derivation of the covariance of $\mathbf{X}$, which is ultimately the most generic and relevant quantification of the uncertainty of the hand-eye calibration process. Specifically, we transpose methods for forward and backward propagation of covariance [Hartley and Zisserman, 2003] into the framework of uncertainty in $SE(3)$. The structure of the hand-eye calibration equation raises specific technical difficulties, which we shall address in detail.

The remainder of the Chapter is organized as follows. Section 4.1 reviews the related literature to have a closer look at this problem and to understand why finding the covari-

ance of X is challenging. Section 4.2, we state the hand-eye calibration problem and introduce the mathematical background of the work, which includes the representation of uncertainty in $SE(3)$, and methods for forward and backward propagation of covariance. In Section 4.3, we present our method to estimate the rotation and translation parts of the hand-eye transformation matrix and their associated covariance matrices. In Section 4.4, we show that the method can indeed predict with excellent precision these covariances in synthetic and real calibration datasets. We also use this information to compute the covariance of the object pose estimation in a real setting. Finally, in Section 4.5, we conclude by discussing the advantages and drawbacks of our approach.

This work has been accepted to publish in **Huy Nguyen** and Quang-Cuong Pham (2018), *On the covariance of X in $AX = XB$* , in *IEEE Transactions on Robotics* 34.6, pp. 1651–1657, DOI: [10.1109/TRO.2018.2861905](https://doi.org/10.1109/TRO.2018.2861905).

4.1 Related Works

4.1.1 Solutions to the Hand-eye Calibration

Commonly, this problem can be formulated as: solve for X in $AX = XB$, where X is the unknown 4×4 hand-eye transformation matrix and A and B are known 4×4 transformation matrices. In the case that there is no error in the measurements, $AX = XB$ has a one-parameter family of solutions (as first shown by [Shiu and Ahmad, 1989]). Two pairs of (A_i, B_i) satisfying certain constraints are required in order to obtain a unique solution. Unfortunately, in actual sensor calibration applications, noises usually exist in the measured values of A and B . Therefore, a more practical approach is to make several measurements $(A_1, B_1), (A_2, B_2), \dots, (A_k, B_k)$, and to find an X that is a best fit to the k matrix equations.

Early solutions decoupled the equation into the rotational and translational parts and optimize for rotation first, then translation. Depending on the techniques used to

represent rotational parts, these approaches can be categorized as follows: angle-axis representation [Shiu and Ahmad, 1989, Tsai and Lenz, 1989], quaternions representation [Chou and Kamel, 1991, Zhuang and Roth, 1996] and the Euclidean Group [Park and Martin, 1994]. These methods often yield uncomplex, fast, but error-prone formulations, since rotation estimation errors propagate to the translational part. [Chen, 1991] for the first time does not decompose the equation into rotational and translational terms by using the screw motion approach. [Dornaika and Horaud, 1998] applied nonlinear optimization to simultaneously solve the rotational and the translational parts. They formulated rotational parts with quaternions. [Daniilidis, 1999] used dual quaternions to represent the screw. [Schmidt et al., 2003] further discussed the selection of movement pairs for increased numerical stability and the elimination of outliers. [Strobl and Hirzinger, 2006] presented a calibration method for eye-in-hand systems in terms of a parametrization of a stochastic model. They proposed a metric on the group of the rigid transformations $SE(3)$ and the corresponding error model for nonlinear optimization.

4.1.2 Uncertainties in the Hand-eye Calibration

Finding the covariance of $\mathbf{X}$ is challenging for several reasons. First, as $\mathbf{X}$, $\mathbf{A}$ and $\mathbf{B}$ represent rigid-body transformations, they live in $SE(3)$, a subset of the space of 4×4 matrices endowed with a non-trivial Lie group structure [Murray et al., 1994b]. Second, how to represent and calculate uncertainties in $SE(3)$ is by itself a complex issue, which has prompted advanced mathematical developments (see Section 2.1). Finally, merely solving for $\mathbf{X}$ in $\mathbf{AX} = \mathbf{XB}$ is already a difficult problem, let alone evaluating the uncertainty of the solution.

There are a number of works dealing with the uncertainty of hand-eye calibration. In [Tsai and Lenz, 1989], based on a sensitivity analysis of closed-form solutions, some critical factors and criteria influencing the accuracy of the result are analyzed. For instance, one may try to maximize the angle between rotation axes of relative movement

Figure 4.1: Depend on the system setting, a camera can be mounted on the end-effector of the robot or on a fixed tri-pod. In both cases, the relative transformation of the camera w.r.t the end-effector (or w.r.t the robot base) can be formulated as $\mathbf{AX} = \mathbf{XB}$ equations.

to reduce the influence on error in rotation, or to minimize the distance between the optical center of the camera and the calibration pattern to reduce the influence on error in translation. Based on this analysis, [Shi et al., 2005] present a algorithm to select movement pairs automatically from a series of measurements to reduce the error of the estimate. [Schmidt and Niemann, 2008] also introduce similar approach based on a vector quantization method. [Aron et al., 2004] present an error estimation method of the rotation part of $\mathbf{X}$ based on an Euler angles parameterization. The authors do not discuss how that error propagates to the translation part of $\mathbf{X}$ and their vision tracking measurements are also assumed to be noise-free. More fundamentally, the Euler angles formulation, as opposed to the $\mathbf{SE}(3)$ formulation, is well-known to involve singularities.

In [Ackerman and Chirikjian, 2013, Ackerman et al., 2014, Ma et al., 2016], authors introduced the concept of using mean and covariance of distributions on $\mathbf{SE}(3)$ to formulate the hand-eye calibration problem. An information-theoretic approach is proposed in [Ackerman et al., 2014] by viewing the problem in terms of distributions on the group $\mathbf{SE}(3)$, and minimizing the Kullback-Leibler divergence of these distributions with re-

spect to the unknown $\mathbf{X}$. This allows them to handle the problem without a priori knowledge of the correspondence of the given data sets. In [Ma et al., 2016] the algorithm is further improved by introducing two new probabilistic approaches built on top of the Batch method. Although the idea of applying probabilistic theory into the hand-eye calibration problem is novel, our focus here is on the derivation of the covariance of the hand-eye transformation, which is different from the mentioned works as they only focus on finding a solution to $\mathbf{X}$.

The idea of estimating explicitly uncertainties in the system is also by no means new. While many have studied the problem of uncertainty in the camera model (intrinsic and extrinsic parameters) [Hartley and Zisserman, 2003] and the propagation of uncertainties through the camera model [Barfoot and Furgale, 2014], we place emphasis on the hand-eye transformation and its uncertainty in this work.

It can be noted that none of the aforementioned works has provided a derivation of the covariance of $\mathbf{X}$, which is ultimately the most generic and relevant quantification of the uncertainty of the hand-eye calibration process. To address this problem, we present in Chapter 4 a novel algorithm to rigorously work out such a derivation. In particular, our work will exploit the benefits of applying optimization techniques directly on $SE(3)$ to obtain the derivation of the covariance of the hand-eye transformation. Specific technical difficulties arising from the structure of the hand-eye calibration equation shall be addressed in detail.

It can be noted that none of the aforementioned works has provided a derivation of the covariance of $\mathbf{X}$, which is ultimately the most generic and relevant quantification of the uncertainty of the hand-eye calibration process. The goal of this chapter is to rigorously work out such a derivation. In particular, our work will exploit the benefits of applying optimization techniques directly on $SE(3)$ to obtain the derivation of the covariance of the hand-eye transformation. The structure of the hand-eye calibration equation raises specific technical difficulties, which we shall address in detail.

It also worths noting that, in other cases where the camera is not mounted on the end-effector but on a fixed tri-pod/stand, finding the relative transformation between the camera and the robot base can also be formulated as the $\mathbf{AX} = \mathbf{XB}$ problem and can be treated by the same method (see Figure 4.1).

4.2 Background

4.2.1 Formulation of the Hand-eye Calibration Problem

The classical hand-eye calibration method consists in looking at a fixed pattern from two different viewpoints, say 1 and 2, giving rise to the following equation

$${}_b\mathbf{T}_1^e {}_e\mathbf{T}^c {}_c\mathbf{T}_1^o = {}_b\mathbf{T}^o = {}_b\mathbf{T}_2^e {}_e\mathbf{T}^c {}_c\mathbf{T}_2^o, \quad (4.1)$$

where

- ${}_b\mathbf{T}_i^e$ is the transformation of the end-effector with respect to the fixed robot base at configuration i ;
- ${}_e\mathbf{T}^c$ is the constant transformation of the camera with respect to the end-effector;
- ${}_c\mathbf{T}_i^o$ is the transformation of the pattern (object) with respect to the camera at configuration i ;
- ${}_b\mathbf{T}^o$ is the constant transformation of the pattern with respect to the robot base (see Fig. 4.2).

Next, one can transform the above equation into

$${}_b\mathbf{T}_2^{e^{-1}} {}_b\mathbf{T}_1^e {}_e\mathbf{T}^c = {}_e\mathbf{T}^c {}_c\mathbf{T}_2^o {}_c\mathbf{T}_1^{o^{-1}}, \quad (4.2)$$

which has the form of $\mathbf{AX} = \mathbf{XB}$, where $\mathbf{X} := {}_e\mathbf{T}^c$ is the unknown hand-eye transformation, and $\mathbf{A} := {}_b\mathbf{T}_2^{e-1} {}_b\mathbf{T}_1^e$ and $\mathbf{B} := {}_c\mathbf{T}_{2c}^o {}_1^o$ can be computed from respectively the robot kinematics and pattern pose estimation [Park and Martin, 1994]. Next, if the fixed pattern is viewed from a large number of viewpoints, one can collect many different $\mathbf{A}$'s and $\mathbf{B}$'s. Suppose that we have a set of k measurements $(\mathbf{A}_1, \mathbf{B}_1), (\mathbf{A}_2, \mathbf{B}_2), \dots, (\mathbf{A}_k, \mathbf{B}_k)$. Since in practice these measurements are perturbed by actuator/sensor noise, the exact solution for the set of k equations $\mathbf{A}_i\mathbf{X} = \mathbf{XB}_i$ will not exist. Instead, the problem is commonly framed as an optimization problem in which $\mathbf{X}$ is found as the transformation that “best” fits the k equalities.

Figure 4.2: The hand-eye calibration problem consists in identifying the rigid-body transformation ${}_e\mathbf{T}^c$ between a camera mounted on the end-effector of a robot and the end-effector itself.

In some cases, the hand-eye calibration problem is also formulated as the “ $\mathbf{A}'\mathbf{X} = \mathbf{ZB}'$ problem” where $\mathbf{A}' := {}_b\mathbf{T}_j^e$, $\mathbf{B}' := {}_c\mathbf{T}_j^{o-1}$ and $\mathbf{Z} := {}_b\mathbf{T}^o$. The solution of $\mathbf{X}$ in this formulation has been proposed by many works, e.g. [Strobl and Hirzinger, 2006, Dornaika and Horaud, 1998]. However, we only focus on the $\mathbf{AX} = \mathbf{XB}$ formulation in

this work.

In other cases, the camera is not mounted on the end-effector but on a fixed stand. In these cases, finding the relative transformation between the camera and the robot base can also be formulated as the $\mathbf{AX} = \mathbf{XB}$ problem and can be treated by the same method.

4.2.2 Representation of Rigid-body Transformations and of their Uncertainties

We first recall some facts and formulae involving the rotation group $\mathbf{SO}(3)$ and the special Euclidean group $\mathbf{SE}(3)$ [Murray et al., 1994b]. Elements of $\mathbf{SO}(3)$ are given by the 3×3 real matrices $\mathbf{R}$ satisfying $\mathbf{R}^\top \mathbf{R} = \mathbb{I}$ and $\det \mathbf{R} = 1$. The group $\mathbf{SE}(3)$ is defined to be

$$\mathbf{SE}(3) = \left\{ \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \middle| \mathbf{R} \in \mathbf{SO}(3), \mathbf{t} \in \mathbb{R}^3 \right\}. \quad (4.3)$$

$\mathbf{SO}(3)$ is a matrix Lie group, and its associated Lie algebra, denoted $\mathfrak{so}(3)$, is given by the set of 3×3 real skew-symmetric matrices. The $[\]$ operator turns $\boldsymbol{\phi} \in \mathbb{R}^3$ into a 3×3 member of the Lie algebra $\mathfrak{so}(3)$:

$$[\boldsymbol{\phi}] := \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix}. \quad (4.4)$$

We will use $\vee$ as the inverse operation of $[\]$. By using the matrix exponential $\exp()$ on elements of $\mathfrak{so}(3)$, we can obtain group elements of $\mathbf{SO}(3)$ as $\mathbf{R} = \exp([\boldsymbol{\phi}])$. The inverse of the exponential, or logarithm, is used to obtain the vector of exponential coordinates $\boldsymbol{\phi}$ from the group element $\mathbf{R} \in \mathbf{SO}(3)$ as $\boldsymbol{\phi} = (\log \mathbf{R})^\vee$.

To model the uncertainty on $\mathbf{SE}(3)$, we follow the approach proposed in [Barfoot and Furgale, 2014]. Since there is in general no bi-invariant distance on $\mathbf{SE}(3)$ [Park

and Ravani, 1997]¹, finding the rotation and translation components of $\mathbf{X}$ simultaneously would require a non-trivial rotation/translation weighting in any cases. Instead, we choose to solve them separately, which entails a number of simplifications [Park and Martin, 1994]. As a consequence, the uncertainties of the rotation and the translation parts are also modeled separately.

Specifically, we assume that the rotation parts of the observations $\mathbf{A}_i$ and $\mathbf{B}_i$ are corrupted as follows

$$\mathbf{R}_{A_i} = \exp([\boldsymbol{\xi}_{R_{A_i}}])\bar{\mathbf{R}}_{A_i}, \quad \mathbf{R}_{B_i} = \exp([\boldsymbol{\xi}_{R_{B_i}}])\bar{\mathbf{R}}_{B_i}, \quad (4.5)$$

where $\bar{\mathbf{R}}_{A_i}, \bar{\mathbf{R}}_{B_i} \in SO(3)$ are the means of $\mathbf{R}_{A_i}, \mathbf{R}_{B_i}$, and $\boldsymbol{\xi}_{R_{A_i}}, \boldsymbol{\xi}_{R_{B_i}} \in \mathbb{R}^3$ are zero-mean Gaussian perturbations with covariance matrices $\Sigma_{R_{A_i}}, \Sigma_{R_{B_i}}$, respectively.

The translation parts of the $\mathbf{A}_i$ and $\mathbf{B}_i$ are corrupted as follows

$$\mathbf{t}_{A_i} = \boldsymbol{\xi}_{t_{A_i}} + \bar{\mathbf{t}}_{A_i}, \quad \mathbf{t}_{B_i} = \boldsymbol{\xi}_{t_{B_i}} + \bar{\mathbf{t}}_{B_i}, \quad (4.6)$$

where $\bar{\mathbf{t}}_{A_i}, \bar{\mathbf{t}}_{B_i} \in \mathbb{R}^3$ are the means of $\mathbf{t}_{A_i}, \mathbf{t}_{B_i}$, and $\boldsymbol{\xi}_{t_{A_i}}, \boldsymbol{\xi}_{t_{B_i}} \in \mathbb{R}^3$ are zero-mean Gaussian perturbations with covariance matrices $\Sigma_{t_{A_i}}, \Sigma_{t_{B_i}}$, respectively.

Note that the above assumptions imply that rotation and translation noises are independent, which on some occasions (e.g. manipulator with large deflections on the robot's links) might require further considerations. Because of this assumptions, the space we consider is, strictly speaking, $SO(3) \times \mathbf{R}^3$, rather than $SE(3)$.

¹We remark that any distance metric on $SE(3)$ fundamentally involves the combination of rotational and translational units, and there is no natural way to do this. In this sense, any distance metric on $SE(3)$ will be *ad hoc* to some extent; this will be captured by the choice of length scale for physical space (see [Park, 1995] for further discussion of these issues).

4.2.3 Forward and Backward Propagation of Covariance

Forward propagation. Let $\mathbf{P}$ be a random vector in $\mathbb{R}^M$ with mean $\bar{\mathbf{P}}$ and covariance matrix Σ . Consider a function $\mathbf{f} : \mathbb{R}^M \rightarrow \mathbb{R}^N$ that is differentiable in a neighbourhood of $\bar{\mathbf{P}}$. Then, at the first order of approximation, $\mathbf{f}(\mathbf{P})$ is a random variable with mean $\mathbf{f}(\bar{\mathbf{P}})$ and covariance matrix

$$\Sigma_f = \mathcal{J}\Sigma\mathcal{J}^\top, \quad (4.7)$$

where $\mathcal{J}$ is the Jacobian matrix of $\mathbf{f}$ at $\bar{\mathbf{P}}$.

Backward propagation. Assume now that $\mathbf{P}$ (the parameter) is unknown, but that $\mathbf{V} := \mathbf{f}(\mathbf{P})$ (the measurement) is known and determined to be a random variable with mean $\bar{\mathbf{V}}$ and covariance matrix Σ_V . Then the best estimate for $\mathbf{P}$ is given by

$$\mathbf{P}^* := \min_{\mathbf{P}} \|\mathbf{V} - \mathbf{f}(\mathbf{P})\|_{\Sigma_V}.$$

To estimate the covariance of $\mathbf{P}$, one can approximate $\mathbf{f}$ by an affine function $\mathbf{f}(\mathbf{P}) = \mathbf{f}(\bar{\mathbf{P}}) + \mathcal{J}(\mathbf{P} - \bar{\mathbf{P}})$, which yields

$$\|\mathbf{V} - \mathbf{f}(\mathbf{P})\|_{\Sigma_V} = \|(\mathbf{V} - \bar{\mathbf{V}}) - \mathcal{J}(\mathbf{P} - \bar{\mathbf{P}})\|_{\Sigma_V}. \quad (4.8)$$

Using the weighted pseudo-inverse, one has

$$\mathbf{P}^* - \bar{\mathbf{P}} = (\mathcal{J}^\top \Sigma_V^{-1} \mathcal{J})^{-1} \mathcal{J}^\top \Sigma_V^{-1} (\mathbf{V} - \bar{\mathbf{V}}). \quad (4.9)$$

From (4.7), the covariance of $\mathbf{P}$ can now be approximated at the first order by

$$\begin{aligned}
\Sigma^* &= [(\mathcal{J}^\top \Sigma_V^{-1} \mathcal{J})^{-1} \mathcal{J}^\top \Sigma_V^{-1}] \Sigma_V \\
&\quad [(\mathcal{J}^\top \Sigma_V^{-1} \mathcal{J})^{-1} \mathcal{J}^\top \Sigma_V^{-1}]^\top \\
&= (\mathcal{J}^\top \Sigma_V^{-1} \mathcal{J})^{-1}.
\end{aligned} \tag{4.10}$$

In practice, when performing an iterative least-squares optimization, one can use (4.10) at the last iteration to obtain the estimation of the covariance of $\mathbf{P}$.

Note that the quality of the approximations given by Equations (4.7) and (4.10) depends in particular on the quality of the linear approximation of $\mathbf{f}$.

