

**POSITION AND ATTITUDE TRACKING
USING $SO(3)$ MANIFOLD**

BY

HAMMAM KHALED MOHAMMED SALEM

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE
In
ELECTRICAL ENGINEERING

DECEMBER 2023

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **HAMMAM KHALED MOHAMMED SALEM** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**.

Thesis Committee

Prof. Ali Hussein Muqaibel (Adviser)

Prof. Tareq Y. Al-Naffouri (Member)

Prof. Fahhad H. Al Harbi (Member)

Dr. Mohammad AlMuhaini
Department Chairman

Prof. Suliman S. Al-Homidan
Dean of Graduate Studies

Date



©Hammam K. Salem
2023

Dedication

This work stands as a testament not just to my own endeavors, but to the unwavering support and endless love that fueled them. To my dearest parents, a lifetime of sacrifices paved the way for this moment. Your tireless work ethic, unwavering faith in my dreams, and quiet encouragement in times of doubt are etched in every page. Thank you for nurturing the seeds of curiosity that blossomed into this pursuit.

My beloved wife, your presence has been the anchor in my academic voyage. Your patience during late nights, understanding during moments of frustration, and unwavering belief in my ability have been my guiding light. You bring joy and laughter to even the most arduous tasks, reminding me of what truly matters beyond the thesis pages. You are my inspiration, my solace, and the sun that warms my academic journey.

And to my precious Khaled, my son, you are the sweetest reward of all. Your laughter is the melody that soothes my weary mind, your innocent curiosity a constant reminder of the boundless joy in learning. This achievement is not just mine, but a bridge to a brighter future I hope to build for you. Your existence fuels my ambition and defines the meaning of this pursuit.

To each of you, a heartfelt thank you. This work is an expression of your love, your faith, and your unwavering belief in me. May it stand as a beacon of gratitude for the endless ways you have illuminated my path.

With all my love,
Hammam

ACKNOWLEDGMENTS

First and foremost, I thank Allah for what he has facilitated for me during my master's degree. Because whoever does not thank people does not thank Allah, I would like to express my sincere gratitude to my thesis supervisor, Prof. Ali Muqaibel, for his patience, guidance, and invaluable support and encouragement throughout my thesis. He has been an advisor, a mentor, and a great friend.

I would also like to thank Prof. Tareq Al-Naffouri, a member committee, for his guidance. I am deeply indebted to him for his willingness to share his knowledge, material, and expertise. In addition, I would like to thank Prof. Fahhad Al Harbi, a member committee, for his valuable feedback and suggestions. His insights and guidance were essential to the completion of this thesis.

I would like also to thank my parents for their love and unwavering support throughout my academic journey. To my wife and child, thank you for your love, patience, and encouragement. I could not have done this without you.

I would like to express my sincere gratitude to King Fahd University of Petroleum and Minerals, the Department of Electrical Engineering, IRC Communication Systems and Sensing, and the Information Science Lab at King Abdullah University of Science and Technology (KAUST) for providing me with the resources, support, and guidance I needed to complete this dissertation.

I would be remiss if I did not extend my sincere gratitude to my dear friend, Mohannad Ahmad, for his invaluable contributions to our discussions about this thesis. His expertise has played a pivotal role in deepening my understanding of the subject matter. I am truly grateful to all of the people who have helped me along the way. This thesis would not have been possible without your support.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF SYMBOLS	xiii
LIST OF ABBREVIATIONS	xv
ABSTRACT (ENGLISH)	xvii
ABSTRACT (ARABIC)	xix
CHAPTER 1 INTRODUCTION	1
1.1 Thesis Contribution	5
1.2 Thesis Outlines	6
CHAPTER 2 LITERATURE REVIEW	7
2.1 Kalman Filters	8
2.2 Riemannian Manifold	9
2.3 Indoor Positioning	12
CHAPTER 3 TECHNICAL BACKGROUND	18
3.1 Target Tracking	19
3.1.1 Position Tracking	19
3.1.2 Orientation Tracking	21

3.2	Kalman Filter and its Variations	23
3.3	Riemannian Manifold	25
3.3.1	Motivation	25
3.3.2	Smooth Manifold	26
3.3.3	Smooth Embedded Submanifolds of Euclidean Space	27
3.3.4	Tangent Vector and Tangent Space	28
3.3.5	Smooth Map on Embedded Smooth Manifold	29
3.3.6	Riemannian Manifold	30
3.3.7	Orthogonal Projection	31
3.3.8	Retraction	31
3.3.9	Vector Transport	32
3.3.10	Special Orthogonal Group Manifold	32
3.4	The Interplay of the Aforementioned Tools and Tracking	34
CHAPTER 4 SYSTEM MODEL AND PROBLEM FORMULATION		36
4.1	System Model	36
4.2	Problem Formulation	39
CHAPTER 5 PROPOSED ALGORITHMS		46
5.1	Pseudo Codes	49
CHAPTER 6 RESULTS AND DISCUSSION		53
6.1	Simulation Setup	54
6.2	Numerical Results - Static Case	55
6.3	Numerical Results - Dynamic Case	57
CHAPTER 7 CONCLUSION		65
7.1	Summary of Results	65
7.2	Future Directions	66
REFERENCES		67
VITAE		80

LIST OF TABLES

2.1	Comparison between some indoor localization works featuring multiple receivers	16
3.1	Riemannian geometry for $SO(3)$ manifold	34
6.1	Position RMSE for U-path	58
6.2	Position RMSE for Stair-path	58

LIST OF FIGURES

3.1	Defining angles of rotation between BCS and GCS where the solid axes, dashed axes, ψ , θ , and ϕ represent the GCS, BCS, angles of rotation around z -axis, y -axis, and x -axis, respectively.	21
3.2	Different geometric shapes to demonstrate the manifold concept, from left: cube (not manifold), Boy's surface, torus, and sphere	27
3.3	Manifold local defining functions: h smoothly maps an open set U (a disk) to R^k ($k = 1$); exactly the intersection of U with the set $\mathcal{M}$ (a curve) is mapped to 0 [1].	28
3.4	Geometric representation of sphere manifold ($\mathcal{S}^{n-1}$) with its tangent space and retraction	29
3.5	Demonstration of retraction operation to allow optimization over smooth manifold [1]	32
4.1	A system diagram to show the moving target with its components, where the dots line, brown triangle, purple asterisks, green axes, red cross, cyan circle, and magenta diamond refer to the moving path, isosceles triangle, transmitters, BCS, target centroid, IMU, and beacons, respectively.	37
5.1	Descriptive diagram of the proposed Algorithm 1 (EKFRie) and proposed Algorithm 2 (UKFRie)	50
6.1	The four true paths under consideration, where the red diamonds refer to the beacons	56
6.2	RMSE vs different values of $1/\sigma_r$ dB for the static case	57

6.3	Estimated path of different algorithms for different σ_r	59
6.4	Positions' RMSE vs different values of $1/\sigma_r$ dB	60
6.5	The <i>RMSE</i> ratio of the Riemannian algorithms vs conventional algorithms for the position under different values of $1/\sigma_r$ dB	61
6.6	Orientation's RMSE vs different values of $1/\sigma_r$ dB	62
6.7	RMSE of U-path vs IMU's updating rate for $\sigma_r^2 = 1 \times 10^{-1}$	63
6.8	RMSE of U-path vs IMU's updating rate for $\sigma_r^2 = 1 \times 10^{-3}$	63
6.9	CDF of the RMSE for Bridge-path for different values of σ_r^2	64

LIST OF SYMBOLS

a	Altitude of the isosceles triangle
$\mathbf{a}_c$	Acceleration of the target's centroid in GCS, $\in \mathbb{R}^3$
$\mathbf{b}_j$	Position of j^{th} receiver in GCS where j is integer number, $\in \mathbb{R}^3$
d	Base of the isosceles triangle
$\mathcal{E}$	Euclidean space
∇	Gradient
$\vec{h}$	Vector in the embedding space of the $\text{SO}(3)$ manifold
$\hbar$	Local defining function
L	Dimension of the state vector
M	Number of anchors
$\mathcal{M}$	Manifold in general, but we reserve it for smooth manifold
ω	Angular velocity of the target's centroid, $\in \mathbb{R}^3$
$\mathbf{p}_c$	Position of the target's centroid in GCS, $\in \mathbb{R}^3$
$Proj_x(\vec{h})$	Projector operator
Ψ	Column vector contains the position and velocity of target's centroid (i.e. $\Psi = [\mathbf{p}_c^\top \quad \mathbf{v}_c^\top]^\top$), $\in \mathbb{R}^6$
$\mathbf{Q}$	Covariance matrix of $\mathbf{u}$

$\mathbf{R}$	Rotation matrix, $\in \mathbb{R}^{3 \times 3}$
r_{ij}	Element located at i^{th} row and j^{th} column of $\mathbf{R}$
$\mathcal{R}_x(\vec{v})$	Retraction operator
T	Time step used in discretization
Θ	Column-major order of $\mathbf{R}$ (i.e. $\Theta = [R_1^\top \ R_2^\top \ R_3^\top]^\top$), $\in \mathbb{R}^9$
$\mathbf{u}$	Column vector contains the angular velocity and the acceleration of the centroid in BCS (i.e. $\mathbf{u} = [\omega^\top \ \mathbf{a}_c^{b^\top}]^\top$), $\in \mathbb{R}^6$
$\vec{v}$	Tangent vector
$\mathbf{v}_c$	Velocity of the target's centroid in GCS, $\in \mathbb{R}^3$
$\mathbf{X}$	Column vector contains states of proposed model such that $\mathbf{X} = [\Theta^\top \ \Psi^\top]^\top$, $\in \mathbb{R}^{15}$
$\hat{\mathbf{y}}_{imu}$	Column vector contains estimated positions of the transmitters depend on state equation, $\in \mathbb{R}^9$
$\mathbf{y}_{ms}$	Column vector contains estimated positions of the transmitters depending on the measurement system i.e. $\mathbf{y} = [\mathbf{p}_1^\top \ \mathbf{p}_2^\top \ \mathbf{p}_3^\top]^\top$, $\in \mathbb{R}^9$
$\mathbf{Z}$	Covariance matrix of $\mathbf{y}$

LIST OF ABBREVIATIONS

AoA	angel of arrival
AWGN	additive white Gaussian noise
BCS	body coordinate system
EKF	extended Kalman filter
EKFRie	extended Kalman filter with Riemannian optimization
GCS	global coordinate system
GPS	global positioning system
ICKF	interpolatory Cubature Kalman Filter
IMU	inertial measurement unit
IoT	Internet of Things
KF	Kalman filter
l.l.m.s.e	linear least mean square error estimator
LS	least squares
ML	machine learning
MS	measurement system
MSE	mean square error

PD	photodiodes
RMSE	root mean square error
RSS	received signal strength
SNR	signal to noise ratio
$\text{SO}(n)$	special orthogonal group manifold
SVD	singular value decomposition
TDoA	time difference of arrival
ToA	time of arrival
ToF	time of flight
UAV	unmanned aerial vehicle
UKF	unscented Kalman filter
UKFRie	unscented Kalman filter with Riemannian optimization
USS	ultrasonic system
UWB	ultra-wideband
ZOH	zero-order hold

THESIS ABSTRACT

NAME: Hammam Khaled Mohammed Salem
TITLE OF STUDY: Position and Attitude Tracking using $SO(3)$ Manifold
MAJOR FIELD: Electrical Engineering
DATE OF DEGREE: December, 2023

Driven by technological breakthroughs, indoor positioning and tracking have gained importance in a wide range of applications. Numerous successful applications, including the Internet of Things (IoT), robotics, and unmanned aerial vehicles (UAVs), heavily depend on indoor positioning and tracking. The rapid advancements in these areas have fueled the demand for high-performance positioning systems. However, the area of indoor positioning and tracking continues to be a fertile ground for research, with no single approach dominating the landscape.

In an effort to tackle some of the inherent challenges associated with indoor tracking, this study explores whether incorporating the $SO(3)$ manifold structure of the rotation matrix into Riemannian optimization can enhance the performance of the extended Kalman filter (EKF) and unscented Kalman filter (UKF) in tracking the 3D position and orientation of a moving target within an indoor environment. The

effect of adopting an isosceles triangle manifold of the transmitter's configuration is also studied.

Our results demonstrate that the proposed extended Kalman filter with Riemannian (EKFRie) and unscented Kalman filter with Riemannian (UKFRie) algorithms consistently outperform the conventional EKF and UKF in terms of position accuracy and often in terms of orientation accuracy. For the considered environment and parameters, while the conventional EKF and UKF achieved an RMSE of $1.08m$ and $1.39m$, respectively, the proposed EKFRie and UKFRie algorithms achieved a significantly lower RMSE of $0.31m$ and $0.39m$, respectively. This improvement in accuracy is further corroborated by the *90th* percentile of the RMSE, which for EKFRie and UKFRie falls below $0.22m$ and $0.27m$, respectively, compared to $0.52m$ and $0.67m$ for EKF and UKF, respectively. These results demonstrate the enhanced performance of the proposed algorithms.

ملخص الرسالة

الاسم: همام خالد محمد سالم

عنوان الدراسة: تتبع الموقع والاتجاه باستعمال المتشعب الريماني $SO(3)$

التخصص: قسم الهندسة الكهربائية

تاريخ الدرجة العلمية: ديسمبر 2023

بفضل التقدم التكنولوجي، أصبح التتبع الداخلي (تتبع موقع جسم واتجاهه في بيئة معزولة عن الأقمار الصناعية) من الأهمية بمكان للعديد من التقنيات الحديثة. فلا يخفى على المتابع مدى اعتماد العديد من التقنيات الحديثة على التتبع الداخلي في أداء مهامها المختلفة، كما هو الحال في إنترنت الأشياء والروبوتات والطائرات بدون طيار. وقد أدى التقدم السريع في هذه المجالات إلى زيادة الطلب على أنظمة تحديد المواقع عالية الجودة. وبالرغم من ذلك فلا يزال مجال التتبع الداخلي يمثل أرضاً خصبة تستهوي الباحثين والمهتمين لاقتحاره لتطبيقات مهيمنة يمكن اعتبارها حلاً جذرياً لهذه المشكلة.

وفي محاولة منها لسد شيء من الثغرات في هذا المجال، تسعى هذه الأطروحة لبحث ما إذا كان بالإمكان الانتفاع من بنية مصفوفة الدوران، والتي تنتمي للفضاء الريماني، لتحسين أداء التتبع وتقليل نسبة الخطأ. من أجل تحقيق ذلك تبنت الأطروحة دمج التحسين الريماني بمرشحين هما: مرشح كالمن المطور (EKF) ومرشح كالمن المحسن (UKF)، وقامت بدراسة مدى تأثير هذا الدمج على دقة التتبع وتقليل مقدار الخطأ. بالإضافة لذلك، تسعى هذه الأطروحة أيضاً لدراسة ما إذا كان تبني البنية الريمانية للمرسلات المكوّنة لمثلث متطابق الساقين قد تساعد في تحسين التتبع وتقليل الأخطاء.

أظهرت نتائجنا التي قمنا بها عن طريق برنامج المحاكاة (مانالاب) أن الخوارزمية المقترحة باستعمال التحسين الريماني دائماً ما تتفوق على الخوارزمية التقليدية لمرشحي كالمن آنفي الذكر سواء في ما يتعلق بتتبع الموقع أو الاتجاه. فبينما حقق مرشحي كالمن التقليديين متوسط خطأ بمقدار

1.08م و 1.39م، حققت النسختين المحدثتين منهما متوسط خطأ بمقدار 0.31م و 0.39م على التوالي. وفيما كان 90% من متوسط الخطأ في

الخوارزميتين التقليديتين دون 0.52م و 0.67م، كان في نظيريه المحسنين دون 0.22م و 0.27م على التوالي. وبهذا يتبين أن تبني التحسين الريماني

قد أدى لتحسين دقة التتبع في الحالات محل الدراسة.

CHAPTER 1

INTRODUCTION

With the technological progress that the world is witnessing, positioning and tracking appear as important factors of this progress [2]. Positioning (sometimes called localization) means estimating the coordinates and orientation of an object in a static case. If an object is moving and we seek to estimate its changing position and orientation; this is known as tracking. However, if the context provides clarity, it is sometimes acceptable to use positioning and tracking interchangeably.

Many applications of the current technological progress rely on positioning and tracking. The Internet of Things (IoT) is one of the prominent areas of research [3], [4] where the rapid development has created a heightened demand for efficient positioning and tracking systems [5]. Smart parking [6], smart cities [7], and asset Management [8] are some examples of localization-based IoT applications.

The robotics revolution is another example where an enormous number of researchers are working to boost the robot's role in human life. But to effectively control these smart assistants, we need to accurately track their positions and orientations [9], [10]. The precise localization technology is crucial for diverse robotics applications,

from delivering packages [11] to navigating self-driving cars and automating tasks in warehouses [12]. Even in delicate surgical operations, robot positioning plays a vital role in ensuring accuracy and safety [13]. As robots continue to evolve, precise localization will be the cornerstone of their success.

The third example that shows the importance of positioning and tracking is unmanned aerial vehicle (UAV) and drone activities due to their roles in national security, surveillance, media, entertainment, etc. Enhancement in positioning and tracking systems will encourage additional applications.

Tracking the orientation of a moving target in an indoor environment is crucial for various applications in areas like IoT [14] and industrial automation [15]. Also, it is important for enhancing precision, safety, and efficiency in numerous scenarios. For example, knowing the target's orientation allows for precise trajectory prediction, enabling robots and autonomous vehicles to navigate around obstacles and plan efficient paths within complex indoor spaces. Additionally, real-time orientation data helps predict potential collisions between moving objects in confined spaces. This is vital for ensuring safety in warehouses with forklifts or factories with autonomous robots operating near humans.

Positioning and tracking can be categorized into self and network frameworks. In the self framework, the object itself performs the processing. Conversely, in the network framework, the processing occurs in the network infrastructure, where the ranges between the object and nodes are estimated based on the transmitted signal. More details about these types of tracking are provided in section 3.1.

Localization and tracking can also be broadly categorized into two categories:

outdoor and indoor [16], [17]. Outdoor tracking deals with the localization problem in open spaces as several techniques can be used to achieve this. For instance, one of the most popular methods is based on satellites to conduct outdoor tracking. This type of localization problem has been extensively researched and much work has been done over the last five decades. Specifically, with the advent of the global positioning system (GPS); this problem is solved in many aspects [18]. However, there is some room for enhancing the accuracy and increasing the update rate.

In indoor tracking, which is the interest of this thesis, the process depends mainly on systems that serve limited areas like ultrasonic system (USS) and built-in sensors like inertial measurement unit (IMU). This kind of tracking is challenging and it is a trend research topic with no prevailing solution so far [19]. Some of the indoor tracking challenges include the lack of reliable GPS signals due to signal penetration issues [20], multipath caused by walls and furniture [21], and the demand for high accuracy, which is often in decimeter-level [22].

