中图分类号： 单位代号：10280

密　　级： 学　　号：

上海大学

博士学位论文

SHANGHAI UNIVERSITY DOCTORAL DISSERTATION

题目	基于特征的自适应滤波SLAM算法设计及在轮式移动机器人中的应用

作　　者＿＿＿＿＿＿＿＿

学科专业＿＿＿＿＿＿＿＿

导　　师＿＿＿＿＿＿＿＿

完成日期＿＿＿＿＿＿＿＿

姓　　名：　　　　　　　　　　　　　　　　　　　　　　　　　　学号：

论文题目：

上海大学

本论文经答辩委员会全体委员审查，确认符合上海大学博士学位论文质量要求。

答辩委员会签名：

主任：

委员：

导　　师：

答辩日期：

姓　　名：　　　　　　　　　　　　　　　　　　　　学号：

论文题目：

原 创 性 声 明

本人声明：所呈交的论文是本人在导师指导下进行的研究工作。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已发表或撰写过的研究成果。参与同一工作的其他同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签　名：____________日 期：__________

本论文使用授权说明

本人完全了解上海大学有关保留、使用学位论文的规定，即：学校有权保留论文及送交论文复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容。

（保密的论文在解密后应遵守此规定）

签 名：___________导师签名：___________日期：___________

上海大学工学博士学位论文

基于特征的自适应滤波 SLAM 算法设计及在轮式移动机器人中的应用

姓　　名：
导　　师：
学科专业：

上海大学机电工程与自动化学院

2020 年 11 月

A Dissertation Submitted to Shanghai University for the

Degree of Doctor in Engineering

Designing Adaptive Filtering for Feature-Based SLAM Algorithm of Wheeled Mobile Robot

MA Candidate ： Heru Suwoyo

Supervisor ： Yingzhong Tian

Major ： Mechatronic Engineering

School of Mechatronic Engineering and Automation,

Shanghai University

November, 2020

摘 要

移动机器人在未知环境下的同步定位和地图构建(Simultaneous Localization and Mapping, SLAM)同时包含定位和建立地图两个问题，被认为是机器人实现完全自主运动的关键技术之一，对机器人的导航、控制、任务规划等领域有着重要意义。用数学模型描述来描述 SLAM 问题，常用的模型分为基于滤波器与基于优化问题两种模型。本文建立的 SLAM 模型是基于滤波器的 SLAM 系统，因此 SLAM 问题可以描述为：估计机器人与空间中路标位置的均值和协方差。

由于传感器和惯性测量设备存在误差，得到的轨迹与地图通常和真实情况存在着一定的误差，如何减少噪声带来的影响，建立准确、一致的地图，是 SLAM 研究者们关注的主要问题。

传统的滤波器无法对系统过程中的噪声进行估计和处理，噪声误差会随着机器人的运动不断累积，最终造成系统的失效。本文的主要工作在于使扩展卡尔曼滤波(Extended Kalman Filter, EKF)和平滑变结构滤波(Smooth Variable Structure Filter, SVSF)能够估计系统过程和测量过程的随机噪声并统计及其相应的协方差。本文的主要研究工作：

(1) 在滤波器中加入自适应机制，即自适应滤波。自适应滤波器的主要设计目的是考虑到滤波过程中的参数变化或噪声统计来调整滤波器增益。自适应滤波器方法使得常规滤波器具有递归估计噪声统计量及其对应协方差的能力。采用离线计算参数来调节自适应参数的批量参数估计自适应滤波方法，将提出的方法称为自适应扩展卡尔曼滤波器(Adaptive Extended Kalman Filter, AEKF)和自适应平滑变结构滤波器(Adaptive Smooth Variable Structure Filter, ASVSF)

(2) 分别参考最大后验指数和加权指数原理，推导了传统的 EKF 和 SVSF。由于原始形式没有估计值，采用一步平滑法对 EKF 和 SVSF 进行了改进。使用平滑参数对公式进行离线推导，然后计算出所有未知参数的次优值。但是，由于存在多步平滑项，自适应过程需要简化，简化处理可能会导致系统过程和测量协方差的非正定矩阵的不准确，因此还涉及到抑制发散的处理方法，使

得自适应滤波器可以发散抑制。

(3) EKF 和 SVSF 的采用相同的自适应策略更新协方差。由于 EKF 或 SVSF 的原始公式中没有估计值，无法运用数学公式推导，因此为了弥补这个估计值的误差，在一步平滑的基础上对 EKF 和 SVSF 进行了改进。为了防止协方差的复杂度过高而引起发散，还引入了无偏估计和创新协方差估计。因此，改进后的滤波器可以在时间积分下递归的更新噪声统计。

(4) 将设计的 SLAM 算法应用于轮式机器人进行仿真实验。设计了基于速度的运动模型和基于直接测量的测量模型，采用 Turtlebot2 机器人的真实参数进行仿真实验。分别将传统的 EKF-SLAM, AEKF-SLAM, SVSF-SLAM, ASVSF-SLAM 算法估计的轨迹和特征地图与参考轨迹和特征地图进行对比，本文提出的方法在精度和全局一致性方面优于传统的算法。

(5) 本文使用 The Victoria Park dataset 数据集进行实验验证，该数据集通常用于验证基于二维在线特征的 SLAM 算法的鲁棒性和有效性。该数据集是澳大利亚野战机器人中心在 2009 年使用 Nebot 机器人在维多利亚公园采集的，该数据集运行时间为 26 分钟，路径长度达到了 4 公里，数据场景的区域面积 18321 平方米。所有实验都在搭载主频为 2.3GHz，Intel Core i5 双核处理器，内存频率为 2133MHz 的 8G 内存的电脑进行测试。从均方根误差和归一化误差平方两个方面分析比较了所有算法的精度和一致性。比较表明，本文提出的 ASVSF-SLAM 算法优于传统算法。

关键词：同时定位和地图构建(SLAM)，特征，激光雷达，极大似然估计(MLE)，最大后验估计(MAP)，自适应卡尔曼滤波 SLAM 算法，自适应平滑可变结构滤波 SLAM 算法

ABSTRACT

Simultaneous Localization and Mapping (SLAM) is a relatively widespread problem that needs to be solved to make a robot fully autonomous. Given the noisy measurement and process, the system architecture should find the accurate position of the robot and construct the map concurrently. Determining the inaccurate position of the robot can makes an improper construction of the map and vice versa. Linguistically, the main objective of addressing the SLAM problem lies in estimating the mean of the robot pose and feature-based map and their covariances. According to this brief description, it is not surprising that the use of the Extended Kalman Filter (EKF) and Smooth Variable Structure Filter (SVSF) have been significantly solving the problem of Simultaneous Localization and Mapping (SLAM).

The implementation requires an accurate system model and known prior knowledge. However, the theoretical perspective illustrates no precise system model due to some considerations, such as avoidance of physic laws. Besides that, the prior knowledge of noise statistics is usually unknown or partially known in the real application. Therefore, by manually defining them as the conventional and common way, both the performance of the SVSF and EKF possess a high risk of degradation. The inaccuracy of the modeling system might enlarge the estimation error. The uncertainty caused by the unpredictable and random error might affect the characteristic statistical change, which undoubtedly leads to the filter divergence condition.

Hence, the traditional form of both filtering strategies should be initially enhanced. The significant contribution of this research is to equip EKF and SVSF with an ability to estimate the noise statistic of the process and measurement and its corresponding covariances. This strategy is well-known as an adaptive filtering method based on a batch estimation of parameters. It is a popular method to tune the gain by offline-calculating the unknown parameter. Henceforth, they are termed as Adaptive Extended Kalman Filter and Adaptive Smooth Variable Structure Filter (AEKF and ASVSF). As

an effort to accomplish these goals, in the first case, the conventional EKF and SVSF are respectively derived by referring to the principle of maximum a posterior and weighted exponent. Due to the absence of estimated values from the original form, the EKF and SVSF are modified based on the strategy of a one-step smoothing method. This process allows the system to have the smoothed parameter that can be utilized to proceed with the offline-derivation process. Afterward, the suboptimal values of all unknown parameters can be calculated. However, due to the presence of a multistep smoothing term, the adaptive process needs to be simplified. Therefore, the inaccuracy might occur because of a non-positive definite matrix to the covariances corresponding to either process or measurement. For this reason, the use of the divergence suppression method is also involved. Besides that, both the suboptimal estimate values are also estimated using the unbiased estimation method. In the second case, the conventional EKF and SVSF are assisted by a different approach to the previous one, named Maximum A Likelihood Estimator and Expectation-Maximization.

In this design, the adaptive forms are generated by assuming that the updated covariance form of EKF and SVSF are the same. However, the mathematical derivation chokes temporarily due to the presence of estimated values which is unavailable from the original formulation of the EKF or SVSF. Therefore, aiming to cover this lack of estimate values, the EKF and SVSF are modified based on a one-step smoothing method. Furthermore, to prevent the divergence caused by covariances' complexity, the unbiased estimation and innovation covariance estimation is involved. Hence, the proposed methods can recursively update the noise statistic under time integration. All the adjusted parameters based on the previous calculation make the filtering learn and improve without changing their characteristics. Furthermore, the proposed methods are applied to solve the SLAM problem of a wheeled mobile robot. Henceforth, both are named as AEKF-SLAM and ASVSF-SLAM algorithm.

The proposed method's verification and validation are conducted in two different cases, the synthetic-based simulation and real-experiment. The synthetic-based simulation considers that the robot moves from the initial position to the goal position by executing wheel rotation ticks per second. The user gives these values, but they are assumed to be always followed by the small additive noise. Sequentially, it measures all distinguishable features by presenting the range and bearing values to the system. Similarly, the measurement values are considered to be noisy. Therefore, the synthetic-based simulation assumes the reference path when the robot is moved using a motion model without any perturbation. The map is supposed to be known by placing some features around the robot path. The motion model and measurement model are designed by adopting the differential steering system and direct point-based observation, respectively. Meanwhile, the Victoria Park dataset recorded by Nebot, 2009, at the Australian Centre for Field Robotics is used for the real-experiment. This popular dataset is commonly used to verify the adaption or invention to a 2D online feature-based SLAM algorithm. A path through an area of around 197m x 93m is described in this dataset. This sequence consists of 7247 frames, captured over a total period of 26 minutes, along a 4 km trajectory. The data set includes steering and rear-axis wheel sensor readings (odometry) and laser range finder readings (one scan of 360 degrees per second) along with GPS data. A tree detector feature is given for the laser range data along with the dataset. Invariably, they have a wide distance to each other and can be isolated or classified with standard data association techniques. However, spurious data is found in some instances and must be deleted. All of the tests are conducted on a 2.3 GHz Dual-Core Intel Core i5, 8 GB 2133 MHz LPDDR3. The purpose of this experiment is to evaluate our approach's consistency and to examine the computational complexity. According to these simulations, all algorithms' accuracy and consistency are analyzed/compared in terms of average RMSE and NEES under the Monte Carlo Simulation. The comparison shows that the proposed method, ASVSF-SLAM algorithm, is better than the conventional method.

Keywords: Simultaneous Localization and Mapping, Feature, Laser Scanner, MLE, MAP, Adaptive EKF-SLAM, Adaptive SVSF-SLAM.

Table of Contents

Title 1
摘　要 1
ABSTRACT vi
Table of Contents xi
Chapter I Introduction 1
1.1 Background 1
1.2 Problem Statement and Hypothesis 4
1.3 Related Works 6
1.4 Research Contribution and Novelty 8
1.5 Organization of The Dissertation 11
Chapter II Literature Review on Gaussian State Estimation and Feature-Based SLAM 13
2.1 Introduction 13
2.2 State Estimation for Solving Stochastic Dynamic System 14
2.2.1 State Estimation from the side of Bayesian Paradigms 15
2.2.2 State Estimation from the side of Gaussian Distribution of The State 16
2.3 Kalman Filter 18
2.4 Extended Kalman Filter 20
2.5 Variable Structure Filter 22
2.6 Extended Variable Structure Filter 26
2.7 Smooth Variable Structure Filter 28
2.8 The Feature-Based SLAM 31
2.8.1 Motion Model 32
2.8.2 Measurement Model, Feature Extraction and Data Association 33
2.8.3 Localization 36
2.8.4 Simultaneous Localization and Mapping 36
Chapter III Adaptively Determining The Recursive Formulation of Noise Statistic for The Conventional Filtering Method 40
3.1 Designing Adaptive Extended Kalman Filter Using Principle of Maximum Likelihood Estimator and Expectation-Maximization 42
3.2 Designing Adaptive Extended Kalman Filter Based on Maximum A Posteriori Estimation and Weighted Exponent 54
3.3 Designing Adaptive Smooth Variable Structure Filter Based on Maximum A Posterior Estimation and Weighted Exponent 66
3.4 Designing Adaptive Smooth Variable Structure Filter Based on Maximum Likelihood Estimation and Expectation-Maximization Principle 77
Chapter IV Applying Proposed Method for 2D Feature-Based SLAM Algorithm 88
4.1 Simulation-Based Experiment 88

4.1.1 Robot Configuration and Motion Model 89
4.1.2 Measurement Model 93
4.2 Experiment Based on The Real-Time Application 96
4.2.1 Motion Model based on Linear and Angular Velocity 96
4.2.2 Measurement Model for Second Experiment 98
4.2.3 Data Association 99
Chapter V Experiment, Result, and Discussion 102
5.1 Realistic-Simulation of Feature-Based SLAM 102
5.2 Verification using the Victoria Dataset 113
5.3 Consistency Validation of The Proposed Algorithm 117
5.4 Verification Based on The Simulation with Fixed Velocity and Known Landmark in the Global Map 123
Chapter VI Conclusion 128
6.1 Summary 128
6.2 Future Research 129
6.3 List of Related Publication 129
Appendix 131
Bibliography 152
Acknowledgement 159

Chapter I Introduction

This chapter consists of the background, problem statement, hypothesis, related works, contribution and novelty, scope and limitation of study, content, and organization of the dissertation.

1.1 Background

The mobile robot has been widely applied in many sectors[1]–[5]. Therefore, it is not surprising to hear that it has been attracting much attention from many researchers. In the case of autonomous applications, the mobile robot should be able to execute all the given tasks without any[6]–[11] human interference. In order to achieve this provision, the robot should be able to initially perform the fundamental task of robot navigation, which is localization[9], [10], [12], [12]–[17]. Localization is the problem of determining the pose of a robot relative to the given map of the environment and establishing the correspondence between the local robot and map system coordinate[18]. There exist many limitations affecting this task, such as the unstructured surface, avoidance of physic laws, sensor limitation, and other unpredictable factors causing the uncertainty[4], [19]–[23]. Consequently, the accuracy of navigation might be decreased. Thus, the robot should first improve the information related to its pose, which can be done by gathering all the vital information. It can be established by utilizing the exteroceptive sensor, such as a camera, laser scanner, and sonar[15], [24]. Continuously, all the perceived information about its surrounding are used for building the global environment map that can be further utilized for localizing itself in the environment[12], [18], [25]–[27]. In the localization, the robot usually utilizes the proprioceptive sensors such as wheel encoder, gyroscope-based orientation sensor, and accelerometer[6], [28]–[31]. The precise and accuracy will be dependent on the location of the current robot. Now, these tasks sound complicated. The robot should collect the coordinate of all the features as well as knowing its pose[16],

[18], [32]–[34]. Nevertheless, it has been famously noticed as the popular problem of a mobile robot named as Simultaneous Localization and Mapping (SLAM) problem[1], [4], [35], [5], [8]–[10], [13], [14], [17], [19]. The SLAM framework based on the stochastic map approach was firstly introduced by Smith and Cheesman in 1986[7], [34], [36].

As an effort to acquire a reliable map, the robot should be having an accurate estimation of its state. Meanwhile, the accurate estimation state can only be satisfied when an accurate estimation map is available. Therefore, the perspective of probabilistic and calculus has been considered as a core of the estimation-based SLAM strategies[2], [37]–[45]. Among all the various approaches, Extended Kalman Filter (EKF) is the most popular and widely used[9], [22], [28], [32], [46]–[51]. There also existed many similar recursive filtering methods used for SLAM, that includes the relatively newest form of filtering, named as Smooth Variable Structure Filter (SVSF)[1], [4], [22], [39], [43], [44], [52]–[55]. As the highlight in SLAM for large-scale application, the computation complexity is the main issue. This complexity is remarkably growing up under the time integration while the new landmark is detected and added to the state vector[10], [12], [18], [32], [37]. Consequently, the number of both the state vector and their corresponding covariances will also be grown[4], [8]. Furthermore, to satisfy the effectiveness and accuracy, the mentioned estimators require the precise and accurate system model and known noise statistic[20], [29], [45], [48], [56]–[66]. However, there almost no exists an accurate model. It is due to the system's uncertainty are characteristically unpredictable and unobservable. Additionally, the prior knowledge of noise statistics is only predictable and suffers from the accurateness. Because of these reasons, there have been existing some approaches. It adopts the principle of batch estimation, which equipping the traditional form with the time-varying noise statistic. Similarly, this research proposes the adaptive filtering used for the robust estimator, SVSF. It aims to replace the role of EKF for the SLAM algorithm. Unlike the tradition, the SVSF is completed with the recursive noise statistic of the process and measurement as well as their corresponding covariances. In which it gives a novelty and innovation among the existing adaptive filter. As a way to present significant validation, it is also compared with an Adaptive EKF. Thus, adaptive EKF is also designed using the same strategy. It is noted that, this strategy aims to improve EKF and SVSF to precisely predict and update the state vector and its corresponding covariance. In this case, all the small additive noise will characteristically be recursive and responsive, referring to the current system.

The process of reaching an adaptive filter from the traditional form required some approaches. Initially, both EKF and SVSF were modified by using the principle of one-step smoothing point[22], [41], [46], [58], [61], [63], [67]–[69]. It aims to provide the multiple-step of estimate values, originally ungiven by the traditional forms, that might occur while the mathematical derivation is being stated. Consequentially, the involvement of different estimator such as maximum a posterior[4], [56]–[58], [65], [69], maximum likelihood estimation[8], [26], [59]–[61], [70], [71], expectation-maximization[8], [60], [64], [70]–[72], and weighted exponent[58], [69]. Besides that, the different methods were also attached to remove the possible effect that might be causing the divergence. The first method is the divergence suppression method[4], [65], [73] used to reupdate the predicted covariance of the state vector. And the second one is the innovation covariance estimation[8], [20], [21], [56], [59], [73], [74] used to prevent the existence of non-positive definite matrices of the process and measurement covariance. As the predecessor of these additions, the unbiased estimation[8], [58], [75] was also completed to the estimation process. It aims to generate more robust estimate values of the process and measurement noises and their corresponding covariances. The process of achieving Adaptive EKF and SVSF is separately introduced. To validate the effectiveness, they are directly implemented to solve the online SLAM problem of a mobile robot. Henceforth, they are termed as AEKF-SLAM and ASVSF-SLAM algorithm[4], [8]. They are realistically simulated, referring to the feature-based mapping and landmark-based localization processes. As an effort to analyze their effects, both of them are compared to the conventional strategies in the form of Root Mean Square Error (RMSE)[1], [30], [44], [45], [76] of Estimated Path Coordinate (EPC) and Estimated Map Coordinate (EMC). According to these comparative results, the presence of adaptive strategy has been significantly improving the traditional method of EKF and SVSF.

1.2 Problem Statement and Hypothesis

By solving Simultaneous Localization and Mapping, the robot might have the ability to fully and autonomously accomplish the navigation task without human interference. This process requires the robot to locate the current position and build the map based on all the acquired information of the sensor data in each iteration of movement. Besides performing two different tasks simultaneously, the difficulty of solving this problem lies in the existence of the unavoidable noise that always follows the measurement. Furthermore, the irregular surface where the robot is operated might also affect the system's stability. Thus, the filtering method is commonly used against these causes of uncertainty by conducting the estimation process. Using filtering for SLAM is intended to approximate the robot coordinate and acquire the coordinate of the features on the specific environment, given the control command and real measurement. After that, these types of coordinates are then gathered as the representation of the robot trajectory and feature-based map in the global map representation. Shortly, the main tasks of solving SLAM problems can be divided into two parts: localization and Mapping, then combined with being operated at the same time.

Localization

The problem of self-locating the pose of the robot along its navigation. Commonly, in order to perform this task, the system learn the knowledge of the environment, the control command, the observation and command history, initial state, and the motion process at the environment. Mathematically it can be described by using the following probability distribution.

$$p(x_k|z_{1:k}, u_{1:k}, m) \tag{1.1}$$

where k is the time discrete index, x_k is the robot state vector, $z_{1:k}$ is the measurement data given by the exteroceptive sensor, laser scanner, and $u_{1:k}$ is the control command sent to the robot and m represents the acquired information of all the tracked position.

Mapping

The mapping problem lies on the way to determine the information representing all the pose of the features spread on the environment. These tasks can be done by approximating all the coordinate of the features based on the previous robot position and the sensor reading, laser scanner. Analytically it can be represented by the following probability distribution.

$$p(m|x_{1:k}, z_{1:k}) \tag{1.2}$$

where m_k refers to the information of the position of the feature in the environment.

Simultaneous Localization and Mapping

The SLAM is the task of determining the consistent map of all the features available on the environment and simultaneously conducting the localization. It can be done by constructing the map and updating the robot's location to the global map representation based on the previous data of measurement and control command. The following probability distribution can represent this analogy.

$$p(x_k, m|z_{1:k}, u_{1:k}) \tag{1.3}$$

Simultaneous Localization and Mapping is challenging when the information of the previous robot position and the data of the measurement are noisy.

Adaptive Filtering Problem

Adaptive filtering is the main crucial problem that needs to be concerned before utilizing any conventional filtering method, such as the Extended Kalman Filter and Smooth Variable Structure Filter. The main objective of this problem is to determine how to obtain the noise statistic of the process and measurement recursively. Many researchers have stated that the optimality of any filtering method is significantly affected by the noise statistic. The mathematical derivation and estimation process are required which aims to generate time-varying noise statistic and their corresponding covariances. There are some recommended approaches before using EKF or SVSF, such as Multiple Model Estimation and Gain-Tuning Adaptive Filter. The urgency of this problem is to optimize the ability of filtering method used for solving 2D feature-based Simultaneous Localization and Mapping.

1.3 Related Works

As an effort to improve the filtering method, the batch estimation of parameter has been strongly recommended and traditionally conducted before applying the optimal filtering method. A. H. Mohamed and K. P. Schwarz Department introduced innovation-based adaptive Kalman Filter which is obtained by adopting the maximum a likelihood estimation to estimate the covariances corresponding to the process and measurement noise via the available information in the filter innovation sequence[59]. It is continuously improved with the presences of the fuzzy-innovation based adaptive filter, in which the measurement covariance matrix is regarded as adjustable parameter and can be tuned using fuzzy logic controller. This method is introduced by R Woo et. al [73]. Different from these methods, W Gao et. al. proposed the adaptive Kalman Filter by adopting the principle of Maximum A Posterior and smoothing it[69].

Besides that, the batch estimation of parameter is also used for nonlinear version of Kalman Filter. Y Huang et. al. proposed the adaptive Extended Kalman Filter by estimating the noise covariance matrices based on online expectation-maximization approach[72]. Similarly, S Akhlaghi et. al. proposed adaptive extended Kalman Filter by adopting the residual and innovation-based strategies. It aims to estimate the absence of recursive covariance noise statistic of the process and measurement from the conventional EKF[20]. Additional to that, K H Kim et. al. also proposed an adaptive two-stage extended Kalman filter by using the adaptive fading EKF. It is used to estimate the covariance matrices of the noise statistic of the process and measurement. As a note, that the fading EKF can be achieved by adopting the principle of innovation covariance estimation[77]. Once, it has been considered as effective way to improve EKF, it is continuously used as the base. K A Myers et. al. introduced an adaptive Kalman Filter based on the state and observation noise samples generated in the Kalman filter algorithm to estimate the first- and second order moment of the noise processes[78].

However, the use of EKF is limited nowadays because of some factors. Therefore, the adaptation of EKF has been famously presented such as Cubature Kalman Filter and Unscented Kalman Filter. As the traditional method, they are not originally completed with an ability to update the noise statistic. For this reason, their adaptive formulations are introduced by J He et. al. and D Chen et. al. proposed the hybrid adaptive filter of Unscented Kalman Filter (UKF) by combining the Maximum A Posterior and Maximum Likelihood Creation to estimate the noise covariance for the state and measurement[65]. F Yu et. al. proposed the adaptive form of Cubature Kalman Filter to estimate the noise statistic and their corresponding covariance matrices by referring to Sage-Husa noise statistic estimator[13], [14], [50], [79]. Y Shi et. al. proposed an adaptive unscented Kalman Filter to recursively estimate the system process noise variance using the modified Sage-Husa noise statistic Estimator[80]. Z Gao et. al. proposed Adaptively Random Weighted Cubature Kalman Filter by involving the random weighting theory to estimate system noise statistics and predicted state and measurement together with their associated covariance[81]. J H Wang et. al. introduced an adaptive UKF used for estimating the noise statistic and their corresponding covariance by using the principle of modified Sage-Husa noise statistic estimator[14]. B Gao et. al. School proposed an adaptive UKF for Nonlinear State Estimation via Maximum Likelihood Principle to estimate the covariance matrix of the process noise and measurement noise adaptively. The approach behind this achievement is estimating the noise statistic via the available new information in the filter innovation sequence[61]. The used method is the same method used by Mohamed[59].

Furthermore, the application of filtering method is also implemented to solve the feature-based SLAM algorithm. It is intended and proposed to replace the role of conventional algorithm, EKF-SLAM algorithm. P Yuzhen et. al. used the adaptive EKF based on modified Sage-Husa noise statistic estimator to resolve the problem of the error accumulation in the process of mobile robot localization[82]. Besides that, J H Wang et. al. implemented an adaptive UKF for solving SLAM problem of Unmanned Underwater Vehicle[14] and F Yu et. al. utilized the adaptive Cubature Kalman Filter for solving SLAM problem of mobile robot[13]. As the alternative of optimal filtering, the robust filtering, SVSF, has been also applied for this type of SLAM problem. D Fethi et. al. utilized the New Form of Smooth Variable Structure Filter completed by Time-Varying Covariance Update and smoothing boundary layers for solving SLAM problem of Unmanned Vehicle[19]. D Fethi et. al. also utilized the Smooth Variable Structure Filter with a fixed smoothing boundary layer for solving SLAM problem of Unmanned Vehicle[2]. Y Liu et. al. used the Smooth Variable Structure Filter as the FastSLAM algorithm core used for Unmanned Air Vehicle[9]. Although, the SVSF has been significantly giving much improvement to the feature-based SLAM, its implementation does not concern to noise adaptation. For this reason, the adaptive filtering method is proposed to improve SVSF before utilizing it as the SLAM algorithm. In this dissertation, the batch estimation of the unknown parameter is applied based on MLE and MAP. However, there is a lack of estimate values that cannot be adopted from the original formulation of SVSF. Thus, as the innovation in this research, the SVSF is smoothed using a one-step smoothing point. Besides that, the estimation process is proceeded by involving the unbiased estimation to guarantee that the estimated noise statistics are unbiased from the actual estimate values. By using these strategies, the novelty of this research is emphasized.

1.4 Research Contribution and Novelty

The primary research lies in the filtering strategy area, indicated by the presence of SVSF development. Recently, SVSF has been considered as the robust filtering for state estimation since Gadsden introduced the complete form in 2011. Like the other traditional estimator, it can estimate the posterior state and covariance regarding the corresponding gain. It is very robust and stable compare to the existed filtering method for modeling uncertainties and errors. Nevertheless, the usage of SVSF requires initial predefinition to the characteristic of the prior noise statistic interfering with the process and measurement system. The absence of a time-varying variable in the traditional SVSF might degrade the robustness and stability of SVSF in case of a real application or even realistically simulation. Therefore, the main contribution of this research is to introduce a new form called adaptive SVSF. It is done by firstly estimating the original formulation with some different methods and strategies. The following flowchart can illustrate the process of generating an adaptive filter for SVSF.

Furthermore, all the concepts seen in Figure 1.1 were also adopted to achieve the adaptive form of traditional EKF. The main reason is to present a more comparative method to validate the robustness, stability, and accuracy of Adaptive SVSF. Since the product of this research can be broken down into the field of state and parameter estimation, so that, this methodology can be applied and implemented on the case of target tracking, simultaneous localization and mapping, fault detection and diagnosis problem. These contributions are significant to develop the existence of SVSF and EKF with all their capabilities. To summaries the methods above, the following statement is presented.

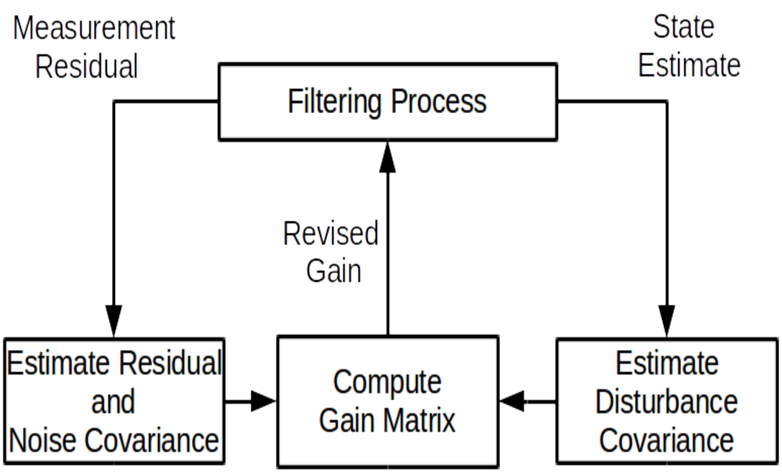


Figure 1.1 Adaptive Concept

Primary Contribution and Innovation

Instead of only using EKF and its adaptive formulation, this dissertation emphasizes the use of a robust filtering method that is adopted from the sliding mode concept, SVSF. According to the characteristic, the formulation of SVSF's gain and its updated covariance is different from EKF. Therefore, even involving the traditional method such as MLE or MAP, it still highlights the novelty in this dissertation. Concurrently, this dissertation presents an innovation that considers the complex formulation of covariance as the base for the whole estimation process.

Besides that, the derivation process is conducted to find the covariance matrices of noise statistic and its mean formulation in which it is different from the usual approach. Furthermore, unlike the traditional, SVSF is smoothed to provide the lack of multiple-step estimates values given by Maximum A Posterior or Maximum Likelihood Estimation derivation. Moreover, the unbiased estimation is also involved aiming to re-check and re-ensure that the suboptimal solution has the zero-bias. In the end, to prevent the occurrence of divergence conditions that might be caused by nonpositive definite matrices, the use of innovation covariance estimation is involved. It is used to replace the covariance of innovation error. It aims to improve the optimality of the proposed method more.

Secondary Contribution and Innovation

The former method commonly used as 2D feature-based algorithm is EKF. Therefore, by replacing it with SVSF, it gives innovation and novelty alternatively. Additionally, the use of SVSF is also enhanced by applying the adaptive filtering method, named as a batch estimation of the parameter. Thus, it gives a significant contribution to the new formulation of SVSF. The presence of adaptive SVSF is used to solve the online SLAM problem of a mobile robot with the feature-based mapping and landmark-based localization.

Scope and Limitation

The recursive estimation process for SVSF is conducted by assuming that there is no lack of prior measurement. The saturation function is on a steady-state condition when the estimated values are inside the boundary layer width. Furthermore, instead of using the simplified formulation of update covariance as introduced by Joseph, designing adaptive SVSF refers to the compound formulation. It aims to keep the accuracy of uncertainty about the state. The validation of the proposed method is taken by comparing it with the other filtering method to solve the feature-based SLAM problem. The simulation is conducted by initially assuming that this robot moves around the environment based on the control command sent by the user. When it moves, this robot acquires the position of all the features. This analogy is considered as a reference. Continuously, to provide a realistic simulation, the motion and measurement of the robot are perturbated with randomly generated small noise. Therefore, it is stated that the main role of the proposed method is to estimate the robot path and feature coordinate, simultaneously, given the inaccurate control command and noisy measurement. Accordingly, by knowing the truth and estimated values, the validation and comparison are conducted in terms of Root Mean Square Error.

1.5 Organization of The Dissertation

The rest part of this dissertation is organized as follows. Chapter II reviews the primary literature related to Gaussian State Estimation, including State Estimation for Solving the Stochastic Dynamic Problem, Kalman Filter, Extended Kalman Filter, Variable Structure Filter, and Its Extended Version, and Smooth Variable Structure Filter. This chapter also presents a Literature Review on Localization, Mapping, and Simultaneous Localization and Mapping Problem, including General Problem of Feature 2D SLAM, Motion Model, Measurement Model, Feature Extraction using Derivative of Recorded Scan, Landmark Measurement, Localization using EKF, Simultaneous Localization and Mapping using EKF. Chapter III presents the process of designing adaptive EKF and adaptive SVSF. The mathematical derivation behind this process is also presented. Moreover, the Adaptive EKF and SVSF are separately validated by implementing them to solve the online SLAM problem. In which, the discussion is presented in Chapter IV. The experiment, result, and discussion will be presented in Chapter V. Chapter VI presents the conclusion.

Chapter II Literature Review on Gaussian State Estimation and Feature-Based SLAM

The state estimation plays a significant role in solving the problem of Simultaneous Localization and Mapping. This chapter presents the popular filtering method for solving the SLAM-like dynamic system. It presents the Bayesian Framework, Kalman Filter, the most popular linearization-based filtering, the Extended Kalman Filter, and Smooth Variable Structure Filter basics. This chapter also presents the general definition of a feature-based 2D SLAM problem applied for a wheeled mobile robot. Some fundamental prerequisites commonly used to solve this problem are also presented sequentially. It includes the motion model, direct point-based observation, and inverse point-based observation.

2.1 Introduction

The primary role of estimation is to extract the actual state from the system's noisy measurement or observation and form a state estimate. This extracting aims to minimize the estimation error caused the uncertainty, which often interfered with the processing system and measurement. There almost no exact manner to against and remove this uncertainty except the probability-based approaches. Thus, the probability is considered the central core behind the estimation theory used for mathematically modeling the uncertainty[18].

The enormous contribution of estimation theory is presenting how to obtain an accurate state from the noisy data. The development of estimation strategies has involved a large number of contributors from different fields listed as follows. Girolamo Cardan (1501 - 1576) was considered the first contributor who presents probability's systematic treatment[83]. Moreover, after some development, Jacob Bernoulli (1654-1705) introduced the first rigorous proof of the Law of Large Numbers for repeated independent trials, which is now popularly known as Bernoulli Trials. Thomas Bayes (1702-1761) derived the famous rule for the statistical inference that provides the Bayesian Estimation method[44]. Carl Friedrich Gauss (1777-1855) introduced the optimal estimate from the noisy data, named as a method of least squares, in 1795[84]. Continued, Andrei Markov (1856–1922) developed a theory of random process, which usually now termed as Markov process as well as Markov Chain[85]. Furthermore, Andrei N. Kolmogorov (1903-1987) reestablished the foundation of probability theory on measure theory, which became the basis for integration theory and the mathematical basis of probability and random process[83].

2.2 State Estimation for Solving Stochastic Dynamic System

State Estimation is the task of extracting the state variables from the noisy measurement. The main objective of state estimation is to minimize the estimation error. The estimation error is the difference between the estimated values and the real values projected as the output on the particular space. The limitation of the measurement process and unpredictable noises cause the measurement to suffer from exactness. Thus, the framework needs to be stated to construct the state estimation of a stochastic dynamic system. And of the popular one might be adopted from the first-order Markov-Process that can be expressed as follows

$$\begin{cases} x_k = f\left(x_{k-1}, u_k, \omega_{k-1}\right) \\ z_k = h\left(x_k, v_k\right) \end{cases} \tag{2.2.1}$$

where k is discrete time index, x_k is the representation of the state vector and z_k is measurement model. Meanwhile, ω_{k-1} and v_k are the representation of small additive noises concerning the process and measurement, respectively. It is assumed that both additive noises are characteristically mutually independent and white. Moreover, the second assumption is both the transition model $f(\cdot)$ and measurement model $h(\cdot)$ are known as well as the control command u_k is given. The filtering problem is regarded to calculate the estimated values of the state vector recursively.

x_{k-1}. Fundamentally, two common methods can be used to achieve this formulation: by either adopting the Bayesian Paradigm or the Gaussian distribution.

2.2.1 State Estimation from the side of Bayesian Paradigms

By using the Bayesian paradigm, allows us to calculate the conditional a posterior state of the probability function x_k of given the set of measurements Z_k. Analytically, it can be modeled as $p(x_k|Z_k)$ where $Z_k = \{z_1, z_2, \ldots, z_k\}$. Continuously, it is initially assumed that the conditional probability of its prior state x_{k-1} given the prior measurement z_{k-1} is denoted as $p(x_{k-1}|z_{k-1})$, then the prediction and update step can be conducted. According to [44], the prediction step of Bayesian perception can be done by using the Chapman-Kolmogorov equation.

$$p(x_k|Z_{k-1}) = \int p(x_k|x_{k-1})p(x_k|Z_{k-1})dx_k \tag{2.2.1.1}$$

where the state transition modeled $p(x_k|x_{k-1})$ can be calculated by referring to Equation (2.2.1). Note that the initial state $p(x_0)$ is known as satisfying $p(x_0|z_0)$. Then by using the Bayesian Rule as the basis, the update step of this process can be analytically calculated as follows.

$$p(x_k|Z_k) = \frac{p(z_k|x_k)p(x_k|Z_{k-1})}{p(z_k|Z_{k-1})} \tag{2.2.1.2}$$

where $p(z_k|Z_{k-1})$ is the normalizing constant which can be calculated as follows

$$p(z_k|Z_{k-1}) = \int p(z_k|x_k)p(x_k|Z_{k-1})dx_{k-1} \tag{2.2.1.3}$$

Once the likelihood function $p(z_k|x_k)$ is obtained by referring to the Equation (2.2.1), then Equation (2.2.1.3) can be calculated. As a note, that above calculation are modeled by referring to the following assumptions

1. The state transition satisfies the first order Markov Process, i.e. $(x_k|X_k, Z_k) = p\left(x_k|x_{k-1}\right)$, where $X_k = \{x_0, x_1, \ldots, x_{k-1}\}$
2. The measurement is conditionally independent given the state, $p(z_k|X_k, Z_{k-1}) = p(z_k|x_k)$

The main purpose of the filtering is to construct the posterior Probability Density Function (PDF) accurately based on all the available information. Equation (2.2.1.1) – (2.2.1.3) present the base scheme of recursive state estimation. However, it cannot cover some scenarios because it only contains the conceptional solution. For this reason, the role of Gaussian PDF might be used.

2.2.2 State Estimation from the side of Gaussian Distribution of The State

The recursive equation of the estimated posterior stated can be analytically solved by taking consideration that the linear state transition and measurement model are subjected to addictive noises with Gaussian PDF[43], [44]. It is commonly used to simplify the complexity of the calculation of the Bayesian Paradigm. The assumption provides the normal distribution to both the prior state $p(x_k|Z_{k-1})$ and the likelihood $p(z_k|Z_{k-1})$, in which it will return Gaussian distribution for the posterior PDF $p(x_k|Z_k)$. Under this assumption, the Bayesian filter is reformulated to as Gaussian Filter by converting the recursive computation of the former Bayesian filter to algebraic computations of the first moment (mean) and the second moment (covariance) of the existing conditional, which both are followed by the time and measurement update as can be presented as follows

Time Update

This step generates the prior state estimate $x_{k|k-1}$ and the prior error state covariance $P_{k|k-1}$ by applying the expectation operator. This step can be analytically formulated as follows

$$\begin{aligned} x_{k|k-1} &= E\{f(x_{k-1}, u_{k-1})|Z_{k-1}\} \\ &= \int^{R^{nx}} f(x_{k-1}, u_{k-1}) N(x_k; \hat{x}_{k-1|k-1}, P_{k-1|k-1}) dx_k \end{aligned} \quad (2.2.2.1)$$

where $N(\ldots, \ldots)$ represents the Gaussian PDF.

Measurement Update

This step produces a posteriori state estimate and its updated covariances by assuming that the error can be approximated as Gaussian since the error in the prior measurement is a zero-mean white stochastic process[43], [44], [52], [85]. By this assumption, the likelihood density can be restated as the initial process of measurement update as shown below.

$$p(z_k|Z_{k-1}) = N(z_k; \hat{z}_{k|k-1}, P_{zz,k|k-1}) \tag{2.2.2.2}$$

where the prior measurement is stated as follows

$$\hat{z}_{k|k-1} = \int^{R^{nx}} h(x_k, u_k) x N(x_k; \hat{x}_{k|k-1}, P_{k|k-1}) dx_k, \tag{2.2.2.3}$$

Meanwhile, its covariance and cross-covariances are respectively described by the following equations.

$$P_{zz,k|k-1} = \int^{R^{nx}} h(x_k, u_k) h^T(x_k, u_k) x N(x_k; \hat{x}_{k|k-1}, P_{k|k-1}) dx_k - \hat{z}_{k|k-1} \hat{z}^T_{k|k-1} + R_k \tag{2.2.2.4}$$

$$P_{xz,k|k-1} = \int^{R^{nx}} x_k h^T(x_k, u_k) x N(x_k; \hat{x}_{k|k-1}, P_{k|k-1}) dx_k - x_{k|\hat{k}-1} \hat{z}^T_{k|k-1} \tag{2.2.2.5}$$

Then by using the new measurement z_k, the concept of the Gaussian filter leads the initial calculation of the posterior state and its covariances

$$p(x_k|Z_k) = N(x_k; \hat{x}_{k|k}, P_{k|k}) \tag{2.2.2.6}$$

and by respectively calculating the gain and error representation as shown below

$$K_k = P_{xz,k|k-1} P_{zz,k|k-1} \tag{2.2.2.7}$$

$$e_{z,k|k-1} = z_k - \hat{z}_{k|k-1} \tag{2.2.2.8}$$

Hence, both the posteriori updated state and its updated covariance can be respectively formulated as follows.

$$x_{k|k} = \hat{x}_{k|k-1} + K_k e_{z,k|k-1} \tag{2.2.2.9}$$

$$P_{k|k} = P_{k|k-1} K_k P_{zz,k|k-1} K_k^T \tag{2.2.2.10}$$

Up to this point, it can be simply declared that the formulation Gaussian Filter above can reduce the complexity of Kalman Filter in the case of both the linear state and measurement, which are subjected to an additive zero-mean white Gaussian noise. However, the main basis of the Gaussian filter is concentrated on how to calculate the Gaussian weighted integrals that are all formulated as nonlinear functions with Gaussian densities[30], [42]–[44], [46], [85]–[87]. Therefore, in some cases of a nonlinear system with non-Gaussian noise, a particular technique is required to solve the estimation problem. This technique leads to such linearization or approximation of probability density function, which is initially regarded to obtain the exact analytical solution due to this limitation. The most common Gaussian method for solving recursive nonlinear estimation problems through linearization is Extended Kalman Filter[35], [41], [46], [86], [88]–[90]

2.3 Kalman Filter

Kalman Filter (KF) was invented by Rudolph Emil Kalman in the 1950s[83], [91]. The significance of KF has been statistically proven to give the optimal solution to the linear system model, under the assumption of the noise is modeled as Gaussian. Generally, there are two versions of KF, which are continuous-time version and discrete-time domain. The continuous-time version was developed by Kalman and Bucy, known as Kalman-Bucy filter[44], [85]. However, the discrete-time estimation problems are the only ones concerned in this dissertation.

The KF requires the dynamic system model, known control input, and measurement followed by the noise. Based on these requirements, the KF can provide the optimal state estimates: by predicting the state referring to the initial state, calculating the covariance of the predicted state, obtaining the weighted average of the predicted variable and measured values, and presenting new state as well as its corresponding covariance for next iteration.

All these processes can be classified into two stages, which are termed as prediction and update. In the prediction step, the KF utilizes the previous state vector, control input, small additive noise assumed to interfere with the process, and their corresponding predetermined matrices. Meanwhile, in the update step, the KF utilizes and combines the predicted state or termed the prior state estimate and the current measurement to obtain the new estimate state commonly named as posterior state estimate [92].

Initially, to formulate KF, the form represented by Equation (2.1) should be linearized. It is assumed that the linearized models are respectively given as follows

$$\begin{cases} x_k = f\left(x_{k-1}, u_k, \omega_{k-1}\right) \\ z_k = h\left(x_k, v_k\right) \end{cases} \tag{2.3.1}$$

Then by referring to [83] the formulation of the recursive KF can be summarized. First, the prediction stage of KF can be summarized as follows

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1} \tag{2.3.2}$$

$$P_{k|k-1} = A\hat{x}_{k|k-1}A^T + Q_k \tag{2.3.3}$$

where the x represents the state vector, A and B are known matrices corresponding to the state and control command, respectively. Meanwhile, Q_k refers to the error covariance of the small additive noise of the process. Since, both the predicted state and measurement are computed, then the update stage of KF can be done as follows

Firstly, the innovation (error measurement) and its covariances are respectively calculated as follows

$$e_{z,k|k-1} = z_k - \hat{z}_{k|k-1} \tag{2.3.4}$$

$$S_{k|k-1} = CP_{k|k-1}C^T + R_k \tag{2.3.5}$$

where $z_{k|k-1} = Cx_{k|k-1} + v_k$ and R_k represents the error covariance corresponding to the small additive noise of the measurement. Once, Equation (2.3.4) and Equation (2.3.5) are calculated, the corresponding gain of KF can be computed as follows

$$K_k = P_{k|k-1}C^T S_{k|k-1}^{-1} \tag{2.3.6}$$

Finally, the updated state and its error covariance are determined by referring to the following equations, respectively.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k e_{z,k|k-1} \quad (2.3.7)$$

$$P_{k|k} = P_{k|k-1} K_k S_{k|k-1} K_k^T \quad (2.3.8)$$

Kalman Filter is known well as one of the optimal state estimation methods. It is also regarded as the popular Gaussian filtering used for solving the linear system, which is the successor of the Wiener-Kolmogorov filter (WF)[44], [85]. The main key of the optimal filtering method is minimizing the estimation error instead of designing the fixed filtering to generate the acceptable performance for a wide range of modeled uncertainties caused by the large dynamic. There have been existed many documented KF references with detail derivation[38], [85], [91], [93], [94]. The optimality of KF is strongly dependent on stability and robustness[85]. The KF assumes that the system model is known and linear, the system and measurement noises are white, and the states have initial conditions with known means and variances[10], [38], [44], [93]. However, these assumptions are always rare in the real application, resulting in suboptimal state estimates or even being unstable[95]. Besides that, the convergence condition is strongly dependent on the precise of computer and the complexity of matrix inversion[83].

2.4 Extended Kalman Filter

According to the description above, it is impossible to construct and obtain the analytical solution to such nonlinear state transition and non-Gaussian noise. It leads to the condition that the predicted distribution $p(x_k|Z_{k-1})$ will not be computed precisely. Accordingly, a particular approach is required to be involved in solving this kind of estimation problem. There are two popular approaches, namely, the approximation of PDF and linearization. By linearizing the nonlinear system, the similar stages existed on Kalman Filter can be alternatively used. The most popular linearization-based filtering method is the Extended Kalman Filter, which is the extension of the Kalman Filter.

The EKF approximates the nonlinear filtering problem by assuming that the distribution is Gaussian, and the direct numerical approximation in a descriptive sense is used to calculate a posteriori distribution. Thus, EKF is classified into the local approach-based nonlinear filtering instead of the global approach one. Unlike, KF the use of EKF is initially started by approximating the nonlinearity of the state or measurement model at the operating point[43], [44], [86], [96]. It is proceeded to updating both the state and its corresponding covariance by first calculating corrective gain respect the diversity between the actual measurement and predicted measurement. According to[2], [8], [44], [46], [47], [53], [72], [85], [89], [97], the process of EKF for both the prediction and correction stage are summarized as follows.

Prediction Stage

Like KF, the prediction stage of EKF assumes that the initial state value and the control command are available. The mean and covariance noise of the process and measurement are also predetermined. Then, the prediction stage of EKF starts by calculating the prior state vector by the following equation.

$$x_{k|k-1} = f(x_{k-1|k-1}, u_k) + \omega_{k-1} \tag{2.4.1}$$

Once, the predicted state estimate is computed, then its corresponding covariance can be sequentially calculated as follows

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_{k-1} \tag{2.4.2}$$

where F_k is the Jacobian matrix of the state transition function $f(\cdot)$. It is calculated by taking partial derivative of function $f(\cdot)$ with respect to the state at time $k-1$. Meanwhile, Q_{k-1} refers to the error covariance matrix of the small additive noise following the process. All variables found in this stage are used to conduct the second stage of EKF, which is the update stage.

