ADAPTIVE FILTERING FOR VISION-AIDED INERTIAL NAVIGATION

A Dissertation
Presented to
The Academic Faculty

By

Kyuman Lee

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

December 2018

Copyright © Kyuman Lee 2018

ADAPTIVE FILTERING FOR VISION-AIDED INERTIAL NAVIGATION

Approved by:

Professor Eric N. Johnson, Advisor
Department of Aerospace Engineering
The Pennsylvania State University

Professor Eric Feron, Co-Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor E. Glenn Lightsey
School of Aerospace Engineering
Georgia Institute of Technology

Professor Marcus J. Holzinger
Ann and H.J. Smead Aerospace Engineering Sciences
University of Colorado Boulder

Professor Byron Boots
School of Interactive Computing
Georgia Institute of Technology

Date Approved: November 9, 2018

That’s one small step for man, one giant leap for mankind.

Neil Armstrong

To my wife and family,

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor, Dr. Eric N. Johnson since he has provided me encouragement and valuable advice during times I have spent with him. I would also like to extend my sincere thanks to the members of my committee: Dr. Eric Feron (co-advisor), Dr. E. Glenn Lightsey, Dr. Marcus J. Holzinger, and Dr. Byron Boots. I must also thank Dr. Javier Irizarry, for his unwavering financial support. Moreover, many thanks to lab members: Jeong Hur, Dr. Daniel P. Magree, Dr. Dmitry Bershadsky, Stephen Haviland, Dr. Takuma Nakamura, Lee Whitcher, Yong Eun Yoon, Toshinobu Watanabe, Jeffrey Lewis, Joel Dunham, Jintasit Pravitra, Jeremy Epps, and Thanakorn Khamvilai. I'd like to acknowledge the assistance of Korean students in Georgia Tech including Sungjin Kim, Youngjun Choi, Pileun Kim, Sanggyu Min, Jung-hyun Kim, Jaein Lim and so on.

Special thanks to my family: Sangsin Lee (father), Youngbok Park (mother), Soongon Chae (father-in-law), Sukki Lee (mother-in-law), Gukhee Lee (sister), Yongsig Lee (brother-in-law), Jongseok Chae (brother-in-law), and Sooahn Lee(nephew). Lastly, I'm extremely grateful to my lovely wife, Minjo Chae.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS OR SYMBOLS

EKF	Extended Kalman Filter.
UKF	Unscented Kalman Filter.
GNC	Guidance, Navigation, and Control.
UAV	Unmanned Aerial Vehicle.
IMU	Inertial Measurement Unit.
GPS	Global Positioning System.
NASA	National Aeronautics and Space Administration.
SLAM	Simultaneous Localization and Mapping.
V-INS	Vision-aided Inertial Navigation System.
VIO	Visual Inertial Odometry.
ROVIO	Robust Visual Inertial Odometry.
SVO	Semi-Direct Visual Odometry.
MSF	Multi-Sensor Fusion.
MSCKF	Multi-State Constraint Kalman Filter.
OKVIS	Open Keyframe-based Visual-Inertial SLAM.
VINS-MONO	Vision-aided Inertial Navigation System-MONOcular camera.
EuRoC	European Robotics Challenge.
MAV	Micro Aerial Vehicle.
RMS	Root Mean Squared.
RANSAC	RANdom SAmple Consensus.
TEDA	Typicality and Eccentricity Data Analysis.
ORKF	Outlier Robust Kalman Filter.
EORKF	Extended ORKF.
IID	Independent and Identically Distributed.
ST	Student's t-Distribution.
PDF	Probability Density Function.
KL	Kullback-Leibler divergence.
ATE	Absolute Trajectory Error.
RPE	Relative Pose Error.
ROS	Robot Operating System .
FAST	Features from Accelerated Segment Test.
KLT	Kanade–Lucas–Tomasi.

x	state.
u	control input.
y	measurement.
t	continuous time.
k	discrete time.
P	error state covariance.
Q	process noise covariance.
R	measurement noise covariance.
K	Kalman gain.
A	Jacobian of dynamic function with respect to state.
B	Jacobian of dynamic function with respect to with process noise.
C	Jacobian of measurement function with respect to state..
J	Jacobian of feature function with respect to state..
Φ	state transition matrix.
η	process noise.
ζ	measurement noise.
r	residual.
j	index of measurements.
i	inertial frame.
b	body frame.
c	camera frame.
c_1	left camera frame.
c_2	right camera frame.
f	feature.
b	bias.
p	position.
v	velocity.
a	acceleration.
q	quaternion.
ω	angular rate.
$\delta\theta$	error quaternion.
$\mathcal{T}$	transformation matrix.
$\mathit{\Lambda}$	precision noise matrix.

SUMMARY

With the advent of unmanned aerial vehicles (UAVs), a major area of interest in the research field of UAVs has been vision-aided inertial navigation systems (V-INS). Many missions of UAVs—reconnaissance, damage assessment, exploration, and other guidance, navigation, and control (GNC) tasks—often demand V-INS in more operational environments such as indoors, hostilities, and disasters. In V-INS, inertial measurement unit (IMU) dead reckoning generates the dynamic models of UAVs, and vision sensors extract information about the surrounding environment and determine features or points of interest. With these sensors, the most widely used algorithm for estimating vehicle and feature states of V-INS is an extended Kalman filter (EKF). The design of the standard EKF does not inherently allow for time offsets between the timestamps of the IMU and vision data, and the necessary assumption of the EKF is Gaussian and white noise. In fact, sensor-related delays and measurement outliers that arise in various realistic conditions are unknown parameters. A lack of compensation of unknown parameters leads to a serious impact on the accuracy of the navigation systems. To compensate for uncertainties of the parameters, we require modified versions of the estimator or the incorporation of other techniques into the filter.

The main purpose of this thesis is to develop accurate and robust V-INS for UAVs, in particular, those for situations pertaining to such unknown parameters. First, to fuse measurements with unknown time delays, this study incorporates parameter estimation into feature initialization and state estimation. Utilizing estimated delays and cross covariance, online temporal calibration, called "latency-adaptive filtering," corrects residual, Jacobian, and covariance. In addition, feature correspondence in image processing front end rejects vision outliers, and then a chi-squared statistic test in filtering back end detects the remaining outliers of the vision data. For frequent outliers, variational approximation for Bayesian inference derives how to compute the optimal noise precision matrices of the measurement outliers. These overall processes of outlier removal and adaptation refer to as "noise-adaptive filtering." Even though almost all of V-INS remove outliers by their own methods, unfortunately, few researchers have treated outlier adaptation in V-INS in great detail. Results from flight dataset tests validate the improved accuracy of V-INS employing these adaptive filtering frameworks.

CHAPTER 1
INTRODUCTION

This chapter describes a clear statement of motivation and implications of problems that we handle in this document. Next, the literature review presents a systematic overview of what has been done and what questions remain unanswered. The next section in this chapter articulates the purpose and contributions of this study, and the last section provides an outline of the content of this dissertation.

1.1 Motivation

The most widely used algorithms for estimating the states of a dynamic system are a Kalman Filter [1, 2] and its nonlinear versions (e.g., extended Kalman filter (EKF) [3, 4] and unscented Kalman filter (UKF) [5]). After the NASA Ames Research Center first implemented the Kalman filter into the navigation computer to estimate the trajectory for the Apollo program, engineers have developed myriad applications of the Kalman filter in guidance, navigation, and control (GNC) research areas [6]. For example, Gaylor and Lightsey [7] designed the GPS/INS Kalman filter for spacecraft operating in the proximity of the international space station, and Holzinger et al. [8] developed the photometric attitude estimator for agile space objects with shape uncertainty. Furthermore, Le Ny et al. [9] scheduled sensor/target assignments and ran their corresponding Kalman filters to solve an attention-control problem in continuous time. Despite the development of numerous applications of the Kalman filter in various fields, it suffers from inaccurate estimation when required assumptions fail.

The design of the standard Kalman filter does not inherently allow for significant sensor-related delays in computation. Fig. 3.1 shows that the *delay* is the time difference between an instant when a measurement is taken by a sensor and another instant when the measurement is available in the filter. As an example of a delay, some complex sensors such as vision processors for navigation often require extensive computations to obtain higher-level information from raw sensor data. Furthermore, a closed-loop system including control logics may be an overall computational burden to a single processing center. Delays resulting from heavy computation may distort the quality of state estimation since a current measurement corresponds to the past states of a system model. In other words, unless compensating delays in Kalman filtering, large estimation errors may accumulate over time.

The delay value is typically unknown and variable in many real applications. For example, even though a local clock is initially forced to synchronize with the centralized clock, deviations between clocks would occur because of clock drift, skew, or bias. In sensor fusion systems, when the timestamps of each sensor are typically recorded by triggered signals, non-deterministic or non-quantized transmission delays lead to unknown time offsets on sensor streams. Moreover, if low-cost sensors such as rolling shutter cameras or software triggered devices are mounted on a vehicle, variance of the uncertainty of timestamps might be larger. In particular, in vision-aided inertial navigation, since a camera does not have its essential clock, we do not know exact time instants when a camera opens and captures images. Exposure time depends on surrounding illumination conditions. The timestamp of a latest grabbed image by some cameras corresponds to the time at which the entire image was available in the memory of a host computer. As such, the timestamps ignore the communication delays that are not precisely known. In fact, when estimating faster motions or using progressive scan cameras, the unknown time delays may worsen the navigation quality. Without calibrating unknown latency in filtering, we cannot guarantee the reliability of the navigation algorithm in practice. In fact, according to an anecdote from our research flying, we observed the necessity of the time delay compensation to be correct for a UAV in flight flying closed loop on the vision-based solution; for, otherwise, we can end up with oscillations that badly exacerbate time delay errors. For example, when a UAV maneuvers with rapid rotation, the bigger time errors produce the larger position estimation errors. Thus, for reliability, an adaptive filtering technique is required to deal with unknown time delays.

The estimation of the Kalman filter is optimal when process and measurement noise are Gaussian. However, sensor measurements are often corrupted by unmodeled non-Gaussian or heavy-tailed noise. An abnormal value relative to an overall pattern of the nominal Gaussian noise distribution is called an *outlier*. In other words, in statistics, an outlier is an observation that deviates so much from other observations as to arouse suspicion that it is generated by a different mechanism [10]. Such outliers have many anomalous causes. They arise due to unanticipated changes in system behavior (e.g., temporary sensor failure or transient environmental disturbance) or unmodeled factors (e.g., human errors or unknown characteristic of intrinsic noise). As an example of measurement outliers in many navigation systems, either computer vision data contaminated by outliers or sonar data corrupted by phase noise lead to erroneous measurements. Process outliers also occur by chance. Inertial measurement unit (IMU) dead reckoning and wheel odometry as a proxy often generate inaccurate dynamic models in visual inertial odometry (VIO) and simultaneous localization and mapping (SLAM) algorithms, respectively. Without accounting for outliers, the accuracy of the estimator significantly degrades, and control systems that rely on high-quality estimation lose stability.

1.2 Related Work

This section reviews the literature pertaining to sensor-related delays and outliers in state estimation including V-INS.

1.2.1 Vision-Aided Inertial Navigation

In recent years, an increasing demand for the research of unmanned aerial vehicles (UAVs) has prompted substantial interest in vision-aided inertial navigation systems (V-INS) [11, 12, 13]. Delmerico and Scaramuzza [14] provide a benchmark comparison of monocular visual inertial odometry (VIO) algorithms for flying robots. Similar to their comparison, Table 1.1 illustrates state-of-the-art VIO techniques even including stereo VIO. Let us explain some relevant terminologies for clarity. VIO uses only data from an IMU and camera vision, but V-INS can fuse other sensors such as altimeters, compass, and GPS with VIO. The tightly-coupled V-INS jointly optimize over all sensor measurements (i.e., visual and inertial cost terms in VIO) which results in higher accuracy. The opposite refers to as loosely coupled.

Table 1.1: State-of-the-art Visual Inertial Odometry

Name	ROVIO [15]	VINS-MONO [16]	SVO +MSF [17]	Alternating Stereo VINS [18]	S-MSCKF [19]	OKVIS [20]
Year	2015 IROS	2017	2016	2018 CVPR	2018 ICRA	2015
Monocular	×	×	×			
Stereo				×	×	×
Indirect		×		×	×	×
Semi-direct			×			
Direct	×					
Loosely Coupled			×			
Tightly Coupled	×	×		×	×	×
Optimization-based		×				×
Filtering-based	×		×	×	×	
Open-source	×	×	×		×	×

VINS-MONO [16, 21] is optimization-based visual SLAM including loop closure. Some processes in this approach is not efficient. VINS duplicates integration with same IMU data at different timestamps for prediction and optimization purposes. That is, for publishing odometry at IMU rate, it integrates whenever IMU data arrived, whereas IMU data are also accumulated in a buffer for batch processing of integration at the time of image measurement update steps. Mourikis first introduced a multi-state constraint Kalman filter (MSCKF) [22, 23] and Sun et al. [19] recently provided its stereo version. Although the real-time high frequency VIO outputs might be crucial for UAV attitude control, MSCKF does not publish the odometry at the IMU rate but at the image rate. Furthermore, batch processing for IMU data integration in MSCKF may add redundant time delays to the filter when vision measurements are available. VINS-MONO and IMU are adequate to only IMU and vision fusion. If we fuse other sensors such as GPS and altimeters in navigation systems, those approaches may not be operable since measurements from other sensors are available to update between images and assumptions for IMU pre-integration between key frames and backward propagation with loop closure in their approaches does not hold. Hence, the EKF-based V-INS frameworks cover more scopes of sensor fusion.

Faessler et al. [17] combined semi-direct visual odometry [24, 25] with modular multi-sensor fusion [26]. Even though this approach uses IMU data for fusing, since it is loosely coupled, its results are suboptimal. Paul et al. [27, 18] recently proposed alternating stereo V-INS that requires computation comparable to monocular V-INS yet provides scale information from the visual observations. However, this method may be not sufficient for tracking fast motions in low-latency demanding applications. Since the implementation is not open-sourced, this is not used for comparison. Leutenegger et al. [20] introduced a consistent keyframe-based stereo SLAM algorithm that performs nonlinear optimization over both visual and inertial cost terms. To maintain the sparsity of the system, their approach employs the some approximation rendering sub-optimal. Since it requires much computation resource or specific levels of sensors such as industrial grade IMUs, operating OKVIS in real-time is more challenging. Among six algorithms in Table 1.1, only S-MSCKF handles an unknown latency and only SVO+MSF deeply considers outlier rejection. Hence, I will use their estimation results as prior work in comparisons to this study.

When we use sensors in V-INS, their numerous unknown parameters affect the navigation solution. Chapter 1.1 introduced the significance of sensor-related unknown delays and outliers. A few among the above state-of-the art VIO have extended to investigate the unknown time delays of vision data, and Section 1.2.2 will present the details of the extensions. However, in fact, few researchers have treated noise-adaptive filtering for V-

INS in great details. Instead, Section 1.2.3 will explore how to handle outliers in general filtering areas.

1.2.2 State Estimation Using Time-Delayed Measurements

In a number of applications, a vital problem for combining data from various sensors is the fusion of delayed observations, and if the computational delay is crucial, fusing the data in a Kalman filter is challenging. During the last 20 years, the sensor time-delay problem have been solved by a number of methods, most of which modify the Kalman filter so that it handles delay in the sensor fusion algorithm. Alexander [28] derived a method of calculating a correction term and then added it to filter estimates when lagged measurements arrive. However, because the uncertainty of measurements is often an unknown quantity until the data are processed, applying the method in time-varying systems is impossible. To overcome the shortcoming of Alexander's method, Larsen et al. [29] extrapolated a measurement to a current time using the past and present estimates of the Kalman filter and calculated an optimal gain for this extrapolated measurement. However, Larsen's approach is exact for only linear systems, but if the system dynamics and measurement equations are significantly nonlinear, it can be highly inaccurate. For optimally fusing lagged sensor data in a general nonlinear system, Van Der Merwe et al. [30, 31] introduced a new technique called "sample-state augmentation," based on the Schmidt-Kalman filter [32] or the stochastic cloning [33]. Appendix C provides detailed background information about the new technique. Lastly, Gopalakrishnan et al. [34] provided a survey of all previously noted methods.

Unknown Time Delays

All of the above methods assume that the amount of delay is known. As an illustration, those methods only work with a few strictly hardware synchronized sensors. However, the hardware synchronization of most low-cost or customized sensors is not available.

Moreover, situations in which a current, accurate time delay might not be known can arise in real applications. To deal with the unknown time delays, Julier and Uhlmann [35] introduced the covariance union algorithm, and Sinopoli et al. [36] modeled the arrival of intermittent observations as a random variable with a probability. In addition, Choi et al. [37] and Yoon et al. [38] augmented a state vector with as many past states as the maximum number of delayed steps. The size of this augmented state vector is extremely large, and calculations with the large-size vector might require additional extensive computational effort. Recently, for the uncertainty of time delays in state estimation, Lee and Johnson [39] also suggested an approach combined with multiple-model adaptive estimation. However, because of imperfect information on a certain range of the delay value, this method might not be suitable if too many models are candidates with delay values.

Instead, we directly estimate the time delay as an additional state since augmentation is a straightforward means of handling unknown delay. Nilsson et al. [40] investigated this idea using Taylor series expansion for small delays. However, delay values are typically larger than a time step, and the linearization in their approach does not hold for large delays. Li and Mourikis [41] also examined the state augmentation for estimating an unknown time offset between the timestamps of two sensors. However, their approach is not optimal since it performs the measurement update of delayed sensor data without the covariance correction that uses the cross-covariance term computed during the delay period. Furthermore, in the recent optimization-based method proposed by Qin and Shen [42], if cameras move at non-constant speed during the short time period like progressive scan cameras, then their assumption does not hold any longer. Despite the short time period, the camera coordinate frame is still changing and moving. Their another assumption in which the time offset is a constant variable is also not general since the unknown delay may be varying. Even though their analysis is meaningful, the purpose of the online temporal calibration is calibrating the existing time offsets whose some parts are possibly uncertain. However, to validate their proposal, they forced to adjust the timestamps of datasets and defined the adjustment as time offsets, but this newly defined time offsets can be different from the really existing time offsets of the datasets. To overcome all previously noted drawbacks, Chapter 3 proposes a novel approach, “latency-adaptive filtering” based on the combined parameter-state estimator [43, 44].

1.2.3 State Estimation for Measurements with Outliers

Since the performance of the Kalman filter degrades at the presence of measurement outliers, many researchers have investigated other approaches to mitigate the impact of outliers. Mehra [45] created adaptive filtering with the identification of noise covariance matrices and showed the asymptotic convergence of the estimates towards their true values. Maybeck [46] and Stengel [47] found other noise-adaptive filtering such as covariance matching. However, all of these filters performed only offline and required filter tuning. To estimate parameter values in unknown covariances without the need for manual parameter tuning, Ting et al. [48] used a variational expectation-maximization (EM) framework. That is, they introduced a scalar weight for each observed data sample and modeled the weights to be Gamma distributed random variables. However, it assumed that noise characteristics are homogeneous across all measurements even though sensors have distinct properties. Särkkä and Nummenmaa [49] provided the online learning of the parameters of the measurement noise variance, but to simultaneously track the system states and the levels of sensor noise, they additionally defined a heuristic transition model for the noise parameters. Piché et al. [50] developed Gaussian assumed density filtering and smoothing framework for nonlinear systems using the multivariate Student t-distribution, and Roth et al. [51] included an approximation step for heavy tailed process noise, but this filter are not applicable in high dimensions. Next, Solin and Särkkä [52] found that the added flexibility of Student-t processes over Gaussian processes robustifies inference in outlier-contaminated noisy data, but they treated only analytic solutions enabled by the noise entanglement.