It is possible to employ either a single transceiver or multiple transceivers to locate the object. In literature, many works have used multiple receivers for various purposes [23], [24]. Some works used multiple receivers to reduce the computation complexity in estimating the moving target orientation as in [25], [26]. Yeonseok Par *et al.* [27] used a single microphone and multiple receivers to overcome the complexity of synchronization in the conventional sound source positioning system. Three ultra-wideband (UWB) tags were used by [28] for mapping to decrease the probability of unobservability between the beacon and the target. Multiple receivers were also used to enhance tracking accuracy [26], [29].

Alejandro Correa *et al.* [26] used a multiple-receivers approach combined with machine learning (ML) algorithms to track a pedestrian in the indoor environment. Their work achieved 90% of root mean square error (RMSE) below $2m$ and $2.8m$ for areas around $100m^2$ and $533.12m^2$ respectively. Multiple optical receivers were used by [23] for tracking in the indoor scenario and achieved position error around $0.17m$ at large luminous flux. The RMSE of $0.89m$ is achieved by utilizing the Kalman filter and machine learning [30]. The authors in [30] used ML (ridge regression) combined with the Kalman filter algorithm in a wireless sensor networks environment. Multiple receivers are also used in outdoor tracking to enhance performance. For example, two GPS receivers positioned 3 meters apart from each other were utilized by [29] to enhance their performance. Also, [29] showed that two receivers provided precise estimates of location with low cost compared with one expensive receiver.

The tracking performance is influenced by a number of factors, including the formulation of the optimization problem and its constraints. Instead of directly monitoring the rotation angles, this thesis opts to express them as a rotation matrix and track the matrix itself. This work exploits the fact that the rotation matrix belongs to the special orthogonal group manifold ($SO(n)$), the subgroup of the Riemannian manifold where n refers to the size of the square matrix, to explore whether the performance of tracking can be improved by using Riemannian geometry. A more detailed discussion of the contributions made by this thesis is presented in the next section

1.1 Thesis Contribution

The work aims to investigate how adopting $SO(3)$ manifold structure can aid in improving indoor tracking accuracy when the isosceles triangular transmitter configuration is used in the static and dynamic setting. Three transmitters are employed for both orientation estimation and enhanced position accuracy. This work leverages recent advances in Riemannian geometry [31], [1], [32], [33] to apply Riemannian optimization in the tracking problem. The extended Kalman filter (EKF) and unscented Kalman filter (UKF) are used for comparison purposes. The contributions of this work related to localization accuracy can be outlined as follows:

1. Propose the incorporation of the Riemannian optimization by means of $SO(3)$ manifold with the EKF and quantify the performance improvement in terms of RMSE,
2. Explore whether the incorporation of the Riemannian manifold by means of $SO(3)$ manifold can improve the performance of the UKF in terms of RMSE,
3. Examine the impact of utilizing the isosceles triangle manifold of the three transmitters on the proposed algorithms under a specific range of the measurement reading's noise variance and specific values of update rate,
4. Evaluate the influence of varying values of measurement reading's noise variance and IMU's update rates on the proposed algorithms.

The outcomes of this thesis will be presented in both conference and journal papers

submitted to IEEE Insha' Allah. The subsequent section outlines the overall framework of the thesis.

1.2 Thesis Outlines

The rest of this thesis is organized as follows: Chapter 2 provides the reader with a literature review about enhancing the performance of KFs, localization by using Riemannian geometry, and merging KF with Riemannian optimization to handle tracking problems. The essential theoretical background about tracking and navigation, the Kalman filter and its variations, and Riemannian manifold optimization are presented in Chapter 3. Readers seeking a deeper understanding of Kalman filters, tracking, and Riemannian optimization before delving into the literature review are encouraged to cover Chapter 3 prior to proceeding to Chapter 2. The proposed system model and problem formulation are presented in detail in Chapter 4. Chapter 5 discusses the proposed algorithms: extended Kalman filter with Riemannian optimization (EKFRie) and unscented Kalman filter with Riemannian optimization (UKFRie). Finally, Chapters 6 and 7 present the results and conclusion.

CHAPTER 2

LITERATURE REVIEW

Moving target tracking problems appeared in the middle of the twentieth century to enable radars to track missiles [34], [35]. The work continued to evolve until 1960 when R. E. Kalman published his paper [36] which provided a very vital linear estimator, known as Kalman filter (KF), to estimate the linearly modeled hidden states. Since then, KF has become a widely used estimator in tracking. Extended Kalman filter (EKF) and unscented Kalman filter (UKF) [37], [38], [39], [40], [41] appeared as nonlinear versions of KF to deal with nonlinear systems. Although KF and its nonlinear versions are widely used in tracking problems because of their high efficiency and accuracy in linear and weakly nonlinear models, they are strongly affected by the behavior of the moving target and the nonlinearity degree of the model.

This literature review examines recent attempts to enhance indoor positioning performance. It focuses on three main approaches: modifications to the Kalman filter, the use of Riemannian optimization, and the fusion of Kalman filters with Riemannian manifolds.

2.1 Kalman Filters

In the literature, there are many techniques that have been used to enhance KF's performance in tracking. For instance, C. Liu *et al.* [42] tackled the problem of high maneuvering by modifying Kalman gain and scaling the state noise covariance by an estimated factor. On the other hand, in an attempt to address the deterioration in EKF performance caused by increased system nonlinearity and model uncertainty, J. Wang *et al.* [43] proposed a fifth-degree interpolatory Cubature Kalman Filter (ICKF) to get acceptable performance at the abrupt change of states. The fading factor was used to satisfy the orthogonality principle by adjusting Kalman gain adaptively, (i.e residuals at state k are uncorrelated with residuals at the next state $k + 1$). To deal with the situation of uncertain measurement noise, the author used inverse Wishart (IW) distribution and variational Bayesian-based Gaussian approximate filter.

Another method to enhance KF is by using many nodes to estimate the target position. In their work, X. Wang *et al.* introduced a novel approach to enhance the performance of EKF in maneuvering scenarios [44]. This involves gathering data from diverse sensors and leveraging the nested probabilistic-numerical linguistic information (NPNLI) method for optimization purposes. Using multiple sensors, a network is proposed by Z. Zhang *et al.* [45] to enhance KF's performance such that each sensor estimates states locally and enhances its estimates based on neighbors' readings. The final result depends on the consensus process of all nodes. Furthermore, [44] highlighted the advantages of incorporating additional sensors to enhance the accuracy and to smooth the tracked path.

In another direction, machine learning methods have been used in the literature to improve the tracking performance of different forms of Kalman filter [30], [46], [47], [48]. For instance, S. Mahfouz *et al.* [30] used radio fingerprints of received signal strength indicators as inputs to the ML algorithm to estimate the target position. Then a KF uses these estimates to update the estimated position based on acceleration and velocity.

Some works used an analytical approach to overcome the sensitivity and complexity of the inverse operation of the covariance matrix in the Kalman filter. For instance, W. Zhou and J. Hou [49] extracted singular value decomposition (SVD) of the state covariance and used it rather than the original state covariance to calculate the Kalman gain to overcome the matrix inversion sensitivity. To avoid the violation of the positive definite condition of the covariance matrix, the authors replaced the covariance matrix with its **QR** decomposition where **Q** is an orthonormal matrix and **R** is an upper triangular matrix.

2.2 Riemannian Manifold

While Euclidean geometry has provided a robust foundation for various applications, advancements in differential geometry, particularly in the Riemannian manifolds, have opened up new avenues for enhancing tracking algorithms. Some researchers have leveraged the Riemannian structure in tracking applications to enhance performance [50], [51], [52], [53], [54], [55]. Working with Riemannian geometry poses challenges due to the relative lack of mathematical tools compared to Euclidean space. However, B.

Bićanić *et al.* [50] extended the tracking problems from Euclidean space to Riemannian manifold space by utilizing probabilistic data association and leveraging tools such as exponential mapping, logarithmic mapping, and parallel transport. This approach showed improved performance compared to conventional methods.

In a similar vein, Yun *et al.* [51] proposed a tracking algorithm that individually utilizes multiple different views of cameras to estimate the position of a moving target. They defined a similarity metric based on the Riemannian manifold structure and applied a manifold-based maximum likelihood estimator to estimate the target location while considering geometric constraints. However, their approach assumed that target had upright movement, which limits the model generality. There are some works applying EKF/UKF on states belonging to the Lie group (some groups, such as $SO(3)$ group, belong to both Lie group and Riemannian manifold) [56], [57].

[58, Hertzberg] and [59, Hauberg] discussed applying UKF with Riemannian manifold, where UKF is preferred compared to EKF because it is a derivative-free filter and performs better generally in nonlinear situations [60]. Hertzberg focused on generalizing the needed mathematical operations in sensor fusion algorithms, such as summation and subtraction, to make them applicable in Riemannian space. So, the authors replaced $+/-$ with $[+]/[-]$ where the latter represents the summation and subtraction defined in Riemannian space. As a result, one can go through Riemannian space as in Euclidean space but with those extended operations. Rather than that, Hauberg generalized the unscented transformation to be effective in both Riemannian and Euclidean environments. Both works presented their definitions of the mean in the context of a manifold, which formed the basis of their research efforts.

[58, Hertzberg] and [59, Hauberg] deal with the covariance differently. Hertzberg calculated the covariance straightforwardly as in Euclidean space but by using its extended operations. On the other hand, Hauberg used eigendecomposition to decompose the covariance matrix into its eigenvalues and eigenvectors. Then parallel transport operation is used to transport these eigenvectors along the shortest curve (geodesic path) connecting the given and estimated means. Finally, the covariance matrix is constructed from the original eigenvalues and transported eigenvectors. One needs to note that while the mean should belong to the manifold, the covariance doesn't, but its eigenvectors should belong to the tangent space (which belongs to Euclidean space) of its related mean.

Hauberg presented the generalization of the unscented transformation, the core of UKF, for two different cases: a function that transforms sigma points from a manifold into a manifold and a function that transforms sigma points from a tangent space of a manifold into a tangent space. Then [59] constructed UKF which is applicable to the Riemannian manifold scenario. Hertzberg showed the consistency of his model in different cases. He also showed that the performance of his model degrades as the tracked orientation goes towards singularity. In this context, [61] also extended some concepts and operators from Euclidean space into Riemannian Manifold. The authors defined the statistics (first and second moments) of random vectors and weighted sets belonging to the Riemannian manifold. Then, they extended the sigma point from Euclidean space into the Riemannian manifold in a closed form. After that, [61] provided their form of Riemannian UKF.

2.3 Indoor Positioning

As our work introduces a target equipped with three transmitters, this review also includes a comparison with literature related to localization and tracking using multiple transmitters/receivers for indoor positioning. Although the acceptable accuracy depends on the application, decimeter-level accuracy is often sufficient for indoor tracking in many contexts [22].

A. Correa *et al.* [26] aimed to estimate the location of a moving target in an indoor setting. They employed EKF to merge signal strength-based readings from wireless sensor networks with other readings obtained through machine learning algorithms. Their work demonstrated improved performance when using four receivers compared to a single receiver, resulting in RMSE of $1.25m$ and $3.6m$, respectively.

Multiple receivers are used also for outdoor tracking. For instance, Elnaz Namaz *et al.* [29] focused on estimating the target location by merging two GPS receivers with map matching. However, this work limited its investigation to the intersection and straight road scenarios, which restricts the generalizability of their findings.

UWB and IMU were used in [62] to track pen-like instruments. UWB signal was used to localize and transfer IMU's information simultaneously which saves resources. The UWB and IMU-based system achieved 90% error within $7mm$ for a $5m^2$ area. However, as the work was evaluated in small areas, its applicability and validity for larger areas may be the subject of further investigation. B. V. Krishnaven *et al.* [63] combined readings from an IMU and UWB technology using EKF and UKF for indoor tracking. However, their algorithms exhibited position error rates within $2m$

for 75% and 85% of the time, respectively, which may not be suitable for many indoor applications. Also, two UWB receivers and an odometer system were adopted by S. Busanelli and G. Ferrari [25] to estimate the position and orientation of a moving target by applying EKF. Two receivers were used to determine the heading angle with RMSE of less than $0.8m$.

Focusing on acoustic-based indoor localization, Ruizhi Chen *et al.* [64] introduced least squares (LS) algorithm based on time difference of arrival (TDoA) for static localization and the EKF algorithm with IMU for dynamic tracking. However, their outcomes showed relatively low accuracy for many indoor applications, with 95% of the error falling within $1.6m$. In contrast, S. Xu *et al.* [65] proposed an acoustic-based indoor localization algorithm that incorporated prior knowledge of the floor plan. By employing a particle filter with dead reckoning, they achieved decimeter accuracy in a smartphone scenario, presenting a promising approach for enhancing localization precision.

The work conducted by Carman Ka Man Lee *et al.* [8] delved into the exploration of utilizing Bluetooth signal strength for indoor tracking in a warehouse setting. They proposed a Kalman-LULU filter in the signal processing stage of estimating the ranges. Subsequently, the trilateration algorithm was employed to estimate the positions of the target based on these ranges. However, their findings revealed an average RMSE of $2.13m$, indicating limited accuracy in their system. In a separate study, Jun Qi *et al.* [66] employed an ultrasonic-based time of flight (ToF) system for indoor tracking and proposed an algorithm for enhancing ToF estimation through envelope detection. Then, the LS algorithm was used to estimate the target position based on the

estimated ranges. Although they achieved a maximum location error of $10.2mm$, it is worth noting that their investigation was constrained to circular trajectory shapes with a maximum diameter below $1.5m$, raising uncertainties about the generalizability of their results to other trajectories with longer distances.

A study by M. Yasir *et al.* [23] proposed a method for tracking a moving target using an LED as a transmitter and 4 photodiodes (PD) with one accelerometer at the receiver. The mean position error was calculated and it was around $0.06m$. However, mean position error alone may not provide a comprehensive evaluation compared to metrics like mean square error (MSE) or RMSE. Some other works aimed to improve indoor localization performance from a communication perspective [67]. The authors introduced a new modulation technique to increase the data rate of the acoustic signal. They also proposed a normalization method to address communication issues such as the near-far effect and multipath during message detection at beacons which offered an improvement in positioning accuracy.

Table 2.1 presents a comparison of the mentioned indoor localization and tracking studies, providing additional details about the algorithms used, noise statistics, evaluation techniques, and any identified limitations. As the table shows, most of the works covered are based on EKF/UKF, and most of the shortcomings are related to the level of accuracy and generality of the work.

Based on the literature review conducted, we have identified two significant gaps in the existing research on utilizing Riemannian optimization with Kalman filters in localization and tracking. Firstly, there is a lack of exploration in leveraging the rotation matrix's $SO(3)$ manifold structure to enhance the performance. Secondly, the

impact of the receivers' isosceles triangle geometry on tracking performance has not been thoroughly discussed in the literature to the best of our knowledge. Therefore, this work aims to address these gaps by utilizing EKF and UKF algorithms with Riemannian optimization to advance indoor localization and tracking.

Table 2.1: Comparison between some indoor localization works featuring multiple receivers

Ref.	Algorithm	Noise Statistics	Evaluation	Shortcomings
[26]	1) RSS-based distance estimation using machine learning 2) Distance-based position estimation with weighted least square algorithm 3) EKF for distance-IMU position fusion	σ_{state}^2 : position = 5, speed = 0.5 $\sigma_{meas.}^2$: position = 1, and speed = 0.1 Experimental Configuration	Linear regression: RMSE = 1.2m and 0.9m for path = 44m and 74.4m Neural network: RMSE = 1m and 0.75m for path = 44m and 74.4m	Cost: 4 receivers and 6 beacons, assumed constant velocity
[23]	Multiple-Photodiode-based indoor positioning algorithm is used by utilizing the received light intensities and accelerometer readings.	$\sigma_{accelerometer}^2 = 5.7 \times 10^{-5}(m/s^2)^2$ Experimental Configuration	Mean error around 0.06m Error bounded by 0.1m	Didn't provide MSE/ RMSE for comprehensive evaluation, high cost, required LOS.
[62]	Kalman filter for sensor fusion of IMU and TDoA-based ranges estimation from UWB system	9DoF IMU and fully integrated single UWB chip (DW1000)	Median error around 2.9mm 90 percentile of error within 7mm	Tested for a small area (5m ²)
[25]	EKF for sensor fusion of odometer readings and TDoA-based ranges estimation from UWB system	$\sigma_{states}^2 = 1 \times 10^{-4}m$ and $3.06\mu rad$ for position and angle	RMSE for position and angles less than 0.8m and 0.5rad	RMSE is very oscillated with small-ranging errors, requiring synchronization
[63]	EKF & UKF for sensor fusion of IMU readings and ToA-based ranges estimation from UWB system	$\sigma_{meas.}^2 = 1 \times 10^{-4}$ $\sigma_{process}^2 = 1 \times 10^{-10}$	90 and 75 percentile of error for UKF and EKF within 2m	Examined over one path shape only, low accuracy regarding many indoor applications
[64]	EKF for sensor fusion of IMU readings and ToA-based ranges estimation from UWB system	None mentioned	95 percentile of error bellow 1.6m	Examined 2D only, low accuracy for many indoor applications

[65]	Particle filter with dead reckoning and platform restraint	$\sigma_{meas.}^2 = 0.2$, $\sigma_{position}^2 = 0.4$, and $\sigma_{angle}^2 = 3$	95 percentile of error within $0.6m$	Considered 2D only
[8]	Bluetooth-based indoor positioning using trilateration algorithm and RSS measurements	Receiver: raspberry Pi 3, model B Bluetooth: 4.1 classic, BLE	RMSE = $2.13m$	Low accuracy for many indoor applications
[66]	ToF estimation by extracting the envelope of the ultrasonic signals. Then, the position is estimated based on the least square method.	None mentioned	Maximum error of $10.24mm$	Considered one trajectory shape with a small distance
[67]	Particle filters utilize acoustic Doppler velocity measurements and beacon timestamps to refine the probability of each particle's state.	INMP411 high performance microphone Raspberry Pi 3 board	90 percentile of error within $0.49m$	Acoustic requires more hardware because of the low covering area.

CHAPTER 3

TECHNICAL BACKGROUND

This chapter introduces brief explanations of three topics:

1. Tracking and navigation,
2. Kalman filter and its nonlinear versions, and
3. Riemannian manifold and Riemannian optimization.

To fully grasp this work, a solid understanding of these topics is essential. The first section discusses various tracking techniques, providing a concise overview. This is followed by a comparative explanation of the Kalman filters (KF, EKF, and UKF) in the second section. The third section then elucidates key concepts related to manifold optimization, Riemannian manifold, smooth mapping, and certain aspects of Riemannian operators. Finally, the focus narrows to a specific group of Riemannian manifolds known as the special orthogonal group ($SO(n)$) due to its relevance to our work.

3.1 Target Tracking

Tracking refers to estimating the dynamic states of a moving object. These states, which encompass the target location and orientation, are typically represented by position coordinates, rotation angles, or alternative rotation representations such as rotation matrices. The following two subsections provide more about tracking the position and orientation of a moving target.