Update Stage

The update stage is intended to produce the updated estimate values of the state vector and its corresponding covariance. It is done by concerning the corrective gain. In which, the innovation (error measurement) is firstly calculated before determining this gain value.

$$e_{z,k|k-1} = z_k - \hat{z}_{k|k-1} \tag{2.4.3}$$

where the predicted measurement is calculated by utilizing the predicted state estimated $x_{k|k-1}$ as follows

$$\hat{z}_{k|k-1} = h(x_{k|k-1}) + v_k \tag{2.4.4}$$

where $h(.)$ refers to the model of measurement function and v_k is the small additive noise of the measurement. Once, the innovation is calculated, then its covariance can also be computed. Mathematically it can be calculated as follows

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{2.4.5}$$

where R_k is the representation of the error covariance relative to the measurement noise. The calculation of obtaining the covariance of the innovation error utilize the predicted covariance of the state estimate. Hence, it can be declared that there is a connection between the prediction and update stage of EKF. Up to this point, the corrective gain of EKF can be calculated, which characteristically corrects the estimated value to minimize the innovation error computed above. Mathematically, it can be expressed as follows.

$$K_k = P_{k|k-1} H_k^T S^{-1} \tag{2.4.6}$$

Then, respectively the updated state estimate and its corresponding covariance can be computed below.

$$x_{k|k} = x_{k|k-1} + K_k e_{z,k|k-1} \tag{2.4.7}$$

$$P_{k|k} = P_{k|k-1}(I - K_k H_k) \tag{2.4.8}$$

Note that the Jacobian matrices of the state transition and measurement function are described

$$F_k = \frac{\partial f}{\partial x} |_{\hat{x}_{k-1|k-1}, u_{k-1}} \tag{2.4.9}$$

$$H_k = \frac{\partial h}{\partial x} |_{\hat{x}_{k|k-1}} \tag{2.4.10}$$

2.5 Variable Structure Filter

As discussed earlier that Kalman filtering-type methods critically requires both the accurate system model and white modeled noise. Unfortunately, they are rarely found in the real application due to some effects, such as the model structure, level and noise distribution, the availability of the initial condition, and the avoidance and reduction of the physical parameters. Therefore, to cover these limitations, the existence of a robust filtering method is introduced. Some methods can be classified into these types, such as robust Kalman, H-infinity, and Variable Structure Filtering-types[44], [45], [85], [86], [89]. And one of them is presented and discussed in this thesis, namely Smooth Variable Structure Filter, which is one of the successor filtering methods of VSF. The presence of SVSF was intended as an effort to improve the stability and robustness of KF. It was introduced by using the close principle of Variable Structure Filter.

As a leading introduction to SVSF, a brief discussion of the Variable Structure Filter is presented. VSF strongly refers to Variable Structure Control theory, which guarantees the stability given some bounded parametric uncertainty. And the Sliding Mode Control is the most popular form derived from the main principle of VSC. It can solve the estimation problem by utilizing a discontinuous switching plane along some desired trajectory. This plane is referred to as a sliding surface used to minimize the estimation error by keeping the state values along this surface. Although the VSF uses the discontinuous component for correcting the estimate as same as SMC, it has a different formulation. It similarly uses the principle of prediction and correction stage to KF. To perform the VSF, the VSF requires the known knowledge at the time k-1 and calculates the predicted state (prior state estimate) $\hat{x}_{k|k-1}$. Like Kalman Filtering-type, it obtains the updated state $\hat{x}_{k|k}$ by firstly utilizing the presence of system measurement. Similarly, to KF, using the linearized form of Equation (2.2.1), which is also expressed by Equation (2.3.1), the summary of the VSF form is presented as follows[30], [43], [44], [53], [85], [89], [90].

Prediction Stage

This stage is used to determine the prior state estimate by using all the available information including the transition matrix A, corresponding control matrix B, the initial state estimate $\hat{x}_{k-1|k-1}$, and the control command u_{k-1}

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1} \tag{2.5.1}$$

Since the predicted estimate is obtained, then the innovation (error measurement) can also be calculated as shown below.

$$e_{z,k|k-1} = z_k - \hat{z}_{k|k-1} \tag{2.5.2}$$

where the priori measurement $\hat{z}_{k|k-1}$ is computed using the following equation.

$$\hat{z}_{k|k-1} = C\hat{x}_{k|k-1} \tag{2.5.3}$$

Up to this point, there is no difference between the prediction stage of VSF and KF.

Update Stage

This stage is used to determine the updated state estimate by calculating the gain of VSF from all the obtained variables in the prediction stage and the following variables.

$$\begin{aligned} \widetilde{\xi} &= \xi - \hat{\xi} \\ \widetilde{\delta} &= \delta - \hat{\delta} \end{aligned} \tag{2.5.4}$$

where $\xi = CAC^{+}$ and $\hat{\xi} = \hat{C}\hat{A}\hat{\widetilde{\xi}}^{+}$. Meanwhile, $\delta = CB$ and $\widetilde{\delta} = \hat{C}\hat{B}$. Then, the gain of VSF can be calculated follows

$$\begin{aligned} K_k^{VSF} = \hat{A}^{-1}\hat{C}^{+}\Big(&||\hat{C}\hat{A}|_{abs}\{\gamma|\hat{C}|_{abs}|e_{z,k|k-1}|_{abs} + |\hat{A}^{-1}\hat{C}^{+}\widetilde{\xi}_{max}z_k|_a bs + \big[|\hat{C}^{+}|_{abs} + \\ &|\hat{A}^{-1}\hat{C}^{+}|_{abs}(\widetilde{\xi}_{max} + I)\big]V_{max} + |\hat{A}^{-1}\hat{C}^{+}\widetilde{\delta}_{max}u_{k-1}|_{abs} + \big[|\hat{A}^{-1}|_{abs} + |\hat{A}^{-1}\hat{C}^{+} \\ &\widetilde{C}_{max}|_{abs}\big]W_{max}\}\,|_{abs} \circ sign(e_{z,k|k-1})\Big) \end{aligned} \tag{2.5.5}$$

where $\circ$ represents the Schur product, $\widetilde{\cdot}$ is representing the error, max refers to the upper bound, I is the identity matrix, and $\widetilde{\xi}$, $\widetilde{C}$, and $\widetilde{\delta}$ represent the upper bound on modeling uncertainties of ξ, C, and δ, respectively[42], [43], [46], [85], [86]. Meanwhile, sign$(e_{z,k|k-1})$ is described as follows

$$\begin{bmatrix} e_{z_1,k|k-1} \\ \vdots \\ e_{z_m,k|k-1} \end{bmatrix} \tag{2.5.6}$$

Generally, Equation (2.5.6) satisfies the following function.

$$sign(e_{z,k|k-1}) = \begin{cases} +1 & e_{z,k|k-1} > 0 \\ 0 \quad if & e_{z,k|k-1} = 0 \\ -1 & e_{z,k|k-1} < 0 \end{cases} \tag{2.5.7}$$

Due to the complexity of gain calculation resulting in high-frequency switching, the performance of VSF is limited with the existence of chattering to the estimation states. As an effort to reduce the effect of this chattering, the presence of smoothing boundary layer ψ was introduced. It is intended to obtain the smooth function and ensure the robustness by maintaining the sign function when it is outside of this boundary. Analytically, the saturation function $sat(.)$ replacing the sign function can be expressed as follows

$$sign(e_{z,k|k-1}) = sat(\frac{e_{z_i,k|k-1}}{\psi_i}) \tag{2.5.8}$$

in which the saturation function is defined as

$$sat(\frac{e_{z_i,k|k-1}}{\psi_i}) = \begin{cases} +1 & \frac{e_{z_i,k|k-1}}{\psi_i} \geq 1 \\ \frac{e_{z_i,k|k-1}}{\psi_i} \quad if & -1 < \frac{e_{z_i,k|k-1}}{\psi_i} < 1 \\ -1 & \frac{e_{z_i,k|k-1}}{\psi_i} \leq -1 \end{cases} \tag{2.5.9}$$

large enough size to overcome the behavior of dynamic change. Thus, it leads to the existence of a relationship between the magnitude of VSF gain and the handling uncertainty level. Additionally, the smoothing boundary layer width should also be sufficiently large, so it can embrace the maximum reached values of the gain of VSF when conducting the estimation process. The smaller boundary layer width represents accuracy to the estimate with fewer uncertainties available. This analogy leads to the following function.

$$\psi = |\hat{C}^{+}\hat{A}^{-1}|_{abs}|\hat{A}\hat{C}|_{abs}\Big\{|\hat{A}^{-1}\hat{C}^{+}\widetilde{\xi}_{max}z_{max}|_{abs} + \Big[|\hat{C}^{+}|_{abs} + |\hat{A}^{-1}\hat{C}^{+}|_{abs}(\hat{\xi} + \widetilde{\xi}_{max} + I)\Big]V_{max} + |\hat{A}^{-1}\hat{C}^{+}\widetilde{\delta}_{max}u_{max}|_{abs} + \Big[|\hat{A}^{-1}|_{abs} + |\hat{A}^{-1}\hat{C}^{+}\widetilde{C}_{max}|_{abs}\Big] W_{max}\Big\} \tag{2.5.10}$$

According to Equation (2.5.10), it can be declared that the boundary layer width is a representative function associated with the presence of uncertainties in the estimation process. It is also apparent to say that the corrective gain of VSF can be calculated since both the upper bound and the level of noise are well-defined. And although the gain provides robustness and stability to the estimation strategy, VSF yields a non-optimal estimation result[30], [44], [76], [85], [87]. It also cannot be applied for such a nonlinear system. For this reason, the nonlinear form of VSF is required.

2.6 Extended Variable Structure Filter

As an effort to improve the capability of VSF for such a nonlinear system, an Extended Variable Structure Filter was introduced. Like KF and EKF, the estimation process of EVSF assumes that the nonlinear system and its measurements are respectively defined as follows

$$\begin{cases} x_k = f\left(x_{k-1}, u_k, \omega_{k-1}\right) \\ z_k = h\left(x_k, v_k\right) \end{cases} \tag{2.6.1}$$

Conceptually, there is no much difference to VSF that the form of EVSF is referred to as the predictor-corrector principle. Therefore, there are also different stages, the prediction and update stage.

Prediction Stage

The predicted state estimate $x_{k|k-1}$ can be obtained by utilizing the previous knowledge of the initial state $x_{k-1|k-1}$ and control command u_{k-1}. Furthermore, the small additive noise to the process ω_{k-1} and measurement v_k and their corresponding covariances should predetermined. Analytically, the prediction stage of EVSF can be expressed as follows

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}) \tag{2.6.2}$$

where $f(.)$ refers to the state transition function. Similar to EKF form, once the predicted state $\hat{x}_{k|k-1}$ is obtained, the error measurement can also be computed. Mathematically, it can be expressed as follows

$$e_{z,k|k-1} = z_k - \hat{z}_{k|k-1} \tag{2.6.3}$$

where the prior measurement is calculated using the following equation.

$$\hat{z}_{k|k-1} = h(\hat{x}_{k|k-1}) \tag{2.6.4}$$

Update Stage

Similar to the VSF, the core of update process lies on the corrective gain. It is calculated as follows

$$K_k^{EVSF} = F^+ H^+ |H_{max}|_{abs} |F_{max}|_{abs} \left\{ \gamma |H_{max}^+|_{abs} |e_{z,k|k-1}|_{abs} + |F_{max}^{-1}|_{abs} \left[|f(x_{k-1}, u_{k-1} - f(\hat{x}_{k-1} u_{k-1}))|_{abs} + W_{max} + |H_{max}^+|_{abs} V_{max} \right] \right\} \circ sign(e_{z,k|k-1}) \tag{2.6.5}$$

where F and H are the Jacobian Matrices as the representative linearized form of the process and measurement. They are respectively calculated as follows

$$F_k = \frac{\partial f}{\partial x} |_{\hat{x}_{k-1|k-1}, u_{k-1}} \tag{2.6.6}$$

$$H_k = \frac{\partial h}{\partial x} |_{\hat{x}_{k|k-1}} \tag{2.6.7}$$

Then by utilizing the corrective gain, Equation (2.6.5), the updated state of EVSF can be computed below.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k^{EVSF} \tag{2.6.8}$$

Although, conceptually, the gain of EVSF is similar to the gain of VSF. They also have the same disadvantages and advantages. The only thing that makes them different is that due to linearization, the EVSF can be applied for a nonlinear system. However, it is frequently conducted when calculating the correcting gain, in which the linearization sufficiently increases the numerical result.

2.7 Smooth Variable Structure Filter

A similar limitation exists on EKF can also be met on EVSF. Accordingly, the revised form was introduced in 2011 to improve their performance, namely Smooth Variable Structure Filter[85]. It is a relatively new predictor-corrector estimator that can be applied for both linear and nonlinear systems. The SVSF is formulated based on the Sliding Mode Concept[1], [22], [39], [45], [53], [55], which utilizes the switching gain to converge the estimates to within a boundary of the actual state values.

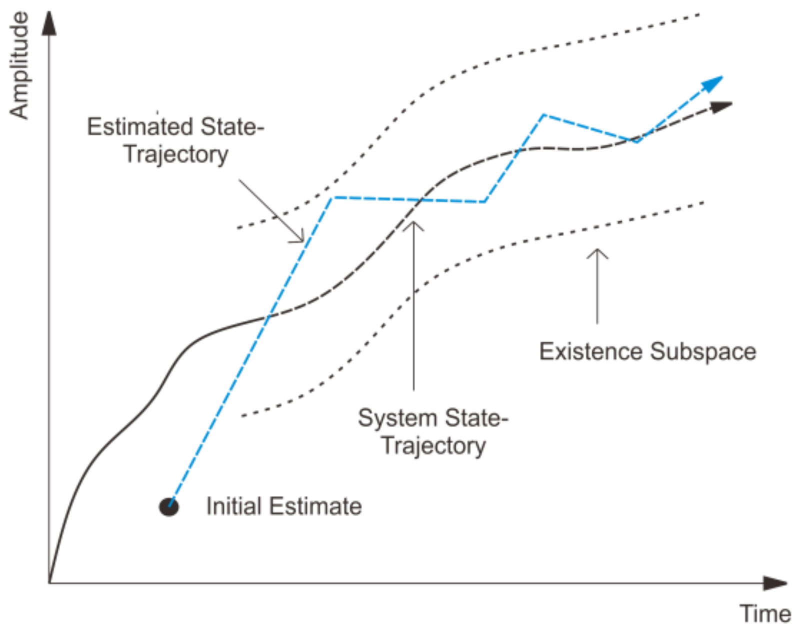


Figure 2.1 Smooth Variable Structure Filtering Concept

Assuming that the nonlinear system is modelled as follows

$$\begin{cases} x_k = f(x_{k-1}, u_k) + \omega_{k-1} \\ z_k = h(x_k) + v_k \end{cases} \tag{2.7.1}$$

According to [85], the SVSF can be summarized as follows.

Prediction Stage

This stage determines the prior state estimate given the state vector's initial information, the control command, and all the relevant variables representing the noise statistic and their covariances. The first step of this process can be seen as follows

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}) \tag{2.7.2}$$

where $f(\cdot)$ represents the state transition function. Once, the predicted state value $\hat{x}_{k|k-1}$ is calculated, the corresponding measurement can be obtained as follows

$$\hat{z}_{k|k-1} = h(\hat{x}_{k|k-1}) \tag{2.7.3}$$

Then, by calculating the difference between the actual/real measurement z_k and the predicted measurement $\hat{z}_{k|k-1}$, the innovation error $e_{z,k|k-1}$ is calculated as

$$e_{z,k|k-1} = z_k - \hat{z}_{k|k-1} \tag{2.7.4}$$

Update Stage

This process utilizes the corrective gain K_k^{SVSF} for calculating the updated or posteriori state estimate $\hat{x}_{k|k}$. The gain calculation is expressed as follows

$$K_k^{SVSF} = H^+\big(|e_{z,k|k-1}|_{abs} + \gamma|e_{z,k-1|k-1}|_{abs}\big) \circ sat\Big(\frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i}\Big) \tag{2.7.5}$$

where $\circ$ and $.^+$ represents the Schur product of matrix multiplication and pseudo inverse of matrix, respectively. Meanwhile, ψ refers to the smoothing boundary layer width and γ refers to the memory or convergence rate satisfying $0 < \psi_{ii} < 1$. According to Equation (2.7.5), the initial error measurement $e_{z,k-1|k-1}$ is required. Therefore, the posterior error measurement should be defined when the noise statistic and their corresponding covariance, the boundary layer width, and convergence rate are predetermined. This scenario leads to the process of SVSF requires to present the posterior error measurement for the next step of estimation. It can be concerned once the updated state estimate is obtained, where the posteriori state estimate is calculated by using the corrective gain as

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k^{SVSF} e_{z,k|k-1} \tag{2.7.6}$$

The corresponding measurement of the update state $\hat{x}_{k|k}$ is firstly calculated as

$$\hat{z}_{k|k} = h(\hat{x}_{k|k}) \tag{2.7.7}$$

then the posteriori error measurement is computed as follows

$$e_{z,k|k} = z_k - \hat{z}_{k|k} \tag{2.7.8}$$

Up to this point, the process of SVSF can be iteratively repeated. The stability and convergence of the existence subspace of SVSF can be evaluated by

$$|e_{z,k|k}|_{abs} < |e_{k-1|k-1}|_{abs} \tag{2.7.9}$$

where $|e|_{abs}$ represents the absolute values of the error measurement which can also be calculated as $|e|_{abs} = e.sign(e)$

The revised SVSF

The SVSF is a relatively new estimator. And all the previously discussed process is the first form SVSF, which is not completed with any ability to update the state vector's covariance recursively. Accordingly, Gadsden introduced the use of covariance to tune corrective gain in the presence of uncertainties. Besides that, the presence of time-varying boundary layer width was also introduced. These additions also give a correction to the corresponding gain K_k^{SVSF}. Referring to[43], [44], [53], once the predicted state estimate $\hat{x}_{k|k-1}$ is obtained, the prediction stage of SVSF is added with its corresponding covariance as

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q_{k-1} \tag{2.7.10}$$

where it can be calculated using the initially predefined covariance $P_{k-1|k-1}$. It is not that this initial covariance represents the uncertainty about the initial state $\hat{x}_{k-1|k-1}$. Meanwhile, Q_{k-1} refers to the covariance of additive noise to the process and F refers to the Jacobian matrix of state transition function $f(\cdot)$, which is calculated as follows

$$F = \frac{\partial f}{\partial x}|_{\hat{x}_{k-1|k-1}, u_{k-1}} \tag{2.7.11}$$

by utilizing the predicted covariance $P_{k|k-1}$ in Equation (2.7.10), the calculation of the covariance of innovation error can be conducted as follows

$$S_k = HP_{k|k-1}H^T + R_k \tag{2.7.12}$$

where R_k refers to the covariance matrix relative to small additive noise of the measurement. Meanwhile, H is the Jacobian matrix of measurement function calculated as follows

$$H = \frac{\partial h}{\partial x}|_{\hat{x}_{k|k-1}} \tag{2.7.13}$$

The covariance of error measurement S_k is then used to calculated the time-varying boundary layer width ψ. This process is analytically expressed by the equation below

$$\psi = \left(\overline{\overline{A}}^{-1} H P_{k|k-1} H^T S_k^{-1}\right)^{-1} \tag{2.7.14}$$

This boundary layer width is also used to design a saturation function, which is expressed as follows

$$sat\left(\frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i}\right) = \begin{cases} +1 & \frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i} \geq 1 \\ \frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i} \quad if & -1 < \frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i} < 1 \\ -1 & \frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i} \leq -1 \end{cases} \tag{2.7.15}$$

Meanwhile, A is defined as the function expressed below.

$$A = \left(|e_{z,k|k-1}|_{abs} + \gamma |e_{z,k-1|k-1}|_{abs}\right) \tag{2.7.16}$$

It is note that, γ refers to the convergence rate. Then the revised gain of SVSF can be calculated as follows

$$K_k^{SVSF} = H^{+}\left\{\overline{\overline{A}} \circ sat\left(\frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i}\right)\right\}\left[\overline{\overline{e_{z,k|k-1}}}\right]^{-1} \tag{2.7.17}$$

By using new gain, the updated state estimate is calculated using Equation (2.7.6). Accordingly, the update covariance of this state can be also calculated as

$$P_{k|k} = (I - K_k^{SVSF} H) P_{k|k-1} (I - K_k^{SVSF} H)^T + K_k^{SVSF} R_k K_k^{SVSF T} \tag{2.7.18}$$

Where $\overline{\overline{\cdot}}$ indicate the diagonal term of matrix.

2.8 The Feature-Based SLAM

Simultaneous Localization and Mapping is a relatively new problem of the mobile robot [2], [10], [26], [37]. It allows mobile robots to autonomously navigate in an environment without prior knowledge of the map without access to independent position information. As the name, the objective of the SLAM problem is to localize all the tracked pose of the robot and simultaneously to construct the map based on the observation through the exteroceptive sensor, such as laser scanner[15]. Although the 3D type of laser scanner is available nowadays, the 2D one is commonly used to solve the feature-based SLAM problem. By knowing the pose of the robot and landmarks, a mobile robot could quickly and safely navigate to reach the goal position from a specific position. It is also the reason that SLAM is considered as the essential concerning problem before performing the path planning or path tracking. It is due to the easiness of making a proper decision after solving it. However, once the SLAM requires the approach to know where the robot is and address how the environment looks, the problem becomes chicken-like. The robot requires an accurate map of its environment when it localizes the position. In order to determine the precise map, a mobile robot needs to know the location in the environment.

By this analogy, observation becomes a crucial part that needs to be carefully considered. As the common device interfered with some factors, such as dependence on the power supply, tolerance of the accuracy, and outer interference, thus the observation is followed by noise. The real-world sensor gives noisy values that make the measurement of either the robot pose and the environment subject to uncertainty and bias. Therefore, as the initial way to solve the SLAM problem, it is often characterized and modeled as the Gaussian distribution, which allows the user to jointly parameterize both the robot pose and environment using the multivariate-type of the Gaussian distributions. Accordingly, the main problem of SLAM problem can be solved by adopting the probabilistic manner to estimate the mean and covariance, which are mean as the expectation of both parameters and covariance represents the uncertainty due to the unpredictable and random noise following the system and measurement.

2.8.1 Motion Model

As mentioned above, the main objective of solving the SLAM problem lies in the estimation process against the uncertainty caused by the noisy measurement and system. Therefore, it is not surprising that the role of filtering strategy is often involved. Before implementing the SLAM-based filtering strategy, the essential consideration of the motion model is firstly concerned[4], [8]. It is approached to apply the state transition process relative to the presence of the control command. The model comprises the probability of feature state with uncertainty caused by the random noise following the input and the situation relative to the environment where the robot is operated. A motion model is an approach used to know the pose of the robot given the previous pose and the velocity command. Behind the motion model's probabilistic configuration, the kinematic configuration is commonly concerned as an easy way to model the mobile robot. Kinematic[31] is a strategy to observe the robot motion by taking off some consideration, such as the mass, force, and interference that might affect the robot. It is only assumed that there is no much factor of the cause of motion. However, to realistically used the kinematic for motion model, some elements lie on the move and turn the action of the robot are considered. It aims to satisfy the possible situation when the odometer gives noisy information. Mathematically, it can be expressed by the conditional probability $p\left(x_{R,k}|x_{R,k-1},u_k\right)$. Where k represents the discrete time index. Note that instead of represents the x-coordinate pose of the robot, the x_R refers to the state of the robot consisting both the spatial pose x, y and heading θ.

2.8.2 Measurement Model, Feature Extraction and Data Association

Besides of motion model, the second crucial part need to be involved before working SLAM-based filter is a measurement model[4], [5], [8]. Measurement is the observation process by utilizing the data generated using the sensor device relative to the physical frame. There are many types of sensors that can be used to collect data around the environment where the robot is operated, such as ranges sensor or camera as the visual. Like on the part of the motion model, there exist unavoidable and unpredictable noise cause the presence of uncertainty. Therefore, instead of using the deterministic model $z_k = f(x_k)$, the measurement is also represented by the conditional probability density function $p(z_k|x_k)$. In this dissertation, the type of sensor used as the primary measurement device is a laser scanner. It can be involved to sense the environment from the robot frame. There are two common noises classified to follow the range and bearing data. As the commonly generated input, the laser scanner gives some measurements depending on the ranges of each angle of measurement. Accordingly, it can be mathematically represented by the following expression

$$z_k = z_k^1, z_k^2, \ldots, z_k^N \tag{2.8.2.1}$$

Where *N* refers to the numbers of entire per measuring data using a laser scanner and z_k^1 refers to a single value of ranges on once measurement. Therefore, it can be known that if the laser scanner has capability to sense 180 ranges and the angle increment of the index to index, the maximum number of measurements is 180. Further, in order to approximate the single measurement of laser scanner, the conditional probability of $p(z_k|x_k, m)$ can be concerned as the product of individual measurement likelihoods. Mathematically it can be expressed as follows

$$p(z_k|x_k, m) = \prod_{n=1}^{N} p(x_k^n|x_k, m) \tag{2.8.2.2}$$

It is noted that, to approximate the single measurement, the state is firstly known. Meanwhile, m represents the map of the environment. It tells that the map of the environment should be specified before conducting the measurement. By definition, the map stores the information relative to the series of the sensed object of the environment. Accordingly, the mathematical representation of the object list of measurement can be expressed as follows

$$m = m_1, m_2, \ldots, m_M \tag{2.8.2.3}$$

Where M refers to the number of objects available on the environment. There are some ways to index the map, such as feature-based and location-based. Although they are essentially the same, feature-based is the most frequently used to represent the single object available on the environment. Instead of indexing with a specific location, the feature-based uses the value of m_M contains in Cartesian location. Although, the specific location is most frequently used in 2D SLAM algorithm perspective. Therefore, it is suppressing that many researchers prefer to represent $m_{x,y}$ instead of using m_M. It aims to firmly show that all the objects are located on world coordinate x, y. In this dissertation, the usage of sensor model is concerned as the feature-based, which involves the specific approach to extract the feature from single raw measurement. Referring to[18] and since the function of feature extractor is denoted by f, then the information of the features that is generated from the range measurement can be expressed as $f(z_k)$. The main challenges of extracting feature using this approach is to recognize the feature from large data of range measurement resulting only small number of features on environment.

The second fundamental component of feature-based mapping and SLAM systems is feature extraction. It seeks to leverage the characteristic of different object in the environment according the raw-sensed data. The low-level of feature extraction algorithm return the classification of the feature such as points, edges, center of curve segments and virtual corners.

Naturally, it can be achieved by using a geometrical feature identification approach. This method involves a local curvature scale. Due to the needless of the constructed scale space of map, this approach has been considered having an effectiveness and low-cost computation. The important of using feature extraction is to predict the measurement according the sensing data. Based on this observation, the extracted features are then proceeded into the step of data association in the SLAM systems. The key of data association is to manage the detected landmark in this observation. It is processed and should be matched with the landmark produced by prior observation. All the detected landmarks are then used as the part on the update process. The newly detected landmark is added into the state vector as the observed landmark. This process is a predecessor step used to find the correspondence to the existing landmark. In which it is an importation part before calculating the innovation error of measurement. Commonly, the data association that is conducted based on the diversity between the newly and observed landmark is Nearest-Neighbor approach.

2.8.3 Localization

Localization is the process to determine the robot position x_R relative to the world environment[16], [18]. It has been regarded as one of the crucial parts to make the robot autonomously navigate itself[27], [87], [98]–[100]. There are commonly two general types of localization, namely relative localization and absolute localization. These models are classified based on determining the position of the robot in the environment, whether by utilizing the exteroceptive sensor or not. Since the localization does not use any measurement data from external sensors, it can be classified as the relative localization. Contrary, since the system utilizes the information acquired by the external sensor about the environment, it is well-known as absolute localization. Since the localization process's objective is to estimate the position of the robot given the information of the previous position information, mathematically it can be expressed as follows

$$m = m_1, m_2, \ldots, m_M \tag{2.8.3.1}$$

It is clear to declare that since the previous state of the robot is determined from an external sensor, the robot pose in the environment can also be discovered. The most popular techniques used for this type of localization is a dead-reckoning. Recursively, the previous robot pose is used as a base in the next step. It is used to determine the new pose of the robot in the environment. Therefore, since the determination of the next location can be predicted using the control command and the information of the base data, the motion model algorithm mentioned above can be directly implemented.

2.8.4 Simultaneous Localization and Mapping

This section presents the big problem concerned as the manner to make the mobile robot to be truly autonomous. It is a termed problem that arises when there is no map given to the robot, and the robot's location relative to the environment is unknown, instead of the set of measurement $z_{1:k}$ and control command $u_{1:k}$. It is called Simultaneous Localization and Mapping (SLAM) or termed as Concurrent Mapping and Localization (CML). In SLAM, a mobile robot acquires the map of the environment and simultaneously localize its position relative to the global coordinate system. Solving the SLAM problem has been considered to have a more challenging task than only solving the localization problem. Since generating a map of the environment requires the precise coordinate position of the robot and obtaining the pose of the robot in the environment requires the known map, the SLAM problem has been regarded as the chicken-egg-like problem. As discussed earlier that, since the presence of the noise always unpredictably and time-invariant follows the system process and measurement, the uncertainty on the perspective of solving the SLAM problem is sufficiently hard. However, there has been an existing method to appropriately addressing this problem, which is approaching the strategy based on the statistical estimation from the probabilistic perspective. From the probabilistic approach, there are two types of SLAM problems classified based on the posterior generated map, online SLAM, and full SLAM problem. The online SLAM problem involves the posterior estimation process over the momentary of robot pose along with the map [18]. Meanwhile, the full SLAM problem is a problem of estimating the posterior over all the robot's pose, instead of the single entire along with the map. Graphically, they can be illustrated as shown in Figure 2.2 and Figure 2.3, respectively.

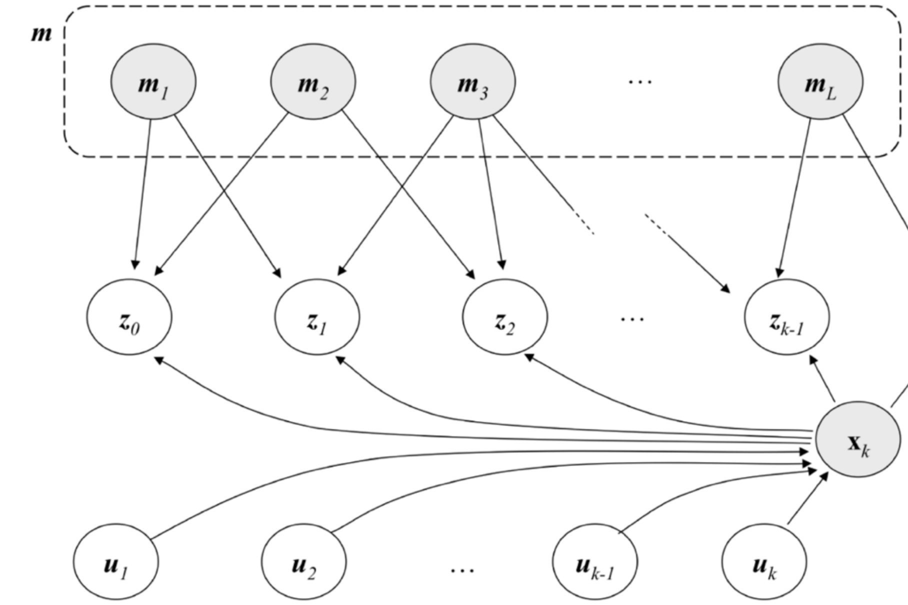


Figure 2.2 The Graphical Representation of Online SLAM problem

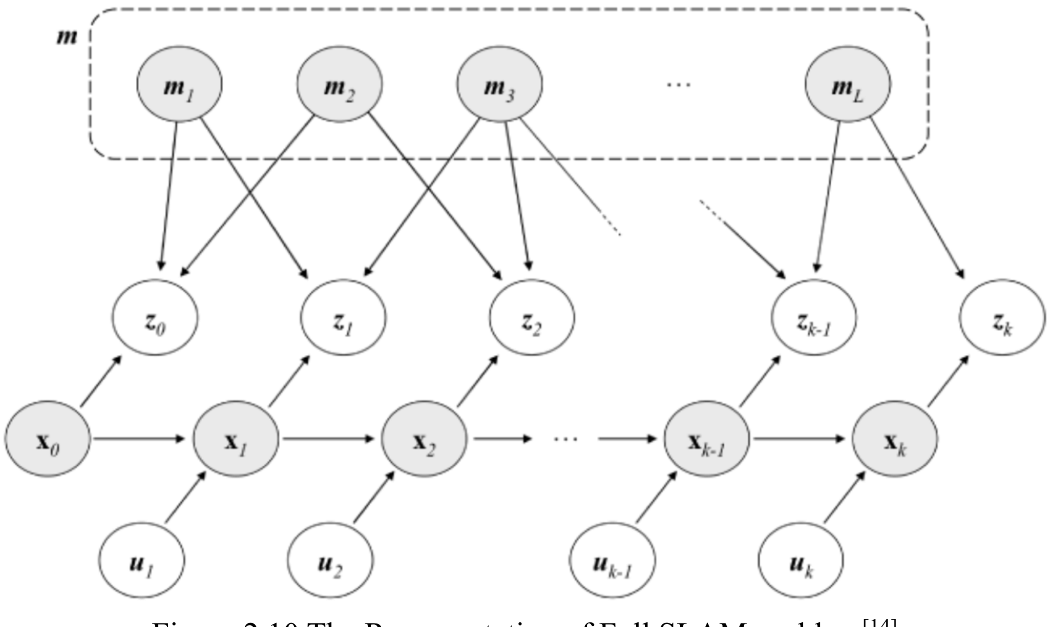


Figure 2.10 The Representation of Full SLAM problem[14]

The difference between the two SLAM problems described above can be probabilistically recognized. Since the objective of online SLAM problem is to generate the marginalized pose of the robot along with map at the time k, it can be probabilistically represented as follows

$$p\left(x_k, m_{1:k} | z_{1:k}, u_{1:k}\right) \tag{2.8.4.1}$$

where x_k refers to the pose of the robot relative to the global coordinate system representing both its spatial position and orientation $x_R = \left[x_{R,k}, y_{R,k}, \theta_{R,k}\right]^T$, which is not the x-coordinate of the robot pose. Meanwhile, m represents the map of environment. And $z_{1:k}$ and $u_{1:k}$ are respectively representing the measurement and control command. Meanwhile, the probabilistic consideration of full SLAM problem can be described as follows

$$p\left(x_{1:k}, m | z_{1:k}, u_{1:k}\right) \quad (2.8.4.2)$$

Regarding the graphical representation and Equation (3.7.2), it is clear to declare that the full SLAM problem is estimating all the entire pose $x_{1:k}$ given the set of measurement $z_{1:k}$ and $u_{1:k}$. However, the full SLAM is beyond the scope of this dissertation. Furthermore, when the consideration of correspondence between the measurement and map is involved, the probabilistic representation of online SLAM can be described as follows.

$$p\left(x_k, m, c_k | z_{1:k}, u_{1:k}\right) \quad (2.8.4.3)$$

Note, the c_k represents the correspondence about the global coordinate system as discussed on the localization algorithm. Since the main objective solving feature-based SLAM problem is to approximate the location of all the landmarks on the environment and the marginalized position of the robot in the global coordinate system, all the Gaussian State Estimator discussed earlier can be applied. As the composed algorithm of SLAM, it requires the transition of the robot and the way how the robot detects the landmark. Additionally, the system requires to know how to differentiate the seen or unseen landmark when it performs a measurement and construct the appropriately new landmark from the raw measurement and find the correspondences. For these reasons, the motion model, direct-based observation model, inverse-point based observation, feature extraction, and landmark registration, and incremental likelihood principle are also used to build a feature-based online SLAM algorithm based on Gaussian State Estimation. It is discussed clearly in the second next chapter herein this dissertation.

Chapter III Adaptively Determining The Recursive Formulation of Noise Statistic for The Conventional Filtering Method

The traditional filtering method requires the accurate system model and known noise statistic. However, in real applications, there is almost no specific system model caused by some factors, including the values of physical parameters, initial conditions, or noise characteristics. Furthermore, there is no exact manner to predefine all the noise statistics of the process, measurement, and corresponding covariances. Consequently, applying the filter without any modification approach might degrade the estimation method's optimality, which sufficiently increases the estimation error. Thus, all the uncertain parameters and noise statistics should be estimated as an effort to alleviate such effects. This estimation can be done during the filtering process by augmenting the adaptation mechanism, well-known as the Adaptive Filtering process.

The main objective of adaptive filters is to tune the filter gain based on the parametric variation or noise statistic that it is considered into the filtering process. The modification approach of adaptive filter leads to the conventional filter for having the ability to estimate the noise statistics and their corresponding covariance recursively. Henceforth, the time-varying noise statistic and covariance are available in the filtering process. The gain adaptation-based adaptive filter can be classified into three different approaches[44].

Joint Filtering of State and Parameters

All the unknown parameter of the systems is considered as the additional state. Therefore, the new state vector contains the former and other state representing the unknown parameter. Both of them are then used to calculate the posterior estimate of the new state vector. There is a common method adopting the principle of Extended Kalman Filter and Particle Filter. Although the corrective gain is tuned by referring to both former and additional states, this combination often causes numerical instability in general for some cases.

On-line noise tuning

According to the error measurement, the filtering performance is regarded whether there is divergence condition that occurs or not. If it is, the levels of measurement noise and or modeling uncertainties are then tuned by adopting some techniques. If it is not, the filtering process keeps the solution.

Batch estimation of parameters

In this approach, specific off-line techniques are adopted for estimating the system and noise statistic parameters based on a batch of measurements.

All the types of gain adaptation-based adaptive filters aim to estimate the unknown parameters in the purposes to improve the gain effect to the posterior state estimate. In this thesis, the types of batch estimation of parameters are discussed. The discussion of the adaptive filter is used not only to improve the performance of optimal filtering type, Extended Kalman Filter but also to improve the performance of the robust filtering type, Smooth Variable Structure Filter.

First, the dynamic system of the Gaussian Nonlinear System is considered for having the following characteristic.

$$\begin{cases} x_k = f(x_{k-1}, u_{k-1}) + \omega_k \\ z_k = h(x_k) + v_k \end{cases} \tag{3.1}$$

where k refers to the discrete time index, $x \in \mathbb{R}^n$ is the representation of state vector, u refers to the control vector, and $z \in \mathbb{R}^m$ is the representation of measurement vector. Meanwhile, ω and v are the small additive noises of the process and measurement, respectively, in which their corresponding covariances are denoted by Q and R, respectively. Furthermore, $f(\cdot)$ and $h(\cdot)$ refer to the state transition and measurement function, respectively. It is assumed that the characteristic of this dynamic system model is expressed as follows

$$\begin{cases} E[\omega_k] = q_k, Cov[\omega_k, \omega_j] = Q_k \delta_{kj}, \\ E[v_k] = r_k, Cov[v_k, v_j] = R_k \delta_{kj}, \\ E[\omega_k, v_j] = 0 \end{cases} \tag{3.2}$$

where δ is Kronecker delta function. Whereas, $E[\cdot]$ and $Cov[\cdot]$ represent mean and covariance term, respectively. Since, the values of the process and measurement noise are nonzero mean but instead q and r, respectively, then

$$\begin{cases} \mu_k = \omega_k - q \\ \eta_k = v_k - r \end{cases} \tag{3.3}$$

Therefore, the equivalent formulation of equation (3.1) can alternatively be written as follows

$$\begin{cases} x_k = f(x_{k-1}, u_{k-1}) + \mu_k + q \\ z_k = h(x_k)\eta_k + r \end{cases} \tag{3.4}$$

To equip the conventional EKF for having an ability to recursively estimate the noise statistic, the Maximum A Posterior and Maximum Likelihood Estimation[4], [8] are separately used refers to Batch Estimation of Parameters.

3.1 Designing Adaptive Extended Kalman Filter Using Principle of Maximum Likelihood Estimator and Expectation-Maximization

Besides having an ability to approximate the process and measurement noise recursively, this adaptation approach allows the new filtering method for also estimating the corresponding covariance representing the uncertainty to the process and measurement. Firstly, the MLE and EM are separately used to derive the conventional EKF. It aims to find the unknown parameters of the noise statistic. The derivation seems to be unobservable because of the requirement of estimates values from the original form. However, the required values are essentially unavailable in the form of EKF. Thus, EKF is modified and improved to tune the estimated value given by the MLE and EM creation. This modification is adopted from the principle of one lag smoothing point introduced by Gustafsson in 2000[41], [63], [67]. This process allows the EKF for having the tuned gain before it is proceeded to derive the main process of adaption. As a note that, the adaption process using MLE and EM creation also approximating the simplification into the multistep smoothing of some estimate values. Consequently, the risk of having the degradation to the filter stability is high. Additional to modify the EKF, the unbiased estimation is also involved to guarantee the filter quality in terms of stability and robustness. By conducting this process, the solution of recursive noise statistic is then almost complete. For the purpose of completing the last step of obtaining the time-varying noise statistic and their corresponding covariances, the weighted exponent strategy is involved. Finally, the optimal adaptive filter is done up to this point. However, the presence of non-positive definite covariance noise statistics should be concerned, which can diverge the filter solution. Accordingly, an additional Innovation Covariance Estimation is utilized, aiming to guarantee the time-varying covariance noise statistic of the process and measurement. This approach utilizes the principle of ICE[8], [61] to tune the quadratic error measurement, which can precisely give positive values to all the elements of the representative matrices of the covariances. The process of finding the recursive noise statistic and their covariance is detail discussed as follows.

Firstly, the model of the non-Gaussian system presented by Equation (3.1) and all its characteristics are recalled. Secondly, assuming that $\theta = (q, r, Q, R)$ are the unknown parameters representing the noise statistic of the process and measurement and their corresponding covariances, respectively. It is assumed as the desire of the adaption process, which can be found by recursively estimating it. Afterward, the estimated values of θ can be obtained by utilizing the Maximum Likelihood Estimation [8]. It can be expressed as follows

$$\hat{\theta}^{mle} = \text{argmax}_{\theta}\{ln[L(q, r, Q, R|Z_k, X_k)]\} \tag{3.1.1}$$

where $L(q, r, Q, R|Z_k, X_k)$ is termed as the likelihood function of $\theta = (q, r, Q, R)$. It can be expanded as follows

$$L(q,r,Q,R|Z_k,X_k) = p(Z_k,X_k|q,r,Q,R) = p(X_k|q,r,Q,R)p(Z_k|X_k,q,r,Q,R) \tag{3.1.2}$$

For $X_k = [x_1, x_2, \ldots, \ldots, x_{k-1}, x_k]$ and $Z_k = [z_1, z_2, \ldots, \ldots, z_{k-1}, z_k]$. Since Equation (3.1) is the first-order Markov process, Equation (3.1.2) can be expressed as follows

$$p(Z_k,X_k|\theta) = p[x_0]\prod_{i=1}^{k} p[x_i|x_{i-1},q,Q]\prod_{i=1}^{k} p[z_i|x_i,r,R] \tag{3.1.3}$$

Then by considering that these prior knowledges are normally distributed, then Equation (3.1.3) can be derived as follows

$$p(Z_k,X_k|q,r,Q,R) = \frac{1}{2\pi^{\frac{k(n+m)+n}{2}}}|P_0|^{-\frac{1}{2}}|Q|^{-\frac{k}{2}}|R|^{-\frac{k}{2}}\exp\left\{-\frac{1}{2}\left[\,\|x_0-\hat{x}_0\|^2_{P_0^{-1}} + \sum_{i=1}^{k}\|x_i - f(x_{i-1}) - q_{i-1}\|^2_{Q^{-1}} + \sum_{i=1}^{k}\|z_i - h(x_i) - r_i\|^2_{R^{-1}}\right]\right\} \tag{3.1.4}$$

by taking logarithm, it yields

$$\ln[L(q,r,Q,R|Z_k,X_k)] = -\frac{k(n+m)+n}{2}\ln(2\pi) - \frac{1}{2}\ln(|P_0|) - \frac{k}{2}\ln(|Q|) - \frac{k}{2}(|R|) - \frac{1}{2}\|x_0-\hat{x}_0\|^2_{P_0^{-1}} - \frac{1}{2}\sum_{i=1}^{k}\|x_i - f(x_{i-1}) - q_{i-1}\|^2_{Q^{-1}} + \sum_{i=1}^{k}\|z_i - h(x_i) - r_i\|^2_{Q_{-1}}$$

In this point, the role of Expectation-Maximization[71] is utilized to solve all the equations of suboptimal-Maximum Likelihood Estimation above. It aims to approximate both the process and measurement noise statistic as the suboptimal solution. Basically, the process under this principle follows two main steps, namely expectation-based strategy and maximization-based strategy. These steps are done sequentially. For the detail, it can be presented as follows

This manner proceeds the derived equation by taking the expectation of the logarithm form $\ln[L(q,r,Q,R|Z_k,X_k)]$. Analytically, it can be presented below

$$E\big[\ln[L(q,r,Q,R|Z_k,X_k)]\big]=-\frac{k(n+m)+n}{2}\ln(2\pi)-\frac{1}{2}\ln(|P_0|)-\frac{k}{2}\ln(|Q|)-\frac{k}{2}\ln(|R|)-\frac{1}{2}E\Bigg[\|x_0-\hat{x}_0\|^2_{P_0^{-1}}+\sum_{i=1}^{k}\|x_i-f(x_{i-1})-q_{i-1}{}^2_{Q^{-1}}+\sum_{i=1}^{k}\|z_i-h(x_i)-r_i\|^2_{R^{-1}}\Bigg] \tag{3.1.5}$$

Supposing that C is a representative form of all the constant

$$C=-\frac{k(n+m)+n}{2}\ln(2\pi)-\frac{1}{2}\ln(|P_0|)-\frac{1}{2}E\big[\|x_0-\hat{x}_0\|^2_{P_0^{-1}}\big] \tag{3.1. 6}$$

then the equivalent form of Equation (3.1.5) can be alternatively written as

$$E\big[\ln[L(q,r,Q,R|Z_k,X_k)]\big]=C-\frac{k}{2}\ln(|Q|)-\frac{k}{2}\ln(|R|)-\frac{1}{2}E\Bigg[\sum_{i=1}^{k}\|x_i-f(x_{i-1})-q_{i-1}{}^2_{Q^{-1}}+\sum_{i=1}^{k}\|z_i-h(x_i)-r_i\|^2_{R^{-1}}\Bigg] \tag{3.1.7}$$

by definition that $\|a\|^2_{b^{-1}}=a^Tb^{-1}a$, thus Equation (3.1.7) can be sufficiently derived as follows

$$J=C-\frac{k}{2}\ln(|Q|)-\frac{k}{2}\ln(|R|)-\frac{1}{2}\sum_{i=1}^{k}E\Big\{tr\big[Q^{-1}\big(x_i-f(x_{i-1})-q_{i-1}\big)\big(x_i-f(x_{i-1})-q_{i-1}\big)^T\big]\Big\}-\frac{1}{2}\sum_{i=1}^{k}E\Big\{tr\big[R^{-1}\big(z_i-h(x_i)-r_i\big)\big(z_i-h(x_i)-r_i\big)^T\big]\Big\} \tag{3.1.8}$$

where J represent the objective function of $E\big[\ln[L(q,r,Q,R|Z_k,X_k)]\big]$.