Recently, Agamennoni et al. developed the outlier robust Kalman filter (ORKF) [53, 54] to obtain the optimal precision matrices of measurement outliers by variational approximation for Bayesian inference [55]. However, this method requires iterations at every time, even when observed data contain no outliers. Graham et al. also established the ℓ_1-norm filter [56] for both types of sparse outliers. However, the filter might not work for nonlinear systems since they derived the constraint of ℓ_1-norm optimization based on only linear system equations. Similar to the ORKF, the ℓ_1-norm filter needs the constrained optimization at all times, even when no additional noise present as outliers. Hence, these two approaches demand some extensive computational complexity for either iterations or optimization. Since outliers do not always arise (i.e., are rare), we reduce such computation cost if a tests detect the time when outliers occur. All of the above methods were not validated for complicated systems such as unmanned aerial vehicles or vision-aided inertial navigation including the sequential measurement update.

Outlier Rejection Techniques

One of the primary problems in VIO is incorrect data associations. Matched features between two different camera views are corrupted by outliers because of image noise, occlusions, and illumination changes that are not modeled by the feature matching techniques. To provide cleaned measurement data to the filter, outlier removal in image processing front end is essential. One of standard outlier rejection techniques is RANdom SAmple Consensus (RANSAC) [57]. RANSAC is an iterative approach to estimate the parameters of a mathematical model from a set of observed data contaminated by outliers. An underlying assumption is that the data consists of inliers whose distribution is explained by some set of the model parameters and outliers that do not fit the model. The generated parameters are then verified on the remaining subset of the data, and the model with the highest consensus is a selected solution. In particular, 2-point RANSAC [58, 59] is an extended RANSAC-based method for two consecutive views of a camera rigidly mounted on a vehicle platform. Given gyroscopic data from IMU measurements, randomly selected two feature correspondences hypothesize an ego-motion of the vehicle. This motion constraint discards wrong data associations in the feature matching processes.

For detecting remaining outliers that were not rejected in image processing front end, outlier detection tests are required in filtering back end. Most of statistical tests [60] that require access to the entire set of data samples for detecting outliers might not be a viable option in real-time applications. For example, the typicality and eccentricity data analysis (TEDA) [61, 62, 63] used in [64] is an inadequate measure in V-INS since computing the means and variances of each residual of sequential measurements is challenging. The tracking of some feature measurements is possibly lost due to out of sight and new feature measurements are coming for initialization.

For the real-time outlier detection of sequential measurements in V-INS, the Mahalanobis gating test [65] is a useful measure based on the analysis of residual and covariance signals at each feature measurement. The approach builds upon each Mahalanobis distance [66] of residuals and compares its value against a threshold given by the quantile of the chi-squared distribution with degrees of freedom. The confidence level of the threshold is designated prior to examining the data. Most commonly, the 95% confidence level is used. This hypothesis testing, called goodness of fit, is a commonly used outlier detection method in practice. Because of such suitability of the Mahalanobis gating test to real-time detection in V-INS, this thesis combined the test with outlier robust EKF (ORKF) [53, 64] to detect and handle measurement outliers in vision-aided estimation problems. Similar to the derivation of update steps for handling measurement outliers in the ORKF, for computing the optimal precision matrices of unmodeled outliers in V-INS, Chapter 4 will derive feasible update procedures by variational inference. In other words, whenever unexpected outliers appear, the noise-adaptive filtering in Chapter 4 updates and marginalizes measurement outliers to improve robustness of the navigation systems.

1.3 Summary of Contributions

This thesis presents several adaptive and robust estimation solutions for vision-aided inertial navigation systems (V-INS) and evaluates their performance with flight datasets testing. For problem statements, the objectives of this thesis are as follows:

- Development of a practical EKF-based V-INS accounting for vehicle-feature correlations.

- Development of tightly coupled visual inertial odometry (VIO) for autonomous flight of UAVs.

- More precise definition of time delays of vision data measurements in V-INS.

- Development of a reliable and accurate filtering formulation for measurements with unknown time delays.

- Improved utilization of outlier removal techniques in image processing front end.

- Development of a robust and adaptive state estimation framework for V-INS under frequent outliers occurrence.

- Test of the performance of V-INS employing the adaptive filtering algorithms in the benchmark flight datasets for comparison to other state-of-the-art VIO algorithms.

- Validation of improved accuracy of V-INS employing the latency-adaptive filtering in the fast motion flight dataset.

- Validation of improved accuracy of V-INS employing the noise-adaptive filtering in the motion blur flight dataset.

This research is conducted within the following scope.

- We handle only unknown time offsets between the timestamps of the IMU and vision sensor data. However, if V-INS fuse another sensor, its unknown time delays may be investigated by the similar way of this study.

- Similar to global shutter cameras, all features from one image have one delay value. However, if relaxing the assumption, we might solve rolling shutter effects by augmenting more state variables regarding multiple unknown delays.

- The unknown part of time delays are static or varying slowly.

- We solve for the situation in which only independent and identically distributed (IID) measurement outliers occur.

- We assume that two unknown parameters—time-delayed measurements and outliers—are independent although they can occur together. Hence, each uncertainty of the unknown parameters generates each distinct adaptive estimation problem.

Starting from the architecture of the existing navigation system, this dissertation focuses on contributing the development of red boxes in Figure 1.1.

1.4 A Guide to This Document

The remainder of this document contains the following chapters. Chapter 2 introduces background for all of this study. To estimate unknown time delays and the states of V-INS, Chapter 3 presents a novel combination of the parameter-adaptive filtering technique with the modified EKF that compensates delayed measurements. To estimate the states of V-INS in which frequent outliers arise, Chapter 4 examines outlier rejection techniques in image processing front end and formulates a novel implementation of robust noise-adaptive filtering. The last chapter concludes and plans future work.

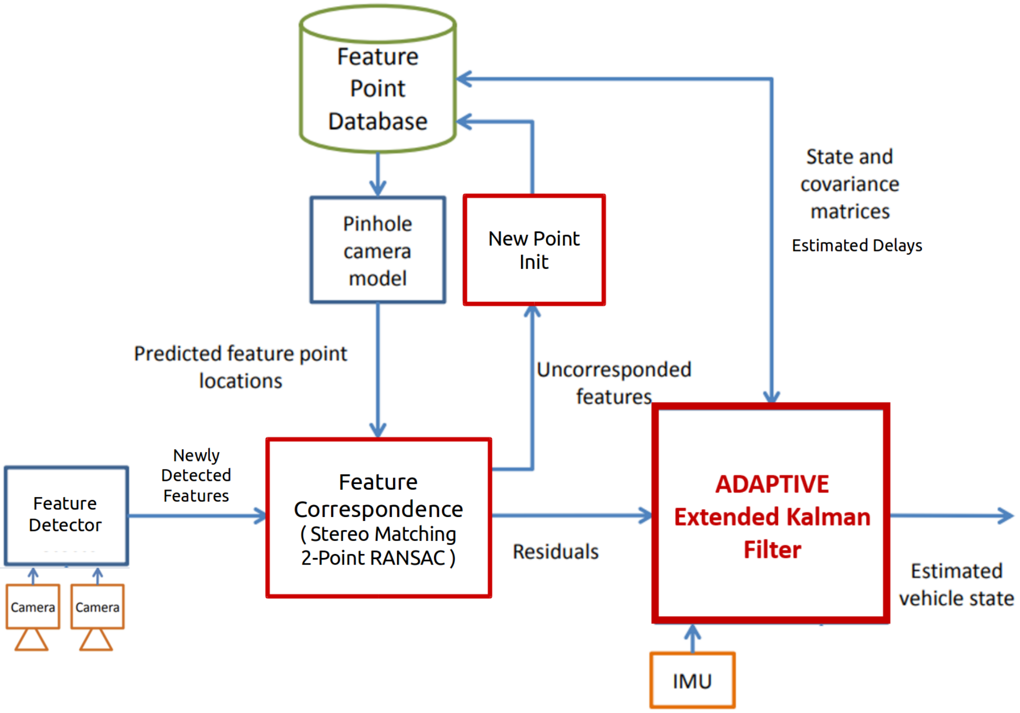


Figure 1.1: A Block Diagram of the Vision-Aided Inertial Navigation System Employing Adaptive Filtering

CHAPTER 2

PRELIMINARIES

This chapter provides background for chapters that follow. The first section presents an overview of the EKF that forms the basis of all proposed adaptive filters in this thesis. The next sections introduce the vehicle model for state propagation and the camera model for measurement update in the filters, respectively.

2.1 The Extended Kalman Filter

The system equations with continuous-time dynamics and a discrete-time sensor are as follows:

$$\dot{x}(t) = f(\,x(t),\, \eta(t)\,) + B_u\, u(t) \tag{2.1}$$

$$y(t_k) = h(\,x(t_k)\,) + \zeta(t_k), \tag{2.2}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^l$ a control input, and $y \in \mathbb{R}^m$ a measurement. $f(\cdot)$ and $h(\cdot)$ are the nonlinear dynamic and measurement functions, respectively, and B_u is the input matrix. Let's assume that these functions are known based on each equation of motion and modeling. To clarify, t denotes continuous time, subscript k represents the k-th time step, and initial condition $x(0) = x_0$ is given. Moreover, let's assume that both propagation and measurements are corrupted by additive zero-mean white Gaussian noise; that is, $\eta(t) \sim \mathcal{N}(0, Q(t))$ and $\zeta(t_k) \sim \mathcal{N}(0, R(t_k))$.

2.1.1 Time Update

To estimate the state variables of the system, we design a hybrid EKF in the following steps. In the propagation step, state estimate $\hat{x} := \mathbb{E}[x]$ and its error-covariance $P :=$

$\mathbb{E}[\,(x-\hat{x})(x-\hat{x})^{\mathrm{T}}\,]$ are integrated from time $(k-1)^{+}$ to time k^{-} with respect to variable τ

$$\dot{\hat{x}}_{k-1} = f(\hat{x}^{+}_{k-1}) + B_u\, u_{k-1} \tag{2.3}$$

$$\hat{x}^{-}_{k} = \hat{x}^{+}_{k-1} + \int_{t_{k-1}}^{t_k} \dot{\hat{x}}(\tau)\, d\tau \;\simeq\; \hat{x}^{+}_{k-1} + \left(\frac{3}{2}\dot{\hat{x}}_{k-1} - \frac{1}{2}\dot{\hat{x}}_{k-2}\right)\Delta t_{k-1}, \tag{2.4}$$

where $\Delta t_{k-1} = t_k - t_{k-1}$, let $\hat{x}_k = \hat{x}(t_k)$ and $u_k = u(t_k)$. Hat "ˆ" denotes an estimate, and superscript $-$ and $+$ a priori and a posteriori estimates, respectively. Here, for one numerical solution of the ordinary differential equation, the Heun's method [67] that refers to the improved Euler's method or a similar two-stage Runge–Kutta method is used. Jacobian A, B and state transition matrix Φ are defined by

$$A_{k-1} = \left.\frac{\partial f(x)}{\partial x}\right|_{\hat{x}^{+}_{k-1}}, \quad B_{k-1} = \left.\frac{\partial f(x)}{\partial \eta}\right|_{\hat{x}^{+}_{k-1}} \tag{2.5}$$

$$\Phi_{k-1} = \exp(A_{k-1}\,\Delta t_{k-1}) \approx I + A_{k-1}\,\Delta t_{k-1}. \tag{2.6}$$

Letting $P_k = P(t_k)$ and $Q_k = Q(t_k)$, the time update of error covariance is

$$P^{-}_{k} = \Phi_{k-1}\, P^{+}_{k-1}\, \Phi^{\mathrm{T}}_{k-1} + B_{k-1}\, Q_{k-1} B^{\mathrm{T}}_{k-1}\, \Delta t_{k-1}. \tag{2.7}$$

2.1.2 Measurement Update

Using actual sensor measurements, the measurement update step of the EKF corrects state estimate and its corresponding error covariance after propagation. Letting $y_k = y(t_k)$ and $R_k = R(t_k)$, at every time k,

$$K_k = P^{-}_{k} C^{\mathrm{T}}_{k}\,(\,C_k P^{-}_{k} C^{\mathrm{T}}_{k} + R_k\,)^{-1} \tag{2.8}$$

$$\hat{x}^{+}_{k} = \hat{x}^{-}_{k} + K_k\,(\,y_k - h(\hat{x}^{-}_{k})\,) \tag{2.9}$$

$$P^{+}_{k} = P^{-}_{k} - K_k C_k P^{-}_{k}, \tag{2.10}$$

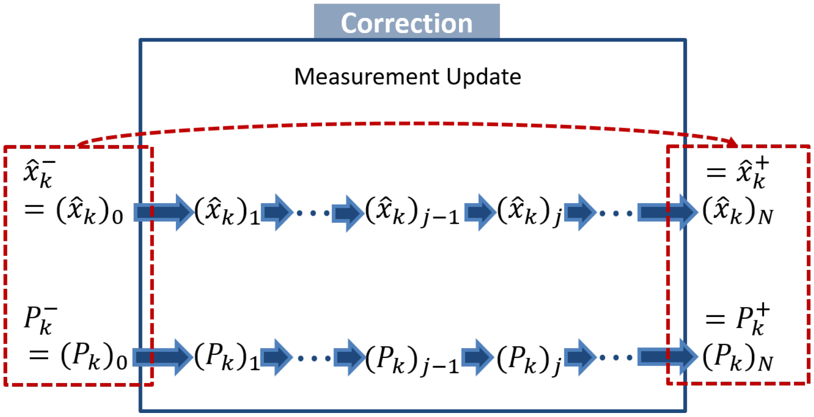


Figure 2.1: A Schematic of the Sequential Measurement Update

where K is called the Kalman gain and Jacobian C is defined as

$$C_k = \left. \frac{\partial h(x)}{\partial x} \right|_{\hat{x}_k^-} . \tag{2.11}$$

Equation (2.10) is the Joseph's form [68] of the covariance measurement update, so this form preserves its symmetry and positive definite. For more details such as optimality and derivation, see references [69, 70].

2.1.3 Sequential Kalman Filter

When myriad measurements are observed at one time, sequential Kalman filtering is useful. In fact, we obtain N measurements, $y_1, y_2, \cdots, y_N$, at time k; that is, we first measure y_1, then y_2, $\cdots$, and finally y_N, shown in Figure 2.1.

We first initialize a posteriori estimate and covariance after zero measurement is processed; that is, they are equal to the a priori estimate and covariance. For $i = 1, \cdots, N$, perform the general measurement update using the i-th measurement. We lastly assign the a posteriori estimate and covariance as

$$(\hat{x}_k)_0 \leftarrow \hat{x}_k^-, \quad (P_k)_0 \leftarrow P_k^- \tag{2.12}$$

$$(\hat{x}_k)_1 = (\hat{x}_k)_0 + K_1 \left(y_1 - h_1 \left((\hat{x}_k)_0\right)\right), \quad (P_k)_1 = (P_k)_0 - K_1 (C_1)_k (P_k)_0 \tag{2.13}$$

$$\vdots$$

$$(\hat{x}_k)_j = (\hat{x}_k)_{j-1} + K_j \left(y_j - h_j \left((\hat{x}_k)_{j-1}\right)\right), \quad (P_k)_j = (P_k)_{j-1} - K_j (C_j)_k (P_k)_{j-1} \tag{2.14}$$

$$\vdots$$

$$\hat{x}_k^+ \leftarrow (\hat{x}_k)_N, \quad P_k^+ \leftarrow (P_k)_N. \tag{2.15}$$

Since Simon [69] proved that the sequential Kalman filtering is equivalent formulation of the standard EKF, the order of updates does not affect overall performance of estimation.

2.2 Vehicle Model

The nonlinear dynamics of a vehicle is driven by raw inertial measurement unit (IMU) sensor data including specific force and angular velocity inputs. Figure 2.2 illustrates the key reference frames used in this document: the inerial frame i, the vehicle body frame b, and the camera frame c. In general, the origins of the IMU sensor frame and the body frames are in different locations, but for simplicity of presentation, we assume that they are co-located.

The vehicle state is given by

$$\hat{x}_V = \begin{bmatrix} {}^i\hat{p}_{b/i}^{\mathrm{T}} & {}^i\hat{v}_{b/i}^{\mathrm{T}} & \delta\hat{\theta}^{\mathrm{T}} & \hat{b}_a^{\mathrm{T}} & \hat{b}_\omega^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{2.16}$$

where $p_{b/i}$, $v_{b/i}$ are the position and velocity of the vehicle with respect to the inertial frame, respectively. $\delta\theta$ is the error quaternion of the attitude of the vehicle, and its more details will be explained in Equations (2.27) – (2.28) or references [71, 72, 73]. b_a, b_ω are the

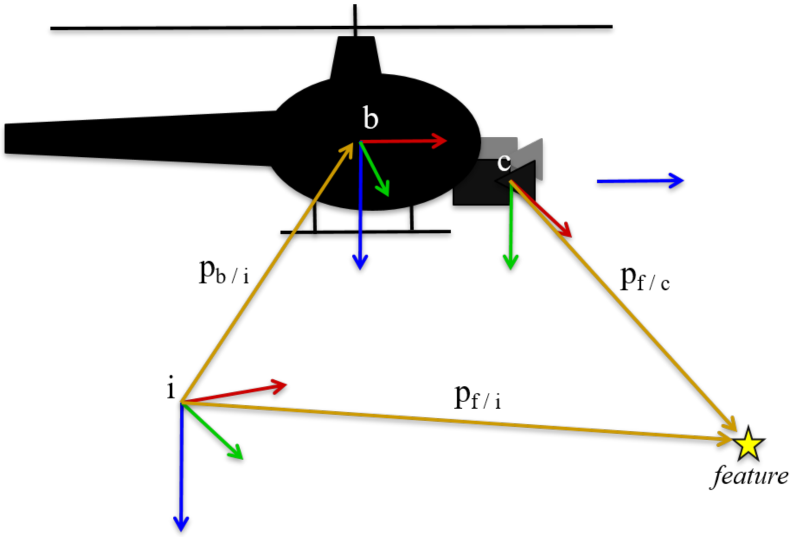


Figure 2.2: A Schematic of the Key Reference Frames

acceleration and gyroscope biases of the IMU, respectively. Left superscript i denotes a vector expressed in the inertial frame. The EKF propagates the vehicle state vector by dead reckoning with data from the IMU. Raw IMU sensor measurements a_{raw} and ω_{raw} are corrupted by noise and bias as follows:

$$a_{\text{raw}} = a_{\text{true}} - \mathcal{T}_{b/i}\,{}^{i}g + b_a + \eta_a, \qquad \dot{b}_a = \eta_{b_a} \tag{2.17}$$

$$\omega_{\text{raw}} = \omega_{\text{true}} + b_\omega + \eta_\omega, \qquad \dot{b}_\omega = \eta_{b_\omega}, \tag{2.18}$$

where a_{true}, ω_{true} are the true acceleration and angular rate, respectively, and g is the gravitational acceleration in the inertial frame. η_a, η_ω are zero-mean, white, Gaussian noise of the accelerometer and gyroscope measurement, and η_{b_a}, η_{b_ω} are the random walk rate of the acceleration and gyroscope biases. The rotation matrix from the inertial frame to the body frame denotes $\mathcal{T}_{b/i} = \mathcal{T}_{i/b}{}^{\mathrm{T}}$.