3.1.1 Position Tracking

Many techniques may be used to track the position of a moving target [68]. This brief presents two relevant ones:

1. **Position's tracking by IMU:** An IMU is a device that is attached to the target's body to provide information about the instantaneous acceleration and angular velocity of the target. Tracking the target position by IMU requires knowledge about the initial values of the position and velocity [69], [70]. Consequentially, the instantaneous velocity is extracted from the instantaneous acceleration by using the following equation [71] while v , a , and T denote linear velocity, acceleration, and time step, respectively:

$$v_{t+1} = a_t T + v_t \quad (3.1)$$

Once we know the instantaneous velocity, we can find the new position of the

target by using the following equation where p denotes the position:

$$p_{t+1} = p_t + v_t T + \frac{1}{2} a_t T^2 \quad (3.2)$$

Many researchers have used this method such as [72] and [73], which is known in the literature as the dead reckoning method [22].

2. Position's tracking by the network: In the second approach, the target is presumed to have a transmitter (or receiver) for sending (or receiving) signals. The network will include beacon (anchor) nodes equipped with receivers (or transmitters). Consequently, the tracking process starts by extracting range information between the target and beacons, using the received signal [74]. Various technologies can be utilized to extract range information, including Bluetooth, WiFi, ultrasound, and optics. There are multiple techniques available for range estimation based on transmitted signals, such as time of arrival (ToA), angle of arrival (AoA), and received signal strength (RSS). For this work, we assume that the estimated ranges are readily accessible, and therefore, we will not delve into these specific technologies and techniques. For more detailed information on these technologies and techniques, interested readers can refer to [22].

Knowing the beacons' positions allows us to estimate the target position once the ranges are determined. The following equation illustrates the relationship between these three quantities, where s_i , p and b_i represent the distance between the i^{th} beacon and target, the target position, and the position of the i^{th} beacon

respectively:

$$s_i = ||p - b_i|| \quad (3.3)$$

3.1.2 Orientation Tracking

The orientation of a target can be defined by specifying the angles between the target's body coordinate system (BCS), and the coordinate of the defined reference system, called the global coordinate system (GCS) as shown in Fig.3.1.

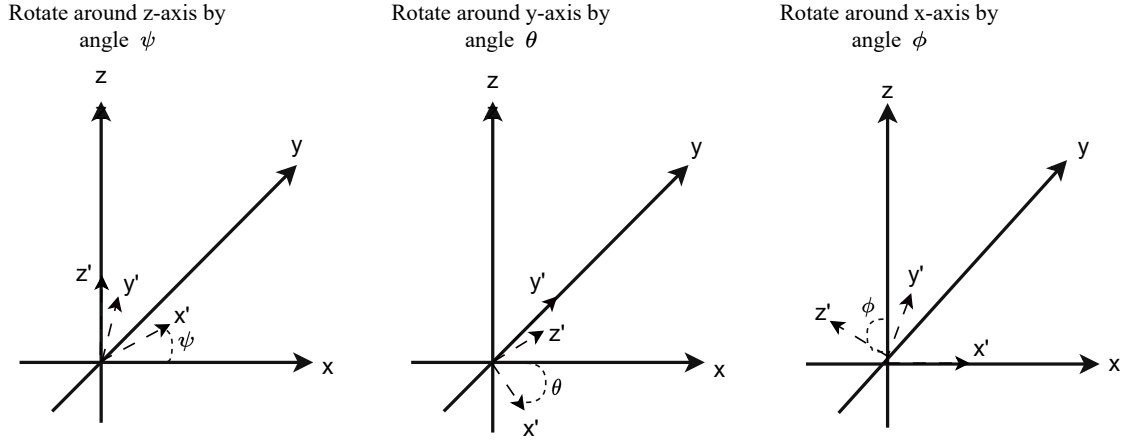


Figure 3.1: Defining angles of rotation between BCS and GCS where the solid axes, dashed axes, ψ , θ , and ϕ represent the GCS, BCS, angles of rotation around z -axis, y -axis, and x -axis, respectively.

The transformation from GCS into BCS can be done by rotating around z , y and x axes by the angles ψ , θ and ϕ , respectively. These angles are well known in the literature by Euler's angles and their order is important. One important quantity in the context of rotation is the *angular velocity* (ω) which defines the changing rate of these angles with time. To interested readers, please see [75] which includes the derivation of all equations in this subsection.

Tracking the new states of these Euler's angles for moving targets could be con-

ducted if their initial values and the angular velocity are known as following:

$$\begin{bmatrix} \phi_{t+1} \\ \theta_{t+1} \\ \psi_{t+1} \end{bmatrix} = \begin{bmatrix} \phi_t \\ \theta_t \\ \psi_t \end{bmatrix} + T \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (3.4)$$

Another representation of target orientation is by **rotation matrix** ($\mathbf{R}$) where $\mathbf{R}$ is a function of Euler's angles as follows:

$$\mathbf{R} = \mathbf{R}_1 \mathbf{R}_2 \mathbf{R}_3 \quad (3.5)$$

where

$$\mathbf{R}_1 = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}, \quad \text{and } \mathbf{R}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

One can track the new state of $\mathbf{R}$ by following equation:

$$\mathbf{R}_{t+1} \approx \mathbf{R}_t + T\dot{\mathbf{R}} \quad (3.6)$$

where $\dot{\mathbf{R}}$ refers to the rate of change of $\mathbf{R}$, and it is defined as follows:

$$\dot{\mathbf{R}} = \mathbf{R} \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (3.7)$$

Eq. (3.7) can be written in component form as follows where r_{ij} refers to the the element of $\mathbf{R}$ located in i row and j column:

$$\dot{r}_{11} = r_{12}\omega_z - r_{13}\omega_y \quad \dot{r}_{12} = r_{13}\omega_x - r_{11}\omega_z \quad \dot{r}_{13} = r_{11}\omega_y - r_{12}\omega_x \quad (3.8)$$

$$\dot{r}_{21} = r_{22}\omega_z - r_{23}\omega_y \quad \dot{r}_{22} = r_{23}\omega_x - r_{21}\omega_z \quad \dot{r}_{23} = r_{21}\omega_y - r_{22}\omega_x \quad (3.9)$$

$$\dot{r}_{31} = r_{32}\omega_z - r_{33}\omega_y \quad \dot{r}_{32} = r_{33}\omega_x - r_{31}\omega_z \quad \dot{r}_{33} = r_{31}\omega_y - r_{32}\omega_x \quad (3.10)$$

3.2 Kalman Filter and its Variations

Before delving into the details of the Kalman filter, let's take a brief look at the linear least mean square error estimator (l.l.m.s.e). The l.l.m.s.e is a less efficient version of the least mean square error estimator in terms of MSE. It operates by minimizing a mean square error function to estimate the values of hidden states. The word *linear* refers to the linearity assumption of the estimator model. In the case of a Gaussian distribution, the l.l.m.s.e estimator is equivalent to the least mean square error estimator, meaning that its performance approaches that of the optimal estimator.

One of the most powerful applications of the l.l.m.s.e is the Kalman filter, which is the focus of this thesis. Consequently, The Kalman filter is an optimal estimator in the Gaussian distribution case, but it could be suboptimal in other cases.

Another way to think about the Kalman filter is as a sensor fusion technique that combines readings from two different subsystems. One subsystem represents the relationship between the current and next states of the system model, assuming the Markovian property. The other subsystem represents the relationship between the current states of the system model and some observed quantities, called observations or measurements, assuming the existence of an independent measurement system. The Kalman filter requires that these observations be uncorrelated with each other.

The Kalman filter estimates the mean and covariance matrix of the system's states by combining readings from the two subsystems mentioned above using a gain matrix called the *Kalman gain*. The Kalman gain plays a vital role in the outputs of the Kalman filter, and it depends on the covariance of the two subsystems and the covariance of the system states. There are many forms of the Kalman filter, but this material focuses on one of them called *the time and measurement update form*. The time update estimates the next states based on the current states, while the measurement update filters the current states based on the observations.

One challenge of the Kalman filter is that it requires the system model to be linear. However, most real-world systems follow a nonlinear state space model. Two nonlinear variants of the Kalman filter have been developed to address this challenge: the extended Kalman filter (EKF) and the unscented Kalman filter (UKF).

The EKF linearizes the state space model around the most recent states using

a zeroth- and first-order Taylor series expansion. However, the partial derivative computation increases the computational complexity of the EKF.

The UKF, on the other hand, estimates the mean and covariance matrix of the nonlinear model by propagating a set of points called *sigma points*. These sigma points are initially generated to capture the distribution of the current states. They are then propagated through the nonlinear model to represent the distribution of the next states. Finally, statistics are used to compute the updated mean and covariance matrix based on the propagated points. For more information on Kalman filters, please see [41].

3.3 Riemannian Manifold

This section provides background information about the Riemannian manifold ($\mathcal{M}$) and Riemannian optimization. Because this thesis relies particularly on a group of Riemannian manifolds, called $\text{SO}(3)$ manifold, this section ends by focusing specifically on that manifold and defining its mathematical operators. The majority of the background material presented here is drawn from [1] and [32].

3.3.1 Motivation

Conventional optimization problems can be represented as follows:

$$\min_{x \in \mathcal{E}} f(x)$$

This means that we want to find a value of x that minimizes the cost function $f(x)$, where x belongs to Euclidean space ($\mathcal{E}$). Suppose that we know that the true value of x always lies on the circumference of a unit circle (i.e. $x \in \mathbb{R}^2 : ||x|| = 1$). In this case, there is no need to search the entire Euclidean space $\mathcal{E}$, since $\mathcal{E}$ contains both points that lie on the circumference of the circle and points that do not. The most efficient approach in this case is to define a new space that only contains points that lie on the circumference of the unit circle. This new space, which we will call the *circular space*, can be defined as a **manifold** and denoted by $\mathcal{S}^1$.

Defining this new space (manifold) necessitates developing novel mathematical operations that are analogous to those employed in Euclidean space. To elaborate, the addition operation is not defined in our circular space because if $a, b \in \mathcal{S}^1$, then there's no guarantee that $a + b \in \mathcal{S}^1$. So, for algorithms like gradient descent relying on addition, we either need a manifold addition operation or an equivalent gradient descent algorithm on the manifold.

3.3.2 Smooth Manifold

A manifold is a topological space that could be linearized (i.e. differentiable) at any belonging point x . Obviously, all the following structures in Fig. 3.2 are manifolds except the most left one since it includes edges. The manifold is called a smooth manifold ($\mathcal{M}$) if it is infinitely differentiable at each $x \in \mathcal{M}$. We are usually interested in smooth manifolds because they are easy to handle mathematically.

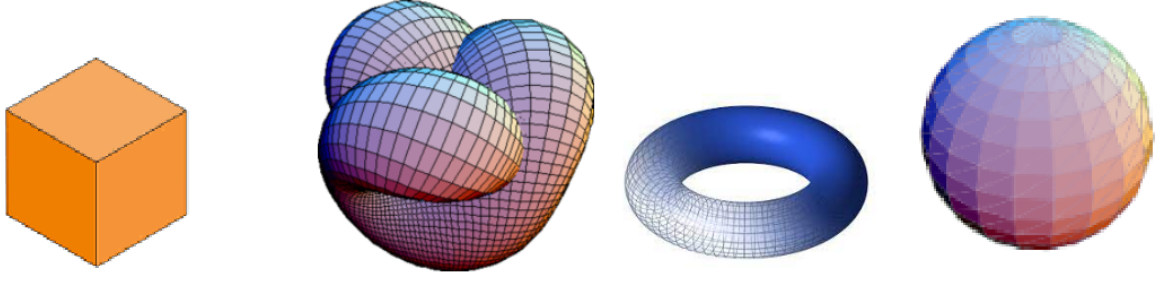


Figure 3.2: Different geometric shapes to demonstrate the manifold concept, from left: cube (not manifold), Boy's surface, torus, and sphere

3.3.3 Smooth Embedded Submanifolds of Euclidean Space

We are usually interested in embedded submanifolds since they can inherit the inner product definition from their embedding space. The following definition is a definition of the smooth embedded submanifolds of Euclidean space ($\mathcal{E}$) from a differential geometry perspective.

Definition 2.1: Let $\mathcal{E}$ be a linear space of dimension n and let $\mathcal{M}$ be a subset of $\mathcal{E}$.

Then, $\mathcal{M}$ is a smooth embedded submanifold of $\mathcal{E}$ of dimension d (where $d < n$) if:

- for a fixed integer $k \geq 1$ and for each $x \in \mathcal{M}$, there exists a neighborhood U of x in $\mathcal{E}$ and a smooth function $\hbar: U \rightarrow \mathbb{R}^k$ such that:

1. If y is in U , then $\hbar(y) = 0$ if and only if $y \in \mathcal{M}$; and
2. $\text{rank} D\hbar(x) = k$

where such function $\hbar$ is called a local defining function for $\mathcal{M}$ at x , and

$D\hbar(x)$ is the differential of $\hbar$ at x .

Fig. 3.3 clarifies this definition for a circular space with dimension 1 embedded in $\mathcal{E}$ with dimension 2 (i.e. the circumference of the circle represents the manifold). It is preferred usually to deal with $\mathcal{M}$ as an embedded submanifold in another manifold

called an embedding manifold, which is usually $\mathcal{E}$. Henceforth, we will typically imply smooth embedded submanifolds even though we may simply use the term ‘manifold’ for brevity.

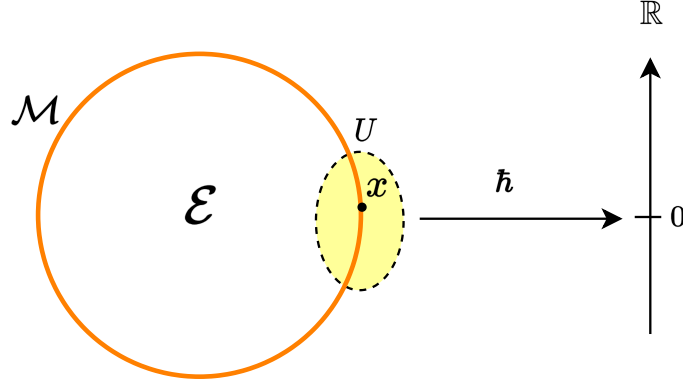


Figure 3.3: Manifold local defining functions: h smoothly maps an open set U (a disk) to $\mathbb{R}^k$ ($k = 1$); exactly the intersection of U with the set $\mathcal{M}$ (a curve) is mapped to 0 [1].

3.3.4 Tangent Vector and Tangent Space

As mentioned previously, smooth manifold $\mathcal{M}$ could be linearized at any $x \in \mathcal{M}$.

In differential geometry, this linearized space is defined as the tangent space of $\mathcal{M}$ at x , denoted by $\mathcal{T}_x\mathcal{M}$.

$\mathcal{T}_x\mathcal{M}$ encompasses all tangent vectors ($\vec{v}$) of $\mathcal{M}$ at x . After the general concept of the tangent space, we introduce its strict definition as follows:

Theorem 2.1: Let $\mathcal{M}$ be an embedded submanifold of $\mathcal{E}$ where $x \in \mathcal{M}$:

1. if $\mathcal{M}$ is an open submanifold, then $\mathcal{T}_x\mathcal{M} = \mathcal{E}$,
2. otherwise, $\mathcal{T}_x\mathcal{M} = \ker D\hbar(x)$ with $\hbar$ any local defining function at x , where $\ker D\hbar(x)$ means zeros of $D\hbar(x)$.

Fig. 3.4 illustrates the geometric interpretation of the tangent space, where $\mathcal{T}_x\mathcal{S}^{n-1}$ refers to tangent space of the sphere manifold ($\mathcal{S}^{n-1}$) at point $x \in \mathcal{S}^{n-1}$, $\vec{v}$ refers to the tangent vector included in $\mathcal{T}_x\mathcal{S}^{n-1}$ and $\mathcal{R}_x(\vec{v})$ denotes the retraction operator which will be discussed later. It is important to note that since $\mathcal{M}$ is an embedded submanifold of $\mathcal{E}$, $\mathcal{T}_x\mathcal{M}$ is a linear subspace of $\mathcal{E}$.

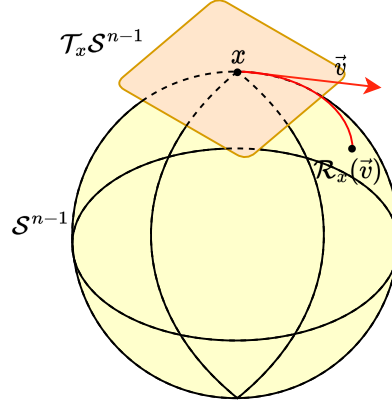


Figure 3.4: Geometric representation of sphere manifold ($\mathcal{S}^{n-1}$) with its tangent space and retraction

3.3.5 Smooth Map on Embedded Smooth Manifold

Given our focus on manifold optimization over $x \in \mathcal{M}$, it is natural to define a smooth map on $\mathcal{M}$. The concept of a smooth map on $\mathcal{M}$ draws heavily upon the classical notion of smooth maps between linear spaces. Due to the relative simplicity of defining smooth maps on $\mathcal{M}$, this section directly presents the strict definition.

Definition 2.2: Let $\mathcal{M}$ and $\mathcal{M}'$ be embedded submanifolds of $\mathcal{E}$ and $\mathcal{E}'$ respectively. A map $F : \mathcal{M} \rightarrow \mathcal{M}'$ is smooth at $x \in \mathcal{M}$ if there exists a function $\bar{F} : U \rightarrow \mathcal{E}'$ which is smooth (in the usual sense) on a neighborhood U of x in $\mathcal{E}$ and such that $F(y) = \bar{F}(y)$ for all $y \in (\mathcal{M} \cap U)$. That is: F is the restriction of $\bar{F}$ to $\mathcal{M}$: $F = \bar{F}|_{\mathcal{M}}$.

3.3.6 Riemannian Manifold

Prior to delving into the concept of a Riemannian manifold, let's first define the Riemannian metric. Consider an embedded submanifold $\mathcal{M}$ of $\mathcal{E}$. We say that $\mathcal{M}$ is equipped with Riemannian metric if the inner product $\langle \cdot, \cdot \rangle_x$ is defined on $\mathcal{T}_x\mathcal{M}$ for all x and for all tangent vectors belong to $\mathcal{T}_x\mathcal{M}$ where this inner product changes smoothly with x .

A manifold $\mathcal{M}$ is called a Riemannian manifold if it is smooth and equipped with a Riemannian metric. The significance of Riemannian manifolds lies in their ability to support the definition of the gradient operation (∇) on their tangent space ($\mathcal{T}\mathcal{M}$). To clarify that, recall the relationship between the gradient and the inner product, which can be expressed as follows:

$$Df(x)[\vec{v}] = \langle \vec{v}, \nabla f(x) \rangle \quad (3.11)$$

Eq. 3.11 implies that the directional derivative of a function $f(x)$ in the direction of $\vec{v}$ is equal to the inner product of the gradient of $f(x)$ and $\vec{v}$. This means that one way to find the gradient is to obtain an expression of $Df(x)[\vec{v}]$ and then manipulate it to be in the form of $\langle \vec{v}, \cdot \rangle_x$. The term that has been taken as the inner product with $\vec{v}$ represents the gradient. Fortunately, when we assume that $\mathcal{M}$ is a smooth manifold embedded in Euclidean space $\mathcal{E}$, its tangent space $\mathcal{T}\mathcal{M}$ automatically becomes a linear subset of $\mathcal{E}$ as is evident from the definition of an embedded submanifold. This implies that the inner product is defined on $\mathcal{T}\mathcal{M}$.