At this point the estimate noise statistic can be obtained by maximizing J function. It can be done by taking the partial derivative of J with respect to the unknown parameters θ, which are q,r,Q and R. Afterward it is proceeded by equating all the partial derivative to be equal to zero. Analytically, it can be expressed as

$$\frac{\partial J}{\partial q}=0,\frac{\partial J}{\partial r}=0,\frac{\partial J}{\partial Q}=0,\frac{\partial J}{\partial R}=0 \tag{3.1.9}$$

Since Q and R are assumed to be positive definite matrices, then it can be described as follows

$$\begin{cases} |Q|=tr(Q) \\ |R|=tr(R) \end{cases} \tag{3.1.10}$$

For $|\cdot|$ represents the determinant matrix operation and $tr(.)$ represents the trace of matrix. Therefore, the partial derivative of J with respect to q, r, Q, and R, can be respectively calculated as follows

$$\hat{q}_k=\frac{\partial J}{\partial q}=\frac{1}{k}\sum_{i=1}^{k} x_{i|k}-f(x_{i-1|k}) \tag{3.1.11}$$

$$\hat{r}_k=\frac{\partial J}{\partial r}=\frac{1}{k}\sum_{i=1}^{k} z_{i|k}-h(x_{i|k}) \tag{3.1.12}$$

$$\hat{Q}_k=\frac{\partial J}{\partial Q}=\frac{1}{k}\sum_{i=1}^{k}\left(x_{i|k}-f(x_{i-1|k})-q\right)\left(x_{i|k}-f(x_{i-1|k})-q\right)^T \tag{3.1.13}$$

$$\hat{R}_k=\frac{\partial J}{\partial R}=\frac{1}{k}\sum_{i=1}^{k}\left(z_{i|k}-h(x_{i|k})-r\right)\left(z_{i|k}-h(x_{i|k})-r\right)^T \tag{3.1.14}$$

The complicated multi-step of smoothing term $x_{i|k}$ and $x_{i-1|k}$ in Equation (3.1.11) – Equation (3.1.14) is required to be simplified in order to ably continue the adaption process. It can be done by replacing $x_{i-1|k}$ with $x_{i-1|i}$ and $x_{i|k}$ with $x_{i|i}$. Similarly, it is conducted to the complicated form of $z_{i|k}$ as well. Therefore, the following equations are presented

$$\hat{q}_k=\frac{1}{k}\sum_{i=1}^{k} x_{i|i}-f(x_{i-1|i}) \tag{3.1.15}$$

$$\hat{r}_k=\frac{1}{k}\sum_{i=1}^{k} z_{i|i}-h(x_{i|i}) \tag{3.1.16}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k}\left(x_{i|i} - f(x_{i-1|i}) - q\right)\left(x_{i|i} - f(x_{i-1|i}) - q\right)^T \quad (3.1.17)$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}\left(z_{i|i} - h(x_{i|i}) - r\right)\left(z_{i|i} - h(x_{i|i}) - r\right)^T \quad (3.1.18)$$

Regarding to Equation (3.1.15) – Equation (3.1.18), it can be noted that $x_{i-1|i}$ is precisely not given by the original form of Extended Kalman Filter. For this reason, besides deriving its form under assumptions of Maximum Likelihood Estimation and Expectation-Maximization creation[8], the use of one step smoothing point is also involved aiming to tune the corrective gain. Besides that, it aims to provide the estimate values which cannot be found directly from the conventional EKF.

By referring to Appendix A, the estimate value of $\hat{x}_{i-1|i}$ and $x_{i|i}$ can be adopted from Equation (A.6) and Equation (A.13), respectively. However, the simplification of multistep smoothing conducted above might degrade the quality of the optimal Maximum Likelihood Estimation and Expectation-Maximization form. Therefore, in order to anticipate the risk, the unbiased estimation is involved. The process can be mathematically shown below

First, by substituting Equation (A.7) into Equation (A.12), the general term of $\hat{x}_{i|i} - f(x_{i-1|i})$ in Equation (3.1.15) and Equation (3.1.17) can be sufficiently rewritten as

$$\hat{x}_{i|i} - f(\hat{x}_{i-1|i}) = K_i e_{z,i|i-1} + q_{i-1} \quad (3.1.19)$$

Then replacing $\hat{x}_{i|i}$ with Equation (A.12), the general term of $\hat{z}_i - h(\hat{x}_{i|i})$ in Equation (3.1.16) and Equation (3.1.18) can also be rewritten as

$$z_i - h(\hat{x}_{i|i}) = z_i - h(\hat{x}_{i|i-1} + K_i e_{z,i|i-1}) \quad (3.1.20)$$

Therefore, it yields

$$\hat{q}_k = \frac{1}{k}\sum_{i=1}^{k} K_i e_{z,i|i-1} + q_{i-1} \quad (3.1.21)$$

$$\hat{r}_k = \frac{1}{k}\sum_{i=1}^{k}(I - HK_i)e_{z,i|i-1} + r_i \tag{3.1.22}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k}K_i e_{z,i|i-1}e_{x,i|i-1}^T K_i^T \tag{3.1.23}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}(I - HK_i)e_{z,i|i-1}e_{x,i|i-1}^T(I - HK_i)^T \tag{3.1.24}$$

Since the innovation or error measurement $e_{z,k|k-1}$ and its covariance $e_{k|k-1}e_{k|k-1}^T$ are contained in the estimate values of the process and measurement noise statistic, thus

$$e_{z,i|i-1} = h(\tilde{x}_{i|i-1}) + v_i - r_i \tag{3.1.25}$$

Assuming that Equation (4.1.49) is satisfied

$$e_{z,k|k-1}e_{z,k|k-1}^T = S_i \tag{3.1.26}$$

It is noted that the corrective gain in Equation (A.5) can be derived as

$$\begin{aligned} K_i =& P_{i|i-1}H^T(S_i)^{-1} \\ & P_{i|i-1}H^T(e_{z,i|i-1}e_{z,k|k-1}^T)^{-1} \end{aligned} \tag{3.1.27}$$

It is obvious to obtain the following equations

$$K_i e_{z,i|i-1}e_{z,i|i-1}^T = P_{i|i-1}H^T \tag{3.1.28}$$

$$e_{z,i|i-1}e_{z,i|i-1}^T K_i^T = (K_i e_{z,i|i-1}e_{z,i|i-1}^T)^T = HP_{i|i-1}^T = HP_{i|i-1} \tag{3.1.29}$$

$$K_i e_{z,i|i-1}e_{z,i|i-1}^T K_i^T = K_i HP_{i|i-1}^T = K_i HP_{i|i-1} \tag{3.1.30}$$

Once Equation (3.1.30) is calculated, then it is clear to have

$$\begin{aligned} P_{i|i} &= (I - K_i H)P_{i|i-1} \\ P_{i|i} &= P_{i|i-1} - P_{i|i-1}K_i H \\ P_{i|i-1}K_i H &= P_{i|i-1} - P_{i|i} \end{aligned} \tag{3.1.31}$$

Substituting Equation (3.1.31) into Equation (3.1.30), then

$$K_i e_{z,i|i-1}e_{z,i|i-1}^T K_i^T = P_{i|i-1}P_{i|i} = FP_{i-1|i-1}F^T + Q_{i-1}P_{i|i} \tag{3.1.32}$$

Now assuming that the expectation form of $e_{z,i|i-1}$ and $e_{z,i|i-1}e_{z,i|i-1}^T$ are

$$E[e_{z,i|i-1}] = 0 \tag{3.1.33}$$

$$E[e_{z,i|i-1}e_{z,i|i-1}^T] = HP_{i|i-1}H^T + R_i \tag{3.1.34}$$

By substituting Equation (3.1.25) – Equation (3.1.34) into Equation (3.1.21) – Equation (3.1.24), it is clear to have

$$E[\hat{q}_k] = q_k \tag{3.1.35}$$

$$E[\hat{r}_k] = r_k \tag{3.1.36}$$

$$E[\hat{Q}_k] = Q_k + E\left[\frac{1}{k}\sum_{i=1}^{k} FP_{i|i-1}F^T - P_{i|i}\right] \tag{3.1.37}$$

Once Equation (3.1.32) is calculated, it can be used compactly reform the noise statistic R. The process is sequentially done as follows. The formulation of covariance matrix of measurement noise statistic R can be derived as

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}(I - K_iH)e_{z,i|i-1}e_{z,i|i-1}^T(I - K_iH)^T \tag{3.1.38}$$

for

$$\begin{aligned}(I - K_iH)e_{z,i|i-1}e_{z,i|i-1}^T(I - K_iH)^T = e_{z,i|i-1}e_{z,i|i-1}^T - HK_ie_{z,i|i-1}e_{z,i|i-1}^T - \\ e_{z,i|i-1}e_{z,i|i-1}^TK_i^TH^T + HK_ie_{z,i|i-1}e_{z,i|i-1}^TK_i^TH^T\end{aligned} \tag{3.1.39}$$

$$\begin{aligned}(I - K_iH)e_{z,i|i-1}e_{z,i|i-1}^T(I - K_iH)^T &= HP_{i|i-1}H^T - 2HP_{i|i-1}H^T + HK_iHP_{i|i-1}H^T \\ &= R_i - HP_{i|i-1}H^T + HK_iHP_{i|i-1}H^T \\ &= R_i - (I - HK_i)HP_{i|i-1}H^T\end{aligned} \tag{3.1.40}$$

Then by substituting Equation (3.1.25) – Equation (3.1.34) into Equation (3.1.21) – Equation (3.1.24), it is clear to also have

$$\begin{aligned} E[\hat{R}_k] =&E\left\{\frac{1}{k}\sum_{i=1}^{k}(I-K_iH)e_{z,i|i-1}e_{z,i|i-1}^T(I-K_iH)^T\right\} \\ =&E\left\{\frac{1}{k}\sum_{i=1}^{k}R_i-(I-K_iH)HP_{i|i-1}H^T\right\} \\ =&R_k+E\left\{\frac{1}{k}\sum_{i=1}^{k}-(HP_{i|i-1}H^T-HK_iHP_{i|i-1}H^T)\right\} \\ =&R_k+E\left\{\frac{1}{k}\sum_{i=1}^{k}HK_iHP_{i|i-1}H^T-HP_{i|i-1}H^T\right\} \end{aligned} \tag{3.1.41}$$

where R_k refers to Equation (3.1.24), then Equation (3.1.41) becomes

$$\hat{R}_k=\frac{1}{k}\sum_{i=1}^{k}(I-K_iH)e_{z,i|i-1}e_{z,i|i-1}^T(I-K_iH)^T+HK_iHP_{i|i-1}H^T-HP_{i|i-1}H^T \tag{3.1.42}$$

where

$$\begin{aligned} (I-K_iH)&e_{z,i|i-1}e_{z,i|i-1}^T(I-K_iH)^T+HK_iHP_{i|i-1}H^T-HP_{i|i-1}H^T \\ =&e_{z,i|i-1}e_{z,i|i-1}^T-HK_ie_{z,i|i-1}e_{z,i|i-1}^T-e_{z,i|i-1}e_{z,i|i-1}^TK_i^TH^T+ \\ &HK_ie_{z,i|i-1}e_{z,i|i-1}^TK_i^TH^T+HK_iHP_{i|i-1}H^T-HP_{i|i}H^T \\ =&e_{z,i|i-1}e_{z,i|i-1}^T-HK_ie_{z,i|i-1}e_{z,i|i-1}^T-e_{z,i|i-1}e_{z,i|i-1}^TK_i^TH^T+ \\ &HK_ie_{z,i|i-1}e_{z,i|i-1}^TK_i^TH^T+(I-K_iH)(HP_{i|i-1}H^T) \\ =&(I-HK_i)\left[e_{z,i|i-1}e_{z,i|i-1}^T(I-K_iH)^T+HP_{i|i-1}H^T\right] \end{aligned} \tag{3.1.43}$$

Then the equivalent estimate values of R is obtained as

$$\hat{R}_k=\frac{1}{k}\sum_{i=1}^{k}(I-K_iH)\left[e_{z,i|i-1}e_{z,i|i-1}^T(I-K_iH)^T+HP_{i|i-1}H^T\right] \tag{3.1.44}$$

Similarly, since q_k, r_k, and Q_k in Equation (3.1.35) – Equation (3.1.37) are respectively $\hat{q}_k, \hat{r}_k$, and $\hat{Q}_k$ in Equation (3.1.21) – Equation (3.1.23), then their recursive forms of noise statistic are

$$\hat{q}_k=\frac{1}{k}\sum_{i=1}^{k}K_ie_{z,i|i-1}+q_{i-1} \tag{3.1.45}$$

$$\hat{r}_k = \frac{1}{k}\sum_{i=1}^{k}(I - HK_i)e_{z,i|i-1} + r_i \tag{3.1.46}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k} K_i e_{z,i|i-1} e_{z,i|i-1}^T K_i^T + FP_{i-1|i-1}F^T - P_{i|i} \tag{3.1.47}$$

Note that Equation (3.1.45) – Equation (3.1.47) represent the properties of the unbiased recursive noise statistic of Equation (3.1.21) – Equation (3.1.24). Therefore, according to Equation (3.1.44) – Equation (3.1.47), the time-varying noise properties can be derived as follows

$$\hat{q}_k = \hat{q}_{k-1} + \frac{1}{k}(K_k e_{z,k|k-1}) \tag{3.1.48}$$

$$\hat{r}_k = \hat{r}_{k-1} + \frac{1}{k}\left[(I - K_k H)e_{z,k|k-1}\right] \tag{3.1.49}$$

$$\hat{Q}_k = \frac{k-1}{k}\hat{Q}_{k-1} + \frac{1}{k}\left[K_k e_{z,k|k-1} e_{z,k|k-1}^T K_k^T + F_{k-1}P_{k|k-1}F_{k-1}^T - P_{k|k}\right] \tag{3.1.50}$$

$$\hat{R}_k = \frac{k-1}{k}\hat{R}_{k-1} + \frac{1}{k}(I - K_k H)\left[e_{z,k|k-1} e_{z,k|k-1}^T (I - K_k H)^T + HP_{k|k-1}H^T\right] \tag{3.1.51}$$

Therefore, by applying the weighted exponent[4], [8], [13], [14], [79], in where the weighting coefficient d_k is formulated to replace the exponential $\frac{1}{k}$, then the alternative formulation of Equation (3.1.48) – Equation (3.1.51) become as follows

$$\hat{q}_k = \hat{q}_{k-1} + d_k(K_k e_{z,k|k-1}) \tag{3.1.52}$$

$$\hat{r}_k = \hat{r}_{k-1} + d_k\left[(I - K_k H)e_{z,k|k-1}\right] \tag{3.1.53}$$

$$\hat{Q}_k = (1 - d_k)\hat{Q}_{k-1} + d_k\left[K_k e_{z,k|k-1} e_{z,k|k-1}^T K_k^T + FP_{k-1|k-1}F^T - P_{k|k}\right] \tag{3.1.54}$$

$$\hat{R}_k = (1 - d_k)\hat{R}_{k-1} + d_k(I - K_k H)\left[e_{z,k|k-1} e_{z,k|k-1}^T (I - K_k H)^T + HP_{k|k-1}H^T \right. \tag{3.1.55}$$

where the weighting exponent are expressed as

$$\begin{cases} \beta_i = d_k \beta^{i-1} \\ d_k = \frac{1-b}{1-b^k} \end{cases}, for\ i = 1,,n \tag{3.1.56}$$

where b is a fading factor satisfying $0 < b < 1$ and β_i refers to i-th weighting factor defined as $\beta_i = \beta_{i-1} b$ and satisfying $\sum_{i=1}^{k} \beta_i$. Up to this point, considering that Q and R are positive definite matrices. Thus, to guarantee Equation (3.1.54) and Equation (3.1.55) to be positive definite matrices, the innovation covariance estimation is involved to the proposed method. It has been proven to be able to depress the filter divergence as introduced in[20], [21], [59], [61], [73]. The process can be summarized as follows. First, assuming that the following form is representation of innovation covariance indexed by k

$$ICE_k = \frac{1}{k} \sum_{j=0}^{N} e_{z,k|j-1} e_{z,k|j-1}^T = ICE_k + \frac{1}{N} \left[e_{z,k} e_{z,k}^T e_{z,k-N} e_{z,k-N}^T \right] \tag{3.1.57}$$

by replacing $e_{z,k|k-1} e_{z,k|k-1}^T$ with ICE_k, equation (3.1.54) and Equation (3.1.55) can be alternatively rewritten as follows

$$\hat{Q}_k = (1 - d_k) \hat{Q}_{k-1} + d_k \left[K_k ICE_k K_k^T + F P_{k-1|k-1} F^T - P_{k|k} \right] \tag{3.1.58}$$

$$\hat{R}_k = (1 - d_k) \hat{R}_{k-1} + d_k (I - K_k H) \left[ICE_k (I - K_k H)^T + H P_{k|k-1} H^T \right] \tag{3.1.59}$$

As described by Equation (3.1.57), the window size N plays a role of achieving an ICE accuracy. N is able to prevent the occurrence of the biased situation in ICE by setting it to small value for the fast change of the dynamic system. Besides that, it is also able to improve the stability of the unbiased ICE by setting it to large value for the slow change of the dynamic system. Simply, the adjustment of window size is strongly depending on the characteristic of dynamic system which can empirically to be adjusted. Then, it is noted that the noise estimator is stable when the following definition is satisfied[6]

$$tr(ICE_k) \le \kappa . tr(S) \tag{3.1.60}$$

where κ is a reserve factor that satisfying $\kappa \leq 1$ and $tr(\cdot)$ refers to the matrix trace. The stability equation expressed by Equation (4.1.83) shows that κ also plays an important role as a threshold to know the incorrectness of the noise statistic when the measurement outliers occurs. By means, it is an issue when the ratio between $tr(ICE)$ and $tr(S)$ is out of the threshold value κ at current step k. Therefore, an additional of the innovation covariance estimation will obviously be keeping the stability of estimated value Q and R by isolating the current innovation covariance $e_{z,k|k-1}e_{z,k|k-1}^T$ with its calculated value. Up to this point, the adaptive EKF can be graphically summarized as follows

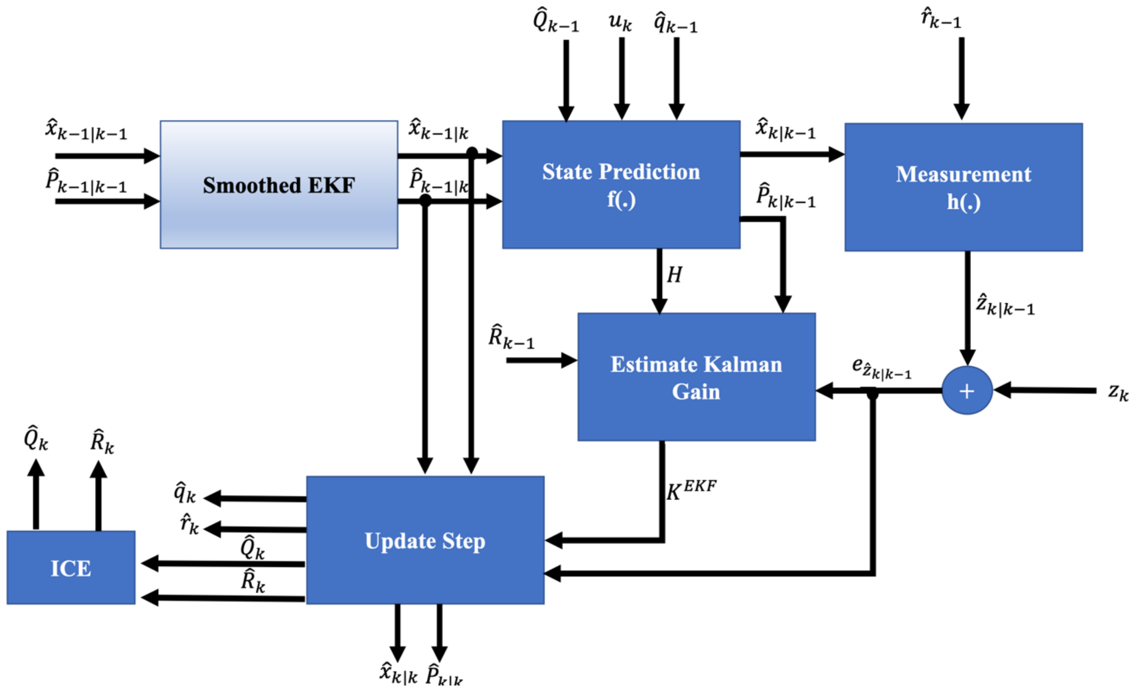


Figure 3.1 Working Principle of Adaptive EKF Based on MLE and ICE

3.2 Designing Adaptive Extended Kalman Filter Based on Maximum A Posteriori Estimation and Weighted Exponent

Besides using Maximum Likelihood Estimation and Expectation-Maximization creation, the adaptive Extended Kalman Filter can be formulated by involving the principle of Maximum A Posteriori and Weighted Exponent. There is no difference in this strategy since the solution offered in this way is the same as the previous solution. But, in this way, the process adaption is more straightforward and faster. Sequentially, the Maximum A Posterior[13], [50], [58], [79], [82] is used to derive the traditional EKF by assuming that the unknown parameters are the noise statistic of the process and measurements, which are not zero mean anymore. Additional to these parameters, their corresponding covariance are also estimated. There are some estimate values, which cannot be adopted directly from the original form of EKF. Therefore, the same strategy of EKF improvement is also conducted. It involves the use of a one-lag smoothing point. Essentially, by performing this improvement, the estimation error is sufficiently reduced because of the smoothing process. This smoothing process is commonly intended to tune the corrective gain aiming to present more responsive and proper gain to the estimation process. The improvement process is conducted to make the mathematical derivation to be observable. It returns the sub-optimal solution, which is the embryo of the recursive noise statistic and covariance for both the process and measurement. Furthermore, the simplification of the multistep smoothing point is also concerned. Consequentially, it might reduce the stability and quality of the estimated unknown parameters, increasing the risk of having a non-positive definite character to the covariance of the noise statistic for either the process or measurement. For this reason, a certain approach should be concerned. In this research, the divergence suppression method is used to tune and reconstruct the covariance of the error state $P_{k|k-1}$ of the smoothed EKF. This stage is also the reason that makes the approach of

Maximum A Posterior, and Maximum Likelihood Estimation is different. Before this stage is conducted, the unbiased estimation is also involved to guarantee that the adaptive EKF is still kept to have the unbiased characteristic to the estimated values. It is sequentially and separately conducted after performing the approach of the weighted exponent. In order to give a clear process of this method, the analytical process of Maximum A Posterior and Weighted Exponent Principle-assisted Extended Kalman Filter is presented below. Firstly, the model of the non-Gaussian system presented by Equation (3.1) – Equation (3.2) and all its characteristic Equation (3.3) – Equation (3.4) are recalled. It is then used as the base for the EKF formulation.

Secondly, a classical EKF was estimated by utilizing MAP creation[56], [69], [71], [81], [102] in order to responsively generate the noise statistic for the next iteration based on the previous iteration. Assuming that, the unknown parameters are the process q and measurement r noise statistics with their process covariance Q and measurement R which are characteristically assumed to be positive definite symmetric matrices. Assuming that $J = [X_k, q, r, Q, R, Z_k]$ is a joint probability density function described as follows

$$J = p[X_k, q, r, Q, R, Z_k] = p[Z_k|X_k, q, r, Q, R]p[X_k|q, r, Q, R]p[q, r, Q, R] \quad (3.2.1)$$

and considering that the conditional density function J^*

$$J^* = p[X_k, q, r, Q, R|Z_k] = \frac{p[X_k, q, r, Q, R, Z_k]}{p[Z_k]} \quad (3.2.2)$$

where $X_k = [x_1, x_2, \ldots, x_k]$ and $Z_k = [z_1, z_2, \ldots, z_k]$. It is regarded that $p[Z_k]$ plays no role in optimization, then by utilizing MAP the estimated values of $\theta = q, r, Q, R$ denoted by $\hat{\theta} = \hat{q}, \hat{r}, \hat{Q}, \hat{R}$ can be calculated by solving the following expression

$$\hat{\theta}_{MAP} = \arg\ \max_\theta\big(ln[J]\big) = \arg\ \max_\theta\big(ln[p[Z_k|X_k, q, r, Q, R]p[X_k|q, r, Q, R]]\big) \quad (3.2.3)$$

Where $p[q, r, Q, R]$ can be obtained from the initial information. Then, by assuming that Equation (3.1) is first order Markov process then $p[Z_k|X_k, q, r, Q, R]$ and

$p[X_k|q,r,Q,R]$ can be respectively factorized as follows

$$p[Z_k|X_k,q,r,Q,R] = \sum_{i=1}^{k} p[z_i|x_i, rR] \tag{3.2.4}$$

$$p[X_k|q,r,Q,R] = p[x_0|P_0] \sum_{i=1}^{k} p[x_i|x_{i-1}, q, Q] \tag{3.2.5}$$

Now, considering that Equation (3.2.5) is normally distributed, then

$$\begin{aligned} p[X_k|q,r,Q,R] =& \frac{1}{2\pi^{\frac{n}{2}}|P_0|^{\frac{1}{2}}} \exp\left[-\frac{1}{2} \|x_0 - \hat{x}_0\|^2_{P_0^{-1}} \right] \times \prod_{i=1}^{k} \frac{1}{2\pi^{-\frac{n}{2}}|Q|^{\frac{k}{2}}} \exp\left[-\frac{1}{2} \right. \\ & \left. \|x_i - f(x_{i-1}) - q\|^2_{Q^{-1}} \right] \\ =& \frac{1}{2\pi^{\frac{n}{2}}|P_0|^{\frac{1}{2}}} \exp\left[-\frac{1}{2} \|x_0 - \hat{x}_0\|^2_{P_0^{-1}} \right] \times \frac{1}{2\pi^{\frac{nk}{2}}|Q|^{\frac{k}{2}}} \exp\left[-\frac{1}{2} \sum_{i=1}^{k} \|x_i \right. \\ & \left. - f(x_{i-1}) - q^2_{Q^{-1}} \right] \end{aligned} \tag{3.2.6}$$

assuming that $C_1 = 2\pi^{(} - \frac{n(k+1)}{2}{)}$ represent a constant, then Equation (3.2.6) can be rewritten as follows

$$\begin{aligned} p[X_k,q,r,Q,R] =& C_1|P_0|^{-\frac{1}{2}}|Q|^{\frac{k}{2}} \times \exp\left\{ -\frac{1}{2}\left[\|x_0 - \hat{x}_0\|^2_{P_0^{-1}} \right] + \left[\|x_i - f(x_{i-1}) - \right. \right. \\ & \left. \left. q^2_{Q^{-1}} \right] \right\} \end{aligned} \tag{3.2.7}$$

Similarly, Equation (3.2.7) is also normally distributed then

$$\begin{aligned} p\left[Z_k|q,r,Q,R\right] =& \prod_{i=1}^{k} \frac{1}{2\pi^{\frac{m}{2}}|R|^{\frac{1}{2}}} \exp\left[-\frac{1}{2} \|z_i - h(x_i) - r\|^2_{R^{-1}} \right] \\ =& \frac{1}{2\pi^{\frac{mk}{2}}}|R|^{-\frac{k}{2}} \exp\left[-\frac{1}{2} \sum_{i=1}^{k} \|z_i - h(x_i) - r\|^2_{R^{-1}} \right] \end{aligned} \tag{3.2.8}$$

Assuming that $C_2 = 2\pi^{\frac{mk}{2}}$ represents a constant, then Equation (3.2.8) can be rewritten as follows

$$p[Z_k|X_k,q,r,Q,R] = C_2|R|^{-\frac{k}{2}} \exp\left[\sum_{i=1}^{k} \|z_i - h(x_i) - r\|^2_{R^{-1}}\right] \tag{3.2.9}$$

Substituting (3.2.7) and (3.2.9) into (3.2.2), then

$$\begin{aligned} J =& C_1C_2|P_0|^{-\frac{1}{2}}|Q|^{-\frac{k}{2}}|R|^{-\frac{k}{2}} \exp\Bigg\{ -\frac{1}{2}\Bigg[\|x_0 - \hat{x}_0\|^2_{P_0^{-1}} + \sum_{i=1}^{k} \|x_i - f(x_{i-1}) - q\| \\ &+ \sum_{i=1}^{k} \|z_i - h(x_i) - r\|^2_{R^{-1}} \Bigg]\Bigg\} \times p[q,r,Q,R] \end{aligned} \tag{3.2.10}$$

by assuming that $C = C_1C_2|P_0|^{-\frac{1}{2}} \exp[-\frac{1}{2}\|x_0 - \hat{x}_0\|^2_{P_0^{-1}}] \times p[q,r,Q,R]$, then (3.2.10) can be rewritten as follows

$$\begin{aligned} J =& C|Q|^{-\frac{k}{2}}|R|^{-\frac{k}{2}} \exp\Bigg\{ -\frac{1}{2}\Bigg[\sum_{i=1}^{k} \|x_i - f(x_{i-1}) - q\|^2_{Q^{-1}} + \sum_{i=1}^{k} \|z_i - h(x_i) - \\ & r^2_{R^{-1}} \Bigg]\Bigg\} \end{aligned} \tag{3.2.11}$$

At this point, by ignoring the constant and taking logarithm of Equation (3.2.11), yield

$$\begin{aligned} ln(J) =& -\frac{k}{2}ln|Q| - \frac{k}{2}ln|R| - \frac{1}{2}\Bigg[\sum_{i=1}^{k} \|x_i - f(x_{i-i}) - q\|^2_{Q^{-1}} + \sum_{i=1}^{k} \|z_i - h(x_i) - \\ & r^2_{R^{-1}} \Bigg] \end{aligned} \tag{3.2.12}$$

Substituting Equation (3.2.12) into Equation (3.2.2), then it is obvious the estimated $\hat{\theta}$ can be obtained by taking partial derivative an equating its result to zero as expressed below

$$\begin{cases} \hat{q} = \dfrac{\partial ln(J)}{\partial q}|_{q=\hat{q}_k} \\ \hat{r} = \dfrac{\partial ln(J)}{\partial r}|_{r=\hat{r}_k} \\ \hat{Q} = \dfrac{\partial ln(J)}{\partial Q}|_{Q=\hat{Q}_k} \\ \hat{R} = \dfrac{\partial ln(J)}{\partial R}|_{R=\hat{R}_k} \end{cases} \tag{3.2.13}$$

Then $\hat{q}$, $\hat{r}$, $\hat{Q}$, and $\hat{R}$ can respectively written as follows

$$\hat{q}_k = \frac{1}{k}\sum_{i=1}^{k} x_{i|k} - f(x_{i-1|k}) \tag{3.2.14}$$

$$\hat{r}_k = \frac{1}{k}\sum_{i=1}^{k} z_{i|k} - h(x_{i|k}) \tag{3.2.15}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k} \left(x_{i|k} - f(x_{i-1|k}) - q\right) \times \left(x_{i|k} - f(x_{i-1|k}) - q\right)^T \tag{3.2.16}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k} \left(z_{i|k} - h(x_{i|k}) - r\right) \times \left(z_{i|k} - h(x_{i|k}) - r\right)^T \tag{3.2.17}$$

The complicated multi-step smoothing term $x_{i|k}$ and $x_{i-1|k}$ in Equation (3.2.14) – Equation (3.2.16) might cause inefficiency of the MAP estimate. Therefore, in order to find the conventional and efficient recursive form the simplification is needed. Note, that the recursive update process only utilizes the estimate value at time k-1 and k hence the simplification can be conducted by replacing $x_{i-1|k}$ with $\hat{x}_{i-1|i}$ and $x_{i|k}$ with $\hat{x}_{i|i}$. Therefore, the suboptimal of MAP noise estimator can be expressed as follows

$$\hat{q}_k = \frac{1}{k}\sum_{i=1}^{k} \hat{x}_{i|i} - f(x_{i-1|i}) \tag{3.2.18}$$

$$\hat{r}_k = \frac{1}{k}\sum_{i=1}^{k} z_{i|i} - h(\hat{x}_{i|i}) \tag{3.2.19}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k}\left(\hat{x}_{i|i} - f(\hat{x}_{i-1|i}) - q\right) \times \left(\hat{x}_{i|i} - f(\hat{x}_{i-1|i}) - q\right)^T \tag{3.2.20}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}\left(z_{i|i} - h(\hat{x}_{i|i}) - r\right) \times \left(z_{i|i} - h(\hat{x}_{i|i}) - r\right)^T \tag{3.2.21}$$

As can be analyzed from the sequence equations above that the estimate value of $\hat{x}_{i-1|i}$ is not provided obviously by classical EKF. Therefore, modifying the original forms of EKF is required. It aims to compute the noise statistics estimator effectively. The process of modifying the EKF can be done by calculating the one-step smoothing of the EKF gain and its corresponding estimate value using the one-smoothing point algorithm[23], [67], [69] (see Appendix A).

Referring to Appendix A, the estimate value of $\hat{x}_{i-1|i}$ and $x_{i|i}$ can be adopted from Equation (A.6) and Equation (A.13), respectively. The simplification of multi-step smoothing conducted above might degrade the quality of MAP estimate. Thus, in order to depress this possibility, the unbiased estimation concept was utilized. The process can be described as follows.

First, by substituting Equation (A.7) into Equation (A.13), the general term $\hat{x}_{i|i} - f(x_{i-1|i})$ in Equation (3.2.18) and Equation (3.2.20) can be rewritten as follows

$$\hat{x}_{i|i} - f(\hat{x}_{i-1|i}) = K_k e_{z,k-1|k} + q \tag{3.2.22}$$

Then replacing $\hat{x}_{i|i}$ with Equation (A.13), the general term $z_i - h(\hat{x}_{i|i-1})$ in Equation (3.2.19) and Equation (3.2.21) can be written as follows

$$z_i - h(\hat{x}_{i|i-1} + K_i e_{z,i|i-1}) \tag{3.2.23}$$

and the suboptimal MAP estimation in (3.2.18) - (3.2.21) can be arranged as follows

$$\hat{q}_k = \frac{1}{k}\sum_{i=1}^{k} K_i e_{z,i|i-1} + q \tag{3.2.24}$$

$$\hat{r}_k = \frac{1}{k}\sum_{i=1}^{k}(I - HK_i) e_{z,i|i-1} + r \tag{3.2.25}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k} K_i e_{z,i|i-1} e_{z,i|i-1}^T K_i^T \tag{3.2.26}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k} (I - HK_i) e_{z,i|i-1} e_{z,i|i-1}^T (I - HK_i)^T \tag{3.2.27}$$

Since the innovation $e_{z,i|i-1}$ and its covariance $e_{z,i|i-1}e_{z,i|i-1}^T$ are contained in the process and measurement noise estimator, therefore

$$e_{z,i-1|i} = h(\widetilde{x}_{i|i-1}) + v_i - r \tag{3.2.28}$$

Assuming that the equation below is satisfied

$$e_{z,k|k-1} e_{z,k|k-1}^T = S_i \tag{3.2.29}$$

then the corrective gain in Equation (A.12) can be derived as

$$\begin{aligned} K_i =& P_{i|i-1} H^T (S_i)^{-1} \\ =& P_{i|i-1} H^T (e_{z,i|i-1} e_{z,k|k-1}^T)^{-1} \end{aligned} \tag{3.2.30}$$

$$e_{z,i|i-1} e_{z,i|i-1}^T K_i^T = (K_i e_{z,i|i-1} e_{z,i|i-1}^T)^T = HP_{i|i-1}^T = HP_{i|i-1} \tag{3.2.31}$$

$$K_i e_{z,i|i-1} e_{z,i|i-1}^T K_i^T = K_i HP_{i|i-1}^T = K_i HP_{i|i-1} \tag{3.2.32}$$

$$\begin{aligned} P_{i|i} &= (I - K_i H) P_{i|i-1} \\ P_{i|i} &= P_{i|i-1} - P_{i|i-1} K_i H \\ P_{i|i-1} K_i H &= P_{i|i-1} - P_{i|i} \end{aligned} \tag{3.2.33}$$

Substituting Equation (3.2.33) into Equation (3.2.32), it yields

$$K_i e_{z,i|i-1} e_{z,i|i-1}^T K_i^T = P_{i|i-1} P_{i|i} = FP_{i-1|i-1} F^T + Q_{i-1} P_{i|i} \tag{3.2.34}$$

taking expectation of $e_{z,i|i-1}$ and $e_{z,i|i-1}e_{z,i|i-1}^T$ then

$$E[e_{z,i|i-1}] = 0 \tag{3.2.35}$$

$$E[e_{z,i|i-1} e_{z,i|i-1}^T] = HP_{i|i-1} H^T + R \tag{3.2.35}$$

Now substituting Equation (3.2.28) – Equation (3.2.35) into Equation (3.2.24) –

Equation (3.2.26), it yields

$$E[\hat{q}_k] = q \tag{3.2.36}$$

$$E[\hat{r}_k] = r \tag{3.2.37}$$

$$E[\hat{Q}_k] = Q + E\left[\frac{1}{k}\sum_{i=1}^{k} FP_{i-1|i-1}F^T P_{i|i}\right] \tag{3.2.38}$$

However, to ease our calculation, the formulation of covariance matrix for the measurement noise statistic R should be derived first. It can be sequentially described as follows

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}(I - K_i H)e_{z,i|i-1}e_{z,i|i-1}^T(I - K_i H)^T \tag{3.2.39}$$

for

$$\begin{aligned}(I - K_i H)e_{z,i|i-1}e_{z,i|i-1}^T(I - K_i H)^T = e_{z,i|i-1}e_{z,i|i-1}^T - HK_i e_{z,i|i-1}e_{z,i|i-1}^T - \\ e_{z,i|i-1}e_{z,i|i-1}^T K_i^T H^T + HK_i e_{z,i|i-1}e_{z,i|i-1}^T K_i^T H^T\end{aligned} \tag{3.2.40}$$

Compactly, Equation (3.2.40) can be expressed as follows

$$\begin{aligned}(I - K_i H)e_{z,i|i-1}e_{z,i|i-1}^T(I - K_i H)^T =&(HP_{i|i-1}H^T + R_i) - 2HP_{i|i-1}H^T + \\ &HK_i HP_{i|i-1}H^T \\ =&R_i - HP_{i|i-1}H^T + HK_i HP_{i|i-1}H^T \\ =&R_i - (I - K_i H)HP_{i|i-1}H^T\end{aligned} \tag{3.2.41}$$

Then by substituting Equation (3.2.28) – Equation (3.2.35) and (3.2.41) into Equation (3.2.27), it yields

$$\begin{aligned} E[\hat{R}_k] =& E\left\{ \frac{1}{k} \sum_{i=1}^{k} (I - K_i H) e_{z,i|i-1} e_{z,i|i-1}^T (I - K_i H)^T \right\} \\ =& E\left\{ \frac{1}{k} \sum_{i=1}^{k} R_i - (I - K_i H) H P_{i|i-1} H^T \right\} \\ =& R_k + E\left\{ \frac{1}{k} \sum_{i=1}^{k} -(H P_{i|i-1} H^T - H K_i H P_{i|i-1} H^T) \right\} \\ =& R_k + E\left\{ \frac{1}{k} \sum_{i=1}^{k} H K_i H P_{i|i-1} H^T - H P_{i|i-1} H^T \right\} \end{aligned} \tag{3.2.42}$$

According to Equation (3.2.36), Equation (3.2.37), Equation (3.2.38), and Equation (3.2.42), it is known that $q, r, Q,$ and R are $\hat{q}_k$, $\hat{r}_k$, $\hat{Q}_k$, and $\hat{R}_k$, respectively. Therefore, it is clear to have the suboptimal recursive noise statistic as presented below

$$\hat{q}_k = \frac{1}{k} \sum_{i=1}^{k} K_i e_{z,i|i-1} + q \tag{3.2.43}$$

$$\hat{r}_k = \frac{1}{k} \sum_{i=1}^{k} (I - H K_i) e_{z,i|i-1} + r \tag{3.2.44}$$

$$\hat{Q}_k = \frac{1}{k} \sum_{i=1}^{k} K_i e_{z,i|i-1} e_{z,i|i-1}^T K_i^T + F P_{i-1|i-1} F^T - P_{i|i} \tag{3.2.45}$$

Since R in Equation (3.2.42) is R_k referring to Equation (3.2.27), then its recursive form can be derived as

$$\hat{R}_k = \frac{1}{k} \sum_{i=1}^{k} (I - K_i H) e_{z,i|i-1} e_{z,i|i-1}^T (I - K_i H)^T + H K_i H P_{i|i-1} H^T - H P_{i|i-1} H^T \tag{3.2.46}$$

where

$$
\begin{aligned}
&(I-K_iH)e_{z,i|i-1}e_{z,i|i-1}^T(I-K_iH)^T+HK_iHP_{i|i-1}H^T-HP_{i|i-1}H^T \\
&=e_{z,i|i-1}e_{z,i|i-1}^T-HK_ie_{z,i|i-1}e_{z,i|i-1}^T-e_{z,i|i-1}e_{z,i|i-1}^TK_i^TH^T+ \\
&\quad HK_ie_{z,i|i-1}e_{z,i|i-1}^TK_i^TH^T+HK_iHP_{i|i-1}H^T-HP_{i|i}H^T \\
&=e_{z,i|i-1}e_{z,i|i-1}^T-HK_ie_{z,i|i-1}e_{z,i|i-1}^T-e_{z,i|i-1}e_{z,i|i-1}^TK_i^TH^T+ \\
&\quad HK_ie_{z,i|i-1}e_{z,i|i-1}^TK_i^TH^T+(I-K_iH)(HP_{i|i-1}H^T) \\
&=(I-HK_i)\left[e_{z,i|i-1}e_{z,i|i-1}^T(I-K_iH)^T+HP_{i|i-1}H^T\right]
\end{aligned}
\tag{3.2.47}
$$

Then the equivalent estimate values of R is obtained as

$$
\hat{R}_k=\frac{1}{k}\sum_{i=1}^{k}(I-K_iH)\left[e_{z,i|i-1}e_{z,i|i-1}^T(I-K_iH)^T+HP_{i|i-1}H^T\right] \tag{3.2.48}
$$

Note that Equation (3.2.43), Equation (3.2.44), Equation (3.2.45), and Equation (3.2.48) represent the properties of unbiased recursive noise statistic of Equation (4.2.24) – Equation (4.2.27) , respectively. Therefore, according to Equation (3.2.43), Equation (3.2.44), Equation (3.2.45), and Equation (3.2.48), the time-varying noise properties can be derived as follows

$$
\hat{q}_k=\hat{q}+\frac{1}{k}(K_ke_{z,k|k-1})] \tag{3.2.49}
$$

$$
\hat{r}_k=\hat{r}+\frac{1}{k}\left[(i-HK_i)e_{z,k|k-1}\right] \tag{3.2.50}
$$

$$
\hat{Q}_k=\frac{k-1}{k}\hat{Q}+\frac{1}{k}[K_ke_{z,k|k-1}e_{z,k|k-1}^TK_k^T+FP_{k-1|k-1}F^T-P_{k|k}] \tag{3.2.51}
$$

$$
\hat{R}_k=\frac{k-1}{k}\hat{R}+\frac{1}{k}(I-HK_k)\times[e_{z,k|k-1}e_{z,k|k-1}^T(I-HK_i)^T+HP_{k|k-1}H^T] \tag{3.2.52}
$$

Where $\hat{q}$,$\hat{r}$,$\hat{Q}$, and $\hat{R}$ are $\hat{q}_{k-1}$,$\hat{r}_{k-1}$,$\hat{Q}_{k-1}$, and $\hat{R}_{k-1}$, respectively. Therefore by applying the weighted exponent[14], [69], [102] , where the weighting coefficient d_k is formulated to replace the exponential $\frac{1}{k}$, then the alternative formulations of Equation (3.2.49) – Equation (3.2.52) become as follows

$$
\hat{q}_k=\hat{q}_{k-1}+d_k(K_ke_{z,k|k-1}) \tag{3.2.53}
$$

$$\hat{r}_k = \hat{r}_{k-1} + d_k[(I - HK_k)e_{z,k|k-1}] \tag{3.2.54}$$

$$\hat{Q}_k = (1 - d_k)Q_{k-1} + d_k[K_k e_{z,k|k-1} e^T_{z,k|k-1} \hat{} K^T_k + FP_{k-1|k-1}F^T - P_{k|k}] \tag{3.2.55}$$

$$\hat{R}_k = (1 - d_k)\hat{R}_{k-1|k-1} + d_k(I - HK_k)[e_{z,k|k-1} e^T_{z,k|k-1}(I - HK_k)^T] + HP_{k|k-1}H^T \tag{3.2.56}$$

the weighting exponent is expressed as follows

$$\begin{cases} \beta_i = d_k \beta^{i-1} \\ d_k = \frac{1-b}{1-b^k} \end{cases}, for\ i = 1, , n \tag{3.2.57}$$

where b is a fading factor satisfied $0 < b < 1$ and β_i is the i-th weighting factor defined as $\beta_i = \beta_{i-1} b^k$ and satisfied $\sum_{i=1}^{k} \beta_i$. Next, to prevent the occurrence of a filter divergence, the covariance correction based on divergent suppression concept was applied. First, by referring to the covariance matching creation, the convergence condition can be described as follows

$$v^T_k v_k \geq S.tr[E(v_k v^T_k)] \tag{3.2.58}$$

where S is an adjustable coefficient presetting that satisfied $(S \geq 1)$ and v_k refers to other forms of innovation sequence that is $v_k = z_k - h(\hat{x}_{k|k-1})$. The main point of this process is correcting the error covariance matrix $P_{k|k-1}$ when the convergence condition above is not satisfied. The contrary, Equation (3.2.53) – Equation (3.2.56) will directly be used in this proposed method. Mathematically, this analogy can be described as follows.

$$\begin{cases} v^T_k v_k & , \quad k = 1 \\ \dfrac{\rho C_{0,k}}{v^T_k v_k} & , \quad k > 1 \end{cases} \tag{3.2.59}$$

$$P_{k|k-1} = \lambda_k \dot{P}_{k|k-1} \tag{3.2.60}$$

Where λ_k is well-known as the adaptive weighting coefficient which is calculated based on the fading factor formula [74], [97] as described as follows

$$\lambda_0 = tr\frac{[N_k]}{tr[M_k]} \tag{3.2.61}$$

$$\lambda_0 = \begin{cases} \lambda_0 \ , & \lambda_0 \geq 1 \\ 1 \ , & \lambda_0 < 1 \end{cases} \tag{3.2.62}$$

where N and M refers to the following equation

$$N_k = tr(C_{0,k} - R)^T \tag{3.2.63}$$

$$M_k = tr(P_{k|k-1}) \tag{3.2.64}$$

Where $tr(.)$ is representation of a matrix trace operator and ρ is forgetful factor satisfying $0 < \rho \geq 1$ commonly adjusted to 0.95. Note that by increasing this factor will create a smaller proportion of the information before time k[14]. It causes the residual vector effect to become prominent so that the ability of filter tracking increase. Up to this point, then the Adaptive EKF can be graphically summarized as follows.

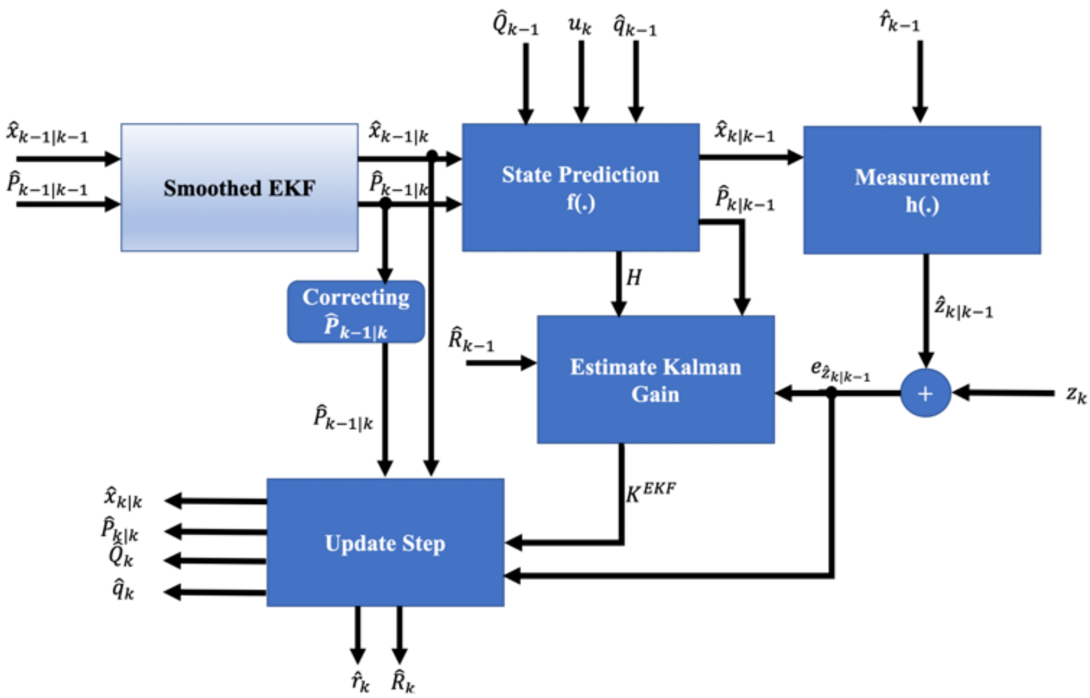


Figure 3.2 Working Principle of Adaptive EKF Based on MAP - Weighted Exponent

3.3 Designing Adaptive Smooth Variable Structure Filter Based on Maximum A Posterior Estimation and Weighted Exponent

Like Extended Kalman Filter, the Smooth Variable Structure Filter traditionally has no ability to update the noise statistic of the process and measurement recursively. For this reason, the modification approach is also recommended to be concerned before using it. As proposed in this dissertation, this modification is intended to effectively and responsively update the predefined noise statistic parameters. It leads to the definition of an adaptive filtering method. In this thesis, the adaptive approach based on a batch estimation of parameters is also conducted as an effort to improve the capability of SVSF for estimating purpose. In order to equip SVSF with an ability to recursively provide the responsive noise statistic, the maximum a posterior and weighted exponent are used in this experiment. Firstly, SVSF is mathematically derived using a maximum posterior to get the suboptimal solution of the time-varying noise statistic. Due to the lack of multi-step smoothing values on the estimated variables, the SVSF is firstly modified. The discrete index of some complicated estimate values seems to be unobservable. Therefore, as the purpose of continuing the derivation process, the simplification is conducted. However, the modification and simplification might degrade the quality of the adaptive SVSF or even lead to the divergence condition. Consequently, the new adaptive form of SVSF has the risk of being unstable, having bias conditions, and inaccurate solutions. It is because of the presence of a nonpositive definite matrix for covariances of noise statistics either to the process or measurement. For this reason, the unbiased estimation is also involved. It is used to guarantee the solution given by the adaption process. Besides that, the divergence suppression method is also concerned to guarantee that the adaptive solution under this approach has high convergence. This suppression method is conducted after performing the weighted exponent used to more derive the suboptimal solution given by maximum a posterior as the final effort to keep the stability, robustness, and effectiveness of SVSF as the robust estimation method. The detailed process of this adaption is presented in this dissertation as follows.