The vehicle dynamics is given by

$$
\begin{aligned}
{}^{i}\dot{\hat{p}}_{b/i} &= {}^{i}\hat{v}_{b/i} && (2.19)\\
{}^{i}\dot{\hat{v}}_{b/i} &= \hat{\mathcal{T}}_{i/b}\,(\,a_{\text{raw}} - \hat{b}_a\,) + {}^{i}g && (2.20)\\
\dot{\hat{q}}_{i/b'} &= \frac{1}{2}\,\mathcal{Q}(\,\omega_{\text{raw}} - \hat{b}_\omega\,)\,\hat{q}_{i/b'} && (2.21)\\
\delta\dot{\hat{\theta}} &= -\left\lceil\,(\,\omega_{\text{raw}} - \hat{b}_\omega\,)\times\right\rfloor\delta\hat{\theta} && (2.22)\\
\dot{\hat{b}}_a &= 0 && (2.23)\\
\dot{\hat{b}}_\omega &= 0, && (2.24)
\end{aligned}
$$

where a skew symmetric matrix is defined

$$
\lceil\alpha\,\times\rfloor = \left\lceil\,[\alpha_1\;\;\alpha_2\;\;\alpha_3]^{\mathrm{T}}\times\,\right\rfloor = \begin{bmatrix} 0 & -\alpha_3 & \alpha_2 \\ \alpha_3 & 0 & -\alpha_1 \\ -\alpha_2 & \alpha_1 & 0 \end{bmatrix}, \tag{2.25}
$$

so function $\mathcal{Q}(\cdot)$ maps a 3 by 1 vector of the angular velocity into a 4 by 4 matrix as follows:

$$
\mathcal{Q}(\omega) = \begin{bmatrix} 0 & -\omega^{\mathrm{T}} \\ \omega & -\lceil\omega\times\rfloor \end{bmatrix}. \tag{2.26}
$$

The use of the 4 by 1 quaternion representation in state estimation causes the covariance matrix to become singular, so it requires considerable accounting for the quaternion constraints. To avoid these difficulties, engineers developed the error-state Kalman filter in which 3 by 1 infinitesimal error quaternion $\delta\theta$ is used instead of 4 by 1 quaternion q in the state vector. In other words, we use attitude error quaternion $\delta q_{b/b'}$ to express the incremental difference between tracked reference body frame b' and actual body frame b for the vehicle, shown in

Figure 2.3.

$$q_{i/b} = \hat{q}_{i/b'} \otimes \delta q_{b'/b} \tag{2.27}$$

$$\delta q_{b'/b} = \hat{q}_{i/b'}^{-1} \otimes q_{i/b} \simeq \begin{bmatrix} 1 \\ \frac{1}{2}\delta\theta \end{bmatrix} \tag{2.28}$$

Resulting rotation matrices with error quaternion and with respect to the nominal reference

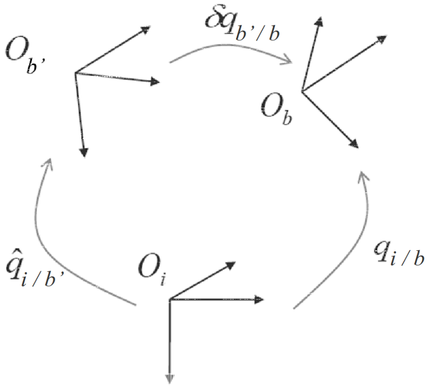


Figure 2.3: Definition of an Attitude Error Quaternion

body frame are

$$\mathcal{T}(q_{i/b}) = \hat{\mathcal{T}}_{b/i} = \hat{\mathcal{T}}_{b/b'}\,\hat{\mathcal{T}}_{b'/i} \tag{2.29}$$

$$\hat{\mathcal{T}}_{i/b'} = \hat{\mathcal{T}}_{b'/i}^{\mathrm{T}} = \mathcal{T}(\hat{q}_{i/b'})^{\mathrm{T}} \tag{2.30}$$

$$\hat{\mathcal{T}}_{b'/b} = \hat{\mathcal{T}}_{b/b'}^{\mathrm{T}} \simeq \Big(I + [\hat{\delta\theta}\times]\Big)^{\mathrm{T}}. \tag{2.31}$$

Jacobian matrix $A = \frac{\partial \dot{x}}{\partial x}|_{\hat{x}}$ and $B = \frac{\partial \dot{x}}{\partial \eta}$, where $\eta = [\, \eta_a^{\mathrm{T}}, \eta_\omega^{\mathrm{T}}, \eta_{b_a}^{\mathrm{T}}, \eta_{b_\omega}^{\mathrm{T}} \,]^{\mathrm{T}}$, are computed

$$A = \begin{bmatrix} 0 & \frac{\partial\, {}^i\dot{\hat{p}}_{b/i}}{\partial\, {}^i\hat{v}_{b/i}} & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial\, {}^i\dot{\hat{v}}_{b/i}}{\partial\, \delta\hat{\theta}} & \frac{\partial\, {}^i\dot{\hat{v}}_{b/i}}{\partial\, \hat{b}_a} & 0 \\ 0 & 0 & \frac{\partial\, \delta\dot{\hat{\theta}}}{\partial\, \delta\hat{\theta}} & 0 & \frac{\partial\, \delta\dot{\hat{\theta}}}{\partial\, \hat{b}_\omega} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{\partial\, {}^i\dot{\hat{v}}_{b/i}}{\partial\, \eta_a} & 0 & 0 & 0 \\ 0 & \frac{\partial\, \delta\dot{\hat{\theta}}}{\partial\, \eta_\omega} & 0 & 0 \\ 0 & 0 & \frac{\partial\, \dot{b}_a}{\partial\, \eta_{b_a}} & 0 \\ 0 & 0 & 0 & \frac{\partial\, \dot{b}_\omega}{\partial\, \eta_{b_\omega}} \end{bmatrix} \tag{2.32}$$

$$\frac{\partial\, {}^i\dot{\hat{p}}_{b/i}}{\partial\, {}^i\hat{v}_{b/i}} = I_{3\times3}, \quad \frac{\partial\, {}^i\dot{\hat{v}}_{b/i}}{\partial\, \delta\hat{\theta}} = -\hat{\mathcal{T}}_{i/b'} \left\lceil \,(\, a_{\text{raw}} - \hat{b}_a \,) \times \right\rfloor, \quad \frac{\partial\, {}^i\dot{\hat{v}}_{b/i}}{\partial\, \hat{b}_a} = -\hat{\mathcal{T}}_{i/b},$$
$$\frac{\partial\, \delta\dot{\hat{\theta}}}{\partial\, \delta\hat{\theta}} = -\left\lceil \,(\, \omega_{\text{raw}} - \hat{b}_\omega \,) \times \right\rfloor, \quad \frac{\partial\, \delta\dot{\hat{\theta}}}{\partial\, \hat{b}_\omega} = -I_{3\times3},$$
$$\frac{\partial\, {}^i\dot{\hat{v}}_{b/i}}{\partial\, \eta_a} = -\hat{\mathcal{T}}_{i/b}, \quad \frac{\partial\, \delta\dot{\hat{\theta}}}{\partial\, \eta_\omega} = -I_{3\times3}, \quad \frac{\partial\, \dot{b}_a}{\partial\, \eta_{b_a}} = I_{3\times3}, \quad \frac{\partial\, \dot{b}_\omega}{\partial\, \eta_{b_\omega}} = I_{3\times3},$$

where for more detailed derivations, see a reference [74].

2.3 Camera Model

An intrinsically calibrated pinhole camera model depicted in Figure 2.4 is given by

$$\begin{bmatrix} u_j \\ v_j \end{bmatrix} = y_j = h_j\left(x\right) + \zeta_j = \begin{bmatrix} f_u \frac{{}^cX_j}{{}^cZ_j} + \zeta_{u_j} \\ f_v \frac{{}^cY_j}{{}^cZ_j} + \zeta_{v_j} \end{bmatrix} \tag{2.33}$$

$$\begin{aligned} \begin{bmatrix} {}^cX_j, & {}^cY_j, & {}^cZ_j \end{bmatrix}^{\mathrm{T}} &= {}^cp_{f_j/c} = \mathcal{T}_{c/i}\, \left({}^ip_{f_j/i} - {}^ip_{c/i} \right) \\ &= \mathcal{T}_{c/b}\, \mathcal{T}\left(q_{i/b}\right) \left({}^ip_{f_j/i} - {}^ip_{b/i} \right) - \mathcal{T}_{c/b}\, {}^bp_{c/b} \end{aligned} \tag{2.34}$$

where measurement y_j is the j-th feature $2D$ location on the image plane. f_u, f_v are the horizontal and vertical focal lengths, respectively, and ζ_u, ζ_v are additive, zero-mean, white, Gaussian noise of the measurement. Vectors $p_{f_j/c}$, $p_{f_j/i}$ defined in Figure 2.2 are the j-th feature $3D$ position with respect to the camera frame and the inertial frame, respectively. Extrinsic parameter $\mathcal{T}_{c/b}$ and ${}^bp_{c/b}$ are known and constant, and rotation matrix $\hat{\mathcal{T}}_{c/i} =$

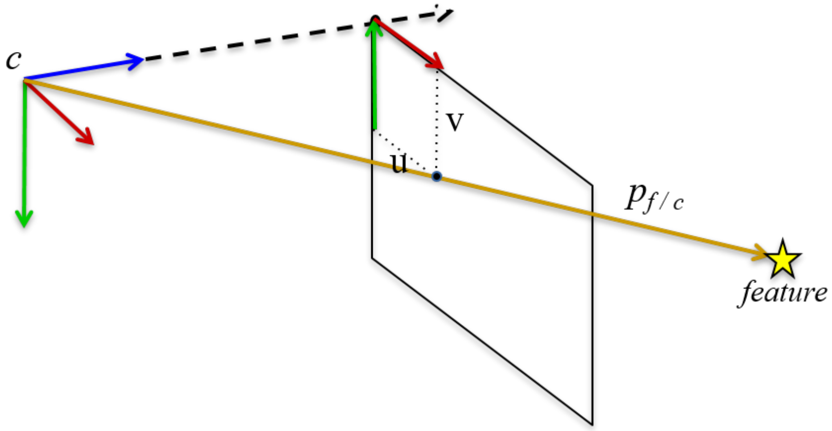


Figure 2.4: A Schematic of the Camera Model.

$\mathcal{T}_{c/b}\,\hat{\mathcal{T}}_{b/b'}\,\hat{\mathcal{T}}_{b'/i}$.

Jacobian matrix $C_j = \frac{\partial y_j}{\partial x}|_{\hat{x}}$ is computed as follows:

$$C_j = \left[\begin{array}{cccccc|ccc} \frac{\partial y_j}{\partial\, {}^i\hat{p}_{b/i}} & 0 & \frac{\partial y_j}{\partial\, \delta\hat{\theta}} & 0 & 0 & & 0\cdots & \frac{\partial y_j}{\partial\, {}^i\hat{p}_{f_j/i}} & \cdots 0 \end{array}\right] \tag{2.35}$$

$$\frac{\partial y_j}{\partial\, {}^i\hat{p}_{b/i}} = \left(\frac{\partial y_j}{\partial\, {}^c\hat{p}_{f_j/c}}\right)(-\hat{\mathcal{T}}_{c/i}), \qquad \frac{\partial y_j}{\partial\, {}^i\hat{p}_{f_j/i}} = -\frac{\partial y_j}{\partial\, {}^i\hat{p}_{b/i}}, \tag{2.36}$$

$$\frac{\partial y_j}{\partial\, \delta\hat{\theta}} = \left(\frac{\partial y_j}{\partial\, {}^c\hat{p}_{f_j/c}}\right)\mathcal{T}_{c/b}\left[\,{}^{b'}\hat{p}_{f_j/b}\times\right]$$

$$\left(\frac{\partial y_j}{\partial\, {}^c\hat{p}_{f_j/c}}\right) = \frac{1}{{}^c\hat{Z}_j}\begin{bmatrix} f_u & 0 & -f_u\frac{{}^c\hat{X}_j}{{}^c\hat{Z}_j} \\ 0 & f_v & -f_v\frac{{}^c\hat{Y}_j}{{}^c\hat{Z}_j} \end{bmatrix}, \tag{2.37}$$

where for more detailed derivations, see a reference [74].

2.3.1 Feature Initialization

From Equation (2.34), if j-th measurement y_j on an image is a new feature, then ${}^ip_{f_j/i}$ is unknown so need to be initialized. In the first step of the measurement update, we employ Gauss-Newton least-squares minimization [75, 22] to estimate feature $3D$ position ${}^i\hat{p}_{f_j/i}$. To avoid local minima, we apply the inverse depth parameterization of the feature position [76] that is numerically more stable than the Cartesian parameterization.

We assume that the intrinsic and extrinsic parameters of a stereo camera are known and constant values. c_1, c_2 frames are the left and right camera frame of the stereo, respectively. Since the baseline of the stereo is fixed, rotation $\mathcal{T}_{c_2/c_1}$ and translation ${}^{c_2}p_{c_1/c_2}$ between both cameras are constant and known values. Feature coordinates ${}^{c}[X, Y, Z]^{\mathrm{T}}$ with respect to both cameras are

$$
\begin{aligned}
{}^{c_2}p_{f_j/c_2} &= \mathcal{T}_{c_2/c_1}\, {}^{c_1}p_{f_j/c_1} + {}^{c_2}p_{c_1/c_2} && (2.38)\\
\begin{bmatrix} {}^{c_2}X_j & {}^{c_2}Y_j & {}^{c_2}Z_j \end{bmatrix}^{\mathrm{T}} &= \mathcal{T}_{c_2/c_1} \begin{bmatrix} {}^{c_1}X_j & {}^{c_1}Y_j & {}^{c_1}Z_j \end{bmatrix}^{\mathrm{T}} + {}^{c_2}p_{c_1/c_2} && (2.39)\\
&= {}^{c_1}Z_j \left(\mathcal{T}_{c_2/c_1} \begin{bmatrix} \frac{{}^{c_1}X_j}{{}^{c_1}Z_j} \\ \frac{{}^{c_1}Y_j}{{}^{c_1}Z_j} \\ 1 \end{bmatrix} + \frac{1}{{}^{c_1}Z_j}\, {}^{c_2}p_{c_1/c_2} \right) && (2.40)\\
&= {}^{c_1}\hat{Z}_j \left(\mathcal{T}_{c_2/c_1} \begin{bmatrix} \hat{u}_{j,1}\,/f_{u_1} \\ \hat{v}_{j,1}\,/f_{v_1} \\ 1 \end{bmatrix} + \frac{1}{{}^{c_1}\hat{Z}_j}\, {}^{c_2}p_{c_1/c_2} \right). && (2.41)
\end{aligned}
$$

Simplifying Equation (2.41),

$$
\begin{bmatrix} {}^{c_2}X_j \\ {}^{c_2}Y_j \\ {}^{c_2}Z_j \end{bmatrix} = {}^{c_1}\hat{Z}_j \left(\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} + \frac{1}{{}^{c_1}\hat{Z}_j} \begin{bmatrix} tr_x \\ tr_y \\ tr_z \end{bmatrix} \right), \tag{2.42}
$$

where m_x, m_y, and m_z are scalar functions of given j-th measurement and constant extrinsic rotation matrix. Based on Equation (2.33), since measurement equations from the stereo camera are

$$
y_j = \begin{bmatrix} {}^{f_j}u_1 \\ v_{j_1} \\ u_{2_j} \\ {}^{2}v_j \end{bmatrix} = \begin{bmatrix} f_{u_1} \frac{{}^{c_1}X_j}{{}^{c_1}Z_j} \\ f_{v_1} \frac{{}^{c_1}Y_j}{{}^{c_1}Z_j} \\ f_{u_2} \frac{{}^{c_2}X_j}{{}^{c_2}Z_j} \\ f_{v_2} \frac{{}^{c_2}Y_j}{{}^{c_2}Z_j} \end{bmatrix} + \zeta_j,
$$

right c_2 camera measurements are expressed in $Ax = b$ form.

$$\begin{bmatrix} \hat{u}_{j,2} \,/\, f_{u_2} \\ \\ \hat{v}_{j,2} \,/\, f_{v_2} \end{bmatrix} = \begin{bmatrix} \frac{m_x + (\, tr_x \,/\, {}^{c_1}\hat{Z}_j \,)}{m_z + (\, tr_z \,/\, {}^{c_1}\hat{Z}_j \,)} \\ \frac{m_y + (\, tr_y \,/\, {}^{c_1}\hat{Z}_j \,)}{m_z + (\, tr_z \,/\, {}^{c_1}\hat{Z}_j \,)} \end{bmatrix} \tag{2.43}$$

$$\begin{bmatrix} m_x - (\hat{u}_{j,2} \,/\, f_{u_2})\, m_z \\ \\ m_y - (\hat{v}_{j,2} \,/\, f_{v_2})\, m_z \end{bmatrix} {}^{c_1}\hat{Z} = \begin{bmatrix} (\hat{u}_{j,2} \,/\, f_{u_2})\, tr_z - tr_x \\ \\ (\hat{v}_{j,2} \,/\, f_{v_2})\, tr_z - tr_y \end{bmatrix}, \tag{2.44}$$

where let

$$x = {}^{c_1}\hat{Z}, \quad A = \begin{bmatrix} m_x - (\hat{u}_{j,2} \,/\, f_{u_2})\, m_z \\ \\ m_y - (\hat{v}_{j,2} \,/\, f_{v_2})\, m_z \end{bmatrix}, \quad b = \begin{bmatrix} (\hat{u}_{j,2} \,/\, f_{u_2})\, tr_z - tr_x \\ \\ (\hat{v}_{j,2} \,/\, f_{v_2})\, tr_z - tr_y \end{bmatrix}. \tag{2.45}$$

Hence, Gauss-Newton least-squares minimization estimates depth ${}^{c_1}Z$ of left c_1 camera using the pseudo-inverse of A:

$$Ax = b \;\Rightarrow\; (A^{\mathrm{T}} A)x = A^{\mathrm{T}}\, b \;\Rightarrow\; \hat{x} = (A^{\mathrm{T}} A)^{-1} A^{\mathrm{T}}\, b.$$

If either estimated depth ${}^{c_1}\hat{Z}$ or ${}^{c_2}\hat{Z}$ is negative, the solution of the minimization is invalid since the feature is always in front of both camera frames observing it. By substituting estimated ${}^{c_1}\hat{Z}$ into Equation (2.41),

$${}^{c_1}\hat{p}_{f_j/c_1} = \begin{bmatrix} (\hat{u}_{j,1} \,/\, f_{u_1})\, {}^{c_1}\hat{Z} & ((\hat{v}_{j,1} \,/\, f_{v_1})\, {}^{c_1}\hat{Z} & {}^{c_1}\hat{Z} \end{bmatrix}^{\mathrm{T}}, \tag{2.46}$$

where $\hat{p}_{f_j/c}$ is not related to the pose of the vehicle. Likewise, if a monocular camera is used instead, c_1 is the camera frame in which the feature was observed at the first time, and c_2 is the camera frame at a different time instance.