3.3.7 Orthogonal Projection

Several crucial tools including smooth vector fields, orthogonal projection, retractions, vector transport, exponential map, facilitate optimization over Riemannian manifolds. This brief overview focuses on orthogonal projection, retraction, and vector transport, as they are essential for comprehending this work.

Orthogonal projection ($Proj_x(\vec{h})$) is an operation that projects any vector $\vec{h}$ belonging to the embedding space of the manifold ($\mathcal{M}$) orthogonally onto the tangent space of that manifold at a point x . Let $\vec{h} \in \mathcal{E}$, then $Proj_x(\vec{h}) : \mathcal{E} \rightarrow \mathcal{T}_x\mathcal{M}$.

3.3.8 Retraction

Retraction ($\mathcal{R}_x(\vec{v})$) is a mapping from the tangent space $\mathcal{T}_x\mathcal{M}$ of the manifold $\mathcal{M}$ at a point x to the manifold itself such that: $\mathcal{R}_x(\vec{v}) : \mathcal{T}_x\mathcal{M} \rightarrow \mathcal{M}$. Retraction is an essential tool for optimization on Riemannian manifolds, as it allows us to update points during optimization iterations while ensuring that they remain on the manifold. Fig. 3.5 illustrates the geometric concept of retraction.

From a differential geometry standpoint, employing retraction in optimization can be visualized as traversing a curve $c(t)$, known as a geodesic, on the manifold, connecting the current point to the updated point. Curve $c(t)$ is depicted in Fig. 3.5 as the curve connecting x and $\mathcal{R}_x(\vec{v})$. The following formal definition of the retraction operator is rooted in this concept.

Definition 2.3: A retraction on $\mathcal{M}$ is a smooth map $\mathcal{R} : \mathcal{T}\mathcal{M} \rightarrow \mathcal{M}$ with the following properties. For each $x \in \mathcal{M}$, let $\mathcal{R}_x : \mathcal{T}_x\mathcal{M} \rightarrow \mathcal{M}$ be the restriction of $\mathcal{R}$ at x , so

that $\mathcal{R}_x(\vec{v}) = \mathcal{R}(x, \vec{v})$. Then, $\mathcal{R}_x(0) = x$ and $D\mathcal{R}_x(0)$ is the identity map such that $D\mathcal{R}_x(0)[\vec{v}] = \vec{v}$.

So, retraction is equivalent to each curve $c(t)$ (i.e. $c(t) = \mathcal{R}_x(tv)$), satisfies that: $c(0) = x$ and $c'(0) = \vec{v}$.

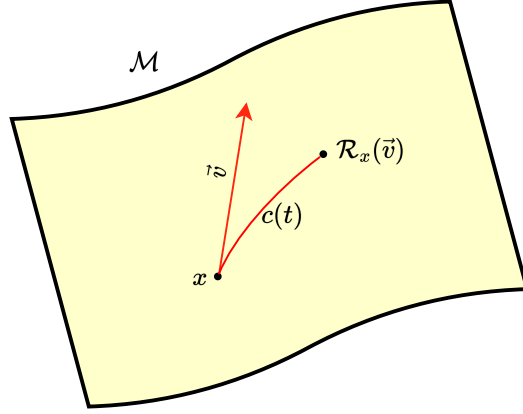


Figure 3.5: Demonstration of retraction operation to allow optimization over smooth manifold [1]

3.3.9 Vector Transport

Vector transport $(\mathcal{T}_{y \leftarrow x}(\vec{v}))$ is an operation that transfers a tangent vector $\vec{v}$ from the tangent space of point x on manifold $\mathcal{M}$ to the tangent space of point y on the same manifold while preserving the geometric properties of the vector. Let $x, y \in \mathcal{M}$ and $\vec{v} \in \mathcal{T}_x\mathcal{M}$, then $\mathcal{T}_{y \leftarrow x}(\vec{v}): \mathcal{T}_x\mathcal{M} \rightarrow \mathcal{T}_y\mathcal{M}$.

3.3.10 Special Orthogonal Group Manifold

This work assumes that the reader is familiar now with the following fundamental concepts in Riemannian geometry: smooth manifolds, Riemannian manifolds, defining functions, tangent spaces, tangent vectors, orthogonal projection, retraction, and

vector transport. This part of the work presents the specific expressions of these operators for the special orthogonal group manifold ($\text{SO}(3)$), which is the main focus of this thesis.

This work maintains the orientation of the moving target by continuously tracking its rotation matrix. The rotation matrix is also employed in positioning to transform the acceleration from BCS to GCS. Fortunately, the rotation matrix is obeyed inherently to the $\text{SO}(3)$ manifold. Leveraging this property, this work incorporates Riemannian optimization into its proposed tracking algorithm. For a more in-depth exploration of the Riemannian geometry of $\text{SO}(3)$, please refer to the work presented in [76].

Definition 2.4: $\text{SO}(3)$ manifold is a set of square matrices with size (3×3) embedded in $\mathbb{R}^{3 \times 3}$ (where $\mathbb{R}^{3 \times 3}$ is the embedding space endowed with the inner product) have orthonormal columns and their determinant equal to $(+1)$.

This definition can be stated equivalently as:

$$\text{SO}(3) = \{X \in \mathbb{R}^{3 \times 3} : X^\top X = X X^\top = I_3 \text{ and } \det(X) = +1\}$$

Table 3.1 summarizes the formulas of the aforementioned geometrical tools for $\text{SO}(3)$. It provides the formulas for the $\text{SO}(3)$ defining function, tangent space, orthogonal projector, vector transport, and retraction operator.

Table 3.1: Riemannian geometry for $SO(3)$ manifold

Name	Formula
Local defining function ($\hbar$)	$\hbar : \mathbb{R}^{3 \times 3} \rightarrow \text{sym}(3) : X \rightarrow \hbar(X) = X^\top X - I_3$, where $\det(X) = +1$ and $\text{sym}(3)$ is symmetric matrices of size 3 in the linear space.
Tangent space ($\mathcal{T}_X SO(3)$)	$\mathcal{T}_X SO(3) = \{X\Omega \in \mathbb{R}^{3 \times 3} : X \in SO(3) \ \& \ \Omega \in \text{Skew}(3) \text{ (that means: } \Omega \in \mathbb{R}^{3 \times 3} : \Omega^\top = -\Omega)\}$
Embedding space	$\mathbb{R}^{3 \times 3}$
Orthogonal Projection ($\text{Proj}_X(\vec{h})$)	$\text{Proj}_X(\vec{h}) = X\Omega$ where $\vec{h} \in \mathbb{R}^{3 \times 3}$ is a vector in the embedding space, and $\Omega = \text{Skew}(X^\top \vec{h})$ such that $\text{Skew}(A) = \frac{A - A^\top}{2}$
Vector Transport ($\mathcal{T}_{Y \leftarrow X}(\vec{v})$)	$Y\Omega$, such that $\vec{v} = X\Omega$
Retraction ($\mathcal{R}_X(\vec{v})$)	$qfactor(X + \vec{v})$, where $qfactor$ extracts the Q -factor of the QR decomposition with positive elements on the diagonal of R .

3.4 The Interplay of the Aforementioned Tools and Tracking

This work employs Riemannian optimization techniques to track the rotation matrix and its associated covariance matrix. The rotation matrix resides within the $SO(3)$ manifold, while the eigenvectors of the covariance matrix belong to the tangent space of $SO(3)$ [59]. Due to the presence of noise, it is assumed that the estimated rotation matrix will lie on the tangent space of $SO(3)$ rather than on $SO(3)$ itself. Consequently, this work proposes utilizing the retraction operator to project the estimated rotation matrix from the tangent space onto $SO(3)$. Similarly, vector transport is employed to transfer the eigenvectors of the corresponding covariance matrix from the tangent space of the current rotation matrix to the updated one. The Riemannian tools discussed in this chapter are leveraged to track the position and orientation of a

moving target based on the system model and problem formulation outlined in the subsequent chapter.

CHAPTER 4

SYSTEM MODEL AND PROBLEM FORMULATION

In this chapter, we describe the considered system model and the mathematical formulation of the tracking problem.

4.1 System Model

This work considers tracking the position and orientation of a moving target equipped with IMU in an environment equipped with a measurement system (MS). The measurement system in this context means a system that uses the transmitted/ received signal for positioning. The positioning quantities are usually represented in two different but related coordinate systems: the body coordinate system (BCS), which is attached to the target, and the global coordinate system (GCS), which is an external reference frame. To avoid confusion, the quantities in the BCS are denoted with a superscript b in this script, while symbols without superscripts are used for those in

the GCS.

The considered MS consists of M anchors (beacons) and three transmitters attached to the target. The position of the i^{th} transmitter in the BCS and the j^{th} anchor in GCS are known and denoted as $\mathbf{p}_i^b$ and $\mathbf{b}_j$, respectively, where both $\in \mathbb{R}^3$. The three transmitters are arranged to align the vertices of an isosceles triangle with a base of length d and an altitude of length a , as depicted in Fig. 4.1. To form an isosceles triangle, the transmitters' positions are formulated in terms of a and d as $\mathbf{p}_1^b = (0, 0, \frac{2a}{3})$, $\mathbf{p}_2^b = (\frac{d}{2}, 0, \frac{-a}{3})$, and $\mathbf{p}_3^b = (\frac{-d}{2}, 0, \frac{-a}{3})$. Furthermore, the s_{ij} denotes to the noisy range between i^{th} transmitter and j^{th} anchor such that $s = s_{true} + s_n$; where s_n is an additive Gaussian noise with zero mean and σ_s^2 variance.

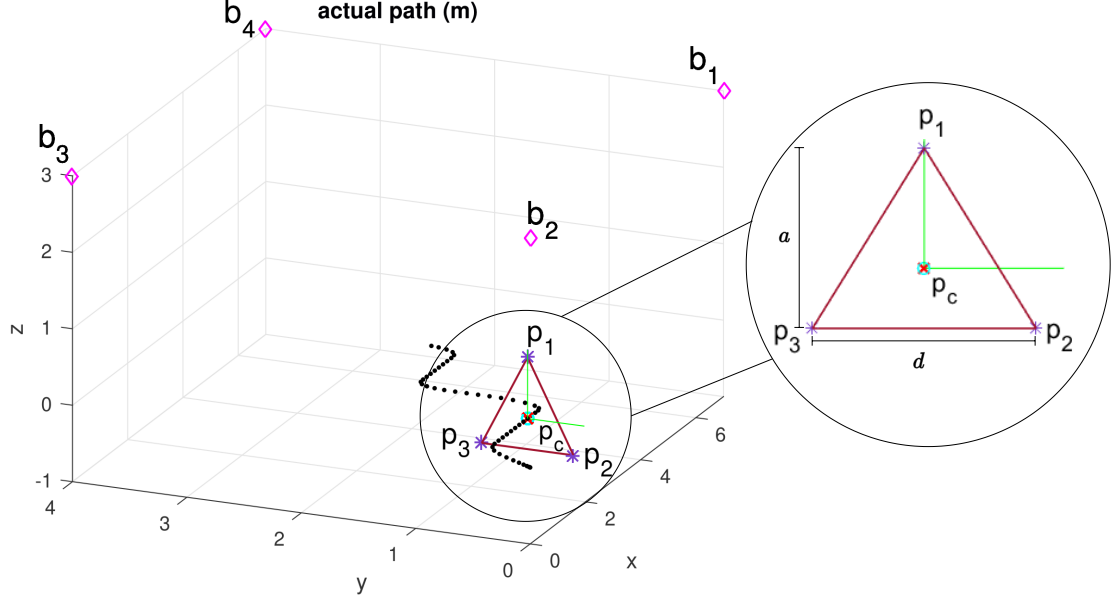


Figure 4.1: A system diagram to show the moving target with its components, where the dots line, brown triangle, purple asterisks, green axes, red cross, cyan circle, and magenta diamond refer to the moving path, isosceles triangle, transmitters, BCS, target centroid, IMU, and beacons, respectively.

The IMU gives two quantities: the target centroid's acceleration in BCS ($\mathbf{a}_c^b \in \mathbb{R}^3$) and the angular velocity ($\omega \in \mathbb{R}^3$). Both $\mathbf{a}_c^b$ and ω form a column vector $\mathbf{u} \in \mathbb{R}^6$ such

that $\mathbf{u} = [\omega^\top \quad \mathbf{a}_c^{b\top}]^\top$, where $\mathbf{u}$ represents the input of our system.

The input $\mathbf{u}$ is a noisy quantity such that $\mathbf{u} = \mathbf{u}_{true} + \mathbf{u}_n$ where $\mathbf{u}_n$ is a noise process described as additive white Gaussian noise (AWGN) such that $\mathbf{u}_n \sim \mathcal{N}(0, \mathbf{Q})$. $\mathbf{Q} \in \mathbb{R}^{6 \times 6}$ is a diagonal matrix that represents the covariance matrix of the input. The first three diagonal elements represent the variance of the angular velocity σ_ω^2 and the other three diagonal elements represent the variance of the acceleration σ_a^2 as appeared in Eq. (4.1).

$$\mathbf{Q} = \begin{bmatrix} \sigma_\omega^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\omega^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_\omega^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_a^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_a^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_a^2 \end{bmatrix} \quad (4.1)$$

The position of the transmitters $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ in GCS represents the output $\mathbf{y} \in \mathbb{R}^9$ of our system such that $\mathbf{y} = [\mathbf{p}_1^\top \quad \mathbf{p}_2^\top \quad \mathbf{p}_3^\top]^\top$. This work utilizes two independent subsystems to estimate $\mathbf{y}$: the IMU system and MS system. To avoid confusion $\hat{\mathbf{y}}_{imu}$ and $\mathbf{y}_{ms}$ refer to the estimated $\mathbf{y}$ based on the IMU and MS, respectively. The $\mathbf{y}_{ms}$ is a noisy quantity such that $\mathbf{y}_{ms} = \mathbf{y}_{ms,true} + \mathbf{y}_{ms,n}$ where $\mathbf{y}_{ms,n} \sim \mathcal{N}(0, \mathbf{Z})$, and $\mathbf{Z} = \sigma_s^2 I_9 \in \mathbb{R}^{9 \times 9}$ is the covariance matrix.

The ultimate purpose of this work is to provide an algorithm that estimates the position and orientation of the moving target obeying the mentioned circumstances by assuming a prior knowledge of the initial values of its position, velocity, and orientation in the global coordinate system.

The IMU, the origin of the BCS, the centroid of the target, and the centroid of the isosceles triangle coincide with each other. As a result, the position, velocity, and acceleration of the target are measured and estimated at its centroid, and they are denoted as $\mathbf{p}_c$, $\mathbf{v}_c$, and $\mathbf{a}_c$ respectively, all belonging to $\mathbb{R}^3$. Since the target centroid coincides with the BCS's origin, $\mathbf{p}_c^b = (0, 0, 0)$.

Although we have access to some of the aforementioned positioning quantities in BCS solely, practically we are interested in them in the GCS. So, the rotation matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$, belonging to $\text{SO}(3)$ manifold, is used to transform the quantities from the BCS into the GCS and vice versa, as described by details in [75]. The rotation matrix could be expressed as $\mathbf{R} = [R_1 \ R_2 \ R_3]$, where $R_i \in \mathbb{R}^3$ refers to the i^{th} column of $\mathbf{R}$. Moreover, the element of $\mathbf{R}$ at i^{th} row and j^{th} column is denoted as r_{ij} .

4.2 Problem Formulation

This work adopts a state space model to formulate tracking the position and orientation of a moving target. This section is divided into two parts. The first part introduces the state space model in continuous time, while the second part focuses on discretizing the state space model.

1. **State Space Model:** The state space model consists of two equations: a state equation and a measurement equation. The state equation of our model is described as follows.

Let $\Theta \in \mathbb{R}^9$ be a column vector with a column-major order of $\mathbf{R}$ (i.e. $\Theta =$

$[R_1^\top \ R_2^\top \ R_3^\top]^\top$). Also, let $\Psi \in \mathbb{R}^6$ be a column vector that contains the target position and velocity in GCS (i.e. $\Psi = [\mathbf{p}_c^\top \ \mathbf{v}_c^\top]^\top$). Then, we define $\mathbf{X}$ a state vector of our system with dimension L such that $\mathbf{X} = [\Theta^\top \ \Psi^\top]^\top$ where $\mathbf{X} \in \mathbb{R}^{15}$. From [75, Sec 3.6] and from the kinematic equations [71], [77], it can be shown that the state equation of $\mathbf{X}$ can be represented as follows:

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + f(\mathbf{X})\mathbf{u} \quad (4.2)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{9 \times 12} & \mathbf{0}_{9 \times 3} \\ \mathbf{0}_{3 \times 12} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 12} & \mathbf{0}_{3 \times 3} \end{bmatrix}_{15 \times 15}, \quad f(\mathbf{X}) = \begin{bmatrix} \mathbf{\Pi}_{9 \times 3} & \mathbf{0}_{9 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{R}_{3 \times 3} \end{bmatrix}_{15 \times 6} \quad (4.3)$$

and where the matrix $\mathbf{\Pi}$ is defined as:

$$\mathbf{\Pi} = \begin{bmatrix} 0 & 0 & 0 & r_{13} & r_{23} & r_{33} & -r_{12} & -r_{22} & -r_{32} \\ -r_{13} & -r_{23} & -r_{33} & 0 & 0 & 0 & r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} & -r_{11} & -r_{21} & -r_{31} & 0 & 0 & 0 \end{bmatrix}^\top \quad (4.4)$$

Now we discuss the measurement equation of our model. Since the $\mathbf{R}$ and $\mathbf{p}_c$ are known from the state equation, and since the transmitters' positions in BCS are fixed and also known, the output of the system could be estimated based on

the IMU readings as:

$$\hat{\mathbf{y}}_{imu} = h(\mathbf{X}) = \begin{bmatrix} \mathbf{R}\mathbf{p}_1^b + \mathbf{p}_c \\ \mathbf{R}\mathbf{p}_2^b + \mathbf{p}_c \\ \mathbf{R}\mathbf{p}_3^b + \mathbf{p}_c \end{bmatrix} \quad (4.5)$$

Eq. (4.5) can be decomposed into two components: a rotational component and a translational component. The rotational component, represented by the matrix multiplication $\mathbf{R}\mathbf{p}_i^b$ transforms the transmitter position from the BCS to the GCS. The translational component, represented by the vector $\mathbf{p}_c$ accounts for the difference in origin between the BCS and the GCS.