Firstly, it is assumed that the dynamic system model used in this process can be recalled from the Equations (3.1). It is completed with the characteristic of the noise statistic for the process and measurement as well as their corresponding covariances as expressed by Equation (3.2) – Equation (3.4). Secondly, in order to ease the mathematical derivation, the summary of the traditional form of SVSF is represented as follows

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \tag{3.3.1}$$

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q_k \tag{3.3.2}$$

$$\hat{z}_{k|k-1} = h(\hat{x}_{k|k-1}) \tag{3.3.3}$$

$$e_{z,k|k-1} = z_k - \hat{z}_{k|k-1} \tag{3.3.4}$$

$$S_k = HP_{k|k-1}H^T + R_k \tag{3.3.5}$$

$$A = \Big(|e_{z,k|k-1}|_{abs} + \gamma |e_{z,k-1|k-1}|_{abs}\Big) \tag{3.3.6}$$

$$\psi = \left(\overline{\overline{A}}^{-1} HP_{k|k-1}H^T S_k^{-1}\right)^{-1} \tag{3.3.7}$$

$$sat\Big(\frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i}\Big) = \begin{cases} +1 & \\ \frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i} & if \quad -1 < \frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i} < 1 \\ -1 & \end{cases} \quad \begin{matrix} \frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i} \geq 1 \\ \\ \frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i} \leq -1 \end{matrix} \tag{3.3.8}$$

$$K_k^{SVSF} = H^+ \left\{ \overline{\overline{A}} \circ sat\left(\frac{\overline{\overline{e_{z_k|k-1}}}}{\psi_i}\right) \right\} \overline{\overline{e_{z,k|k-1}}}^{-1} \tag{3.3.9}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k^{SVSF} \tag{3.3.10}$$

$$P_{k|k} = (I - K_k^{SVSF}H)P_{k|k-1}(I - K_k^{SVSF}H)^T + K_k^{SVSF}R_k K_k^{SVSF\,T} \tag{3.3.11}$$

$$\hat{z}_{k|k} = h(\hat{x}_{k|k}) \tag{3.3.12}$$

$$e_{z,k|k} = z_k - \hat{z}_{k|k} \tag{3.3.13}$$

$$|e_{z,k|k}|_{abs} < |e_{k-1|k-1}|_{abs} \tag{3.3.14}$$

Note that all the representative equation above are the original form of SVSF for one-cycle of working. It is assumed that the noise statistic is predefined and kept to be constant for the whole estimation process. This form aims to reduce the estimation error, which is the different values of the predicted and real measurement. Accordingly, the innovation or error measurement is then used to generate the corrective gain by using the principle of the sliding model concept. And as the final stage of SVSF, the corrective gain is used to update the state value and its covariance and posteriori error measurement, which is also predefined at the beginning. However, in the real application, it is impossible to define those parameters accurately by means they are partially known or even unknown. It might degrade the filtering performance. For these reasons, the adaptive SVSF algorithm is mainly concerned. The process was initially started by reconsidering the prior information of the nonlinear dynamic system is modeled as described in Equation (3.1). Secondly, it is considered that the process noise ω_k, measurement noise v_k and initial state vector x_0 are assumed to be mutually uncorrelated for any discrete time index j or k, then the mean $E[\cdot]$ and covariance $Cov[.]$ of the process and measurement noise can be redefined in order to clearly analogize the identity of the system.

$$\begin{cases} E[\omega_k] = q_k, Cov[\omega_k, \omega_j] = Q_k \delta_{kj} \\ E[v_k] = r_k, Cov[v_k, v_j] = R_k \delta_{kj} \\ Cov[\omega_k, v_j] = 0 \end{cases} \tag{3.3.15}$$

where δ refers to Kronecker delta function. The prior information above is initially assumed to be not equals to zero. Additionally, Q_k and R_k are positive definite symmetric matrices, then the MAP estimates of q_k,r_k,Q_k, R_k, and X_k can be obtained by calculating the maximum value of the following conditional probability density function

$$L = p[X_k, q, r, Q, R|Z_K] \tag{3.3.16}$$

where $X_k = [x_1, x_2, \ldots, x_k]$ and $Z_k = [z_1, z_2, \ldots, z_k]$. Next, applying the Bayes rule and referring to the property of the conditional probability, where L is proportional to $p[X_k, q, r, Q, R, Z_k]/p[Z_k]$. Therefore, since its marginal likelihood $p[Z_k]$ plays no role in the optimization, then

$$L = p[Z_k|X_k, q, r, Q, R]p[X_k|q, r, Q, R]p[q, r, Q, R] \tag{3.3.17}$$

where $p[q, r, Q, R]$ is regarded as the constants obtained from the prior information. Then a posteriori distribution $p[X_k, q, r, Q, R|Z_k]$ can be calculated by multiplying $p[Z_k|q, r, Q, R]p[Z_k|q, r, Q, R]$ with $p[X_k, q, r, Q, R|Z_k]$ as derived below

$$p[X_k|q, r, Q, R] = p[x_0]\prod_{i=1}^{k} p[x_i|x_{i-1}, q, R] \tag{3.3.18}$$

which can be derived and expanded as follows

$$\begin{aligned}
p[x_0]\prod_{i=1}^{k} p[x_i|x_{i-1}, q, R] =& \frac{1}{2\pi^{\frac{m}{2}}|P_0|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}\|x_0 - \hat{x}_0\|^2_{P_0^{-1}} \right] \prod_{i=1}^{k} \frac{1}{2\pi^{\frac{m}{2}}|Q|^2} \exp \\
& -\frac{1}{2}\|x_i - f(x_{i-1}) - q\|^2_{Q^{-1}} \Big] \\
=& \frac{1}{2\pi^{\frac{m}{2}}|P_0|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}\|x_0 - \hat{x}_0\|^2_{P_0^{-1}} \times \frac{1}{2\pi^{\frac{mk}{2}}|Q|^{\frac{k}{2}}} \exp \right[\\
& -\frac{1}{2}\sum_{i=1}^{k}\|x_i - f(x_{i-1}) - q\|^2_{Q^{-1}} \Big] \\
=& \frac{1}{2\pi^{\frac{m}{2}}2\pi^{\frac{mk}{2}}}|P_0|^{-\frac{1}{2}}|Q|^{-\frac{k}{2}} \exp\{- \\
& \frac{1}{2}\left[\|x_0 - \hat{x}_0\|^2_{P_0^{-1}} + \sum_{i=1}^{k}\|x_i - f(x_{i-1}) - q\|^2_{Q^{-1}} \right]\} \\
=& \frac{1}{2\pi^{\frac{mk+m}{2}}}|P_0|^{-\frac{1}{2}}|Q|^{-\frac{k}{2}} \exp\{ \\
& -\frac{1}{2}\left[\|x_0 - \hat{x}_0\|^2_{P_0^{-1}} + \sum_{i=1}^{k}\|x_i - f(x_{i-1}) - q\|^2_{Q^{-1}} \right]\}
\end{aligned} \tag{3.3.19}$$

$$p[Z_k|X_k, q, r, Q, R] = \prod_{i=1}^{k} p[z_i|x_i, r, R] \tag{3.3.20}$$

which can be derived and expanded as follows

$$\begin{aligned}\prod_{i=1}^{k} p[z_i|x_i,r,R] =& \prod_{i=1}^{k} \frac{1}{2\pi^{\frac{n}{2}}|R|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}\left\|z_i-h(x_i)-r\right\|_{R^{-1}}^{2}\right] \\ =& \frac{1}{2\pi^{\frac{n}{2}}}|R|^{-\frac{1}{2}} \exp-\frac{1}{2}\left\|z_i-h(x_i)-r\right\|_{R^{-1}}^{2}\Big]\end{aligned} \tag{3.3.21}$$

Therefore, by substituting both Equation (3.3.18) and (3.3.20) into Equation (3.3.17), it yields

$$\begin{aligned}p[Z_k|X_k,q,r,Q,R]p[X_k|q,r,Q,R]p[q,r,Q,R] =& \frac{1}{2\pi^{\frac{n}{2}}}|R|^{-\frac{1}{2}} \exp-\frac{1}{2}\left\|z_i-h(x_i)-\right. \\ & \left. r\right\|_{R^{-1}}^{2}\Big] \times \frac{1}{2\pi^{\frac{mk+m}{2}}}|P_0|^{-\frac{1}{2}}|Q|^{-\frac{k}{2}} \exp\{-\frac{1}{2}\Big[\left\|x_0-\hat{x}_0\right\|_{P_0^{-1}}^{2} \\ & +\sum_{i=1}^{k}\left\|x_i-f(x_{i-1})-q\right\|_{Q^{-1}}^{2}\Big]\} \times p[q,r,Q,R]\end{aligned} \tag{3.3.22}$$

Now supposing that

$$C = \frac{1}{2\pi^{-\frac{nk}{2}}2\pi^{\frac{mk+m}{2}}}|P_0|^{-\frac{1}{2}} \exp\left[\left\|x_0-\hat{x}_0\right\|_{P_0^{-1}}^{2}\right] \times p[q,r,Q,R] \tag{3.3.23}$$

referring Equation (3.3.23) that the constants are obtained from the initially predefined parameters, there Equation (3.3.21) can be compactly simplified as follows

$$\begin{aligned}L =& C|R|^{-\frac{k}{2}}|Q|^{-\frac{k}{2}} \exp\{-\frac{1}{2}\Big[\sum_{i=1}^{k}\left\|z_i-h(x_i)-r\right\|_{R^{-1}}^{2}+\sum_{i=1}^{k}\left\|x_i-f(x_{i-1})-\right. \\ & \left. q\right\|_{Q^{-1}}^{2}\Big]\}\end{aligned} \tag{3.3.24}$$

Furthermore, to find the maximized parameter of the posterior distribution; firstly, taking a logarithm as the monotonic function to simplify the calculation; secondly, find the first derivative of L with respect to the q_k,r_k,Q_k, and R_k, separately; then finally ended by equating it with zero. These steps are organized and derived as follows

$$J = ln(L) = ln(C) - \frac{k}{2}ln|R| - \frac{k}{2}ln|Q| - \frac{1}{2}\left[\sum_{i=1}^{k}\|z_i - h(x_i) - r\|_{R}^{2}{}^{-1} + \sum_{i=1}^{k}\|x_i - f(x_{i-1}) - q\|_{Q}^{2}{}^{-1}\right] \tag{3.3.25}$$

Then $\hat{q}_k$, $\hat{r}_k$, $\hat{Q}_k$ and $\hat{R}_k$ are respectively presented as follows

$$\hat{q}_k = \frac{\partial J}{\partial q} = \frac{1}{k}\sum_{i=1}^{k}\hat{x}_{i|k} - f(\hat{x}_{i-1|k}) \tag{3.3.26}$$

$$\hat{r}_k = \frac{\partial J}{\partial r} = \frac{1}{k}\sum_{i=1}^{k}\hat{z}_i - h(\hat{x}_{i|k}) \tag{3.3.27}$$

$$\hat{Q}_k = \frac{\partial J}{\partial Q} = \frac{1}{k}\sum_{i=1}^{k}\left(\hat{x}_{i|k} - f(\hat{x}_{i-1|k}) - q\right)\left(\hat{x}_{i|k} - f(\hat{x}_{i-1|k}) - q\right)^T \tag{3.3.28}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}\left(\hat{z}_i - h(\hat{x}_{i|k}) - r\right)\left(\hat{z}_i - h(\hat{x}_{i|k}) - r\right)^T \tag{3.3.29}$$

The complicated multi-step smoothing term $\hat{x}_{i|k}$ and $\hat{x}_{i-1|k}$ in Equation (3.3.26) – Equation (3.3.29) might cause inefficiency of the MAP estimate, therefore to find the conventional and efficient recursive form the simplification is needed. Note that the recursive update process only utilizes the estimate value at time $k-1$ and k, thus the simplification can be conducted by replacing $\hat{x}_{i-1|k}$ with $\hat{x}_{i-1|i}$ in Equation (3.3.26) and Equation (3.3.28) and $\hat{x}_{i|k}$ with $\hat{x}_{i|i}$ in Equation (3.3.26) – Equation (3.3.29). Therefore, the suboptimal of MAP noise estimator can be expressed as follows

$$\hat{q}_k = \frac{\partial J}{\partial q} = \frac{1}{k}\sum_{i=1}^{k}\hat{x}_{i|i} - f(\hat{x}_{i-1|i}) \tag{3.3.30}$$

$$\hat{r}_k = \frac{\partial J}{\partial r} = \frac{1}{k}\sum_{i=1}^{k}\hat{z}_i - h(\hat{x}_{i|i}) \tag{3.3.31}$$

$$\hat{Q}_k = \frac{\partial J}{\partial Q} = \frac{1}{k}\sum_{i=1}^{k}\left(\hat{x}_{i|i} - f(\hat{x}_{i-1|i}) - q\right)\left(\hat{x}_{i|i} - f(\hat{x}_{i-1|i}) - q\right)^T \tag{3.3.32}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}\left(\hat{z}_i - h(\hat{x}_{i|i}) - r\right)\left(\hat{z}_i - h(\hat{x}_{i|i}) - r\right)^T \tag{3.3.33}$$

As can be analyzed from the sequence equations above that the estimate value of $\hat{x}_{i-1|i}$ is not provided by the SVSF. Therefore, modifying the original form of SVSF is needed to execute the noise statistics estimator effectively. The process of modifying the SVSF can be done by calculating the one-step smoothing of the SVSF gain and its corresponding estimate value using the fixed-point smoothing algorithm[41], [63], [67]. (see Appendix B)

According to Appendix B, the estimate value $\hat{x}_{i|i}$ and $\hat{x}_{i-1|i}$ can be adopted from Equation (B.10) and Equation (B.19), respectively. Next, to guarantee that the recursive process and measurement noise statistics are unbiased, the modified suboptimal MAP noise estimators are then derived refer to unbiased estimation.

First, by substituting Equation (B.11) into Equation (B.19), the general term $\hat{x}_{i|i} - f(\hat{x}_{i|i-1})$ in Equation (3.3.30) and Equation (3.3.32) can be rewritten as follows

$$\hat{x}_{i|i} - f(\hat{x}_{i|i-1}) = K_i^{SVSF} e_{z,i|i-1} + q_{i-1} \tag{3.3.34}$$

Similarly, replacing $\hat{x}_{i|i}$ with Equation (3.3.52), the general term $z_i - h(\hat{x}_{i|i})$ in Equation (3.3.31) and Equation (3.3.33) can be written as follows

$$z_i - h(\hat{x}_{i|i}) = z_i - h(\hat{x}_{i|i-1} + K_i^{SVSF} e_{z,i|i-1}) \tag{3.3.35}$$

and the suboptimal MAP estimation in Equation (3.3.30) – Equation (3.3.33) can be rearranged as follows

$$\hat{q}_k = \frac{1}{k}\sum_{i=1}^{k} K_i^{SVSF} e_{z,i|i-1} + q_{i-1} \tag{3.3.36}$$

$$\hat{r}_k = \frac{1}{k}\sum_{i=1}^{k}(I - HK_i^{SVSF}) e_{z,i|i-1} + r_i \tag{3.3.37}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k} K_i^{SVSF} e_{z,i|i-1} e_{z,i|i-1}^T K_i^{SVSFT} \tag{3.3.38}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}(I - HK_i^{SVSF})e_{z,i|i-1}e_{z,i|i-1}^T(I - HK_i^{SVSF})^T \tag{3.3.39}$$

Since the innovation $e_{z,i|i-1}$ and its covariance $e_{z,i|i-1}e_{z,i|i-1}^T$ are contained in the process and measurement noise estimator, therefore

$$e_{z,i|i-1} = h(\tilde{x}_{i|i-1}) + v_i - r_i \tag{3.3.40}$$

and referring to state error covariance in Equation (B.20), it is obvious to have

$$\begin{aligned} K_i^{SVSF}e_{z,i|i-1}e_{z,i|i-1}^T K_i^{SVSFT} =& P_{i|i} - P_{i|i-1} + P_{i|i-1}H^T K_i^{SVSFT} \\ & HK_i^{SVSF}P_{i|i-1} \end{aligned} \tag{3.3.41}$$

Next, considering that the expectations $e_{z,i|i-1}$ and $e_{z,i|i-1}e_{z,i|i-1}^T$, therefore,

$$E[e_{z,i|i-1}] = 0 \tag{3.3.42}$$

$$E[e_{z,i|i-1}e_{z,i|i-1}^T] = HP_{i|i-1}H^T + R_i \tag{3.3.43}$$

Then by substituting Equation (3.3.40) – Equation (3.3.43), it is clear to have

$$E[\hat{q}_k] = q_k \tag{3.3.44}$$

$$E[\hat{r}_k] = r_k \tag{3.3.45}$$

$$\begin{aligned} E[\hat{Q}_k] =& - Q_k + E\left[\frac{1}{k}\sum_{i=1}^{k} P_{i|i} + P_{i|i-1}H^T K_i^{SVSFT} + HK_i^{SVSF}P_{i|i-1} - \right. \\ & \left. FP_{i-1|i-1}F^T \right] \end{aligned} \tag{3.3.46}$$

$$\begin{aligned} E[\hat{R}_k] =& R_k + E\left[\frac{1}{k}\sum_{i=1}^{k} HK_i^{SVSF}e_{z,i|i-1}e_{z,i|i-1}^T H^T K_i^{SVSFT} - \right. \\ & \left. HK_i^{SVSF}e_{z,i|i-1}e_{z,i|i-1}^T - e_{z,i|i-1}e_{z,i|i-1}^T H^T K_i^{SVSFT} HP_{i|i-1}H^T \right] \end{aligned} \tag{3.3.47}$$

Note that q_k, r_k, Q_k, and R_k are the representation of the suboptimal MAP estimation in Equation (3.3.36) – Equation (3.3.39), thus the unbiased MAP estimation can be summarized as follows

$$\hat{q}_k = \frac{1}{k}\sum_{i=1}^{k} K_i^{SVSF} e_{z,i|i-1} + q_{i-1} \tag{3.3.48}$$

$$\hat{r}_k = \frac{1}{k}\sum_{i=1}^{k}(I - HK_k^{SVSF})e_{z,i|i-1} + r_i \tag{3.3.49}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k} P_{i|i} - FP_{i-1|i-1}F^T \tag{3.3.50}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}\left[2\left((I - HK_i^{SVSF})e_{z,i|i-1}e_{z,i|i-1}^T(I - HK_i^{SVSF})^T\right) - e_{z,i|i-1}e_{z,i|i-1}^T \right. \\ HP_{i|i-1}H^T \tag{3.3.51}$$

The time-varying noise estimator is proposed. According to the unbiased suboptimal MAP estimator calculated above, the time-varying unbiased noise estimator are derived as follows

$$\hat{q}_k = \hat{q}_{k-1} + \frac{1}{k}(K_k^{SVSF} e_{z,k|k-1}) \tag{3.3.52}$$

$$\hat{r}_k = \hat{r}_{k-1} + \frac{1}{k}\left[(I - HK_k^{SVSF})e_{z,k|k-1}\right] \tag{3.3.53}$$

$$\hat{Q}_k = \frac{k-1}{k}\hat{Q}_{k-1} + \frac{1}{k}\left[P_{k|k-1} - FP_{k-1|k-1}F^T\right] \tag{3.3.54}$$

$$\hat{R}_k = \frac{k-1}{k}\hat{R}_{k-1} + \frac{1}{k}\left[2\left((I - HK_k^{SVSF})e_{z,k|k-1}e_{z,k|k-1}^T(I - HK_i^{SVSF})^T\right) - \right. \\ e_{z,k|k-1}e_{z,k|k-1}^T + HP_{k|k-1}H^T \tag{3.3.55}$$

the following alternative forms are regarded as the modified form of the time-varying unbiased noise statistics estimator Equation (3.3.52) – Equation (3.3.55). This method refers to the fading memory weighted exponent method and then by utilizing the weighting coefficient d_k to replace the exponential $1/k$.

$$\hat{q}_k = \hat{q}_{k-1} + d_k(K_k^{SVSF} e_{z,k|k-1}) \tag{3.3.56}$$

$$\hat{r}_k = \hat{r}_{k-1} + d_k\left[(I - HK_k^{SVSF})e_{z,k|k-1}\right] \tag{3.3.57}$$

$$\hat{Q}_k = (1-d_k)\hat{Q}_{k-1} + d_k\left[P_{k|k-1} - FP_{k-1|k-1}F^T\right] \tag{3.3.58}$$

$$\begin{aligned}\hat{R}_k =&(1-d_k)\hat{R}_{k-1} + d_k\left[2\left((I-HK_k^{SVSF})e_{z,k|k-1}e_{z,k|k-1}^T(I-HK_k^{SVSF})^T\right) - \right.\\ &e_{z,k|k-1}e_{z,k|k-1}^T + HP_{k|k-1}H^T\end{aligned} \tag{3.3.59}$$

where the weighting coefficient d_k can be calculated as follows

$$\begin{cases}\beta_i = d_k b^{i-1} \ , \ i=1,,n \\ d_k = \frac{1-b}{1-b^k}\end{cases} \tag{3.3.60}$$

where b is fading factor, which is satisfied $0 < b < 1$ and β_i is the i-th weighting factor, which is defined as $\beta_i = \beta_{i-1}b$ and satisfied $\sum_{i=1}^{k}$. Next, to prevent the occurrence of filter divergence, the covariance correction step based on the divergence suppression concept [7] is involved. First, the convergence condition is derived referring to the covariance matching creation as described below

$$v_k^T v_k \le S.tr[E(v_k^T v_k)] \tag{3.3.61}$$

where S is an adjustable coefficient presetting which is satisfied ($S \ge 1$) and the residual sequence $v_k = z_k - h(\hat{x}_{k|k-1})$. By executing Equation (3.3.60) if the convergence condition Equation (4.3.61) is satisfied, Equation (3.3.56) – Equation (3.3.59) are applied otherwise the correction method of the covariance $P_{k|k-1}$ is suggested against the divergence occurrence.

$$P_{k|k-1} = \lambda_k \dot{P}_{k|k-1} \tag{3.3.62}$$

where λ_k is known as the adaptive weighting coefficient which is calculated based on the fading factor formula [27,28] as summarized as follows

$$C_{0,k} = \begin{cases} v_k^T v_k, k=1 \\ \frac{\rho C_{0,k} + v_k^T v_k}{1+\rho}, k>1\end{cases} \tag{3.3.63}$$

$$N_k = tr(C_{0,k} - R)^T \tag{3.3.64}$$

$$M_k = tr(P_{k|k-1}) \tag{3.3.65}$$

$$\lambda_0 = tr\frac{[N_k]}{tr[M_k]} \tag{3.3.66}$$

$$\lambda_k = \begin{cases} \lambda_0, \lambda_0 \geq 1 \\ 1, \lambda_0 < 1 \end{cases} \tag{3.3.67}$$

where $tr(\cdot$ refers to the matrix trace, ρ is the forgetful factor satisfied (typically to be set 0.95). Note that increasing the factor will create a smaller proportion of the information before time k. It causes the residual vector effect to become prominent, thus consequently the filter tracking ability will increase. Finally, the summary of Adaptive SVSF can be presented by the following flowchart.

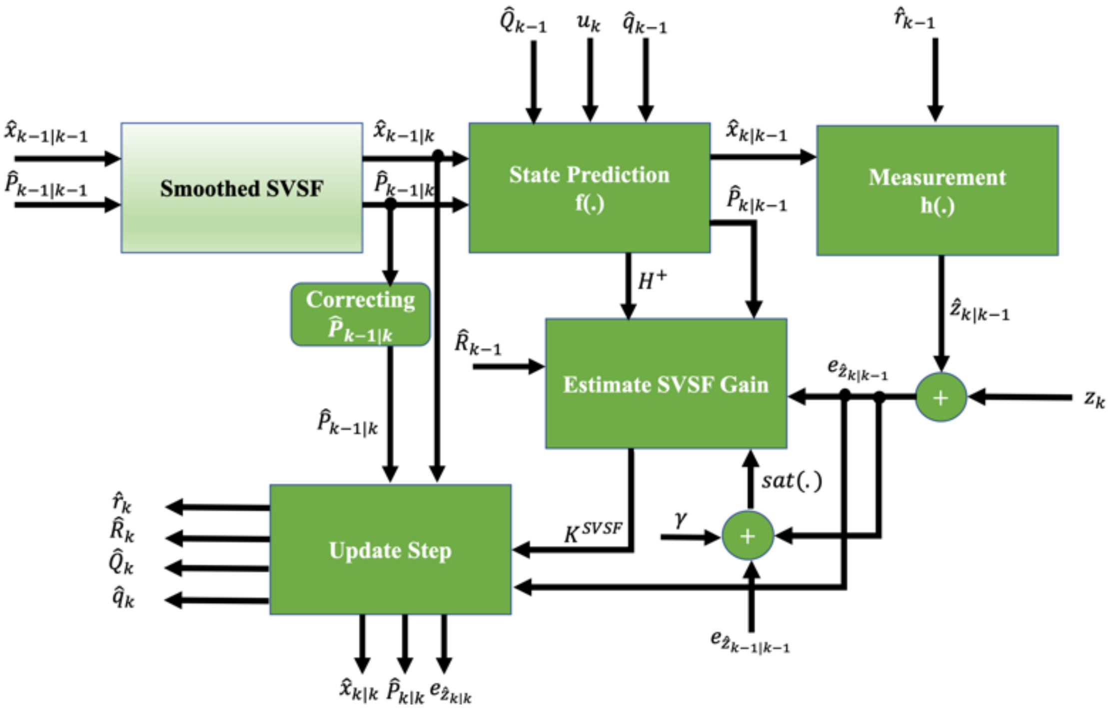


Figure 3.3 Working Principle of Adaptive SVSF Based on MAP and Weighted Exponent

3.4 Designing Adaptive Smooth Variable Structure Filter Based on Maximum Likelihood Estimation and Expectation-Maximization Principle

For the second proposed method, the maximum likelihood estimation and expectation-maximization are used as the batch estimation of parameters-based adaptive filtering. Likely, the purpose of this method is to equip the Smooth Variable Structure Filter with the recursive noise formulation corresponding to its traditional formulation. The main objective of this approach is to let the SVSF having the ability to estimate noise statistic and their corresponding covariance recursively.

Assuming that $\theta = (q, r, Q, R)$ represents the unknown noise statistic of the nonlinear system described by Equation (3.1). Then its estimated value $\hat{\theta}$ can be obtained by utilizing MLE as expressed as follows

$$\hat{\theta}^{ML} = \arg\max_{\theta} \left\{ \ln \left[L(q, r, Q, R | Z_k, X_k) \right] \right\} \tag{3.4.1}$$

where $L(q, r, Q, R|Z_k, X_k)$ is the likelihood function of θ. It can be expressed below

$$L(q, r, Q, R|Z_k, X_k) = p(Z_k, X_k|q, r, Q, R) = p(X_k|q, Q, r, R)p(Z_k|X_k, q, r, Q, Rr) \tag{3.4.2}$$

For $X_k = [x_0, ..., x_k]$ and $Z_k = [z_1, ..., z_k]$. Since Equation (3.1) is the first-order of Markov process then Equation (4.4.2) can be factorized as follows

$$p(Z_k, X_k|\theta) = p[x_0] \prod_{i=1}^{k} p[x_i|x_{i-1}, q, Q] \prod_{i=1}^{k} p[z_i|x_i, r, R] \tag{3.4.3}$$

Then by considering that these knowledges comply with Gaussian distribution, then Equation (3.4.3) can be rewritten as follows

$$p(Z_k,X_k|q,r,Q,R) = \frac{1}{(2\pi)^{-\frac{k(n+m)+n}{2}}}|P_0|^{-\frac{1}{2}}|Q|^{-\frac{k}{2}}|R|^{-\frac{k}{2}} \times \exp\left\{-\frac{1}{2}\left[\|x_0 - \hat{x}_0\|^2_{P_0^{-1}} + \sum_{i=1}^{k}\|x_i - f(x_{i-1}) - q\|^2_{Q^{-1}} + \sum_{i=1}^{k}\|z_i - h(x_i) - r\|^2_{R^{-1}}\right]\right\} \tag{3.4.4}$$

taking logarithm of Equation (3.4.4), it yields

$$\ln[L(q,r,Q,R|Z_k,X_k)] = -\frac{k(n+m)+n}{2}\ln(2\pi) - \frac{1}{2}\ln(|P_0|) - \frac{k}{2}\ln(|R|) - \frac{k}{2}\ln(|Q|) - \frac{1}{2}\|x_0 - \hat{x}_0\|^2_{P_0^{-1}} - \frac{1}{2}\left[\sum_{i=1}^{k}\|x_i - f(x_{i-1}) - q\|^2_{Q^{-1}} + \sum_{i=1}^{k}\|z_i - h(x_i) - r\|^2_{R^{-1}}\right] \tag{3.4.5}$$

In this experiment, the role of EM estimation is utilized to solve the derived MLE solution above. It aims to approximate both process and measurement noise statistic. Basically, there exist two main steps in EM namely expectation-based and maximization-based solution[8], [57], [60], [64], [71], [72]. Both are conducted sequentially. Its process can be described as follows

Under Expectation-Based Solution (E-Step)

The expectation process can be done by first taking the conditional expectation and sequentially equating the result to zero as shown below

$$E\left[\ln[L(q,r,Q,R|Z_k,X_k)]\right] = -\frac{k(n+m)+n}{2}\ln(2\pi) - \frac{1}{2}\ln(|P_0|) - \frac{1}{2}E\left[\|x_0 - \hat{x}_0\|^2_{P_0^{-1}}\right] - \frac{k}{2}\ln(|R|) - \frac{k}{2}\ln(|Q|) - \frac{1}{2}E\left[\sum_{i=1}^{k}\|x_i - f(x_{i-1}) - q\|^2_{Q^{-1}} + \sum_{i=1}^{k}\|z_i - h(x_i) - r\|^2_{R^{-1}}\right] \tag{3.4.6}$$

Suppose that

$$C = -\frac{k(n+m)+n}{2}ln(2\pi) - \frac{1}{2}ln(|P_0|) - \frac{1}{2}E\left[\|x_0 - \hat{x}_0\|^2_{P_0^{-1}}\right] \tag{3.4.7}$$

then Equation (3.4.6) can be re-expressed as follows

$$E\bigg[\ln[L(q,r,Q,R|Z_k,X_k)]\bigg] = C - \frac{k}{2}ln(|R|) - \frac{k}{2}ln(|Q|) - \frac{1}{2}E\Bigg[\sum_{i=1}^{k}\|x_i - f(x_{i-1}) - q\|^2_{Q^{-1}} + \sum_{i=1}^{k}\|z_i - h(x_i) - r\|^2_{R^{-1}}\Bigg] \tag{3.4.8}$$

By definition $\|a\|^2_{b^{-1}} = a^T b^{-1} a$. Then by applying the identity $tr(a^T a) = tr(aa^T)$, Equation (3.4.8) can be derived as follows

$$J = C - \frac{k}{2}\ln(|R|) - \frac{k}{2}\ln(|Q|) - \frac{1}{2}\sum_{i=1}^{k}E\Bigg\{tr\bigg[Q^{-1}(x_i - f(x_{i-1}) - q)(x_i - f(x_{i-1}) - q)^T\bigg]\Bigg\} - \frac{1}{2}\sum_{i=1}^{k}E\Bigg\{tr\bigg[R^{-1}(z_i - h(x_i) - r)(z_i - h(x_i) - r)^T\bigg]\Bigg\} \tag{3.4.9}$$

where J refers to an objective function of $E\big[ln[L(q,r,Q,R|Z_k,X_k)]\big]$.

Under Maximization-Based Solution (M-Step)

At this point the estimated noise statistic can be obtained by maximizing J. It can be done by taking partial derivative with respect to q, Q, r, R and equating to zero as can be calculated as follows

$$\frac{\partial J}{\partial q} = 0\,, \frac{\partial J}{\partial r} = 0\,, \frac{\partial J}{\partial Q} = 0\,, \frac{\partial J}{\partial R} = 0 \tag{3.4.10}$$

Since R and Q are positive definite matrix it can be described that

$$\begin{cases} |Q| = tr(Q) \\ |R| = tr(R) \end{cases} \tag{3.4.11}$$

For $|\cdot|$ and $tr(.)$ refer to the determinant and trace of matrix, respectively. Then, it is obtained

$$\hat{q}_k = \frac{1}{k}\sum_{i=1}^{k} x_{i|k} - f(x_{i-1|k}) \tag{3.4.12}$$

$$\hat{r}_k = \frac{1}{k}\sum_{i=1}^{k} z_{i|k} - h(x_{i|k}) \tag{3.4.13}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k}(x_{i|k} - f(x_{i-1|k}) - q)(x_i - f(x_{i-1|k}) - q)^T \tag{3.4.14}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}(z_{i|k} - h(x_{i|k}) - r)(z_{i|k} - h(x_{i|k}) - r)^T \tag{3.4.15}$$

The complicated multi-step of smoothing term $x_{i|k}$ and $x_{i-1|k}$ in Equation (3.4.12) – Equation (3.4.15) might cause inefficiency. For this reason, the simplification is approached at this point. It can be done by replacing $x_{i-1|k}$ with $x_{i-1|i}$ and $x_{i|k}$ with $x_{i|i}$.

$$\hat{q}_k = \frac{1}{k}\sum_{i=1}^{k} x_{i|i} - f(x_{i-1|i}) \tag{3.4.16}$$

$$\hat{r}_k = \frac{1}{k}\sum_{i=1}^{k} z_{i|i} - h(x_{i|i}) \tag{3.4.17}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k}(x_{i|i} - f(x_{i-1|i}) - q)(x_i - f(x_{i-1|i}) - q)^T \tag{3.4.18}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}(z_{i|i} - h(x_{i|i}) - r)(z_{i|i} - h(x_{i|i}) - r)^T \tag{3.4.19}$$

At this point, it is clear that $x_{i-1|i}$ is not provided by the classical SVSF. Hence, SVSF was modified/improved. The process of modifying SVSF was done by calculating the one-step smoothing of the SVSF gain and its corresponding estimate value using the fixed-point smoothing algorithm[41], [63], [67]. (see Appendix B)

According to Appendix B, the estimate value $x_{i|i}$ and $x_{i-1|i}$ can be adopted from Equation (B.10) and Equation (B.19), respectively. The simplification of the complicated multistep smoothing in Equation (3.4.16) – Equation (3.4.19) might reduce the optimal solution obtained by MLE-EM. For this reason, the recursive noise statistics Equation (3.4.16) – Equation (3.4.19) are derived as follows.

Substituting Equation (B.11) into Equation (B.19), the general term $x_{i|i} - f(x_{i-1|i})$ in Equation (3.4.16) and Equation (3.4.18) can be rewritten

$$x_{i|i} - f(\hat{x}_{i-1|i}) = K_i^{SVSF} e_{z,i|i-1} + q \tag{3.4.20}$$

Then substituting Equation (4.4.39) into $z_i - h(x_{i|i})$

$$z_{i|i} - h(x_{i|i}) = z_i - h(x_{i|i-1}) + K_i^{SVSF} e_{z,i|i-1} \tag{3.4.21}$$

then the equivalent form of Equation (3.4.16) – Equation (3.4.19) are

$$\hat{q}_k = \frac{1}{k}\sum_{i=1}^{k} K_i^{SVSF} e_{z,i|i-1} + q \tag{3.4.22}$$

$$\hat{r}_k = \frac{1}{k}\sum_{i=1}^{k} (I - HK_i^{SVSF}) e_{z,i|i-1} + r \tag{3.4.23}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k} (K_i^{SVSF} e_{z,i|i-1} e_{z,i|i-1}^T K_i^{SVSF^T}) \tag{3.4.24}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k} (I - HK_i^{SVSF}) e_{z,i|i-1} e_{z,i|i-1}^T (I - HK_i^{SVSF}) \tag{3.4.25}$$

Since the innovation $e_{z,i|i-1}$ and its covariance $e_{z,i|i-1}e_{z,i|i-1}^T$ are contained in the process and measurement noise estimator, therefore

$$e_{z,i|i-1} = h(\tilde{x}_{i|i-1}) + v_i - r \tag{3.4.26}$$

Referring to the state error covariance in Equation (B.20), then

$$K_i^{SVSF} e_{z,i|i-1} e_{z,i|i-1}^T K_i^{SVSF^T} = P_{i|i} - P_{i|i-1} + P_{i|i-1} H^T K_i^{SVSF^T} + HK_i^{SVSF} P_{i|i-1} \tag{3.4.27}$$

Next, assuming that their expectations are

$$E[e_{z,i|i-1}] = 0 \tag{3.4.28}$$

$$E[e_{z,i|i-1} e_{z,i|i-1}^T] = HP_{i|i-1}H^T + R_i \tag{3.4.29}$$

Then by substituting Equation (3.4.26) – Equation (3.4.29) into Equation (3.4.22) – Equation (3.4.25), the expectation of the noise statistics is

$$E[\hat{q}_k] = q_k \tag{3.4.30}$$

$$E[\hat{r}_k] = r_k \tag{3.4.31}$$

$$\begin{aligned} E[\hat{Q}_k] =& -Q_k + E\left[\frac{1}{k}\sum_{i=1}^{k}(P_{i|i} + P_{i|i-1}H^T K_i^{SVSF^T} + HK_i^{SVSF}P_{i|i-1} - \right. \\ & \left. FP_{i-1|i-1}F^T)\right] \end{aligned} \tag{3.4.32}$$

$$\begin{aligned} E[\hat{R}_k] =& R_k + E\left[\frac{1}{k}\sum_{i=1}^{k}(HK_i^{SVSF}e_{z,i}e_{z,i}^T H^T K_i^{SVSF^T} - HK_i^{SVSF}e_{z,i}e_{z,i}^T - \right. \\ & \left. e_{z,i}e_{z,i}^T H^T K_i^{SVSF^T} HP_{i|i-1}H^T)\right] \end{aligned} \tag{3.4.33}$$

Note that, q_k, r_k, Q_k, and R_k in Equation (3.4.30) – Equation (3.4.33) are the representation of Equation (3.4.22) – Equation (3.4.25), thus the unbiased noise statistic properties can be calculated as follows

$$\hat{q}_k = \frac{1}{k}\sum_{i=1}^{k}(K_i^{SVSF}e_{z,i}) + q \tag{3.4.34}$$

$$\hat{r}_k = \frac{1}{k}\sum_{i=1}^{k}(I - HK_i^{SVSF})e_{z,i} + r \tag{3.4.35}$$

$$\hat{Q}_k = \frac{1}{k}\sum_{i=1}^{k}(P_{i|i-1} - FP_{i-1|i-1}F^T) \tag{3.4.36}$$

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}\left[2\big((I - HK_i^{SVSF})e_{z,i}e_{z,i}^T(I - HK_i^{SVSF})^T\big) - e_{z,i}e_{z,i}^T + HP_{i|i-1}H^T\right] \tag{3.4.37}$$

Up to this point, some derived equations are presented in order to ease the adaption process. Considering that the original Joseph covariance is able to be derived then the alternative measurement noise statistic covariance is

$$P_{i|i} = (I - K_i^{SVSF} H) P_{i|i-1} (I - K_i^{SVSF} H)^T + K_i^{SVSF} R K_i^{SVSF^T}$$
$$= P_{i|i-1} - P_{i|i-1} H^T K_i^{SVSF^T} - K_i^{SVSF} H P_{i|i-1} + K_i^{SVSF} H P_{i|i-1} K_i^{SVSF^T} H^T$$
$$K_i^{SVSF} R K_i^{SVSF^T} \tag{3.4.38}$$

Note that the trace of matrix equals to the trace of its transpose. Therefore

$$tr[P_{i|i}] = tr[P_{i|i-1}] - 2tr[H K_i^{SVSF} P_{i|i-1}] + tr[K_i^{SVSF} (H P_{i|i-1} H^T + R) K_i^{SVSF^T}] \tag{3.4.39}$$

Differentiating with respect to K_i^{SVSF}, then

$$\frac{\partial\, tr[P_{i|i}]}{\partial\, K_i^{SVSF}} = -2[H P_{i|i-1}]^T + 2K_i^{SVSF} (H P_{i|i-1} H^T + R) \tag{3.4.40}$$

Equating to zero, it yields

$$\begin{aligned} [H P_{i|i-1}]^T &= K_i^{SVSF} (H P_{i|i-1} H^T + R) \\ K_i^{SVSF} &= [H P_{i|i-1}]^T (H P_{i|i} H^T + R)^{-1} \\ K_i^{SVSF} &= P_{i|i-1} H^T (H P_{i|i-1} H^T + R)^{-1} \end{aligned} \tag{3.4.41}$$

Substituting Equation (3.4.41) into Equation (3.4.38) it is obtained

$$\begin{aligned} P_{i|i} &= P_{i|i-1} - H K_i^{SVSF} P_{i|i-1} \\ P_{i|i} &= (I - K_i^{SVSF} H) P_{i|i-1} \end{aligned} \tag{3.4.42}$$

Suppose that the following equation is satisfied

$$e_{z,i|i-1} e_{z,i|i-1}^T = H P_{i|i-1} H^T + R \tag{3.4.43}$$

Then by substituting Equation (3.4.43) into Equation (3.4.41), it yields

$$K_i^{SVSF} e_{z,i|i-1} e_{z,i|i-1}^T = P_{i|i-1} H^T \tag{3.4.44}$$

Hence, the formulation the covariance matrix of measurement noise statistic R can be derived as follows

$$\hat{R}_k = \frac{1}{k} \sum_{i=1}^{k} (I - H K_i^{SVSF}) e_{z,i} e_{z,i}^T (I - H K_i^{SVSF})^T \tag{3.4.45}$$

where

$$(I-K_i^{SVSF}H)e_{z,i}e_{z,i}^T(I-K_i^{SVSF}H)^T = (HP_{i|i-1}H^T+R)-2HP_{i|i-1}H^T+ HK_i^{SVSF}HP_{i|i-1}H^T = R-(I-HK_i^{SVSF})HP_{i|i-1}H^T \tag{3.4.46}$$

$$\begin{aligned} E[\hat{R}_k] &= E\Big\{\frac{1}{k}\sum_{i=1}^{k}(I-HK_i^{SVSF})e_{z,i}e_{z,i}^T(I-HK_i^{SVSF})^T\Big\} \\ &= E\Big\{\frac{1}{k}\sum_{i=1}^{k}R-(I-HK_i^{SVSF})HP_{i|i-1}H^T\Big\} \\ &= R+E\Big\{\frac{1}{k}\sum_{i=1}^{k}-(HP_{i|i-1}H^T-HK_i^{SVSF}HP_{i|i-1}H^T)\Big\} \\ &= R+E\Big\{\frac{1}{k}\sum_{i=1}^{k}HK_i^{SVSF}HP_{i|i-1}H^T-HP_{i|i-1}H^T\Big\} \end{aligned} \tag{3.4.47}$$

Then it can be obtained

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}(I-HK_i^{SVSF})e_{z,i}e_{z,i}^T(I-HK_i^{SVSF})^T + HK_i^{SVSF}HP_{i|i-1}H^T - HP_{i|i-1}H^T \tag{3.4.48}$$

where

$$\begin{aligned} &(I-HK_i^{SVSF})e_{z,i}e_{z,i}^T(I-HK_i^{SVSF})^T+HK_i^{SVSF}HP_{i|i-1}H^T-HP_{i|i-1}H^T \\ =& e_{z,i}e_{z,i}^T-K_i^{SVSF}He_{z,i}e_{z,i}^T-e_{z,i}e_{z,i}^TK_i^{SVSF^T}H^T+K_i^{SVSF}He_{z,i}e_{z,i}^TK_i^{SVSF^T}H^T- \\ &HK_i^{SVSF}HP_{i|i-1}H^T-HP_{i|i-1}H^T \\ =& e_{z,i}e_{z,i}^T-K_i^{SVSF}He_{z,i}e_{z,i}^T-e_{z,i}e_{z,i}^TK_i^{SVSF^T}H^T+K_i^{SVSF}He_{z,i}e_{z,i}^TK_i^{SVSF^T}H^T- \\ &(I-K_i^{SVSF}H)(HP_{i|i-1}H^T) \\ =& (I-K_i^{SVSF}H)[e_{z,i}e_{z,i}^T(I-K_i^{SVSF}H)+HP_{i|i-1}H^T] \end{aligned} \tag{3.4.49}$$

Therefore, it is clear to have

$$\hat{R}_k = \frac{1}{k}\sum_{i=1}^{k}(I-K_i^{SVSF}H)[e_{z,i}e_{z,i}^T(I-K_i^{SVSF}H)+HP_{i|i-1}H^T] \tag{3.4.50}$$

As can be seen that Equation (3.4.37) has the complicated formulation. It might cause the reduction of filter accuracy. Therefore, by referring to Equation (3.4.38) – Equation (3.4.50) above, then Equation (3.4.37) is replaced with Equation (3.4.50). Furthermore, the time-varying noise estimator is proposed in this dissertation. According to Equation (3.4.34), Equation (3.4.56), Equation (3.4.57), and Equation (3.4.50), then respectively their time-varying unbiased noise properties are

$$\hat{q}_k = \hat{q} + \frac{1}{k}(K_i^{SVSF} e_{z,i}) \tag{3.4.51}$$

$$\hat{r}_k = \hat{r} + \frac{1}{k}\left[(I - HK_i^{SVSF})e_{z,i}\right] \tag{3.4.52}$$

$$\hat{Q}_k = \frac{k-1}{k}\hat{Q} + \frac{1}{k}\left[P_{i|i-1} - FP_{i-1|i-1}F^T\right] \tag{3.4.53}$$

$$\hat{R}_k = \frac{k-1}{k}\hat{R} + \frac{1}{k}(I - HK_i^{SVSF})\left[e_{z,i}e_{z,i}^T(I - HK_i^{SVSF}) + HP_{i|i-1}H^T\right] \tag{3.4.54}$$

where q, r, Q, and R are q_{k-1}, r_{k-1}, Q_{k-1}, and R_{k-1}, respectively. Now by applying the weighted exponent [8], [14], [58], [74], where the weighting coefficient d_k is formulated to replace the exponential $\frac{1}{k}$, then the alternative form of Equation (3.4.51) - Equation (3.4.54) become as follows

$$\hat{q}_k = \hat{q}_{k-1} + d_k(K_k^{SVSF} e_{z,k}) \tag{3.4.55}$$

$$\hat{r}_k = \hat{r}_{k-1} + d_k\left[(I - HK_k^{SVSF})e_{z,k}\right] \tag{3.4.56}$$

$$\hat{Q}_k = (1 - d_k)\hat{Q}_{k-1} + d_k\left[P_{k|k-1} - FP_{k-1|k-1}F^T\right] \tag{3.4.57}$$

$$\hat{R}_k = (1 - d_k)\hat{R}_{k-1} + d_k(I - HK_i^{SVSF})\left[e_{z,k}e_{z,k}^T(I - HK_k^{SVSF}) + HP_{k|k-1}H^T\right] \tag{3.4.58}$$

where the residual measurement $e_{z,k} = e_{z,k|k-1}$ and the weighting coefficient is expressed as follows

$$\begin{cases} \beta_i = d_k b^{i-1} \quad , \quad i = 1, ..., n \\ d_k = \frac{(1-b)}{(1-b^k)} \end{cases} \tag{3.4.59}$$

where b is fading factor satisfied $0 < b < 1$ and β_i is the i-th weighting factor defined as $\beta_i = \beta_{i-1}b$ and satisfied $\sum_{i=1}^{k}\beta_i$. Referring to the definition mentioned by Equation (4.2), both Q and R should be positive definite symmetric matrix. Unfortunately, the complicated formulation shown in Equation (3.4.58) will present the negative definite matrix for the measurement noise statistic covariance. In order to depress this possibility, the alternative formulation adopted from the innovation covariance estimator concept[8], [21], [57], [59], [61] was used. It can be presented below.