4.3 Derivation of the Covariance of $\mathbf{X}$

Equation $\mathbf{A}_i \mathbf{X} = \mathbf{X} \mathbf{B}_i$ can be decomposed as

$$\mathbf{R}_{Ai} \mathbf{R} = \mathbf{R} \mathbf{R}_{Bi}, \tag{4.11}$$

$$\mathbf{R}_{Ai} \mathbf{t} + \mathbf{t}_{Ai} = \mathbf{R} \mathbf{t}_{Bi} + \mathbf{t}, \tag{4.12}$$

where $\mathbf{R}, \mathbf{t}$ denote respectively the rotation and translation parts of $\mathbf{X}$.

4.3.1 Covariance of the Rotation Part of $\mathbf{X}$

We first consider the rotation part $\mathbf{R}$ of $\mathbf{X}$. Let $[\alpha_i], [\beta_i] \in \mathfrak{so}(3)$ denote the logarithms of $\mathbf{R}_{Ai}$ and $\mathbf{R}_{Bi}$ respectively, i.e.

$$[\alpha_i] := \log \mathbf{R}_{Ai}, \quad [\beta_i] := \log \mathbf{R}_{Bi}. \tag{4.13}$$

Note that the covariance matrices of α_i and β_i can be obtained by applying the forward propagation of covariance

$$\Sigma_{\alpha_i} = \mathbf{J}(\alpha_i)^{-1} \Sigma_{R_{Ai}} \mathbf{J}(\alpha_i)^{-1\top}, \quad (4.14)$$

$$\Sigma_{\beta_i} = \mathbf{J}(\beta_i)^{-1} \Sigma_{R_{Bi}} \mathbf{J}(\beta_i)^{-1\top}, \quad (4.15)$$

where $\mathbf{J}(\alpha_i)$ denotes the (left) Jacobian of $SO(3)$ at α_i , see [Barfoot and Furgale, 2014] for more details.

Next, via logarithm mapping, equation (4.11) can be written as

$$\log R_{Ai} = \log R R_{Bi} R^\top = R[\beta_i] R^\top. \quad (4.16)$$

Applying the rule $R[\omega] R^\top = [R\omega]$ for $R \in SO(3)$ and $[\omega] \in \mathfrak{so}(3)$, one has

$$\alpha_i = R\beta_i. \quad (4.17)$$

In order to use the uncertainty model in $SO(3)$, we define a random variable ξ_R that represents the difference between R and the current estimate $\hat{R}$ by

$$R = \exp([\xi_R]) \hat{R}.$$

Next, to apply the backward propagation of covariance, one needs the measurement vectors α_i and β_i to appear on the same side of the equation. To achieve this without making it too complex, we use a trick from [Hartley and Zisserman, 2003], which consists

in “copying” the β_i ’s on both sides, as follows

$$\underbrace{\begin{pmatrix} \beta_1 \\ \alpha_1 \\ \vdots \\ \beta_k \\ \alpha_k \end{pmatrix}}_V = \underbrace{\begin{pmatrix} \beta_1 \\ \exp([\xi_R])\hat{R}\beta_1 \\ \vdots \\ \beta_k \\ \exp([\xi_R])\hat{R}\beta_k \end{pmatrix}}_{f(P)}. \quad (4.18)$$

Now, the measurement vector is given by $V := (V_1, \dots, V_k)$, where $V_i := (\beta_i, \alpha_i)$, and the parameter vector is given by $P := (\xi_R, \beta_1, \dots, \beta_k)$.

Since the noise of α_i ’s and β_i ’s are independent (α is caused by robot kinematics while β_i is caused by object pose estimation in the camera frame), the covariance matrix of the measurement vector is given by

$$\Sigma_V := \text{diag}(\Sigma_{V_1}, \Sigma_{V_2}, \dots, \Sigma_{V_k}), \quad (4.19)$$

with $\Sigma_{V_i} := \text{diag}(\Sigma_{\beta_i}, \Sigma_{\alpha_i})$.

Now, the covariance-weighted minimization is given by

$$\min_P \|V - f(P)\| = \min_P \sum_i^k (V_i - f(P)_i)^\top \Sigma_{V_i}^{-1} (V_i - f(P)_i).$$

This minimization problem can be solved by iteratively updating the estimate of the parameter vector by the rules

$$\hat{R}_{(j+1)} = \exp([\xi_R])\hat{R}_{(j)}, \quad (4.20)$$

$$\hat{\beta}_{i(j+1)} = \hat{\beta}_{i(j)} + \delta_{\beta_i}, \quad (4.21)$$

where at each step (j) the update vector $\delta := (\xi_R, \delta_{\beta_1}, \dots, \delta_{\beta_k})$ is found by solving the

normal equation

$$\mathcal{J}_f^\top \Sigma_V^{-1} \mathcal{J}_f \delta = \mathcal{J}_f^\top \Sigma_V^{-1} (V - f(\mathbf{P})). \quad (4.22)$$

The Jacobian of f has the form

$$\begin{aligned} \mathcal{J}_f &= [\mathcal{J}^{\xi_R} | \mathcal{J}^\beta] \\ &= \left[\begin{array}{c|ccc} \mathcal{J}_1^{\xi_R} & \mathcal{J}_1^\beta & & \\ \mathcal{J}_2^{\xi_R} & & \mathcal{J}_2^\beta & \\ \vdots & & & \ddots \\ \mathcal{J}_k^{\xi_R} & & & \mathcal{J}_k^\beta \end{array} \right], \end{aligned} \quad (4.23)$$

$$\text{where } \mathcal{J}_i^{\xi_R} := \begin{bmatrix} \mathbf{0} \\ -[\hat{\mathbf{R}} \hat{\boldsymbol{\beta}}_i] \end{bmatrix}, \mathcal{J}_i^\beta := \begin{bmatrix} \mathbb{I} \\ \hat{\mathbf{R}} \end{bmatrix}. \quad (4.24)$$

The set of equations (4.22) may now written in block form as

$$\begin{aligned} &\begin{bmatrix} \mathcal{J}^{\xi_R \top} \Sigma_V^{-1} \mathcal{J}^{\xi_R} & \mathcal{J}^{\xi_R \top} \Sigma_V^{-1} \mathcal{J}^\beta \\ \mathcal{J}^{\beta \top} \Sigma_V^{-1} \mathcal{J}^{\xi_R} & \mathcal{J}^{\beta \top} \Sigma_V^{-1} \mathcal{J}^\beta \end{bmatrix} \begin{pmatrix} \boldsymbol{\xi}_R \\ \boldsymbol{\delta}_\beta \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{J}^{\xi_R \top} \Sigma_V^{-1} (V - f(\mathbf{P})) \\ \mathcal{J}^{\beta \top} \Sigma_V^{-1} (V - f(\mathbf{P})) \end{pmatrix}. \end{aligned} \quad (4.25)$$

To simplify the left-hand side of (4.25), let

$$\mathbf{U} := \mathcal{J}^{\xi_R \top} \Sigma_V^{-1} \mathcal{J}^{\xi_R} = \sum_i^k \mathcal{J}_i^{\xi_R \top} \Sigma_{V_i}^{-1} \mathcal{J}_i^{\xi_R}, \quad (4.26)$$

$$\mathbf{W} := \mathcal{J}^{\xi_R \top} \Sigma_V^{-1} \mathcal{J}^\beta = [\mathbf{W}_1, \dots, \mathbf{W}_k], \quad (4.27)$$

$$\mathbf{Z} := \mathcal{J}^{\beta \top} \Sigma_V^{-1} \mathcal{J}^\beta = \text{diag}(\mathbf{Z}_1, \dots, \mathbf{Z}_k), \quad (4.28)$$

$$\text{where } \mathbf{W}_i := \mathcal{J}_i^{\xi_R \top} \Sigma_{V_i}^{-1} \mathcal{J}_i^\beta \text{ and } \mathbf{Z}_i := \mathcal{J}_i^{\beta \top} \Sigma_{V_i}^{-1} \mathcal{J}_i^\beta. \quad (4.29)$$

As for the right-hand side of (4.25), let

$$\begin{aligned}\boldsymbol{\epsilon}_{\xi_R} &:= \mathcal{J}^{\xi_R \top} \Sigma_V^{-1} (V - f(\mathbf{P})) \\ &= \sum_i^k \mathcal{J}_i^{\xi_R \top} \Sigma_{V_i}^{-1} (V_i - f(\mathbf{P})_i),\end{aligned}\tag{4.30}$$

$$\boldsymbol{\epsilon}_\beta := \mathcal{J}^{\beta \top} \Sigma_V^{-1} (V - f(\mathbf{P})) = (\boldsymbol{\epsilon}_{\beta_1}, \dots, \boldsymbol{\epsilon}_{\beta_k}),\tag{4.31}$$

$$\text{where } \boldsymbol{\epsilon}_{\beta_i} := \mathcal{J}_i^{\beta \top} \Sigma_{V_i}^{-1} (V_i - f(\mathbf{P})_i).\tag{4.32}$$

To solve equations (4.25), one can left-multiply both sides by $\begin{bmatrix} \mathbb{I} & \mathbf{W}\mathbf{Z}^{-1} \\ \mathbf{0} & \mathbb{I} \end{bmatrix}$, which yields

$$(\mathbf{U} - \mathbf{W}\mathbf{Z}^{-1}\mathbf{W}^\top)\boldsymbol{\xi}_R = \boldsymbol{\epsilon}_{\xi_R} - \mathbf{W}\mathbf{Z}^{-1}\boldsymbol{\epsilon}_\beta,\tag{4.33}$$

$$\mathbf{Z}\boldsymbol{\delta}_\beta = \boldsymbol{\epsilon}_\beta - \mathbf{W}^\top\boldsymbol{\xi}_R.\tag{4.34}$$

The above equations can now be solved to find the updating vectors $\boldsymbol{\xi}_R$ and $\boldsymbol{\delta}_\beta$.

Applying backward propagation of covariance, a first-order approximation of the covariance of $\mathbf{P}$ is given the following matrix, taken at the last iteration,

$$\begin{aligned}\Sigma^* &= (\mathcal{J}_f^\top \Sigma_V^{-1} \mathcal{J}_f)^{-1} \\ &= \begin{bmatrix} \Sigma_R & -\Sigma_R \mathbf{W}\mathbf{Z}^{-1} \\ -(\Sigma_R \mathbf{W}\mathbf{Z}^{-1})^\top & (\mathbf{W}\mathbf{Z}^{-1})^\top \Sigma_R \mathbf{W}\mathbf{Z}^{-1} + \mathbf{Z}^{-1} \end{bmatrix}.\end{aligned}$$

Truncation of the covariance matrix Σ^* gives covariance matrices for parameters $\boldsymbol{\xi}_R$ and $\boldsymbol{\beta}_i^*$ separately. The covariance of $\boldsymbol{\xi}_R$ is given by the top-left block of Σ^* , that is:

$$\Sigma_R = (\mathbf{U} - \mathbf{W}\mathbf{Z}^{-1}\mathbf{W}^\top)^{-1} = (\mathbf{U} - \sum_i^k \mathbf{W}_i \mathbf{Z}_i^{-1} \mathbf{W}_i^\top)^{-1}.\tag{4.35}$$

The covariance of $\boldsymbol{\beta}_i^*$ is given by:

$$\Sigma_{\beta_i^*} = (\mathbf{W}_i \mathbf{Z}_i^{-1})^\top \Sigma_R \mathbf{W}_i \mathbf{Z}_i^{-1} + \mathbf{Z}_i^{-1}.\tag{4.36}$$

The cross-covariances of ξ_R and β_i^* are $\Sigma_{R\beta_i^*} = -\Sigma_R W_i Z_i^{-1}$.

4.3.2 Covariance of the Translation Part of X

We now consider the translation part t of X . Let $q_i := R t_{B_i} - t_{A_i}$. Equations (4.12) can be written as

$$q_i = (R_{A_i} - \mathbb{I})t. \quad (4.37)$$

Note that the covariance matrices of q_i can be approximated by applying the forward propagation of covariance

$$\Sigma_{q_i} = \Sigma_{t_A} + R^* \Sigma_{t_B} R^{*\top} + [R^* t_{B_i}] \Sigma_R [R^* t_{B_i}]^\top, \quad (4.38)$$

where R^* is the optimal rotation found in the previous section, and Σ_R is the corresponding covariance.

We then update the α_i by substituting the optimal values of R and β_i found in last section into equation (4.17). The covariance matrices of the corresponding $R_{A_i}^*$ can be approximated by forward propagation of covariance

$$\begin{aligned} \Sigma_{R_{A_i}^*} = & J(\alpha_i^*) \{ [R^* \beta_i^*] \Sigma_R [R^* \beta_i^*]^\top - R^* \Sigma_{R\beta_i^*}^\top [R^* \beta_i^*]^\top \\ & - [R^* \beta_i^*] \Sigma_{R\beta_i^*} R^{*\top} + R^* \Sigma_{\beta_i^*} R^{*\top} \} J(\alpha_i^*)^\top. \end{aligned} \quad (4.39)$$

Applying the same trick as previously, we “copy” the $R_{A_i}^*$ ’s on both sides of the equation,

as follows

$$\underbrace{\begin{pmatrix} \mathbf{R}_{A_1}^* \\ \mathbf{q}_1 \\ \vdots \\ \mathbf{R}_{A_k}^* \\ \mathbf{q}_k \end{pmatrix}}_{\mathbf{V}} = \underbrace{\begin{pmatrix} \mathbf{R}_{A_1}^* \\ (\mathbf{R}_{A_1}^* - \mathbb{I})\hat{\mathbf{t}} \\ \vdots \\ \mathbf{R}_{A_k}^* \\ (\mathbf{R}_{A_k}^* - \mathbb{I})\hat{\mathbf{t}} \end{pmatrix}}_{\mathbf{f}(\mathbf{P})}. \quad (4.40)$$

Now the measurement vector is given by $\mathbf{V} := (\mathbf{V}_1, \dots, \mathbf{V}_k)$, where $\mathbf{V}_i := (\mathbf{R}_{A_i}^*, \mathbf{q}_i)$, and the parameter vector is given by $\mathbf{P} := (\mathbf{t}, \mathbf{R}_{A_1}^*, \dots, \mathbf{R}_{A_k}^*)$.

The covariance matrix of the measurement vector is given by

$$\Sigma_{\mathbf{V}} := \text{diag}(\Sigma_{\mathbf{V}_1}, \Sigma_{\mathbf{V}_2}, \dots, \Sigma_{\mathbf{V}_k}), \quad (4.41)$$

where

$$\Sigma_{\mathbf{V}_i} := \begin{bmatrix} \Sigma_{\mathbf{R}_{A_i}^*} & \Sigma_{\mathbf{R}_{A_i}^* \mathbf{q}_i} \\ \Sigma_{\mathbf{q}_i \mathbf{R}_{A_i}^*} & \Sigma_{\mathbf{q}_i} \end{bmatrix}. \quad (4.42)$$

The cross-covariance matrices $\Sigma_{\mathbf{q} \mathbf{R}_{A_i}^*}$ can be estimated by applying forward propagation of covariance:

$$\Sigma_{\mathbf{q} \mathbf{R}_{A_i}^*} = -[\mathbf{R}^*, \mathbf{t}_A] \Sigma_{\mathbf{R} \mathbf{R}_{A_i}^*} \quad (4.43)$$

$$= [\mathbf{R}^*, \mathbf{t}_A] (\Sigma_{\mathbf{R}} [\mathbf{R}^* \boldsymbol{\beta}_i^*]^\top - \Sigma_{\mathbf{R} \boldsymbol{\beta}_i^*} \mathbf{R}^{*\top}) \mathbf{J}(\alpha_i^*)^\top. \quad (4.44)$$

Now, the covariance-weighted minimization is given by

$$\min_{\mathbf{P}} \|\mathbf{V} - \mathbf{f}(\mathbf{P})\| = \min_{\mathbf{P}} \sum_i^k (\mathbf{V}_i - \mathbf{f}(\mathbf{P})_i)^\top \Sigma_{\mathbf{V}_i}^{-1} (\mathbf{V}_i - \mathbf{f}(\mathbf{P})_i).$$

We solve this by iteratively updating the estimate of the parameter vector by the rules

$$\hat{\mathbf{t}}_{(j+1)} = \hat{\mathbf{t}}_{(j)} + \boldsymbol{\delta}_t, \quad (4.45)$$

$$\hat{\mathbf{R}}_{A_i(j+1)}^* = \exp([\boldsymbol{\xi}_{R_{A_i}}])\hat{\mathbf{R}}_{A_i(j)}^*, \quad (4.46)$$

where at each step (j) the update vector $\boldsymbol{\delta} := (\boldsymbol{\delta}_t, \boldsymbol{\xi}_{R_A}) = (\boldsymbol{\delta}_t, \boldsymbol{\xi}_{R_{A_1}}, \dots, \boldsymbol{\xi}_{R_{A_k}})$ is found by solving the normal equation

$$\mathcal{J}_f^\top \Sigma_V^{-1} \mathcal{J}_f \boldsymbol{\delta} = \mathcal{J}_f^\top \Sigma_V^{-1} (\mathbf{V} - \mathbf{f}(\mathbf{P})). \quad (4.47)$$

The Jacobian matrix has the form

$$\mathcal{J}_f = \begin{bmatrix} \mathcal{J}^t | \mathcal{J}^{\xi_{R_A}} \end{bmatrix}, \quad (4.48)$$

$$\text{where } \mathcal{J}^t := (\mathcal{J}_1^t, \dots, \mathcal{J}_k^t), \quad (4.49)$$

$$\mathcal{J}^{\xi_{R_A}} := \text{diag}(\mathcal{J}_1^{\xi_{R_A}}, \dots, \mathcal{J}_k^{\xi_{R_A}}), \quad (4.50)$$

$$\mathcal{J}_i^t := \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{R}}_{A_i}^* - \mathbb{I} \end{bmatrix}, \quad (4.51)$$

$$\mathcal{J}_i^{\xi_{R_A}} := \begin{bmatrix} \mathbb{I} \\ -[\hat{\mathbf{R}}_{A_i}^* \hat{\mathbf{t}}_X] \end{bmatrix}. \quad (4.52)$$

For the rest of the derivation, we following the same procedure as previously derived. One thus can obtain the update vectors from

$$(\mathbf{U} - \mathbf{W}\mathbf{Z}^{-1}\mathbf{W}^\top)\boldsymbol{\delta}_t = \boldsymbol{\epsilon}_t - \mathbf{W}\mathbf{Z}^{-1}\boldsymbol{\epsilon}_{\xi_{R_A}}, \quad (4.53)$$

$$\mathbf{Z}\boldsymbol{\xi}_{R_A} = \boldsymbol{\epsilon}_{\xi_{R_A}} - \mathbf{W}^\top \boldsymbol{\delta}_t, \quad (4.54)$$

$$\text{where } \mathbf{U} := \sum_i^k \mathcal{J}_i^{\top} \Sigma_{V_i}^{-1} \mathcal{J}_i, \quad (4.55)$$

$$\mathbf{W} := [\mathbf{W}_1, \dots, \mathbf{W}_k] \text{ with } \mathbf{W}_i := \mathcal{J}_i^{\top} \Sigma_{V_i}^{-1} \mathcal{J}_i^{\xi_{RA}}, \quad (4.56)$$

$$\mathbf{Z} := \text{diag}(\mathbf{Z}_1, \dots, \mathbf{Z}_k) \text{ with } \mathbf{Z}_i := \mathcal{J}_i^{\xi_{RA}^{\top}} \Sigma_{V_i}^{-1} \mathcal{J}_i^{\xi_{RA}}, \quad (4.57)$$

$$\boldsymbol{\epsilon}_t := \sum_i^k \mathcal{J}_i^{\top} \Sigma_{V_i}^{-1} (\mathbf{V}_i - \mathbf{f}(\mathbf{P})_i), \quad (4.58)$$

$$\boldsymbol{\epsilon}_{\xi_{RA}} := (\boldsymbol{\epsilon}_{\xi_{RA_1}}, \dots, \boldsymbol{\epsilon}_{\xi_{RA_k}}), \quad (4.59)$$

$$\text{with } \boldsymbol{\epsilon}_{\xi_{RA_i}} = \mathcal{J}_i^{\xi_{RA}^{\top}} \Sigma_{V_i}^{-1} (\mathbf{V}_i - \mathbf{f}(\mathbf{P})_i). \quad (4.60)$$

At the last iteration, a first-order approximation of the covariance matrix of $\mathbf{t}$ is given by

$$\Sigma_t = (\mathbf{U} - \mathbf{W} \mathbf{Z}^{-1} \mathbf{W}^{\top})^{-1} = (\mathbf{U} - \sum_i^k \mathbf{W}_i \mathbf{Z}_i^{-1} \mathbf{W}_i^{\top})^{-1}. \quad (4.61)$$

4.4 Experiments

We now validate the proposed method in predicting the covariance of the hand-eye transformation using synthetic and real calibration data. Using the covariance of $\mathbf{X}$, we are then in a position to compute the covariance of the object pose estimation in a real setting. Our implementation is open-source and is available at <https://bit.ly/2LwUVjh>.