The j-th feature $3D$ position with respect to the inertial frame is

$$\begin{aligned}
{}^{i}\hat{p}_{f_j/i} &= \hat{\mathcal{T}}_{i/c_1}\,{}^{c_1}\hat{p}_{f_j/c_1} + {}^{i}\hat{p}_{c_1/i} \\
&= \hat{\mathcal{T}}_{i/b}\,\mathcal{T}_{b/c_1}\,{}^{c_1}\hat{p}_{f_j/c_1} + \left({}^{i}\hat{p}_{b/i} + \hat{\mathcal{T}}_{i/b}\,{}^{b}p_{c_1/b}\right) \\
&= \hat{\mathcal{T}}_{i/b'}\hat{\mathcal{T}}_{b'/b}\,\left(\mathcal{T}_{b/c_1}\,{}^{c_1}\hat{p}_{f_j/c_1} + {}^{b}p_{c_1/b}\right) + {}^{i}\hat{p}_{b/i}. \qquad (2.47)
\end{aligned}$$

The new feature is initialized using only one image in which the feature is first observed. Although the new feature is initialized, since it still entails uncertainty, the EKF recursively estimates and updates its $3D$ position by augmenting into the state vector.

$$\hat{x} = \begin{bmatrix} \hat{x}_V^{\mathrm{T}} & {}^{i}\hat{p}_{f_j/i}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \qquad (2.48)$$

where $\hat{x}_V$ is the vehicle state vector defined in Equation (2.16). The overall initialization includes the initial value of the feature state and its error covariance assignment. The error covariance of the new feature are initialized using state augmentation with Jacobian J.

$$\begin{bmatrix} P & * \\ * & * \end{bmatrix} = \begin{bmatrix} I \\ J \end{bmatrix} P \begin{bmatrix} I & J^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} P & P\,J^{\mathrm{T}} \\ J\,P & JPJ^{\mathrm{T}} + P_{f_{\mathrm{new}}} \end{bmatrix}, \qquad (2.49)$$

where Jacobian $J = \frac{\partial p_{f/i}}{\partial x}\big|_{\hat{x}}$ is computed as follows:

$$J = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & -\hat{\mathcal{T}}_{i/b'}\,\lceil\,(\,\mathcal{T}_{b/c_1}{}^{c_1}\hat{p}_{f_j/c_1} + {}^{b}p_{c_1/b}\,)\times\rfloor & 0_{3\times6} & \big|0\cdots \end{bmatrix}. \qquad (2.50)$$

$P_{f_{\mathrm{new}}}$ is own uncertainty of the initialized new feature. The error pertains to measurement noise and the error of the least-squares minimization and so on. In fact, since Montiel et al. [76] validate that the initial uncertainty is coded as Gaussian, the EKF including the feature initialization still holds optimality.

Once initialized, the EKF processes the feature state in the prediction-update loop. In the time update of the EKF, we propagate P by Equation (2.7).

$$\begin{bmatrix} \Phi & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} P_{vv} & P_{vf} \\ P_{fv} & P_{ff} \end{bmatrix} \begin{bmatrix} \Phi^{\mathrm{T}} & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} Q_v & 0 \\ 0 & Q_{ff} \end{bmatrix} = \begin{bmatrix} \Phi\, P_{vv}\, \Phi^{\mathrm{T}} + Q_v & \Phi\, P_{vf} \\ P_{fv}\, \Phi^{\mathrm{T}} & P_{ff} + Q_f \end{bmatrix}, \tag{2.51}$$

where state transition matrix $\Phi \approx I + A\,\Delta t$. In addition, we assume that surrounding is static, so the dynamics of features $\dot{\hat{p}}_{f_j/i} = 0$. In the measurement update of the EKF, only tracked features are used for the update. For the efficient management of the map database, if the size of the state vector exceeds than the maximum limit, then the feature with the least number of observations is pruned and marginalized.

CHAPTER 3
LATENCY-ADAPTIVE FILTERING FOR MEASUREMENTS WITH UNKNOWN TIME DELAYS

To fuse measurements of vision data with unknown time delays, this chapter incorporates three correction techniques into state estimation. Similar to the combined parameter-state estimator [43], we directly estimate the unknown part of the delay value as an additional state and simultaneously obtain refined state estimates in the modified Kalman filter that corrects Jacobian, residual, and covariance for compensating of delayed measurements. Testing results of this study on flight dataset show that this approach is more reliable than the existing other approaches for state estimation using measurements with unknown time delays.

3.1 Definition of Time Delays

Based on dead reckoning, the EKF propagates state x and its error covariance P at time t when IMU sensor data a_{raw} and ω_{raw} are measured. Since an IMU is a discrete-time sensor, the time update of the EKF is processed in discrete time step $k = \text{(integer)}\,(t\,/\,\Delta\, t_{\text{IMU}})$, where continuous time $t \in [0, t_{\text{final}}]$ and $\Delta\, t_{\text{IMU}}$ is the sampling rate of the IMU. $\Delta\, t_{\text{IMU}}$ is generally almost constant since a micro controller such as Arduino and Pixhawk calculates precise timestamps in millisecond for each IMU measurement. Next, whenever a new vision data from an image are arrived at the filter, the EKF performs the measurement update for correcting the state estimate and its error covariance. As introduced in Section 1.1, various reasons such as image processing produce time delays that the time stamps of vision data contain. For clarity, this section defines the latency in details.

Latency is the time difference between when an image was grabbed and when vision data from the image are updated in the filter, shown in Figure 3.1. That is, true delays $\Delta\, t_d$

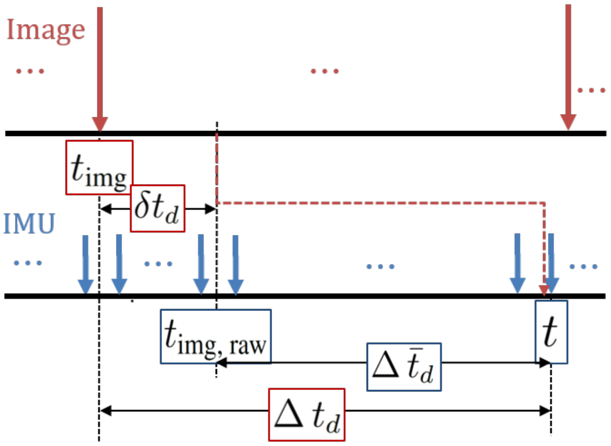


Figure 3.1: Data Streams of the IMU and the Delayed Vision Data

is written as

$$t = t_{\text{img}} + \Delta\,t_d, \tag{3.1}$$

where t is current IMU time and t_{img} is the time when current image was captured. Since cameras do not equip with clock, we do not know exact time when images are grabbed. The timestamps of each image are encoded by indirect ways such as triggers. In other words, true image time t_{img} constitutes readable timestamps $t_{\text{img, raw}}$ and unknown δt_d such as clock bias and drift. Let us define time differences $\Delta\,\bar{t}_d$ between the time readouts of sensors as follows:

$$t_{\text{img, raw}} = t_{\text{img}} + \delta t_d \tag{3.2}$$

$$\Delta\,\bar{t}_d \coloneqq t - t_{\text{img, raw}} = \Delta\,t_d - \delta t_d \tag{3.3}$$

$$\Delta\,t_d = \Delta\,\bar{t}_d + \delta t_d, \tag{3.4}$$

where $\Delta\,\bar{t}_d$ and δt_d are the approximately known and the unknown parts of true delays t_d, respectively.

3.2 Approximately Known Part of Time Delays

$\Delta\,\bar{t}_d$ is either a fixed value determined by offline beforehand tuning or readable differences between the time stamps of image and the time stamps of IMU data. Indeed, regardless of a constant value or readable varying delays, approximate delay $\Delta\,\bar{t}_d$ is a known value. Let the discrete steps of the approximately known part be $d =$ (integer) $(\Delta\,\bar{t}_d\,/\,\Delta\,t_{\text{IMU}})$, where (integer) means type conversion to integer from other types; that is, d is the quotient of division $\frac{\Delta\,\bar{t}_d}{\Delta\,t_{\text{IMU}}}$.

3.2.1 Jacobian and Residual - "Baseline Correction"

Since δt_d is unknown, we first consider the only $\Delta\,\bar{t}_d$ term as delays of the system. From the system models given in Sections 2.2 and 2.3, only measurements from the camera model depend on the time delays. To correct Jacobian and residual with approximately known delays, interpolation and quaternion slerp are required.

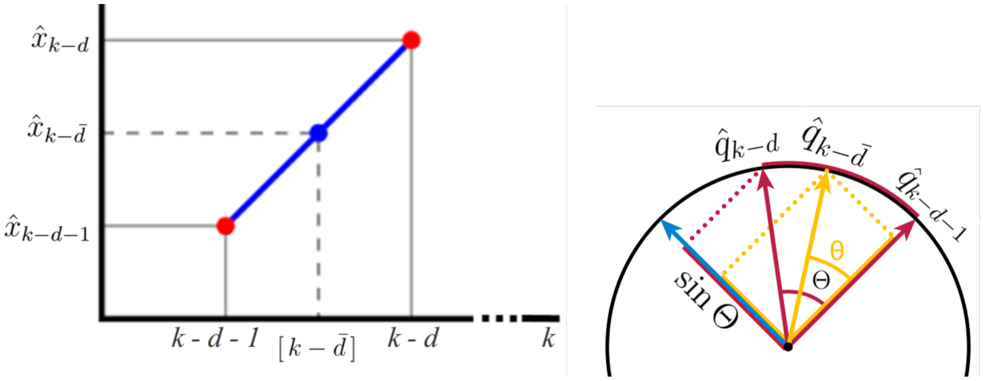
Interpolation

Since $k - d \neq$ (integer) $\left(\frac{t-\Delta\,\bar{t}_d}{\Delta t_{\text{IMU}}}\right)$, we define new time notation $[\,k - \bar{d}\,]$ as

$$[\,k - \bar{d}\,] := \frac{t_{\text{img, raw}}}{\Delta t_{\text{IMU}}} = \frac{t - \Delta\,\bar{t}_d}{\Delta t_{\text{IMU}}}.$$

When time $[\,k - \bar{d}\,]$ is expressed at subscript (e.g., $x_{[\,k-\bar{d}]}$, $P_{[\,k-\bar{d}]}$), we will use the shorthand notation without [ ] (e.g., $x_{k-\bar{d}}$, $P_{k-\bar{d}}$).

Although delay d in discrete-time systems is the number of delayed samples, time $[\,k - \bar{d}\,]$ is not required to be an integer by reading timestamps of each sensor. Since $[\,k - \bar{d}\,]$ is not an integer, we cannot directly access the values of either $\hat{x}_{k-\bar{d}}$ or its corresponding error covariance $P_{k-\bar{d}}$, so relevant interpolation is required instead. Mathematically, linear interpolation constructs a new data point within the range of two known adjacent data points by the same slope of two lines [77]. Let us take the nearest integer time step $k - d$,



(a) Linear Inteportation (b) Quaternion Slerp

Figure 3.2: Examples of Interpolation and Slerp

which is greater than or equal to $[\,k-\bar{d}\,]$, shown in Figure 3.2(a). With two data points, either $(k-d-1,\, \hat{x}_{k-d-1})$ and $(k-d,\, \hat{x}_{k-d})$ or $(k-d-1,\, P_{k-d-1})$ and $(k-d,\, P_{k-d})$, the interpolants at time $[\,k-\bar{d}\,]$ are given by

$$\hat{x}_{k-d} - \hat{x}_{k-d-1} = \frac{\hat{x}_{k-d} - \hat{x}_{k-\bar{d}}}{k-d-[\,k-\bar{d}\,]}$$

$$\hat{x}_{k-d} - \hat{x}_{k-\bar{d}} \simeq \left(\frac{t}{\Delta\, t_{\text{IMU}}} - d - \frac{t-\Delta\,\bar{t}_d}{\Delta t_{\text{IMU}}}\right)(\hat{x}_{k-d} - \hat{x}_{k-d-1}) \tag{3.5}$$

$$\hat{x}_{k-\bar{d}} = \left(1 - \frac{\Delta\,\bar{t}_d}{\Delta t_{\text{IMU}}} + d\right)\hat{x}_{k-d} + \left(\frac{\Delta\,\bar{t}_d}{\Delta t_{\text{IMU}}} - d\right)\hat{x}_{k-d-1}, \tag{3.6}$$

where $k = (\text{integer})\left(\frac{t}{\Delta\, t_{\text{IMU}}}\right) \approx \frac{t}{\Delta\, t_{\text{IMU}}}$ in Equation (3.5). Likewise,

$$P_{k-\bar{d}} = \left(1 - \frac{\Delta\,\bar{t}_d}{\Delta t_{\text{IMU}}} + d\right)P_{k-d} + \left(\frac{\Delta\,\bar{t}_d}{\Delta t_{\text{IMU}}} - d\right)P_{k-d-1}.$$

Quaternion Slerp

Although we compute the interpolants at time $[\,k-\bar{d}\,]$ using linear interpolation, because of the constraint and specialty of quaternion, another adequate interpolation is required. Slerp is shorthand for spherical linear interpolation, introduced by Ken Shoemake [78] in the context of quaternion interpolation for the purpose of animating $3D$ rotation. Interpolants refer to constant-speed motion along a unit-radius circle arc, shown in Figure 3.2(b). Based on the fact that any point on the curve is linear combination of the given ends, the geometric formula [78, 79] is

$$\Theta = \cos^{-1}\left(q_{k-d} \cdot q_{k-d+1}\right) \tag{3.7}$$

$$\hat{q}_{k-\bar{d}} = \frac{\sin\left[\left(1 - \frac{\Delta\,\bar{t}_d}{\Delta t_{\text{IMU}}} + d\right)\Theta\right]}{\sin\Theta}\,\hat{q}_{k-d} + \frac{\sin\left[\left(\frac{\Delta\,\bar{t}_d}{\Delta t_{\text{IMU}}} - d\right)\Theta\right]}{\sin\Theta}\,\hat{q}_{k-d-1}, \tag{3.8}$$

where since only unit quaternions are valid rotations, normalization of each quaternion before applying Slerp is a prerequisite.

Θ is a smaller angle between two end quaternions, so we ensure that $-90\,\mathrm{deg} \leq \Theta \leq 90\,\mathrm{deg}$. If the dot product in Equation (3.7) is negative, Slerp does not represent the shortest path. To prevent long paths, we negate one of end quaternions since q and $-q$ are equivalent when the negation is applied to all four components. If the input quaternions are too close, then interpolants by linear interpolation explained in Section 3.2.1 is acceptable. Otherwise, since the dot product is in range of threshold, $\cos^{-1}(\cdot)$ is safe computation.

Baseline Correction

With suitable interpolants at time $[\,k - \bar{d}\,]$, a baseline approach modifies the feature initialization in Section 2.3.1 and the measurement update in Section 2.1.2. At time k, the vision data of an image grabbed at time $(t - \Delta\,t_d)$ arrives at the filter for either the feature initializations or the sequential measurement updates.

If j-th measurement y_j on the last image is a new feature, then from Equations (2.48) and (2.49), state $\hat{x}$ and covariance P at current time k are augmented as follows.

$$\hat{x}_k \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} \hat{x}_k^{\mathrm{T}} & {}^{i}\hat{p}_{f_j/i}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{3.9}$$

$$P_k \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P_k & P_k\,(J_j)_{k-\bar{d}}^{\mathrm{T}} \\ (J_j)_{k-\bar{d}}\,P_k & (J_j)_{k-\bar{d}}\,P_k\,(J_j)_{k-\bar{d}}^{\mathrm{T}} + P_{f_{j\text{new}}} \end{bmatrix}, \tag{3.10}$$

where ${}^{i}\hat{p}_{f_j/i} = \mathcal{T}_{i/b}\big|_{k-\bar{d}}\, {}^{b}\hat{p}_{f_j/b} + \left({}^{i}p_{b/i}\right)_{k-\bar{d}}$ and $(J_j)_{k-\bar{d}} = \frac{\partial p_{f_j/i}}{\partial x}\big|_{\hat{x}_{k-\bar{d}}}$. ${}^{b}\hat{p}_{f_j/b}$ is initialized by Gaussian-Newton least-squares minimization derived in Section 2.3.1. Although we assume static features, since the feature initialization is related to estimated camera pose at the time when the delays begin, corrected Jacobian J_j is required in the initialization steps.

If j-th measurement y_j on the image is a tracked feature, then we correct only residual r and Jacobian C in the following measurement update.

$$K_j = (P_k)_{j-1}\,(C_j)^{\mathrm{T}}_{k-\bar{d}}\left((C_j)_{k-\bar{d}}\,(P_k)_{j-1}\,(C_j)^{\mathrm{T}}_{k-\bar{d}} + R\right)^{-1} \tag{3.11}$$

$$(\hat{x}_k)_j = (\hat{x}_k)_{j-1} + K_j\left(y_j\big|_{t-\Delta\, t_d} - h_j(\hat{x}_{k-\bar{d}})\right) \tag{3.12}$$

$$(P_k)_j = (P_k)_{j-1} - K_j\,(C_j)_{k-\bar{d}}\,(P_k)_{j-1} \tag{3.13}$$

where corrected residual $(r_j)_{k-\bar{d}} = y_j\big|_{t-\Delta\, t_d} - h_j(\hat{x}_{k-\bar{d}})$ and Jacobian $(C_j)_{k-\bar{d}} = \frac{\partial\, h_j(x)}{\partial\, x}\Big|_{\hat{x}_{k-\bar{d}}}$. K_j is sub-optimal Kalman gain computed by current covariance. As sequential Kalman Filtering introduced in Section 2.1.3, if j is the first feature on the current image (i.e., j=0), then assign $(\hat{x}_k)_0 \leftarrow \hat{x}_k^-$, $(P_k)_0 \leftarrow P_k^-$, and if j is the last feature on the current image (i.e., j=N_k), then assign $\hat{x}_k^+ \leftarrow (\hat{x}_k)_N$, $P_k^+ \leftarrow (P_k)_N$. Before measurement updates (3.11) – (3.13), a chi-squared gating test rejects outliers of each measurement. For only this test purpose in the case of baseline correction, we add uncertainty due to time delay. Procedures in Equations (3.9) – (3.13) are referred to as ”baseline correction.”

3.2.2 Cross Covariance - “Covariance Correction”

During the delay period, even though an image was already captured in the past, since vision data from the image have not arrived yet at the filter because of various reasons, the EKF is not ready to perform the measurement update. Indeed, the filter processes only time update in Equations (2.3) – (2.7). When a vision data packet from the image finally arrives and is ready to update in the filter, we simply execute the Jacobian and residual correction in Equations (3.11) – (3.13) using the delayed measurements. However, unlike the baseline correction, if the filter acts the update as if the measurements arrives immediately without delays (like red lines in Figure 3.3), then filter can achieve more accurate estimation quality. In fact, covariance correction presented in this section (like blue lines in Figure 3.3) is a way as if the filter accomplished the general measurement update in Equations (2.8) – (2.10) at the time instant when the image was captured. In other words, red lines in Figure 3.3 are ideal but unrealistic, blue lines in the figure are practical. The red lines process the measurement update first and then time update; however, the order of the processes of the blue lines are opposite. Only the order of the processes has changed.