Assuming that the innovation covariance is expressed as follows

$$ICE_k = \frac{1}{N}\sum_{j=0}^{N} e_{z,k-j}e_{z,k-j}^T = ICE_{k-1} + \frac{1}{N}\left[e_{z,k}e_{z,k}^T - e_{z,k-j}e_{z,k-j}^T\right] \tag{3.4.60}$$

As described by Equation (3.4.60), the window size N plays a role of achieving an ICE accuracy. N is able to prevent the occurrence of the biased situation in ICE by setting it to small value for the fast change of the dynamic system. Besides that, it is also able to improve the stability of the unbiased ICE by setting it to large value for the slow change of the dynamic system. Simply, the adjustment of window size is strongly depending on the characteristic of dynamic system which can empirically to be adjusted. Then, it is noted that the noise estimator is stable when the following definition is satisfied.

$$tr(ICE_k) \leq \kappa \,.\, tr(S) \tag{3.4.61}$$

where κ is a reserve factor that satisfies $\kappa \geq 1$ and $tr(.)$ refers to trace of matrix. The stability equation expressed by Equation (3.4.61) shows that κ also plays an important role as a threshold to know the incorrectness of the noise statistic when the measurement outliers occurs. By means, it is an issue when the ratio between $tr(ICE)$ and $tr(S)$ is out of the threshold value κ at current step k. Therefore, an additional of the innovation covariance estimation will obviously be keeping the stability of estimated value Q and R by isolating the current innovation covariance $e_{z,k}e_{z,k}^T$ with its calculated value. Then by replacing $e_{z,k}e_{z,k}^T$ with ICE_k, Equation (3.4.58) becomes

$$\hat{R}_k = (1 - d_k)\hat{R}_{k-1} + d_k(I - HK_i^{SVSF})\left[ICE_k(I - HK_i^{SVSF}) + HP_{i|i-1}H^T\right] \tag{3.4.62}$$

At the last, the working principle of the adaptive SVSF can be graphically summarized as shown in Figure 3.4 below

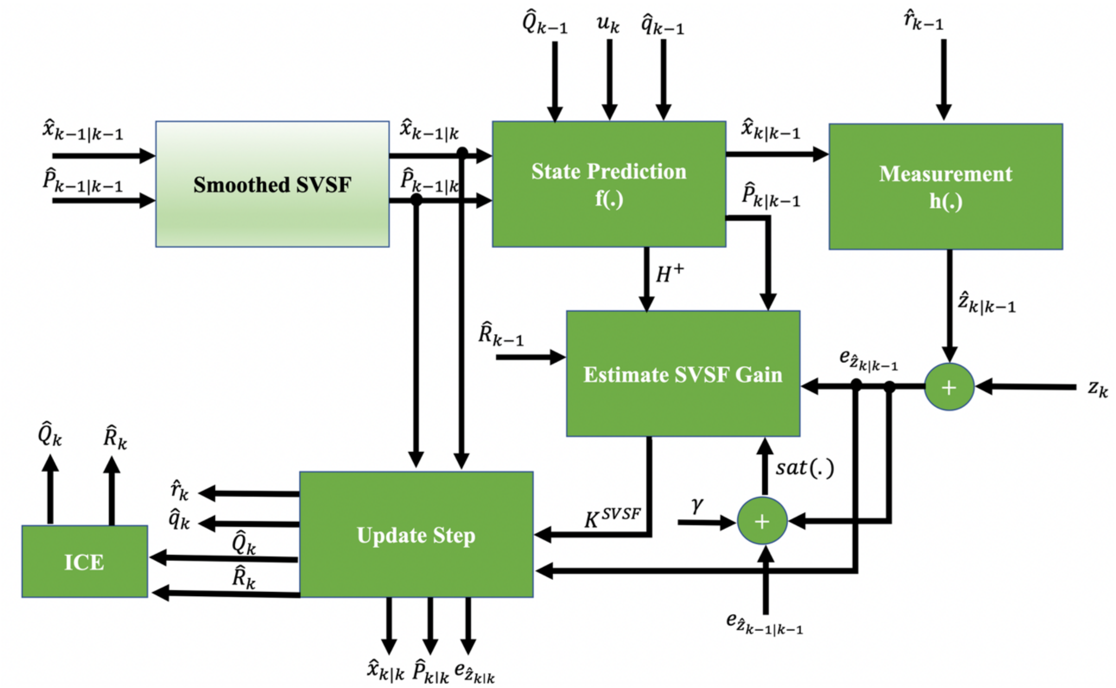


Figure 3.4 Working Principle of Adaptive SVSF Based on MLE and ICE

Chapter IV Applying Proposed Method for 2D Feature-Based SLAM Algorithm

This chapter presents the process of constructing the feature-based SLAM algorithm. All filtering methods presented in the previous chapter are applied in both simulation and real application. This implementation requires the motion model used in the prediction step, and the measurement model used to predict the measurement and calculate the innovation error. Moreover, the Jacobian matrixes of the state, control, and measurement needs to define. There are two types of experiment in this dissertation which are based on a realistic environment-based simulation and real-time-based simulation.

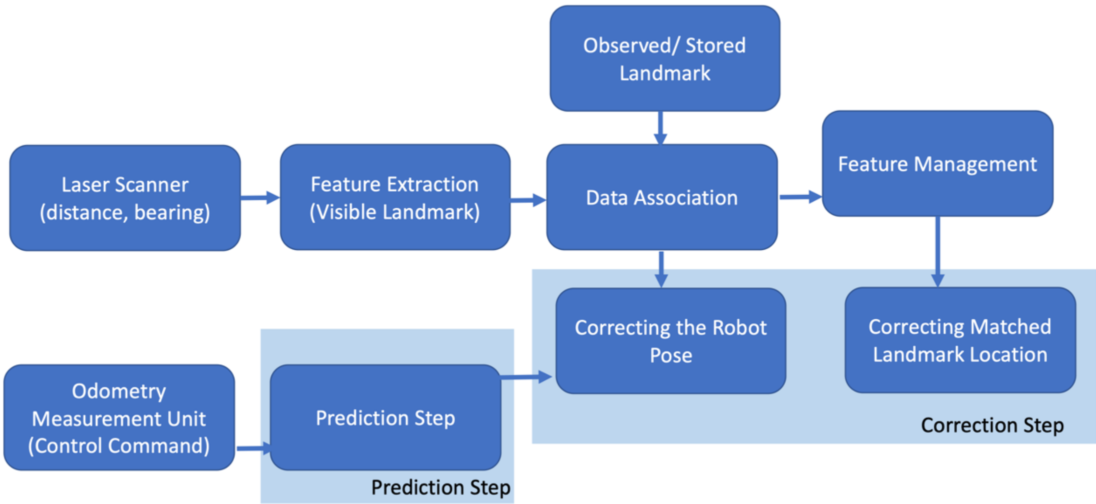


Figure 4.1 Flowchart of Feature-Based SLAM Algorithm

4.1 Simulation-Based Experiment

The simulation is realistically conducted by referring to the parameter of Turtlebot2 which equipped by the laser scanner. The robot moves depend on the different velocity of the right and wheels. It is henceforth called a control command which can be measured utilizing the odometry sensor. Naturally, the robot generates its path after executing all the control command with assumption both the system and control are not perturbated by any small noise. The robot is analogized to sense and measure the features in every step of state transition. Considering that the location of all the features are known by the user, therefore, the corresponding is known. Once, the reference is available, the robot is assumed to inaccurately moves because of some factors and the measurement is noisy due to incorrect robot base and unavoidable error perturbating the distance and bearing data. Consequently, the filtering-based SLAM algorithm is used to overcomes these issues by estimating the robot path and features coordinate in the global environment.

4.1.1 Robot Configuration and Motion Model

Assuming that the spatial and orientation of robot pose is denoted as follows

$$x_{R,k} = \begin{bmatrix} x_{r,k} \\ y_{r,k} \\ \theta_{r,k} \end{bmatrix} \tag{4.1.1.1}$$

Where k represents the discrete time index. Since the robot configuration and motion principle is graphically depicted in Figure 4.1, thus, the next position of the robot can be known once the current pose of the robot and control command are given.

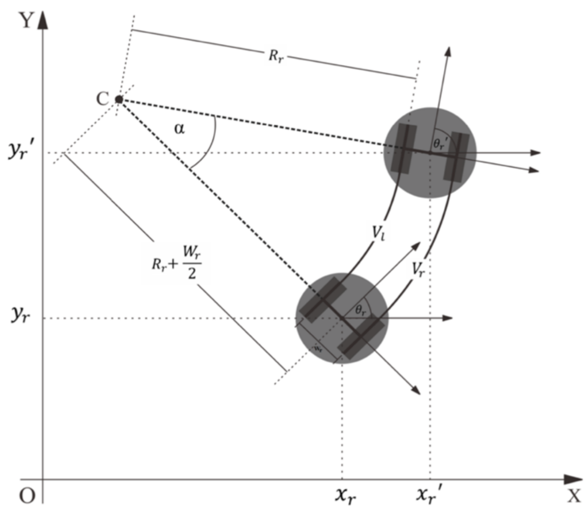


Figure 4.2 Kinematic Configuration of The Robot

Figure 4.1 represents the pose of the robot on the 2D planar environment as the global frame representation. It refers to the distance of the separated wheels W_r and the length between the robot's outer wheel R_r to the point of the turn causing angle C. As can be seen from Figure 4.1, the next location is achieved after the right v_r and left speed v_l are propagated. In which they are the perturbated velocity caused by the turn and move factor $\zeta = [\zeta_1 \quad \zeta_2]^T$.

$$\begin{cases} v_r = \zeta_1 u_r + \zeta_2(u_r - u_l) \\ v_l = \zeta_1 u_l + \zeta_2(u_r - u_l) \end{cases} \tag{4.1.1.2}$$

where, the right u_r and left u_l are the true control command given by the used. Now by assuming that the factors are random and unknown. Thus, there will be two different types of motion classified based on the diversity of speed. In which, this factor will make the robot move with or without a turn.

Then, by assuming that the right and left velocity are the same, the robot will move without changing its heading. Mathematically, it is expressed as follows

$$x_R^A = \begin{bmatrix} x_{r,k}^A \\ y_{r,k}^A \\ \theta_{r,k}^A \end{bmatrix} = \begin{bmatrix} x_{r,k-1}^A \\ y_{r,k-1}^A \\ \theta_{r,k-1}^A \end{bmatrix} + v_r \begin{bmatrix} cos(\theta_{r,k-1}^A) \\ sin(\theta_{r,k-1}^A) \\ 0 \end{bmatrix} \tag{4.1.1.3}$$

Contrary, when the right and left velocity are not same, in a certain angle the robot will turn depending on this diversity. Therefore, the analogy of this motion can be expressed as follows

$$x_R^B = \begin{bmatrix} x_{r,k}^B \\ y_{r,k}^B \\ \theta_{r,k}^B \end{bmatrix} = \begin{bmatrix} x_{r,k-1}^B \\ y_{r,k-1}^B \\ \theta_{r,k-1}^B \end{bmatrix} + \begin{bmatrix} \left(R + \frac{W}{2}\right)\left(sin\left(\theta_{r,k-1}^B + \alpha\right)\right) + sin(\alpha) \\ \left(R + \frac{W}{2}\right)\left(-cos\left(\theta_{r,k-1}^B + \alpha\right) + cos(\alpha)\right) \\ \alpha \end{bmatrix} \tag{4.1.1.4}$$

Where *A* is for the first case and *B* is for the second case. It is noted that the motion model is used to predict the robot state. Therefore, the conditional probability $p\left(x_{R,k}|x_{R,k-1}, u_k\right)$ is now satisfied. Moreover, the Jacobian matrices which are calculated based on the partial derivative of function *f(.)* with respect to the state and control are described as follows (see Appendix).

$$F_s = \begin{bmatrix} \frac{\partial x_{r,k}}{\partial x_{r,k-1}} & \frac{\partial x_{r,k}}{\partial y_{r,k-1}} & \frac{\partial x_{r,k}}{\partial \theta_{r,k-1}} \\ \frac{\partial y_{r,k}}{\partial x_{r,k-1}} & \frac{\partial y_{r,k}}{\partial y_{r,k-1}} & \frac{\partial y_{r,k}}{\partial \theta_{r,k-1}} \\ \frac{\partial \theta_{r,k}}{\partial x_{r,k-1}} & \frac{\partial \theta_{r,k}}{\partial y_{r,k-1}} & \frac{\partial \theta_{r,k}}{\partial \theta_{r,k-1}} \end{bmatrix} \tag{4.1.1.5}$$

As known that there are two different motion model in this case, therefore, the specified Jacobiana Fs is presented as follows.

$$F_s^A = \begin{bmatrix} 1 & 0 & \left(R + \frac{W}{2}\right)\left(cos(\theta_{r,k-1} + \alpha) - cos(\theta_{r,k-1})\right) \\ 0 & 1 & \left(R + \frac{W}{2}\right)\left(sin(\theta_{r,k-1} + \alpha) - sin(\theta_{r,k-1})\right) \\ 0 & 0 & 1 \end{bmatrix} \tag{4.1.1.6}$$

$$F_s^B = \begin{bmatrix} 1 & 0 & -u_l \,.\, sin(\theta_{r,k-1}) \\ 0 & 1 & u_l \,.\, cos(\theta_{r,k-1}) \\ 0 & 0 & 1 \end{bmatrix} \tag{4.1.1.7}$$

Up to this point, the state and its covariance can be calculated as follows

$$x_k = f(x_{k-1}, u_k) \tag{4.1.1.8}$$

$$P_k = F_s P_{k-1} F_s^T + Q_k \tag{4.1.1.9}$$

Where *f(.)* represents the motion model in Equation (4.1.1.3) – Equation (4.1.1.4). Meanwhile Q_k is calculated using the following equation.

$$Q_k = F_c \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} F_c^T \tag{4.1.1.10}$$

where σ_l and σ_r are direct variable obtained based on the relative effect and F_c refers to the Jacobian matrix calculated by taking the partial derivative of function *f(.)* with respect to the control (see Appendix).

$$F_c = \begin{bmatrix} \frac{\partial x_{r,k}}{\partial u_l} & \frac{\partial x_{r,k}}{\partial u_r} \\ \frac{\partial y_{r,k}}{\partial u_l} & \frac{\partial y_{r,k}}{\partial u_r} \\ \frac{\partial \theta_{r,k}}{\partial u_l} & \frac{\partial \theta_{r,k}}{\partial u_r} \end{bmatrix} \tag{4.1.1.11}$$

Remembering that the motion model in this case is depend on the different velocity of right and left, therefore, the construction of F_c are two. For the first F_c when the velocities are different all the completeness in Equation 4.1.6 is sequentially presented as follows

$$\frac{\partial x_{r,k}}{\partial u_l} = \frac{WR}{(u_r - u_l)^2}\Big(sin(\theta'_{r,k-1} - sin(\theta_{r,k-1}))\Big) - \frac{u_r + u_l}{2(u_r - u_l)}cos(\theta'_{r,k-1})$$

(4.1.1.12)

$$\frac{\partial y_{r,k}}{\partial u_l} = \frac{WR}{(u_r - u_l)^2}\Big(-cos(\theta'_{r,k-1} + cos(\theta_{r,k-1}))\Big) - \frac{u_r + u_l}{2(u_r - u_l)}sin(\theta'_{r,k-1})$$

(4.1.1.13)

$$\frac{\partial \theta_{r,k}}{\partial u_l} = -\frac{1}{W} \tag{4.1.1.14}$$

$$\frac{\partial x_{r,k}}{\partial u_r} = -\frac{WR}{(u_r - u_l)^2}\Big(sin(\theta'_{r,k-1} - sin(\theta_{r,k-1}))\Big) + \frac{u_r + u_l}{2(u_r - u_l)}cos(\theta'_{r,k-1})$$

(4.1.1.15)

$$\frac{\partial y_{r,k}}{\partial u_r} = -\frac{WR}{(u_r - u_l)^2}\Big(-cos(\theta'_{r,k-1} + cos(\theta_{r,k-1}))\Big) + \frac{u_r + u_l}{2(u_r - u_l)}sin(\theta'_{r,k-1})$$

(4.1.1.16)

$$\frac{\partial \theta_{r,k}}{\partial u_r} = \frac{1}{W} \tag{4.1.1.17}$$

Meanwhile, the right v_r and left speed v_l are the same, then all the part of the partial derivative of *f(.)* with respect to the control are sequentially described as follows

$$\frac{\partial x_{r,k}}{\partial u_l} = \frac{1}{2}\Big(cos(\theta_{r,k-1}) + \frac{u_l}{W}sin(\theta_{r,k-1})\Big) \tag{4.1.1.18}$$

$$\frac{\partial y_{r,k}}{\partial u_l} = \frac{1}{2}\Big(sin(\theta_{r,k-1}) + \frac{u_l}{W}cos(\theta_{r,k-1})\Big)) \tag{4.1.1.19}$$

$$\frac{\partial \theta_{r,k}}{\partial u_l} = 0 \tag{4.1.1.20}$$

$$\frac{\partial x_{r,k}}{\partial u_r} = \frac{1}{2}\Big(cos(\theta_{r,k-1}) - \frac{u_l}{W}sin(\theta_{r,k-1})\Big) \tag{4.1.1.21}$$

$$\frac{\partial y_{r,k}}{\partial u_r} = \frac{1}{2}\Big(sin(\theta_{r,k-1}) + \frac{u_l}{W}cos(\theta_{r,k-1})\Big) \tag{4.1.1.22}$$

$$\frac{\partial \theta_{r,k}}{\partial u_r} = 0 \tag{4.1.1.23}$$

4.1.2 Measurement Model

As commonly known that the state vector is the SLAM perception not only contains the robot pose but also all the feature location. Therefore, its representation is modelled as follows.

$$x_k = \begin{bmatrix} x_{R,k} \\ x^i_{L,k} \end{bmatrix} \tag{4.1.2.1}$$

where $x_{R,k}$ represents the robot pose variable at time k in previous discussion. It consists both the spatial location and its heading or orientation. Meanwhile $x^i_{L,k}$ gives the information of the i-th landmark coordinate consisting both the coordinate respect to x-axes $x^i_{l,k}$ and y-axes $y^i_{l,k}$ for $i = 1, 2, \ldots, \ldots, N-1, N$ at time k. It is noted that in the simulation-based experiment this set of measurement is determined when the robot senses the corresponding feature location in the environment using the measurement model as discussed below. Therefore, the set of observed landmarks is described as follows

$$x^i_{L,k} = \begin{bmatrix} x^1_{l,k} \\ y^1_{l,k} \\ \cdots \\ \cdots \\ x^N_{l,k} \\ y^N_{l,k} \end{bmatrix} \tag{4.1.2.2}$$

Where N is the number of landmarks available on the global coordinate system as the point-based map. Using the certain method of adding feature to state vector, the composed state vector can be fully presented as follows

$$x_k = \begin{bmatrix} x_{r,k} \\ y_{r,k} \\ \theta_{r,k} \\ x_{l,k}^1 \\ y_{l,k}^1 \\ \cdots \\ \cdots \\ x_{l,k}^N \\ y_{l,k}^N \end{bmatrix} \tag{4.1.2.3}$$

Where $x_{L,k}^i$ can be calculated using the direct-point-based observation as expressed below. The landmark detection is illustrated as shown in Figure 4.3.

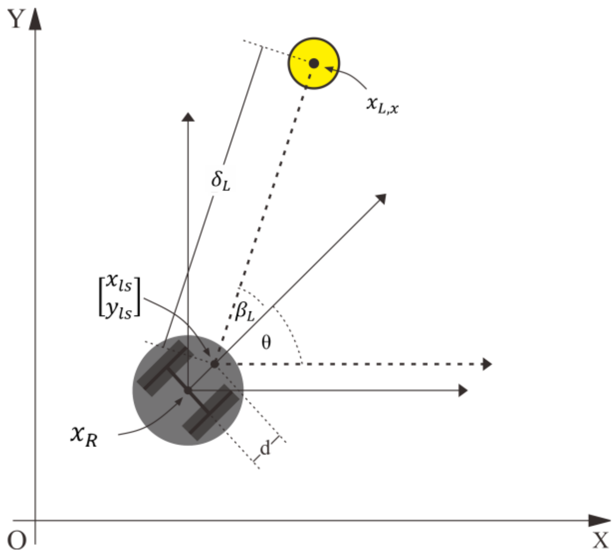


Figure 4.3 Landmark Detection

where $x_{ls}^R = [x_{ls} \;\; y_{ls}]^T$ refers to the position relative to the robot in local frame, and d or d_{ls} refers to the laser scanner displacement. Meanwhile, a single measurement consists the value of z_i. Therefore, given the current pose of the robot x_R, the location of the laser scanner in Equation (4.1.2.4), the direct-based observation model can be mathematically described in Equation (4.1.2.5).

$$\begin{bmatrix} x_{ls} \\ y_{ls} \end{bmatrix} = \begin{bmatrix} x_r \\ y_r \end{bmatrix} + d_{ls} \begin{bmatrix} cos(\theta_r) \\ sin(\theta_r) \end{bmatrix} \tag{4.1.2.4}$$

Sequentially, once the single landmark is found after applying the feature extraction algorithm, its values $x_L = [x_{L,x} \ \ x_{L,y}]$ is then used to produce the distance δ_L and bearing β_L. It can be calculated as follows

$$\begin{bmatrix} \delta_k^i \\ \beta_k^i \end{bmatrix} = \begin{bmatrix} \sqrt{\left(x_{l,k}^i - x_{ls,k}\right)^2 + \left(y_{l,k}^i - y_{ls,k}\right)^2} \\ atan2\left(\dfrac{y_{l,k}^i - y_{ls,k}}{x_{l,k}^i - x_{ls,k}}\right) - \theta_{r,k} \end{bmatrix} \tag{4.1.2.5}$$

Now, in order to make this observation satisfy the probabilistic model, the noise is assumed to always follows the measurement. It perturbates the measured distance δ_L and bearing β_L. Then, by supposing that these small noises are denoted as $r = [r_\delta \ \ r_\beta]^T$, which corresponding to the distance δ_L and bearing β_L, the final measurement model returns the measured landmark z_i.

$$z_i = \begin{bmatrix} \delta_L^i \\ \beta_L^i \end{bmatrix} + \begin{bmatrix} r_\delta \\ r_\beta \end{bmatrix} \tag{4.1.2.6}$$

Based on Equation (4.1.2.6), the predicted measurement and innovation sequence error are satisfied.

$$\hat{z}_{k|k-1} = h(\hat{x}_{k|k-1}) \tag{4.1.2.7}$$

$$e_{z,k|k-1} = z_k - \hat{z}_{k|k-1} \tag{4.1.2.8}$$

Where *h(.)* represents the measurement model and $\hat{z}_{k|k-1}$ is predicted measurement in Equation (4.1.2.6) and z_k is true corresponding measurement. The corresponding covariance of the predicted measurement is calculated using the following equation.

$$H = \frac{\partial z_k^i}{\partial x_{R,k}} = \begin{bmatrix} \frac{\partial \delta_k^i}{\partial x_{r,k}} & \frac{\partial \delta_k^i}{\partial y_{r,k}} & \frac{\partial \delta_k^i}{\partial \theta_{r,k}} \\ \frac{\partial \beta_k^i}{\partial x_{r,k}} & \frac{\partial \beta_k^i}{\partial y_{r,k}} & \frac{\partial \beta_k^i}{\partial \theta_{r,k}} \end{bmatrix} \tag{4.1.2.9}$$

Where *H* is partial derivative of h(.) with respect to the state (see Appendix).

$$H = \begin{bmatrix} \frac{-\Delta x}{\sqrt{q}} & \frac{-\Delta y}{\sqrt{q}} & \frac{d_{ls}}{\sqrt{q}}\left(\Delta x sin(\theta_{r,k}) - \Delta y cos(\theta_{r,k})\right) \\ \frac{\Delta y}{q} & \frac{-\Delta x}{q} & \frac{-d_{ls}}{q}\left(\Delta x cos(\theta_{r,k}) + \Delta y sin(\theta_{r,k})\right) - 1 \end{bmatrix} \tag{4.1.2.10}$$

where

$$\Delta x = x_{l,k} - x_{ls} \tag{4.1.2.11}$$

$$\Delta y = y_{l,k} - y_{ls} \tag{4.1.2.12}$$

$$q = (x_{l,k} - x_{ls})^2 + (y_{l,k} - y_{ls})^2 \tag{4.1.2.13}$$

4.2 Experiment Based on The Real-Time Application

In order to present more extensive and reliable validation, the proposed method is also practically implemented to solve the real-time application. It employs the real experiment data provided by University of Sydney. Accordingly, this subsection presents a step of constructing the algorithm based on all the filtering-based strategies in the previous Chapter. The real data is adopted from the real vehicle that was moved around the Victoria Park, Sydney, Australia. This vehicle is utilized to perceive the possible information of the park by using the Laser Scanner. Based on this information, the goal is to construct the feature-based map. Besides that, this map is generated together with the vehicle path. For this reason, the additional exteroceptive, odometry sensor, was also equipped to the vehicle. It is used to know the linear and angular velocity per time-duration. The odometry data is then used to process the filter as the path estimator. The GPS was also completed to collect the coordinate of vehicle pose. Inaccuracy of GPS is motivation to get the estimate values. Therefore, the feature-based SLAM is designed, in which it requires the motion model, observation model, Jacobian calculation, and Data Association.

4.2.1 Motion Model based on Linear and Angular Velocity

Different from the one used in the verification based on simulation above, the motion model is designed based on the following vehicle. This modelling refers to the real experiment conducted by Dr. Jose Guivant from the University of Sydney. The configuration of this vehicle is shown in Figure 4.4

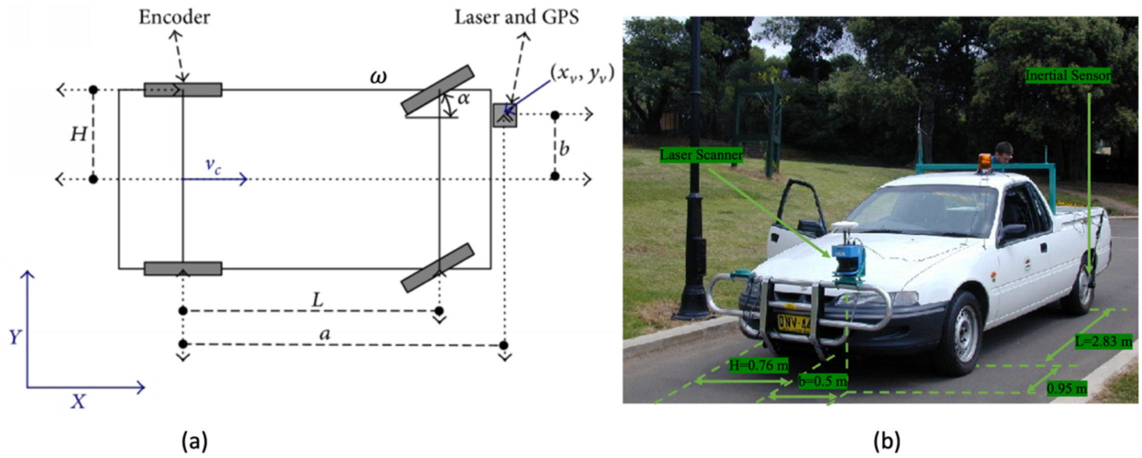


Figure 4.4 Kinematic Configuration (a) and Real Appearance of Vehicle (b)

Where the parameters of this vehicle is given as follows

$$L = 2.83, H = 0.76, b = 0.5, a = 3.78$$

By using the data provided by the odometer and GPS, the user can easily to know the robot pose. However, both types of data are bad and imprecise due to the present of noise. Additionally, the data of measurement of laser scanner is also noisy. Therefore, as the objective of this experiment, it is quite difficult to determine the accurate pose of vehicle and location of the feature. Accordingly, by this analogy the SLAM algorithm is applied to address this kind of issue. Initially, the reference trajectory is generated from the robot path based on GPS and odometer. And based on these benchmarks, the use of filtering method is approached. A filtering-SLAM algorithm predict and update the pose of the robot as well as construct the landmark-based map, simultaneously. The prediction requires the motion model as the state transition method caused by the control command. For this reason, the following motion model is presented.

$$x_{R,k} = \begin{bmatrix} x_{r,k} \\ y_{r,k} \\ \theta_{r,k} \end{bmatrix} = \begin{bmatrix} x_{r,k-1} + \Delta T(v_c.cos(\theta_{r,k-1})) - \frac{v_c}{L}.tan(\theta_{r,k-1})(a.sin(\theta_{r,k-1})) + b.cos(\theta_{r,k-1})) \\ y_{r,k-1} + \Delta T(v_c.sin(\theta_{r,k-1})) + \frac{v_c}{L}.tan(\theta_{r,k-1})(a.cos(\theta_{r,k-1})) - b.sin(\theta_{r,k-1})) \\ \theta_{r,k-1} + \Delta T \frac{v_c}{L}.tan(\alpha) \end{bmatrix}$$

(4.2.1.1)

It is noted that this modelling system is also provided together with the dataset. Next, the Jacobian matric F can be described as follows.

$$\frac{\delta f}{\delta x} = \begin{bmatrix} 1 & 0 & -\Delta T(v_c.cos(\theta_{r,k-1})) + \frac{v_c}{L}.tan\alpha(a.cos(\theta_{r,k-1})) - b.sin(\theta_{r,k-1})) \\ 0 & 1 & \Delta T(v_c.sin(\theta_{r,k-1})) - \frac{v_c}{L}.tan\alpha(a.sin(\theta_{r,k-1})) + b.cos(\theta_{r,k-1})) \\ 0 & 0 & 1 \end{bmatrix}$$

(4.2.1.2)

Similar to the previous discussion, up to this point, the prediction step for the state and its corresponding covariance used for all algorithms presented in Chapter 3 is satisfied.

4.2.2 Measurement Model for Second Experiment

The measurement is intended to predict the location of the landmark, when the current pose of the robot is given together with set of sensed/detected landmark in the global environment. It is also the initiated step before calculating the innovation sequence error and gain of filtering. The model of this observation can be illustrated in Figure below.

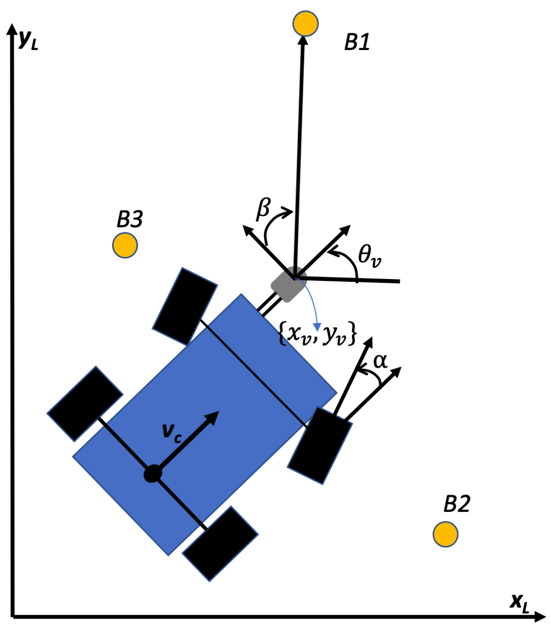


Figure 4.5 Landmark Detection

All newly observed landmarks are combined together as the state vector. Therefore, given the current pose of vehicle and raw data of laser scanner, the feature extraction and data association are conducted before it is proceeded to the step of measurement. In this experiment, the feature or landmarks are the tree trunks. Based on the raw sensing data, it is detected using the feature extractor. It is noted that this algorithm is also provided together with the dataset. Then, the extracted or detected landmark is measure as the prediction step. It is done by using the following measurement model.

$$\begin{bmatrix} z_r \\ z_l \end{bmatrix} = \begin{bmatrix} \sqrt{(x_L - x_r)^2 + (y_L - y_r)^2} \\ atan(\frac{(y_L - y_r)}{(x_L - x_r)}) - \theta_{r,k-1} + \frac{\pi}{2} \end{bmatrix} \tag{4.2.2.1}$$

Since the measurement is predicted, then the innovation sequence error and gain of filtering-based SLAM algorithm can be determined. However, the calculation also requires the Jacobian matrix H. It is done by taking partial derivative of h(.) with respect to the state as can be presented as follows

$$H = \frac{\delta h}{\delta x} = \begin{bmatrix} \frac{\delta h_\delta}{\delta x} \\ \frac{\delta h_\beta}{\delta x} \end{bmatrix} = \begin{bmatrix} \frac{\delta z_\delta}{\delta x_v, y_v, \theta_v} \\ \frac{\delta z_\beta}{\delta x_v, y_v, \theta_v} \end{bmatrix} \tag{4.2.2.2}$$

For

$$\frac{\delta h_\delta}{\delta x} = \frac{1}{\Delta}[-\Delta x, -\Delta y, 0] \tag{4.2.2.3}$$

$$\frac{\delta h_\beta}{\delta x} = \frac{1}{\Delta^2}[-\Delta y, -\Delta x, -1] \tag{4.2.2.4}$$

$$\Delta x = (x_L - x_v), \Delta y = [y_L - y_v], \Delta = \sqrt{\Delta x^2 + \Delta y^2} \tag{4.2.2.5}$$

Where $\{x_v, y_v, \theta_v\}$ and $\{x_L, y_L\}$ are robot and landmark pose, respectively. Therefore, the partial derivative with respect to the state (including the landmark pose) is

$$\frac{\delta h_\delta}{\delta x} = \frac{1}{\Delta}[-\Delta x, -\Delta y, 0, 0, 0, \ldots, \Delta x, \Delta y, 0, 0, \ldots 0] \tag{4.2.2.6}$$

$$\frac{\delta h_\beta}{\delta x} = -\frac{\Delta y}{\Delta^2}, \frac{\Delta x}{\Delta^2}, -1, 0, 0, \ldots, -\frac{\Delta y}{\Delta^2}, \frac{\Delta x}{\Delta^2}, 0, 0 \ldots 0] \tag{4.2.2.7}$$

4.2.3 Data Association

Although the feature extraction is already given, in this experiment still requires the data association. It is used to find the correspondence between the detected feature and the previously observed landmark. Naturally, the data association is a step used to check whether the newly observed landmark is available in the state vector or not. Therefore, if it is not stored the location of the feature is added into the state vector. Otherwise, the existing one is chosen as the base. The data association is important before conducting a measurement step. There are some techniques used, but the common nearest neighbor is approached in this experiment. The reason is, the landmarks have a large distance to each other, therefore, it can be separated or uniquely identified easily. The nearest neighbor firstly sets the associated gate to restrict the possible number of decisions to be made, and the associated gate 's preliminary screen makes the echoes a candidate. The Associated Door is a subspace in the tracking field. The center is located in the forecast of the tracked target position. The size of the configuration will ensure that the appropriate likelihood of an echo can be obtained to some degree. The nearest neighbor method always selects a point trace that falls into the association gate and is closest to the tracked target position. Usually, it is judged by the statistical distance. Through analysis, it is not difficult to find that the nearest neighbor data association is primarily appropriate for tracking domain targets, but only for a limited number of target instances or target tracking of a sparse area only. Defining the statistical distance: Hypothesis before the first k-fold scan, we have identified the N path. New results for the first k cycles are Z_j^k for $j = 1,2,\dots,N$. In the association gate of track i the difference vector between the observed j and track i is defined as the diversity of the measured value and the predicted value. The residual error is given as follows

$$e_{ij}(k) = z_j - h(\hat{x}^j_{k|k-1}) \tag{4.2.3.1}$$

Where *h(.)* is the measurement model. And by referring to the definition this residual has the covariance S_{ij}. Therefore, the mentioned statistical distance is

$$d_{ij} = [e_{ij}(k)^T S_{ij}^{-1} e_{ij}]^{1/2} \tag{4.2.3.2}$$

It is noted that d_{ij} is known as the Mahalanobis distance of the invention, taking into account the uncertainty of the expected calculation. Based on this judgement, then by setting a parameter γ_g to specifies the gate, the decision is made based on the following equation

$$d_{ij}^2 < \gamma_g \tag{4.2.3.3}$$

It means that if the norm vector of Mahalanobis distance d_{ij}^2 is smaller than the threshold γ_g, the associated landmark is selected. Contrary, the newly observed landmark is added as the state vector with appropriate index. Finally, by referring to all the designed filtering in Chapter 3 and sequent discussion in this chapter, the feature-based SLAM algorithm is generally given as follows.

Algorithm Feature-Based SLAM Algorithm

Require: Initial State Estimate, Covariance, Convergence Rate, and Initial Error
1: **loop**
2: **Prediction Step: If proprioceptive data is available**
3: Propagate the state estimate
4: Compute the Jacobian of *f(.)*
5: Propagate the covariance relative to the state
6: **Update: If the observation data is available**
7: Compute the innovation sequence error
8: Calculate Gain
9: Update the State, and Covariance
10: Compute the noise statistic
11: **end loop**

This algorithm is applicable for all adaptive filters that have been described in Chapter 3. For the record, the implementation using EKF-SLAM and SVSF-SLAM stage 10 is not required. This confirms that while adaptive filtering can recursively update all associated noise properties, conventional methods cannot. As a further note, the parameters of the EKF-SLAM algorithm do not need to have an initial definition of the convergence rate as is required in SVSF-SLAM.

Chapter V Experiment, Result, and Discussion

This chapter presents the verification and validation of the proposed method that is implemented for solving feature-based SLAM problem both in simulation-case and real-environment. These steps are initiated by comparing some existing algorithms and the proposed method, ASVSF-SLAM algorithm. As the objective of the SLAM problem, all the presented algorithms are used to estimate the truth robot path and features location. For the real environment, the reference trajectory and features location are provided by the GPS data adopted from the popular Victoria Park dataset. Meanwhile, the simulation case uses the generated path and measurement by assuming there is no noise following the control and measurement. Up to this point, there will be the real/provided data and estimated data given by all the filtering-based algorithms. For this reason, some parameters are used to compared all mentioned algorithms, such as the RMSE, prediction and update time, detected landmark, and Monte Carlo simulation. Based on these parameters, the effectiveness and performances are respectively verified and evaluated.

5.1 Realistic-Simulation of Feature-Based SLAM

For the simulation, the robot trajectory and features are designed. They are considered as the reference or the true data unfollowed by the noise. The second assumption is that a known-locating robot is equipped with the odometer and laser scanner used to respectively detect the wheel revolution and measure the feature. As the tradition, a robot moves when it executes the control command given by the users. However, all the control commands are perturbated with random-small noise to either right and left wheels in every step. Since, this command is velocity of the right and left wheel, the robot executes the noisy velocities which makes the generated path diverges from the truth. Additionally, the random error is also assumed to disturb the given distance and bearing data when the robot senses the location of the robot. By this analogy, the velocity-based motion model (in Chapter 2) and direct-based measurement are used. It is clear to see, in order to apply the motion model, both the distance between the left and right wheel, and displacement of the laser scanner should be given. Then, by adopting the parameters from a real robot platform, Turtlebot 2, the parameterization for the simulation case is presented in Table 5.1 below.

Table 5.1 Parameterization of All the Algorithms

No	Parameter	Value
1	W (length of body)	33 cm
2	d_{ls} (displacement of laser scanner)	14 cm

Up to this point, all the complement initially requires to execute the motion model are satisfied. As the user, the control commands are known and sent to the robot to make robot moves. Suppose that the robot is initially placed on the global map and there is no noise follows the command, the reference trajectory and map are assumed as depicted by the following figure.

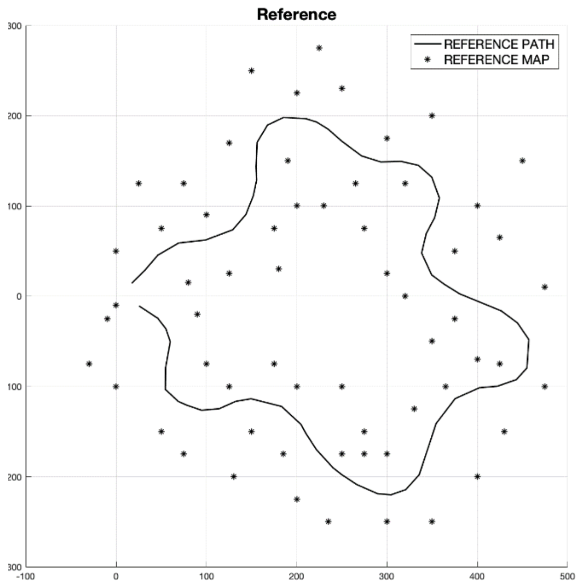


Figure 5.1 Reference Path and Map

The initial pose of the robot is assumed to be

$$x_{R,0} = \begin{bmatrix} x_{r,0} \\ y_{r,0} \\ \theta_{r,0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{35\pi}{180} \end{bmatrix} \tag{5.1.1}$$

Now, assuming that the robot strongly believes about its initial position, the initial covariance of this robot pose can be defined as

$$P_{R,0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.1.2}$$

Besides that, the initial noise statistics should be defined in order to applied all the algorithms. There are two simulations with different initial additive noise (as can be seen in Table 5.1) in this verification process. It is intended to validate the consistency and stability of the proposed method when the noise is increased.

Table 5.2 The Defined Noise Statistic for Two Different Simulation Parameter

Num. Sim	q_0	Q_0	r_0	R_0	γ	$e_{z,0}$
1st Sim.	$\left[0.3, \frac{3\pi}{180}\right]^T$	$\begin{bmatrix} 0.3^2 & 0 \\ 0 & \left(\frac{3\pi}{180}\right)^2 \end{bmatrix}$	$\left[0.2, \frac{6\pi}{180}\right]^T$	$\begin{bmatrix} 0.2^2 & 0 \\ 0 & \left(\frac{6\pi}{180}\right)^2 \end{bmatrix}$	1.5e -2	$[0\ 0]^T$
2nd Sim.	$\left[0.4, \frac{5\pi}{180}\right]^T$	$\begin{bmatrix} 0.4^2 & 0 \\ 0 & \left(\frac{5\pi}{180}\right)^2 \end{bmatrix}$	$\left[0.5, \frac{6\pi}{180}\right]^T$	$\begin{bmatrix} 0.5^2 & 0 \\ 0 & \left(\frac{6\pi}{180}\right)^2 \end{bmatrix}$	1.5e -2	$[0\ 0]^T$

Up to this point, the performance of the former EKF and alternative SVSF-based SLAM algorithm can be compared. The comparison uses the Root Mean Square Error as the term to evaluate the convergence of the proposed method.

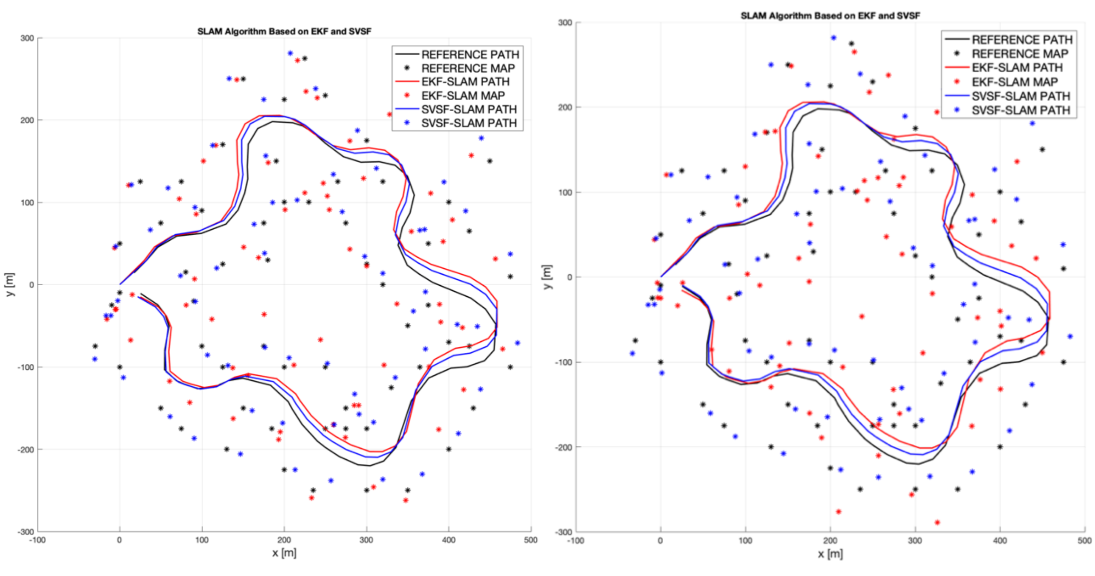


Figure 5.2 Performance of EKF and SVSF-based SLAM algorithm for 1st Simulation (Left) and 2nd Simulation (Right)

According to Figure 5.2, it can be declared that both optimal and robust filtering method can successfully estimate the robot path and map. This graphical result also proves that the SVSF-SLAM, can be alternatively replace the EKF-SLAM algorithm.

Moreover, in order to see clearly the EKF-SLAM performance, it is directly compared with its advanced, EKF-MAPWE-SLAM algorithm and AEKF-MLEEM-SLAM algorithm. They are compared in terms of RMSE for their Estimated Path Coordinate (EPC) and Estimated Map Coordinated. Remembering, that they are initially simulated referring to all initialization and parameterization shown in Table 5.1. Then, to prove that both adaptive EKF-SLAMs are convergence, the result is presented with the graphical performance as shown in Figure 5.3.

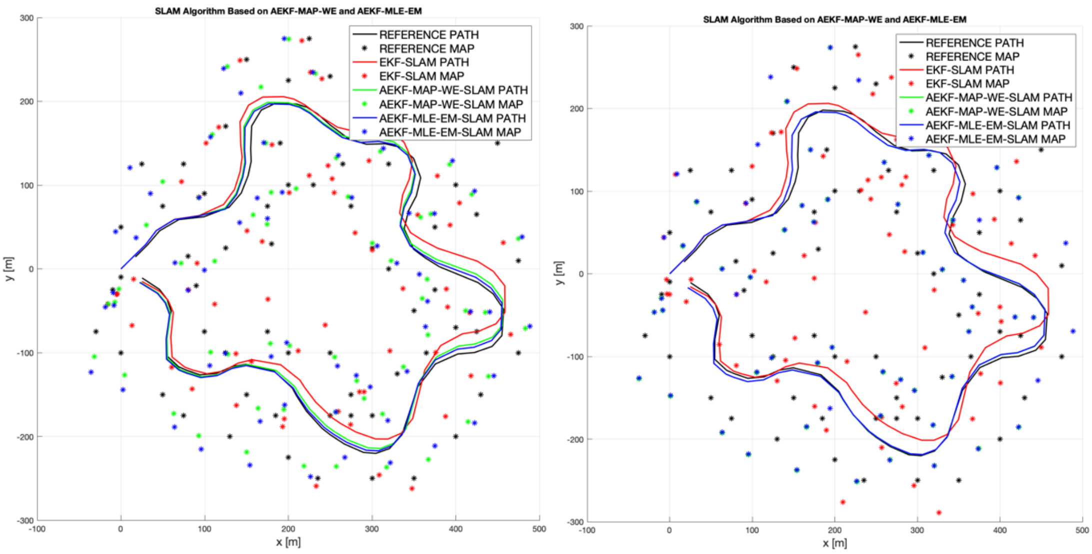
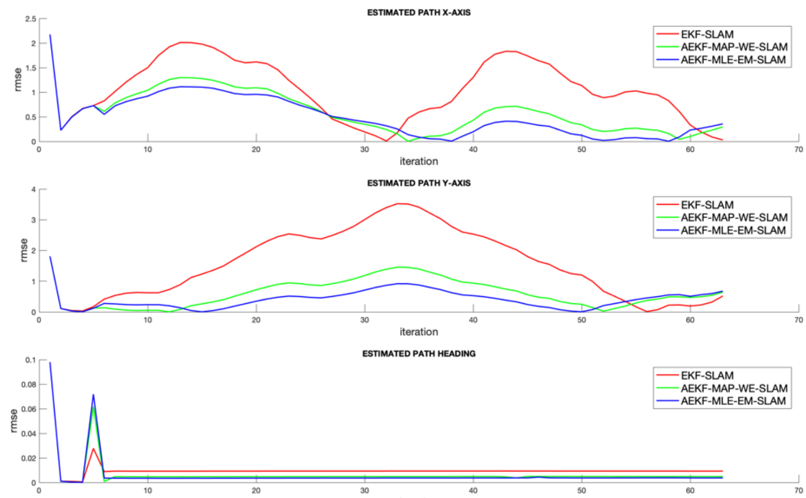
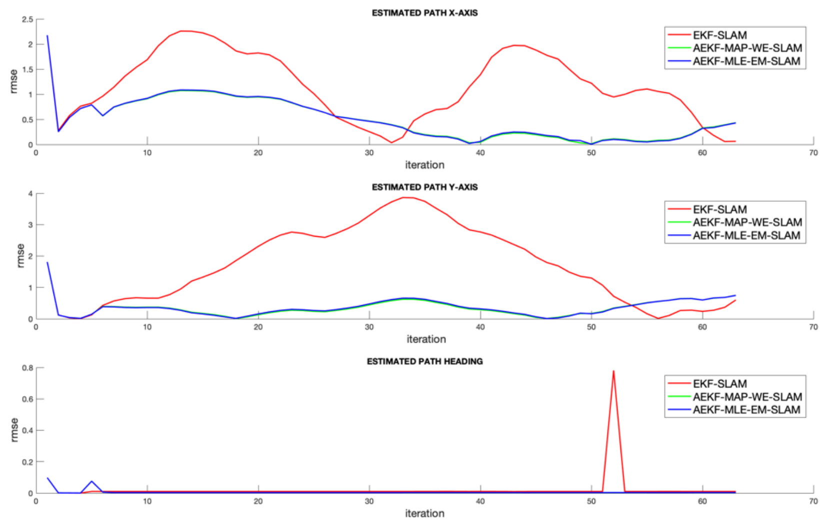


Figure 5.3 Performance of the EKF, AEKF-MAPWE and AEKF-MLEEM-SLAM Algorithm for 1st Simulation (left) and 2nd Simulation (right).