In addition to estimating the output based on the IMU, KF needs to measure the output by utilizing the measurement system (MS). Many algorithms could be utilized by the MS to estimate the transmitters' positions based on the given ranges. Two of these algorithms are the Gauss-Newton algorithm and the isosceles triangle manifold algorithm. For more detailed information on these algorithms, please refer to [78], [79, Sec 10.2].

2. Discretization of State Space Model

In order to derive the discrete-time model, certain assumptions are made about the discretized signals between the sampling instants. Here, we assume that the inputs remain constant, which is commonly referred to as the zero-order hold (ZOH). Since each of Θ and Ψ has a different characteristic, they are

discretized separately. Then, these discretized equations are merged to form the overall discretized model.

Since $\dot{\Theta}_k \approx \frac{\Theta_k - \Theta_{k-1}}{T} \implies \Theta_k \approx \Theta_{k-1} + T\dot{\Theta}_k$. From (4.2), we can conclude that:

$$\dot{\Theta} = \mathbf{\Pi}\omega \quad (4.6)$$

So, the discretized state space equation of Θ will be as:

$$\hat{\Theta}_k \approx \Theta_{k-1} + T\mathbf{\Pi}\omega \quad (4.7)$$

From (4.2) also, we can conclude that:

$$\dot{\Psi} = \Upsilon.\Psi + \mathbf{B}.\mathbf{a}_c^b \quad (4.8)$$

where

$$\Upsilon = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{R}_{3 \times 3} \end{bmatrix} \quad (4.9)$$

Since the state equation of Ψ is linear, it is convenient to use linear ZOH discretization equations. We need to find two matrices $\Upsilon_D, \mathbf{B}_D$ describing the discrete-time system [80, Sec 4.2]. These are given in terms of the continuous-

time system matrices $\Upsilon, \mathbf{B}$ and sampling time T by:

$$\Upsilon_D = e^{\Upsilon T}, \quad \mathbf{B}_D = \int_0^T e^{\Upsilon t} \mathbf{B} dt \quad (4.10)$$

Applying these formulae we get:

$$\Upsilon_D = \begin{bmatrix} \mathbf{I}_3 & T\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}, \quad \mathbf{B}_D = \begin{bmatrix} \frac{T^2}{2} \mathbf{R} \\ T\mathbf{R} \end{bmatrix}$$

$$\hat{\Psi}_{k+1} \approx \Upsilon_D \Psi_k + \mathbf{B}_D \mathbf{a}_c^b \quad (4.11)$$

Note that the discretization errors of Eq. (4.7) and Eq. (4.11) are proportion to T and T^2 respectively.

To define the discretized overall state space model, we combine the results of discretizing Θ and Ψ (Eq. (4.7) and Eq. (4.11)) as follows:

$$\hat{\mathbf{X}}_k = \mathbf{F} \mathbf{X}_{k-1} + g(\mathbf{X}_{k-1}) \mathbf{u}_{k-1} \quad (4.12)$$

$$\text{where } \mathbf{F} = \begin{bmatrix} \mathbf{I}_{9 \times 9} & \mathbf{0}_{9 \times 3} & \mathbf{0}_{9 \times 3} \\ \mathbf{0}_{3 \times 9} & \mathbf{I}_{3 \times 3} & T\mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 9} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (4.13)$$

and

$$g(\mathbf{X}_{k-1}) = \begin{bmatrix} T\mathbf{\Pi}_{9 \times 3} & \mathbf{0}_{9 \times 3} \\ \mathbf{0}_{3 \times 3} & \frac{T^2}{2}\mathbf{R} \\ \mathbf{0}_{3 \times 3} & T\mathbf{R} \end{bmatrix} \quad (4.14)$$

In summary, this thesis introduces algorithms for tracking the position and orientation of a moving target. The target has two subsystems: IMU and a measurement system comprising three transmitters arranged in an isosceles triangle configuration. The IMU provides both the angular velocity and the acceleration of the target. Using the angular velocity, the target's orientation is estimated for the next time step, represented as a rotation matrix within the $\text{SO}(3)$ manifold, based on the current orientation. Employing the laws of motion, the target's position is estimated for the next time step using the acceleration, the rotation matrix, and the current position.

Due to the accumulation of errors in IMU-based estimation, three transmitters are employed to mitigate the increasing drift over time. This thesis assumes that the noisy range measurements between the transmitters and beacons are available. Subsequently, the Gauss-Newton method and the isosceles triangle manifold method are utilized to estimate the target position based on the provided range measurements.

The readings from these two subsystems are seamlessly integrated using EKF and UKF, leading to improved tracking performance. Furthermore, Riemannian optimization is applied to further enhance the accuracy and efficiency of the tracking system by exploiting the $\text{SO}(3)$ manifold. The subsequent chapter delves into the two algorithms

introduced in this thesis. These algorithms represent advanced versions of EKF and UKF that leverage Riemannian tools and operations.

CHAPTER 5

PROPOSED ALGORITHMS

This chapter outlines the proposed algorithms for tracking moving targets. These proposed algorithms are updated versions of the EKF and UKF, which leverage Riemannian optimization operators.

The conventional EKF and UKF algorithms consist of two parts: time update and measurement update. In the time update, the algorithms predict the new states and associated covariance matrix based on the current state and readings from the first subsystem (IMU in our case). In the measurement update, the EKF and UKF algorithms update the outputs of the previous part (time update) based on observations from the second subsystem (the measurement system).

It's important to note that the two subsystems should be independent of each other to achieve satisfactory results. A *Kalman gain* factor controls the influence of the two subsystems. As the Kalman gain decreases, the effect of the first subsystem increases and vice versa. Conversely, the effect of the second subsystem increases as the Kalman gain increases.

Both EKF and UKF are approximation methods used to handle nonlinear models.

EKF is based on Taylor series expansions and on calculating Jacobian matrices to linearize the nonlinear model around the estimated states. In contrast, UKF is based on creating some samples (called sigma points) to capture the first and second moments of the probability density function (pdf). Then, these sigma points are passed through the nonlinear model to estimate the first and second moments of the next state's pdf. For more detailed information about EKF and UKF, please refer to [41].

The rotation matrix belongs to the $SO(3)$ manifold, and the eigenvectors of its covariance matrix are assumed to be on the corresponding tangent space [59]. If the effector noise is high, this assumption will not hold true. In the estimation process and due to noise, the estimated rotation matrix is more likely to be on the tangent space of the $SO(3)$ manifold.

To address this, we use the retraction operator to project the rotation matrix from the tangent space into the $SO(3)$ manifold. Hauberg *et al.* [59] proposed a method to update the corresponding covariance matrix as follows. First, the corresponding covariance matrix is decomposed into its eigenvectors and eigenvalues. Then, the parallel transport shifts the eigenvectors from the tangent space of the original rotation matrix into the tangent space of the updated one. After that, the transported eigenvectors are recomposed with the original eigenvalues to form the new covariance matrix. However, as parallel transport is computationally expensive [1], [32], we replace it with the vector transport operation. These operators (retraction and vector transport) are used after each part of the EKF and UKF algorithms.

This study faces one of the challenges of the UKF, which is losing the positive definiteness property of the covariance matrix. That happens in this work, particularly

after applying vector transport. As a result of that, the Cholesky factorization fails. This challenge of the UKF is known in the literature as stated in [41, Sec. 30.8], [81]. To overcome this issue, this thesis utilizes a Matlab function developed by John D’Errico [82] to replace the transported covariance matrix with its nearest positive definite matrix. The proposed pseudo codes of this thesis are presented and discussed in the following section.

As outlined in table 3.1, the retraction operator for the rotation matrix can be expressed as a Q-factor of the sum of the rotation matrix and the related tangent vector. Furthermore, Section 4.2 introduces the time update equation for Θ , given by Eq. (4.7) as:

$$\hat{\Theta}_k \approx \Theta_{k-1} + T\Pi\omega$$

We aim to demonstrate that the right-hand side of Eq. (4.7) can be restructured to match the structure of the *qfactor(.)* function argument, namely, a matrix belonging to the SO(3) manifold plus a tangent vector of the SO(3) manifold. This allows us to directly apply the retraction operator by simply putting the restructured right-hand side of Eq. (4.7) as an argument of the *qfactor(.)* function.

To show that, let us convert Eq. (4.7) from its vector form into matrix form as follows:

$$\hat{\mathbf{R}}_k \approx \mathbf{R}_{k-1} + T\mathbf{R}_{k-1}\boldsymbol{\Omega} \tag{5.1}$$

where

$$\mathbf{\Omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (5.2)$$

Eq. (4.7) and Eq. (5.1) are equivalent to each other. The first term of Eq. (5.1) is clearly on the SO(3) manifold. The $\mathbf{\Omega}$, in the second term, is a skew symmetric matrix (meaning $\mathbf{\Omega}^\top = -\mathbf{\Omega}$), and its scaling is also skew symmetric. Since the tangent space of SO(3) is defined in Table 3.1 as the product of a matrix on the SO(3) manifold and a skew symmetric matrix, the second term of Equation (5.1) is in the tangent space of the SO(3) manifold. Therefore, it has been shown that the right-hand side of Equation (4.7) can be reformulated to match the structure of the argument of the *qfactor(.)* function.

5.1 Pseudo Codes

This section provides and discusses the proposed algorithms, EKF with Riemannian (EKFRie) and UKF with Riemannian (UKFRie). The overall mechanism of the algorithms is illustrated in Fig. 5.1 where GN and Iso refer to the Gauss-Newton algorithm and isosceles triangle manifold algorithm, respectively. The proposed algorithms are presented in Algorithms 1 and 2, respectively, where the red lines represent the Riemannian part of the algorithm.

The EKFRie algorithm comprises two primary loop: the time update stage and the measurement update stage. The time update stage, represented by the outer loop,

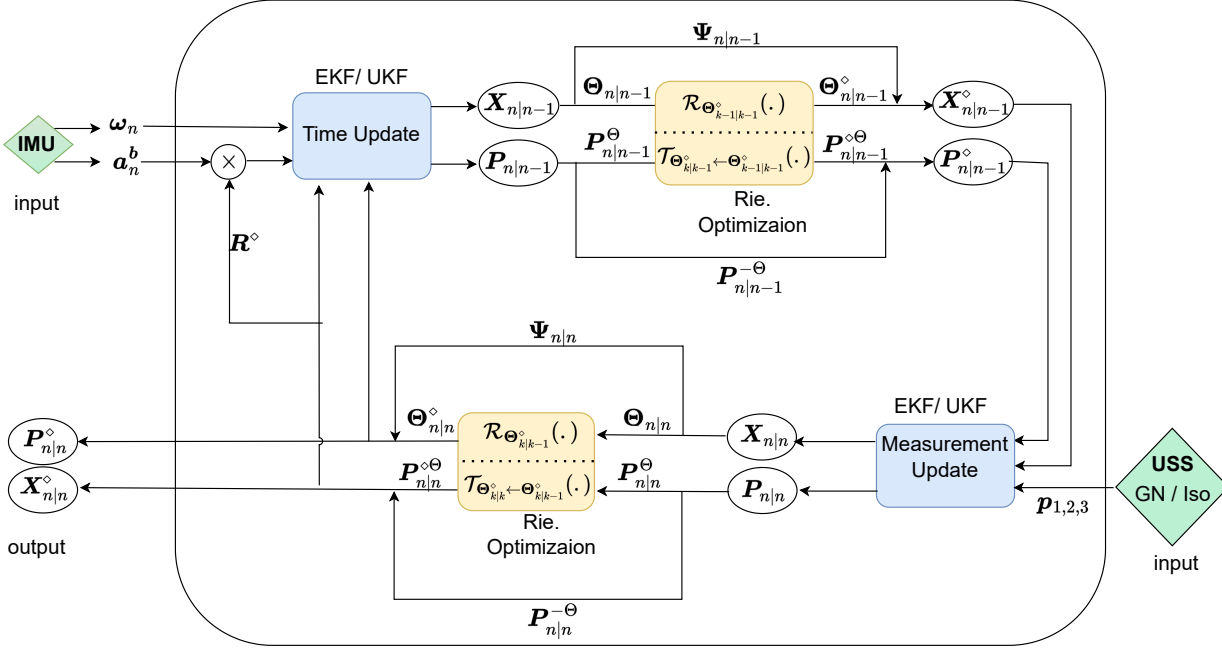


Figure 5.1: Descriptive diagram of the proposed Algorithm 1 (EKFRie) and proposed Algorithm 2 (UKFRie)

handles the estimation of the target's position and orientation based on IMU readings. The first four steps of this loop correspond to the conventional EKF, while steps 5 – 7 incorporate the Riemannian elements of the time update process.

In contrast, the measurement update stage, represented by the inner loop, focuses on correcting error drift using the MS's readings. Steps 9 – 13 of this loop align with the conventional EKF, while steps 14 – 16 integrate the Riemannian elements of the measurement update process. The UKFRie algorithm maintains the same overall structure but replaces the conventional EKF section with the conventional UKF counterpart.

In the pseudo codes presented in this section, the superscript ($\diamond$) indicates those quantities subjected to $SO(3)$; and $\mathbf{P}^\Theta$ indicates the submatrix of $\mathbf{P}$ associated with the elements in the state vector that represent the rotation matrix (Θ). These two

algorithms will be discussed under real scenarios and assumptions in the next chapter.

Algorithm 1 : Proposed EKFRie algorithm

Input: $\hat{\mathbf{X}}_{k-1|k-1}^\diamond$, $\mathbf{y}_{ms,k} = [\mathbf{p}_1^\top, \mathbf{p}_2^\top, \mathbf{p}_3^\top]^\top$, $\mathbf{a}_{k-1}^b$, ω_{k-1} , $\mathbf{P}_{k-1|k-1}^\diamond$,
and $\mathbf{p}_{1,2,3}^b$.

Output: $\hat{\mathbf{X}}_{k|k}^\diamond$ and $\mathbf{P}_{k|k}^\diamond$.

Steps:

- 1: $k_1 = IMU_{DataRate}/MS_{DataRate}$
- repeat for** $k > 0$:
- 2: Find $g(\mathbf{X}_{k-1|k-1}^\diamond)$ by using (4.14)
- 3: Find $\hat{\mathbf{X}}_{k|k-1}$ by using (4.12) and (4.13)
- 4: $\mathbf{P}_{k|k-1} \leftarrow \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^\top + g(\mathbf{X}_{k-1|k-1}^\diamond)\mathbf{Q}g(\mathbf{X}_{k-1|k-1}^\diamond)^\top$
- 5: $\hat{\Theta}_{k|k-1}^\diamond \leftarrow \mathcal{R}_{\Theta_{k-1|k-1}^\diamond}(T\Pi\omega_{k-1})$
- 6: $\mathbf{P}_{k|k-1}^{\diamond\Theta} \leftarrow \mathbf{V}^\diamond\Lambda\mathbf{V}^{\diamond\top} \leftarrow \mathbf{V}^\diamond \leftarrow \mathcal{T}_{\Theta_{k|k-1}^\diamond \leftarrow \Theta_{k-1|k-1}^\diamond}(\mathbf{V}) \leftarrow \mathbf{V}\Lambda\mathbf{V}^\top \leftarrow eig(\mathbf{P}_{k-1|k-1}^{\diamond\Theta})$
- 7: $\hat{\mathbf{X}}_{k|k-1}^\diamond \leftarrow \begin{bmatrix} \hat{\Theta}_{k|k-1}^\diamond & \hat{\Psi}_{k|k-1} \end{bmatrix}^\top$
- 8: **if** $k_1 = IMU_{DataRate}/MS_{DataRate}$ **then**
- 9: Find $\hat{\mathbf{y}}_{imu,k|k-1}$ by using (4.5)
- 10: $\mathbf{H}_k \leftarrow \frac{\partial h(\mathbf{X})}{\partial \mathbf{X}}|_{\mathbf{X}=\hat{\mathbf{X}}_{k|k-1}^\diamond}$
- 11: $\mathbf{K}_{f,k} \leftarrow \mathbf{P}_{k|k-1}^\diamond \mathbf{H}_k^\top (\mathbf{H}_k \mathbf{P}_{k|k-1}^\diamond \mathbf{H}_k^\top + \mathbf{Z}_k)^{-1}$
- 12: $\hat{\mathbf{X}}_{k|k} \leftarrow \hat{\mathbf{X}}_{k|k-1}^\diamond + \mathbf{K}_{f,k}(\mathbf{y}_{ms,k} - \hat{\mathbf{y}}_{imu,k|k-1})$
- 13: $\mathbf{P}_{k|k} \leftarrow (\mathbf{I} - \mathbf{K}_{f,k}\mathbf{H}_k)\mathbf{P}_{k|k-1}^\diamond$
- 14: $\hat{\Theta}_{k|k}^\diamond \leftarrow \mathcal{R}_{\Theta_{k|k-1}^\diamond}(\text{the corresponding part of } \mathbf{K}_{f,k}(\mathbf{y}_{ms,k} - \hat{\mathbf{y}}_{imu,k|k-1}) \text{ to } \Theta)$
- 15: $\mathbf{P}_{k|k}^{\diamond\Theta} \leftarrow \mathbf{V}^\diamond\Lambda\mathbf{V}^{\diamond\top} \leftarrow \mathbf{V}^\diamond \leftarrow \mathcal{T}_{\Theta_{k|k}^\diamond \leftarrow \Theta_{k|k-1}^\diamond}(\mathbf{V}) \leftarrow \mathbf{V}\Lambda\mathbf{V}^\top \leftarrow eig(\mathbf{P}_{k|k-1}^{\diamond\Theta})$
- 16: $\hat{\mathbf{X}}_{k|k}^\diamond \leftarrow \begin{bmatrix} \hat{\Theta}_{k|k}^\diamond & \hat{\Psi}_{k|k} \end{bmatrix}^\top$
- 17: $k_1 = 0$
- 18: **end if**
- 19: $k_1 = k_1 + 1$
- end.**

Algorithm 2 :Proposed UKFRie algorithm

Input: $\hat{\mathbf{X}}_{k-1|k-1}^\diamond$, $\mathbf{y}_{ms,k} = [\mathbf{p}_1^\top, \mathbf{p}_2^\top, \mathbf{p}_3^\top]^\top$, $\mathbf{a}_{k-1}^b$, ω_{k-1} , $\mathbf{P}_{k-1|k-1}^\diamond$, and $\mathbf{p}_{1,2,3}^b$.

Output: $\hat{\mathbf{X}}_{k|k}^\diamond$ and $\mathbf{P}_{k|k}^\diamond$.