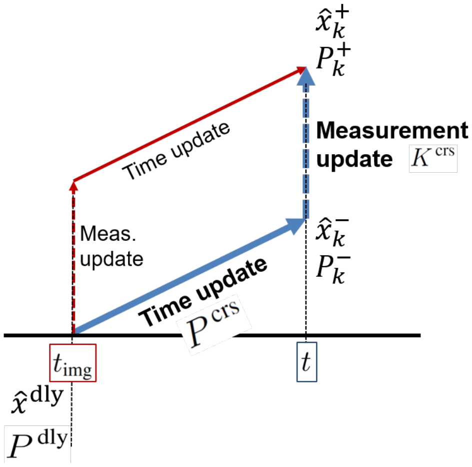


Figure 3.3: A Schematic of Modified Measurement Update Using Covariance Correction

Among a variety of fusing techniques for time-delayed observations discussed in Section-1.2.2, the stochastic cloning [33]-based method (i.e., the Schmidt EKF [30, 31]), is applicable to varying delays and nonlinear functions such as the vehicle and camera models described in Sections 2.2 and 2.3, respectively. Thus, this study modifies the method for finding the optimal navigation solution of vision-aided inertial navigation systems.

Let us introduce new notation P^{dly}. P^{dly} is P covariance matrix at the time when the true delays begin. In the scope of this section, $P^{\text{dly}} \simeq P_{k-\bar{d}}$. In addition, when this section uses corrected residual $(r_j)_{k-\bar{d}}$ and Jocobians $(J_j)_{k-\bar{d}}$, $(C_j)_{k-\bar{d}}$, we will use their shorthand notations as r_j and J_j, C_j, respectively. That is, each residual and Jacobian is corrected based on Section 3.2.1. In addition to the baseline correction, we correct error covariance in both the feature initialization and the measurement update when delayed vision data are available in the filter.

If j-th feature measurement y_j on the recent image is a new feature, the augmentation of $P^{\,\text{dly}}$ in the feature initialization is similar to Equation (3.10). On the other hand, since Jacobian J_j is computed at the time when the delays begin, the augmentation of covariance matrix P_k at current time is following by a different way.

$$P_k \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P_k & 0 \\ 0 & d\,Q_f \end{bmatrix} \tag{3.14}$$

$$P^{\,\text{dly}} \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P^{\,\text{dly}} & P^{\,\text{dly}}\,J_j^{\text{T}} \\ J_j\,P^{\,\text{dly}} & J_j\,P^{\,\text{dly}}\,J_j^{\text{T}} + P_{f_{j_{\text{new}}}} \end{bmatrix}, \tag{3.15}$$

where $d = (\text{integer})\,\frac{\Delta\,\bar{t}_d}{\Delta\,t_{\text{IMU}}}$ and $J_j = \frac{\partial p_{f_j/i}}{\partial x}\Big|_{\hat{x}_{k-\bar{d}}}$. State estimate $\hat{x}_k$ is augmented by Equation (3.9).

When j-th delayed vision data y_j is ready to update at time k, we modify the measurement update steps of the sequential Kalman filtering as follows:

$$S_j = C_j\,\left(P^{\,\text{dly}}\right)_{j-1}\,C_j^{\text{T}} + R$$

$$K_j^{\text{crs}} = \left(P^{\,\text{crs}}\right)_{j-1}\,C_j^{\text{T}}\,S_j^{-1} \tag{3.16}$$

$$\left(\hat{x}_k\right)_j = \left(\hat{x}_k\right)_{j-1} + K_j^{\text{crs}}\,r_j \tag{3.17}$$

$$\left(P_k\right)_j = \left(P_k\right)_{j-1} - K_j^{\text{crs}}\,C_j\,\left(P^{\,\text{crs}}\right)_{j-1}^{\text{T}}, \tag{3.18}$$

where $r_j = y_j\big|_{t-\Delta\,t_d} - h_j(\hat{x}_{k-\bar{d}})$ and $C_j = \frac{\partial\,h_j(x)}{\partial\,x}\Bigg|_{\hat{x}_{k-\bar{d}}}$. $P^{\,\text{crs}}$ is the relevant cross-covariance term during the delay period. This term, which fuses a current prediction of the state with an observation related to the lagged state of the system, is used for formulating modified Kalman gain matrix K^{crs}. Equation (3.18), like (2.10), still holds Jeseph's form [68] that preserves the symmetry of the updated covariance and ensures its the positive definiteness.

By sequential update provisions, the state estimate and covariance at time $[\,k - \bar{d}\,]$ are also updated as follows:

$$K_j^{\text{dly}} = \left(P^{\,\text{dly}}\right)_{j-1} C_j^{\text{T}}\, S_j^{-1} \tag{3.19}$$

$$(\hat{x}_{k-\bar{d}})_j = (\hat{x}_{k-\bar{d}})_{j-1} + K_j^{\text{dly}}\, r_j \tag{3.20}$$

$$\left(P^{\,\text{dly}}\right)_j = \left(P^{\,\text{dly}}\right)_{j-1} - K_j^{\text{dly}}\, C_j\, \left(P^{\,\text{dly}}\right)_{j-1}. \tag{3.21}$$

At time t_{img}, when cameras open for capturing the image, the cross-covariance matrix is initialized with covariance at that time; that is, $P^{\,\text{crs}} \leftarrow P^{\,\text{dly}} \approx P_{k-\bar{d}}$. During the delay period, from time $[\,k - \bar{d}\,]$ to current time k, if no other measurements are fused into the filter, the cross covariance is only propagated by the following computation based on the Schmidt-Kalman filter [30, 33, 32].

$$\mathit{\Phi}^{\,\text{crs}} = \prod_{i=k-1}^{k-d} \begin{bmatrix} \Phi_i & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} \left(\prod_{i=k-1}^{k-d} \Phi_i\right) & 0 \\ 0 & I \end{bmatrix} \tag{3.22}$$

$$P^{\,\text{crs}} = \mathit{\Phi}^{\,\text{crs}}\, P^{\,\text{dly}} \tag{3.23}$$

$$= \begin{bmatrix} \left(\prod_{i=k-1}^{k-d} \Phi_i\right) P_{vv}^{\text{dly}} & \left(\prod_{i=k-1}^{k-d} \Phi_i\right) P_{vf}^{\text{dly}} \\ P_{fv}^{\text{dly}} & P_{ff}^{\text{dly}} \end{bmatrix} \tag{3.24}$$

where Φ is the state transition matrix defined in Equation (2.6). In the sequential measurement update, based on updated $\left(P^{\,\text{dly}}\right)_j$ in Equation (3.21), updating $(P^{\,\text{crs}})_{j-1}$ is straightforward as follows:

$$(P^{\,\text{crs}})_j = \mathit{\Phi}^{\,\text{crs}}\, \left(P^{\,\text{dly}}\right)_j \tag{3.25}$$

If other measurements from other sensors such as an altimeter and GPS are fused during the delay period, then $P^{\,\text{dly}}$ and cross covariance $P^{\,\text{crs}}$ are also recursively updated using the Kalman gain of the other measurements. For this case, Equation (3.22) – (3.25) do not hold any longer. For more details, see Appendix C. All modification in this section is referred to as ”covariance correction.” Furthermore, the optimality of this covariance correction is guaranteed based on the fact that the standard Kalman filter is an optimal fitler since Appendix C proves that the covariance correction is identical to the standard EKF. Hence, the proposed correction still holds its optimality. Section 3.4 will describe ways of its efficient implementation.

3.3 Unknown Part of Time Delays - “Online Calibration”

Although residual, Jacobians, covariance are corrected for measurements with time delays, if $\Delta \bar{t}_d$ is uncertain readouts or δt_d is the larger portion of true delays, we cannot guarantee the reliability of the correction algorithm (Figure 3.4). For robustness of vision-aided navigation systems, we need to additionally investigate the unknown part of true delays. Figure 3.4 shows three corrections in the latency-adaptive filter presented in Chapter 3.

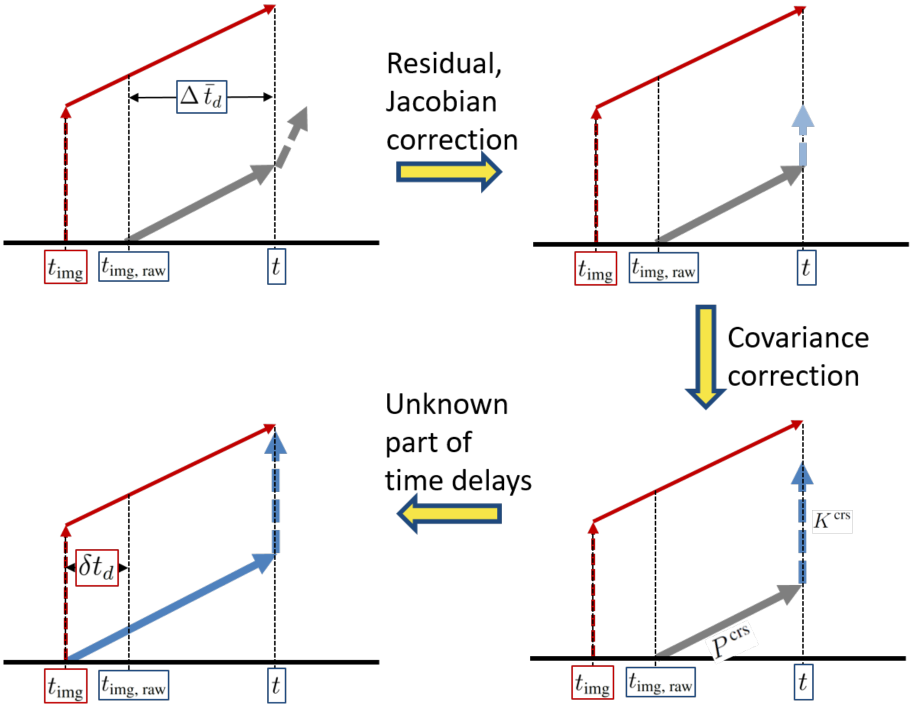


Figure 3.4: Three Corrections in the Latency-Adaptive Filter

From the standard Kalman filter, if one does not account for time delay, propagation and measurement update look like grey lines in Figure 3.4. For the last correction, we estimate the unknown part of time delays to obtain more precise time instant when the delays begin. As discussed in Section 1.1, unknown phenomena such as clock bias, drift, skews, asynchronization cause δt_d, so δt_d may be a positive or negative value.

State estimation theory can be used to estimate not only the states but also the unknown parameters of the system [80]. Numerous researchers [81, 82, 83] have proved that state augmentation functions are easy to use with state observers, so we enable design a state observer by state augmentation to estimate the unknown part of the time delays. To estimate unknown delay value δt_d, we first augment state estimates $\hat{x}_V$ and covariance P_{vv} of the vehicle as follows:

$$x_V \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} x_V^{\mathrm{T}}, & \delta t_d \end{bmatrix}^{\mathrm{T}}, \quad P_{vv} \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P_{vv} & P_{v\,\delta t_d} \\ P_{\delta t_d\,v} & P_{\delta t_d} \end{bmatrix}. \tag{3.26}$$

Like the modeling of the IMU biases in Equations (2.17) and (2.18), we model the dynamics of δt_d using a small artificial noise term

$$\dot{\delta t}_d = \eta_d, \quad \dot{\hat{\delta t}}_d = 0, \tag{3.27}$$

where η_p is a random walk rate that allows the EKF to change its estimate of δt_d; that is, the power spectral density of η_p represents the variability of δt_d. In fact, this is a conventional random walk model for an unknown parameter that may be varying—commonly seen for things like gyro bias, as done here. If additional modeling information about the way time delays are expected to vary is known, then it could be captured here with a more complex model.

Let us rewrite the definition of time delays.

$$\begin{aligned} t - t_{\text{img}} &= \Delta\, t_d \\ \Delta\, t_d &= \Delta\, \bar{t}_d + \delta t_d = (t - t_{\text{img, raw}}) + \delta t_d. \end{aligned}$$

For clarity, we define new time notation $[\, k - \hat{d}\,]$ as

$$\begin{aligned} [\, k - \hat{d}\,] &:= \frac{t_{\text{img, raw}} - \hat{\delta t}_d}{\Delta t_{\text{IMU}}} \approx \frac{t_{\text{img}}}{\Delta t_{\text{IMU}}} \\ &= \frac{t - (\Delta\, \bar{t}_d + \hat{\delta t}_d)}{\Delta t_{\text{IMU}}}, \end{aligned}$$

where now time $[\, k - \hat{d}\,]$ is the most precise time instant when the image was captured.

To apply the relevant interpolation techniques in Sections 3.2.1 and 3.2.1 to the state estimates and covariance at time $[\, k - \hat{d}\,]$, we access their values at the nearest integer time step $k - s$, where $s = \text{(integer)} \left(\Delta\, \bar{t}_d + \hat{\delta t}_d\right) / \Delta t_{\text{IMU}}$. In other words, s, discrete delayed samples including estimated latency, is greater than or equal to $[\, k - \hat{d}\,]$, shown in Figure 3.1.

To operate the augmented system, we match its dimension by augmenting other matrices. In the time update, since $\dot{\hat{\delta t}}_d = 0$, the state transition matrix and the process noise covariance matrix are augmented

$$\Phi \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} \Phi & 0 \\ 0 & I \end{bmatrix}, \quad Q_v \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} Q_v & 0 \\ 0 & Q_d \end{bmatrix} \tag{3.28}$$

where I is due to $\dot{\hat{\delta t}}_d = 0$ and the Gaussian white noise $\eta_d \sim \mathcal{N}(0, Q_d)$. Under assumption of static features, since estimated latency δt_d is pertain to only vision measurements, we compute augmented elements of Jacobian matrices J and C [41, 43]. In fact, from Equation-

(2.50), Jacobian J_j in the feature initialization is augmented as follows:

$$(J_j)_{k-\hat{d}} \overset{\text{aug}}{\Longrightarrow} \left[I_{3\times3} \quad 0_{3\times3} \quad \left.\frac{\partial\,\hat{p}_{f_j/i}}{\partial\,\delta\hat{\theta}}\right|_{[k-\hat{d}]} \quad 0_{3\times6} \quad J_{j_{\delta t_d}} \quad \middle| 0\cdots \right],$$

where

$$\begin{aligned}
J_{j_{\delta t_d}} &= \left.\frac{\partial\,\hat{p}_{f_j/i}}{\partial\,\hat{\delta t}_d}\right|_{[k-\hat{d}]} \simeq \left.\frac{\partial\,\hat{p}_{f_j/i}}{\partial\,x}\right|_{\hat{x}_{k-\hat{d}}} \cdot \left.\frac{\partial\,x}{\partial\,t}\right|_{[k-\hat{d}]} \cdot \left.\frac{\partial\,t}{\partial\,\hat{\delta t}_d}\right|_{[k-\hat{d}]} \\
&= (J_j)_{k-\hat{d}}\,\dot{\hat{x}}_{k-\hat{d}} \\
&= \left[I_{3\times3} \quad 0_{3\times3} \quad \left.\frac{\partial\,\hat{p}_{f_j/i}}{\partial\,\delta\hat{\theta}}\right|_{[k-\hat{d}]} \quad 0_{3\times3} \quad 0_{3\times3} \quad \middle| 0\cdots \right] \begin{bmatrix} {}^i\dot{\hat{p}}_{b/i} \\ {}^i\dot{\hat{v}}_{b/i} \\ \delta\dot{\hat{\theta}} \\ 0 \\ 0 \\ \hline (0\cdots0)^{\mathrm{T}} \end{bmatrix}_{[k-\hat{d}]} \\
&= \left({}^i\hat{v}_{b/i}\right)_{k-\hat{d}} + \left(\left.\frac{\partial\,\hat{p}_{f_j/i}}{\partial\,\delta\hat{\theta}}\right|_{[k-\hat{d}]} \right) \delta\dot{\hat{\theta}}_{k-\hat{d}}
\end{aligned} \tag{3.29}$$

Furthermore, from Equation (2.35), augmented Jacobian C_j in the measurement update is

$$(C_j)_{k-\hat{d}} \overset{\text{aug}}{\Longrightarrow} \left[\frac{\partial\,y_j}{\partial\,{}^i\hat{p}_{b/i}} \quad 0 \quad \frac{\partial\,y_j}{\partial\,\delta\hat{\theta}} \quad 0 \quad 0 \quad \frac{\partial\,y_j}{\partial\,\hat{\delta t}_d} \quad \middle| \, 0\cdots \quad \frac{\partial\,y_j}{\partial\,{}^i\hat{p}_{f_j/i}} \quad \cdots 0 \right]_{[k-\hat{d}]},$$

where

$$\begin{aligned}
\left.\frac{\partial\,y_j}{\partial\,\hat{\delta t}_d}\right|_{[k-\hat{d}]} &\simeq \left.\frac{\partial\,y_j}{\partial\,x}\right|_{\hat{x}_{k-\hat{d}}} \cdot \left.\frac{\partial\,x}{\partial\,t}\right|_{[k-\hat{d}]} \cdot \left.\frac{\partial\,t}{\partial\,\hat{\delta t}_d}\right|_{[k-\hat{d}]} = (C_j)_{k-\hat{d}}\,\dot{\hat{x}}_{k-\hat{d}} \\
&= \left(\left.\frac{\partial\,y_j}{\partial\,{}^i\hat{p}_{b/i}}\right|_{[k-\hat{d}]} \right) \left({}^i\hat{v}_{b/i}\right)_{k-\hat{d}} + \left(\left.\frac{\partial\,y_j}{\partial\,\delta\hat{\theta}}\right|_{[k-\hat{d}]} \right) \delta\dot{\hat{\theta}}_{k-\hat{d}}.
\end{aligned} \tag{3.30}$$

Here, let us call the combination of the estimation of the unknown latency in this section with the baseline correction ”online calibration.” Therefore, to reliably estimate the state variable and effectively compensate the total delays, we incorporate all three corrections, called ”latency-adaptive filtering.”

3.4 Implementation

This section presents everything to solve the problem, and Figure 3.5 illustrates a flow chart of the overall process.

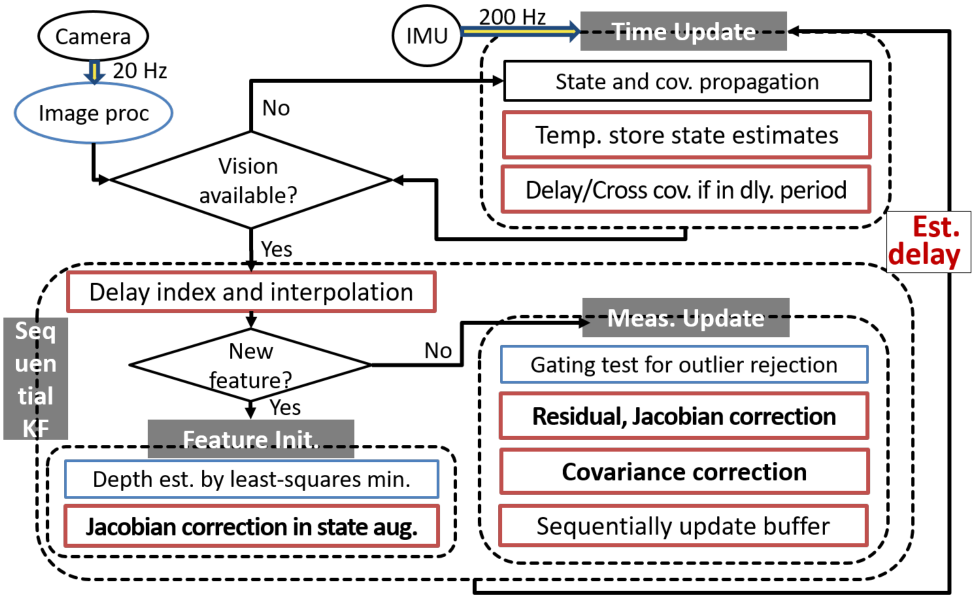

Camera
20 Hz
Image proc
IMU
200 Hz
Time Update
State and cov. propagation
Temp. store state estimates
Delay/Cross cov. if in dly. period
No
Vision available?
Yes
Est. delay
Delay index and interpolation
Seq uen tial KF
New feature?
No
Yes
Meas. Update
Gating test for outlier rejection
Residual, Jacobian correction
Covariance correction
Sequentially update buffer
Feature Init.
Depth est. by least-squares min.
Jacobian correction in state aug.