Graphically, Figure 5.3 shows that by involving adaptive filtering method, the performance of EKF-SLAM improves. It is proven from smaller gap between the estimated and reference path. The consistency and stability of adaptive filtering method can also be evaluated by this figure. In which, there is no much effect to the AEKF-SLAM algorithm when the noise statistic is increased. However, it is hard to see clearly the quality of estimated path and map by only referring to Figure 5.3. It is especially for the performance of adaptive EKF in estimating map. For this reason, the evaluation is also conducted by analyzing the comparison of the estimated path and map in term of RMSE.



(Top)

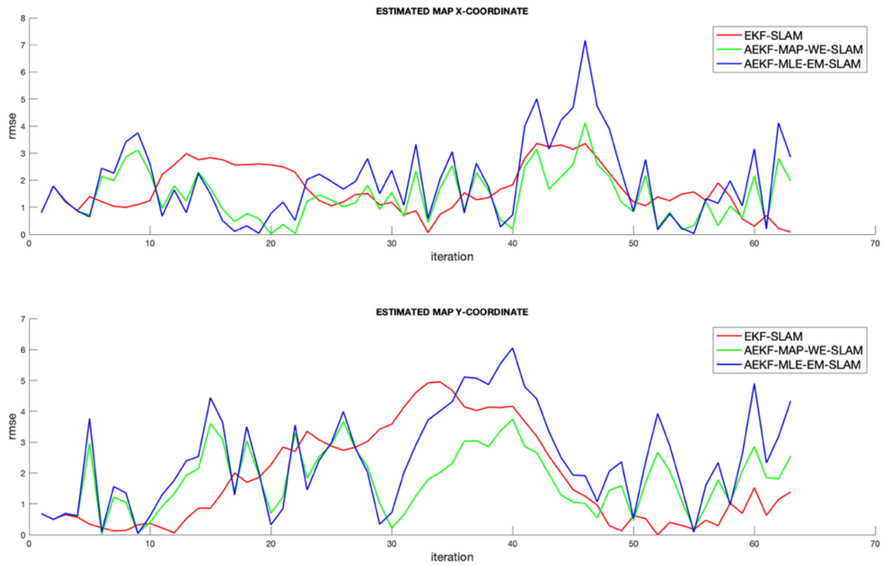


(Bottom)

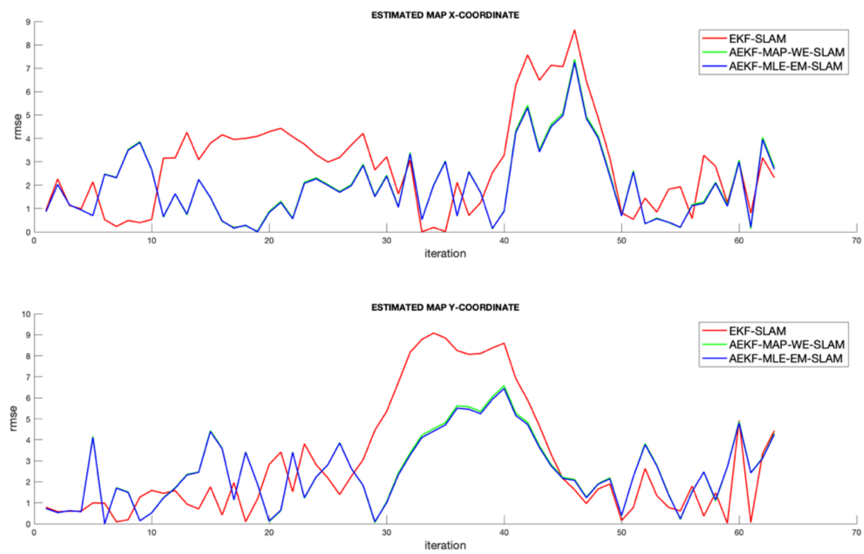
Figure 5.4 Estimated Path of EKF-SLAM, AEKF-MAPWE-SLAM, and AEKF-MLEEM-SLAM algorithm for 1st Simulation (Top) and 2nd Simulation (Bottom)

Figure 5.4 depicts the different RMSE values of the estimated path coordinate, including the robot's spatial coordinate $[x, y]^T$ and the robot heading θ. According to Figure 5.4, it can be now seen clearly the diversity between the EKF-SLAM and its advanced algorithm. Based on the estimated path for x-coordinate both the Adaptive EKF using MAP-WE with a divergence suppression method, and Adaptive EKF using MLE-EM with an Innovation Covariance Estimation are better than EKF-SLAM. It is shown from the smaller values generated with respect to the time step in Figure 5.4. The estimated path coordinate for y and θ-coordinate given by the adaptive filter are also better than EKF-SLAM. Besides that, the stability and consistency of the adaptive EKF-SLAM algorithms are also proven. It can be seen from their performance (2nd Simulation), which is stable even when the initial noise statistic is increased.

Furthermore, the performance is also evaluated based on the value of RMSE for the estimated map coordinate. It can be seen from Figure 5.5



(Top)



(Bottom)

Figure 5.5 Estimated Map of EKF-SLAM, AEKF-MAPWE-SLAM, and AEKF-MLEEM-SLAM algorithm for 1st Simulation (Top) and 2nd Simulation (Bottom)

Figure 5.5 depicts the performance of EKF-SLAM, AEKF-MAPWE-SLAM with divergence suppression method, and AEKF-MLEEM-SLAM with Innovation Covariance Estimate (ICE). According to the 1st Simulation, it might be hard to see the performance of adaptive EKF in estimating the map. But the significant difference can be evaluated from the 2nd Simulation. Therefore, it can now be stated that the adaptive EKF-SLAM algorithm significantly improves its predecessor with the guaranteed stability and consistency under invariant additive noise.

Additionally, the performance of adaptive SVSF-based SLAM algorithms is also validated. Like the previous manner, this validation involves the RMSE to evaluate its capability to estimate the robot path and map. First of all, the convergence of all algorithm based on Adaptive SVSF is evaluated. It can be done by seeing the graphical performance in estimating the path and map for both 1st and 2nd simulation.

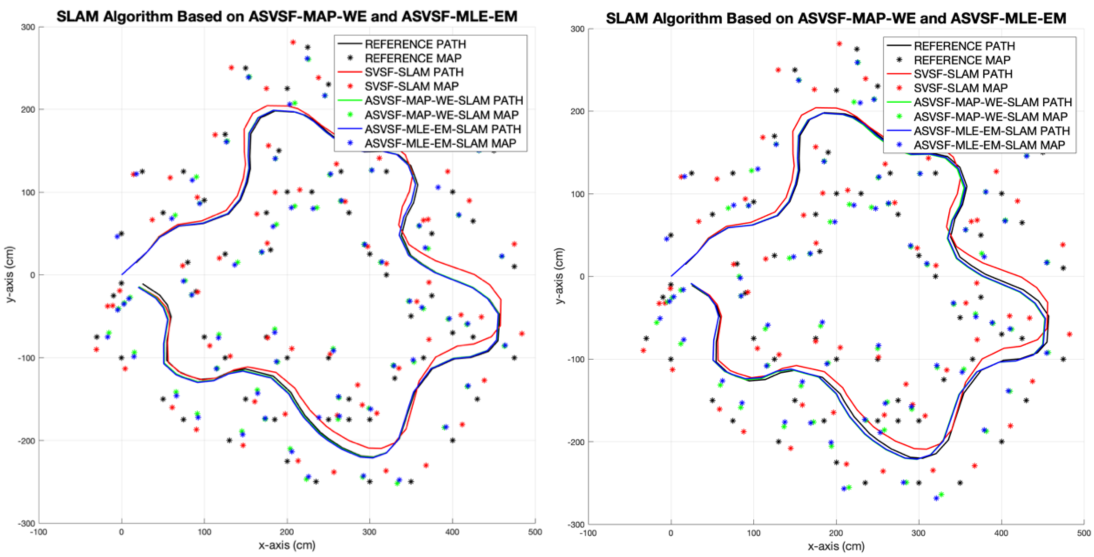
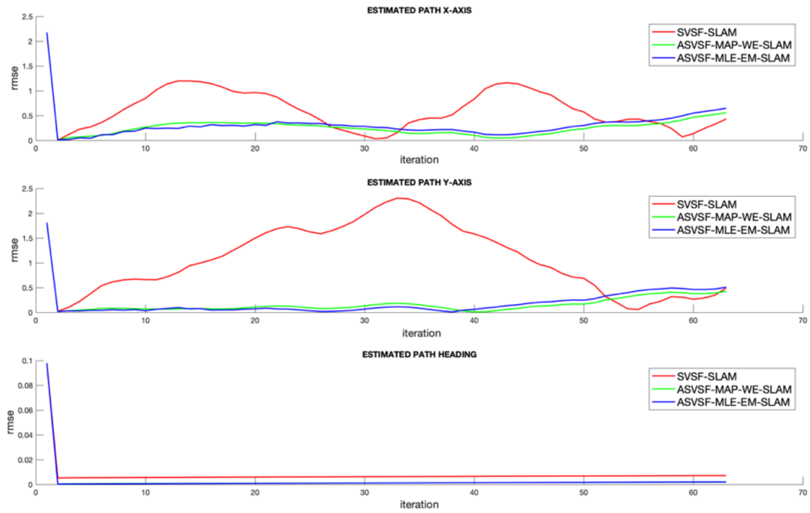
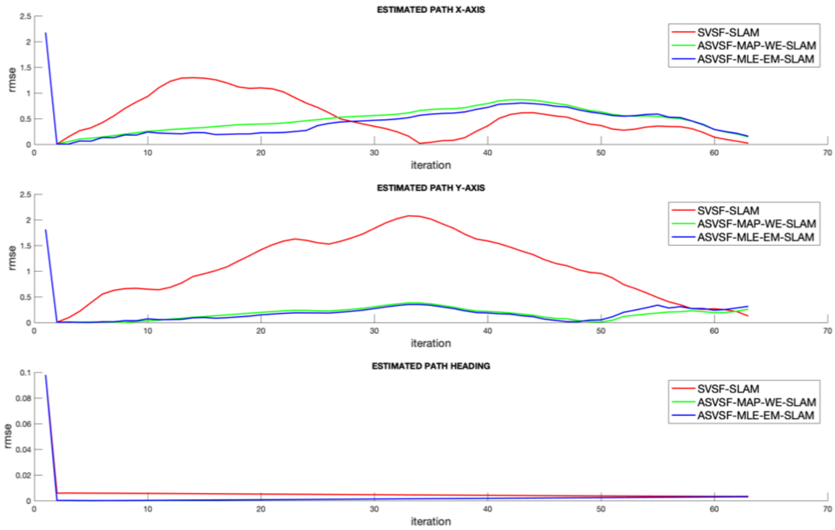


Figure 5.6 Performance Comparison between SVSF-SLAM, ASVSF-MAPWE and ASVSF-MLEEM-SLAM Algorithm for 1st Simulation (Left) and 2nd Simulation (Right)

Figure 5.6 illustrates the difference in performance between SVSF-SLAM, ASVSF-MAPWE-SLAM, and ASVSF-MLEEM-SLAM algorithm. Thus, it can be declared that the adaptive version significantly improves the performance of SVSF-SLAM algorithm when the small additive noise statistics are predetermined in Table 5.1. The consistency and stability are also guaranteed. It can be seen from Figure 5.6 which shows that the increment of the initially predetermined noise statistic gives no significant effect to the performance of all adaptive SVSF. Although, the effectiveness of ASVSF-SLAM algorithm in estimating the robot path can be easily evaluated from Figure 5.6, but its capability to estimate the map is difficult to be evaluated. Therefore, the graph of RMSE values for the estimated path and map coordinate is presented.



(Top)

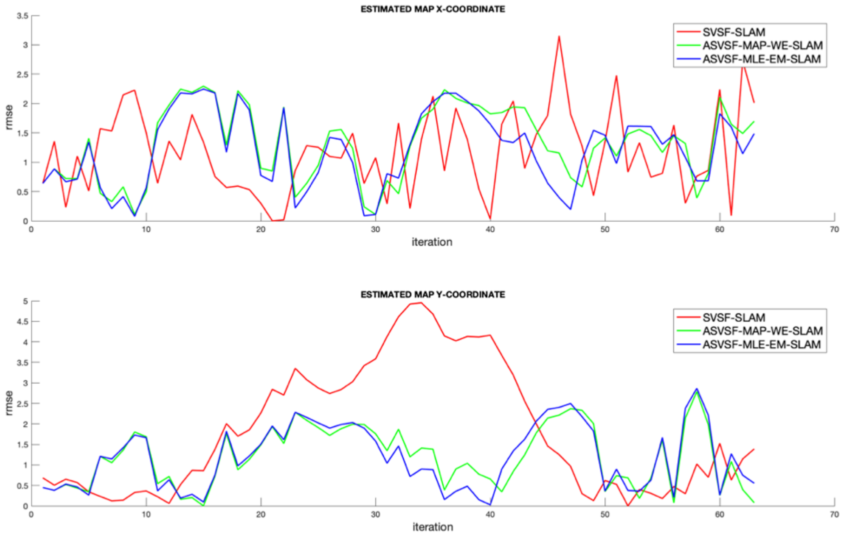


(Bottom)

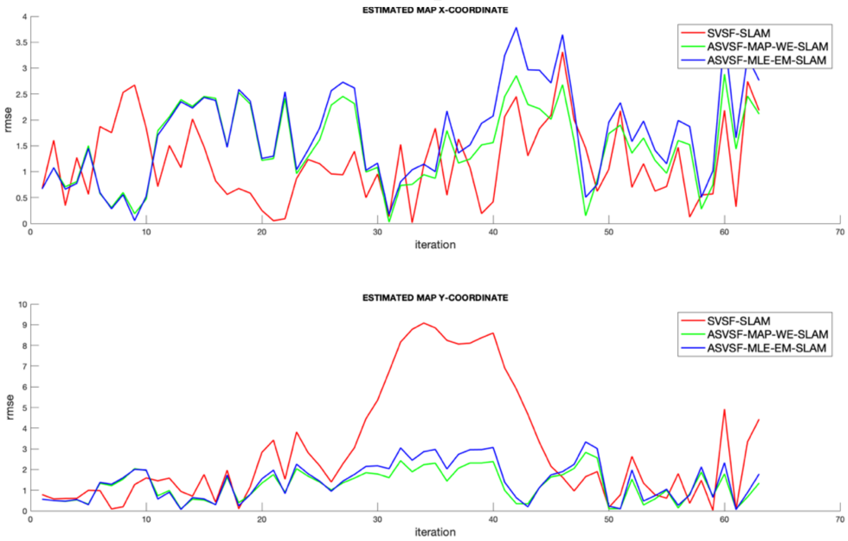
Figure 5.7 Estimated Path Coordinate of SVSF-SLAM, ASVSF-MAPWE and ASVSF-MLEEM-SLAM Algorithm for the 1st Simulation (Top) and 2nd Simulation (Bottom)

Figure 5.7 depicts the performance of SVSF-SLAM, ASVSF-MAPWE, and ASVSF-MLEEM-SLAM Algorithm for both estimating the spatial coordinate of a wheeled mobile robot and its heading direction in radian. According to Figure 5.7, it is clearly to see that all the adaptive filter relative to SVSF successfully improve the

SVSF-SLAM algorithm. It is proven based on better RMSE values for all the benchmarks, in which all the RMSE values are smaller than the SVSF-SLAM algorithm. This result proves that the previous statement that is stated based on the graphical performance (Figure 5.6). According to two different simulations in Figure 5.7, the consistency and stability of Adaptive SVSF are also satisfied. Next, to see the diversity of adaptive SVSF-SLAM performance, their RMSEs of estimated map coordinate are also presented as can be seen from Figure 5.8.



(Top)



(Bottom)

Figure 5.8 Estimated Map Coordinate of SVSF-SLAM, ASVSF-MAPWE and ASVSF-MLEEM-SLAM Algorithm for 1st Simulation (Top) and 2nd Simulation (Bottom)

According to Figure 5.8, it can be declared that the adaptive SVSF-SLAM algorithm is better than the SVSF-SLAM algorithm in approximating the landmark in two different simulation. It can be seen from its performance in estimating the y-coordinate of all the landmark. However, similar to Figure 5.5, the performance of SLAM algorithm in estimating the x-coordinate of the landmark is difficult to be evaluated. For this reason, Table 5.3 and Table 5.4 are presented to respectively show the different SLAM algorithm's performance in term of RMSE for Estimated Path and Estimated Map Coordinate.

Table 5.3 RMSE values of The Feature-Based SLAM algorithm based on EKF, AEKF-MAPWE, AEKF-MLEEM, SVSF, ASVSF-MAPWE, and ASVSF-MLEEM (1^{st} Simulation)

No	Name of Algorithm	Estimated Path Coordinate			Estimated Map Coordinate	
		x	y	θ	x	y
1.	EKF-SLAM	9.6867	14.980	0.1240	14.8801	18.8703
2.	AEKF-MAPWE-SLAM	5.8164	5.9584	0.1214	13.3618	16.1547
3.	AEKF-MLEEM-SLAM	5.1243	4.0567	0.1247	19.3317	23.1409
4.	SVSF-SLAM	5.9065	10.045	0.1099	2.0095	2.2145
5.	ASVSF-MAPWE-SLAM	3.0666	2.3328	0.0985	11.4657	11.0790
6.	ASVSF-MLEEM-SLAM	3.2337	2.5043	0.0985	10.8512	11.1995

Table 5.3 presents the different values of RMSE for the estimated path coordinate and the estimated map coordinate for all the algorithm in the 1^{st} Simulation. According to Table 5.3, two adaptive EKF-SLAM algorithms give significant improvement to the EKF-SLAM algorithm. The RMSE reduction shows it to almost all benchmark. This values also highlight the previous analysis which is conducted based on Figure 5.2 – Figure 5.4. Besides that, by comparing to its former version, all adaptive SVSF-SLAM algorithm also shows great result indicated by smaller RMSE values for all the benchmark.

Table 5.4 RMSE values of The Feature-Based SLAM algorithm based on EKF,

AEKF-MAPWE, AEKF-MLEEM, SVSF, ASVSF-MAPWE, and ASVSF-MLEEM

(2nd Simulation)

No	Name of Algorithm	Estimated Path Coordinate			Estimated Map Coordinate	
		x	y	θ	x	y
1.	EKF-SLAM	10.666	16.333	0.7913	27.7747	32.0915
2.	AEKF-MAPWE-SLAM	5.1149	3.4892	0.1265	20.0421	24.2682
3.	AEKF-MLEEM-SLAM	5.1421	3.5478	0.1262	19.7067	23.8953
4.	SVSF-SLAM	5.5823	9.7952	0.1045	17.1811	14.4686
5.	ASVSF-MAPWE-SLAM	4.6835	2.3872	0.0989	13.2414	11.5277
6.	ASVSF-MLEEM-SLAM	4.6835	2.3807	0.0987	15.4925	13.5943

Table 5.4 shows that all the proposed algorithms are effectively solving the problem of feature-based SLAM. According to Table 5.3 and Table 5.4, the AEKF-SLAM algorithm shows better improvement to its conventional algorithm, EKF-SLAM algorithm. The effectiveness of using an adaptive approach to the traditional filtering is also proven by all the performance of ASVSF-SLAM in this 2nd Simulation.

Based on Table 5.3 and Table 5.4, it is clear to declare that the AEKF and ASVSF can be alternatively used for feature-based SLAM algorithm. The noisy process and measurement can be represented based on how large the initial additive noise statistic. Accordingly, the AEKF-SLAM algorithms can be applied when the uncertainty is average, and the ASVSF-SLAM can be used for either condition of the uncertainty is significant or not. Furthermore, referring to the Bayes-Rules, which composed by Maximum A Posterior (MAP) and Maximum Likelihood Estimation (MLE), there should be different for the optimal and robust result of adaptive by using the MAP and MLE, since the prior knowledge is assumed to be available. However, they can be characteristically different since the supportable methods are respectively adopted from the divergence suppression method and Innovation Covariance Estimation. Regarding the discussion in Chapter 3, the divergence suppression method is intended to correct the predicted covariance matrix about the state in the prediction step. The derived formulation does not have any effect because of the gain. Meanwhile, the use of Innovation Covariance Estimation is applied once the corrective gain is calculated, and it replaces the covariance of innovation error. Therefore, it is not surprising when the result of the AEKF-MLEEM-SLAM is good and better for the AEFK-MAPWE-SLAM; the ASVSF-MLEEM is not better than the ASVSF-MAPWE-SLAM algorithm (see Table 5.3 and Table 5.4). The reason is that there is a different formulation of corrective gain between EKF and SVSF, which of course directly affected due to the change of additive noise statistic. However, the designed adaptive filtering for both EKF and SVSF is successfully designed. Additionally, the proposed method in this dissertation also successfully validated in terms of effectiveness and stability. By means, since the conventional algorithm, EKF or SVSF-SLAM, does not have the ability to respond to the system through the recursive noise statistic, their proposed method does have. Obviously, it overcomes the issues of keeping noise statistic to be same whole the estimation process is not recommended

5.2 Verification using the Victoria Dataset

Besides comparing the proposed algorithm in the simulation case, its effectiveness and performance are also verified and evaluated for the real application. In which all the algorithms are applied to solve the feature-based SLAM problem of vehicle in Victoria Park. The data of this experiment was collected by Nebot (2009) at the Australian Centre for Field Robotics, Sydney. It was done using a vehicle (a truck) equipped with a laser scanner, odometer, and GPS sensor. The vehicle moves in 4 kilometers along the park over a total time about 26 minutes. The trees are detected during the vehicle is being moved based on the measurement in each step. The local minima detection is used as the algorithm for extracting the feature in the scanning data. This tree extraction function is provided together with the dataset compatible with Matlab. The Victoria Park consists of some trees separated with large distance which makes the common data association applicable for this experiment. In this dissertation, the experiment is conducted on a Dual-Core Intel Core i5-2.3 GHz.

The interest is that this dataset has a bad quality on the odometer data with much unexpected perturbations and the truth given by GPS's data seems not accurate (see Figure 5.9). The trees in the park are distinctive features detectable by the laser scanner, which makes this dataset becomes the popular option to validate any feature-based SLAM algorithm. Different from the above simulation, this validation approaches the second creation of the motion model, measurement and all the relative Jacobians in Chapter 4. At first, in order to implement all algorithms introduced in Chapter 3, the following characteristic are defined as the parameterization step.

Table 5.5 The initial noise statistic

γ	$e_{z,0}$	q_0	Q_0	r_0	R_0
0.8	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}0.02\\ \frac{2\pi}{180}\end{bmatrix}$	$\begin{bmatrix}0.02^2 & 0\\ 0 & \frac{2\pi}{180}^2\end{bmatrix}$	$\begin{bmatrix}0.05\\ \frac{\pi}{180}\end{bmatrix}$	$\begin{bmatrix}0.05^2 & 0\\ 0 & \frac{\pi}{180}^2\end{bmatrix}$

Where q_0 contains the small additive noise following the linear velocity (*m/s*) and angular velocity (*rad/s*), and Q_0 refers to covariance matrix of the control. Meanwhile, r_0 is small additive noise following the range (*m*) and bearing (*rad*) of laser scanner measurement, and R_0 refers to the covariance matrix relative to the measurement. It is noted that these parameters will be kept to be constant by EKF-SLAM and SVSF-SLAM algorithm, but they are recursively updated by AEKF-SLAM and ASVSF-SLAM algorithm. Due to the closed similarity between AEKF-MAP-SLAM and AEKF-MLE-SLAM algorithm and ASVSF-MAP-SLAM and ASVSF-MLE-SLAM, this experiment only considers AEKF-MLE-SLAM and ASVSF-MLE-SLAM. In this case, the truth of vehicle path can be seen from the following figure.

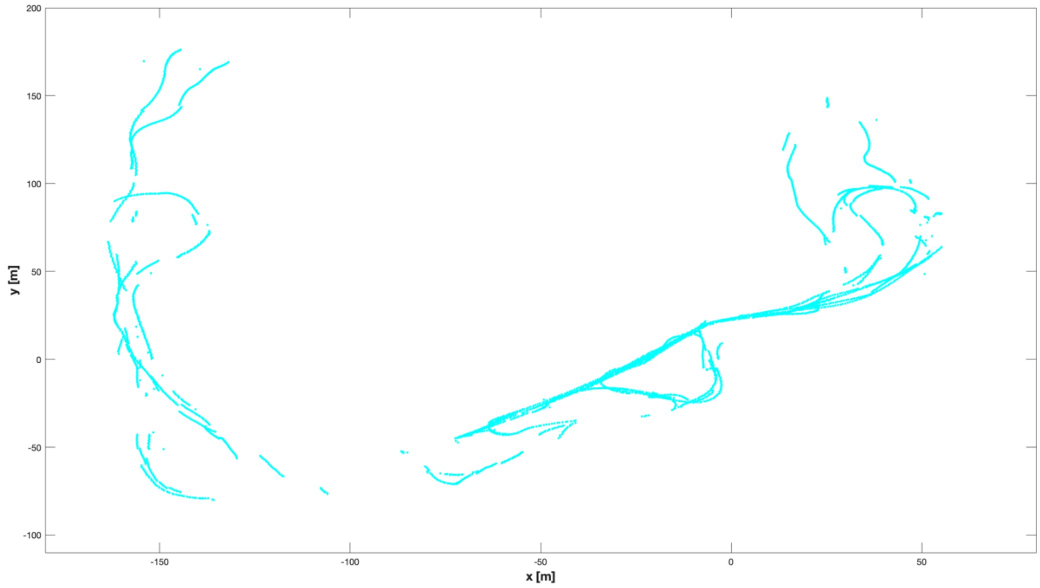


Figure 5.9 GPS-generated Vehicle Path

With a task to locate the robot pose and the landmark position in Victoria Park, all the feature-based algorithms are applied. This implementation is can be illustrated in Figure 5.10. The number of landmarks is reduced by using the feature elimination method based on the negative evidence information. It aims to determine more reliable representation after the data association is applied.

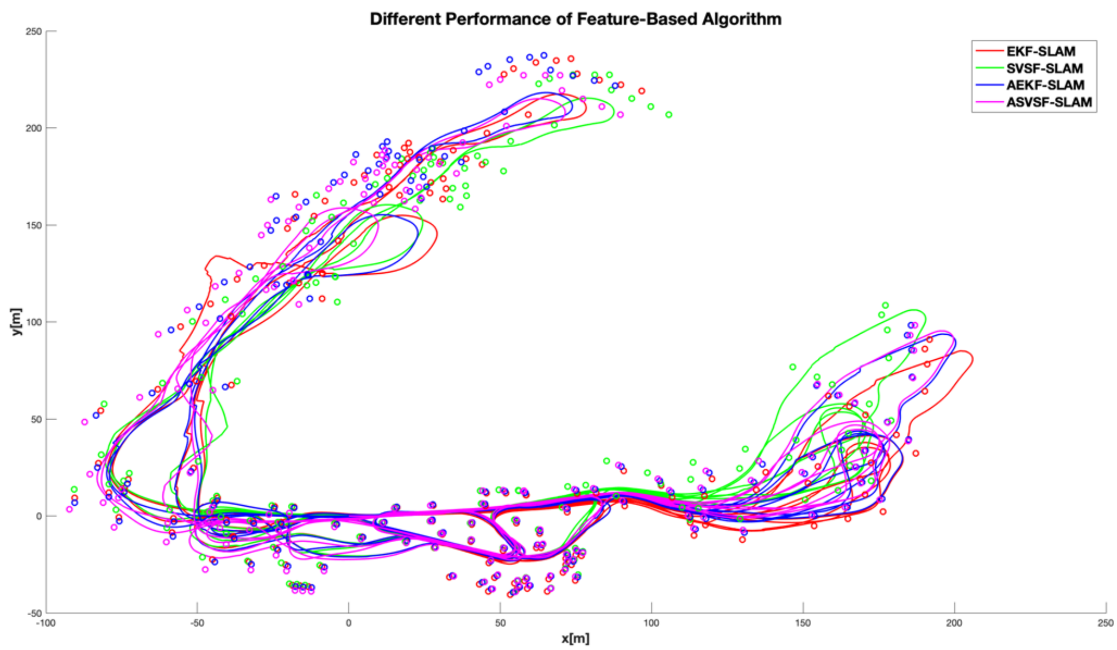


Figure 5.10 Performance of Different SLAM algorithm

As can be seen, that the proposed method gives significant improvement compared to the conventional algorithm, EKF-SLAM. It can be validated according to the smoother and better performance in estimating the robot path. As the second effort to validate the effectiveness of the proposed method, the obtained number of landmarks is also analyzed. Graphically, it can be seen as follows.

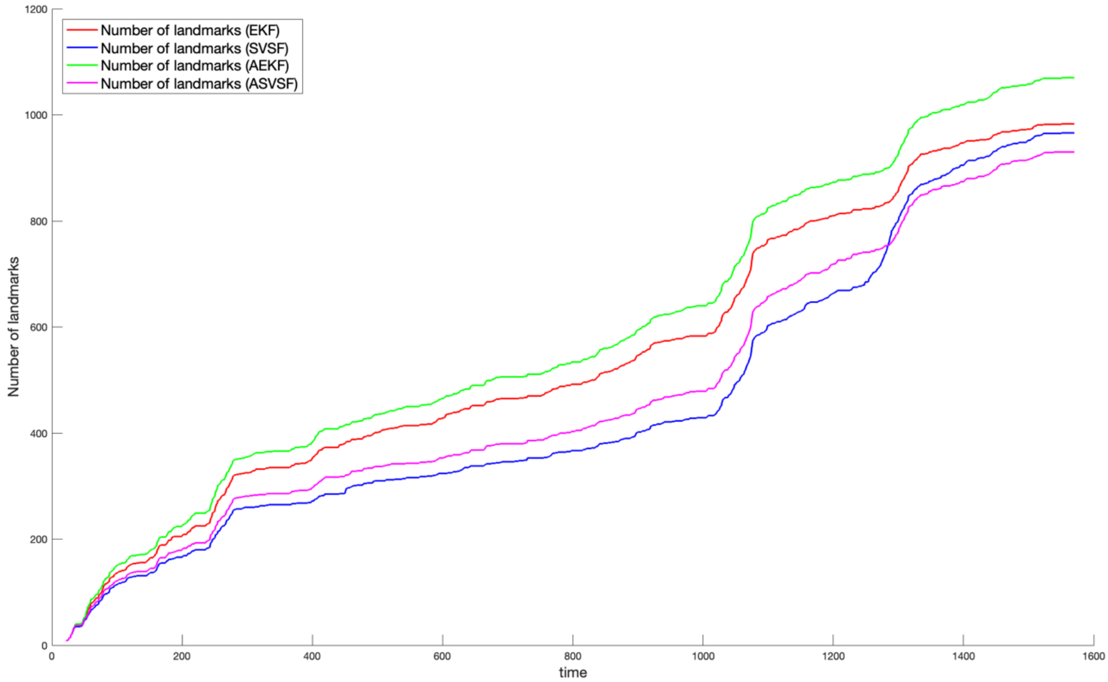


Figure 5.11 Number of The Stored Landmarks in 1600 (s)

Figure 5.11 illustrates the number of stored landmarks after executing all the control command. The proposed method is able to store less compared to the conventional one after applying the data association. In which the collected coordinates are kept and add new landmark. Additionally, the computational cost containing the prediction and update stage is also evaluated in this verification. Theoretically, this comparison lies on the different of update time process only. It is because all the algorithms perform the same prediction step. The following figure represents the duration of all the algorithms in predicting and updating the state (robot pose and map).

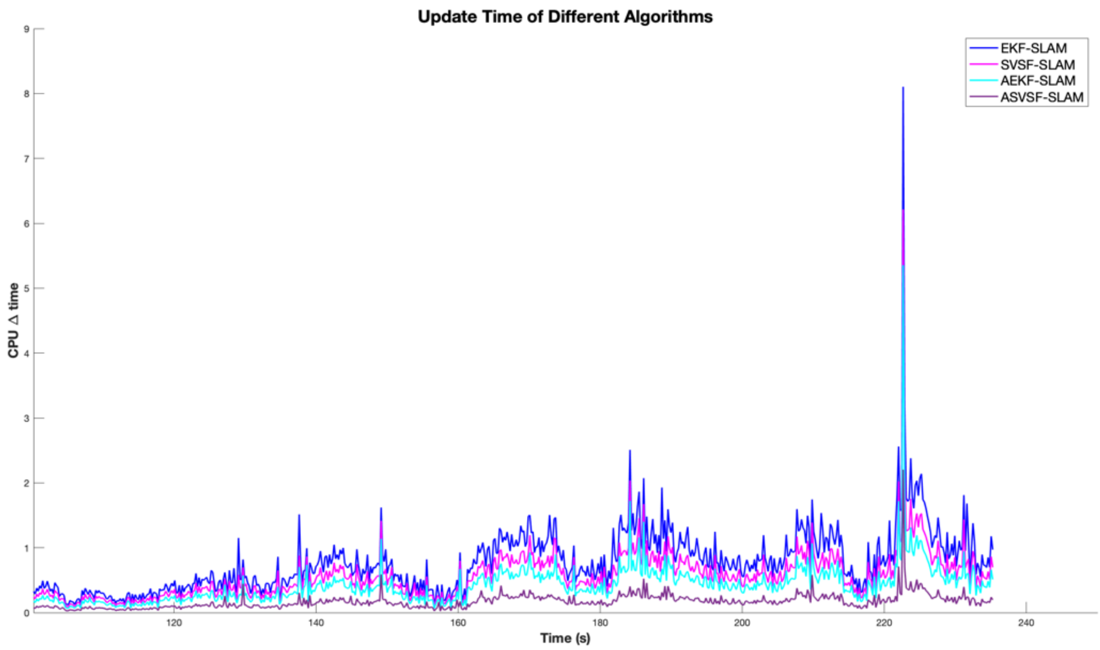


Figure 5.12 Update Time

5.3 Consistency Validation of The Proposed Algorithm

In all the theoretical convergence properties used in the last section, it is assumed that the Jacobians are evaluated at the true observation point and at the true landmark location. However, in fact of SLAM application both the true of robot pose and landmark position are not available/visible. Accordingly, the Jacobians must be measured at the approximate values. It is well known that this kind of linearization error can be incorrect. For this reason, the traditional metrics used to measure the efficiency of the estimation process, such as root mean squared (RMS) error (as conducted above), do not include consistency information. It is because they do not take into account the uncertainty returned by the filter. In order to provide more extensive verification and cover the lack of validation, the both the average of RMSE and Normalized Approximation Error Squared (NEES) are used to evaluate the all algorithms under repeated runs of Monte Carlo Simulation.

The NEES can be applied by measuring the mean squared value of the error that is normalized through the covariance matrix of all filtering-based strategy (e.g. EKF, AEKF, SVSF, or ASVSF). The NEES can be calculated when the density of probability is unknown but the availability of ground truth is known. This type of term is commonly used to characterize whether the filtering is consistence or not. The average NEES can be computed based on determination of the average values of Mahalonobis distance of the estimator.

$$\epsilon_k = (x_k - \hat{x}_{k|k})^T P_{k|k-1}^{-1}(x_k - \hat{x}_{k|k-1}) \tag{5.2.1}$$

The filter is consistence if the following properties are satisfied.

$$\tilde{x}(k|k) = x(k) - \hat{x}(k|k) \tag{5.2.2}$$

$$\epsilon(k) = \tilde{x}(k|k)^T P(k|k)^{-1}\tilde{x}(k|k) \tag{5.2.3}$$

Linguistically, it means that the filter is unbiased and the estimated covariance is matched to its theoretical one. Therefore, by the evaluation in term of average NEES can also prove whether the designed filter in this dissertation is unbiased or not. Under the assumption that the filter is consistent and roughly linear-Gaussian, the Mahalanobis complies with the chi-square distribution with the dimension $dim(x_k)$. Consequently, the average value of $\epsilon(k)$ tends towards the state dimension as N approaches infinity.

$$E[\epsilon(k)] = n_x \tag{5.2.4}$$

As well-known that the average NEES can also be implemented based on the real-time application (single-online run, N=1) and simulation (multiple N-runs or Monte Carlo Simulation). Given N runs and the average error of estimator, then the average NEES is computed as follows

$$\bar{\epsilon}(k) = \frac{1}{N}\Sigma_{i=1}^{N}\epsilon^i(k) \tag{5.2.5}$$

Given a hypothesis of a consistent linear-Gaussian filter, $N\bar{\epsilon}\ (k)$ has χ^2 density with $N.dim(x_k)$ degree of freedom. Therefore, for 3-dimentional of robot state, the 95% probability concentration region for $\bar{\epsilon}(k)$ is bounded by the interval [0.1198, 9.7218] for real-time application (N=1). Meanwhile, in the case of simulation with constant velocity and known number and corresponding landmark, the probability concentration region is bounded by the interval [2.6701 3.3491]. In which the interval is also known as acceptance region which used to evaluate the average NEES. Filter is considered to be consistent, if its average NEES falls into this region. Contrary, the filter is optimistic and conservative if its average NEES rises the upper bound and lower bound, respectively.

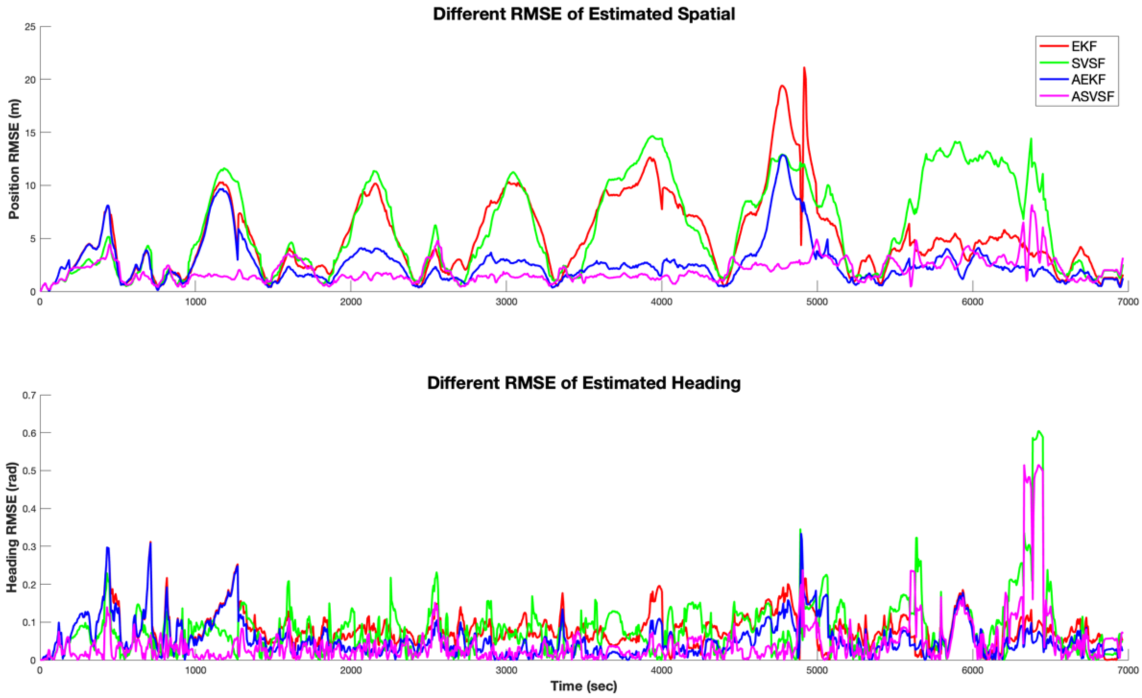


Figure 5.13 Average RMSE of different algorithm under Monte Carlo Simulation with N=1

It is clear shown in Figure 5.13, that the proposed method significantly improves the existing methods. The SVSF-SLAM algorithm can also be alternatively used to replace the EKF-SLAM. This result is detail presented in Table 5.3.

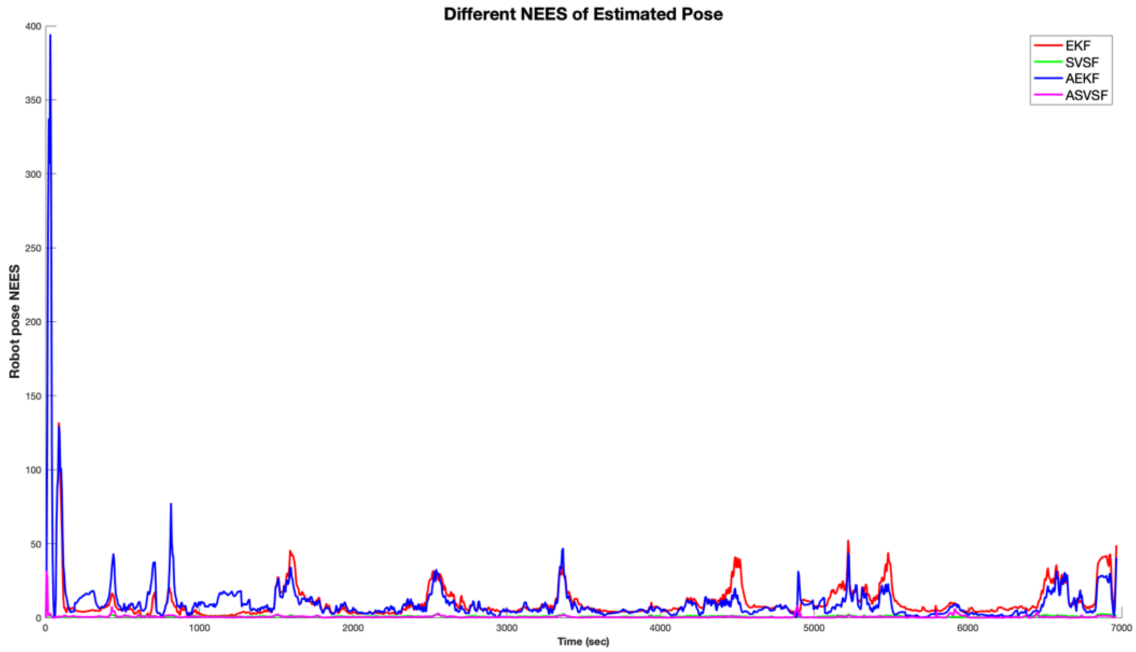


Figure 5.14 Average NEES for Robot Pose of Different Algorithms

Figure 5.14 presents the graphic of average NEES for each step performing the SLAM-algorithm. However, the validation cannot be easily conducted only by referring to this result. Accordingly, each performance is separately validated by involving the firstly mentioned interval, [0.1198, 9.7218]. It is computed based on the N-runs=1 of Monte Carlo Simulation.