Steps:

- 1: $k_1 = IMU_{DataRate}/MS_{DataRate}$
 - repeat for** $k > 0$:
 - 2: Finding sigma points of $\hat{\mathbf{X}}_{k-1|k-1}^\diamond$ by using following equations with $\iota = 1$ and $\alpha = 0.001$:

$$[\mathbf{a}_1 \quad \mathbf{a}_2 \quad \dots \quad \mathbf{a}_L] \leftarrow \mathbf{P}_{k-1|k-1}^{\diamond 1/2}, \quad \rho \leftarrow \alpha^2(L + \iota) - L, \quad \xi_0 \leftarrow \frac{\rho}{L + \rho},$$

$$\chi_0 \leftarrow \hat{\mathbf{X}}_{k-1|k-1}^\diamond, \quad \chi_{\ell \setminus \ell + L} \leftarrow \hat{\mathbf{X}}_{k-1|k-1}^\diamond (+ \setminus -) (\frac{L}{1 - \xi_0})^{1/2} \mathbf{a}_\ell \quad \text{and} \quad \xi_\ell \leftarrow \frac{1}{2(L + \rho)}$$
 for $\ell = 1, 2, \dots, 2L$
 - 3: Finding $g(\chi_\ell)$ by using (4.14)
 - 4: Finding $\mathbf{X}_{k,\ell}$ by using (4.12) and (4.13)
 - 5: $\hat{\mathbf{X}}_{k|k-1} \leftarrow \sum_{\ell=0}^{2L} \xi_\ell \mathbf{X}_{k,\ell}$
 - 6: $\mathbf{P}_{k|k-1} \leftarrow \sum_{\ell=0}^{2L} \xi_\ell (\mathbf{X}_{k,\ell} - \hat{\mathbf{X}}_{k|k-1})(\mathbf{X}_{k,\ell} - \hat{\mathbf{X}}_{k|k-1})^\top + \xi_\ell g(\chi_\ell) \mathbf{Q}_k g(\chi_\ell)^\top$
 - 7: $\hat{\Theta}_{k|k-1}^\diamond \leftarrow \mathcal{R}_{\Theta_{k-1|k-1}^\diamond}(\sum_{\ell=0}^{2L} \xi_\ell T \Pi_\ell \omega_{k-1})$ where Π_ℓ corresponding to the χ_ℓ
 - 8: $\mathbf{P}_{k|k-1}^{\diamond \Theta} \leftarrow \mathbf{V}^\diamond \Lambda \mathbf{V}^{\diamond \top} \leftarrow \mathbf{V}^\diamond \leftarrow \mathcal{T}_{\Theta_{k|k-1}^\diamond \leftarrow \Theta_{k-1|k-1}^\diamond}(\mathbf{V}) \leftarrow \mathbf{V} \Lambda \mathbf{V}^\top \leftarrow eig(\mathbf{P}_{k-1|k-1}^{\diamond \Theta})$
 - 9: $\mathbf{P}_{k|k-1}^{\diamond \Theta} \leftarrow nearestSPD(\mathbf{P}_{k|k-1}^{\diamond \Theta})$
 - 10: $\hat{\mathbf{X}}_{k|k-1}^\diamond \leftarrow [\hat{\Theta}_{k|k-1}^\diamond \quad \hat{\Psi}_{k|k-1}]^\top$
 - 11: **if** $k_1 = IMU_{DataRate}/MS_{DataRate}$ **then**
 - 12: Finding $\hat{\mathbf{y}}_{k,\ell}$ by using Eq. (4.5) and utilizing the same χ_ℓ
 - 13: $\hat{\mathbf{y}}_{imu,k|k-1} \leftarrow \sum_{\ell=0}^{2L} \xi_\ell \hat{\mathbf{y}}_{k,\ell}$
 - 14: $\delta_{e,k} \leftarrow \sum_{\ell=0}^{2L} \xi_\ell (\hat{\mathbf{y}}_{k,\ell} - \hat{\mathbf{y}}_{imu,k|k-1})(\hat{\mathbf{y}}_{k,\ell} - \hat{\mathbf{y}}_{imu,k|k-1})^\top + \mathbf{Z}_k$
 - 15: $\delta_{xe,k} \leftarrow \sum_{\ell=0}^{2L} \xi_\ell (\mathbf{X}_{k,\ell} - \hat{\mathbf{X}}_{k|k-1})(\hat{\mathbf{y}}_{k,\ell} - \hat{\mathbf{y}}_{imu,k|k-1})^\top$
 - 16: $\mathbf{K}_{f,k} \leftarrow \delta_{xe,k} \delta_{e,k}^{-1}$
 - 17: $\hat{\mathbf{X}}_{k|k} \leftarrow \mathbf{X}_{k|k-1}^\diamond + \mathbf{K}_{f,k}(\mathbf{y}_{ms,k} - \hat{\mathbf{y}}_{imu,k|k-1})$
 - 18: $\mathbf{P}_{k|k} \leftarrow \mathbf{P}_{k|k-1}^\diamond - \mathbf{K}_{f,k} \delta_{e,k} \mathbf{K}_{f,k}^\top$
 - 19: $\hat{\Theta}_{k|k}^\diamond \leftarrow \mathcal{R}_{\Theta_{k|k-1}^\diamond}(\text{the corresponding part of } \mathbf{K}_{f,k}(\mathbf{y}_{ms,k} - \hat{\mathbf{y}}_{imu,k|k-1}) \text{ to } \Theta)$
 - 20: $\mathbf{P}_{k|k}^{\diamond \Theta} \leftarrow \mathbf{V}^\diamond \Lambda \mathbf{V}^{\diamond \top} \leftarrow \mathbf{V}^\diamond \leftarrow \mathcal{T}_{\Theta_{k|k}^\diamond \leftarrow \Theta_{k|k-1}^\diamond}(\mathbf{V}) \leftarrow \mathbf{V} \Lambda \mathbf{V}^\top \leftarrow eig(\mathbf{P}_{k|k-1}^{\diamond \Theta})$
 - 21: $\mathbf{P}_{k|k}^{\diamond \Theta} \leftarrow nearestSPD(\mathbf{P}_{k|k}^{\diamond \Theta})$
 - 22: $\hat{\mathbf{X}}_{k|k}^\diamond \leftarrow [\hat{\Theta}_{k|k}^\diamond \quad \hat{\Psi}_{k|k}]^\top$
 - 23: $k_1 = 0$
 - 24: **end if**
 - 25: $k_1 = k_1 + 1$
 - end while.**
-

CHAPTER 6

RESULTS AND DISCUSSION

This chapter evaluates the effectiveness of the proposed algorithms (EKFRie and UKFRie) by comparing them to the conventional EKF and UKF in terms of the RMSE. Additionally, the impact of incorporating the isosceles triangle manifold of the transmitters' geometry is discussed. The position RMSE is calculated according to the following formula, where i refers to X , Y , and Z axes, $iter$ refers to the iteration, and p and $\hat{p}$ refer to the true and estimated target centroid position, respectively:

$$\text{RMSE} = \text{avraging over time of } \left(\sum_{i=1}^3 \sqrt{\frac{1}{N} \sum_{iter=1}^N (p_{i,iter} - \hat{p}_{i,iter})^2} \right) \quad (6.1)$$

The orientation RMSE is analogous to the position RMSE, except that the target centroid position p_i is supplanted by the orientation angle of the target centroid around i -axis.

This thesis utilizes a simulation approach to assess the performance of the proposed algorithms against conventional algorithms using MATLAB. The simulation setup is detailed in the opening section of this chapter. The second section delves into the

performance of the proposed algorithms in a static scenario, where the target position and orientation remain constant. The final section examines the performance under dynamic conditions.

6.1 Simulation Setup

The simulations were conducted using a MATLAB program running on a 64 – *bit* computer with 6GB of RAM, powered by an Intel(R) Core(TM) i7-3630QM CPU (2.40GHz). All simulations were run for 3000 iterations. The MPU-9250 IMU was chosen for this thesis due to its small size, low cost, and low complexity [83]. It includes a 3–axis gyroscope, 3–axis accelerometer, and 3–axis magnetometer. The MPU-9250 has a gyroscope update rate range of 4 – 8000Hz and an accelerometer update rate range of 4 – 4000Hz. In our simulations, the gyroscope and accelerometer update rates were fixed to 10Hz to reduce simulation time, unless otherwise stated.

IMU noise characteristics were incorporated into the simulations based on the MPU-9250 datasheet [83]. Gyroscope noise was modeled with a variance of $5 \times 10^{-4} \text{degree/s}$, and accelerometer noise was modeled with a variance of $0.43 \times 10^{-3} \text{m/s}^2$. These variances were calculated based on the gyroscope’s rate noise spectral density ($0.01 \text{degree/s/sqrt(Hz)}$) and the accelerometer’s noise power spectral density ($300 \mu\text{g/sqrt(Hz)}$), as provided in the datasheet. The variances were calculated using the following formulas: $\sigma_a^2 = (300 \times 9.8 \times 10^{-6})^2 \times (f_s/2)$ and $\sigma_\omega^2 = (0.01)^2 \times (f_s/2)$, where f_s is the updating rates.

The accuracy of measurement systems can vary depending on the type of sensor

used. For example, Cricket RF-ultrasonic sensors [84] provide $1 - 3cm$ position precision, Marvelmind sensors [85] provide accuracy of $\pm 2cm$ or better, and UWB DW1000 sensors [86] provide around $10cm$ position precision. Ultrasonic systems generally provide accuracy within the range $3 - 100cm$ [23]. Therefore, this work studies the effect of MS standard deviation values within the range $1cm$ to $1m$. The measurement system update rate was fixed at $1Hz$ which is lower than the IMU's update rate to be consistent with real-world applications. The isosceles triangle used in this work had a base and altitude of $10cm$ and $30cm$, respectively.

The study considered four different paths, as shown in Fig. 6.1, to ensure the generality of the results. These paths take 10, 5, 5, and 85 seconds, respectively, with a maximum velocity of less than $2m/s$ to simulate human and robot [87] motion.

6.2 Numerical Results - Static Case

This section focuses on evaluating the proposed algorithms for static scenarios. To simulate a static case, we employed the previously described model with the velocity set to zero. Fig. 6.2 illustrates the RMSE for the estimated position and orientation plotted against the reciprocal of different values of the measurement system's standard deviation (σ_r). The figure demonstrates that the EKF outperforms the UKF regarding both position and orientation estimation. While the RMSE of the EKF decreases linearly as $1/\sigma_r$ increases, the RMSE of the UKF exhibits a less pronounced decrease, highlighting the EKF's advantage in this scenario. This superiority can be attributed to the relatively low degree of nonlinearity in our model, which reduces the impact of

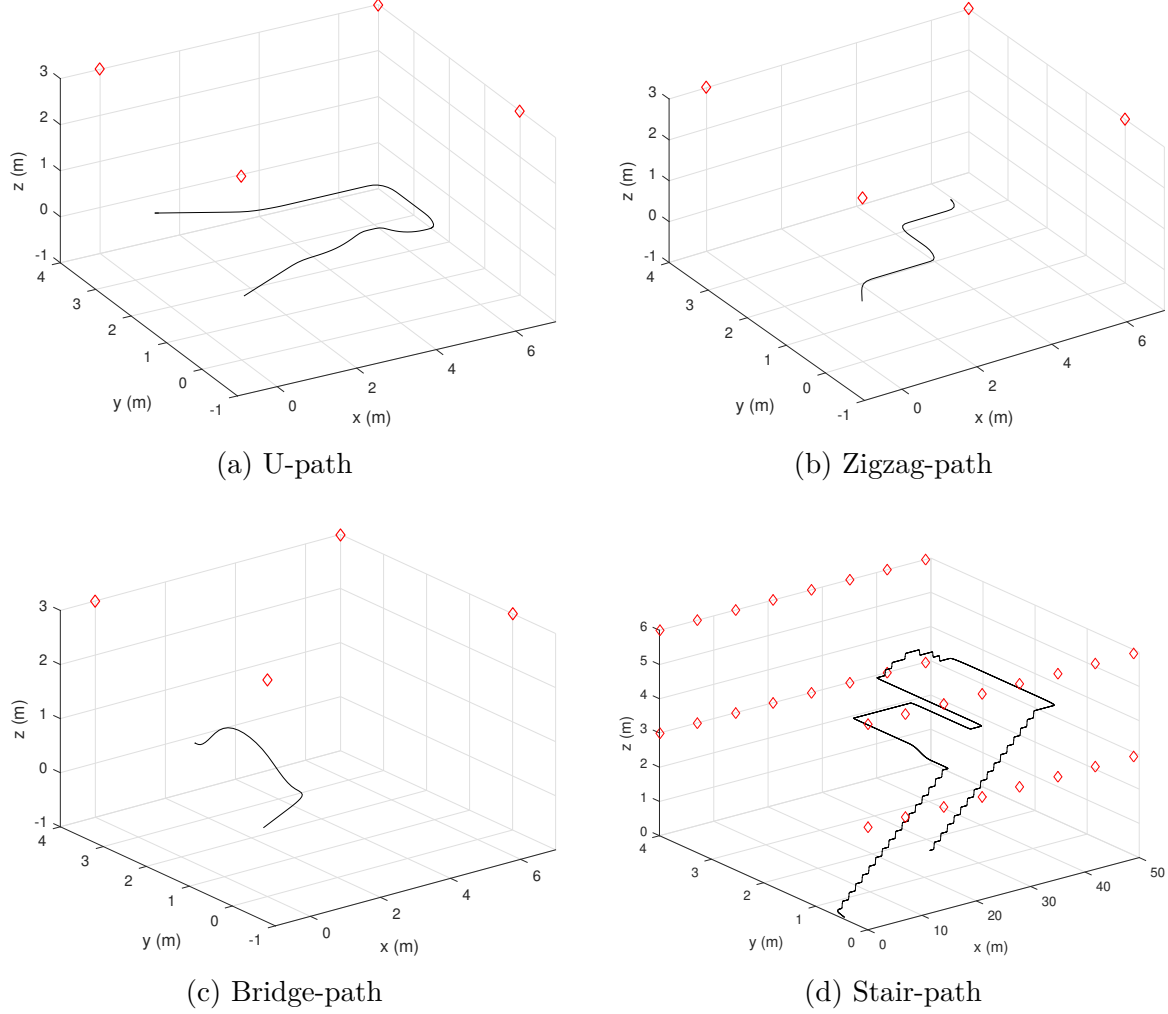


Figure 6.1: The four true paths under consideration, where the red diamonds refer to the beacons

the linearization error associated with the Taylor expansion employed in the EKF.

Furthermore, Fig. 6.2 shows that the performance of all EKFs (EKF, EKFIso, EKFRie, and EKFRieIso) are aligned closely, as the performance of all UKFs (UKF, UKFIso, UKFRie, and UKFRieIso). In the legend, EKFs refer to all four algorithms related to the EKF while UKFs refer to those related to the UKF.

That is because the angular velocity is zero, which implies that the orientation matrix remains constant in the static scenario. Consequently, EKF and UKF perform

better in this case, rendering the exploitation of the $SO(3)$ structure of the rotation matrix redundant. This observation suggests that Riemannian optimization is practically useless in static scenarios.

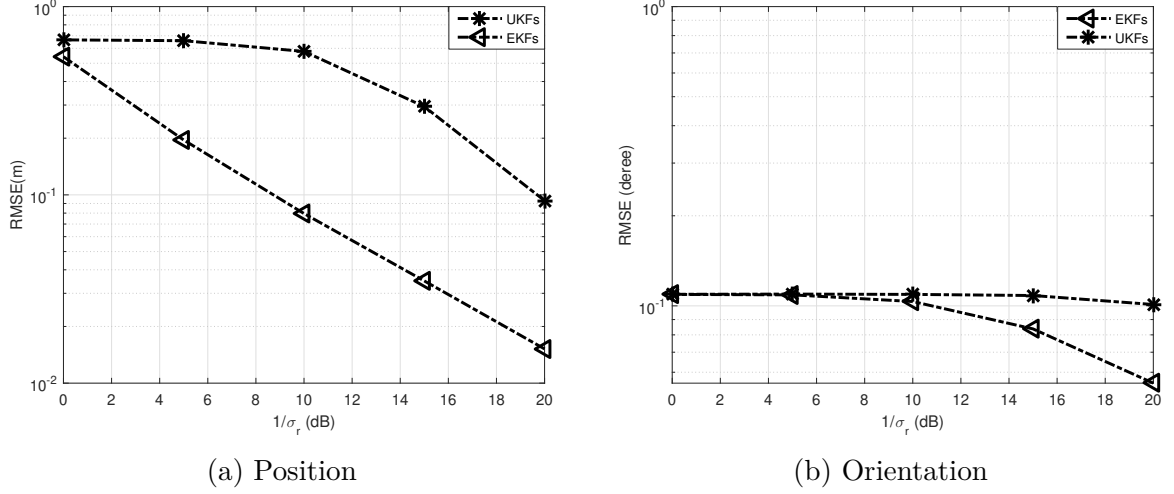


Figure 6.2: RMSE vs different values of $1/\sigma_r$ dB for the static case

6.3 Numerical Results - Dynamic Case

The performance of the proposed algorithms (EKFRie and UKFRie) is compared to the conventional EKF and UKF in terms of the RMSE. Across all paths and noise levels σ_r , our proposed algorithms consistently surpass the EKF and UKF in terms of both position and, in most cases, orientation. For U-path and $1/\sigma_r = 10$ dB, the RMSE for the EKF and EKFRie were $1.08m$ and $0.31m$, respectively, and for the UKF and UKFRie were $1.39m$ and $0.39m$, respectively. For Stair-path and $1/\sigma_r = 10$ dB, the RMSE for the EKF and EKFRie were $1.22m$ and $0.206m$, respectively, and for the UKF and UKFRie were $17.79m$ and $5.139m$, respectively. It should be noted that UKF performed poorly on Stair-path (the longest path). Although the RMSE

of UKFRie is still relatively large, it has significantly improved UKF's performance. Table. 6.1 and 6.2 summarize the position RMSE for all algorithms under all values of $1/\sigma_r$ for U-path and Stair-path, respectively. The smallest RMSE for each $1/\sigma_r$ value is in bold to indicate the best algorithm.

Table 6.1: Position RMSE for U-path

$1/\sigma_r =$	0 dB	5 dB	10 dB	15 dB	20 dB
EKF	1.381	1.339	1.078	0.665	0.356
EKFIso	1.381	1.339	1.078	0.665	0.356
EKFRie	0.390	0.379	0.307	0.188	0.120
EKFRieIso	0.390	0.379	0.307	0.188	0.120
UKF	1.389	1.388	1.386	1.360	1.172
UKFIso	1.389	1.388	1.386	1.360	1.172
UKFRie	0.391	0.391	0.390	0.382	0.339
UKFRieIso	0.391	0.391	0.389	0.383	0.339

Table 6.2: Position RMSE for Stair-path

$1/\sigma_r =$	0 dB	5 dB	10 dB	15 dB	20 dB
EKF	6.488	2.763	1.216	0.574	0.241
EKFIso	6.388	2.750	1.204	0.560	0.237
EKFRie	1.115	0.469	0.206	0.095	0.050
EKFRieIso	0.972	0.458	0.205	0.102	0.067
UKF	23.280	23.830	17.790	9.988	3.392
UKFIso	23.410	27.160	19.880	10.410	3.452
UKFRie	5.325	5.570	5.139	2.659	1.154
UKFRieIso	5.228	5.971	5.473	2.915	1.218

Fig. 6.3 illustrates the estimated paths obtained using different algorithms compared to the true path, highlighting the benefits of Riemannian optimization. Fig. 6.3a shows the estimated path using the EKFs at $\sigma_r = 1 \times 10^{-4}$. This figure demonstrates the superior performance of Riemannian optimization. Additionally, the influence of the measurement system is evident, as the estimated path is drawn back towards the true path after each second. Fig. 6.3b depicts the estimated path using the UKFs at

$\sigma_r = 1 \times 10^{-1}$. Because the variance of the measurement system noise is relatively high here, the corrective effect of the measurements is less pronounced. Nevertheless, the advantage of the Riemannian optimization is still evident in this case, as it provides more accurate path estimates compared to the other method.