Figure 3.5: A Flow Chart of the Overall Process of the Latency-Adaptive Filtering

3.4.1 Forward Computation of Cross Covariance

Even though delays begins Δt_d time prior, estimated delay value $\hat{t}_d$ is only accessible when delay finished. That is, during the delay period from t_{img} to t, $\Delta\hat{t}_d$ is unknown yet. $\Delta\hat{t}_d$ is estimated at current time k. Since estimated delay value $\Delta\hat{t}_d$ is unknown up to time k, we are not sure when the covariance correction begins computing cross covariance P^{crs}. Theoretically, when $\Delta\hat{t}_d$ is estimated at time k, we compute P^{dly} and P^{crs} by backward from time k to time $[\,k-\hat{d}\,]$ with saved Jacobians and covariance during the delay period. This is ideal computation, but not realistic. Backward computation that used in [43] is impossible for real-time operations since storing large matrices such as sequences of Jacobian and covariance matrices allocates huge memory uses. Furthermore, the backward computing is not efficient because it iterates backward at time k like batch processing.

Instead, for real-time framework, an approximated way of forward computation of cross covariance is introduced. Since $\hat{\delta t}_d = 0$, we assume that the time delay does not change in state propagation during the delay period, so a posteriori estimate of time delay when the last measurement update is assumed to be a priori estimate of the delay at current time. Next, under this assumption, we predict when the time delay of the next image begins. At the predicted time instant, we store the covariance matrix once for P^{dly} and recursively calculate Φ^{crs} for P^{crs}.

3.4.2 Summarized Algorithm

When the size of the state after augmentation in the feature initialization steps exceeds a maximum threshold, we prune the number of features in database. The system in this study finds an index for the best place to insert a new point in the database. The one with the least number of observations or frequent outliers is marginalized. Unlike Lee at al. [43], this thesis does not estimate the total parts of time delays, so the latency-adaptive filter does not entail a specific constraints. That is, this study estimates only unknown part δt_d that is a possibly positive or negative value. To save computation, constrained Kalman filtering is not necessary. Instead, interpolation and quaternion Slerp explained in Section 3.2.1 are tractable.

From the definition of time delays presented in Section 3.1, total time delay is not estimated as negative. For example, if estimated delay is negative (i.e., an exceeded index), estimation is impossible since this case is forecasting states or obtaining measurements from the future, so the total delay has to be bounded by zero. Moreover, in the sequential measurement update, if estimated time delay δt_d is larger than sampling time of the IMU, Δt_{IMU}, then we indicate another slot in the delay buffer. Algorithm 1 is a summarized algorithm of overall processes of the latency-adaptive filter.

Algorithm 1 The Latency-Adaptive Filtering

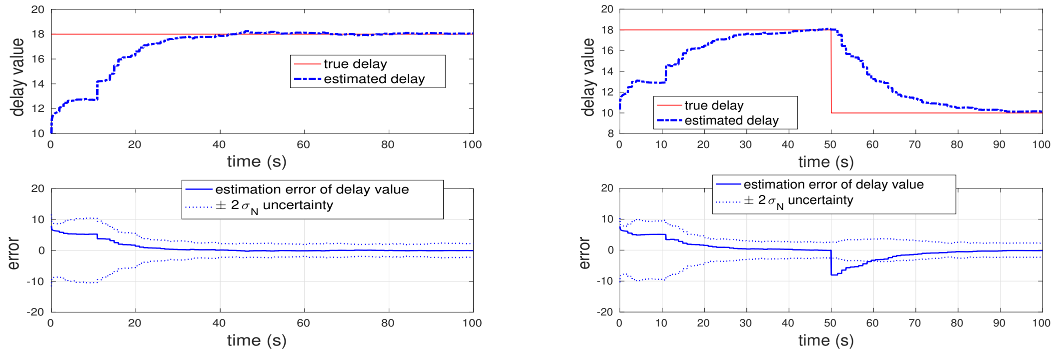
```
Require: $\hat{x}_0^+, P_0^+, Q, R,\ P^{\text{dly}}(= P_0^+), \Phi^{\text{crs}}(= I),\ \chi^2$
for $k = 1 : T$ do
    if new IMU packet arrival then
Time Update:
        $\dot{\hat{x}}_V = f\left(\hat{x}^+_{V_{k-1}},\ a_{\text{raw}}, \omega_{\text{raw}}\right)$    ▷ static features
        Numerically integrate with $\Delta\, t_{\text{IMU}}(= t_k - t_{k-1})$
        $\hat{x}_k^- = \hat{x}_{k-1}^+ + \int_{t_{k-1}}^{t_k} \dot{\hat{x}}(\tau)\, d\tau$
        $\Phi_{k-1} = \exp\left(\int_{t_{k-1}}^{t_k} A(\tau)\, d\tau\right)$    ▷ $A = \frac{\partial f}{\partial x}\big|_{\hat{x}_V}$
        $(P_{vv})_k^- = \Phi_{k-1}\,(P_{vv})_{k-1}^+\,\Phi_{k-1}^{\mathrm{T}} + B_{k-1}\,Q_v\,B_{k-1}^{\mathrm{T}}$    ▷ $B = \frac{\partial f}{\partial \eta}\big|_{\hat{x}_V}$
        $P_k^- = \begin{bmatrix} (P_{vv})_k^- & \Phi_{k-1}\,(P_{vf})_{k-1}^+ \\ (P_{fv})_{k-1}^+\,\Phi_{k-1}^{\mathrm{T}} & (P_{ff})_{k-1}^+ + Q_f \end{bmatrix}$
        Store the state estimates into the delay buffer
        if during delay period then
            $\Phi^{\text{crs}} \leftarrow \Phi_{k-1}\,\Phi^{\text{crs}}$    ▷ recursive
        else
            $P^{\text{dly}} \leftarrow P_k^-$
            $\Phi^{\text{crs}} \leftarrow I$
        end if
    end if
```

3.5 Monte Carlo Simulations

To show actual time delays being estimated accurately, this section simulates a simple example problem by 100 Monte Carlo trials. The vehicle and measurement models of this simulation are direct from Lee and Johnson' previous work [43]. The models are a second order dynamic system with a non-delayed speed measurement and two delayed bearing angles measured from each location of two stations. From Equation (3.28), variance Q_d value of this simulation is $0.25\,[s^2]$. The actual time delay of the delayed measurements in this simulation is $0.9\,[s]$, and this value is identical to 18 delayed samples since the

```
    **if** new vision data packet arrival **then**
        Compute index $\hat{d}^{-}$ of delay
        Interpolate using the state estimates from the buffer
        **for** $j = 1 : \sharp$ of observed features **do**
            **if** new feature **then**
*Feature Initialization*:
                $\hat{p}_{f_j/i} = g_j\left(\hat{x}_{k-\hat{d}},\ y_j\right)$    $\triangleright$ least-squares minimization
                Augment state if feature is valid    $\triangleright$ if positive depth
                $\hat{x}_k \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix}\hat{x}_k^{\mathrm{T}} & \hat{p}_{f_j/i}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$
                $P^{\text{dly}} \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P^{\text{dly}} & P^{\text{dly}}\,J_j^{\mathrm{T}} \\ J_j\,P^{\text{dly}} & J_j\,P^{\text{dly}}\,J_j^{\mathrm{T}} + P_{f_{j_{\text{new}}}} \end{bmatrix}$    $\triangleright J_j = \frac{\partial p_{f_j/i}}{\partial x}\Big|_{\hat{x}_{k-\hat{d}}}$
                $P_k \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P_k & 0 \\ 0 & \hat{d}\,Q_f \end{bmatrix}$
                Prune state vector if exceed maximum
            **else**    $\triangleright$ tracked feature
*Measurement Update*:
                Update if gating test is passed    $\triangleright r_j^{\mathrm{T}} S_j^{-1} r_j \overset{?}{<} \chi_j^2$
                $r_j = y_j - h_j(\hat{x}_{k-\hat{d}})$
                $S_j = (C_j)_{k-\hat{d}}\,P^{\text{dly}}\,(C_j)_{k-\hat{d}}^{\mathrm{T}} + R$    $\triangleright (C_j)_{k-\hat{d}} = \frac{\partial h_j}{\partial x}\Big|_{\hat{x}_{k-\hat{d}}}$
                $P^{\text{crs}} \leftarrow \begin{bmatrix} \Phi^{\text{crs}}\,P_{vv}^{\text{dly}} & \Phi^{\text{crs}}\,P_{vf}^{\text{dly}} \\ P_{fv}^{\text{dly}} & P_{ff}^{\text{dly}} \end{bmatrix}$
                $K_j^{\text{crs}} = P^{\text{crs}}\,(C_j)_{k-\hat{d}}^{\mathrm{T}}\,S_j^{-1}$
                $\Delta\,\hat{x}_k = +K_j^{\text{crs}}\,r_j$    $\triangleright \Delta\hat{t}_d \overset{?}{<} \Delta t_{\text{IMU}}$
                $\Delta\,P_k = -K_j^{\text{crs}}\,(C_j)_{k-\hat{d}}\,(P^{\text{crs}})^{\mathrm{T}}$
                Sequentially update the buffer
                $K_j^{\text{dly}} = P^{\text{dly}}\,(C_j)_{k-\hat{d}}^{\mathrm{T}}\,S_j^{-1}$
                $\Delta\,\hat{x}_{k-\hat{d}} = +K_j^{\text{dly}}\,r_j$
                $\Delta\,P^{\text{dly}} = -K^{\text{dly}}\,(C_j)_{k-\hat{d}}\,P^{\text{dly}}$
            **end if**
        **end for**
        Store index $\hat{d}^{+}$ of the posterior estimated delay
        $P^{\text{dly}} \leftarrow P_k^{+},\ \Phi^{crs} \leftarrow I$
        Erase used slots in the delay buffer
    **end if**
**end for**
```

propagation rate of the simulation is $0.05\ [s]$. Monte Carlo simulations estimate the values



(a) A Static Delay

(b) Varying Delays

Figure 3.6: Estimation of Total Delays in Simulation

of time delays, shown in Figure 3.6. Figure 3.6(a) shows that the estimated delay rapidly converges to the true delay value. That is, the estimation error of the delayed samples gradually decreases toward zero. Moreover, we may wonder whether the latency-adaptive filtering algorithm works when the delay is not static. See Figure 3.6(b) for an answer. Although the values of unknown delays vary over time, estimation resulting from the adaptive method converges to true delay values.

3.6 Flight Datasets Test Results

To validate the reliability of the proposed approach for estimating states and unknown delay values, we test one of benchmark datasets, so-call ”EuRoC MAV datasets [84]”. The visual-inertial sequences of the datasets were recorded onboard a micro aerial vehicle while a pilot manually flied around indoor Vicon environments. For more details, see Appendix A. Although the datasets include noise model parameters from the IMU at rest, we need to tune each variance of process noise covariance Q for the best performance. Likewise, to estimate the unknown part of time delays, we set the standard deviation of random walk η_d in Equation (3.27) as 1.0×10^{-5} since the order of this value is set to same order of the smallest value among the provided noise parameters.

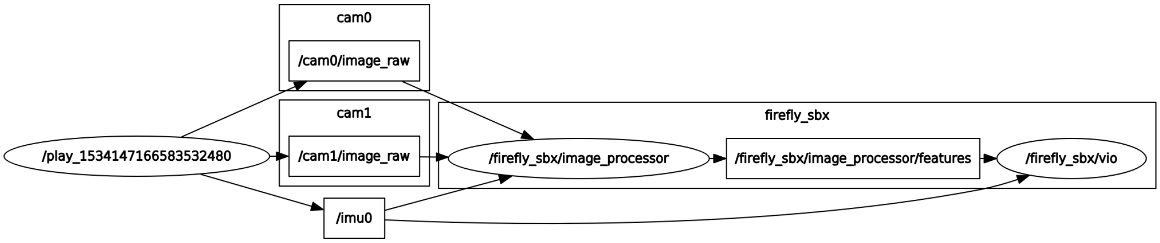

cam0
/cam0/image_raw
cam1
/cam1/image_raw
firefly_sbx
/play_1534147166583532480
/firefly_sbx/image_processor
/firefly_sbx/image_processor/features
/firefly_sbx/vio
/imu0

Figure 3.7: ROS rqt Graph

Given datasets provide various levels of challenging sequences such as faster motion, poor illumination in each environment. To articulate the significance of time delays defined in Section 3.1, we select two datasets of slow motion, called ”EuRoC V1 Easy,” and fast motion, called ”EuRoC V1 Medium.” Since the vehicle in the medium dataset maneuvers twice faster, we hypothesize that the time delays have greater impact on the navigation solution of the medium dataset. Algorithms of image processing and filtering are developed under the robot operating system (ROS) [85], given IMU data and images from the stereo camera are also subscribed under the ROS, shown in Figure 3.7.

The simplest solution to the estimation problem of the given datasets is to run the baseline in Section 3.2.1 that corrects only Jacobians and residual. However, the novel latency-adaptive filter described in Algorithm 1 compensates for delayed measurements at time when the vision data are fused at the filter and estimates the refined state and the delay values. This adaptive filtering follows the processes of all three correction, shown in Figure 3.5.

The EKF estimates relative location from a starting point. Since we do not know the exact absolute location of origin of given datasets, to compare with ground truth data given in the datasets, certain evaluation error metrics such as so-call ”absolute trajectory error [86]” are required. For more details, see Appendix B. After applying the absolute trajectory error, Figure 3.8 illustrates the top down view of the estimated flight trajectory of the medium dataset. Figures 3.9 exhibit estimated x,y,z position and their estimation errors. All estimation errors are bounded within each standard deviation σ bounds. We should expect significant time correlation in error plots and a generally growing error covariance

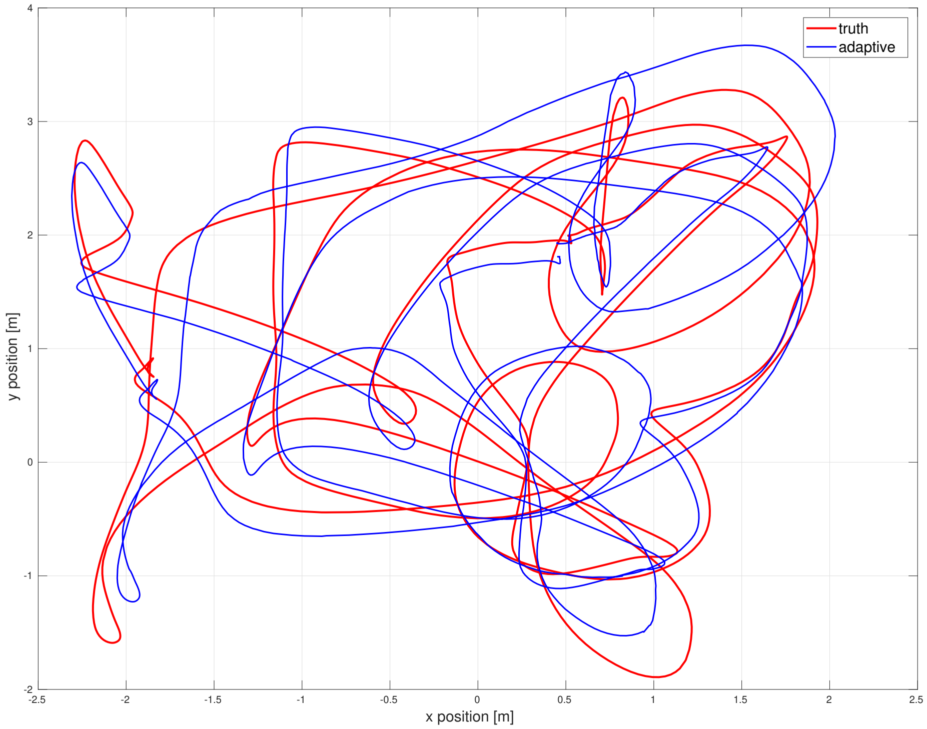


Figure 3.8: Top Down View of Flight Trajectory of the EuRoC V1 Medium Dataset by the Latency-Adaptive Filter

for vision-aided inertial navigation problems like this one. Conceptually, position error gets "locked in" and to the extent new features are being mapped the position error will tend to grow with the length of the trajectory. Starting from the noise model parameters reported for the datasets, the adaptive filter is a well-tuned estimator; for, the performance of doing runs with 3x or 10x (/3 or /10) multiplier on the R term used in the filter is worse for all of those, shown in Table 3.1. In other words, the fact that using those multipliers shows larger RMS estimation errors indicates that our approach is a well-tuned filter.

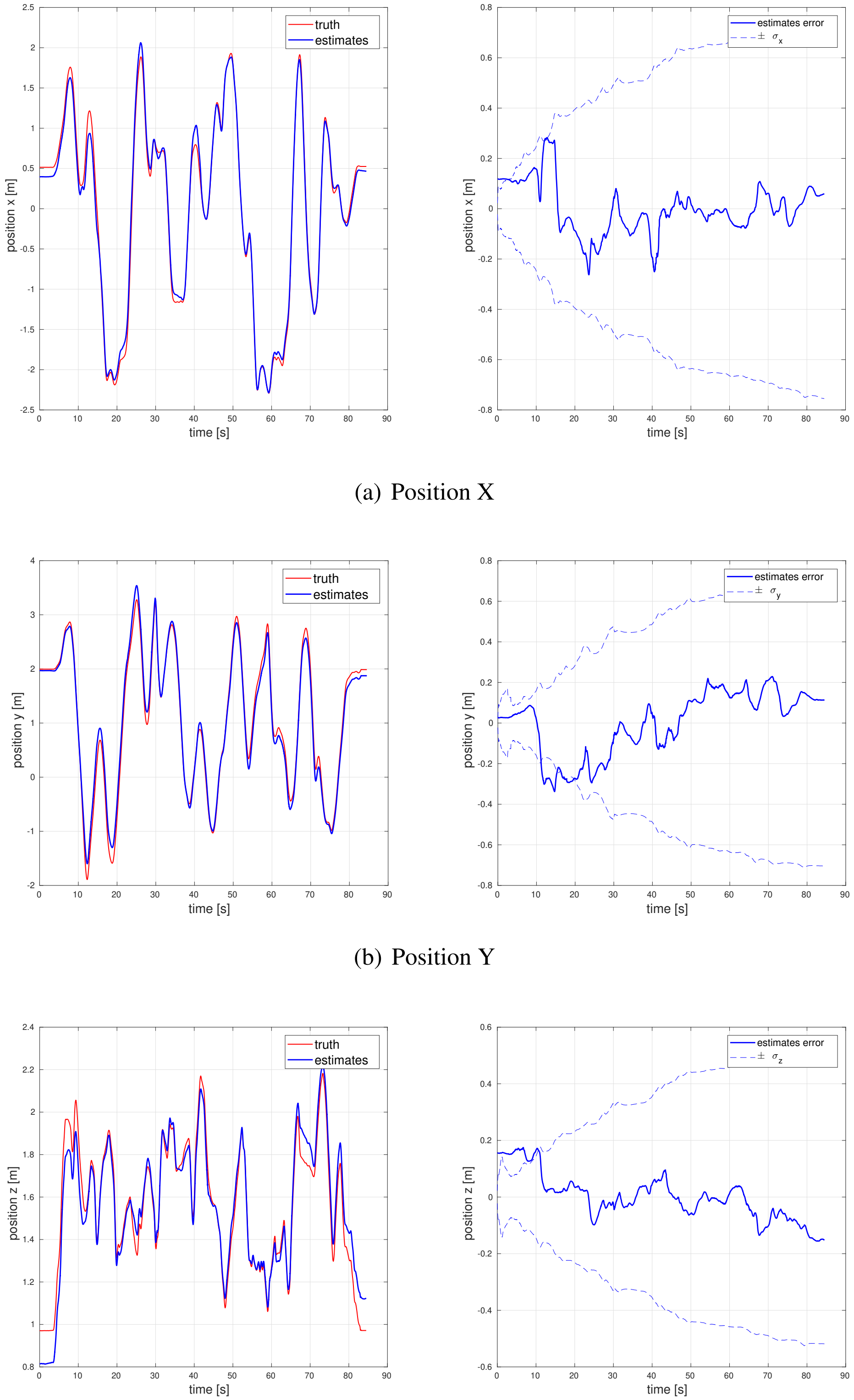
Table 3.1: Indication that the Latency-Adaptive Filter is Well-Tuned for EuRoC V1 Medium Dataset

Multiplier on R	/10	/3	1	x3	x10
RMS error [m]	1.5096	0.1969	0.1619	0.2636	0.2850

Figure 3.10 shows the advantages of each correction in the latency-adaptive filtering by comparing with the baseline and the covariance correction. The baseline discards cross covariance and unknown part of the delays, and although the latency-adaptive filtering might increase the computational effort of the entire system, it significantly improves the accuracy of estimation.