4.4.1 Synthetic Calibration Data

To generate synthetic data, we start by selecting a random transformation matrix $\bar{\mathbf{X}} = (\bar{\mathbf{R}}, \bar{\mathbf{t}})$, which serves as the true hand-eye transformation. We then generate $M = 250$ dataset, each dataset comprising $k = 30$ corrupted pairs $(\mathbf{A}_i, \mathbf{B}_i)_{i \in [1, k]}$. Each corrupted pair is generated as follows. First, we generate a random *uncorrupted* pair $(\bar{\mathbf{A}}_i, \bar{\mathbf{B}}_i)$, which are drawn from a simulation model of the system in Fig. 4.2 and verifies $\bar{\mathbf{A}}_i \bar{\mathbf{X}} = \bar{\mathbf{X}} \bar{\mathbf{B}}_i$ exactly. Next, we add noise to $\bar{\mathbf{A}}_i$ and $\bar{\mathbf{B}}_i$ as explained in Section 4.2.2. The covariance

matrices of the noise are chosen arbitrarily as

$$\Sigma_{\mathbf{R}_{A_i}} := \lambda \text{diag}(5, 2, 3), \quad (4.62)$$

$$\Sigma_{\mathbf{R}_{B_i}} := \lambda \text{diag}(7, 2, 8), \quad (4.63)$$

$$\Sigma_{\mathbf{t}_{A_i}} := \lambda \text{diag}(0.1, 0.2, 0.5), \quad (4.64)$$

$$\Sigma_{\mathbf{t}_{B_i}} := \lambda \text{diag}(0.7, 0.8, 0.1), \quad (4.65)$$

where $\lambda \in \mathbb{R}$ is a scaling parameter that allows us to change the magnitude of the uncertainties.

At each noise level λ , we evaluate the covariance of $\mathbf{X}$ following two methods

- *Our method:* For some dataset $m \in [1, M]$, we compute the covariance matrices $\Sigma_{\mathbf{R}_m}, \Sigma_{\mathbf{t}_m}$ using the proposed method (Predicted). In fact, $\Sigma_{\mathbf{R}_m}, \Sigma_{\mathbf{t}_m}$ are slightly different across the M datasets. This reflects the influence of the data distribution on the estimation of the hand-eye transformation. However, since the differences are very small compared to the standard deviations, we use the averages for our comparison. Also note that the accuracy analyses performed here only focus on the covariance matrices since our experiments show that the standard deviations are always much larger than the biases.;
- *Empirical Estimation:* For each dataset $m \in [1, M]$, we find the rotation and translation $\hat{\mathbf{R}}_m, \hat{\mathbf{t}}_m$ that optimally fit the k hand-eye equations following the proposed method. We then compute the covariance matrices $\Sigma_{\mathbf{R}}^{\text{ES}}, \Sigma_{\mathbf{t}}^{\text{ES}}$ across the M datasets as

$$\Sigma_{\mathbf{R}}^{\text{ES}} := \frac{1}{M} \sum_m^M \boldsymbol{\xi}_{\mathbf{R}_m} \boldsymbol{\xi}_{\mathbf{R}_m}^{\top}, \quad (4.66)$$

$$\Sigma_{\mathbf{t}}^{\text{ES}} := \frac{1}{M} \sum_m^M \boldsymbol{\xi}_{\mathbf{t}_m} \boldsymbol{\xi}_{\mathbf{t}_m}^{\top}, \quad (4.67)$$

Figure 4.3: Projection of the one-standard-deviation covariance ellipsoids on pairs of axes for Σ_R and Σ_t , shown for Empirical Estimation and our method on synthetic data when $\lambda = 10^{-4}$, where $\xi_{R_x}, \xi_{R_y}, \xi_{R_z}$ are errors in rotation around X, Y, Z-axis and $\xi_{t_x}, \xi_{t_y}, \xi_{t_z}$ are errors in translation along X, Y, Z-axis.

Figure 4.4: Difference between the covariances obtained from Empirical Estimation and predicted by our method, at different noise levels.

where $\xi_{R_m} := (\log(\hat{R}_m \bar{R}^{-1}))^\vee$, $\xi_{t_m} := \hat{t}_m - \bar{t}$.²

Fig. 4.3 shows projections of the covariance ellipsoids on pairs of axes for the Empirical Estimation and our method, when $\lambda = 10^{-4}$. It can be noted that the proposed method provides an excellent estimation of the covariance of the hand-eye transformation.

²Since we sample directly on the Lie algebra $\mathfrak{so}(3)$, computing the sampled means and covariances does not require to involve the group Jacobian (see [Wang and Chirikjian, 2008, Kwon et al., 2007] for further discussion on sampling means and covariances on $\mathbf{SO}(3), \mathbf{SE}(3)$).

To gauge the performance at different noise levels, we use the following metrics

$$\varepsilon = \frac{\sqrt{\text{tr}((\Sigma^{\text{PR}} - \Sigma^{\text{ES}})^\top (\Sigma^{\text{PR}} - \Sigma^{\text{ES}}))}}{\sqrt{\text{tr}(\Sigma^{\text{ES}^\top} \Sigma^{\text{ES}})}}. \quad (4.68)$$

Fig 4.4 shows that our algorithm can cope well with increasing magnitudes of the measurement uncertainty. The estimation errors remain low overall, and increases slightly with the magnitude of the noise, since larger noise levels increase the number of local minima at each iteration. Regarding the computation cost, our method is naturally several magnitude faster than the Empirical Estimation.

It is also worth noting that the closed form solution in [Park and Martin, 1994] always yields slightly higher covariance compared to our method. This is because our method does optimally minimize the uncertainty of the estimated transformation by taking to account the measurement noise.

4.4.2 Real Calibration Data

We now validate the proposed method on actual calibration data obtained from our robot system, which consists of a 3D camera mounted on a 6-DOF industrial manipulator, as shown in Fig. 4.2.

Covariances of A and B in the Actual System

We first need to estimate the covariances of the A_i 's and B_i 's in our system, so that we can give them as inputs to our method.

As our industrial manipulator has a very high accuracy compared to that of our camera system³, we assume that the noise on the A_i 's is negligible.

³Using a well-calibrated tooltip and a calibrating plate with well-manufactured ground truth points, our experiments show that the tooltip can achieve up to 0.3mm in error of the translation part. This can be considered as negligible compared to the uncertainty of the pose estimation using our camera system.

Regarding the $\mathbf{B}_i$'s, the $\mathbf{B}_1, \dots, \mathbf{B}_k$ are assumed to have the same noise distributions: $\forall i, \Sigma_{\mathbf{R}_{\mathbf{B}_i}} = \Sigma_{\mathbf{R}_{\mathbf{B}}}, \Sigma_{\mathbf{t}_{\mathbf{B}_i}} = \Sigma_{\mathbf{t}_{\mathbf{B}}}$. We experimentally collect randomly 500 pairs of $\mathbf{A}_i$ and $\mathbf{B}_i$ from our system, with the only restriction that distances from pattern to camera are from 700 mm to 1100 mm. Next, we generate $M = 300$ datasets, each dataset comprising $k = 30$ pairs $(\mathbf{A}_i, \mathbf{B}_i)$ randomly selected from the collected pairs.

The rotation and translation errors of $\mathbf{B}_i$'s are then computed as

$$\xi_{\mathbf{R}_{\mathbf{B}_i}} = (\log(\mathbf{R}_{\mathbf{B}_i} \bar{\mathbf{R}}_{\mathbf{B}_i}^{-1}))^\vee, \quad \xi_{\mathbf{t}_{\mathbf{B}_i}} = \mathbf{t}_{\mathbf{B}_i} - \bar{\mathbf{t}}_{\mathbf{B}_i}, \quad (4.69)$$

where the ground truth is $\bar{\mathbf{B}}_i = \bar{\mathbf{X}}^{-1} \mathbf{A}_i \bar{\mathbf{X}}$. Note that we assumed rotation and translation noises are independent.

Since the true transformation $\bar{\mathbf{X}}$ is unknown in the real system, we use

$$\mathbf{R}^{\text{avg}} = \exp \left(\left[\frac{1}{M} \sum_m^M (\log \hat{\mathbf{R}}_m)^\vee \right] \right), \quad \mathbf{t}^{\text{avg}} = \frac{1}{M} \sum_m^M \hat{\mathbf{t}}_m, \quad (4.70)$$

as the ground truth⁴. Estimating $\hat{\mathbf{R}}_m, \hat{\mathbf{t}}_m$ using our method would require information of $\mathbf{A}_i$'s, $\mathbf{B}_i$'s noise, hence we use [Park and Martin, 1994] instead.

Then the empirical covariance matrices of $\mathbf{B}_i$ can be estimated similarly to Equa-

⁴Note that there are other ways to calculate the mean $\mathbf{R}_{\text{mean}}$ of a rotation distribution (e.g. by the condition $\sum_m^M \log(\mathbf{R}_{\text{mean}}^\top \mathbf{R}_m) = \mathbf{0}$ [Wang and Chirikjian, 2008]). However, as we noticed that the rotations of $\mathbf{X}$ found in our experiment are only distributed around the identity, equation (4.70) is used to simplify the calculation.

tions (4.66, 4.67):

$$\Sigma_{\mathbf{R}_{B_i}} = \begin{bmatrix} 5.7 & -2.3 & -2.8 \\ -2.3 & 18.1 & 3.8 \\ -2.8 & 3.8 & 1.8 \end{bmatrix} 10^{-6} rad^2, \quad (4.71)$$

$$\Sigma_{\mathbf{t}_{B_i}} = \begin{bmatrix} 13.1 & 0.7 & 0.9 \\ 0.7 & 2.1 & -0.3 \\ 0.9 & -0.3 & 0.8 \end{bmatrix} 10^{-6} mm^2. \quad (4.72)$$

Validation

To validate our method, we collect another 500 pairs of $\mathbf{A}_i$ and $\mathbf{B}_i$ from our system. We constrain the robot motion so that it covers the same area as that used for determining the noise on $\mathbf{B}_i$. We then generate $M = 300$ datasets, each dataset comprising $k = 30$ pairs $(\mathbf{A}_i, \mathbf{B}_i)$ randomly selected from the collected pairs. The covariance matrices are computed from these datasets using the Empirical Estimation and our method, in the same manner as previously.

Fig. 4.5 provides projections of the covariance ellipsoids on pairs of axes shown for two methods. We see that the proposed method delivers a good estimation of the covariances. We do not believe there has been another methods of estimating uncertainty of the hand-eye transformation. Moreover, the proposed method is also relatively easy to replicate and use in practical applications.

Assessing a Possible Systematic Bias

Note that the study of the covariance would not be meaningful if there exists a systematic bias in the estimation of $\mathbf{X}$. However, direct evaluation of the systematic bias is difficult – most papers on hand-eye calibration indeed do not contain such a direct evaluation.

Figure 4.5: One-standard-deviation covariance ellipsoids of the hand-eye transformation Σ_R and Σ_t , shown for Empirical Estimation and our algorithm with real data, axes' labels are similar as to Fig. 4.3.

We, therefore, focus on the following indirect evaluation.

Consider first the average X^{avg} computed across the M datasets. We put the object to a fixed position, i.e. ${}_bT^o$ is constant. We then observed the object from 300 robot poses, yielding 300 pairs of ${}_bT_i^e, {}_cT_i^o$. If there is no systematic bias in the estimation of X^{avg} and no uncertainty in ${}_cT_i^o$, then we shall have ${}_bT_i^e X^{\text{avg}} {}_cT_i^o$ constant for all i (they are all equal to ${}_bT^o$). Thus, the “standard deviation” of ${}_bT_i^e X^{\text{avg}} {}_cT_i^o$ is a possible indirect measure of the systematic bias.

In our setting, we found that the “standard deviation” of the translation part of ${}_bT_i^e X^{\text{avg}} {}_cT_i^o$ was 0.7 mm. Since this also contains the uncertainty of ${}_cT_i^o$, the systematic bias must be quite small, compared to the translation part of the covariance of X , which was found to be around 2 mm. As another element of comparison, in case one considers the X resulting from a random dataset m , then the “standard deviation” of the translation part of ${}_bT_i^e X {}_cT_i^o$ would be around 2 mm.

4.4.3 Covariance of the Object Pose Estimation

Using the covariance of $\mathbf{X}$ previously obtained, we are now in a position to predict the covariance of the object pose estimation, which is our ultimate goal. Here, we demonstrate the propagation of uncertainties to the object pose estimation using the same robot system as previously (see Fig. 4.2).

Recall that the constant transformation of the pattern (object) with respect to the robot base is given by

$$\mathbf{Y} := {}_b\mathbf{T}^o = {}_b\mathbf{T}_i^e {}_e\mathbf{T}^c {}_c\mathbf{T}_i^o. \quad (4.73)$$

The covariances of ${}_b\mathbf{T}_i^e$ are assumed to be negligible, and the covariance of ${}_e\mathbf{T}^c (= \mathbf{X})$ can be estimated by using the procedure proposed in Section 4.4.2. Thus, to predict the mean and the covariance of $\mathbf{Y}$, one needs now to estimate the covariances of ${}_c\mathbf{T}_i^o$.

Suppose that ${}_c\mathbf{T}_1^o, \dots, {}_c\mathbf{T}_k^o$ have the same noise distribution, i.e. $\forall i, \Sigma_{\mathbf{R}_{cT^o i}} = \Sigma_{\mathbf{R}_{co}}, \Sigma_{t_{cT^o i}} = \Sigma_{t_{co}}$. We begin by experimentally collecting 500 pairs of ${}_b\mathbf{T}_i^e, {}_c\mathbf{T}_i^o$ from our system. The rotation and translation errors of ${}_c\mathbf{T}_i^o$'s are then computed similar to (4.69), where the ground truth is ${}_c\bar{\mathbf{T}}_i^o = (\bar{\mathbf{Y}}^{-1} {}_b\bar{\mathbf{T}}_i^e \bar{\mathbf{X}})^{-1}$. As discussed earlier, we will use (4.70) instead of the true value of $\bar{\mathbf{X}}$. Regarding $\bar{\mathbf{Y}}$, one can transform (4.73) into

$${}_b\mathbf{T}_{pb}^e {}_q\mathbf{T}^{e-1} \mathbf{Y} = \mathbf{Y} {}_c\mathbf{T}_p^{o-1} {}_c\mathbf{T}_q^o, p \neq q, \quad (4.74)$$

which has the form of $\mathbf{A}'_i \mathbf{Y} = \mathbf{Y} \mathbf{B}'_i$, where $\mathbf{A}' := {}_b\mathbf{T}_{pb}^e {}_q\mathbf{T}^{e-1}$ and $\mathbf{B}' := {}_c\mathbf{T}_p^{o-1} {}_c\mathbf{T}_q^o$. Hence, $\bar{\mathbf{Y}}$ can also be computed in the same manner as computing $\bar{\mathbf{X}}$.

We now collect 500 pairs of ${}_b\mathbf{T}_i^e, {}_c\mathbf{T}_i^o$ from our system. We then generate $M = 300$ datasets, each dataset comprising $k = 30$ $(\mathbf{A}_i, \mathbf{B}_i)$ pairs computed from $({}_b\mathbf{T}_i^e, {}_c\mathbf{T}_i^o)$ pairs randomly selected from the 500 collected pairs.

Next, we evaluate the covariance of $\mathbf{Y}$ following two methods

- *Our method:* For some dataset $m \in [1, M]$, we compute the covariance matrices

Figure 4.6: One-standard-deviation covariance ellipsoids of the object pose (w.r.t the robot base) Σ_{R_Y} and Σ_{t_Y} , shown for Empirical Estimation and our algorithm, axes' labels are similar as to Fig. 4.3. The distances from the pattern to the camera are from 700 mm to 1100 mm.

$\Sigma_{R_{Y_m}}, \Sigma_{t_{Y_m}}$ using the propagation method described in Chapter 3, Section 3.1.2 (PRedicted). Since the $\Sigma_{R_{Y_m}}, \Sigma_{t_{Y_m}}$ are only slightly different across the M datasets, we use the averages for our comparison;

- *Empirical Estimation*: For each dataset $m \in [1, M]$, we compute $Y_m = {}_bT_m^o = {}_b\hat{T}_e^e {}_e\hat{T}_m^c {}_c\hat{T}^o$, where ${}_b\hat{T}_e^e$ and ${}_c\hat{T}^o$ are randomly selected from the collected pairs; the rotation and translation of ${}_e\hat{T}_m^c$ together with their covariances are computed using our method as proposed in Section 4.3 (see equations (4.36, 4.61)). Next, the covariances are computed similarly to Equations (4.66, 4.67).

Fig. 4.6 shows the one-standard-deviation covariance ellipsoids shown for two methods. One can see that our prediction matches very well the covariances from the Empirical Estimation.

In absolute values, the covariance of the hand-eye calibration compounds with that

of the object pose estimation in the camera frame, resulting in a relatively large overall covariance for the object pose estimation in the robot frame, around 2 mm in standard deviation. This again emphasizes the need of having access to the covariance of the hand-eye transformation. This fine-grained information tells us how confident we can be regarding the object pose estimation and shall also enable us to design new perception algorithms and methods for reaching higher precision, by e.g. visuo-tactile sensor fusion.