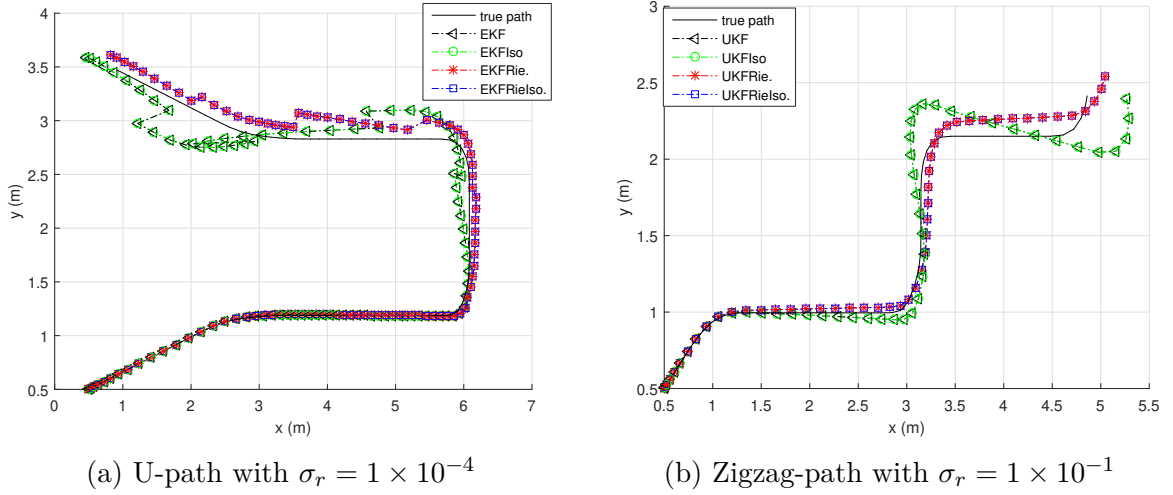


Figure 6.3: Estimated path of different algorithms for different σ_r

Remarkably, incorporating Riemannian optimization generally leads to a noticeable reduction in RMSE. Fig. 6.4 and 6.6 illustrate the RMSE of the estimated position and orientation versus $1/\sigma_r$ for the first and fourth paths, respectively. In the legend, “(Iso)” denotes the use of the isosceles triangle manifold algorithm during the measurement stage to estimate transmitters’ positions, while the Gauss-Newton algorithm was employed for the others.

The isosceles triangle manifold algorithm outperforms the Gauss-Newton algorithm in estimating transmitter positions, as mentioned in [78] and as shown in our simulation (not included as it falls beyond the scope of this thesis). However, the isosceles triangle manifold’s advantage over the Gauss-Newton algorithm is not fully realized by the considered KFs tracking algorithms.

This is because the isosceles triangle manifold's advantage is inversely proportional to signal to noise ratio (SNR). That implies the isosceles triangle manifold algorithm has a greater advantage over the Gauss-Newton algorithm when the ranges are more noisy. This is evident from the results of [78] and our simulation.

However, the relationship between range noise and the contribution of the MS on KFs is also inversely proportional. This means that as the range noise increases, the contribution of the MS to the KFs decreases. As a result, the advantage of the isosceles triangle manifold algorithm is not as noticeable in the considered KFs tracking algorithms. Despite this, a slight improvement in RMSE for EKFRie with the isosceles triangle manifold algorithm at $1/\sigma_r = 0$ dB can be observed in Fig. 6.4b.

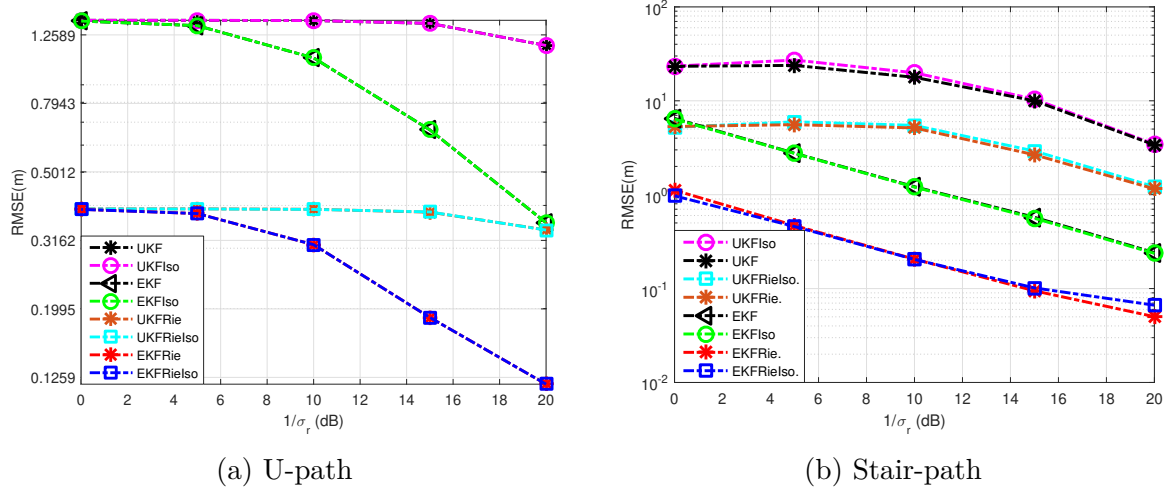


Figure 6.4: Positions' RMSE vs different values of $1/\sigma_r$ dB

In terms of time efficiency, EKF and EKFRie outperform UKF and UKFRie due to the time-consuming sigma point generation process in the latter, especially for high-dimensional state vectors. Gauss-Newton algorithm is more efficient than the isosceles triangle manifold algorithm, which suffers from the limitations of the steepest-descent function.

To investigate how varying σ_r affects the enhancement of the proposed algorithms, Fig. 6.5 shows the ratio between the RMSE of the proposed and conventional algorithms. A ratio of 0.25 indicates that the proposed algorithm's RMSE is 25% of the conventional algorithm's RMSE. Our results demonstrate that the improvement increases as $1/\sigma_r$ decreases in most cases.

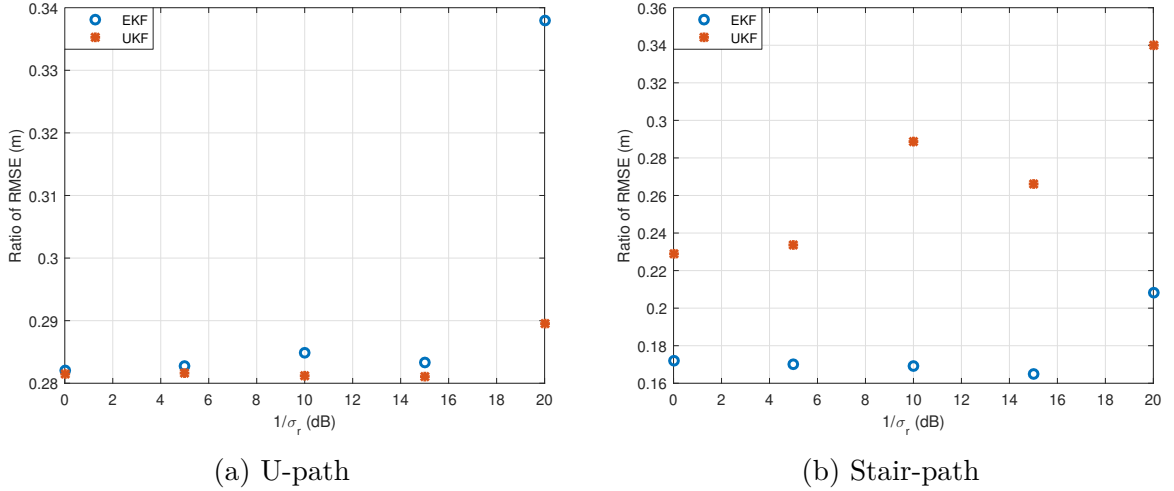


Figure 6.5: The *RMSE* ratio of the Riemannian algorithms vs conventional algorithms for the position under different values of $1/\sigma_r$ dB

The proposed algorithms show marginal improvements in orientation estimation compared to standard Kalman filters, as illustrated in Fig. 6.6. The figure also demonstrates that the impact of σ_r on orientation RMSE diminishes as σ_r increases as the plot becomes flatter. This is because the contribution of the MS into the Kalman filter estimate decreases with increasing σ_r . Fig. 6.6b also highlights that the advantage of our proposed algorithms over conventional methods for orientation can weaken as $1/\sigma_r$ decreases as is the case with UKFs.

The impact of increasing the IMU updating rate on the RMSE of estimated position and orientation is examined in Fig. 6.7 and 6.8 for $\sigma_r^2 = 1 \times 10^{-1}$ and 1×10^{-3} ,

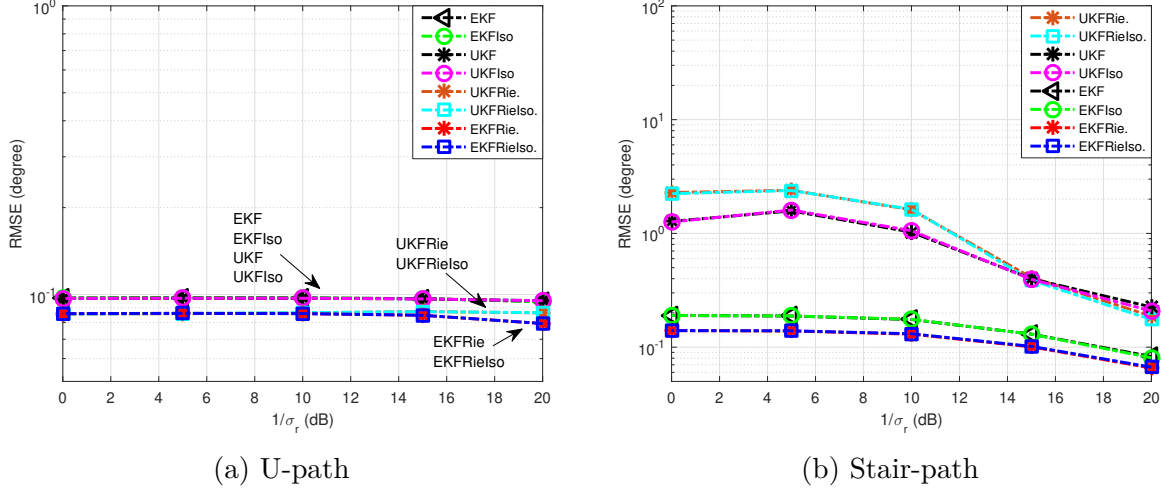


Figure 6.6: Orientation's RMSE vs different values of $1/\sigma_r$ dB

respectively. As expected, the results show that higher updating rates lead to better performance of the algorithms. However, the improvement diminishes with increasing updating rates, suggesting that there is a threshold beyond which further increases in the updating rate yield little benefit.

For example, the position RMSE of EKFRie and UKFRie decreased by around 52.6% when the IMU's data rate increased from 5 to 10 *samples/s* when $\sigma_r^2 = 1 \times 10^{-1}$. However, when the IMU data rate increased from 10 to 50 *samples/s*, the RMSE decreased by only 39.6%.

Analyzing the cumulative distribution of the RMSE provides a more comprehensive understanding of the error distribution and reduces the influence of outliers. Our results show that the proposed algorithms consistently outperform the EKF and UKF across all test cases in terms of position. The CDF of the RMSE of position estimation for Bridge-path at two different of σ_r^2 is shown in Fig. 6.9. The CDFs of RMSE for EKFRie and UKFRie at $\sigma_r^2 = 1 \times 10^{-3}$ indicate that 90% of the RMSE lie below 0.22m and 0.27m, respectively. In contrast, the corresponding values for EKF and

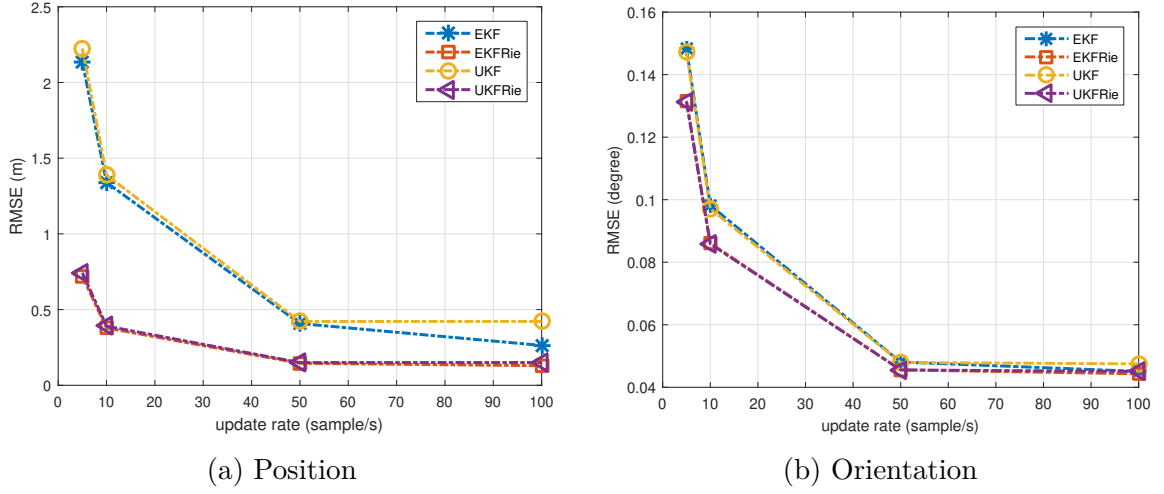


Figure 6.7: RMSE of U-path vs IMU's updating rate for $\sigma_r^2 = 1 \times 10^{-1}$

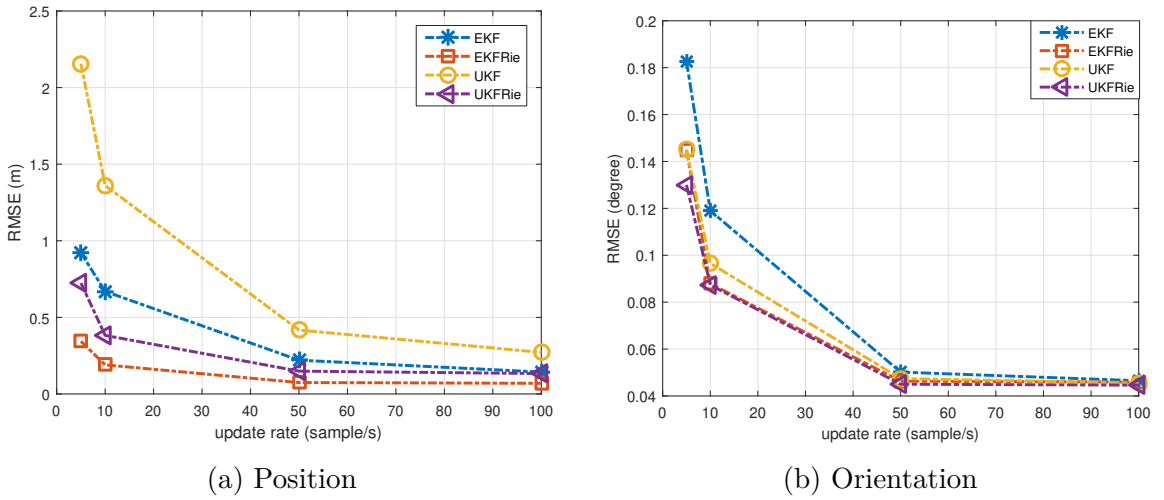
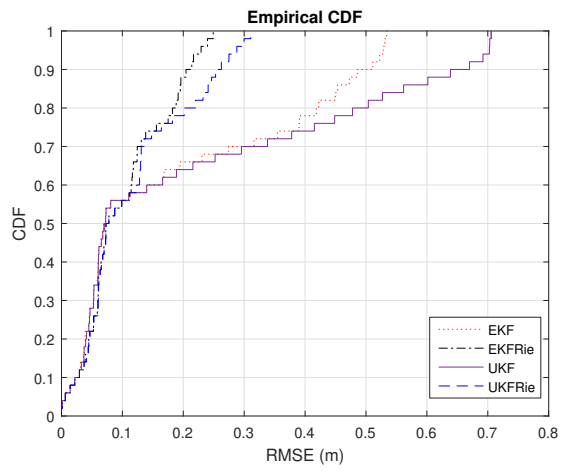
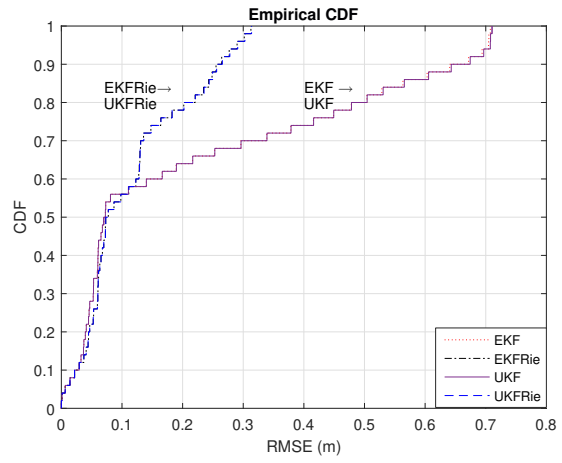


Figure 6.8: RMSE of U-path vs IMU's updating rate for $\sigma_r^2 = 1 \times 10^{-3}$

UKF lie below $0.52m$ and $0.67m$, respectively, which implies an outperforming of the proposed algorithms.



(a) $\sigma_r^2 = 1 \times 10^{-3}$



(b) $\sigma_r^2 = 1 \times 10^{-1}$

Figure 6.9: CDF of the RMSE for Bridge-path for different values of σ_r^2

CHAPTER 7

CONCLUSION

This study explores whether utilizing the $SO(3)$ manifold structure of the rotation matrix to conduct Riemannian optimization can enhance the performance of target tracking in indoor environments. To achieve this, the conventional EKF and UKF were modified by incorporating Riemannian optimization tools, namely the retraction and vector transport, to update the state vector and covariance matrix. Additionally, the impact of adopting the isosceles triangle manifold geometry of the attached transmitters was also examined.