Unlike either the baseline or the covariance correction, the latency-adaptive filter calibrates the unknown part of time delays. Figure 3.11 shows that estimation resulting from the adaptive filter converges to a certain, final delay value, and its variance rapidly decreases although initial uncertainty is high. As shown in Figure 3.12, the average of total estimated delays is around $45\,[ms]$ that could generate about $4\,cm$ drift and offset during the delay period when the vehicle fly at $0.91\,m/s$ average speed. When readable delay values are negative, the timestamps of images might indicate wrong pairs or packet.

Table 3.2 lists the root mean squared (RMS) position errors of cases for sensitivity analysis. Approximately known part of time delays introduced in Section 3.2 is either fixed $\bar{t}_d$ by tuning or readouts $\bar{t}_{d_{\text{raw}}}$ that is the difference of readable timestamps of current IMU and image. In addition, we can directly estimate entire parts of time delays without



(a) Position X

(b) Position Y

(c) Position Z

Figure 3.9: Position and Estimation Error of the EuRoC V1 Medium Dataset by the Latency-Adaptive Filter

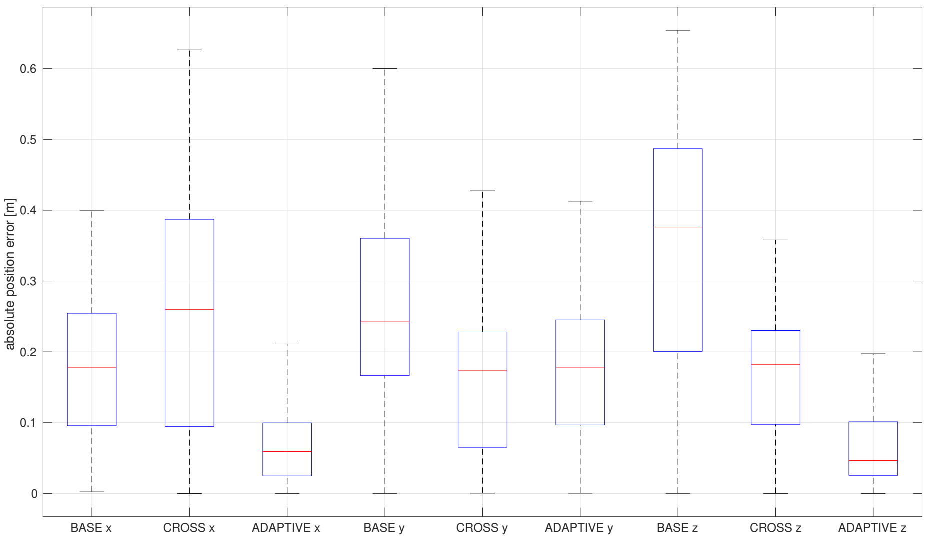


Figure 3.10: Box Plot of Absolute Estimation Error of Position of the EuRoC V1 Medium Dataset by the Latency-Adaptive Filter

information of the approximately known part. For another case, using the final value of estimated unknown part of time delays, we add a fixed $\bar{\delta}t_d$ to the total delays at every time. However, this case might not work when the delay is varying, and we can know the final value only after running the proposed adaptive filter. In other words, before applying the adaptive filtering, fixed $\bar{\delta}t_d$ is still unknown. The estimation results from the latency-adaptive filtering approach depict the influence of the delays and the effectiveness of the corrections in the sensor fusion of the lagged measurements. Fast motion datasets are more sensitive to time delays since the improvement is larger when applied to those datasets.

Table 3.2: Sensitivity Analysis in RMS Position Error [m] of Latency-Adaptive Filtering

	Dataset	EuRoC V1 Easy **Slow Motion** 0.41 m/s, 16.0 deg/s		EuRoC V1 Medium **Fast Motion** 0.91 m/s, 32.1 deg/s	
Method		Cross-Cov OFF	Cross-Cov ON	Cross-Cov OFF	Cross-Cov ON
Fixed $\bar{t}_{d_{\text{const}}}$		0.3376	0.2677	0.4644	0.3135
Entirely Estimated $\hat{t}_d$		0.2282	0.2406	0.4734	0.3538
Readouts $\bar{t}_d$	+ N/A	0.2558	0.2032	0.4163	0.3121
	+ Fixed $\bar{\delta}t_d$	0.2869	0.2285	0.3281	0.2218
	+ Estimated $\hat{\delta}t_d$	0.2019	**0.1461**	0.3353	**0.1619**

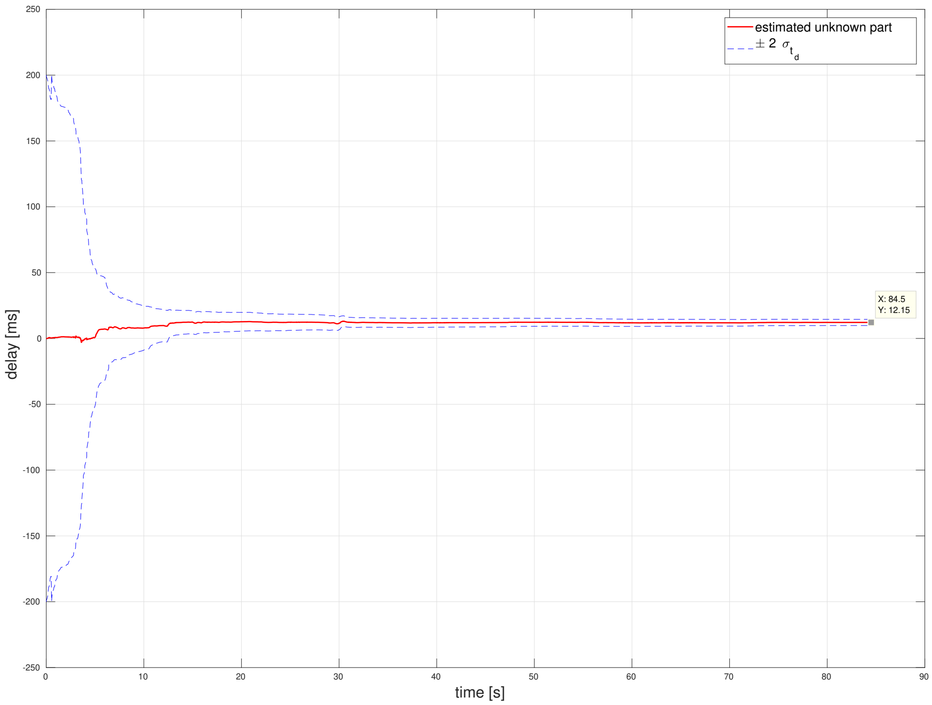


Figure 3.11: Estimation of Unknown Part of Time Delays of the EuRoC V1 Medium Dataset

Although numerous researchers have explored visual inertial odometry of the EuRoC datasets, few of them thoroughly considered measurements with unknown time delays. Table 3.3 reveals that the proposed estimator, the latency-adaptive filter, outperforms the existing state-of-the-art methods, called ”S-MSCKF” and ”SVO+MSF” in which stereo is available.

Table 3.3: Comparison with Other Methods in RMS Position Error [m] of Latency-Adaptive Filtering

Dataset	EuRoC V1 Easy	EuRoC V1 Medium
Method	**Slow Motion** 0.41 m/s, 0.28 rad/s	**Fast Motion** 0.91 m/s, 0.56 rad/s
Latency-Adaptive Filter	0.1461	0.1619
S-MSCKF (stereo-filter)	0.34	0.20
SVO+MSF (loosely coupled)	0.40	0.63

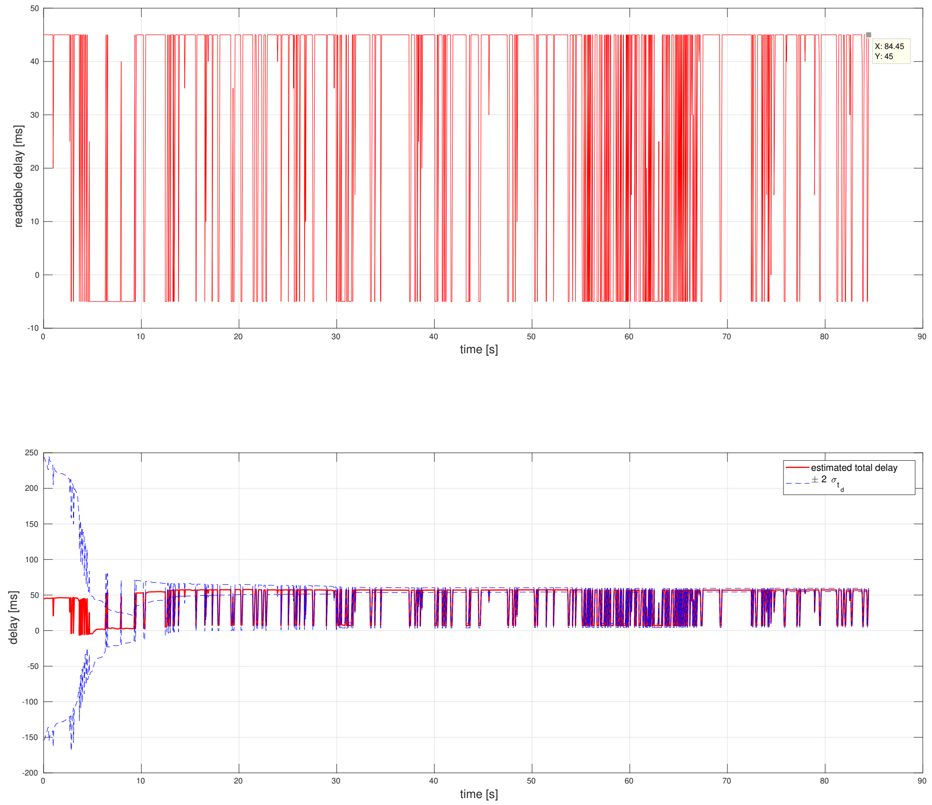


Figure 3.12: Estimation of Time Delays of the EuRoC V1 Medium Dataset

CHAPTER 4
NOISE-ADAPTIVE FILTERING FOR MEASUREMENTS WITH FREQUENT OUTLIERS

For outlier removal in image processing front end, feature correspondence constitutes the following three steps: tracking, stereo matching, and 2-point RANSAC. To estimate the states of V-INS in which vision measurements still remain outliers, this chapter proposes a novel approach [64] that combines a real-time outlier detection technique with an extended version of an outlier robust Kalman filter (ORKF) [53, 54]. Hence, our approach does not restrict noise at either a constant or Gaussian level in filtering. The testing results of a benchmark flight dataset show that our approach leads to greater improvement in robustness under severe environments.

4.1 Outlier Rejection in Image Processing Front End

4.1.1 Feature Correspondence

In this dissertation, a feature detector using the Features from Accelerated Segment Test (FAST) algorithm [87, 88] maintains a minimum number of features in each image. For each new image, a feature extractor using the Kanade–Lucas–Tomasi (KLT) sparse optical flow algorithm [89] tracks the existing features. Even though Paul et al. [27] proved that descriptor-based methods for temporal feature tracking are more accurate than KLT-based methods, since Sun et al. [19] analyzed that descriptor-based methods require much more computing resource with small gain in accuracy, we employ the KLT optical flow algorithm in the image processing front-end of this study. Next, our stereo matching using fixed baseline stereo configuration also applies to the KLT optical flow algorithm for saving computational loads compared to other stereo matching approaches. With the matched features, a 2-point RANdom SAmple Consensus (RANSAC) [57] is applied to remove remaining outliers by utilizing the RANSAC step in the fundamental matrix test [58]. In the scope of this study, we implement the 2-point RANSAC algorithm by simply running one of open source codes.

Similar to [19, 90], our outlier rejection is composed of three steps, shown in Figure 4.1. We assume that features from previous c_1 and c_2 images are outlier-rejected points. The three steps forms a close loop of previous and current frames of left and right cameras. The first step is the stereo matching of tracked features on current c_1 image to c_2 image. The next steps are applying 2-point RANSAC between previous and current images of left camera and another 2-point RANSAC between previous and current images of right camera. For the step 2 and 3, stereo matched features are directly used in each RANSAC.

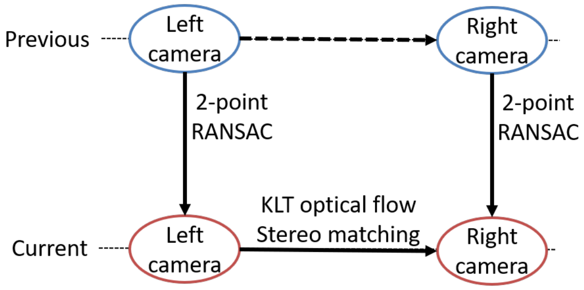


Figure 4.1: Close Loop Steps of Outlier Rejection in Image Processing Front End

4.1.2 Algorithm of Feature Correspondence

Algorithm 2 summarizes the feature correspondence for outlier rejection. For the scope of this thesis, the OpenCV library [91] and open source codes of RANSAC are extremely useful and directly applied. where Pyramid is a type of multi-scale signal representation in which an image is subject to repeated smoothing and sub-sampling.

Algorithm 2 Feature Correspondence for Outlier Rejection

```
Require: Pyramids and outlier-rejected points of previous $c_1$, $c_2$ images
Feature Tracking:
function buildOpticalFlowPyramid(current $c_1$ or $c_2$ image)    ▷ OpenCV
    return pyramd of current $c_1$ or $c_2$
end function
function predictFeatures(outlier-rejected points of previous $c_1$, $\hat{\mathcal{T}}_{\text{curr}\leftarrow\text{prev}}$ of $c_1$, Intrinsic $c_1$)
    return predicted features of current $c_1$
end function
function calcOpticalFlowPyrLK(pyramids of previous and current $c_1$, outlier-rejected points of previous $c_1$, predicted features of current $c_1$)    ▷ OpenCV
    return tracked points of previous $c_1$ and $c_2$, tracked features of current $c_1$
end function
Stereo Matching:
function stereoMatching(tracked points of previous $c_1$ and $c_2$, tracked features of current $c_1$)
    Initialize $c_2$ points by projecting the tracked features of current $c_1$ to $c_2$ using the rotation from stereo extrinsic
    function calcOpticalFlowPyrLK(pyramid of current $c_1$ and $c_2$, tracked features of current $c_1$, initialized $c_2$ points)    ▷ OpenCV
    end function
    Further remove outliers based on the essential matrix
    return matched points of previous $c_1$ and $c_2$, matched features of current $c_1$ and $c_2$
end function
2-Point RANSAC:
function twoPointRansac(matched points of previous $c_1$ or $c_2$, matched features of current $c_1$ or $c_2$, $\hat{\mathcal{T}}_{\text{curr}\leftarrow\text{prev}}$ of $c_1$ or $c_2$, Intrinsic of $c_1$ or $c_2$)
    return outlier-rejected points of current $c_1$ or $c_2$
end function
Addition of Newly Detected Features:
Create a mask to avoid re-detecting existing features
function fastFeatureDetector(current $c_1$ image, mask)
    return new features on current $c_1$
end function
function stereoMatching(new features on current $c_1$)
    return matched new features on current $c_2$
end function
Group all of outlier-rejected features
```

4.2 Outlier Adaptation in Filtering Back End

Even though image processing front end removes outliers by tracking, stereo matching, and 2-point RANSAC, some outlier features still survive and enter the filter as inputs. This section explains the outlier rejection procedure in filtering back end.

4.2.1 Outlier Removal in Feature Initialization

If a measurement is a new feature, our system initializes its $3D$ position with respect to the inertial frame. In feature initialization, Gaussian-Newton least squares minimization in Section 2.3.1 first estimates the depth of left c_1 camera. If either estimated depth of left or right camera is negative, then the solution of the minimization is invalid since features are always in front of both camera frames observing it. The process of removing features that has the invalid depth is referred to as outlier removal in feature initialization.

4.2.2 Outlier Detection by Chi-Squared Statistical Test

Before operating the navigation systems, we initialize the chi-squared test table with the 95% confidence level. While the systems estimate the state variable, if j-th measurement y_j is the existing feature, its residual r_j and Jacobian C_j are computed. Next, we proceed a Mahalanobis gating test [65] for residual r_j to detect remaining outliers. In fact, Mahalanobis distance [66] γ_j is a measure of the distance between residual r_j and covariance matrix $S_j = (C_j)_{k-\hat{d}} P_{j-1}^{\text{dly}} (C_j)_{k-\hat{d}}^{\text{T}} + R$

$$\gamma_j = r_j^{\text{T}} \left((C_j)_{k-\hat{d}} P_{j-1}^{\text{dly}} (C_j)_{k-\hat{d}}^{\text{T}} + R \right)^{-1} r_j. \tag{4.1}$$

In the statistic test, we compare γ_j value against a threshold given by the 95-th percentile of the χ^2 distribution with ν_j degrees of freedom. Here, ν_j is the number of observations of the j-th feature minus one. If the feature passes the test, the EKF uses residual r_j to process the measurement update.

4.2.3 Noise-Adaptive Filtering

Unlike the extended ORKF (EORKF) [64], for a practical estimation approach in V-INS, this study investigates only measurement outliers due to the following reasons. Since the measurement update is not the process performed at every time step, the outlier detection by each residual value cannot directly detect the outliers of IMU measurements. Furthermore, in the sequential measurement update, multiple residuals are computed to update at one IMU time stamp. In other words, since only rare observations among feature measurements from one image are corrupted by the remaining outliers, hypothesizing that the outliers comes from the IMU may be faulty. Hence, in the scope of this thesis, we handle only measurement outliers.