4.5 Conclusion

In this Chapter, we have presented a rigorous derivation of the covariance of the solution $\mathbf{X}$, when $\mathbf{A}$ and $\mathbf{B}$ are randomly perturbed matrices. Our approach consists in transposing methods for forward and backward propagation of covariance into the framework of uncertainty in $SE(3)$. Experiments involving synthetic and real calibration data show that our approach can predict the covariance of the hand-eye transformation with excellent precision. In addition, we also presented a propagation experiment to find the uncertainty of the object pose.

While these estimates could also be provided by the Empirical Estimation, such a method would require collecting a large number of samples, which is not practical. Furthermore, the Empirical Estimation yields no insights into how the uncertainties on the measurements of $\mathbf{A}$ and $\mathbf{B}$ propagate to the uncertainty of the hand-eye transformation. By contrast, in our method, by analyzing critical factors influencing the covariance of $\mathbf{X}$, for instance, based on the formulae (4.36) and (4.61), one may be able to refine the calibration process to achieve a higher precision, by e.g. determining the appropriate number of sample viewpoints or choosing their optimal distribution, which is the object of our future research.

Chapter 5

Improvements to Touch-based Estimation

Touch-based object localization is an important component of autonomous robotic systems that are to perform dexterous tasks in real-world environments. Yet, localization by visual sensors alone might not provide a sufficient precision for many dexterous grasping or manipulation tasks (see Chapter 4, Section 4.4 for the detailed experiments). Consider for instance the assembly a chair [Suarez-Ruiz and Pham, 2016], where one sub-task consists in inserting wooden pins into the holes on a wooden stick. While object localization by commercial 3D cameras can provide at best 1-2 mm precision, the tightness of the insertion task here requires sub-millimeter precision.

One principled approach to address this problem consists in refining the pose estimate by *physically interacting* with the object: the robot would touch the object of interest (without moving it) at multiple positions, see Fig. 5.1. Contact positions and normals are recorded by forward kinematics (some specialized tactile sensors allow detecting the surface normal at contact [Petrovskaya and Khatib, 2011a, Hebert et al., 2013]). The estimation of the object pose from a given a number of such measurements is called the *tactile (or touch-based) localization problem*. Starting from the 1980's, a large amount of literature has been devoted to this problem and efficient methods have been developed,

see Chapter 2, Section 5.1.1.

One major difficulty with these approaches is that, when the target object is placed within a cluttered environment, the robot might touch a different object and/or obstacles instead of the target object, generating thereby an *outlier measurement*. Existing methods are inherently fragile with respect to such outliers.

Here we address the problem of outliers classification by transposing the well-known RANdom SAmple Consensus (RANSAC) method into a Bayesian estimation framework (see Chapter 3, Section 3.1.3 for more details on how we applied the Bayesian filter to solve the touch-based localization problem). The algorithm consists in a series of hypothesize-and-verify iterations to select the “best” set of measurements. As these iterations involve (computationally intensive) Bayesian updates, it is crucial to improve these updates in order to achieve practical running times. Our second contribution is therefore a fast method to find the most probable object face that corresponds to a given touch measurement, which yields a significant acceleration of the Bayesian updating step.

The remainder of the Chapter is organized as follows. Section 5.1 reviews the related literature and provides the necessary mathematical background. Section 5.2 and 5.3 present our contributions in detail. Section 5.4 reports experimental results, which show that our overall algorithm provides accurate localization in practical times, even when the measurements are corrupted by outliers. Finally, Section 5.5 concludes and sketches some future research directions.

Figure 5.1: We study the object localization problem via touch. The photo shows the robot interacting with a wooden stick using a pin and its force/torque sensor.

5.1 Related Works and Problem Setting

5.1.1 Related Works

Over the past years, there has been a surge of interest in applying Bayesian methods in touch-based localization [Gadeyne and Bruyninckx, 2001, Chhatpar and Branicky, 2005, Corcoran and Platt, 2010, Petrovskaya and Khatib, 2011a]. These methods calculate the probability distribution over all possible states, which captures the uncertainty resulting from noisy sensors, inaccurate object models, and other effects present during the sensing process. Over the last decades, these algorithms have been applied to tackle a variety of problems related to object pose estimation with great success [Thrun et al., 2005]. A majority of existing works have been devoted to reducing the computational complexity of problem, which scales exponentially with the number of DOFs and the size of the initial uncertainty region. Hence, they have been using to perform object localization and produce useful information for improving measurements collection, and real-world exploration.

Among the early works in this field, the touch-based localization problem was addressed by [Gadeyne and Bruyninckx, 2001], who performed 3-DOF localization of a box

with initial uncertainty of 300 mm in position and 360 degrees in orientation. To obtain measurements, they used a force-controlled robot and a sampled measurement model. [Chhatpar and Branicky, 2005] used particle filters to achieve object localization in peg-in-hole assembly tasks. Their measurement model was constructed by sampling the object in advance. [Corcoran and Platt, 2010] used an annealed particle filter to estimate a 4-DOF pose and radius of cylindrical objects. The work extended the analytical measurement model proposed by [Petrovskaya et al., 2006] to include negative information. Due to the computational complexity, most approaches have to restrict the number of DOFs and/or initial uncertainty. To deal with this problem, many researchers have come up with a number of approaches which enable high DOFs localization with large initial uncertainty in a timely fashion. For instance, in [Petrovskaya and Khatib, 2011a], authors introduced the Scaling Series method, which achieved 6-DOF localization with large initial uncertainty of 400mm in position and 360 degrees in orientation. Their measurements were collected by making repeated contact with the object using a single end-effector. [Vezzani et al., 2017] proposed the Memory Unscented Particle Filter that combines a modified particle filter and the unscented Kalman filter.

In these works, very often, measurements are obtained through a data collection procedure where the robot’s end effector, equipped with a tactile or a force/torque sensor, approaches the object from several different directions. Though these actions can be generated randomly [Petrovskaya and Khatib, 2011a, Chhatpar and Branicky, 2005] or be chosen to maximize the expected information gain [Hebert et al., 2013, Javdani et al., 2013], there is no guarantee that the set of measurements does not contain *extreme erroneous* measurements, or outliers, which may result from sensor failures or the presence of other objects in the environments. These outliers, however, will shift the distribution of the object states far from the correct state, leading to a significant loss in localization precision.

To mitigate the effect of outlier measurements, one can try to determine whether the

received measurement is an outlier. Subsequently, the updating step of the filtering is only performed on a relevant subset of the data. The idea of solving this correspondence problem by classifying measurements into inliers and outliers is not new. There have been many important works on 6-DOF object localization by the computer vision community [Raguram et al., 2008, Papazov and Burschka, 2010]. However, to our knowledge, 6-DOF touch-based estimation in cluttered environments has not been addressed in prior art.

5.1.2 Bayesian estimation

We start out with a quick summary of the problem: one needs to determine the pose $X \in \text{SE}(3)$ of an object O of known shape based on a set of tactile measurements $\mathbf{y}$. The object is typically represented as a polygonal mesh. The measurements $\mathbf{y} = \mathbf{y}_0, \dots, \mathbf{y}_n$ are obtained by touching the object with the robot's end effector. Each measurement $\mathbf{y}_k := (\mathbf{y}_k^{\text{pos}}, \mathbf{y}_k^{\text{nor}})$ consists of the acquired contact position $\mathbf{y}_k^{\text{pos}}$ and contact normal $\mathbf{y}_k^{\text{nor}}$.

Note that we consider here the case when the measurement data sets fully constrain the problem. In other words, we assume enough data has been collected in order to sufficiently disambiguate the object pose.

Hereafter, the tactile localization problem is cast into the Bayesian framework and addressed as a nonlinear filtering problem.

The uncertain knowledge of the object is represented by a probability distribution. The object to be located is assumed to be static during the measurement collection. This assumption is commonly made [Chhatpar and Branicky, 2005, Petrovskaya and Khatib, 2011a] and is realistic: for instance, the object is heavy or is fixed on a support preventing possible movements, or the contact is very slight. Hence, starting with $P(X_0)$ – the *prior* distribution over the state X – the goal is to recursively updating the following

conditional probability

$$P(X_{t+1}|\mathbf{y}) = \eta P(\mathbf{y}|X_t)P(X_t). \quad (5.1)$$

Here $P(X_{t+1}|\mathbf{y})$ is known as the *posterior*, which represent our uncertain belief about the state $\mathbf{X}$ after having incorporated the measurement $\mathbf{y}$. On the right-hand side, the first factor $P(\mathbf{y}|X_t)$ is the *total measurement probability*, which encodes the likelihood of the measurement given the state (*measurement model*). The second factor $P(X_t)$ is the *prior*, which represents our belief about $\mathbf{X}$ before obtaining the measurements $\mathbf{y}$. The factor η is a normalizing factor independent of the state X_t and needs not be computed.

As mentioned before, many variants of particle filters have been proposed and proven to well suit the nonlinear and multi-modal nature of the problem. This work builds upon these algorithms and provides an automated method to deal with outlier measurements. To illustrate its performance, we apply our method to the Scaling Series algorithm [Petrovskaya and Khatib, 2011a], which is able to solve the 6-DOF localization problem efficiently and reliably. The main idea of the Scaling Series approach is to combine Bayesian Monte-Carlo and annealing techniques. It performs multiple iterations over the data, gradually scaling precision from low to high. The number of particles at each iteration is automatically selected on the basis of the complexity of the annealed posterior.

5.1.3 The measurement model

The total measurement probability is commonly computed based on the *proximity measurement model*, where the measurements are considered independent of each other and where both the position and normal components are corrupted by Gaussian noise. For each measurement, the probability is computed based on the distance between the measurement and the object. This model was first introduced by the authors [Petrovskaya and Khatib, 2011a] and became popular in the literature owing to its computational efficiency.

Assume that the target object is represented as a polygonal mesh, containing a set of

faces $\mathcal{F}_i$ and their corresponding normal vectors $\mathbf{n}_i$. Suppose that the object is at state $\mathbf{X}$, then the distance between a measurement $\mathbf{y}_k$ and the face i of the object is defined by

$$d(\mathbf{y}_k, \mathcal{F}_i^X) := \sqrt{\frac{d(\mathbf{y}_k^{\text{pos}}, \mathcal{F}_i^X)^2}{\sigma_{\text{pos}}^2} + \frac{d(\mathbf{y}_k^{\text{nor}}, \mathbf{n}_i^X)^2}{\sigma_{\text{nor}}^2}}, \quad (5.2)$$

where $d(\mathbf{y}_k^{\text{pos}}, \mathcal{F}_i^X)$ is the shortest Euclidean distance from $\mathbf{y}_k^{\text{pos}}$ to any point on the face $\mathcal{F}_i^X$, $d(\mathbf{y}_k^{\text{nor}}, \mathbf{n}_i^X)$ is the usual angle between two 3D vectors, and $\sigma_{\text{pos}}, \sigma_{\text{nor}}$ are the Gaussian noise variances of the position and normal measurement components respectively. Next, the distance between the measurement $\mathbf{y}_k$ and the object is defined as

$$d(\mathbf{y}_k, \mathcal{O}^X) := \min_i d(\mathbf{y}_k, \mathcal{F}_i^X). \quad (5.3)$$

For the whole set of measurements $\mathbf{y}$, the total measurement error is defined as

$$u(\mathbf{y}, \mathbf{X}) := \sum_k d(\mathbf{y}_k, \mathcal{O}^X)^2. \quad (5.4)$$

Finally, the total measurement probability can be computed as follows

$$P(\mathbf{y}|\mathbf{X}_t) = \eta_{\mathbf{y}} \exp\left(-\frac{1}{2}u(\mathbf{y}, \mathbf{X}_t)^2\right), \quad (5.5)$$

where $\eta_{\mathbf{y}}$ is a constant and will be taken into account during the normalization.

Notice that the model described here is analytical. Hence it can be computed efficiently on the fly and without the need for prior training. The model assumes that the closest point on the object caused the measurement. This is often referred to as a “hard assignment” meaning that the point causing the measurement is assigned to be the closest point. Alternatively, one can consider the contribution from all points to the probability of the measurement [Corcoran and Platt, 2010]. Though such an approach might be more informative, it is much more computationally intensive.

5.2 Accelerating Bayesian Updates by Efficient Face Selection

5.2.1 Outline of the Algorithm

We note that, to compute the likelihood of a measurement, one needs to look for the face that is the most likely to cause the contact and normal measurement, i.e. that minimizes the distance $d(\mathbf{y}_k, \mathcal{F}_i^X)$. Hence the running time of the updating step depends linearly on the number of faces in the mesh model. A key to improve the speed of the updating step then consists in accelerating the face selection process.

Here we propose to do so by pruning out faces based on a pre-computed offline angle dictionary as follows.

Offline stage: We compute the angles α_i between a reference vector $\mathbf{n}_{\text{ref}}$ (e.g. the z-axis in the object reference frame) and the normal vectors $\mathbf{n}_i$ of all faces $\mathcal{F}_i$. The faces are then sorted according to the value of this angle.

Measurement evaluation stage: We first compute the angle α_{y_k} between the measurement normal and the reference unit vector. Then, a binary search is used to find, in the list of the α_i 's, the two angles α_L and α_R that best approximate α_{y_k} from below and from above, respectively. Next, a face $\mathcal{F}_i$ is added to the subset for evaluation if its associated angle α_i satisfies following condition

$$\alpha_L - \delta_\alpha < \alpha_i < \alpha_R + \delta_\alpha, \quad (5.6)$$

where δ_α is a problem-specific threshold. Fig. 5.2 illustrates the face selection algorithm on a partial sphere mesh. In this case, the number of faces considered in the measurement likelihood evaluation has been reduced down to the number of faces in a ring-like region (shown in red). Finally, we compute the distance $d(\mathbf{y}_k, \mathcal{F}_i)$ for all the faces in the subset

and choose the face with the lowest distance as representative of the object in (5.3).

Figure 5.2: An illustration of the algorithm on a partial sphere mesh. The number of faces need to be considered has reduced to the number of faces on a ring-like region (shown in red).

5.2.2 Algorithm Parameters

One can see that if δ_α is too large, the algorithm will be too conservative and select a larger number of faces than needed, resulting in a longer running time. However, if δ_α is too small, the algorithm might not be able to return a good subset of faces for evaluation. A reasonable choice is to set $\delta_\alpha = \sigma_{\text{nor}}$ in order to prune out all faces whose normals are farther than about one-standard-deviation from the measured normal.

Another factor that affects the performance of the algorithm is the choice of the reference vector $\mathbf{n}_{\text{ref}}$. A good choice for $\mathbf{n}_{\text{ref}}$ would induce an “even” distribution of the α_i , which can be quantified by the Shannon entropy as follows. The range $[0, \pi]$ is divided into N equal segments. Then the α_i are grouped into N bins depending on their values.

The Shannon entropy of the distribution is then given by

$$S := - \sum_{c=1}^N p_c \log(p_c), \text{ where } p_c := \frac{\#(\text{bin } c)}{\#\text{faces}}. \quad (5.7)$$

Figure 5.3: An illustration of the computation of the Shannon entropy for two different reference vectors on the mesh model of the back of a chair.

Fig. 5.3 illustrates the computation of the Shannon entropy for two different reference vectors on the mesh model of the back of a chair.

Finally, to choose the best reference vector, we sample random unit reference vectors, compute the Shannon entropy they induce, and choose the one with the highest Shannon entropy.

The efficient face selection technique described above can be applied equally well in many Bayesian estimations where normal and contact point measurements are available. The main idea is that the normals are discriminative and that the calculations of distances between normals are very fast, as compared to the calculations of the distances between points and faces.

5.3 Outlier Classification Using RANSAC

5.3.1 Outline of the Algorithm

The main challenges of the object localization in cluttered environments are: (i) the correspondance problem (whether the measurement belongs to the object), and (ii) computational complexity. In particular, in cluttered environments, the measurement set is usually contaminated by extreme erroneous measurements, or outliers, which may result from sensor failure or the presence of other objects in the environments. When a particle filter receives an outlier measurement, the weight update will shift most of the weights to a few particles that are far from the correct state. This would lead to a significant loss in localization precision. Thus, to mitigate the effect of outlier measurements, one can try to determine via statistical methods whether the received measurement is an outlier or not. Then the updating step of the filtering is only performed on a relevant set of data (i.e. the dataset containing only inliers).

Note that we consider here the case when the measurement data sets fully constrain the problem. In other words, we assume enough data has been collected in order to sufficiently disambiguate the object pose.

Here we adapt the popular Random Sample Consensus (RANSAC) to simultaneously classify the data into inliers (points consistent with the relation) and outliers (points not consistent with the relation)[Fischler and Bolles, 1981]. The algorithm consists of a series of hypothesize-and-verify steps and is presented in pseudo-code in Alg.1.

Given a set $\mathcal{M}$ containing all measurements, we start by randomly sampling a subset of m measurements called the *hypothetical inliers*. Based on the elements of this sample subset we find a best estimation (hypothesis) using a particle filter. As stated previously, we employ in this work the Scaling Series method [Petrovskaya and Khatib, 2011a] for its ability to deal with 6-DOF localization with large initial uncertainty in a timely fashion. Once a model has been hypothesized from this minimal subset, the remaining data points

Algorithm 1: Outlier classification

Input : $\mathcal{M}$: set of all measurements;
 K : max number of iterations;
 m : size of initial subsets;
 d : minimum size for good subsets;
 ϵ : threshold to include in inliers;
 δ : threshold to consider as found.

```
1 for  $i = 1$  to  $K$  do
2   maybe_inliers  $\leftarrow m$  random measurements  $\in \mathcal{M}$ 
3   remainders  $\leftarrow \mathcal{M} \setminus \text{maybe\_inliers}$ 
4   pose_hypo  $\leftarrow \text{EstimatePose}(\text{maybe\_inliers})$ 
5   foreach measurement  $\in$  remainders do
6     if  $d(\text{pose\_hypo}, \text{measurement}) < \epsilon$  then
7       Add measurement to maybe_inliers
8     if  $\# \text{maybe\_inliers} > d$  then
9       new_pose  $\leftarrow \text{EstimatePose}(\text{maybe\_inliers})$ 
10      new_err  $\leftarrow \text{AverageDistance}(\text{new\_pose}, \text{maybe\_inliers})$ 
11      if  $\text{new\_err} < \text{best\_err}$  then
12        best_err  $\leftarrow \text{new\_err}$ 
13        best_pose  $\leftarrow \text{new\_pose}$ 
14        if  $\text{new\_err} < \delta$  then
15          break
```

are examined to determine which agree with the hypothesis (line 6). This can be achieved by evaluating the Mahalanobis distance as in Eqn. 5.2 and 5.3. The points that fit the estimated model well are considered as part of the consensus set. The estimated model is reasonably good if sufficiently many points have been classified as part of the consensus set (lines 8). Subsequently, the model may be improved by re-estimating it using all members of the consensus set, and a measure of how good the model is can be estimated following Eqn. 5.9 (lines 9, 10). These measures are stored and used to select the best hypothesis. This process is then repeated until a termination criterion is met.