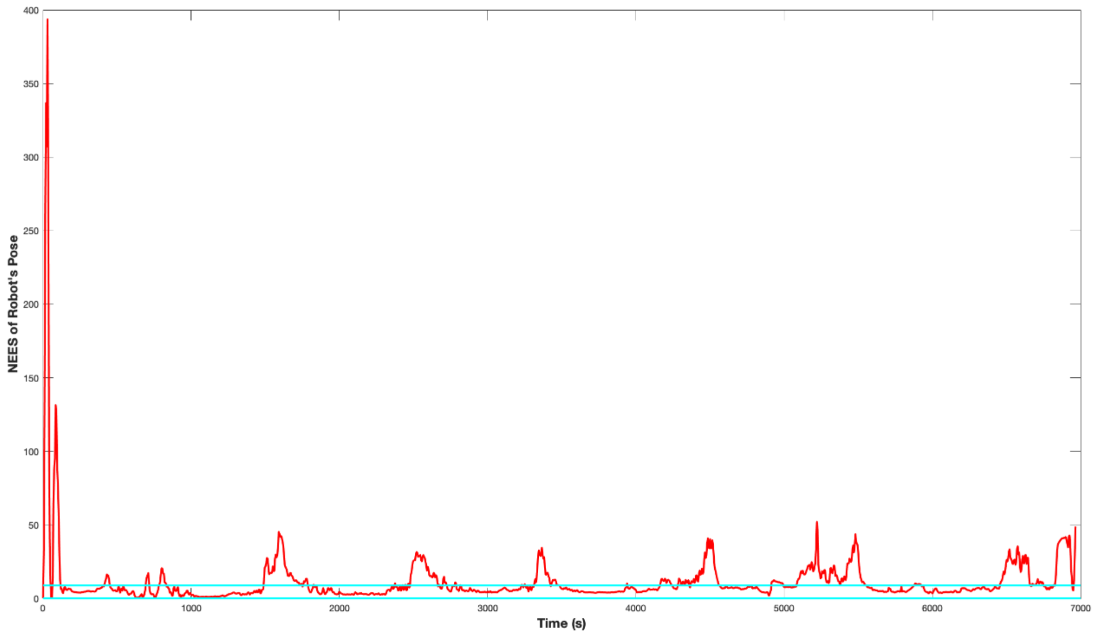


Figure 5.15 Average NEES of Robot Pose for EKF-SLAM Algorithm with two-sided 95% region (N=1)

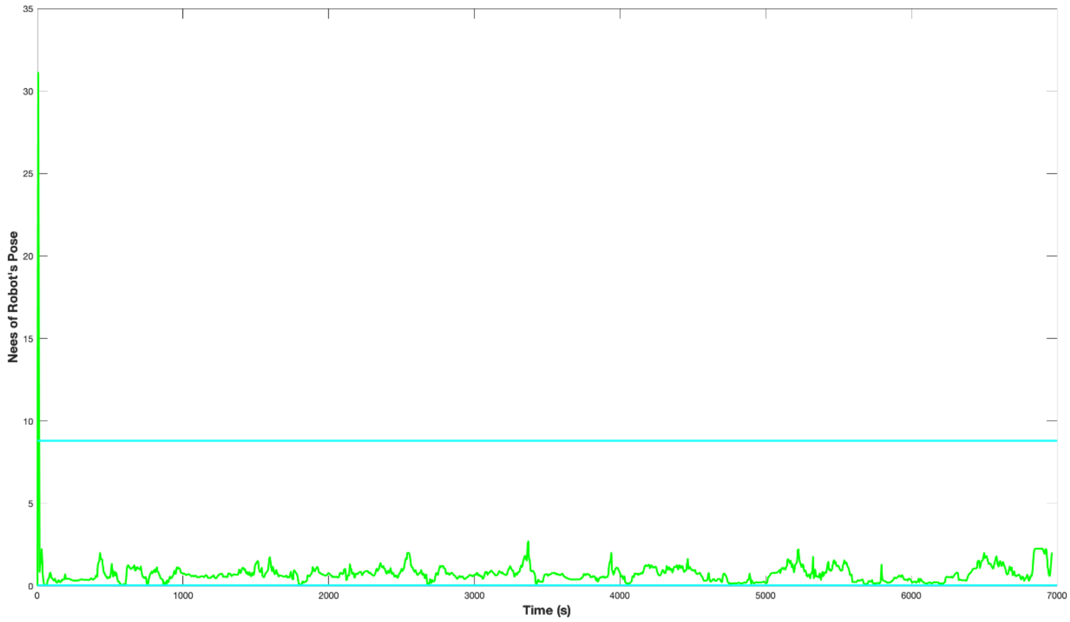


Figure 5.16 NEES's Average of Robot Pose for SVSF-SLAM Algorithm

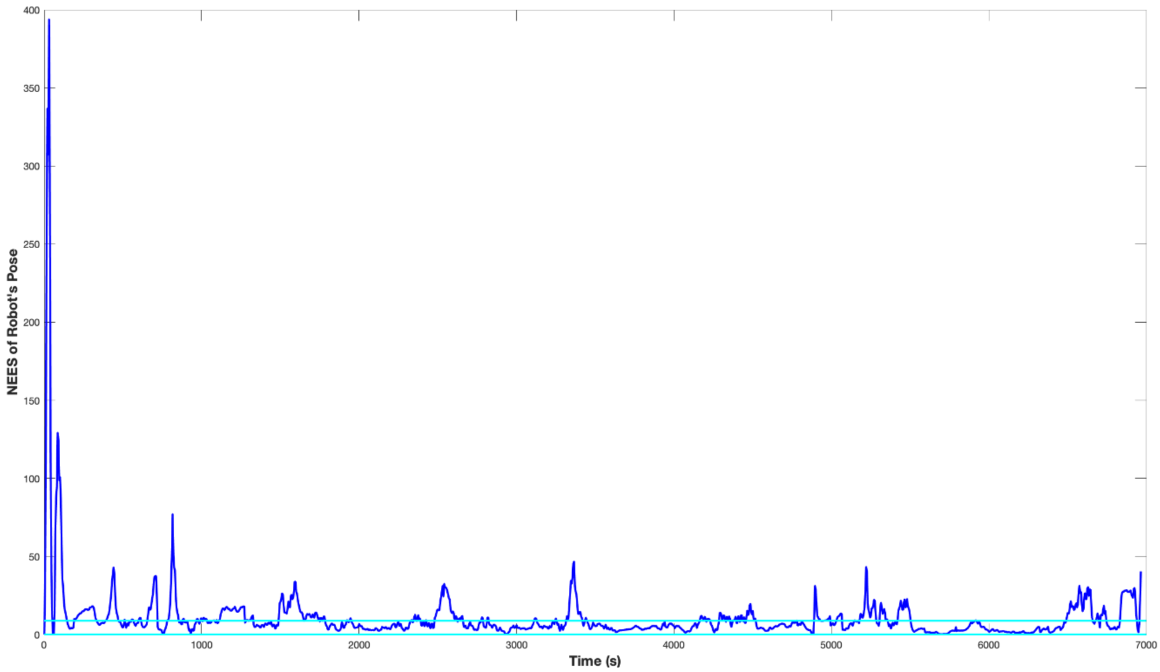


Figure 5.17 NEES's Average of Robot's Pose for AEKF-SLAM Algorithm

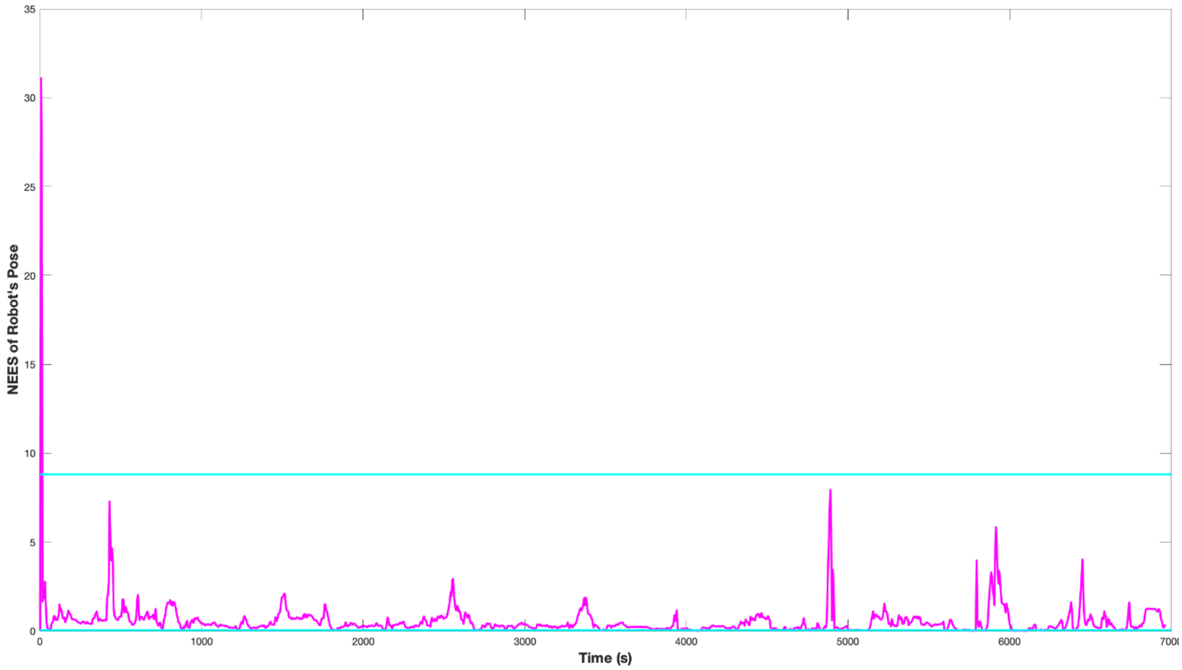


Figure 5.18 NEES's Average of Robot Pose for ASVSF-SLAM Algorithm

Figure 5.15 – Figure 5.18 shows the average NEES of Robot Pose for EKF-SLAM, AEKF-SLAM, SVSF-SLAM, and ASVSF-SLAM algorithm. According to the graphical representations with cyan-color bounds as the benchmark to evaluate, in short or single-run Monte Carlo Simulation, only the proposed method (adaptive filtering) remains consistent. Contrary, based on Figure 5.15 and Figure 5.17, both the EKF-SLAM and SVSF-SLAM show their inconsistency in estimating the robot pose. However, indicated by the values which frequently rises the upper bound, the SVSF-SLAM algorithm is better than EKF-SLAM algorithm. Therefore, based on the real-time verification, our hypothesis is satisfied and true. IN which, the robustness offered by SVSF makes SVSF can replace the EKF-SLAM algorithm. Moreover, the adaptive filtering method can be alternatively used to improve the consistency of any filtering since it is designed with checking its optimal solution with an unbiased estimator (see Chapter 3). The following Table is presented in order to clearly compare all the algorithm.

Table 5.6 Comparative Result Based on Average RMSE and NEES for different algorithm

Metrics	EKF-SLAM	SVSF-SLAM	AEKF-SLAM	ASVSF-SLAM
NEES Robot Pose	11.3658	10.6287	0.7887	0.5948
RMSE Robot Pose	5.1078	4.5829	2.6711	1.9346
RMSE Robot Heading	0.0803	0.0820	0.0551	0.0438
RMSE of Landmark	81.5989	80.0655	77.1436	74.5689

According to Table 5.6 above, all the statement declared previously is proven. All the average NEES analysis is correct that the adaptive filtering is consistent compared with the conventional one. Table 5.6 also presents the average RMSE over the Monte Carlo Simulation. Based on these average values, the optimality of the proposed method, ASVSF-SLAM is also proven. It is shown from all its values which are lower than the other algorithms.

5.4 Verification Based on The Simulation with Fixed Velocity and Known Landmark in the Global Map

If the previous analysis is conducted referring to the average NEES of robot pose when N=1 in the real application, Victoria Park-based SLAM problem, the following verification is intended to ensure that the consistency of the proposed method also appropriate for long-duration of Monte Carlo Simulation. For this case, it is assumed that the objective of all the algorithm presented in Chapter 3 is to estimate the following reference path and map.

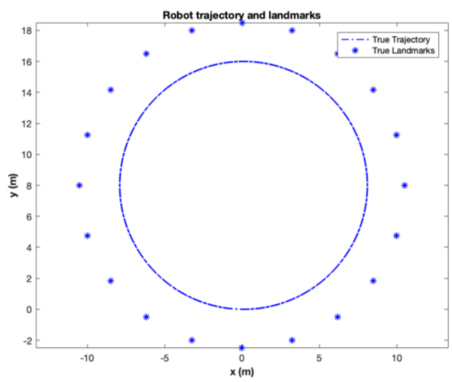


Figure 5.19 Reference Trajectory with 20 Static Features

Knowing the initial position of the robot in the global environment, the robot moves based on the constant linear 0.2 m/s and angular velocity 0.025 rad/s. Referring to the motion model and measurement model designed in the second creation of Chapter 4, the robot moves repeatedly over times of 500. It is evaluated under the Monte Carlo Simulation with N=200. Therefore, the acceptance region is bounded by the interval [2.6701 3.3491]. In other to illustrates that the simulation is realistic, the process is assumed to not accurate caused by the noise, the odometer used to sense the rotated wheels are also noisy, and the measurement sensor is noisy. For this reason, the characteristic noise statistics are defined in this simulation; q=[0.0071;0.0283] corresponding to the linear (m/s) and angular velocity (rad/s) and Q=[0.0071^2 0;0 0.0283^2] corresponding to its covariance. Meanwhile, the observation noise perturbating the measurement of z is defined r=[1;10] relative to the distance (m) and bearing (deg), and $R = [1^2\ 0; 0\ 10\pi/180^2]$. Theoretically, these noises are recursively updated when implementing the EKF-SLAM and SVSF-SLAM. Moreover, the rest parameter such as γ and initial error are defined as the same used in the previous verification. Contrary, they are kept constant/invariant. Based on this parameterization the following result is presented.

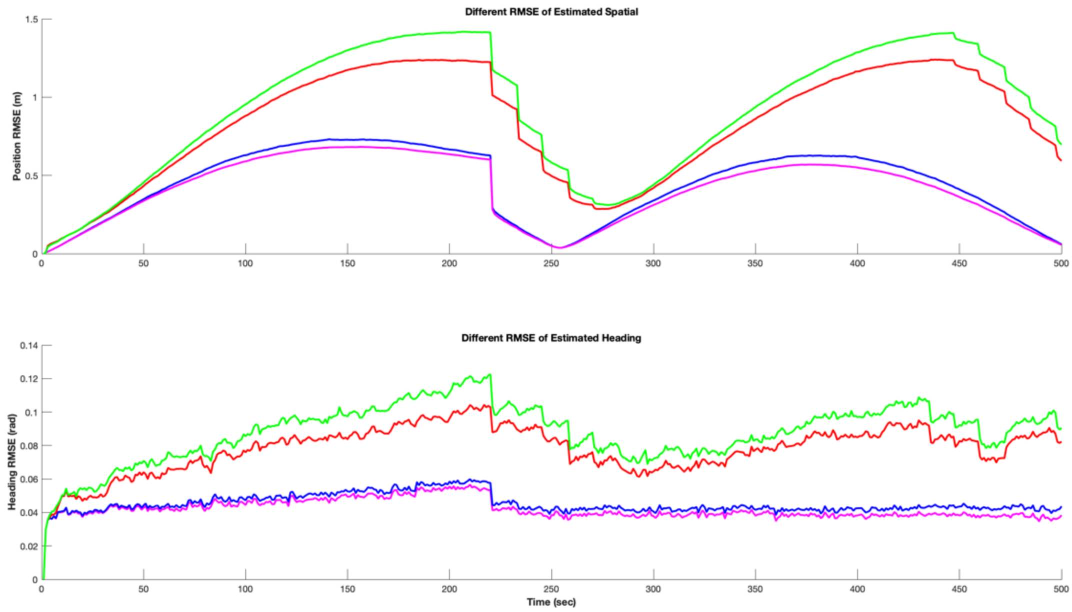


Figure 5.20 Average RMSE of Robot Pose and Heading under Monte Carlo Simulation with N=200

Based on the result depicted in Figure 5.20, the proposed methods present a better performance and accuracy of estimating the robot pose and heading. It is proven and indicated by the smaller values for both the average RMSE whole the step. Although, it again validates the effectiveness and optimality of ASVSF-SLAM algorithm, its consistency cannot be evaluated and analyzed according to this result. For this reason, the corresponding average NEES is presented as follows.

.

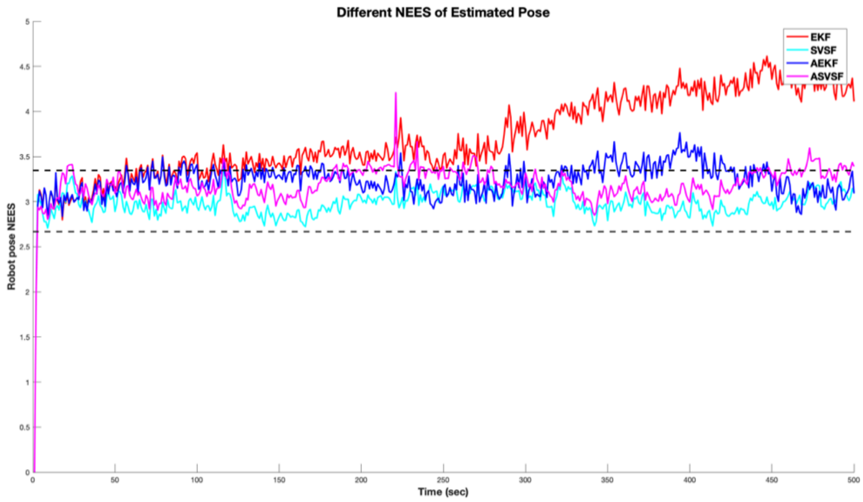


Figure 5.21 Average NEES of Robot Pose and Heading under Monte Carlo Simulation with N=200

Figure 5.21 shows that by bounding the graphical result with the determined acceptance regions, all the algorithms are relatively consistent except the EKF-SLAM algorithm. Like the previous way, the separately determined results are presented.

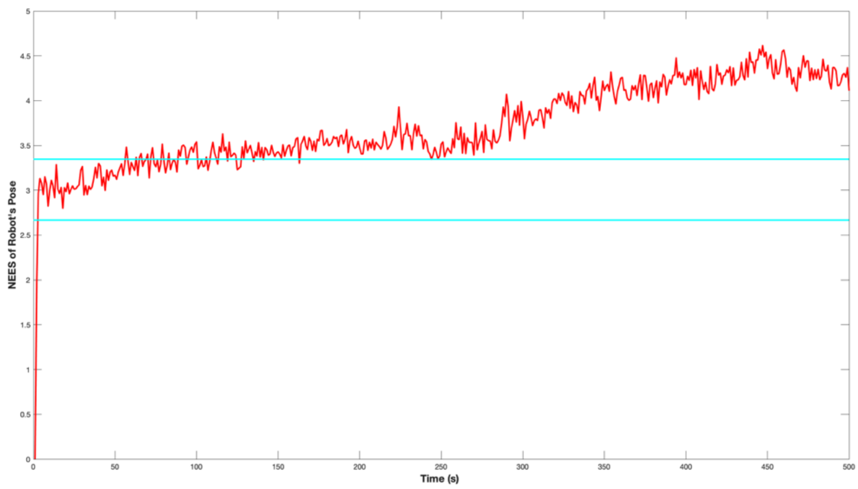


Figure 5.22 Average NEES given by the EKF-SLAM algorithm under the Monte Carlo Simulation with N=200

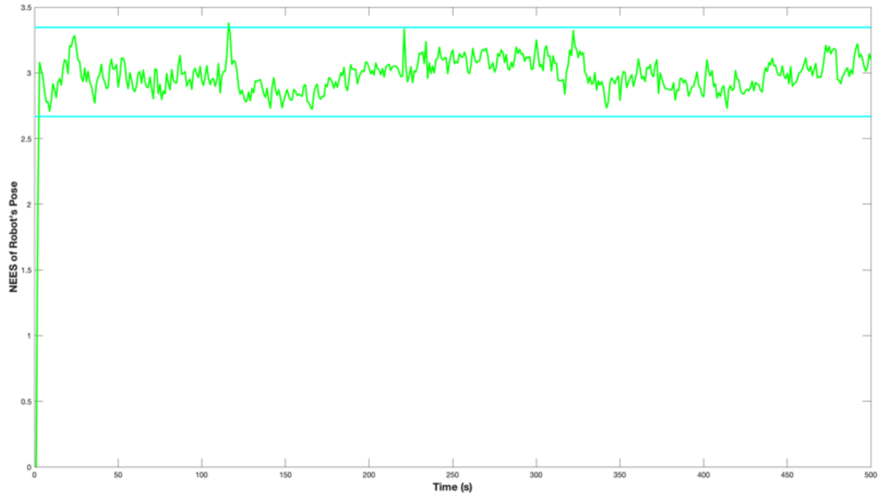


Figure 5.23 Average NEES given by the SVSF-SLAM algorithm under the Monte Carlo Simulation with N=200

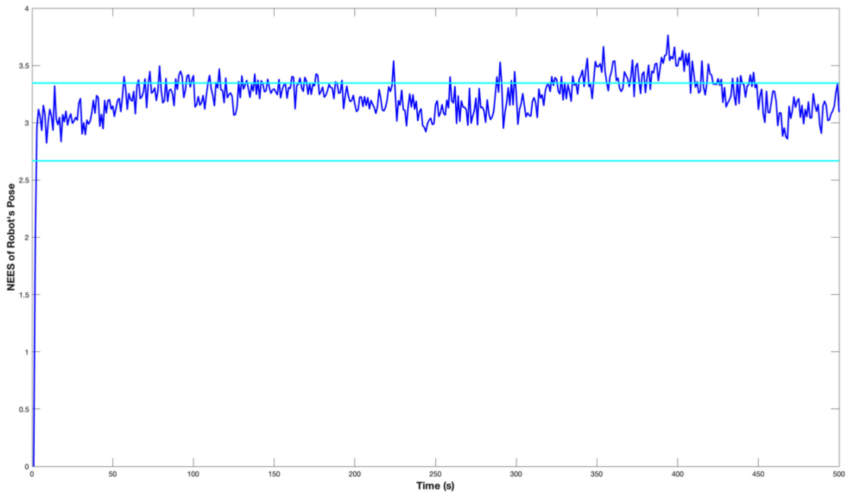


Figure 5.24 Average NEES given by the AEKF-SLAM algorithm under the Monte Carlo Simulation with N=200

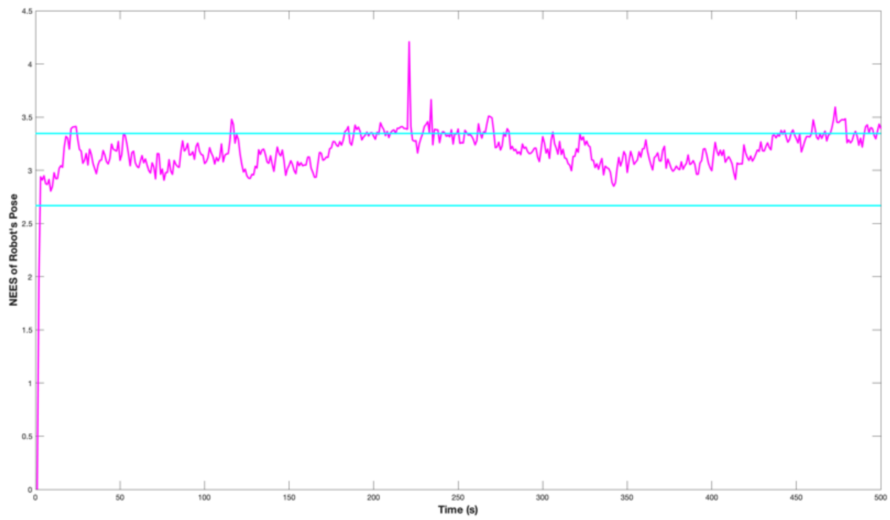


Figure 5.25 Average NEES given by the ASVSF-SLAM algorithm under the Monte Carlo Simulation with N=200

Now, it clear to declare that the consistency of all algorithm except the EKF-SLAM are guaranteed. To more extensive present the proof of the effectiveness, accuracy, consistency of the proposed algorithm, Table 5.7 is presented.

Table 5.7 Average RMSE and NEES of all algorithms

Metrics	EKF-SLAM	SVSF-SLAM	AEKF-SLAM	ASVSF-SLAM
NEES Robot Pose	3.7163	2.9782	3.2315	3.1853
RMSE Robot Pose	0.4449	0.8020	0.4217	0.9254
RMSE Robot Heading	0.4449	0.0762	0.4217	0.9254
NEES of Landmark	2.3166	1.9010	1.8357	1.7499
RMSE of Landmark	0.4807	0.4650	0.4467	0.4208

Chapter VI Conclusion

6.1 Summary

The popular filtering method commonly used to solve the feature-based SLAM algorithm is the Extended Kalman Filter. Alternatively, the role of robust filtering type, Smooth Variable Structure Filter, can replace EKF as the main state and parameter estimation. However, the original formulation of Extended Kalman Filter and Smooth Variable Structure is not completed with an ability to recursively estimate the noise statistic of the process and measurement as well as their corresponding covariance. Consequently, when they are predetermined and kept to be constant throughout the estimation process, the filter has a big possibility leading to the divergence condition respecting the original reference. For this reason, certain modification is strongly recommended to be concerned before using an either optimal and robust estimator. There are many types of modifications, as mentioned and introduced in the previous chapter. And the most popular one is tuning the corrective gain through the adaptive noise statistic, which is termed-well as an adaptive filter. Herein this dissertation, four different adaptive filters distinguished by the presence of guarantor techniques, Divergence Suppression Method and Innovation Covariance Estimator, are discussed, aiming to either improve EKF and SVSF. It is separately done by involving the Maximum A Posterior together with Weighted Exponent and Maximum Likelihood Estimator together with Expectation-Maximum Creation. The detailed derivation obtaining the adaptive form of EKF and SVSF is discussed. Furthermore, by adopting some completeness used for the feature-based SLAM problem, differently, there are converted to be a SLAM algorithm. Furthermore, they are realistically simulated with the presence of small additive noise and their corresponding covariance following the process and measurement. Of course, this analogy is intended to satisfy the probability condition caused by uncertainty. By using RMSE for the estimated path coordinate and the estimated map coordinated, different SLAM algorithm is compared and evaluated.

Accordingly, as the proposed strategies, both EKF and SVSF equipped with an adaptive manner to estimate the noise statistic, shows better and significant improvement than their conventional algorithm. Based on the lastly previous chapter, their effectiveness in term of accuracy and consistency are validated. The contribution containing in this dissertation can be repeatedly presented as follows.

6.2 Future Research

As mentioned earlier in the previous Chapters, theoretically, the Smooth Variable Structure Filter can be combined with any filtering method such as Extended Kalman Filter, Quadrature Kalman Filter, Cubature Kalman Filter, etc., based on the smoothing boundary layer. Essentially, its mechanism allows selecting the correspondingly appropriated gain by initially comparing it with the boundary limit variable. By this statement, the author plans to combine the Adaptive Smooth Variable Structure Filter and Adaptive Extended Kalman Filter. Besides that, the use of different stability and robust guarantor makes the adaptive filter with divergence suppression methods, and innovation covariance estimation is characteristically different. Therefore, mixing both adaptive filters can give a hybrid formulation, which, of course, improves the conventional filter. As the second plan, the author plans to design the hybrid filter based on Extended Kalman Filter and Smooth Variable Structure Filter. Afterward, the results are applied to alternatively and effectively solving the feature-based SLAM problem of wheeled mobile robot.

6.3 List of Related Publication

1. A MAPEKF-SLAM Algorithm with Recursive Mean and Covariance of Process and Measurement Noise Statistic, *Sinergi*, 2020
2. An AEKF-SLAM Algorithm with Recursive Noise Statistic Based on MLE and EM, *Journal of Intelligent & Robotic Systems*, 2019

3. An ASVSF-SLAM Algorithm with Time-Varying Noise Statistics Based on MAP Creation and Weighted Exponent, *Mathematical Problems in Engineering*, 2019.
4. Maximum likelihood estimation-assisted ASVSF through state covariance-based 2D SLAM algorithm, TELKOMNIKA Telecommunication, Computing, Electronics and Control, 2020
5. Adaptive Development of SVSF for A Feature-Based SLAM algorithm using Maximum Likelihood Estimation and Expectation and Maximization, IIUM Engineering Journal, 2020

Appendix

A. Smoothed Formulation of EKF based on one-step smoothing point

$$\breve{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}) + q_{k-1} \tag{A.1}$$

$$\breve{z}_{k|k-1} = h(\breve{x}_{k|k-1}) + r_k \tag{A.2}$$

$$e_{\breve{z},k|k-1} = z_k - \breve{z}_{k|k-1} \tag{A.3}$$

$$S = HP_{k|k-1}H^T + R_k \tag{A.4}$$

$$K = P_{k|k-1}H^T(S)^{-1} \tag{A.5}$$

$$\hat{x}_{k-1|k} = \hat{x}_{k-1|k-1} + Ke_{\breve{z},k|k-1} \tag{A.6}$$

Referring to Equation (A.1) – Equation (A.6), the estimate value of $\hat{x}_{k-1|k}$ can now replace the term of $\hat{x}_{k-1|k-1}$ in the original form of EKF, then the rest forms of smoothed EKF are chained as follows

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k}, u_{k-1}) + q_{k-1} \tag{A.7}$$

$$\hat{z}_{k|k-1} = h(\hat{x}_{k|k-1}) + r_k \tag{A.8}$$

$$e_{z,k|k-1} = z_k - \hat{z}_{k|k-1} \tag{A.9}$$

$$S_k = HP_{k|k-1}H^T + R_k \tag{A.10}$$

$$K_k = P_{k|k-1}H^T(S_k)^{-1} \tag{A.11}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k e_{z,k|k-1} \tag{A.12}$$

$$P_{k|k} = P_{k|k-1}(I - K_k H_k) \tag{A.13}$$

Alternatively, it can be done using, the RTS smoother technique as presented below. Given the Kalman Gain in Equation (A.11), the smoothed forms are

$$\hat{x}_{k|n} = \hat{x}_{k|k} + M_k(\hat{x}_{k|n-1} - \hat{x}_{k|k-1}) \tag{A.13}$$

$$P_{k|n} = P_{k|k} + M_k(P_{k|n-1} - P_{k|k-1})M_k^T \tag{A.13}$$

Where

$$M_k = P_{k|k-1}\bar{F}_k^T P_{k|k-1}^T \tag{A.13}$$

$$\bar{F}_k = F - K_k H \tag{A.13}$$

B. **Smoothed Formulation of SVSF based on one-step smoothing point**

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}) + q_{k-1} \tag{B.1}$$

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q_{k-1} \tag{B.2}$$

$$\hat{z}_{k|k-1} = h(\hat{x}_{k|k-1}) + r_k \tag{B.3}$$

$$\breve{e}_{z,k|k-1} = z_k - \hat{z}_{k|k-1} \tag{B.4}$$

$$A = \left(|\breve{e}_{z,k|k-1}|_{abs} + \gamma|e_{z,k-1|k-1}|_{abs}\right) \tag{B.5}$$

$$S_k = HP_{k|k-1}H^T + R_k \tag{B.6}$$

$$\psi_k = \left(\overline{\overline{A}}^{-1} HP_{k|k-1}H^T S_k^{-1}\right)^{-1} \tag{B.7}$$

$$sat[\psi^{-1}(\overline{\overline{\breve{e}_{z,k|k-1}}}] = \begin{cases} 1 \ , \ \psi^{-1}\overline{\overline{\breve{e}_{z,k|k-1}}} \geq 1 \\ \psi^{-1}\overline{\overline{\breve{e}_{z,k|k-1}}} \ , \ -1 < \psi^{-1}\overline{\overline{\breve{e}_{z,k|k-1}}} < 1 \\ -1 \ , \ \psi^{-1}\overline{\overline{\breve{e}_{z,k|k-1}}} \leq 1 \end{cases} \tag{B.8}$$

$$K_k^{SVSF} = H^+\left\{\overline{\overline{A}}^{-1} \circ sat[\psi^{-1}\overline{\overline{\breve{e}_{z,k|k-1}}}]\right\}[\overline{\overline{\breve{e}_{z,k|k-1}}}]^{-1} \tag{B.9}$$

$$\hat{x}_{k-1|k} = \hat{x}_{k-1|k-1} + K_k^{SVSF}\breve{e}_{z,k|k-1} \tag{B.10}$$

then considering that the prior state $\hat{x}_{k-1|k}$ replaces the term of $\hat{x}_{k-1|k-1}$ in the normal SVSF, the rest part of modified SVSF are chained as follows

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k}) + q_{k-1} \tag{B.11}$$

$$\hat{z}_{k|k-1} = h(\hat{x}_{k|k-1}) + r_k \tag{B.12}$$

$$e_{z,k|k-1} = z_k - \hat{z}_{k|k-1} \tag{B.13}$$

$$S_k = HP_{k|k-1}H^T + R_k \tag{B.14}$$

$$A = \left(|e_{z,k|k-1}|_{abs} + \gamma|e_{z,k-1|k-1}|_{abs}\right) \tag{B.15}$$

$$\psi_k = \left(\overline{\overline{A}}^{-1} H P_{k|k-1} H^T S_k^{-1}\right)^{-1} \tag{B.16}$$

$$sat[\psi^{-1}\overline{\overline{e_{z,k|k-1}}}] = \begin{cases} 1 \ , & \psi^{-1}\overline{\overline{e_{z,k|k-1}}} \geq 1 \\ \psi^{-1}\overline{\overline{e_{z,k|k-1}}}) \ , & -1 < \psi^{-1}\overline{\overline{e_{z,k|k-1}}} < 1 \\ -1 \ , & \psi^{-1}\overline{\overline{e_{z,k|k-1}}} \leq 1 \end{cases} \tag{B.17}$$

$$K_k^{SVSF} = H^+ \left\{\overline{\overline{A}}^{-1} \circ sat\left[\psi^{-1}\overline{\overline{e_{z,k|k-1}}}\right]\right\} \left[\overline{\overline{e_{z,k|k-1}}}\right]^{-1} \tag{B.18}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k^{SVSF} e_{z,k|k-1} \tag{B.19}$$

$$P_{k|k} = (I - HK_k^{SVSF})e_{z,k|k-1}P_{k|k-1}(I - HK_k^{SVSF})^T + K_k^{SVSF}R_kK_k^{SVSF\,T} \tag{B.20}$$

$$e_{z,k|k} = z_k - h(\hat{x}_{k|k}) \tag{B.21}$$

Alternatively, these smoothed forms can be compactly computed by referring to RTS principle. Given the SVSF Gain in Equation (A.18) and recalled as *K*, the smoothed forms are

$$\hat{x}_{k|n} = \hat{x}_{k|k} + M_k(\hat{x}_{k|n-1} - \hat{x}_{k|k-1}) \tag{B.22}$$

$$P_{k|n} = P_{k|k} + M_k(P_{k|n-1} - P_{k|k-1})M_k^T \tag{B.23}$$

Where

$$M_k = P_{k|k-1}\bar{F}_k^T P_{k|k-1}^T \tag{B.24}$$

$$\bar{F}_k = F - K_k H \tag{B.25}$$

C. **The Foundation of the Feature-Based SLAM Algorithms**

In the SLAM application, the full state vector is

$$x_k = \begin{bmatrix} x_{R,k} \\ x_{L,k}^i \end{bmatrix} \tag{C.1}$$

where $x_{R,k}$ represents the robot pose variable at time k consisting both the spatial location and its heading or orientation

$$x_{R,k} = \begin{bmatrix} x_{r,k} \\ y_{r,k} \\ \theta_{r,k} \end{bmatrix} \tag{C.2}$$

Meanwhile $x_{L,k}^i$ gives the information of the i-th landmark coordinate consisting both the coordinate respect to x-axes $x_{l,k}^i$ and y-axes $y_{l,k}^i$ for $i = 1, 2, \ldots, \ldots, N-1, N$ at time k

$$x_{L,k}^i = \begin{bmatrix} x_{l,k}^1 \\ y_{l,k}^1 \\ \cdots \\ \cdots \\ x_{l,k}^N \\ y_{l,k}^N \end{bmatrix} \tag{C.3}$$

Where N is the number of landmark available on the global coordinate system as the point-based map. Thus, Equation C.1 becomes

$$x_k = \begin{bmatrix} x_{r,k} \\ y_{r,k} \\ \theta_{r,k} \\ x_{l,k}^1 \\ y_{l,k}^1 \\ \cdots \\ \cdots \\ x_{l,k}^N \\ y_{l,k}^N \end{bmatrix} \tag{C.4}$$

Suppose every landmark is collected with a signature s_l^i, the full state vector becomes

$$x_k = \begin{bmatrix} x_{r,k} \\ y_{r,k} \\ \theta_{r,k} \\ x_{l,k}^1 \\ y_{l,k}^1 \\ s_{l,k}^1 \\ \cdots \\ \cdots \\ x_{l,k}^N \\ y_{l,k}^N \\ s_{l,k}^N \end{bmatrix} \qquad \text{(C.5)}$$

Equation (C.5) shows that the full state vector has dimension of $(3N + 3)$ for N is the number of landmark available in the map. For this reason, the dimension of its covariance is $(3N + 3) \times (3N + 3)$. Furthermore, as the common way in designing the feature-SLAM algorithm, the initial pose is assumed to be origin. It means that the robot pose has information of its coordinate $x_{R,0} = [0, 0, 0]^T$ as well as it considers that there is no landmark seen at the time $k = 0$. Therefore, it is obvious to have an assumption that $x_{L,0} = [0, 0, 0, \ldots, \ldots, 0, 0, 0]^T$. Therefore, the initial setup of the full state vector is

$$x_0 = \begin{bmatrix} x_{R,0} \\ x_{L,0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dots \\ \dots \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{C.6}$$

Considering that N is infinity when the robot pose does not see any landmark yet in the origin pose, it is clear to describe its covariance P_0 as

$$P_0 = \begin{bmatrix} 0 & 0 & 0 & \infty & \dots & \infty \\ 0 & 0 & 0 & \infty & \dots & \infty \\ 0 & 0 & 0 & \infty & \dots & \infty \\ 0 & 0 & 0 & \infty & \dots & \infty \\ \infty & \infty & \infty & \infty & \dots & \infty \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \infty & \infty & \infty & \infty & \dots & \infty \end{bmatrix} \tag{C.7}$$

Prerequisites for Prediction Step in Optimal and Robust Filter

Next, it assumes, the robot is moves from the initial pose $x_{R,k=0}$ to the next coordinate $x_{R,k=k+1}$ since the robot execute the control command u_k. Then by involving the motion model and execute the control command, the next pose of robot is

$$x_k = x_{k-1} + \begin{bmatrix} \left(R+\frac{W}{2}\right)\left(sin\left(\theta_{r,k-1}+\alpha\right)\right)+sin(\alpha) \\ \left(R+\frac{W}{2}\right)\left(-cos\left(\theta_{r,k-1}+\alpha\right)+cos(\alpha)\right) \\ \alpha \\ 0 \\ 0 \\ 0 \\ \ldots \\ \ldots \\ \ldots \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{C.8}$$

Since the command has the same values of velocity, the another model of C.8 is

$$x_k = x_{k-1} + \begin{bmatrix} u_r cos\left(\theta_{r,k-1}\right) \\ u_r sin\left(\theta_{r,k-1}\right) \\ 0 \\ 0 \\ 0 \\ 0 \\ \ldots \\ \ldots \\ \ldots \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{C.9}$$

Compactly, it can be respectively modeled as

$$x_k = x_{k-1} + A_x^T \begin{bmatrix} \left(R+\frac{W}{2}\right)\left(sin\left(\theta_{r,k-1}+\alpha\right)\right)-sin(\theta_{r,k-1}) \\ \left(R+\frac{W}{2}\right)\left(-cos\left(\theta_{r,k-1}+\alpha\right)+cos(\theta_{r,k-1})\right) \\ \alpha \end{bmatrix} \tag{C.10}$$

and

$$x_k = x_{k-1} + A_x^T \begin{bmatrix} u_r cos\left(\theta_{r,k-1}\right) \\ u_r sin\left(\theta_{r,k-1}\right) \\ 0 \end{bmatrix} \tag{C.11}$$

For A_x is satisfying the dimension of $(3N+3)$. It is expressed as

$$A_x = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ & & & \underbrace{\qquad\qquad}_{3N\ Columns} & & \end{bmatrix} \tag{C.12}$$

Clearly, Equation (C.1) – Equation (C.12) are the foundation to apply the prediction state.

$$x_k = f(x_{k-1}, u_k) \tag{C.13}$$

Assuming that all the landmarks are static, and no affected by the control command. The motion model is expressed as follows

$$x_k = \begin{bmatrix} x_{r,k-1} \\ y_{r,k-1} \\ \theta_{r,k-1} \end{bmatrix} + \begin{bmatrix} \left(R+\frac{W}{2}\right)(sin\,(\theta_{r,k-1}+\alpha)) - sin(\theta_{r,k-1}) \\ \left(R+\frac{W}{2}\right)(-cos\,(\theta_{r,k-1}+\alpha) + cos(\theta_{r,k-1})) \\ \alpha \end{bmatrix} \tag{C.14}$$

$$x_k = \begin{bmatrix} x_{r,k-1} \\ y_{r,k-1} \\ \theta_{r,k-1} \end{bmatrix} + \begin{bmatrix} u_r cos\,(\theta_{r,k-1}) \\ u_r sin\,(\theta_{r,k-1}) \\ 0 \end{bmatrix} \tag{C.15}$$

It shows that only three variables in the old state are changed after including the motion model. Accordingly, the Jacobian matrices of $f(\cdot)$ with respect to the state F_s and with respect to the control command F_c can respectively be calculated as

$$F_s = \begin{bmatrix} \frac{\partial x_{r,k}}{\partial x_{r,k-1}} & \frac{\partial x_{r,k}}{\partial y_{r,k-1}} & \frac{\partial x_{r,k}}{\partial \theta_{r,k-1}} \\ \frac{\partial y_{r,k}}{\partial x_{r,k-1}} & \frac{\partial y_{r,k}}{\partial y_{r,k-1}} & \frac{\partial y_{r,k}}{\partial \theta_{r,k-1}} \\ \frac{\partial \theta_{r,k}}{\partial x_{r,k-1}} & \frac{\partial \theta_{r,k}}{\partial y_{r,k-1}} & \frac{\partial \theta_{r,k}}{\partial \theta_{r,k-1}} \end{bmatrix} \quad \text{C.16)}$$

$$F_c = \begin{bmatrix} \frac{\partial x_{r,k}}{\partial u_l} & \frac{\partial x_{r,k}}{\partial u_r} \\ \frac{\partial y_{r,k}}{\partial u_l} & \frac{\partial y_{r,k}}{\partial u_r} \\ \frac{\partial \theta_{r,k}}{\partial u_l} & \frac{\partial \theta_{r,k}}{\partial u_r} \end{bmatrix} \tag{C.17}$$

Referring to Equation (C.14) and Equation (C.15), it is clear to have

$$x_{r,k} = x_{r,k-1} + \left(R + \frac{W}{2}\right)(sin\,(\theta_{r,k-1}+\alpha)) - sin(\theta_{r,k-1}) \tag{C.18}$$

$$y_{r,k} = y_{r,k-1} + \left(R + \frac{W}{2}\right)(-cos\,(\theta_{r,k-1}+\alpha) + cos(\theta_{r,k-1})) \tag{C.19}$$

$$\theta_{r,k} = \theta_{r,k-1} + \alpha \tag{C.20}$$

and applying both partial derivative of function $f(\cdot)$ with respect to the stateas well as with respect to the control, two possible corresponding results classified based on similarity of right wheel and left wheel velocity are described as follows

$$F_s^A = \begin{bmatrix} 1 & 0 & \left(R+\frac{W}{2}\right)\left(cos(\theta_{r,k-1}+\alpha)-cos(\theta_{r,k-1})\right) \\ 0 & 1 & \left(R+\frac{W}{2}\right)\left(sin(\theta_{r,k-1}+\alpha)-sin(\theta_{r,k-1})\right) \\ 0 & 0 & 1 \end{bmatrix} \tag{C.21}$$

It is noted that Equation (C.21) represents the partial derivative of $f(\cdot)$ with respect to the state when $u_r \neq u_l$, denoted by F_s^A. Contrary, when $u_r = u_l$, the notation becomes F_s^B. It aims to differs the formulation based on the random movement caused by the control command. The randomness is because of the presence of unpredictable noise following the turn and move as the influence factor. From Equation (C.21) can seen that the entries on this matrix only involves the cause of coordinate changer. Therefore, F_s^B should not be much difference to F_s^A . In order to calculate it, let's observe the cause of motion for $x_{r,k}$ by knowledge that $\alpha = 0$ when $u_r = u_l$ (see Chapter 2).

$$\begin{aligned} &(R+\frac{W}{2}\left(cos(\theta_{r,k-1}+\alpha)-cos(\theta_{r,k-1})\right) \\ &=R\left(cos(\theta_{r,k-1}+\alpha)-cos(\theta_{r,k-1})\right)+\frac{W}{2}\left(cos(\theta_{r,k-1}+\alpha)-cos(\theta_{r,k-1}\right. \end{aligned} \tag{C.22}$$

Since $\alpha = 0$, Equation (C.22) seems to return zero. It causes obtaining the partial derivative of $f(\cdot)$ with respect to the state when $u_r = u_l$ becomes unobservable. For this reason, all the expandable variables in Equation (C.22) is observed. It leads to the original formulation of R. In which it is the variable representing the ratio of u_l and α, $R = \frac{u_l}{\alpha}$. For this statement it is clear to have a definition that once $\alpha = 0$, R is infinity. Therefore, to calculate (C.22) as the effort find the entire in matrix F_s^B, the limit approach is involved. It can be clearly derived as follows

$$\lim_{\alpha \to 0} u_l \cdot \frac{cos(\theta_{r,k-1}+\alpha)-cos(\theta_{r,k-1})}{\alpha} \tag{C.23}$$

Using l'hopital French mathematician, it is clear to get

$$\lim_{\alpha \to 0} u_l \ . \ -\frac{sin(\theta_{r,k-1}+\alpha)}{1} = -u_l \ . \ sin(\theta_{r,k-1}) \tag{C.24}$$

Similarly, by recalling $\left(R+\frac{W}{2}\right)(sin(\theta_{r,k-1}+\alpha)-sin(\theta_{r,k-1}))$ for $u_r = u_l$, it gives the element of matrix which equation to $u_l \ . \ cos(\theta_{r,k-1})$. Therefore, F_s^B can be described as follows

$$F_s^B = \begin{bmatrix} 1 & 0 & -u_l \ . \ sin(\theta_{r,k-1}) \\ 0 & 1 & u_l \ . \ cos(\theta_{r,k-1}) \\ 0 & 0 & 1 \end{bmatrix} \tag{C.25}$$

Up to this point, the Jacobian matrix of $f(\cdot)$ relative to the state has been calculated. Next, let's calculate the partial derivative of f(.) with respect to the control. Unfortunately, by referring to the definition of finding the partial derivative of all the element is respected to u_r and u_l, and there is no such visible u_l or u_l. For this reason, all the derived equation from the motion model in Chapter 2 is recalled. They are chained as follows

$$u_l = \alpha * R \tag{C.26}$$

$$u_r = \alpha * (W+R) \tag{C.27}$$

$$u_r - u_l = \alpha(R) - \alpha(W+R) = \alpha W \tag{C.28}$$

$$\alpha = \tfrac{u_r - u_l}{W} \tag{C.29}$$

$$R = \frac{u_l}{\alpha} \tag{C.30}$$

From Equation (C.26) – Equation (C.30), the following equation can be augmented.

$$\alpha = \tfrac{u_r - u_l}{W} \tag{C.31}$$

$$R = \frac{u_l W}{u_r - u_l} \tag{C.32}$$

$$R + \frac{W}{2} = \frac{u_l W}{u_r - W} + \frac{W}{2} = \frac{W}{2} . \frac{u_r + u_l}{u_r - u_l} \tag{C.33}$$

Now Equation (C.13) can be expanded as

$$x_k = \begin{bmatrix} x_{r,k-1} \\ y_{r,k-1} \\ \theta_{r,k-1} \end{bmatrix} + \begin{bmatrix} \left(R + \frac{W}{2}\right)\left(sin\left(\theta^{'}_{r,k-1}\right) - sin(\theta_{r,k-1})\right) \\ \left(R + \frac{W}{2}\right)\left(-cos\left(\theta^{'}_{r,k-1}\right) + cos(\theta_{r,k-1})\right) \\ \alpha \end{bmatrix} \tag{C.34}$$

And referring to Equation (C.26) – Equation (C.33), all the element of the robot pose $x_{r,k}$ are

$$x_{r,k} = x_{r,k-1} + \left(\frac{W}{2} \cdot \frac{u_r + u_l}{u_r - u_l}\right)\left(sin\left(\theta^{'}_{r,k-1}\right) - sin(\theta_{r,k-1})\right) \tag{C.35}$$

$$y_{r,k} = y_{r,k-1} + \left(\frac{W}{2} \cdot \frac{u_r + u_l}{u_r - u_l}\right)\left(-cos\left(\theta^{'}_{r,k-1}\right) + cos(\theta_{r,k-1})\right) \tag{C.36}$$

$$\theta_{r,k} = \theta^{'}_{r,k-1} \tag{C.37}$$

Once $\theta^{'}_{r,k-1} = \theta_{r,k-1} + \alpha$ then the partial derivative of $f(.)$ with respect to the control command can respectively be calculated as follows

$$\frac{\partial x_{r,k}}{\partial u_l} = \frac{WR}{(u_r - u_l)^2}\left(sin(\theta^{'}_{r,k-1} - sin(\theta_{r,k-1}))\right) - \frac{u_r + u_l}{2(u_r - u_l)} cos(\theta^{'}_{r,k-1}) \tag{C.38}$$

$$\frac{\partial y_{r,k}}{\partial u_l} = \frac{WR}{(u_r - u_l)^2}\left(-cos(\theta^{'}_{r,k-1} + cos(\theta_{r,k-1}))\right) - \frac{u_r + u_l}{2(u_r - u_l)} sin(\theta^{'}_{r,k-} \tag{C.39}$$

$$\frac{\partial \theta_{r,k}}{\partial u_l} = -\frac{1}{W} \tag{C.40}$$

$$\frac{\partial x_{r,k}}{\partial u_r} = -\frac{WR}{(u_r - u_l)^2}\left(sin(\theta^{'}_{r,k-1} - sin(\theta_{r,k-1}))\right) + \frac{u_r + u_l}{2(u_r - u_l)} cos(\theta^{'}_{r,k-} \tag{C.41}$$

$$\frac{\partial y_{r,k}}{\partial u_r} = -\frac{WR}{(u_r - u_l)^2}\left(-cos(\theta^{'}_{r,k-1} + cos(\theta_{r,k-1}))\right) + \frac{u_r + u_l}{2(u_r - u_l)} sin(\theta^{'}_{r,k} \tag{C.42}$$

$$\frac{\partial \theta_{r,k}}{\partial u_r} = \frac{1}{W} \tag{C.43}$$

Note that Equation (C.38) – Equation (C.43) represent all the element on the partial derivative of $f(.)$ with respect to the control command, when $u_r \neq u_l$. Obviously, F_c^A is calculated. By the same way and referring to Equation (C.14), the chained formulation of all the element for F_c^B can also calculated as

$$\frac{\partial x_{r,k}}{\partial u_l} = \frac{1}{2}\left(cos(\theta_{r,k-1}) + \frac{u_l}{W} sin(\theta_{r,k-1})\right) \tag{C.44}$$

$$\frac{\partial y_{r,k}}{\partial u_l} = \frac{1}{2}\left(sin(\theta_{r,k-1}) + \frac{u_l}{W}cos(\theta_{r,k-1})\right)) \tag{C.45}$$

$$\frac{\partial \theta_{r,k}}{\partial u_l} = 0 \tag{C.46}$$

$$\frac{\partial x_{r,k}}{\partial u_r} = \frac{1}{2}\left(cos(\theta_{r,k-1}) - \frac{u_l}{W}sin(\theta_{r,k-1})\right) \tag{C.47}$$

$$\frac{\partial y_{r,k}}{\partial u_r} = \frac{1}{2}\left(sin(\theta_{r,k-1}) + \frac{u_l}{W}cos(\theta_{r,k-1})\right) \tag{C.48}$$

$$\frac{\partial \theta_{r,k}}{\partial u_r} = 0 \tag{C.49}$$

Clearly, once Equation (C.12) is discussed, the first step in prediction stage is satisfied. Furthermore, the second step in prediction step is discussed and presented here. It is started by recalling the definition of covariance prediction step after the state is predicted. The form is

$$P_k = F_s P_{k-1} F_s^T + Q_k \tag{C.50}$$

By considering that the model of covariance is used to represent the uncertainty about its corresponding state vector, Equation (C.50) is completed with Q_k, which is the random effect caused by the turn and move factor. It is recalled as

$$Q_k = F_c \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} F_c^T \tag{C.51}$$

where σ_l and σ_r are direct variable obtained based on the relative effect. Now, by assuming that the number of landmark $N = 2$, then it is clear to have the partial derivate with respect to the state as

$$F_s = \begin{bmatrix} 1 & 0 & \frac{\partial x_{r,k}}{\partial \theta_{r,k-1}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{\partial y_{r,k}}{\partial \theta_{r,k-1}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.52}$$

In order to make Equation (5.1.52) to be compact formulation, it can also be modeled as

$$F_s = I + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \begin{bmatrix} 0 & 0 & \frac{\partial x_{r,k}}{\partial \theta_{r,k-1}} \\ 0 & 0 & \frac{\partial y_{r,k}}{\partial \theta_{r,k-1}} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.53}$$

By knowing that since $N = 2$, A_x is the following form

$$A_x = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.54}$$

Consequentially, the compact model of partial derivative with respect to the state F_s step can further be calculated as

$$F_s = I + A_x^T \begin{bmatrix} 0 & 0 & \frac{\partial x_{r,k}}{\partial \theta_{r,k-1}} \\ 0 & 0 & \frac{\partial y_{r,k}}{\partial \theta_{r,k-1}} \\ 0 & 0 & 0 \end{bmatrix} A_x \tag{C.55}$$

Note that $\frac{\partial x_{r,k}}{\partial \theta_{r,k-1}}$ and $\frac{\partial y_{r,k}}{\partial \theta_{r,k-1}}$ are non-zero elements calculated earlier under condition of $u_r = u_l$ or $u_r \neq u_l$, and F_s is now representing all the partial derivative of f(.) with respect to the control with N is the number landmark. The command u_r and u_l only influence the first three variable on the state transition. It gives assumption that all the partial derivate related to the landmark coordinate are now zero. Then, by considering that the number of landmarks is $N = 2$, it is obvious to have

$$F_{ce} = \begin{bmatrix} \frac{\partial x_{r,k}}{\partial u_l} & \frac{\partial x_{r,k}}{\partial u_r} \\ \frac{\partial y_{r,k}}{\partial u_l} & \frac{\partial y_{r,k}}{\partial u_r} \\ \frac{\partial \theta_{r,k}}{\partial u_l} & \frac{\partial \theta_{r,k}}{\partial u_r} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{C.56}$$

Next, by knowing that $\begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}$ is concerned as random effect caused by the turn and move factor, then Q_k in both optimal and robust prediction stage becomes

$$Q_k = F_{ce} \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} F_{ce}^T = \begin{bmatrix} \frac{\partial x_{r,k}}{\partial u_l} & \frac{\partial x_{r,k}}{\partial u_r} \\ \frac{\partial y_{r,k}}{\partial u_l} & \frac{\partial y_{r,k}}{\partial u_r} \\ \frac{\partial \theta_{r,k}}{\partial u_l} & \frac{\partial \theta_{r,k}}{\partial u_r} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} \begin{bmatrix} \frac{\partial x_{r,k}}{\partial u_l} & \frac{\partial x_{r,k}}{\partial u_r} \\ \frac{\partial y_{r,k}}{\partial u_l} & \frac{\partial y_{r,k}}{\partial u_r} \\ \frac{\partial \theta_{r,k}}{\partial u_l} & \frac{\partial \theta_{r,k}}{\partial u_r} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}^T \tag{C.57}$$

$$Q_k = F_{ce} \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} F_{ce}^T = \begin{bmatrix} * & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.58}$$

Where F_{ce} is extended F_c with a consideration of landmark coordinate. Meanwhile (3×3) non zeros matrix indicated by element $*$ in Equation (5.1.58) is

$$Q_{0,k} = F_c \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} F_c^T = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} = \begin{bmatrix} \frac{\partial x_{r,k}}{\partial u_l} & \frac{\partial x_{r,k}}{\partial u_r} \\ \frac{\partial y_{r,k}}{\partial u_l} & \frac{\partial y_{r,k}}{\partial u_r} \\ \frac{\partial \theta_{r,k}}{\partial u_l} & \frac{\partial \theta_{r,k}}{\partial u_r} \end{bmatrix} \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} \begin{bmatrix} \frac{\partial x_{r,k}}{\partial u_l} & \frac{\partial x_{r,k}}{\partial u_r} \\ \frac{\partial y_{r,k}}{\partial u_l} & \frac{\partial y_{r,k}}{\partial u_r} \\ \frac{\partial \theta_{r,k}}{\partial u_l} & \frac{\partial \theta_{r,k}}{\partial u_r} \end{bmatrix}^T \tag{C.59}$$

Where $Q_{0,k}$ is Q_k when F_c is calculated both relative to $u_r \neq u_l$ and $u_r = u_l$. As can be seen, Equation (C.59) can also be modeled as

$$Q_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.60}$$

By knowing that since $N = 2$, A_x is the following form

$$A_x = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.61}$$

Consequentially, to maps from simple size of matrix to the compact one when the number of landmarks is N, Q_k ca n be calculated as follows

$$Q_k = A_x^T Q_{0,k} A_x \tag{C.62}$$

Up to this point, all the completeness for the prediction step for both the optimal and robust filtering are satisfied.