7.1 Summary of Results

Our results demonstrated that in static scenarios, incorporating the $SO(3)$ structure did not improve the tracking performance for both position and orientation. However, for dynamic scenarios, the proposed EKFRie and UKFRie algorithms outperformed the standard EKF and UKF in terms of RMSE for position and orientation. While the RMSE for EKF and UKF were $1.22m$ and $17.79m$ for Stair-path, respectively,

they were $0.21m$ and $5.14m$ for EKFRie and UKFRie, respectively. Our results also revealed that the advantage of the proposed algorithms increased as the σ_r increased. Analyzing the CDF further confirms the superior performance of the proposed algorithms. While the 90th percentile of the RMSE for EKF and UKF fall below $0.52m$ and $0.67m$, respectively, they fall below $0.22m$ and $0.27m$ for EKFRie and UKFRie, respectively. Regarding adopting the isosceles triangle manifold algorithm, our results did not indicate that it effectively enhanced the tracking performance.

7.2 Future Directions

Several avenues exist for further extending the scope of this work. One promising approach involves employing the exponential map and parallel transport as alternatives to retraction and vector transport to investigate their potential for performance enhancement. Additionally, the impact of varying gyroscope and accelerometer variances, as well as adjusting the number of anchors within fixed-size spaces, could be explored. Furthermore, investigating the application of different initialization techniques for the isosceles triangle manifold algorithm within the measurement system could potentially lead to improved performance. These avenues offer valuable opportunities to optimize the approach and enhance its overall effectiveness.

REFERENCES

- [1] N. Boumal, “An introduction to optimization on smooth manifolds,” *Available online, May*, vol. 3, 2020.
- [2] A. A. Khudhair, S. Q. Jabbar, M. Q. Sulttan, and D. Wang, “Wireless indoor localization systems and techniques: survey and comparative study,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 3, no. 2, pp. 392–409, 2016.
- [3] S. Ramnath, A. Javali, B. Narang, P. Mishra, and S. K. Routray, “Iot based localization and tracking,” in *2017 International Conference on IoT and Application (ICIOT)*, 2017, pp. 1–4.
- [4] K. A. Kordi, A. Alhammadi, M. Roslee, M. Y. Alias, and Q. Abdullah, “A review on wireless emerging iot indoor localization,” in *2020 IEEE 5th International Symposium on Telecommunication Technologies (ISTT)*, 2020, pp. 82–87.
- [5] K. Lin, W. Wang, Y. Bi, M. Qiu, and M. M. Hassan, “Human localization based on inertial sensors and fingerprints in the industrial internet of things,” *Computer Networks*, vol. 101, pp. 113–126, 2016.

- [6] A. Khanna and R. Anand, “Iot based smart parking system,” in *2016 international conference on internet of things and applications (IOTA)*. IEEE, 2016, pp. 266–270.
- [7] B. Alsinglawi, M. Elkhodr, Q. V. Nguyen, U. Gunawardana, A. Maeder, and S. Simoff, “Rfid localisation for internet of things smart homes: a survey,” *arXiv preprint arXiv:1702.02311*, 2017.
- [8] C. K. M. Lee, C. Ip, T. Park, and S. Chung, “A bluetooth location-based indoor positioning system for asset tracking in warehouse,” in *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 2019, pp. 1408–1412.
- [9] A. Howard, M. Matark, and G. Sukhatme, “Localization for mobile robot teams using maximum likelihood estimation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2002, pp. 434–439 vol.1.
- [10] K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann, “High-speed laser localization for mobile robots,” *Robotics and Autonomous Systems*, vol. 51, no. 4, pp. 275–296, 2005. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889005000254>
- [11] D. Lee, G. Kang, B. Kim, and D. H. Shim, “Assistive delivery robot application for real-world postal services,” *IEEE Access*, vol. 9, pp. 141 981–141 998, 2021.
- [12] M. Gareis, A. Parr, J. Trabert, T. Mehner, M. Vossiek, and C. Carlowitz, “Stock-taking robots, automatic inventory, and 3d product maps: The smart warehouse

- enabled by uhf-rfid synthetic aperture localization techniques,” *IEEE Microwave Magazine*, vol. 22, no. 3, pp. 57–68, 2021.
- [13] H. Peng, X. Yang, Y.-H. Su, and B. Hannaford, “Real-time data driven precision estimator for raven-ii surgical robot end effector position,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 350–356.
- [14] T. Wei and X. Zhang, “Gyro in the air: tracking 3d orientation of batteryless internet-of-things,” in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016, pp. 55–68.
- [15] G. Qinglei, L. Huawei, M. Shifu, and H. Jian, “Design of a plane inclinometer based on mems accelerometer,” in *2007 International Conference on Information Acquisition*. IEEE, 2007, pp. 320–323.
- [16] E.-E.-L. Lau and W.-Y. Chung, “Enhanced rssi-based real-time user location tracking system for indoor and outdoor environments,” in *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, 2007, pp. 1213–1218.
- [17] K. Bakhru, “A seamless tracking solution for indoor and outdoor position location,” in *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 3, 2005, pp. 2029–2033 Vol. 3.

- [18] P. Carbone, D. Petri, and A. Tsourdos, “I and m in navigation systems [framing i & m topics],” *IEEE Instrumentation and Measurement Magazine*, vol. 18, no. 3, pp. 36–39, 2015.
- [19] D. Dardari, P. Closas, and P. M. Djurić, “Indoor tracking: Theory, methods, and technologies,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, 2015.
- [20] M. Alhafnawi, H. B. Salameh, A. Masadeh, H. Al-Obiedollah, M. Ayyash, R. El-Khazali, and H. Elgala, “A survey of indoor and outdoor uav-based target tracking systems: Current status, challenges, technologies, and future directions,” *IEEE Access*, 2023.
- [21] J.-S. Gutmann, P. Fong, L. Chiu, and M. E. Munich, “Challenges of designing a low-cost indoor localization system using active beacons,” in *2013 IEEE conference on technologies for practical robot applications (TePRA)*. IEEE, 2013, pp. 1–6.
- [22] F. Zafari, A. Gkelias, and K. K. Leung, “A survey of indoor localization systems and technologies,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.
- [23] M. Yasir, S.-W. Ho, and B. N. Vellambi, “Indoor position tracking using multiple optical receivers,” *Journal of Lightwave Technology*, vol. 34, no. 4, pp. 1166–1176, 2015.

- [24] Y. Zhao, Y. Zhao, and C. Zhao, “A novel algebraic solution for moving target localization in multi-transmitter multi-receiver passive radar,” *Signal Processing*, vol. 143, pp. 303–310, 2018.
- [25] S. Busanelli and G. Ferrari, “Uwb-based tracking of autonomous vehicles with multiple receivers,” in *Communication and Networking: International Conference, FGCN 2010, Held as Part of the Future Generation Information Technology Conference, FGIT 2010, Jeju Island, Korea, December 13-15, 2010. Proceedings, Part I*. Springer, 2010, pp. 188–198.
- [26] A. Correa, M. B. Llado, A. Morell, and J. L. Vicario, “Indoor pedestrian tracking by on-body multiple receivers,” *IEEE Sensors Journal*, vol. 16, no. 8, pp. 2545–2553, 2016.
- [27] Y. Park, A. Choi, and K. Kim, “Single-channel multiple-receiver sound source localization system with homomorphic deconvolution and linear regression,” *Sensors*, vol. 21, no. 3, p. 760, 2021.
- [28] M. A. Shalaby, C. C. Cossette, J. L. Ny, and J. R. Forbes, “Ultra-wideband teach and repeat,” *arXiv preprint arXiv:2202.01134*, 2022.
- [29] E. Namazi, R. Mester, C. Lu, M. M. Log, and J. Li, “Improving vehicle localization with two low-cost gps receivers,” in *Innovations in Smart Cities Applications Volume 5: The Proceedings of the 6th International Conference on Smart City Applications*. Springer, 2022, pp. 177–186.

- [30] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, and H. Snoussi, “Target tracking using machine learning and kalman filter in wireless sensor networks,” *IEEE Sensors Journal*, vol. 14, no. 10, pp. 3715–3725, 2014.
- [31] “Man opt: Tool boxes for optimization on manifolds and matrices,” accessed: 2023-03-5. [Online]. Available: <https://www.manopt.org/>
- [32] R. M. P.-A. Absil and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton University Press 41 William Street, Princeton, New Jersey 08540, 2008.
- [33] Q. Li, R. Li, K. Ji, and W. Dai, “Kalman filter and its application,” in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 2015, pp. 74–77.
- [34] R. H. Delano, “A theory of target glint or angular scintillation in radar tracking,” *Proceedings of the IRE*, vol. 41, no. 12, pp. 1778–1784, 1953.
- [35] “Ball tracker speeds missile target detection,” [Online] <https://ieeexplore.ieee.org/document/6434355>, pp. 545–545, 1962.
- [36] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960. [Online]. Available: <https://doi.org/10.1115/1.3662552>
- [37] S. J. Julier and J. K. Uhlmann, “New extension of the kalman filter to nonlinear systems,” in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068. Spie, 1997, pp. 182–193.

- [38] R. Van Der Merwe, *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. Oregon Health & Science University, 2004.
- [39] E. A. Wan and R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee, 2000, pp. 153–158.
- [40] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.
- [41] A. H. Sayed, *Kalman Filter*. Cambridge University Press, 2022, vol. 2, p. 1154–1210.
- [42] C. Liu, P. Shui, G. Wei, and S. Li, “Modified unscented kalman filter using modified filter gain and variance scale factor for highly maneuvering target tracking,” *Journal of Systems Engineering and Electronics*, vol. 25, no. 3, pp. 380–385, 2014.
- [43] J. Wang, T. Zhang, X. Xu, and Y. Li, “A variational bayesian based strong tracking interpolatory cubature kalman filter for maneuvering target tracking,” *IEEE Access*, vol. 6, pp. 52 544–52 560, 2018.
- [44] X. Wang, Z. Xu, X. Gou, and L. Trajković, “Tracking a maneuvering target by multiple sensors using extended kalman filter with nested probabilistic-numerical linguistic information,” *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 2, pp. 346–360, 2019.

- [45] Z. Zhang, Q. Li, L. Han, X. Dong, Y. Liang, and Z. Ren, “Consensus based strong tracking adaptive cubature kalman filtering for nonlinear system distributed estimation,” *IEEE Access*, vol. 7, pp. 98 820–98 831, 2019.
- [46] V. N. Agarwal, A. Kanshi, N. V. Melarkode, H. Krishna, and M. K. Hota, “Neural network aided kalman filter to maximize accuracy,” in *2022 International Conference for Advancement in Technology (ICONAT)*. IEEE, 2022, pp. 1–4.
- [47] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. Van Sloun, and Y. C. Eldar, “Kalmannet: Neural network aided kalman filtering for partially known dynamics,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.
- [48] B. Millidge, A. Tschantz, A. Seth, and C. Buckley, “Neural kalman filtering,” *arXiv preprint arXiv:2102.10021*, 2021.
- [49] W. Zhou and J. Hou, “A new adaptive robust unscented kalman filter for improving the accuracy of target tracking,” *IEEE Access*, vol. 7, pp. 77 476–77 489, 2019.
- [50] B. Bićanić, I. Marković, and I. Petrović, “Multi-target tracking on riemannian manifolds via probabilistic data association,” *IEEE Signal Processing Letters*, vol. 28, pp. 1555–1559, 2021.
- [51] Y. Yun, I. Y.-H. Gu, and H. Aghajan, “Multi-view ml object tracking with online learning on riemannian manifolds by combining geometric constraints,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 2, pp. 185–197, 2013.

- [52] Y. Wu, B. Wu, J. Liu, and H. Lu, “Probabilistic tracking on riemannian manifolds,” in *2008 19th International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–4.
- [53] A. Mashtakov, R. Duits, Y. Sachkov, E. J. Bekkers, and I. Beschastnyi, “Tracking of lines in spherical images via sub-riemannian geodesics in $\mathfrak{so}(3)$,” *Journal of mathematical imaging and vision*, vol. 58, no. 2, pp. 239–264, 2017.
- [54] Z. H. Khan and I. Y.-H. Gu, “Bayesian online learning on riemannian manifolds using a dual model with applications to video object tracking,” in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2011, pp. 1402–1409.
- [55] —, “Tracking visual and infrared objects using joint riemannian manifold appearance and affine shape modeling,” in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2011, pp. 1847–1854.
- [56] E. Kraft, “A quaternion-based unscented kalman filter for orientation tracking,” in *Proceedings of the Sixth International Conference of Information Fusion*, vol. 1. IEEE Cairns, 2003, pp. 47–54.
- [57] J. Kwon and K. M. Lee, “Monocular slam with locally planar landmarks via geometric rao-blackwellized particle filtering on lie groups,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1522–1529.

- [58] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, “Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds,” *Information Fusion*, vol. 14, no. 1, pp. 57–77, 2013.
- [59] S. Hauberg, F. Lauze, and K. S. Pedersen, “Unscented kalman filtering on riemannian manifolds,” *Journal of mathematical imaging and vision*, vol. 46, no. 1, pp. 103–120, 2013.
- [60] N. B. da Silva, D. B. Wilson, and K. R. Branco, “Performance evaluation of the extended kalman filter and unscented kalman filter,” in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2015, pp. 733–741.
- [61] H. M. Menegaz, J. Y. Ishihara, and H. T. Kussaba, “Unscented kalman filters for riemannian state-space systems,” *IEEE Transactions on Automatic Control*, vol. 64, no. 4, pp. 1487–1502, 2018.
- [62] Y. Cao, A. Dhekne, and M. Ammar, “Itracku: tracking a pen-like instrument via uwb-imu fusion,” in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, 2021, pp. 453–466.
- [63] B. V. Krishnaveni, K. S. Reddy, and P. R. Reddy, “Indoor tracking by adding imu and uwb using unscented kalman filter,” *Wireless Personal Communications*, pp. 1–22, 2022.
- [64] R. Chen, Z. Li, F. Ye, G. Guo, S. Xu, L. Qian, Z. Liu, and L. Huang, “Precise indoor positioning based on acoustic ranging in smartphone,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

- [65] S. Xu, R. Chen, G. Guo, Z. Li, L. Qian, F. Ye, Z. Liu, and L. Huang, “Bluetooth, floor-plan, and microelectromechanical systems-assisted wide-area audio indoor localization system: Apply to smartphones,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 11, pp. 11 744–11 754, 2021.
- [66] J. Qi and G.-P. Liu, “A robust high-accuracy ultrasound indoor positioning system based on a wireless sensor network,” *Sensors*, vol. 17, no. 11, p. 2554, 2017.
- [67] C. Cai, R. Zheng, J. Li, L. Zhu, H. Pu, and M. Hu, “Asynchronous acoustic localization and tracking for mobile targets,” *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 830–845, 2019.
- [68] “Top common ways to localize a mobile robot.” [Online]. Available: <https://www.leorover.tech/post/top-common-ways-to-localize-a-mobile-robot>
- [69] I. A. Faisal, T. W. Purboyo, and A. S. R. Ansori, “A review of accelerometer sensor and gyroscope sensor in imu sensors on motion capture,” *J. Eng. Appl. Sci*, vol. 15, no. 3, pp. 826–829, 2019.
- [70] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, “Reviews on various inertial measurement unit (imu) sensor applications,” *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256–262, 2013.
- [71] “Physics.” [Online]. Available: https://d3bxy9euw4e147.cloudfront.net/oscms-prodcms/media/documents/Physics-WEB_Sab7RrQ.pdf
- [72] M. Freydin and B. Or, “Learning car speed using inertial sensors for dead reckoning navigation,” *IEEE Sensors Letters*, vol. 6, no. 9, pp. 1–4, 2022.

- [73] D. M. Bevly and B. Parkinson, “Cascaded kalman filters for accurate estimation of multiple biases, dead-reckoning navigation, and full state feedback control of ground vehicles,” *IEEE Transactions on Control Systems Technology*, vol. 15, no. 2, pp. 199–208, 2007.
- [74] B. Alavi and K. Pahlavan, “Modeling of the toa-based distance measurement error using uwb indoor radio measurements,” *IEEE communications letters*, vol. 10, no. 4, pp. 275–277, 2006.
- [75] D. Titterton, J. L. Weston, and J. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.
- [76] “A riemannian geometry for the special orthogonal group (the set of rotations).” [Online]. Available: https://www.manopt.org/manifold_documentation_rotations.html
- [77] P. P. Teodorescu, *Mechanical Systems, Classical Models: Volume I: Particle Mechanics*. Springer, 2007.
- [78] M. H. AlSharif, A. Douik, M. Ahmed, T. Y. Al-Naffouri, and B. Hassibi, “Manifold optimization for high-accuracy spatial location estimation using ultrasound waves,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 5078–5093, 2021.
- [79] J. E. Dennis Jr and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.
- [80] C.-T. Chen, *Linear system theory and design*. Saunders college publishing, 1984.

- [81] K. Emami, T. Fernando, H. H.-C. Iu, H. Trinh, and K. P. Wong, “Particle filter approach to dynamic state estimation of generators in power systems,” *IEEE transactions on power systems*, vol. 30, no. 5, pp. 2665–2675, 2014.
- [82] John D’Errico, “nearestspd,” MATLAB Central File Exchange, 10 2023, <https://www.mathworks.com/matlabcentral/fileexchange/42885-nearestspd>.
- [83] M. L. TDK InvenSense, “Mpu9250specificationdatasheet, ps-mpu-9250a-01,” TDK InvenSense, 6 2016, rev. 1.1 <https://www.digikey.jp/htmldatasheets/production/1431889/0/0/1/mpu-9250.html>.
- [84] MIT’s Project Oxygen, “The cricket indoor location system,” MIT’s Project Oxygen, 2006, rev. 2.3.2, <http://cricket.csail.mit.edu/>.
- [85] M. robotics, “Startersetsuper-mp-3d,” Marvelmind robotics, rev. 1.1 <https://marvelmind.com/product/starter-set-super-mp-3d/>.
- [86] QORVO, “Dw1000 ieee802.15.4 -2011 uwb transceiver,” Qorvo, 12 2017, rev. 2.22 <https://www.qorvo.com/products/p/DW1000>.
- [87] P. Papcun and J. Jadloviský, “Optimizing industry robot for maximum speed with high accuracy,” *Procedia Engineering*, vol. 48, pp. 533–542, 2012, modelling of Mechanical and Mechatronics Systems. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877705812046139>

VITAE

- Name: Hammam Khaled Mohammed Salem
- Nationality: Yemen
- Date of Birth: 3/8/1992
- Email: *hmmam1413@gmail.com*
- Permanent Address: Qassim, Al Rass, postal address: 58613
- Bsc. in Electrical Engineering from Qassim University
- Secondary Education from King Saud secondary school in Al Rass
- Publications and researches
 - Recent Advances in Data-driven Integrated Sensing and Communication Methods - A Review, submitted.
 - 3D Position and Attitude Tracking using $SO(3)$ Manifold, under preparation.

- Robust 3D Indoor Positioning Algorithm in the Presence of non-Line-of-Sight Signals, under preparation.
- Honors and Awards
 - First class honors in Master's degree
 - First class honors in Bachelor's degree
 - Academic excellence award from Qassim University
 - Academic excellence award from the Department of Education in Al-Rass