5.3.2 Algorithm Parameters

The standard termination criterion for RANSAC is based on the minimum number of samples required to ensure, with some level of confidence, that at least one of the selected minimal subsets is outlier-free. Let K be the number of iterations, K can be chosen as suggested by [Raguram et al., 2008]:

$$K := \frac{\log(1 - p)}{\log(1 - w^m)}, \quad (5.8)$$

where p is the probability we expect the algorithm to select only inliers from the input data set, w is the probability of choosing an inlier each time a single point is selected. Though this gives an estimate of the required number of iterations, it could result the very large number of iterations when the proportion of inliers is relatively small. Therefore, this should be taken only as the upper limit of iterations. In our algorithm, we terminate the process when the “model goodness” falls below a certain threshold. Here we define the model goodness as follows:

$$G := \frac{1}{I} \sum_i^I d_i, \quad (5.9)$$

where I is the total number of measurements in the current consensus set, d_i is the distance between the measurement i in the set and the object at estimated pose (as defined in Eqn. 5.3,5.2). In other words, given the measurement consensus set, the proposed model goodness is the *average measurement distance* between each measurement in the set and the estimated object. Since it encapsulates all the distance errors, G provides a good numerical evaluation of the localization. As a further benefit, it can be computed easily on-line at each iteration and could therefore be monitored to understand when to stop the algorithm. Nevertheless, it is worth noticing that when the measurement errors are too large (indicating by the values of σ_{nor} and σ_{pos}), (5.9) can be non-informative (i.e.

G might be low even if it is associated to a wrong object pose). In that case, using the maximum number of iterations K is suggested.

Since the presented algorithm requires to perform the Bayesian update steps many times (as part of EstimatePose), it is critical to use the efficient face selection technique presented in Section 5.2 in order to achieve practical running times.

5.4 Simulations

In order to validate the performance of our proposed methods, a Python implementation has been tested via simulations on different objects and collection of measurements. All experiments were run on a machine with a 3.40 GHz processor, 4GB RAM. Our implementation is open-source and can be found online at <https://bit.ly/2LwUVjh>.

5.4.1 Efficient Face Selection

Here we evaluate the proposed face selection algorithm. As mentioned earlier, the proposed improvement procedure for the measurement likelihood evaluation can be applied equally well in the context of particle filter and many of its variants. Here, for the sake of comparison, we apply our procedure to the Scaling Series algorithm [Petrovskaya and Khatib, 2011a] and compare it with the vanilla version.

The simulation setup consisted of 3 objects: a rectangle box, the back of a chair, and a simplified mesh of a cash register (Fig. 5.4).

The performance of the proposed algorithm was assessed in terms of both *reliability* and *execution time*. Reliability was measured in terms of the number of successes in all the trials. A trial was considered as a fail if the estimated pose was far from the real pose. Let $\hat{\mathbf{X}} = (\hat{\mathbf{R}}, \hat{\mathbf{t}})$ and $\bar{\mathbf{X}} = (\bar{\mathbf{R}}, \bar{\mathbf{t}})$ be the estimated and the real poses, respectively. We

Objects	Face Selection(FS)	Without FS	Improvement
Box	1.06±0.21	3.93±0.79	3.7x
Back	2.71±0.65	12.03±1.61	4.4x
Register	2.07±0.27	13.50±2.27	6.5x

Table 5.1: Execution time, in seconds, for Scaling Series method with and without the proposed face selection.

defined the distance metrics for rotation and translation as follows:

$$d(\hat{\mathbf{R}}, \bar{\mathbf{R}}) = \sqrt{\|\log(\hat{\mathbf{R}}^{-1} \bar{\mathbf{R}})\|^2}, \quad (5.10)$$

$$d(\hat{\mathbf{t}}, \bar{\mathbf{t}}) = \sqrt{\|\hat{\mathbf{t}} - \bar{\mathbf{t}}\|^2}, \quad (5.11)$$

where $\|\cdot\|$ denoted the Euclidean norm (refer to [Park, 1995] for more details). In our experiment, the thresholds of 0.005 mm in $d(\hat{\mathbf{t}}, \bar{\mathbf{t}})$ and 0.05 rad in $d(\hat{\mathbf{R}}, \bar{\mathbf{R}})$ were used to indicate whether or not a trial was successful.

For each object, we ran both methods over 50 trials. The initial uncertainties for all objects were 50 mm along x, y, z and 0.5 rad in rotations about x, y, z . At each trial, we randomly selected a pose from the uncertainty region and used it as the ground truth. The measurements set used in the simulation tests were then drawn by sampling random points on 3D model faces. These measurements were perturbed by Gaussian noises with variances $\sigma_{\text{pos}} = 2$ mm and $\sigma_{\text{nor}} = 0.09$ rad. For each trial, we drew a sufficient number of measurements to fully constrain the object estimation and used this set of measurements for both methods. The parameters for the Scaling Series algorithm were chosen as suggested in [Petrovskaya and Khatib, 2011a] and $\delta_\alpha = 0.09$ rad was used as the threshold of face selection algorithm. After the algorithm terminated at each trial, the pose was estimated by computing the mean of the resulting distribution.

Tables 5.1 shows execution times for both methods. For all objects, it shows that the proposed algorithm greatly improves the execution time. The improvements were more significant for complex objects (i.e. with a larger number of faces). For example, the

Figure 5.4: The 3 object models used in our experiments: a box, the back of a chair, and a simplified mesh of a cash register. Model complexity ranges from 12 triangle faces (for the box) to over 44 faces (for the back).

time difference ranges from 3.7 times for the box to 6.5 times for the cash register model. This could be expected since our algorithm focuses the computational resources only on a relevant subset of faces during the measurement likelihood evaluation. Note that both methods displayed the same level of reliability since all trials succeeded in both cases.

In this experiment, we did not increase the initial uncertainty to keep the running time of the Python implementation moderate. However, with larger initial uncertainty, the required number of particles and measurement evaluation steps will considerably increase. Hence the improvement on the running time will be even more significant.

5.4.2 Object Localization in a Cluttered Environment

In this experiment, we consider a scene where the object to be located (the box in red) is placed near two other objects (the back of a chair and a wood stick), see Figure 5.5.

To sample measurements, we simulated, by a ray tracing method, approaching actions by a manipulator end-effector from random directions. In practice, even when prior information of the box is known and the approaching actions are planned carefully, a number of outliers still appear because of uncertainties arising during the execution. In our experiment, the number of outliers ranges from 1 to 5 over a total of 15 measurements.

Figure 5.5: The object to be located (in red) is placed in a cluttered environment. Measurements including outliers are shown as spheres and normal vectors.

While some of these outliers might have been discarded by considering e.g. the initial probability of the object, others were very difficult to be differentiated from the inliers. Note that, in our case, no information about the other objects was fed into the algorithm, which brings about the need for an outlier classification procedure.

The initial uncertainty about the object was assumed to be 10 mm along x, y, z and 0.3 rad in rotations about x, y, z. The measurements were generated using the same parameters as last experiment. The parameters for the Scaling Series algorithm were chosen as suggested in [Petrovskaya and Khatib, 2011a] and $\delta_\alpha = 0.09$ rad was used as the threshold of face selection algorithm.

Methods	Trans err (mm)	Rot err (rad)
W/ Outlier Classif. (OC)	0.81 ± 0.45	0.015 ± 0.012
Without OC	6.64 ± 5.21	0.1 ± 0.06

Table 5.2: Average distance errors in translation and rotation for both methods.

We performed 50 trials and recorded the execution time, together with the average distances between the estimated poses and the real ones. Over all trials, the proposed algorithm succeed in locating the object of interest with an average execution time 21.3 ± 17.2 seconds. We also performed the Scaling Series method (without outlier

classification) over the same data set. In this case, outliers always shift the estimated pose distribution away from the real pose and cause significantly larger errors, see Table 5.2.

5.5 Conclusion

This Chapter was concerned with the touch-based localization problem in cluttered environments, where outlier measurements can lead to significant loss in precision in existing approaches. Our main contributions consist in applying RANSAC to a Bayesian estimation framework and in proposing a novel face selection procedure to improve the speed of the measurement likelihood evaluation in the Bayesian updating steps. Experiments showed that our algorithm could provide, in a timely fashion, accurate and reliable localization in cluttered environments, in the presence of outliers.

Chapter 6

Conclusion

6.1 Summary

In this thesis, we are interested in manipulation tasks under the presence of uncertainties. We have presented a novel probabilistic framework to precisely keep track of uncertainties in manipulation tasks. We also discussed various related problems associated with estimating/propagating uncertainties in different stages of manipulation tasks. In addition, we have also presented an improvement to motion planning.

Tracking Uncertainties in Manipulation Tasks

A common assumption when planning for manipulation tasks is that the robot systems have perfect knowledge of the poses of the objects in the environment. However, this does not always hold true, i.e. the vision systems are inherently erroneous, physical interactions between robots and objects can also introduce additional uncertainties. Depend on the task, the lack of precise knowledge about the uncertainties of the object poses can cause damages to the manipulated objects and/or to the robots. We addressed this problem by proposing a new probabilistic framework to keep track of uncertainties in manipulation tasks. In particular, we decomposed the manipulation task into two sub-

sequent stages, namely perception and physical interactions. Since each stage associates with separated sources of uncertainties and requires different techniques to work with, we presented appropriate approaches to address a number of problems commonly arising in each stage. The capabilities of having access to such valuable information is, in fact, complementary to many existing approaches when dealing with uncertainties. We demonstrated this by conducting a part assembly experiment on a real setting.

Uncertainty in the Hand-eye Calibration

Hand-eye calibration is an important part of the perception stage. In practice, the uncertainty of the hand-eye transformation arguably contributes the most to the uncertainty of the object pose. We have presented a rigorous derivation of the covariance of the solution $\mathbf{X}$, when $\mathbf{A}$ and $\mathbf{B}$ are randomly perturbed matrices. Our approach consists in transposing methods for forward and backward propagation of covariance into the framework of uncertainty in $SE(3)$. Experiments involving synthetic and real calibration data showed that our approach can predict the covariance of the hand-eye transformation with excellent precision. In addition, we also presented a propagation experiment to find the uncertainty of the object pose.

In our method, by analyzing critical factors influencing the covariance of $\mathbf{X}$, one may be able to yield insights into how the uncertainties on the measurements of $\mathbf{A}$ and $\mathbf{B}$ propagate to the uncertainty of the hand-eye transformation. Hence, one can refine the calibration process to achieve a higher precision, by e.g. determining the appropriate number of sample viewpoints or choosing their optimal distribution, which is one of the objects of our future research.

Improvement to Touch-based Localization

Touch-based object localization is one of the important actions in the physical interaction stage. A major challenging problem is that, when the target object is placed within a

cluttered environment, the robot might touch a different object and/or obstacles instead of the target object, generating thereby an outlier measurement. Such outlier measurements, however, can lead to significant loss in precision in existing approaches. Our main contributions consist in applying RANSAC to a Bayesian estimation framework and in proposing a novel face selection procedure to improve the speed of the measurement likelihood evaluation in the Bayesian updating steps. Experiments showed that our algorithm could provide, in a timely fashion, accurate and reliable localization in cluttered environments, in the presence of outliers.

Contribution to Motion Planning

Motion planning is an essential part in a manipulation planner. Fast manipulation motions largely benefit from fast trajectories, subject to various constraints, generated by a motion planner.

Unlike the previous contributions which focused on uncertainties in manipulation tasks, this part merely places emphasis on the kinodynamic motion planning problem. In particular, we do not consider motion planning on belief space. Instead, our approach is to plan geometric motions and execute these motions on a compliant robot (as depicted in Chapter 3). After briefly reviewing a number of existing approaches to kinodynamic motion planning, we present the well-known plan-and-shortcut pipeline that was used to generate trajectories in our system. Finally, we introduced the motion planning in $SE(3)$ problem in which we aimed to plan fast and collision-free trajectories in the spaces of three-dimensional rotations $SO(3)$ and three-dimensional rigid body motions $SE(3)$ under kinodynamic constraints. Our main contribution consisted in extending the classical Time-Optimal Path Parameterization (TOPP) algorithm to $SO(3)$. We showed that, overall, the proposed method could very efficiently find fast rotation motions in $SO(3)$ and $SE(3)$ spaces in a cluttered environment.

6.2 Future Works

Looking forward, there are, still, a number of barriers lying ahead along the road to achieve the fully robotic automation of manipulation tasks. In this Section, we will show some challenging but interesting directions for future research.

We have proposed the method to precisely keep track of the uncertainties of the objects poses through all stages of a manipulation task. So far we have only studied the cylindrical peg-in-hole task to exploit the benefits of having access to such precise information about the uncertainties. We believe the proposed techniques can also be applied in various manipulation tasks that require the knowledge of the uncertainties of the object poses. One of the useful practical applications includes the prediction of the success probability in performing a specific action. In fact, knowing the success probability allows the robot to decide whether or not the action should be carried out to complete the intended task. Furthermore, the detailed description of the uncertainties can also be employed in planning particular actions, which contributes to minimize the number of required actions and, at the same time, to decrease the size of the uncertainties. In addition to the two mentioned applications, the fine-grained information about the uncertainties is definitely beneficial to more complex manipulation tasks such as the insertion tasks with multiple pins or non-cylindrical pins/holes. Last but not least, since the framework we have developed is mainly used for unimanual manipulation tasks, the future work can inherit the proposed techniques to employ in bimanual manipulation tasks with higher complexity. Although these directions are considered challenging, they undeniably play a crucial role in making a firm step towards the advancement of robotic manipulation systems.

Appendix A

Contributions to Motion Planning

As execution time is one of the crucial factors in industrial productivity, apart from being safe, the generated motions should be also fast. When planning motions subject to only *kinematic constraints* (e.g. robot joint limits and obstacle avoidance), solutions are purely geometrical objects, or in other words, path descriptions contain no time information. Planning such motions in manipulation tasks often occur in high-dimensional configuration spaces in which sampling-based planners are proven to be highly efficient. These include the probabilistic roadmap (PRM) [Kavraki et al., 1996], rapidly exploring random tree (RRT) [Lavalle, 1998]. However, to execute the motions after planning, time information along the path is required. Other constraints which include higher-order time derivatives of the robot configuration (e.g. bounds on the robot velocity, acceleration, and force/torque) must also be taken into account. This problem is known as *kinodynamic motion planning*, which was first introduced in [Donald et al., 1993]. Unlike the previous Chapters which focus on uncertainties in manipulation tasks, this Appendix merely places emphasis on this kinodynamic motion planning problem. In particular, we do not consider motion planning on belief space. Instead, our approach is to plan geometric motions and execute these motions on a compliant robot (as depicted in Chapter 3).

In the first part of this Appendix, we will briefly review a number of existing approaches to kinodynamic motion planning and discuss the methods we employ to gen-

erate robot trajectories in our system. To be specific, we will provide an overview on the sampling-based planners and the trajectory planning with kinodynamic constraints. We, then, will present the well-known *plan-and-shortcut* pipeline used to generate trajectories in our system. Due to its robustness, this pipeline is widely employed by many research groups and industrial robotics. In the second part, we will introduce the *motion planning in $SE(3)$* problem in which we aim to plan fast and collision-free trajectories in the spaces of three-dimensional rotations $SO(3)$ and three-dimensional rigid body motions $SE(3)$ under kinodynamic constraints in cluttered environments. In fact, planning the trajectories of objects is an essential step in many bimanual manipulation tasks. There have been a number of researchers who find our proposed work helpful to plan rigid body motions in their works. To name but a few, [Xian et al., 2017, Lertkultanon and Pham, 2018] applied this work in planning for multi-arm systems to carry out manipulation tasks involving large or heavy objects. Our work on the motion planning of rigid-body motions has been published in **Huy Nguyen** and Quang-Cuong Pham (2016), “Time-Optimal Path Parameterization of Rigid-Body Motions: Applications to Spacecraft Reorientation”, In *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 7, pp. 1667-1671.

A.1 Approaches to Optimal, Kinodynamic Motion Planning

Over the past decades, motion planning has been widely used in robotics and automation. Although it first appeared as a field of robotics, motion planning has also been successfully applied to numerous of fields aside from robotic applications[Latombe, 1999].

A.1.1 Sampling-based Path Planners

Path planning, a purely geometric process which only concerns on finding a collision-free path regardless of the feasibility of the path, is perhaps the most basic form of the motion planning problem. Many efforts, such as early works in [Asano et al., 1985], [Lozano-Pérez and Wesley, 1979], [Hart et al., 1968], [Stentz, 1993], [Reif, 1979] and [Schwartz and Sharir, 1983], have been proposed for solving such problem. However, they are usually computationally expensive, therefore, not capable for planning in high-dimensional configuration spaces. Sampling-based algorithms, hence, were developed with the aim of being able to overcome the complexity of robot planning algorithms for high-dimensional problems. This randomized approach has demonstrated their efficacy for solving difficult problems. Sampling-based planners can be classified into two families: probabilistic roadmaps (PRM) and rapidly-exploring random trees (RRT). Next, some background on these two approaches will be reviewed.

Probabilistic Roadmap (PRM)

The basic idea behind PRM is to generate a probabilistic roadmap to represent the connectivity of the free configuration space [Kavraki et al., 1996]. As a result of maintaining the roadmap, PRM is able to solve problems where several queries involving the same system. It is also known as a multiple-query method. Most of the planning time is spent on generating the roadmap; a query, hence, can be found very quickly. Many algorithms have been proposed to improve sampling and making connections in roadmaps. In [Kavraki et al., 1996], vertex enhancement approach gives more effort on vertices that were difficult to connect to other vertices in the roadmap construction algorithm. Authors in [Amato et al., 1998a], [Amato and Wu, 1996] showed that it is sometimes better to sample along the boundary of free space. Likewise, Gaussian sampling strategy proposed in [Boor et al., 1999] manages to increase the probability of sampling around obstacles. Bridge-test sampling [Hsu et al., 2003] uses three samples along a short segment at-

tempt to identify narrow corridors in the environment. [Holleman and Kavraki, 2000], [Wilmarth et al., 1999] introduced medial axis sampling methods which force the samples to be as far from the boundary as possible. Some analysis of PRM planners are provided in [Kavraki et al., 1998], [Ladd and Kavraki, 2004]. Further reading on the probabilistic foundations of PRM can be found in [Hsu et al., 2006]

Rapidly-exploring Random Tree (RRT)

The rapidly-exploring random tree (RRT) is another class of sampling-based planners, i.e. single-query planners. Commonly, a tree rooted from start configuration is incrementally grown towards goal configuration. Once a new sample is selected, if it lies in the free space, a connection is attempted to the nearest vertex of the tree. This new sample and its connection are added to the current tree if the connection lies entirely in the free space and satisfies all the constraints. In single-query problems, RRT planners are usually faster compared to PRM planners since they do not need preprocessing. Another important property of RRT planners is that it introduces Voronoi bias, i.e. the tree tends to expand towards large unsearched areas. Due to these reasons, RRT planners can *efficiently and quickly explore high-dimensional spaces* and is suitable to handle various systems involving different constraints.