Prerequisites before Performing the Correction Step in either Optimal and Robust filtering

Either optimal and robust filtering utilizes the measurement model as

$$z_k^i = \begin{bmatrix} \delta_k^i \\ \beta_k^i \end{bmatrix} \tag{C.63}$$

Knowing that the measurement model is going to give the range and bearing of laser scanner from the robot frame relative to the landmark, hence the direct point-based observation is recalled in this Chapter. It is modeled as

$$\begin{bmatrix} \delta_k^i \\ \beta_k^i \end{bmatrix} = \begin{bmatrix} \sqrt{\left(x_{l,k}^i - x_{ls,k}\right)^2 + \left(y_{l,k}^i - y_{ls,k}\right)^2} \\ atan\left(\dfrac{y_{l,k}^i - y_{ls,k}}{x_{l,k}^i - x_{ls,k}}\right) - \theta_{r,k} \end{bmatrix} \tag{C.64}$$

Where

$$\begin{bmatrix} x_{ls} \\ y_{ls} \end{bmatrix} = \begin{bmatrix} x_{r,k} \\ y_{r,k} \end{bmatrix} + d_{ls} \begin{bmatrix} sin(\theta_{r,k}) \\ cos(\theta_{r,k}) \end{bmatrix} \tag{C.65}$$

The measurement model of landmark is assumed to be noisy therefor it is clear to remodeled Equation (C.64) with an addition of random effect influencing both the range δ_k^i and bearing β_k^i. Supposing that r_δ and r_β are considered as the additive noise, respectively. The formulation of measurement z_k^i is

$$z_k^i = \begin{bmatrix} \delta_k^i \\ \beta_k^i \end{bmatrix} + \begin{bmatrix} r_\delta \\ r_\beta \end{bmatrix} \tag{C.66}$$

For i refer to the sequence detected landmark in the map. Now, it is clear since all the variables used for calculating Equation (C.63) – Equation (C.66) is adopted from the robot pose $x_R = [x_r \;\; y_r \;\; \theta_r]^T$ and depend to the real landmark coordinate $x_{L,k} = [x_{l,k}^i \;\; y_{l,k}^i]^T$ from the fix map, then obviously the function relative to the measurement is

$$z_k = h(x_{r,k}) + r_k \tag{C.67}$$

Where $x_{r,k}$ is the predicted state calculated in Equation (C.12) and $r = [r_\delta \;\; r_\beta]^T$ is additive noise relative to the measurement as mentioned in Equation (C.66). Next, similarly to the prediction step, the correction step for the optimal and robust also requires to compute the Jacobian matrix H, in which the partial derivative of z_k with respect to the corresponding state $x_{R,k}$. Knowing that $x_{R,k} = [x_{r,k} \;\; y_{r,k} \;\; \theta_{r,k}]^T$from the prediction step, then it is obvious to have

$$H = \frac{\partial z_k^i}{\partial x_{R,k}} = \begin{bmatrix} \frac{\partial \delta_k^i}{\partial x_{r,k}} & \frac{\partial \delta_k^i}{\partial y_{r,k}} & \frac{\partial \delta_k^i}{\partial \theta_{r,k}} \\ \frac{\partial \beta_k^i}{\partial x_{r,k}} & \frac{\partial \beta_k^i}{\partial y_{r,k}} & \frac{\partial \beta_k^i}{\partial \theta_{r,k}} \end{bmatrix} \tag{C.68}$$

Where correspondingly, all elements in Jacobian matrix H are presented as

$$\begin{aligned} \frac{\partial \delta_k^i}{\partial x_{r,k}} &= \frac{\partial}{\partial x_{r,k}} \cdot \left(\sqrt{\left(x_{l,k}^i - x_{ls,k}\right)^2 + \left(y_{l,k}^i - y_{ls,k}\right)^2} \right) \\ &= \frac{2\left(x_{l,k} - x_{ls}\right)(-1)}{2\sqrt{\left(x_{l,k} - x_{ls}\right)^2 + \left(y_{l,k} - y_{ls}\right)^2}} \end{aligned} \tag{C.69}$$

$$\begin{aligned} \frac{\partial \delta_k^i}{\partial y_{r,k}} &= \frac{\partial}{\partial y_{r,k}} \cdot \left(\sqrt{\left(x_{l,k}^i - x_{ls,k}\right)^2 + \left(y_{l,k}^i - y_{ls,k}\right)^2} \right) \\ &= \frac{2\left(x_{l,k} - y_{ls}\right)(-1)}{2\sqrt{\left(x_{l,k} - x_{ls}\right)^2 + \left(y_{l,k} - y_{ls}\right)^2}} \end{aligned} \tag{C.70}$$

$$
\begin{aligned}
\frac{\partial \delta_k^i}{\partial y_{r,k}} =& \frac{\partial}{\partial \theta_{r,k}} \quad . \quad \left(\sqrt{(x_{l,k}-x_{ls})^2+(y_{l,k}-y_{ls})^2}\right) \\
=& \frac{1}{2\sqrt{(x_{l,k}-x_{ls})^2-(y_{l,k}-y_{ls})^2}} \times \Big(2\,(x_{l,k}-x_{ls})\,d_{ls}sin(\theta_{r,k}) - \\
& 2\,(y_{l,k}-y_{ls})\,d_{ls}cos(\theta_{r,k})\Big) \\
=& \frac{d_{ls}}{\sqrt{(x_{l,k}-x_{ls})^2-(y_{l,k}-y_{ls})^2}} \times \Big(\,(x_{l,k}-x_{ls})\,sin(\theta_{r,k}) - \\
& (y_{l,k}-y_{ls})\,cos(\theta_{r,k})\Big)
\end{aligned}
\tag{C.71}
$$

where Equation (C.69) – Equation (C.71) represent the partial derivative of the range with respect to the state. Next, the partial derivative of the bearing with respect to the state can also be calculated as

$$
\begin{aligned}
\frac{\partial \beta_k^i}{\partial x_{r,k}} =& \frac{\partial}{\partial x_{r,k}} \quad . \quad \left(atan\left(\frac{y_{l,k}-y_{ls}}{x_{l,k}-x_{ls}}\right)-\theta_r\right) \\
=& \frac{1}{1+\left(\frac{y_{l,k}-y_{ls}}{x_{l,k}-x_{ls}}\right)^2} \quad . \quad \frac{-(y_{l,k}-y_{ls})}{(x_{l,k}-x_{ls})^2}(-1) \\
=& \frac{y_{l,k}-y_{ls}}{(x_{l,k}-x_{ls})^2+(y_{l,k}-y_{ls})^2}
\end{aligned}
\tag{C.72}
$$

$$
\begin{aligned}
\frac{\partial \beta_k^i}{\partial y_{r,k}} =& \frac{\partial}{\partial y_{r,k}} \quad . \quad \left(atan\left(\frac{y_{l,k}-y_{ls}}{x_{l,k}-x_{ls}}\right)-\theta_r\right) \\
=& \frac{1}{1+\left(\frac{y_{l,k}-y_{ls}}{x_{l,k}-x_{ls}}\right)^2} \quad . \quad \frac{-(x_{l,k}-x_{ls})}{(x_{l,k}-x_{ls})^2} \\
=& \frac{-(x_{l,k}-x_{ls})}{(x_{l,k}-x_{ls})^2+(y_{l,k}-y_{ls})^2}
\end{aligned}
\tag{C.73}
$$

$$
\begin{aligned}
\frac{\partial \beta_k^i}{\partial \theta_{r,k}} =& \frac{\partial}{\partial \theta_{r,k}} \quad . \; atan\left(\frac{y_{l,k}-y_{ls}}{x_{l,k}-x_{ls}}\right)-\theta_r \\
=& \frac{1}{1+\left(\frac{y_{l,k}-y_{ls}}{x_{l,k}-x_{ls}}\right)^2} \quad . \quad \frac{-d_{ls}cos(\theta_{r,k})(x_{l,k}-x_{ls})-(-d_{ls}\,.\,(-sin(\theta_{r,k}))(y_{l,k}-y_{ls}))}{(y_{l,k}-y_{ls})^2}-1
\end{aligned}
\tag{C.74}
$$

Now by supposing that $\Delta x = x_{l,k}-x_{ls}$, $\Delta y = y_{l,k}-y_{ls}$, and $q=(x_{l,k}-x_{ls})^2+(y_{l,k}-y_{ls})^2$, all the compact elements of H can be presented with respect to Equation (C.69) – Equation (C.74).

$$\frac{\partial \delta_k^i}{\partial x_{r,k}} = \frac{-\Delta x}{\sqrt{q}} \tag{C.75}$$

$$\frac{\partial \delta_k^i}{\partial y_{r,k}} = \frac{-\Delta y}{\sqrt{q}} \tag{C.76}$$

$$\frac{\partial \delta_k^i}{\partial \theta_{r,k}} = \frac{d_{ls}}{\sqrt{q}} \left(\Delta x sin(\theta_{r,k}) - \Delta y cos(\theta_{r,k})\right) \tag{C.77}$$

By following the same way as finding the compact formulation of the partial derivative of *h(.)* with respect to state relative to the range, for the bearing one can also be presented as

$$\frac{\partial \beta_k^i}{\partial x_{r,k}} = \frac{\Delta y}{q} \tag{C.78}$$

$$\frac{\partial \beta_k^i}{\partial y_{r,k}} = \frac{-\Delta x}{q} \tag{C.79}$$

$$\frac{\partial \beta_k^i}{\partial \theta_{r,k}} = \frac{-d_{ls}}{q} \left(\Delta x cos(\theta_{r,k}) + \Delta y sin(\theta_{r,k})\right) - 1 \tag{C.80}$$

Accordingly, the Jacobian matrix of measurement is

$$H = \begin{bmatrix} \frac{-\Delta x}{\sqrt{q}} & \frac{-\Delta y}{\sqrt{q}} & \frac{d_{ls}}{\sqrt{q}} \left(\Delta x sin(\theta_{r,k}) - \Delta y cos(\theta_{r,k})\right) \\ \frac{\Delta y}{q} & \frac{-\Delta x}{q} & \frac{-d_{ls}}{q} \left(\Delta x cos(\theta_{r,k}) + \Delta y sin(\theta_{r,k})\right) - 1 \end{bmatrix} \tag{C.81}$$

Note that Equation (C.63) – Equation (C.81) are considered when the model of measurement only consists the range δ_L and bearing β_L. Therefore, if the signature is always included as the last element for each landmark measurement, it is clear to reconstruct the model of measurement z_k as

$$z_k^i = \begin{bmatrix} \delta_k^i + r\delta \\ \beta_k^i + r\beta \\ s_k^i \end{bmatrix} \tag{C.82}$$

For i represent the i-th landmark and $m_{i,s}$ is the correspondence between the expected measurement and the landmark on the map. Consequently, when the dimension of z_i is increased, the Jacobian matrix H is sufficiently increased as well. The formulation of H can be calculated by taking the partial derivative of h(.) with respect to the full state vector of x_k including the robot pose $x_{R,k}$ and landmark x_L^i.

$$H=\frac{\partial z_k^i}{\partial x_k}=\begin{bmatrix}\frac{\partial \delta_k^i}{\partial x_{r,k}} & \frac{\partial \delta_k^i}{\partial y_{r,k}} & \frac{\partial \delta_k^i}{\partial \theta_{r,k}} & \frac{\partial \delta_k^i}{\partial x_{l,j}} & \frac{\partial \delta_k^i}{\partial y_{l,j}} & \frac{\partial \delta_k^i}{s_{l,j}} \\ \frac{\partial \beta_k^i}{\partial x_{r,k}} & \frac{\partial \beta_k^i}{\partial y_{r,k}} & \frac{\partial \beta_k^i}{\partial \theta_{r,k}} & \frac{\partial \beta_k^i}{\partial x_{l,j}} & \frac{\partial \beta_k^i}{\partial y_{l,j}} & \frac{\partial \beta_k^i}{s_{l,j}} \\ \frac{\partial s_k^i}{\partial x_{r,k}} & \frac{\partial s_k^i}{\partial y_{r,k}} & \frac{\partial s_k^i}{\partial \theta_{r,k}} & \frac{\partial s_k^i}{\partial x_{l,j}} & \frac{\partial s_k^i}{\partial y_{l,j}} & \frac{\partial s_k^i}{s_{l,j}}\end{bmatrix} \tag{C.83}$$

Once the model of Jacobian matrix is known as Equation (5.1.68), according to Equation (C.69) – Equation (C.81) the following element are additionally calculated.

$$\begin{aligned}\frac{\partial \delta_k^i}{\partial x_{l,j}} &= \frac{\partial\sqrt{\left(x_{l,j}-x_{ls}\right)^2+\left(y_{l,j}-y_{ls}\right)^2}}{\partial x_{l,j}} \\ &= \frac{2\left(x_{l,j}-x_{ls}\right)}{2\sqrt{\left(x_{l,j}-x_{ls}\right)^2-\left(y_{l,j}-y_{ls}\right)^2}}\end{aligned} \tag{C.84}$$

$$\begin{aligned}\frac{\partial \delta_k^i}{\partial y_{l,j}} &= \frac{\partial\sqrt{\left(x_{l,j}-x_{ls}\right)^2-\left(y_{l,j}-y_{ls}\right)^2}}{\partial y_{l,j}} \\ &= \frac{2\left(y_{l,j}-y_{ls}\right)}{2\sqrt{\left(x_{l,j}-x_{ls}\right)^2-\left(y_{l,j}-y_{ls}\right)^2}}\end{aligned} \tag{C.85}$$

$$\frac{\partial \delta_k^i}{\partial s_{l,j}}=0 \tag{C.86}$$

Equation (C.84) – Equation (C.86) represent the rest element for the first row in *H*. Similarly, the rest elements for the second row are calculated as

$$\begin{aligned}\frac{\partial \beta_k^i}{\partial x_{l,j}} &= \frac{\partial\ \ atan\left(\frac{y_{l,j}-y_{ls}}{x_{l,j}-x_{ls}}\right)-\theta_r}{\partial x_{l,j}} \\ &= \frac{1}{1+\left(\frac{y_{l,j}-y_{ls}}{x_{l,j}-x_{ls}}\right)^2}\ \cdot\ \frac{-\left(y_{l,j}-y_{ls}\right)}{\left(x_{l,j}-x_{ls}\right)^2} \\ &= \frac{-\left(y_{l,j}-y_{ls}\right)}{\left(x_{l,j}-x_{ls}\right)^2+\left(y_{l,j}-y_{ls}\right)^2}\end{aligned} \tag{C.87}$$

$$\begin{aligned}\frac{\partial \beta_k^i}{\partial y_{l,j}} &= \frac{\partial\ \ atan\left(\frac{y_{l,j}-y_{ls}}{x_{l,j}-x_{ls}}\right)-\theta_r}{\partial y_{l,j}} \\ &= \frac{1}{1+\left(\frac{y_{l,j}-y_{ls}}{x_{l,j}-x_{ls}}\right)^2}\ \cdot\ \frac{x_{l,j}-x_{ls}}{\left(x_{l,j}-x_{ls}\right)^2} \\ &= \frac{x_{l,j}-x_{ls}}{\left(x_{l,j}-x_{ls}\right)^2+\left(y_{l,j}-y_{ls}\right)^2}\end{aligned} \tag{C.88}$$

$$\frac{\partial \beta_k^i}{\partial s_{l,j}} = 0 \tag{C.89}$$

Finally, for the last row, which is the partial derivative of $h(\cdot)$ with respect to s_j , all the elements are equal to zero except the last one which is 1. Therefore

$$\frac{\partial s_k^i}{\partial x_{r,k}} = 0 \tag{C.90}$$

$$\frac{\partial s_k^i}{\partial y_{r,k}} = 0 \tag{C.91}$$

$$\frac{\partial s_k^i}{\partial \theta_{r,k}} = 0 \tag{C.92}$$

$$\frac{\partial s_k^i}{\partial x_{l,j}} = 0 \tag{C.93}$$

$$\frac{\partial s_k^i}{\partial y_{l,j}} = 0 \tag{C.94}$$

$$\frac{\partial s_k^i}{\partial s_{l,j}} = 1 \tag{C.95}$$

By applying $\Delta x = x_{l,k} - x_{ls}$, $\Delta y = y_{l,k} - y_{ls}$, and $q = (x_{l,k} - x_{ls})^2 + (y_{l,k} - y_{ls})^2$, we have

$$\frac{\partial \delta_k^i}{\partial x_{l,j}} = \frac{\Delta x}{\sqrt{q}} \tag{C.96}$$

$$\frac{\partial \delta_k^i}{\partial y_{l,j}} = \frac{\Delta y}{\sqrt{q}} \tag{C.97}$$

$$\frac{\partial \delta_k^i}{\partial s_{l,j}} = 0 \tag{C.98}$$

$$\frac{\partial \beta_k^i}{\partial x_{l,j}} = \frac{-\Delta y}{q} \tag{C.99}$$

$$\frac{\partial \beta_k^i}{\partial y_{l,j}} = \frac{-\Delta x}{q} \tag{C.100}$$

$$\frac{\partial \beta_k^i}{\partial s_{l,j}} = 0 \tag{C.101}$$

Then, the Jacobian matrix H for case when $x_L = [x_{l,j} \;\; y_{l,j} \;\; s_{l,j}]^T$ is included in measurement model becomes

$$H = \begin{bmatrix} \frac{-\Delta x}{\sqrt{q}} & \frac{-\Delta y}{\sqrt{q}} & \frac{d_{ls}}{\sqrt{q}}\left(\Delta x sin(\theta_{r,k}) - \Delta y cos(\theta_{r,k})\right) & \frac{\Delta x}{\sqrt{q}} & \frac{\Delta y}{\sqrt{q}} & 0 \\ \frac{\Delta y}{q} & \frac{-\Delta x}{q} & \frac{-d_{ls}}{q}\left(\Delta x cos(\theta_{r,k}) + \Delta y sin(\theta_{r,k})\right) - 1 & \frac{-\Delta y}{q} & \frac{-\Delta x}{q} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (C.102)$$

As the last formulation, in order to maps the low-dimensional of *H* into a matrix with dimension $(3N+3) \times 3$, the following matrix is defined and further involved.

$$B_{x,m} = \begin{bmatrix} 1 & 0 & 0 & 0\ldots0 & 0 & 0 & 0 & 0\ldots0 \\ 0 & 1 & 0 & 0\ldots0 & 0 & 0 & 0 & 0\ldots0 \\ 0 & 0 & 1 & 0\ldots0 & 0 & 0 & 0 & 0\ldots0 \\ 0 & 0 & 0 & 0\ldots0 & 1 & 0 & 0 & 0\ldots0 \\ 0 & 0 & 0 & 0\ldots0 & 0 & 1 & 0 & 0\ldots0 \\ 0 & 0 & 0 & 0\ldots0 & 0 & 0 & 1 & 0\ldots0 \\ & & & \underbrace{}_{2j-2} & & & & \underbrace{}_{2N-2j} \end{bmatrix} \quad (C.103)$$

Note, that from Equation (C.83) – Equation (C.103), the construction of $x_{L,j}$

$$x_{L,j} = \begin{bmatrix} x_{l,j} \\ y_{l,j} \\ s_{l,j} \end{bmatrix} \quad (C.104)$$

For j represents the landmark j on the map correspondences with c_k^i as the expected measurement given the full state vector x_k. For this reason, in order to initialize the expected landmark based on the direct point-based observation into the map, the inverse point-based observation is involved, which is

$$x_{L,j} = \begin{bmatrix} x_{l,j} \\ y_{l,j} \\ s_{l,j} \end{bmatrix} = \begin{bmatrix} x_{r,k} \\ y_{r,k} \\ s_k^i \end{bmatrix} + \delta_k^i \begin{bmatrix} cos(\theta_{r,k} + \beta_k^i) \\ sin(\theta_{r,k} + \beta_k^i) \\ 0 \end{bmatrix} \quad (C.105)$$

Bibliography

[1] A. Ahmed, N. Abdelkrim, and H. Mustapha, “Smooth Variable Structure Filter VSLAM,” *IFAC-PapersOnLine*, vol. 49, no. 15, pp. 205–211, 2016.

[2] D. Fethi, A. Nemra, K. Louadj, and M. Hamerlain, “Simultaneous localization, mapping, and path planning for unmanned vehicle using optimal control,” *Adv. Mech. Eng.*, vol. 10, no. 1, pp. 1–25, 2018.

[3] E. A. Elsheikh, M. A. El-Bardini, and M. A. Fkirin, “Dynamic path planning and decentralized FLC path following implementation for WMR based on visual servoing,” *2016 3rd MEC Int. Conf. Big Data Smart City, ICBDSC 2016*, pp. 148–154, 2016.

[4] Y. Tian, H. Suwoyo, W. Wang, and L. Li, “An ASVSF-SLAM Algorithm with Time-Varying Noise Statistics Based on MAP Creation and Weighted Exponent,” *Math. Probl. Eng.*, vol. 2019, pp. 28–34, 2019.

[5] F. Demim, A. Nemra, and K. Louadj, “Robust SVSF-SLAM for Unmanned Vehicle in Unknown Environment,” *IFAC-PapersOnLine*, 2016.

[6] C. Caceres, J. M. Rosario, and D. Amaya, “Approach of Kinematic Control for a Nonholonomic Wheeled Robot using Artificial Neural Networks and Genetic Algorithms,” *2017 Int. Work Conf. Bio-Inspired Intell. Intell. Syst. Biodivers. Conserv. IWOBI 2017 - Proc.*, 2017.

[7] D. Rodriguez-Losada, P. San Segundo, F. Matia, R. Galan, A. Jiménez, and L. Pedraza, “Dual of the factored solution to the simultaneous localization and mapping problem,” *IFAC Proc. Vol.*, vol. 6, no. PART 1, pp. 542–547, 2007.

[8] Y. Tian, H. Suwoyo, W. Wang, D. Mbemba, and L. Li, “An AEKF-SLAM Algorithm with Recursive Noise Statistic Based on MLE and EM,” *J. Intell. Robot. Syst. Theory Appl.*, 2019.

[9] Y. Liu and C. Wang, “A FastSLAM Based on the Smooth Variable Structure Filter for UAVs,” *2018 15th Int. Conf. Ubiquitous Robot. UR 2018*, pp. 591–596, 2018.

[10] T. D. Barfoot, “State estimation for robotics,” *State Estim. Robot.*, pp. 1–368, 2017.

[11] J. Berisha, X. Bajrami, A. Shala, and R. Likaj, “Application of Fuzzy Logic Controller for obstacle detection and avoidance on real autonomous mobile robot,” *2016 5th Mediterr. Conf. Embed. Comput. MECO 2016 - Incl. ECyPS 2016, BIOENG.MED 2016, MECO Student Chall. 2016*, no. v, pp. 200–205, 2016.

[12] R. Kuemmerle, “State Estimation and Optimization for Mobile Robot Navigation,” pp. 1–191, 2013.

[13] F. Yu, Q. Sun, C. Lv, Y. Ben, and Y. Fu, “A SLAM algorithm based on adaptive cubature Kalman filter,” *Math. Probl. Eng.*, vol. 2014, 2014.

[14] H. Wang, G. Fu, J. Li, Z. Yan, and X. Bian, “An adaptive UKF based SLAM method for unmanned underwater vehicle,” *Math. Probl. Eng.*, vol. 2013, 2013.

[15] A. Karambakhsh, M. Yousefi Azar Khanian, M. R. Meybodi, and A. Fakharian, “Robot navigation algorithm to wall following using fuzzy Kalman filter,” *IEEE Int. Conf. Control Autom. ICCA*, pp. 440–443, 2011.

[16] H. Choset *et al.*, “ PRINCIPLES OF ROBOT MOTION, Theory, Algorithms and Implementations, by Howie Choset et al., MIT Press, 2005. xix + 603 pp., index, ISBN 0-262-

03327-5, 433 references (Hb. £38.95) ," *Robotica*, vol. 24, no. 2, pp. 271–271, 2006.

[17] C. G. Prince, *Book Review: Computational Principles of Mobile Robotics*, vol. 14, no. 3. 2004.

[18] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[19] F. Demim *et al.*, "A new adaptive smooth variable structure filter SLAM algorithm for unmanned vehicle," *2017 6th Int. Conf. Syst. Control. ICSC 2017*, pp. 6–13, 2017.

[20] S. Akhlaghi, N. Zhou, and Z. Huang, "Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2018-Janua, pp. 1–5, 2018.

[21] Z. Zeng, S. Zhang, Y. Xing, and X. Cao, "Robust adaptive filter for small satellite attitude estimation based on magnetometer and gyro," *Abstr. Appl. Anal.*, vol. 2014, 2014.

[22] F. Outamazirt, L. Fu, Y. Lin, and N. Abdelkrim, "A new SINS/GPS sensor fusion scheme for UAV localization problem using nonlinear SVSF with covariance derivation and an adaptive boundary layer," *Chinese J. Aeronaut.*, 2016.

[23] R. Caballero Aguila, A. Hermoso Carazo, J. Linares Pérez, and J. Jiménez López, "Filtering and fixed-point smoothing from an innovation approach in systems with uncertainty," *Extr. Math.*, vol. 18, no. 1, pp. 119–128, 2003.

[24] H. J. S. Feder, J. J. Leonard, and C. M. Smith, "Adaptive mobile robot navigation and mapping," *Int. J. Rob. Res.*, vol. 18, no. 7, pp. 650–668, 1999.

[25] I. Karabegović and V. Doleček, *Mobile Robotics*. 2017.

[26] G. Grisetti, "Towards a PhD Thesis on Simultaneous Localization and Mapping," pp. 1–38, 2004.

[27] F. Lachekhab and M. Tadjine, "Goal seeking of mobile robot using fuzzy actor critic learning algorithm," *Proc. 2015 7th Int. Conf. Model. Identif. Control. ICMIC 2015*, no. Icmic, pp. 1–6, 2016.

[28] L. Teslić, I. Škrjanc, and G. Klančar, "EKF-based localization of a wheeled mobile robot in structured environments," *J. Intell. Robot. Syst. Theory Appl.*, vol. 62, no. 2, pp. 187–203, 2011.

[29] Á. Odry, R. Fullér, I. J. Rudas, and P. Odry, "Kalman filter for mobile-robot attitude estimation: Novel optimized and adaptive solutions," *Mech. Syst. Signal Process.*, 2018.

[30] U. Farooq, A. Khalid, M. Amar, A. Habiba, S. Shafique, and R. Noor, "Design and low cost implementation of a fuzzy logic controller for wall following behavior of a mobile robot," *ICSPS 2010 - Proc. 2010 2nd Int. Conf. Signal Process. Syst.*, vol. 2, pp. V2-740-V2-746, 2010.

[31] L. H. Prayudhi, A. Widyotriatmo, and K. S. Hong, "Wall following control algorithm for a car-like wheeled-mobile robot with differential-wheels drive," *ICCAS 2015 - 2015 15th Int. Conf. Control. Autom. Syst. Proc.*, no. Iccas, pp. 779–783, 2015.

[32] G. Kosuru, "Design and Implementation of an EKF based SLAM algorithm on a mobile robot," no. October, p. 111, 2011.

[33] R. G. Reid, "Large-scale simultaneous localization and mapping for teams of mobile robots," no. July, 2016.

[34] T. Bailey, "Mobile Robot Localisation and Mapping in Extensive Outdoor Environments," *Philosophy*, vol. 31, no. August, p. 212, 2002.

[35] S. Holmes, G. Klein, and D. W. Murray, "A Square Root Unscented Kalman Filter for visual monoSLAM," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 3710–3716, 2008.

[36] T. Bailey and H. Durrant-whyte, "Simultaneous Localisation and Mapping (SLAM): Part II State of the Art," pp. 1–10.

[37] W. Burgard, C. Stachniss, K. Arras, and M. Bennewitz, “Introduction to Mobile Robotics SLAM : Simultaneous Localization and Mapping What is SLAM ?,” no. June, 2010.

[38] E. R. Dougherty, “Optimal Filtering,” *Random Process. Image Signal Process.*, pp. 307–482, 2009.

[39] R. Ahmed, M. El Sayed, S. A. Gadsden, J. Tjong, and S. Habibi, “Artificial neural network training utilizing the smooth variable structure filter estimation strategy,” *Neural Comput. Appl.*, vol. 27, no. 3, pp. 537–548, 2016.

[40] J. J. LaViola, “A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion,” *Proc. Am. Control Conf.*, vol. 3, pp. 2435–2440, 2003.

[41] U. K. Filters, F. Gustafsson, and G. Hendeby, “Some Relations Between Extended and,” *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 545–555, 2012.

[42] S. A. Gadsden, S. R. Habibi, D. Dunne, and T. Kirubarajan, “Combined particle and smooth variable structure filtering for nonlinear estimation problems,” *Fusion 2011 - 14th Int. Conf. Inf. Fusion*, pp. 1552–1559, 2011.

[43] S. A. Gadsden and H. H. Afshari, “A review of smooth variable structure filters: Recent advances in theory and applications,” *ASME Int. Mech. Eng. Congr. Expo. Proc.*, vol. 4A-2015, 2015.

[44] H. H. Afshari, “The 2nd-Order Smooth Variable Structure Filter (2nd-SVSF) for State Estimation: Theory and Applications,” McMaster University, 2014.

[45] M. Al-Shabi and S. Habibi, “Iterative smooth variable structure filter for parameter estimation,” *ISRN Signal Process.*, vol. 2011, no. 1, 2011.

[46] S. R. H. Mohammed Farag , Mina Attari , S. Andrew Gadsden, “Lithium-Ion Battery State of Charge Estimation Using One State Hysteresis Model with Nonlinear Estimation Strategies,” *Int. J. Chem. Mol. Nucl. Mater. Metall. Eng.*, vol. 11, no. 3, pp. 214–218, 2017.

[47] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-gaussian Bayesian state estimation,” *IEE Proceedings, Part F Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, 1993.

[48] J. N. Yang, S. Lin, H. Huang, and L. Zhou, “An adaptive extended Kalman filter for structural damage identification,” *Struct. Control Heal. Monit.*, vol. 13, no. 4, pp. 849–867, 2006.

[49] J. Sol, “Simulataneous localization and mapping with the extended Kalman filter ‘ A very quick guide ... with Matlab code !,’” pp. 1–35, 2012.

[50] Y. Pi, Q. Yuan, and B. Zhang, “The application of adaptive extended Kalman filter in mobile robot localization,” *Proc. 28th Chinese Control Decis. Conf. CCDC 2016*, pp. 5337–5342, 2016.

[51] B. Yi, L. Kang, S. Tao, X. Zhao, and Z. Jing, “Adaptive two-stage extended kalman filter theory in application of sensorless control for permanent magnet synchronous motor,” *Math. Probl. Eng.*, vol. 2013, 2013.

[52] H. H. Afshari, S. A. Gadsden, and S. Habibi, “A nonlinear second-order filtering strategy for state estimation of uncertain systems,” *Signal Processing*, 2019.

[53] S. A. Gadsden, M. El Sayed, and S. R. Habibi, “Derivation of an optimal boundary layer width for the smooth variable structure filter,” *Proc. Am. Control Conf.*, pp. 4922–4927, 2011.

[54] S. A. Gadsden, M. Al-Shabi, I. Arasaratnam, and S. R. Habibi, “Combined cubature Kalman and smooth variable structure filtering: A robust nonlinear estimation strategy,” *Signal Processing*, vol. 96, no. PART B, pp. 290–299, 2014.

[55] S. A. Gadsden and A. S. Lee, “Advances of the smooth variable structure filter: square-root and

two-pass formulations," *J. Appl. Remote Sens.*, 2017.
[56] Z. L. W. X.-X. SUN and M. D. J.-C. Y. A. N. Chao, "Adaptive UKF Filtering Algorithm Based on Maximum a Posterior Estimation and Exponential Weighting," *Acta Autom. Sin.*, vol. 7, 2010.
[57] Y. Meng, S. Gao, Y. Zhong, G. Hu, and A. Subic, "Covariance matching based adaptive unscented Kalman filter for direct filtering in INS/GNSS integration," *Acta Astronaut.*, 2016.
[58] Z. Li, W. Yang, and D. Ding, "Time-Varying Noise Statistic Estimator Based Adaptive Simplex Cubature Kalman Filter," *Math. Probl. Eng.*, vol. 2017, 2017.
[59] A. H. Mohamed and K. P. Schwarz, "Adaptive Kalman filtering for INS/GPS," *J. Geod.*, vol. 73, no. 4, pp. 193–203, 1999.
[60] B. Gao, S. Gao, G. Hu, Y. Zhong, and C. Gu, "Maximum likelihood principle and moving horizon estimation based adaptive unscented Kalman filter," *Aerosp. Sci. Technol.*, 2018.
[61] B. Gao, S. Gao, L. Gao, and G. Hu, "An adaptive UKF for nonlinear state estimation via maximum likelihood principle," *ICEIEC 2016 - Proc. 2016 IEEE 6th Int. Conf. Electron. Inf. Emerg. Commun.*, no. 4, pp. 117–120, 2016.
[62] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," *IEEE 2000 Adapt. Syst. Signal Process. Commun. Control Symp. AS-SPCC 2000*, pp. 153–158, 2000.
[63] F. Gustafsson, *Adaptive Filtering and Change Detection*, vol. 6. 2001.
[64] X. Wang, B. Song, Y. Liang, and Q. Pan, "EM-based adaptive divided difference filter for nonlinear system with multiplicative parameter," *Int. J. Robust Nonlinear Control*, vol. 27, no. 13, pp. 2167–2197, 2017.
[65] J. He, Y. Chen, Z. Zhang, W. Yin, and D. Chen, "A hybrid adaptive unscented Kalman filter algorithm," *Int. J. Eng. Model.*, vol. 31, no. 3, pp. 51–65, 2018.
[66] W. Chen, H. Zhou, F. Shen, and Z. Guo, "Current statistic model and adaptive tracking algorithm based on Kalman and Smooth Variable Structure Filters," *9th Int. Conf. Microw. Millim. Wave Technol. ICMMT 2016 - Proc.*, vol. 2, pp. 1064–1066, 2016.
[67] F. Gustafsson and F. Gustafsson, *Adaptive filtering and change detection*, vol. 1. Citeseer, 2000.
[68] M. R. H. Al-Dahhan and M. M. Ali, "Path tracking control of a mobile robot using fuzzy logic," *13th Int. Multi-Conference Syst. Signals Devices, SSD 2016*, pp. 82–88, 2016.
[69] W. Gao, J. Li, G. Zhou, and Q. Li, "Adaptive Kalman filtering with recursive noise estimator for integrated SINS/DVL systems," *J. Navig.*, vol. 68, no. 1, pp. 142–161, 2015.
[70] E. Levitan and G. T. Herman, "A Maximum A Posteriori Probability Expectation Maximization Algorithm for Image Reconstruction in Emission Tomography," *IEEE Trans. Med. Imaging*, vol. 6, no. 3, pp. 185–192, 1987.
[71] A. Logothetis and V. Krishnamurthy, "Expectation maximization algorithms for MAP estimation of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 47, no. 8, pp. 2139–2156, 1999.
[72] Y. Huang, Y. Zhang, B. Xu, Z. Wu, and J. A. Chambers, "A New Adaptive Extended Kalman Filter for Cooperative Localization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 1, pp. 353–368, 2018.
[73] R. Woo, E. J. Yang, and D. W. Seo, "A fuzzy-innovation-based adaptive Kalman filter for enhanced vehicle positioning in dense urban environments," *Sensors (Switzerland)*, vol. 19, no. 5, 2019.
[74] Z. Donghua, X. Yugeng, and Z. Zhongjun, "A SUBOPTIMAL MULTIPLE FADING

EXTENDED KALMAN FILTER [J]," *Acta Autom. Sin.*, vol. 6, 1991.

[75] S. Gillijns and B. De Moor, "Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough," *Automatica*, vol. 43, no. 5, pp. 934–937, 2007.

[76] S. Chen, Z. Shi, and J. Ding, "Application of the 2nd-order Smooth Variable Structure Filter algorithm for SINS initial alignment," *2017 Forum Coop. Position. Serv. CPGPS 2017*, no. 1, pp. 43–49, 2017.

[77] K. H. Kim, J. G. Lee, and C. G. Park, "Adaptive two-stage extended kalman filter for a fault-tolerant INS-GPS loosely coupled system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 1, pp. 125–137, 2009.

[78] K. A. Myers and B. D. Tapley, "Adaptive Sequential Estimation with Unknown Noise Statistics," *IEEE Trans. Automat. Contr.*, vol. 21, no. 4, pp. 520–523, 1976.

[79] X. Gao, D. You, and S. Katayama, "Seam tracking monitoring based on adaptive Kalman filter embedded elman neural network during high-power fiber laser welding," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4315–4325, 2012.

[80] S. Yong, H. Chongzhao, and L. Yongqi, "Adaptive UKF for target tracking with unknown process noise statistics," *2009 12th Int. Conf. Inf. Fusion, FUSION 2009*, no. 1, pp. 1815–1820, 2009.

[81] Z. Gao, D. Mu, Y. Zhong, C. Gu, and C. Ren, "Adaptively random weighted cubature kalman filter for nonlinear systems," *Math. Probl. Eng.*, vol. 2019, 2019.

[82] Y. Pi, Q. Yuan, and B. Zhang, "The application of adaptive extended Kalman filter in mobile robot localization," *Proc. 28th Chinese Control Decis. Conf. CCDC 2016*, pp. 5337–5342, 2016.

[83] M. S. G. A. P. Andrews, *Kalman Filtering Theory and Practice Using MATLAB*. .

[84] "Kalman Filtering: Theory and Practice Using Matlab." .

[85] S. A. Gadsden, "Smooth Variable Structure Filter Theory and Application.pdf," McMaster University, 2011.

[86] H. H. Afshari, S. A. Gadsden, and S. Habibi, "Gaussian filters for parameter and state estimation: A general review of theory and recent trends," *Signal Processing*, 2017.

[87] K. Al-Mutib, M. Faisal, M. Alsulaiman, F. Abdessemed, H. Ramdane, and M. Bencherif, "Obstacle avoidance using wall-following strategy for indoor mobile robots," *2016 2nd IEEE Int. Symp. Robot. Manuf. Autom. ROMA 2016*, pp. 1–6, 2017.

[88] M. Reinštein, "From Bayes to Extended Kalman Filter Outline of the lecture: Overview: From MAP to RBE Overview: From LSQ to NLSQ Linear Kalman Filter (LKF) Example: Linear navigation problem," 2006.

[89] B. S. Habibi, "Structure Filter," *Proc. IEEE*, vol. 95, no. 5, 2007.

[90] S. A. Gadsden, S. R. Habibi, D. Dunne, and T. Kirubarajan, "Combined particle and smooth variable structure filtering for nonlinear estimation problems," *Fusion 2011 - 14th Int. Conf. Inf. Fusion*, 2011.

[91] R. E. Kalman and R.Buey, "A new approach to linear filtering and prediction theory.," *Trans. ASME, J. Basic Eng.*, vol. 83, no. Series D, pp. 95–108, 1961.

[92] Y. Bar-Shalon, X.-R. Li, and T. Kirubarajan, "Estimation with Applications to Tracking and Navigation." .

[93] Y. Pei, S. Biswas, D. S. Fussell, and K. Pingali, "An Elementary Introduction to Kalman Filtering," pp. 6–11, 2017.

[94] A. Gelb, *Applied optimal estimation*, vol. 64, no. 4. 2001.

[95] J. Gonçalves and S. Warnick, "Technical notes and correspondence," *IEEE Trans. Automat. Contr.*, vol. 53, no. 7, p. 1671, 2008.

[96] S. A. Gadsden, "Smooth Variable Structure Filtering: Theory and Applications," 2011.

[97] Z. Donghua, X. Yugeng, and Z. Zhongjun, "Suboptimal fading extended kalman filtering for nonlinear systems [J]," *Control Decis.*, vol. 5, no. 5, pp. 1–6, 1990.

[98] A. S. Handayani, T. Dewi, N. L. Husni, S. Nurmaini, and I. Yani, "Target tracking in mobile robot under uncertain environment using fuzzy logic controller," *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, vol. 2017-Decem, no. September, pp. 19–21, 2017.

[99] H. Kretzschmar, "Learning Probabilistic Models for Mobile Robot Navigation," pp. 3131–3135, 2014.

[100] G. Der Wu, Z. W. Zhu, and C. W. Chien, "Sparse-sensing-based wall-following control design for a mobile-robot," *Proc. 2016 IEEE Int. Conf. Control Robot. Eng. ICCRE 2016*, pp. 1–5, 2016.

[101] X. Shao, B. He, J. Guo, and T. Yan, "The application of AUV navigation based on adaptive extended Kalman filter," *Ocean. 2016 - Shanghai*, 2016.

[102] J. He, Q. Zhang, Q. Hu, and G. Sun, "A Hybrid Adaptive Unscented Kalman Filter Algorithm," 2017.

Papers published by the author while studying for a Doctor's degree

1. An AEKF-SLAM Algorithm with Recursive Noise Statistic Based on MLE and EM, *Journal of Intelligent and Robotic Systems*, 2019
2. An ASVSF-SLAM Algorithm with Time-Varying Noise Statistics Based on MAP Creation and Weighted Exponent, *Mathematical Problem in Engineering*, 2019
3. Improving the wall-following robot performance using PID-PSO Controller, *Journal Teknology*, 2019
4. Improving a Wall-Following Robot Performance with a PID-Genetic Algorithm Controller, *Proceeding of the Electrical Engineering Computer Science and Informatics*, 2018
5. Enhancing The Performance of The Wall-Following Robot Based on FLC-GA, *Sinergi,* 2020
6. A MAPAEKF-SLAM Algorithm with Recursive Mean and Covariance of Process and Measurement Noise Statistic, *Sinergi,* 2020
7. Maximum likelihood estimation-assisted ASVSF through state covariance-based 2D SLAM algorithm, TELKOMNIKA Telecommunication, Computing, Electronics and Control, 2020
8. Adaptive Development of SVSF for A Feature-Based SLAM algorithm using Maximum Likelihood Estimation and Expectation and Maximization, IIUM Engineering Journal, 2020

Acknowledgement

Firstly, the author would like to express his sincere gratitude to his supervisor, Dr. Yingzhong Tian for the continuous support, guidance, and advice to the related research as well as for the patience, motivation, and immense knowledge given to the author during his study at Mechatronic Engineering, Shanghai University.

Besides that, the author also would like to thanks to the rest of dissertation committee for their insightful comments and encouragement from variative perspective. The sincere thanks also go to Dr. Long Li and Dr. Guangjie Yuan, who supported and provided the author an opportunity to discuss and sharing about the related research. Special thanks are also given to them for their deeply examination and test to all the results of the related research. The author thanks his fellow lab mates in for stimulating discussions, and sleepless nights during completing this dissertation.

Last but not the least, the author would like to specially thank his family: to his mother, Sumarti, for always spiritually supporting his study from time-to-time, to his brother, Toton Hartanto, for his kindness and patience guides the author, to his father, Parwoto, for showing instance to be patience man, to his wife, Refilia Nur Arafiana, for always mentality supporting the author, to his sister, Elinda Luxita Wati. Not to forget, with the same level of thank, the author also would like to express his thanks to the rest members of his family: to his uncle, grandfather, grandmother, and father-mother in law.

This dissertation is dedicated to author's mother, Sumarti, dedicated to his son, Muhammad Kara Raffasya, and presented to his affiliation, Universitas Mercu Buana.