The RRT planner was first proposed in [Lavalle, 1998] as a new randomized data structure for motion planning problem. To improve the performance of RRT planners, it is suggested that the search can be performed in bidirectional directions. This strategy constructs *two* trees instead of one, i.e. one tree grown from start configuration and another grown from goal configuration. The solution is found when the two trees can be successfully connected. There exist many other approaches to improve performance of the planners [Cheng and LaValle, 2002], [Sánchez and Latombe, 2003], [Lindemann and LaValle, 2004], [Yershova et al., 2005], [Akinc et al., 2005], [Vahrenkamp et al., 2009], [Kesselman et al., 2014]. However, the recommended choice mainly depends on the charac-

teristics of problem to be solved. Some analyses of RRT planners can be found in [Kuffner and LaValle, 2000] and [LaValle and Kuffner Jr, 2000].

A.1.2 Trajectory Planning with Kinodynamic Constraints

In overall, there are two principal approaches used to cope with the problems related to trajectory planning with kinodynamic constraints, which are the *state space approach* and *configuration space approach*.

State Space Approach

The first approach is to plan directly a motion in the *state space* of the robot, a space of robot configurations together with their first derivatives. Standard sampling-based path planners such as PRM and RRT-based planners have been successfully extended to directly plan in the state space, for instance, see [LaValle and Kuffner, 2001],[Hsu et al., 2002]. Nonetheless, the performance of such approaches are inherently sensitive to the deployed distance metric [Amato et al., 1998b, LaValle and Kuffner, 2001]. For example, [Shkolnik et al., 2009] showed that Euclidean distance metric resulted in a poorly-performed planner, even in case of a 1-DOF pendulum under the joint torque limits. Moreover, the manifold structure of the state space entails more difficulties in connecting two states with kinodynamic constraints. In addition to this problem, due to the high dimensionality of the state space, the complexity of the problem is considerably escalated. For these reasons, we decide to employ the configuration space approach in order to plan trajectories for our manipulation tasks.

Configuration Space Approach

Unlike the state space approach, configuration space approach is based on the idea of separating the principal problem into two different subproblems, which are path planning and time-parameterization. The former subproblem, i.e. , path planning problem, is

entirely geometric. Only kinematic constraints, such as the joint limits of the robot and the obstacle avoidance, are investigated carefully in this phase. Moreover, the choice of path planners that can be used is diversified. After obtaining a path from the planner, the latter subproblem aims to assign a *velocity profile* along the path so that the trajectory can be dynamically executed. One of the advantages of such planners is that the search space is only a half compared to it of the state space approach. As a result, the expected operating time of the planner is significantly reduced. This approach is also known as *path-velocity decomposition* approach [Kant and Zucker, 1986, Pham et al., 2017].

However, there is no guarantee that the obtained path will be time-parameterizable by taking into account only kinematic constraints in the first phase. This is a major issue due to the fact that a non-parameterizable path needs to be discarded and the planning has to get back to the starting point. A common method to address this issue is to enforce quasi-static constraints to ensure the existence of time-parameterization functions in path planning stage [Kuffner Jr et al., 2002]. However, this also have another limitation that all dynamically feasible motions that are not quasi-statically feasible are overlooked. In 2013, the Admissible Velocity Propagation algorithm (AVP) has been recommended in [Pham et al., 2013] to particularly deal with this issue. The algorithm is incorporated into the path planing stage and allows the planner to plan a kinodynamically practicable motions while still remaining in the configuration space. In [Pham et al., 2017], AVP algorithm was further extended to the class of bidirectional planners.

A.1.3 Trajectory Planning Pipeline

After reviewing a number of existing motion planning algorithms, in this Section, we will present the well-known *plan-and-shortcut* pipeline that will be used to generate trajectories in our system. Due to its robustness [Luo and Hauser, 2014], the pipeline is widely used in industrial robotics. It is also the default planning pipeline implemented in the robot programming environment OpenRAVE [Diankov, 2010], which is used by

many research groups and robotic companies worldwide.

This pipeline consists of three main steps, including:

1. Plan a collision-free *geometric path* between two robot configurations $\mathbf{x}_{start}$ and $\mathbf{x}_{goal}$ in the robot joint space C ;
2. Time-parameterize that geometric path to obtain a *trajectory* that respects the kinodynamic constraints (bounds on the robot velocities, accelerations, etc.);
3. Repeatedly apply *shortcuts* to decrease the time duration of the obtained trajectory.

In the first step, we use the path planner, i.e. a Rapidly-exploring Random Tree (RRT), to plan a collision-free path between $\mathbf{x}_{start}$ and $\mathbf{x}_{goal}$. In fact, the resulting path is obtained by concatenating linear path segments together. Each linear segment from $\mathbf{x}_a$ to $\mathbf{x}_b$ is represented by a parameterization $\mathbf{x}(s) = \mathbf{x}_a + s(\mathbf{x}_b - \mathbf{x}_a)$, $s \in [0, 1]$. After this step, a piecewise linear path is obtained. Next, in step 2, we optimally time-parameterize each of the *linear* segment a velocity profile. Let T be the total duration of the resulting trajectory, we compute a time-derivative of the path parameterization function $s : [0, T] \rightarrow [0, 1]$ and assign it to the path. To avoid discontinuities of the velocity vector at the junctions of the “linear” segments, we ensure that the velocities at the beginning and the end of each segment are zero. At the end, the time-parameterization step will result a trajectory $\mathbf{x} : [0, T] \rightarrow C$, with $\mathbf{x}(0) = \mathbf{x}_{start}$ and $\mathbf{x}(T) = \mathbf{x}_{goal}$. In step 3, at each shortcut iteration, we randomly select two time instants along the trajectory t_a, t_b and find the interpolation path between $\mathbf{x}(t_a), \dot{\mathbf{x}}(t_a)$ and $\mathbf{x}(t_b), \dot{\mathbf{x}}(t_b)$. Once the path is collision-free, it is then optimally time-parameterized. If the resulting duration is shorter, we then replace the original trajectory by the shortcut.

A.2 Trajectory Planning with Kinodynamic Constraints in $SO(3)$, $SE(3)$

The trajectory planning pipeline presented in the last Section has been proven to be very robust when planning trajectories for industrial robotic manipulation. However, when rigid body motions are involved, the problem of finding fast and collision-free trajectories under kinodynamic bounds is very challenging. This is due to the non-trivial natures of the constraints, and of the underlying space, the space of rigid body configurations, or $SE(3)$. In this Section, we will present how we adopt the plan-and-shortcut method to resolve such problem.

A.2.1 Overview

When only rotations are involved (no translation), the space of possible motions is $SO(3)$. There are two main directions of research in the literature. The aerospace community has considered the problem of finding exact time-optimal reorientation trajectories under kinodynamic constraints by applying directly Pontryagin's minimum principle [Li and Bainum, 1990, Bilimoria and Wie, 1993, Bai and Junkins, 2009]. While the theoretical interest of such an approach is evident, in practice, it is very computationally intensive and does not seem to be generalizable beyond rest-to-rest motions in obstacle-free environments.

In the robotics community, popular sampling-based motion planning methods such as RRT or PRM have been extended to $SO(3)$. Such methods require three components: (i) random sampling, (ii) distance metric, and (iii) interpolation between two points. While straightforward in Euclidean spaces, these three components involve considerable challenges in $SO(3)$. Regarding component (i), a sampling algorithm based on unit quaternions was proposed in [Kuffner, 2004]. As for components (ii) and (iii), methods based

on unit quaternions or on rotation matrices exist, but have rarely been considered in the contexts of kinodynamic constraints and time-optimality.

As a result, to our knowledge, there are no general methods for planning fast trajectories in $\mathbf{SO}(3)$ under kinodynamic constraints and in a cluttered environment. Here we address this problem by following the widely used plan-and-shortcut pipeline, which has shown to be easy to implement, robust and efficient. The pipeline consists of 3 steps (as presented in last Section). Step 1 of the pipeline which plans piece-wise “linear” paths has been addressed, in particular by Kuffner [Kuffner, 2004], by extending the classical Rapidly exploring Random Tree (RRT) algorithm [LaValle and Kuffner, 2001] to $\mathbf{SO}(3)$. Regarding steps 2 and 3, the crucial requirement is the ability to optimally *time-parameterize* a given path in $\mathbf{SO}(3)$ under kinodynamic constraints. We do so by extending the classical [Bobrow et al., 1985, Shin and McKay, 1985, Pham, 2014] Time-Optimal Path Parameterization (TOPP) algorithm to the case of $\mathbf{SO}(3)$, which constitutes the main contribution of this work. We show that, overall, the method we propose can find fast rotation motions in a cluttered environment in less than 10 seconds. We also present an extension to the space of three-dimensional rigid body motions, $\mathbf{SE}(3)$.

A.2.2 Interpolation in $\mathbf{SO}(3)$ and $\mathbf{SE}(3)$

Notations

We first recall some useful notations in the study of $\mathbf{SO}(3)$ [Park and Ravani, 1995]. Given a vector $\mathbf{r} \in \mathbb{R}^3$, one defines the skew-symmetric matrix

$$[\mathbf{r}] \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}.$$

The exponential of a skew-symmetric matrix defined as above is given by

$$\mathbf{e}^{[\mathbf{r}]} \stackrel{\text{def}}{=} \mathbb{I} + \frac{\sin \|\mathbf{r}\|}{\|\mathbf{r}\|} [\mathbf{r}] + \frac{1 - \cos \|\mathbf{r}\|}{\|\mathbf{r}\|^2} [\mathbf{r}]^2. \quad (\text{A.1})$$

Next, for any matrix $\mathbf{R} \in \mathbf{SO}(3)$, there exists $\mathbf{r}$ such that $\mathbf{R} = \mathbf{e}^{[\mathbf{r}]}$, and one denotes $\log \mathbf{R} \stackrel{\text{def}}{=} [\mathbf{r}]$. More precisely, if $\text{tr}(\mathbf{R}) \neq 1$, then

$$\log \mathbf{R} = \frac{\phi}{2 \sin \phi} (\mathbf{R} - \mathbf{R}^\top), \quad (\text{A.2})$$

where ϕ satisfies $1 + 2 \cos \phi = \text{tr}(\mathbf{R})$. If $\text{tr}(\mathbf{R}) = 1$, then

$$\log \mathbf{R} = \pm \pi [\hat{\mathbf{v}}], \quad (\text{A.3})$$

where $\hat{\mathbf{v}}$ is a unit length eigenvector of $\mathbf{R}$ associated with the eigenvalue 1.

Interpolation in $\mathbf{SO}(3)$

Consider two orientation matrices $\mathbf{R}_0, \mathbf{R}_1 \in \mathbf{SO}(3)$ and two vectors $\boldsymbol{\omega}_0, \boldsymbol{\omega}_1 \in \mathbb{R}^3$. We are interested in the problem of finding a smooth, “short” curve $(\mathbf{R}(t))_{t \in [0,1]}$ in $\mathbf{SO}(3)$ such that

$$\mathbf{R}(0) = \mathbf{R}_0, \mathbf{R}(1) = \mathbf{R}_1, \boldsymbol{\omega}(0) = \boldsymbol{\omega}_0, \boldsymbol{\omega}(1) = \boldsymbol{\omega}_1,$$

where $\boldsymbol{\omega}(t)$ denotes the angular velocity vector computed in the moving (body) frame.

Consider a vector $\mathbf{r}_1$ such that $[\mathbf{r}_1] = \log(\mathbf{R}_0^\top \mathbf{R}_1)$. One possible interpolation method is given by [Park and Ravani, 1995, Park and Ravani, 1997]

$$\mathbf{R}(t) = \mathbf{R}_0 \mathbf{e}^{[a_3 t^3 + a_2 t^2 + a_1 t]}, \quad (\text{A.4})$$

where $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 \in \mathbb{R}^3$ are constant vectors satisfying

- (i) $\mathbf{a}_1 = \boldsymbol{\omega}_0$;

$$(ii) \quad \mathbf{a}_3 + \mathbf{a}_2 + \mathbf{a}_1 = \mathbf{r}_1;$$

$$(iii) \quad 3\mathbf{a}_3 + 2\mathbf{a}_2 + \mathbf{a}_1 = \mathbf{A}^{-1}(\mathbf{r}_1)\boldsymbol{\omega}_1 \text{ [the definition of } \mathbf{A} \text{ is given in equation (A.7) below].}$$

This interpolation method has many advantages. First, it approximates the minimum acceleration interpolation, which implies that the underlying path is smooth and short. In particular, if $\boldsymbol{\omega}_0 = \boldsymbol{\omega}_1 = \mathbf{0}$, then this interpolation indeed yields the shortest path in $SO(3)$ [Park and Ravani, 1997]. This interpolation is also bi-invariant: the resulting orientation trajectories are invariant with respect to the choice of fixed or moving reference frames. Finally, the first and second derivatives are smooth and easy to compute, which is useful for TOPP where we need smooth and easy-to-compute first and second derivatives. The angular velocity and acceleration are indeed given by

$$\boldsymbol{\omega}(t) = \mathbf{A}(\mathbf{r})\dot{\mathbf{r}}, \quad (A.5)$$

$$\dot{\boldsymbol{\omega}}(t) = \mathbf{A}(\mathbf{r})\ddot{\mathbf{r}} + \mathbf{C}(\mathbf{r}, \dot{\mathbf{r}}), \quad (A.6)$$

where

$$\mathbf{A}(\mathbf{r}) \stackrel{\text{def}}{=} \mathbb{I} - \frac{1 - \cos \|\mathbf{r}\|}{\|\mathbf{r}\|^2} [\mathbf{r}] + \frac{\|\mathbf{r}\| - \sin \|\mathbf{r}\|}{\|\mathbf{r}\|^3} [\mathbf{r}]^2, \quad (A.7)$$

and

$$\begin{aligned} \mathbf{C}(\mathbf{r}, \dot{\mathbf{r}}) &\stackrel{\text{def}}{=} \frac{\|\mathbf{r}\| - \sin \|\mathbf{r}\|}{\|\mathbf{r}\|^3} \dot{\mathbf{r}} \times (\mathbf{r} \times \dot{\mathbf{r}}) \\ &- \frac{2 \cos \|\mathbf{r}\| + \|\mathbf{r}\| \sin \|\mathbf{r}\| - 2}{\|\mathbf{r}\|^4} \mathbf{r}^\top \dot{\mathbf{r}} (\mathbf{r} \times \dot{\mathbf{r}}) \\ &+ \frac{3 \sin \|\mathbf{r}\| - \|\mathbf{r}\| \cos \|\mathbf{r}\| - 2\|\mathbf{r}\|}{\|\mathbf{r}\|^5} \mathbf{r}^\top \dot{\mathbf{r}} (\mathbf{r} \times (\mathbf{r} \times \dot{\mathbf{r}})). \end{aligned} \quad (A.8)$$

Interpolation in $SE(3)$

While $SO(3)$ is the group of 3D rotations, $SE(3)$ is the special Euclidean group of rigid-body motions. $SE(3)$ includes both rotations and translations, and is of the form

$$\begin{bmatrix} \mathbf{R} & \mathbf{q} \\ 0 & 1 \end{bmatrix},$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{q} \in \mathbb{R}^3$. There are a number of available approaches to interpolate trajectories between two elements of $SE(3)$. Park and Ravani [Park and Ravani, 1995] exploited the Lie group structure of $SE(3)$ to develop an algorithm analogous to the preceding interpolation in $SO(3)$. The resulting motion, however, is a screw motion, which corresponds to a strange motion in physical space. Moreover, in $SE(3)$ there is in general no bi-invariant interpolations [Park and Ravani, 1997]. Therefore, we use a simple method, which consists in interpolating orientations and positions separately. For the orientation part, we proceed as in Section A.2.2. For the translation part, we interpolate between two pairs (position, linear velocity) by a third degree polynomial. Consider two pairs $(\mathbf{p}_0, \mathbf{v}_0)$ and $(\mathbf{p}_1, \mathbf{v}_1)$, the interpolant is given by

$$\mathbf{p}(t)_{t \in [0,1]} = \mathbf{k}_3 t^3 + \mathbf{k}_2 t^2 + \mathbf{k}_1 t + \mathbf{k}_0, \quad (\text{A.9})$$

where $\mathbf{k}_3, \mathbf{k}_2, \mathbf{k}_1, \mathbf{k}_0$ can be easily found using the boundary conditions $\mathbf{p}(0) = \mathbf{p}_0, \mathbf{v}(0) = \mathbf{v}_0, \mathbf{p}(1) = \mathbf{p}_1, \mathbf{v}(1) = \mathbf{v}_1$ ($\mathbf{v}(t)$ denotes the linear velocity vector). Note that the first and second derivatives of the interpolant are trivial to compute.

A.2.3 Time-Optimal Path Parameterization in $SO(3)$

Consider a path $\mathcal{P}$ – represented as the underlying path of a trajectory $\mathbf{r}(s)_{s \in [0, s_{\text{end}}]}$ – in the configuration space. Assume that $\mathbf{r}(s)_{s \in [0, s_{\text{end}}]}$ is C^1 - and piecewise C^2 -continuous. We are interested in *time-parameterizations* of $\mathcal{P}$, which are increasing *scalar functions*

$s : [0, T] \rightarrow [0, s_{\text{end}}]$, under kinodynamic constraints.

If the constraints can be reduced to the form (note that all vector inequalities in this Section are element-wise)

$$\ddot{s}\mathbf{a}(s) + \dot{s}^2\mathbf{b}(s) + \mathbf{c}(s) \leq \mathbf{0}, \quad (\text{A.10})$$

then efficient methods and implementations allow finding the time-optimal parameterization $s(t)$ (see e.g. [Pham, 2014]).

Consider now a rigid body with three independent actuations, such as a rigid spacecraft whose equation of motion is [Li and Bainum, 1990, Bilimoria and Wie, 1993, Bai and Junkins, 2009]

$$\mathbb{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbb{I}\boldsymbol{\omega}) = \boldsymbol{\tau}, \quad (\text{A.11})$$

where $\mathbb{I}$ is the 3×3 inertia matrix of the spacecraft and $\boldsymbol{\tau}$ the 3-dimensional torque vectors. The actuation bounds are given by

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max}. \quad (\text{A.12})$$

Consider now an orientation trajectory $\mathbf{R}(s)_{s \in [0,1]} \in \mathbf{SO}(3)$ given by

$$\mathbf{R}(s) = \mathbf{R}_0 \mathbf{e}^{[\mathbf{r}(s)]}. \quad (\text{A.13})$$

Substituting the expressions of $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$ in terms of the path parameter s

$$\dot{\mathbf{r}} = \dot{s}\mathbf{r}_s, \quad \ddot{\mathbf{r}} = \ddot{s}\mathbf{r}_s + \dot{s}^2\mathbf{r}_{ss} \quad (\text{A.14})$$

into (A.5) and (A.6), one obtains

$$\boldsymbol{\omega} = \dot{s}\mathbf{A}(\mathbf{r})\mathbf{r}_s, \quad (\text{A.15})$$

$$\dot{\omega} = \ddot{s}\mathbf{A}(\mathbf{r})\mathbf{r}_s + \dot{s}^2 \{\mathbf{A}(\mathbf{r})\mathbf{r}_{ss} + \mathbf{C}(\mathbf{r}, \mathbf{r}_s)\}, \quad (\text{A.16})$$

$$\boldsymbol{\tau} = \ddot{s}\mathbb{I}\mathbf{A}(\mathbf{r})\mathbf{r}_s + \dot{s}^2 \{\mathbb{I}\mathbf{A}(\mathbf{r})\mathbf{r}_{ss} + \mathbb{I}\mathbf{C}(\mathbf{r}, \mathbf{r}_s) + (\mathbf{A}(\mathbf{r})\mathbf{r}_s) \times (\mathbb{I}\mathbf{A}(\mathbf{r})\mathbf{r}_s)\}. \quad (\text{A.17})$$

Thus, the condition $\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max}$ can be put in the form of (A.10) with

$$\mathbf{a} = \begin{bmatrix} \mathbb{I}\mathbf{A}(\mathbf{r})\mathbf{r}_s \\ -\mathbb{I}\mathbf{A}(\mathbf{r})\mathbf{r}_s \end{bmatrix}, \quad (\text{A.18})$$

$$\mathbf{b} = \begin{bmatrix} \mathbb{I}\mathbf{A}(\mathbf{r})\mathbf{r}_{ss} + \mathbb{I}\mathbf{C}(\mathbf{r}, \mathbf{r}_s) + (\mathbf{A}(\mathbf{r})\mathbf{r}_s) \times (\mathbb{I}\mathbf{A}(\mathbf{r})\mathbf{r}_s) \\ -(\mathbb{I}\mathbf{A}(\mathbf{r})\mathbf{r}_{ss} + \mathbb{I}\mathbf{C}(\mathbf{r}, \mathbf{r}_s) + (\mathbf{A}(\mathbf{r})\mathbf{r}_s) \times (\mathbb{I}\mathbf{A}(\mathbf{r})\mathbf{r}_s)) \end{bmatrix}, \quad (\text{A.19})$$

$$\mathbf{c} = \begin{bmatrix} -\boldsymbol{\tau}_{\max} \\ \boldsymbol{\tau}_{\min} \end{bmatrix}. \quad (\text{A.20})$$

The motion of the spacecraft is generally constrained, not only by the spacecraft itself (for example, torque limits) but also by task constraints (e.g. bounds on angular accelerations). The form (A.10) allows easily combining different type of constraints by concatenating the vectors $\mathbf{a}(s)$, $\mathbf{b}(s)$, $\mathbf{c}(s)$ corresponding to the different constraints. For instance, bounds on angular accelerations ($\dot{\omega}_{\min} \leq \dot{\omega} \leq \dot{\omega}_{\max}$) can be put in the form of (A.10) by noting that $\dot{\omega} = \ddot{s}\mathbf{A}(\mathbf{r})\mathbf{r}_s + \dot{s}^2 \{\mathbf{A}(\mathbf{r})\mathbf{r}_{ss} + \mathbf{C}(\mathbf{r}, \mathbf{r}_s)\}$.

Bounds on angular velocities can be expressed as $\boldsymbol{\omega}^\top \boldsymbol{\omega} \leq \omega_{\max}$, where $\omega_{\max}$ is a scalar. Such constraints were first addressed in [Žlajpah, 1996]. The treatment was completed and implemented in [Pham, 2014].

For time-parameterization a given path in $SE(3)$ under kinodynamic constraints, we separate the process into 2 steps. First, we re-time the orientation part as presented earlier. Regarding the translation part, since we interpolate two pairs $(\mathbf{p}_0, \mathbf{v}_0)$ and $(\mathbf{p}_1, \mathbf{v}_1)$ by a third degree polynomial (Section II A.2.2), the first and second derivatives are trivial

to compute. Therefore, constraints on linear accelerations ($\alpha_{\min} \leq \alpha \leq \alpha_{\max}$) or forces ($f_{\min} \leq f \leq f_{\max}$) can be easily expressed in the form of (A.10).

A.2.4 Implementation and Evaluation

We experimentally evaluated our approach on some spacecraft maneuver problems. Our implementation is open-source and can be found online at <https://bit.ly/2K1OYp7>. All experiments were run on a machine with an Intel i7 3.40 GHz processor, 4GB RAM. Videos of the resulting simulations can be found at <https://bit.ly/2LKdkbH>.

Planning Fast, Collision-free Trajectories in $SO(3)$ Under Kinodynamic Constraints

Here, we present the algorithm to find a fast, collision-free, trajectory connecting two rotations $\mathbf{R}_0, \mathbf{R}_1$, subject to velocity and torque bounds. As mentioned earlier, we follow the plan-and-shortcut method [Geraerts and Overmars, 2007].

In step 1, we use an RRT-based path planner to find a collision-free piecewise “linear” path that connects $(\mathbf{R}_0, \omega_0)$ and $(\mathbf{R}_1, \omega_1)$ (each “linear” segment is actually a great-circle arc in $SO(3)$).

Next, in step 2, we optimally time-parameterize each of the “linear” segment. To avoid discontinuities of the velocity vector at the junctions of the “linear” segments, we ensure that the velocities at the beginning and the end of each segment are zero.

In step 3, at each shortcut iteration, we select two random time instants t_0, t_1 along the trajectory, which correspond to two orientation matrices $\mathbf{R}_{t_0}, \mathbf{R}_{t_1} \in SO(3)$ and two angular velocity vectors $\omega_{t_0}, \omega_{t_1} \in \mathbb{R}^3$. We then find the interpolation path $\mathbf{R}^*(t)_{t \in [0, T]}$, where $T = t_1 - t_0$, such that

$$\mathbf{R}^*(0) = \mathbf{R}_{t_0}, \mathbf{R}^*(T) = \mathbf{R}_{t_1}, \omega^*(0) = \omega_{t_0}, \omega^*(T) = \omega_{t_1}.$$

If this path involves any collision, we discard it. Otherwise, we use TOPP to optimally

Figure A.1: A fast reorientation trajectory in $SO(3)$ for the Messenger spacecraft in a cluttered environment.

time-parameterize it, while ensuring that the initial and final velocities of the short-cut portion are unchanged to avoid velocity discontinuities. If the resulting duration is smaller than T then we replace the original trajectory portion by the shortcut.

Figure A.2: Angular velocities and torques of the trajectory in Fig. A.1.

We tested the proposed algorithm on a reorientation problem for the Messenger spacecraft subject to bounds on angular velocities and torques in a cluttered environment (model downloaded from <https://go.nasa.gov/2vbIXR7>), see Fig. A.1. We ran 100 trials, with 200 shortcut iterations per trial. The average running time for the three steps of the algorithm are shown in Table A.1. Overall, the algorithm can find fast trajectories for the satellite model in this cluttered environment in 7.42 ± 4.58 s. The average number of successful shortcuts per trial was 11.12 ± 6.31 . Fig. A.1 shows one typical trajectory found by the algorithm. Fig. A.2 shows the corresponding angular velocities and torques as functions of time. Note that, $(\omega^1, \omega^2, \omega^3)$ are the three elements of the angular velocity vector ω , while (τ^1, τ^2, τ^3) are the three elements of the torque vector τ . One can see that, in agreement with time-optimality, at least one constraint is saturated at any time instant.

Planning Fast, Collision-free Trajectories in $SE(3)$ Under Kinodynamic Constraints

We now consider the case of rigid-body motions. Figure A.3 shows the environment we used to test the proposed algorithm. We ran 100 trials, with 200 shortcut iterations per trial. The average running time for the three steps of the algorithm is showed in Table A.1. Overall, the algorithm can find fast trajectories for the satellite model in this cluttered environment in 32.21 ± 6.12 s. The average number of successful shortcuts per trial was 18.32 ± 7.34 . The total running time for problems in $SE(3)$ was longer than in $SO(3)$, because of the higher problem dimension and environment complexity. One typical resulting trajectory found by the algorithm is showed in Figure A.3. The corresponding velocities, torques and forces are showed in Figure A.4. Note that, (v^1, v^2, v^3) are the three elements of the linear velocity vector $\mathbf{v}$, while (f^1, f^2, f^3) are the three elements of the force vector $\mathbf{f}$. One can also see here that, in agreement with time-optimality, at least one constraint is saturated at any time instant.

Figure A.3: A fast maneuver trajectory in $SE(3)$ for the Messenger spacecraft in a cluttered environment.

Figure A.4: Velocities, torques and forces of the trajectory in Fig. A.3.

	RRT	TOPP on “linear” segments	Shortcutting	Total running time
$\mathbf{SO}(3)$	1.87 ± 3.01	0.09 ± 0.07	5.34 ± 2.35	7.42 ± 4.58
$\mathbf{SE}(3)$	6.18 ± 5.23	0.32 ± 0.20	25.54 ± 6.53	32.21 ± 6.12

Table A.1: Average running time, in seconds, for the three steps of the plan-and-shortcut method in $\mathbf{SO}(3)$ and $\mathbf{SE}(3)$

A.2.5 Conclusion

We have addressed the problem of planning fast, collision-free, trajectories in the spaces of three-dimensional rotations $\mathbf{SO}(3)$ and three-dimensional rigid body motions $\mathbf{SE}(3)$ under kinodynamic constraints in cluttered environments. Our main contribution consisted in extending the classical Time-Optimal Path Parameterization (TOPP) algorithm to $\mathbf{SO}(3)$. We integrated the algorithm into a plan-and-shortcut pipeline to yield a complete framework (and open-source implementation) for rigid-body motion planning under kinodynamic constraints. We showed that our implementation could very efficiently find near time-optimal trajectories in $\mathbf{SO}(3)$ and $\mathbf{SE}(3)$ spaces.

Bibliography

- [Ackerman et al., 2014] Ackerman, M. K., Cheng, A., and Chirikjian, G. (2014). An information-theoretic approach to the correspondence-free $ax=xb$ sensor calibration problem. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4893–4899. IEEE.
- [Ackerman and Chirikjian, 2013] Ackerman, M. K. and Chirikjian, G. S. (2013). A probabilistic solution to the $ax=xb$ problem: Sensor calibration without correspondence. In *Geometric Science of Information*, pages 693–701. Springer.
- [Akinc et al., 2005] Akinc, M., Bekris, K. E., Chen, B. Y., Ladd, A. M., Plaku, E., and Kavraki, L. E. (2005). Probabilistic roadmaps of trees for parallel computation of multiple query roadmaps. In *Robotics Research*, pages 80–89. Springer.
- [Amato et al., 1998a] Amato, N. M., Bayazit, O. B., and Dale, L. K. (1998a). Obprm: An obstacle-based prm for 3d workspaces.
- [Amato et al., 1998b] Amato, N. M., Bayazit, O. B., Dale, L. K., Jones, C., and Vallejo, D. (1998b). Choosing good distance metrics and local planners for probabilistic roadmap methods. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 630–637. IEEE.

- [Amato and Wu, 1996] Amato, N. M. and Wu, Y. (1996). A randomized roadmap method for path and manipulation planning. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1, pages 113–120. IEEE.
- [Aron et al., 2004] Aron, M., Simon, G., and Berger, M.-O. (2004). Handling uncertain sensor data in vision-based camera tracking. In *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on*, pages 58–67. IEEE.
- [Asano et al., 1985] Asano, T., Asano, T., Guibas, L., Hershberger, J., and Imai, H. (1985). Visibility-polygon search and euclidean shortest paths. In *Foundations of Computer Science, 1985., 26th Annual Symposium on*, pages 155–164. IEEE.
- [Bai and Junkins, 2009] Bai, X. and Junkins, J. L. (2009). New results for time-optimal three-axis reorientation of a rigid spacecraft. *Journal of guidance, control, and dynamics*, 32(4):1071–1076.
- [Barfoot and Furgale, 2014] Barfoot, T. D. and Furgale, P. T. (2014). Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679 – 693.
- [Bilimoria and Wie, 1993] Bilimoria, K. D. and Wie, B. (1993). Time-optimal three-axis reorientation of a rigid spacecraft. *Journal of Guidance, Control, and Dynamics*, 16(3):446–452.
- [Bobrow et al., 1985] Bobrow, J. E., Dubowsky, S., and Gibson, J. S. (1985). Time-optimal control of robotic manipulators along specified paths. *The International Journal of Robotics Research*, 4(3):3–17.
- [Bohg et al., 2017] Bohg, J., Hausman, K., Sankaran, B., Brock, O., Kragic, D., Schaal, S., and Sukhatme, G. S. (2017). Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291.

- [Boor et al., 1999] Boor, V., Overmars, M. H., and van der Stappen, A. F. (1999). The gaussian sampling strategy for probabilistic roadmap planners. In *Robotics and Automation, 1999. Proceedings. ICRA '99. IEEE International Conference on*, volume 2, pages 1018–1023.
- [Brooks, 1982] Brooks, R. A. (1982). Symbolic error analysis and robot planning. *The International Journal of Robotics Research*, 1(4):29–78.
- [Brooks, 1985] Brooks, R. A. (1985). Visual map making for a mobile robot. In *Robotics and Automation, 1985. Proceedings. IEEE International Conference on*, volume 2, pages 824–829. IEEE.
- [Bruyninckx et al., 1995] Bruyninckx, H., Dutre, S., and De Schutter, J. (1995). Peg-on-hole: a model based solution to peg and hole alignment. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 2, pages 1919–1924. IEEE.
- [Bry and Roy, 2011] Bry, A. and Roy, N. (2011). Rapidly-exploring random belief trees for motion planning under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 723–730. IEEE.
- [Chen, 1991] Chen, H. H. (1991). A screw motion approach to uniqueness analysis of head-eye geometry. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 145–151. IEEE.
- [Chen and Chao, 1987] Chen, J. and Chao, L.-M. (1987). Positioning error analysis for robot manipulators with all rotary joints. *IEEE Journal on Robotics and Automation*, 3(6):539–545.
- [Cheng and LaValle, 2002] Cheng, P. and LaValle, S. M. (2002). Resolution complete rapidly-exploring random trees. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, pages 267–272.

- [Chhatpar and Branicky, 2005] Chhatpar, S. R. and Branicky, M. S. (2005). Particle filtering for localization in robotic assemblies with position uncertainty. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3610–3617. IEEE.
- [Chirikjian, 2009] Chirikjian, G. (2009). *Stochastic Models, Information Theory, and Lie Groups, Volume 1: Classical Results and Geometric Methods*, volume 1. Springer Science & Business Media.
- [Chirikjian, 2011] Chirikjian, G. (2011). *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*, volume 2. Springer Science & Business Media.
- [Chou and Kamel, 1991] Chou, J. C. and Kamel, M. (1991). Finding the position and orientation of a sensor on a robot manipulator using quaternions. *The international journal of robotics research*, 10(3):240–254.
- [Corcoran and Platt, 2010] Corcoran, C. and Platt, R. (2010). A measurement model for tracking hand-object state during dexterous manipulation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4302–4308. IEEE.
- [Daniilidis, 1999] Daniilidis, K. (1999). Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research*, 18(3):286–298.
- [De Schutter et al., 2007] De Schutter, J., De Laet, T., Rutgeerts, J., Decré, W., Smits, R., Aertbeliën, E., Claes, K., and Bruyninckx, H. (2007). Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research*, 26(5):433–455.
- [Diankov, 2010] Diankov, R. (2010). *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute.

- [Donald et al., 1993] Donald, B., Xavier, P., Canny, J., and Reif, J. (1993). Kinodynamic motion planning. *Journal of the ACM (JACM)*, 40:1048–1066.
- [Dornaika and Horaud, 1998] Dornaika, F. and Horaud, R. (1998). Simultaneous robot-world and hand-eye calibration. *IEEE transactions on Robotics and Automation*, 14(4):617–622.
- [Doucet et al., 2001] Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer.
- [Durrant-Whyte and Bailey, 2006] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110.
- [Durrant-Whyte, 1988] Durrant-Whyte, H. F. (1988). Uncertainty geometry in robotics. *IEEE Journal of Robotics and Automation*, 4(1):23–31.
- [Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- [Flanagan et al., 2006] Flanagan, J. R., Bowman, M. C., and Johansson, R. S. (2006). Control strategies in object manipulation tasks. *Current opinion in neurobiology*, 16(6):650–659.
- [Gadeyne and Bruyninckx, 2001] Gadeyne, K. and Bruyninckx, H. (2001). Markov techniques for object localization with force-controlled robots. In *10th Int’l Conf. on Advanced Robotics*.

- [Geraerts and Overmars, 2007] Geraerts, R. and Overmars, M. (2007). Creating high-quality paths for motion planning. *The International Journal of Robotics Research*, 26(8):845–863.
- [Goyal et al., 1991] Goyal, S., Ruina, A., and Papadopoulos, J. (1991). Planar sliding with dry friction part 1. limit surface and moment function. *Wear (Amsterdam, Netherlands)*, 143(2):307–330.
- [Gullapalli et al., 1994] Gullapalli, V., Franklin, J. A., and Benbrahim, H. (1994). Acquiring robot skills via reinforcement learning. *IEEE Control Systems*, 14(1):13–24.
- [Hart et al., 1968] Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107.
- [Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*, pages 132–150. Cambridge university press.
- [Hauser, 2010] Hauser, K. (2010). Randomized belief-space replanning in partially-observable continuous spaces. In *Algorithmic Foundations of Robotics IX*, pages 193–209. Springer.
- [Hauskrecht, 2000] Hauskrecht, M. (2000). Value-function approximations for partially observable markov decision processes. *Journal of artificial intelligence research*, 13:33–94.
- [Hebert et al., 2013] Hebert, P., Howard, T., Hudson, N., Ma, J., and Burdick, J. W. (2013). The next best touch for model-based localization. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 99–106. IEEE.
- [Holleman and Kavraki, 2000] Holleman, C. and Kavraki, L. E. (2000). A framework for using the workspace medial axis in prm planners. In *Robotics and Automation, 2000*.

- Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1408–1413. IEEE.
- [Horaud and Dornaika, 1995] Horaud, R. and Dornaika, F. (1995). Hand-eye calibration. *The international journal of robotics research*, 14(3):195–210.
- [Hsu et al., 2003] Hsu, D., Jiang, T., Reif, J., and Sun, Z. (2003). The bridge test for sampling narrow passages with probabilistic roadmap planners. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*.
- [Hsu et al., 2002] Hsu, D., Kindel, R., Latombe, J.-C., and Rock, S. (2002). Randomized kinodynamic motion planning with moving obstacles. *The International Journal of Robotics Research*, 21(3):233–255.
- [Hsu et al., 2006] Hsu, D., Latombe, J.-C., and Kurniawati, H. (2006). On the probabilistic foundations of probabilistic roadmap planning. *The International Journal of Robotics Research*, 25(7):627–643.
- [Javdani et al., 2013] Javdani, S., Klingensmith, M., Bagnell, J. A., Pollard, N. S., and Srini-vasa, S. S. (2013). Efficient touch based localization through submodularity. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1828–1835. IEEE.
- [Kant and Zucker, 1986] Kant, K. and Zucker, S. W. (1986). Toward efficient trajectory planning: the path-velocity decomposition. *The International Journal of Robotics Research*, 5(3):72–89.
- [Karan and Vukobratovic, 1994] Karan, B. and Vukobratovic, M. (1994). Calibration and accuracy of manipulation robot models-an overview. *Mechanism and Machine Theory*, 29(3):479.

- [Kavraki et al., 1996] Kavraki, L., Svestka, P., Latombe, J., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580.
- [Kavraki et al., 1998] Kavraki, L. E., Kolountzakis, M. N., and Latombe, J.-C. (1998). Analysis of probabilistic roadmaps for path planning. *Robotics and Automation, IEEE Transactions on*, 14(1):166–171.
- [Keselman et al., 2014] Keselman, L., Verriest, E., and Vela, P. A. (2014). Forage RRT—an efficient approach to task-space goal planning for high dimensional systems. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1572–1577.
- [Khalil and Dombre, 2004] Khalil, W. and Dombre, E. (2004). *Modeling, identification and control of robots*. Butterworth-Heinemann.
- [Kim and Likhachev, 2017] Kim, S.-K. and Likhachev, M. (2017). Parts assembly planning under uncertainty with simulation-aided physical reasoning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4074–4081. IEEE.
- [Knepper et al., 2013] Knepper, R. A., Layton, T., Romanishin, J., and Rus, D. (2013). Ikeabot: An autonomous multi-robot coordinated furniture assembly system. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 855–862. IEEE.
- [Koval et al., 2016] Koval, M. C., Pollard, N. S., and Srinivasa, S. S. (2016). Pre-and post-contact policy decomposition for planar contact manipulation under uncertainty. *The International Journal of Robotics Research*, 35(1-3):244–264.
- [Kuffner and LaValle, 2000] Kuffner, J. and LaValle, S. (2000). RRT-connect: An efficient approach to single-query path planning. In *IEEE International Conference on Robotics and Automation*.

- [Kuffner, 2004] Kuffner, J. J. (2004). Effective sampling and distance metrics for 3d rigid body path planning. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3993–3998. IEEE.
- [Kuffner Jr et al., 2002] Kuffner Jr, J. J., Kagami, S., Nishiwaki, K., Inaba, M., and Inoue, H. (2002). Dynamically-stable motion planning for humanoid robots. *Autonomous Robots*, 12(1):105–118.
- [Kwon et al., 2007] Kwon, J., Choi, M., Park, F. C., and Chun, C. (2007). Particle filtering on the euclidean group: framework and applications. *Robotica*, 25(6):725–737.
- [Ladd and Kavraki, 2004] Ladd, A. M. and Kavraki, L. E. (2004). Measure theoretic analysis of probabilistic path planning. *Robotics and Automation, IEEE Transactions on*, 20(2):229–242.
- [Latombe, 1999] Latombe, J.-C. (1999). Motion planning: A journey of robots, molecules, digital actors, and other artifacts. *The International Journal of Robotics Research*, 18(11):1119–1128.
- [Lau et al., 2011] Lau, M., Mitani, J., and Igarashi, T. (2011). Automatic learning of pushing strategy for delivery of irregular-shaped objects. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3733–3738. IEEE.
- [LaValle and Kuffner, 2001] LaValle, S. and Kuffner, J. (2001). Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400.
- [Lavalle, 1998] Lavalle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical report, Iowa State University.
- [LaValle and Kuffner Jr, 2000] LaValle, S. M. and Kuffner Jr, J. J. (2000). Rapidly-exploring random trees: Progress and prospects.

- [Lee et al., 2008] Lee, T., Leok, M., and McClamroch, N. H. (2008). Global symplectic uncertainty propagation on so (3). In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 61–66. IEEE.
- [Lefebvre et al., 2005] Lefebvre, T., Xiao, J., Bruyninckx, H., and De Gersem, G. (2005). Active compliant motion: a survey. *Advanced Robotics*, 19(5):479–499.
- [Lertkultanon and Pham, 2018] Lertkultanon, P. and Pham, Q.-C. (2018). A certified-complete bimanual manipulation planner. *IEEE Transactions on Automation Science and Engineering*, 15(3):1355–1368.
- [Levine et al., 2016] Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373.
- [Levine et al., 2015] Levine, S., Wagener, N., and Abbeel, P. (2015). Learning contact-rich manipulation skills with guided policy search. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 156–163. IEEE.
- [Li and Bainum, 1990] Li, F. and Bainum, P. M. (1990). Numerical approach for solving rigid spacecraft minimum time attitude maneuvers. *Journal of Guidance, Control, and Dynamics*, 13(1):38–45.
- [Lindemann and LaValle, 2004] Lindemann, S. R. and LaValle, S. M. (2004). Incrementally reducing dispersion by increasing voronoi bias in RRTs. In *Robotics and Automation, 2004. Proceedings. ICRA '04. IEEE International Conference on*, volume 4, pages 3251–3257.
- [Liou et al., 1993] Liou, Y., Lin, P. P., Lindeke, R. R., and Chiang, H.-D. (1993). Tolerance specification of robot kinematic parameters using an experimental design technique. *Robotics and computer-integrated manufacturing*, 10(3):199.

- [Littman et al., 1995] Littman, M. L., Cassandra, A. R., and Kaelbling, L. P. (1995). Learning policies for partially observable environments: Scaling up. In *Machine Learning Proceedings 1995*, pages 362–370. Elsevier.
- [Long et al., 2013] Long, A. W., Wolfe, K. C., Mashner, M. J., and Chirikjian, G. S. (2013). The banana distribution is gaussian: a localization study with exponential coordinates. *Robotics: Science and Systems VIII*, page 265.
- [Lozano-Pérez and Wesley, 1979] Lozano-Pérez, T. and Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570.
- [Luo and Hauser, 2014] Luo, J. and Hauser, K. (2014). An empirical study of optimal motion planning. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 1761–1768.
- [Ma et al., 2016] Ma, Q., Li, H., and Chirikjian, G. S. (2016). New probabilistic approaches to the ax=xb hand-eye calibration without correspondence. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 4365–4371. IEEE.
- [Mason, 1986] Mason, M. T. (1986). Mechanics and planning of manipulator pushing operations. *The International Journal of Robotics Research*, 5(3):53–71.
- [Melchior and Simmons, 2007] Melchior, N. A. and Simmons, R. (2007). Particle rrt for path planning with uncertainty. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1617–1624. IEEE.
- [Menq and Borm, 1988] Menq, C.-H. and Borm, J.-H. (1988). Statistical measure and characterization of robot errors. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 926–931. IEEE.

- [Meriçli et al., 2015] Meriçli, T., Veloso, M., and Akın, H. L. (2015). Push-manipulation of complex passive mobile objects using experimentally acquired motion models. *Autonomous Robots*, 38(3):317–329.
- [Murray et al., 1994a] Murray, R. M., Li, Z., and Sastry, S. S. (1994a). *A mathematical introduction to robotic manipulation*. CRC press.
- [Murray et al., 1994b] Murray, R. M., Li, Z., Sastry, S. S., and Sastry, S. S. (1994b). *A mathematical introduction to robotic manipulation*, pages 403–433. CRC press.
- [Nguyen and Pham, 2017] Nguyen, H. and Pham, Q.-C. (2017). Touch-based object localization in cluttered environments. *arXiv preprint arXiv:1709.09317*.
- [Nguyen and Pham, 2018] Nguyen, H. and Pham, Q.-C. (2018). On the covariance of x in $ax=xb$. *IEEE Transactions on Robotics*, 34(6):1651–1658.
- [Nubiola and Bonev, 2013] Nubiola, A. and Bonev, I. A. (2013). Absolute calibration of an abb irb 1600 robot using a laser tracker. *Robotics and Computer-Integrated Manufacturing*, 29(1):236–245.
- [Papazov and Burschka, 2010] Papazov, C. and Burschka, D. (2010). An efficient ransac for 3d object recognition in noisy and occluded scenes. In *Asian Conference on Computer Vision*, pages 135–148. Springer.
- [Park and Ravani, 1995] Park, F. and Ravani, B. (1995). Bezier curves on riemannian manifolds and lie groups with kinematics applications. *Journal of Mechanical Design*, 117(1):36–40.
- [Park, 1991] Park, F. C. (1991). *The optimal kinematic design of mechanisms*. PhD thesis, Havard University.
- [Park, 1995] Park, F. C. (1995). Distance metrics on the rigid-body motions with applications to mechanism design. *Journal of Mechanical Design*, 117(1):48–54.

- [Park and Martin, 1994] Park, F. C. and Martin, B. J. (1994). Robot sensor calibration: solving $ax = xb$ on the euclidean group. *IEEE Transactions on Robotics and Automation*, 10(5).
- [Park and Ravani, 1997] Park, F. C. and Ravani, B. (1997). Smooth invariant interpolation of rotations. *ACM Transactions on Graphics (TOG)*, 16(3):277–295.
- [Petrovskaya and Khatib, 2011a] Petrovskaya, A. and Khatib, O. (2011a). Global localization of objects via touch. *IEEE Transactions on Robotics*, 27(3):569–585.
- [Petrovskaya and Khatib, 2011b] Petrovskaya, A. and Khatib, O. (2011b). Global localization of objects via touch. *IEEE Transactions on Robotics*, 27(3):569–585.
- [Petrovskaya et al., 2006] Petrovskaya, A., Khatib, O., Thrun, S., and Ng, A. Y. (2006). Bayesian estimation for autonomous object manipulation based on tactile sensors. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 707–714. IEEE.
- [Pham, 2014] Pham, Q.-C. (2014). A general, fast, and robust implementation of the time-optimal path parameterization algorithm. *IEEE Transactions on Robotics*, 6:1533–1540.
- [Pham et al., 2017] Pham, Q.-C., Caron, S., Lertkultanon, P., and Nakamura, Y. (2017). Admissible velocity propagation: Beyond quasi-static path planning for high-dimensional robots. *The International Journal of Robotics Research*, 36(1):44–67.
- [Pham et al., 2013] Pham, Q.-C., Caron, S., and Nakamura, Y. (2013). Kinodynamic planning in the configuration space via admissible velocity propagation. In *Proceedings of Robotics: Science and Systems*.
- [Phillips-Grafflin and Berenson, 2017] Phillips-Grafflin, C. and Berenson, D. (2017). Planning and resilient execution of policies for manipulation in contact with actuation uncertainty. *arXiv preprint arXiv:1703.10261*.

- [Platt Jr et al., 2010] Platt Jr, R., Tedrake, R., Kaelbling, L., and Lozano-Perez, T. (2010). Belief space planning assuming maximum likelihood observations. In *Proceedings of the Robotics: Science and Systems Conference, 6th*.
- [Prentice and Roy, 2010] Prentice, S. and Roy, N. (2010). The belief roadmap: Efficient planning in linear pomdps by factoring the covariance. In *Robotics Research*, pages 293–305. Springer.
- [Raguram et al., 2008] Raguram, R., Frahm, J.-M., and Pollefeys, M. (2008). A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. *Computer Vision–ECCV 2008*, pages 500–513.
- [Reif, 1979] Reif, J. H. (1979). Complexity of the mover’s problem and generalizations. In *Proceedings of the 20th Annual IEEE Conference on Foundations of Computer Science*, pages 421–427.
- [Renders et al., 1991] Renders, J.-M., Rossignol, E., Becquet, M., Hanus, R., et al. (1991). Kinematic calibration and geometrical parameter identification for robots. *IEEE Transactions on robotics and automation*, 7(6):721–732.
- [Sallinen, 2003] Sallinen, M. (2003). *Modelling and estimation of spatial relationships in sensor-based robot workcells*. PhD thesis, University of Oulu (Oulu, Finland).
- [Sánchez and Latombe, 2003] Sánchez, G. and Latombe, J.-C. (2003). A single-query bi-directional probabilistic roadmap planner with lazy collision checking. In *Robotics Research*, pages 403–417.
- [Schmidt and Niemann, 2008] Schmidt, J. and Niemann, H. (2008). Data selection for hand-eye calibration: a vector quantization approach. *The International Journal of Robotics Research*, 27(9):1027–1053.

- [Schmidt et al., 2003] Schmidt, J., Vogt, F., and Niemann, H. (2003). Robust hand-eye calibration of an endoscopic surgery robot using dual quaternions. In *Joint Pattern Recognition Symposium*, pages 548–556. Springer.
- [Schwartz and Sharir, 1983] Schwartz, J. T. and Sharir, M. (1983). On the “piano movers” problem. ii. general techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics*, 4(3):298–351.
- [Shi et al., 2005] Shi, F., Wang, J., and Liu, Y. (2005). An approach to improve online hand-eye calibration. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 647–655. Springer.
- [Shin and McKay, 1985] Shin, K. and McKay, N. (1985). Minimum-time control of robotic manipulators with geometric path constraints. *Automatic Control, IEEE Transactions on*, 30(6):531–541.
- [Shiu and Ahmad, 1989] Shiu, Y. C. and Ahmad, S. (1989). Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $ax = xb$. *IEEE Transactions on Robotics and Automation*, 5(1).
- [Shkolnik et al., 2009] Shkolnik, A., Walter, M., and Tedrake, R. (2009). Reachability-guided sampling for planning under differential constraints. In *Robotics and Automation, 2009. Proceedings. ICRA '09. IEEE International Conference on*, pages 2859–2865.
- [Siciliano and Khatib, 2016] Siciliano, B. and Khatib, O. (2016). *Springer handbook of robotics*. Springer.
- [Sieverling et al., 2017] Sieverling, A., Eppner, C., Wolff, F., and Brock, O. (2017). Interleaving motion in contact and in free space for planning under uncertainty. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 4011–4073. IEEE.

- [Smith et al., 2003] Smith, P., Drummond, T., and Roussopoulos, K. (2003). Computing map trajectories by representing, propagating and combining pdfs over groups. In *null*, page 1275. IEEE.
- [Smith et al., 1990] Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer.
- [Smith and Cheeseman, 1986] Smith, R. C. and Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *IEEE Journal of Robotics and Automation*, 5(4):56–68.
- [Stentz, 1993] Stentz, A. (1993). Optimal and efficient path planning for unknown and dynamic environments. Technical report, DTIC Document.
- [Strobl and Hirzinger, 2006] Strobl, K. H. and Hirzinger, G. (2006). Optimal hand-eye calibration. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4647–4653. IEEE.
- [Su and Lee, 1991] Su, S. and Lee, C. (1991). Uncertainty manipulation and propagation and verification of applicability of actions in assembly tasks. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2471–2476.
- [Su and Lee, 1992a] Su, S.-F. and Lee, C. G. (1992a). Manipulation and propagation of uncertainty and verification of applicability of actions in assembly tasks. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6):1376–1389.
- [Su and Lee, 1992b] Su, S.-F. and Lee, C. G. (1992b). Manipulation and propagation of uncertainty and verification of applicability of actions in assembly tasks. *Systems Science and Cybernetics, IEEE Transactions on*, 22(6):1376–1389.

- [Suarez-Ruiz and Pham, 2016] Suarez-Ruiz, F. and Pham, Q.-C. (2016). A framework for fine robotic assembly. In *IEEE International Conference on Robotics and Automation*.
- [Suárez-Ruiz et al., 2018] Suárez-Ruiz, F., Zhou, X., and Pham, Q.-C. (2018). Can robots assemble an ikea chair? *Science Robotics*, 3(17):eaat6385.
- [Taylor, 1976] Taylor, R. H. (1976). *The Synthesis of Manipulator Control Programs from Task-level Specifications*. PhD thesis, Stanford University, Stanford, CA, USA. AAI7707174.
- [Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic robotics*. MIT press.
- [Tsai and Lenz, 1989] Tsai, R. and Lenz, R. K. (1989). A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358.
- [Žlajpah, 1996] Žlajpah, L. (1996). On time optimal path control of manipulators with bounded joint velocities and torques. In *Robotics and Automation, 1996. Proceedings. ICRA '96, IEEE International Conference on*, volume 2, pages 1572–1577.
- [Vahrenkamp et al., 2009] Vahrenkamp, N., Berenson, D., Asfour, T., Kuffner, J., and Dillmann, R. (2009). Humanoid motion planning for dual-arm manipulation and re-grasping task. In *Intelligent Robots and Systems (IROS), 2009 IEEE/RSJ International Conference on*, pages 2464–2470.
- [Van Den Berg et al., 2011] Van Den Berg, J., Abbeel, P., and Goldberg, K. (2011). Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, 30(7):895–913.

- [Van Wyk et al., 2018] Van Wyk, K., Culleton, M., Falco, J., and Kelly, K. (2018). Comparative peg-in-hole testing of a force-based manipulation controlled robotic hand. *IEEE Transactions on Robotics*, 34(2):542–549.
- [Veitschegger and Wu, 1987] Veitschegger, W. and Wu, C.-h. (1987). A method for calibrating and compensating robot kinematic errors. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, volume 4, pages 39–44. IEEE.
- [Vezzani et al., 2017] Vezzani, G., Pattacini, U., Battistelli, G., Chisci, L., and Natale, L. (2017). Memory unscented particle filter for 6-dof tactile localization. *IEEE Transactions on Robotics*.
- [Wahrburg et al., 2014] Wahrburg, A., Zeiss, S., Matthias, B., and Ding, H. (2014). Contact force estimation for robotic assembly using motor torques. In *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, pages 1252–1257. IEEE.
- [Walker and Salisbury, 2008] Walker, S. and Salisbury, J. K. (2008). Pushing using learned manipulation maps. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3808–3813. IEEE.
- [Wang, 1992] Wang, C. C. (1992). Extrinsic calibration of a vision sensor mounted on a robot. *IEEE Transactions on Robotics and Automation*, 8(2):161–175.
- [Wang and Chirikjian, 2008] Wang, Y. and Chirikjian, G. S. (2008). Nonparametric second-order theory of error propagation on motion groups. *The International journal of robotics research*, 27(11-12):1258–1273.
- [Wilmarth et al., 1999] Wilmarth, S. A., Amato, N. M., and Stiller, P. F. (1999). MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *Robotics and Automation, 1999. Proceedings. ICRA '99. IEEE International Conference on*, volume 2, pages 1024–1031.

- [Wirnshofer et al., 2018] Wirnshofer, F., Schmitt, P. S., Feiten, W., Wichert, G. v., and Burgard, W. (2018). Robust, compliant assembly via optimal belief space planning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–5. IEEE.
- [Xian et al., 2017] Xian, Z., Lertkultanon, P., and Pham, Q.-C. (2017). Closed-chain manipulation of large objects by multi-arm robotic systems. *IEEE Robotics and Automation Letters*, 2(4):1832–1839.
- [Yershova et al., 2005] Yershova, A., Jaillet, L., Siméon, T., and LaValle, S. M. (2005). Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain. In *Robotics and Automation, 2005. Proceedings. ICRA '05, IEEE International Conference on*, pages 3856–3861.
- [Zhou et al., 2016] Zhou, J., Paolini, R., Bagnell, J. A., and Mason, M. T. (2016). A convex polynomial force-motion model for planar sliding: Identification and application. *arXiv preprint arXiv:1602.06056*.
- [Zhou et al., 2017] Zhou, J., Paolini, R., Johnson, A. M., Bagnell, J. A., and Mason, M. T. (2017). A probabilistic planning framework for planar grasping under uncertainty. *IEEE Robotics and Automation Letters*, 2(4):2111–2118.
- [Zhuang and Roth, 1996] Zhuang, H. and Roth, Z. S. (1996). *Camera-aided robot calibration*. CRC press.