Student's t*-Distribution*

Despite the true system with outliers, the classical EKF assumes that each model in the filter is corrupted with additive white Gaussian noise. The levels of the noise are assumed to be constant and encoded by sensor covariance matrices Q and R (i.e., $\eta_k \sim \mathcal{N}(0,\, Q)$, $(\zeta_j)_k \sim \mathcal{N}(0,\, R)$). However, since outliers arise in the realistic system, now we do not restrict noise at either a constant or Gaussian level. Instead, their levels vary over time, or noise have heavier tails than the normal distribution as follows:

$$\zeta_j\big|_k \sim \text{ST}(0,\, \tilde{R}_j,\, \nu_j), \quad \text{where} \quad \tilde{R}_j \sim \mathcal{W}^{-1}\left(\,\nu_j \Lambda_j,\, \nu_j\,\right), \tag{4.2}$$

where $\text{ST}(\cdot)$ denotes a Student's *t*-distribution, and $\nu_k > m - 1$ is degrees of freedom. Covariance matrix $\tilde{R}_j$ follows the inverse-Wishart distribution, denoted as $\mathcal{W}^{-1}(\cdot)$. $\Lambda_j \succ 0$ is $m \times m$ precision matrix.

In Bayesian statistics, the inverse-Wishart distribution is used as the conjugate prior for the covariance matrix of a multivariate normal distribution [55]. The probability density function (pdf) of the inverse-Wishart is

$$p(\tilde{R}_j \,|\, \nu_j, \Lambda_j) \propto |\tilde{R}_j|^{-\frac{\nu_j+m+1}{2}} \exp\left[-\frac{\nu_j}{2}\,\mathrm{tr}(\Lambda_j\,\tilde{R}_j^{-1})\right], \tag{4.3}$$

where tr$(\cdot)$ denotes the trace of a square matrix in linear algebra. Moreover, in probability and statistics, a Student's *t*-distribution is any member of a family of continuous probability distributions that arises when estimating the mean of a normally distributed population in situations where the standard deviation of the population is unknown [92]. Whereas a normal distribution describes a full population, a *t*-distribution describes samples drawn from a full population; thus, the larger the sample, the more the distribution resembles a normal distribution. Indeed, as the degree of freedom goes to infinity, the *t*-distribution approaches the standard normal distribution. In other words, when the variance of a normally distributed random variable is unknown and a conjugate prior placed over it that follows an inverse-Wishart distribution, the resulting marginal distribution of the variable follows a Student's *t*-distribution [93]. Then, the Student-*t*, a sub-exponential distribution with much heavier tails than the Gaussian, is more prone to producing outlying values that fall far from its mean.

Variational Inference

The purpose of filtering is generally to find the approximations of posterior distributions $p(x_k \,|\, y_{1:k})$, where $y_{1:k} = [y_1, y_2, \cdots, y_k]$ is the histories of sensor measurements obtained up to time k. For systems with the heavy tailed noise, we also wish to produce another inference about covariance matrices whose priors follow the inverse-Wishart distribution. Hence, our goal in this section is to find both approximations for posterior distribution $p(x_{1:k}, \tilde{R}_{1:k} \big| y_{1:k})$ and model evidence $p(y_{1:k})$. Compared to sampling methods, the variational Bayesian method performs approximate posterior inference at low computational cost for a wide range of models [55, 93]. In the method, we decompose log marginal probability

$$\ln p(y_{1:k}) = \mathrm{KL}\,[\,q\,\|\,p\,] + \mathcal{L}[\,q\,], \tag{4.4}$$

where

$$\mathrm{KL}\,[\,q\,\|\,p\,] = \int q(x_{1:k}, \tilde{R}_{1:k}) \ln \frac{q(x_{1:k}, \tilde{R}_{1:k})}{p(x_{1:k}, \tilde{R}_{1:k} \,|\, y_{1:k})} \tag{4.5}$$

$$\mathcal{L}[\,q\,] = \int q(x_{1:k}, \tilde{R}_{1:k}) \ln \frac{p(x_{1:k}, \tilde{R}_{1:k},\, y_{1:k})}{q(x_{1:k}, \tilde{R}_{1:k})}. \tag{4.6}$$

p is the true distribution that is intractable for non-Gaussian noise models, and q is a tractable approximate distribution.

In probability theory, a measure of the difference between two probability distributions p and q is the Kullback-Leibler divergence, denoted as KL$[\cdot]$. If we allow any possible choice for q such as the Gaussian distribution, then lower bound $\mathcal{L}[q]$ is maximum when the KL divergence vanishes; that is, $q(x_{1:k}, \tilde{R}_{1:k}) = p(x_{1:k}, \tilde{R}_{1:k} \,|\, y_{1:k})$. To minimize the KL divergence, we seek the member of a restricted family of $q(x_{1:k}, \tilde{R}_{1:k})$. Indeed, maximizing $\mathcal{L}[q]$ is equivalent to minimizing another new KL divergence [93], and thus the minimum occurs when factorized distributions $q(x_{1:k}, \tilde{R}_{1:k}) = q(x_{1:k})\, q(\tilde{R}_{1:k})$ and the following Equations (4.7) – (4.8) hold.

$$\ln\, q(x_{1:k}) = \ln\, p(x_1) + \sum_{t=2}^{k} \mathbb{E}_{q(\tilde{R}_{1:t})}[\ln\, p(x_t \,|\, x_{t-1})] + \sum_{t=1}^{k} \mathbb{E}_{q(\tilde{R}_{1:t})}[\ln\, p(y_t \,|\, x_t, \tilde{R}_t)] + \cdots \tag{4.7}$$

$$\ln\, q(\tilde{R}_k) = \mathbb{E}_{q(x_{1:k})}[\,\ln\, p(y_k \,|\, x_k,\, \tilde{R}_k)\,] + \ln\, p(\tilde{R}_k) + \cdots \tag{4.8}$$

where $\mathbb{E}_{q(\cdot)}$ represents the expectation with respect to $q(\cdot)$. With assuming that initial state x_1 is Gaussian, the measurement update with varying noise covariance $\mathbb{E}[\,\tilde{R}_t^{-1}\,] = \mathit{\Lambda}_t^{-1}$, which closely resemble the EKF updates, solve Equation (4.7). Algorithm 3 describes the details of the updates.

Now let us assume that the true priors are IID noise models as the case in this study; that is, $p(\tilde{R}_k)$ follows $\mathcal{W}^{-1}(\nu R,\, \nu)$ distribution. Then second term $\ln p(\tilde{R}_k)$ in the right-hand side of Equation (4.8) is computed using the pdf of the inverse-Wishart distribution in Equation (4.3) with its prior noise model.

$$\ln p(\tilde{R}_k) = -\frac{\nu + m + 1}{2} \ln |\tilde{R}_k| - \frac{\nu}{2}\, \text{tr}(R\, \tilde{R}_k^{-1}). \tag{4.9}$$

Since the term is conjugate prior for Equation (4.2), the approximations of $q(\tilde{R}_k)$ have same mathematical forms as priors; that is, $q(\tilde{R}_k)$ also follows $\mathcal{W}^{-1}(\tilde{\nu}_k\, \mathit{\Lambda}_k,\, \tilde{\nu}_k)$ distribution.

$$\ln q(\tilde{R}_k) = -\frac{\tilde{\nu}_k + m + 1}{2} \ln |\tilde{R}_k| - \frac{\tilde{\nu}_k}{2}\, \text{tr}(\mathit{\Lambda}_k\, \tilde{R}_k^{-1}). \tag{4.10}$$

Since $y_t \,\Big|\{x_t, \tilde{R}_t\} \sim \mathcal{N}(h(x_t), \tilde{R}_t)$,

$$\mathbb{E}\left[\, \ln p(y_k \,|\, x_k, \tilde{R}_k) \right] = -\frac{1}{2} \ln |\tilde{R}_k| - \frac{1}{2}\, \text{tr}\left(\mathbb{E}[\zeta_k\, \zeta_k^{\mathrm{T}}]\, \tilde{R}_k^{-1} \right). \tag{4.11}$$

From Equations (4.8) – (4.11), to handle measurement outliers, similar to Agamennoni et al. [54, 53]'s derivation, we derive how to compute precision matrix $\mathit{\Lambda}_k$ of approximate distribution $q(\tilde{R}_k)$ of $\tilde{R}_k$ as follows:

$$\begin{aligned} \tilde{\nu}_k = 1 + \nu, \quad \tilde{\nu}_k\, \mathit{\Lambda}_k &= \mathbb{E}[\zeta_k\, \zeta_k^{\mathrm{T}}] + \nu R \\ \Rightarrow \mathit{\Lambda}_k &= \frac{\nu R + \mathbb{E}[\zeta_k\, \zeta_k^{\mathrm{T}}]}{\nu + 1}, \end{aligned} \tag{4.12}$$

where each feature from one image is independent and

$$\begin{aligned} (\zeta_j)_k &= (y_j)_k - h(\, x(t_{\text{img}})\,) \\ &= (y_j)_k - h\left(\, (\hat{x}_{k-\hat{d}})_j + e_j \,\right) \\ &\approx (y_j)_k - h\left(\, (\hat{x}_{k-\hat{d}})_j \,\right) - C_j\, e_j. \end{aligned}$$

Next, in Equation (4.12),

$$\begin{aligned}\mathbb{E}\left[\,(\zeta_j)_k\,(\zeta_j)_k^{\mathrm{T}}\,\right] &= \left[(y_j)_k - h\left(\,(\hat{x}_{k-\hat{d}})_j\,\right)\right]\left[(y_j)_k - h\left(\,(\hat{x}_{k-\hat{d}})_j\,\right)\right]^{\mathrm{T}}\dots\\ &\quad + C_j\,\mathbb{E}[e_j\,e_j^{\mathrm{T}}]\,C_j^{\mathrm{T}}\\ &= r_j\,r_j^{\mathrm{T}} + C_j\,(P^{\,\mathrm{dly}})_j\,C_j^{\mathrm{T}},\end{aligned} \tag{4.13}$$

where estimation error $e_j = x(t_{\mathrm{img}}) - (\hat{x}_{k-\hat{d}})_j$ and the Jacobian $C_j = \frac{\partial h_j}{\partial x}\big|_{(\hat{x}_{k-\hat{d}})_j}$. In the sequential measurement update, $(\hat{x}_{k-\hat{d}})_j$ and $(P^{\,\mathrm{dly}})_j$ are corrected by Kalman gain K_j^{dly} that is a function of $(\Lambda_j)_k$, so these update steps are coupled. Hence no a closed-form solution exists, and we can only solve iteratively. The purpose of the iteration seems to be similar to that of the online learning of unknown variances of each noise [45]. In addition, similar to Agamennoni et al.'s interpretation [54], the convergence and optimality of the derived update steps for outliers are guaranteed since the variational lower bound is convex with respect to $(\hat{x}_{k-\hat{d}})_j$, $(P^{\,\mathrm{dly}})_j$, and $(\Lambda_j)_k$. In particular, as the j-th feature is observed countless times (i.e., $\nu_j \rightarrow \infty$), Λ_j converges to R in the limit of an infinitely precise noise distribution, so the iterative update steps reduce to the standard sequential measurement update of the EKF.

If true state $x(t_{\mathrm{img}})$ is significantly different from its estimate $(\hat{x}_{k-\hat{d}})_j$, then statistics $\mathbb{E}\left[\,(\zeta_j)_k\,(\zeta_j)_k^{\mathrm{T}}\,\right]$ dominates, and $(\Lambda_j)_k$ becomes much larger than R. This ζ_j is regarded as a measurement outlier at time k. Since Kalman gain K_j^{dly} is a function of the inverse of precision matrix $(\Lambda_j)_k$, the larger $(\Lambda_j)_k$ values, the smaller the Kalman gain. Therefore, to deal with situations where measurement outliers occur, the iteration for Equations (4.12) and (4.13) corrects the state estimates and its covariance with low weights.

4.3 Implementation

4.3.1 Marginalization of Feature States

If measurement outliers often occur, a few number of sequential updates in the EKF are proceed to correct the state estimates. Without enough number of the measurement updates, the EKF is not robust. Hence, noise-adaptive filtering introduced in Section 4.2.3 performs modified measurement update even when a residual is detected as an outlier. Indeed, to save computation resource, this study operates the noise-adaptive filtering for only features detected frequently outliers. For implementation, we count how many numbers features augmented in state variables are detected as outliers. Once updating feature outliers by the noise-adaptive filtering approach, we prune the used feature states from the state vector. In addition, similar to mention in Section 2.3.1, to maintain a certain size of the state vector, after the feature initialization, we marginalize the features with the least number of observations among tracked features.

4.3.2 Summarized Algorithm

Algorithm 3 illustrates the pseudo-code of the overall process of the noise-adaptive filtering approach for V-INS. From Figure 3.5, the blue boxes and circle in the figure are extended for the noise-adaptive filter presented in this Chapter 4, shown in Figure 4.2.

4.4 Flight Datasets Test Results

To examine the influence of outliers and validate the reliability and robustness of the proposed noise-adaptive approach for systems with outliers, we test the EuroC datasets. To articulate the significance of outliers, we select two datasets of bright scene, called ”EuRoC V1 Easy,” and motion blur, called ”EuRoC V1 Difficult.” Since the images in the difficult dataset are dark scene or motion blur, we hypothesize that outliers occur more frequently in the difficult dataset.

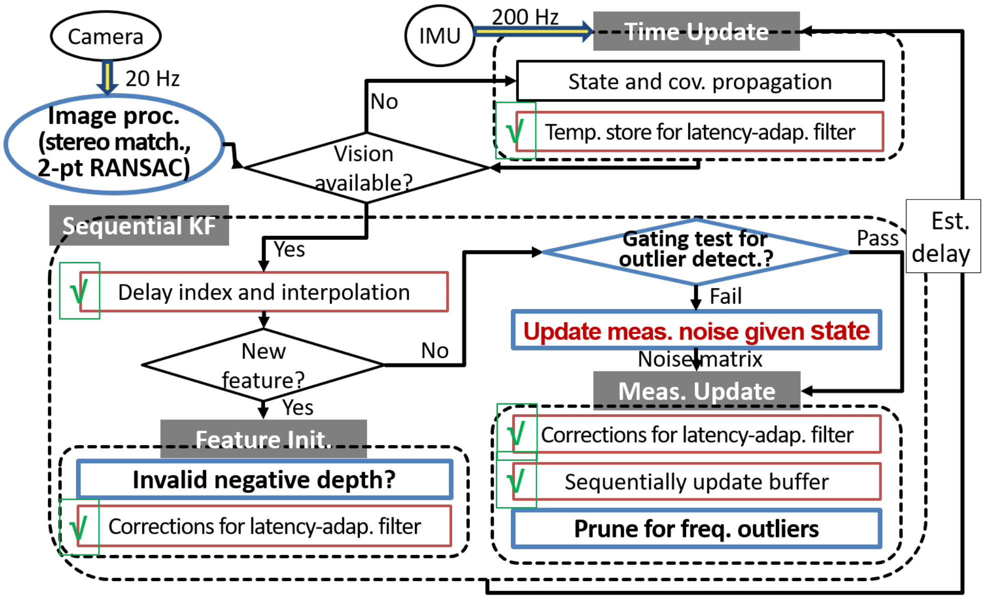

Camera
20 Hz
Image proc. (stereo match., 2-pt RANSAC)
IMU
200 Hz
Time Update
State and cov. propagation
√ Temp. store for latency-adap. filter
No
Vision available?
Sequential KF
Yes
√ Delay index and interpolation
New feature?
No
Yes
Feature Init.
Invalid negative depth?
√ Corrections for latency-adap. filter
Gating test for outlier detect.?
Pass
Est. delay
Fail
Update meas. noise given state
Noise matrix
Meas. Update
√ Corrections for latency-adap. filter
√ Sequentially update buffer
Prune for freq. outliers

Figure 4.2: A Flow Chart of the Overall Process of the Noise-Adaptive Filtering

By similar analysis in Section 3.6, absolute trajectory error as evaluation error metric explained in Appendix B produces multiple comparison plots. Figure 4.3 illustrates the top down view of the estimated flight trajectory of the difficult dataset. Figures 4.4 depict estimated x,y,z position and their each estimation error. All estimation errors are bounded within each standard deviation σ envelopes, so the proposed approach is reliable vision-aided inertial navigation under even poor illumination environment. We should expect significant time correlation in error plots and a generally growing error covariance for vision-aided inertial navigation problems. Conceptually, position error gets “locked in” and to the extent new features are being mapped the position error will tend to grow with the length of the trajectory. Similar to the analysis presented Section 3.6, the adaptive filter is a well-tuned estimator since the performance of doing runs with multipliers on the R matrix used in the filter is worse for all of those, shown in Table 4.1. That is, the fact that using the multipliers reveals larger RMS estimation errors indicates that our filter is well-tuned.

Figure 4.5 shows the advantages of addition of outlier adaptation in the noise-adaptive

Algorithm 3 The Noise-Adaptive Filtering

```
Require: $\hat{x}_0^+, P_0^+, Q, R,\ \chi^2$
for $k = 1 : T$ do
    if new IMU packet arrival then
Time Update:
        State prediction
    end if
    if new image capture then
        Image processing front-end in different thread
        Stereo matching between current images of left camera $c_1$ and right camera $c_2$
        RANSAC between previous and current images of camera $c_1$
        RANSAC between previous and current images of camera $c_2$
    end if
```

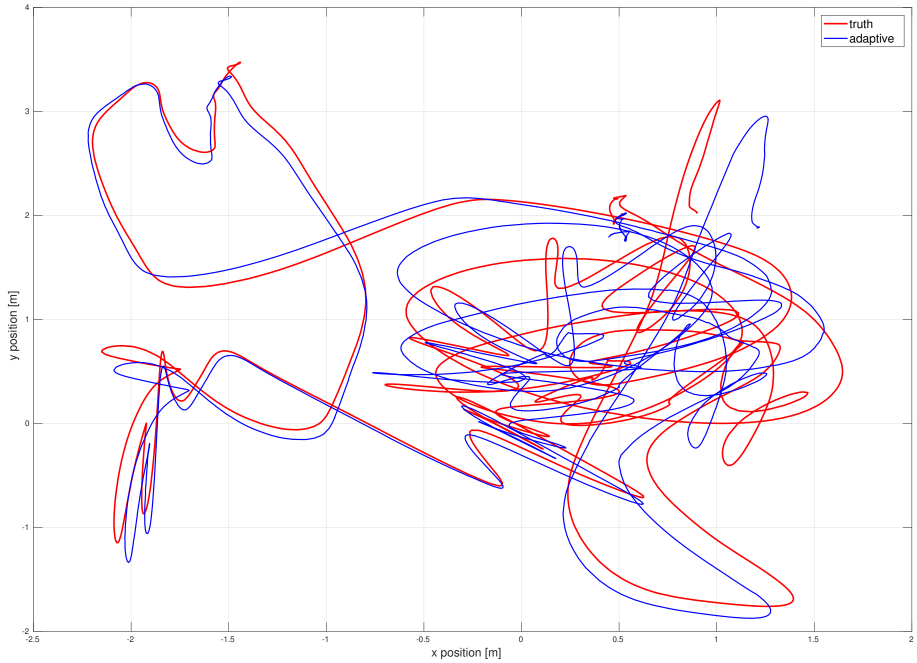
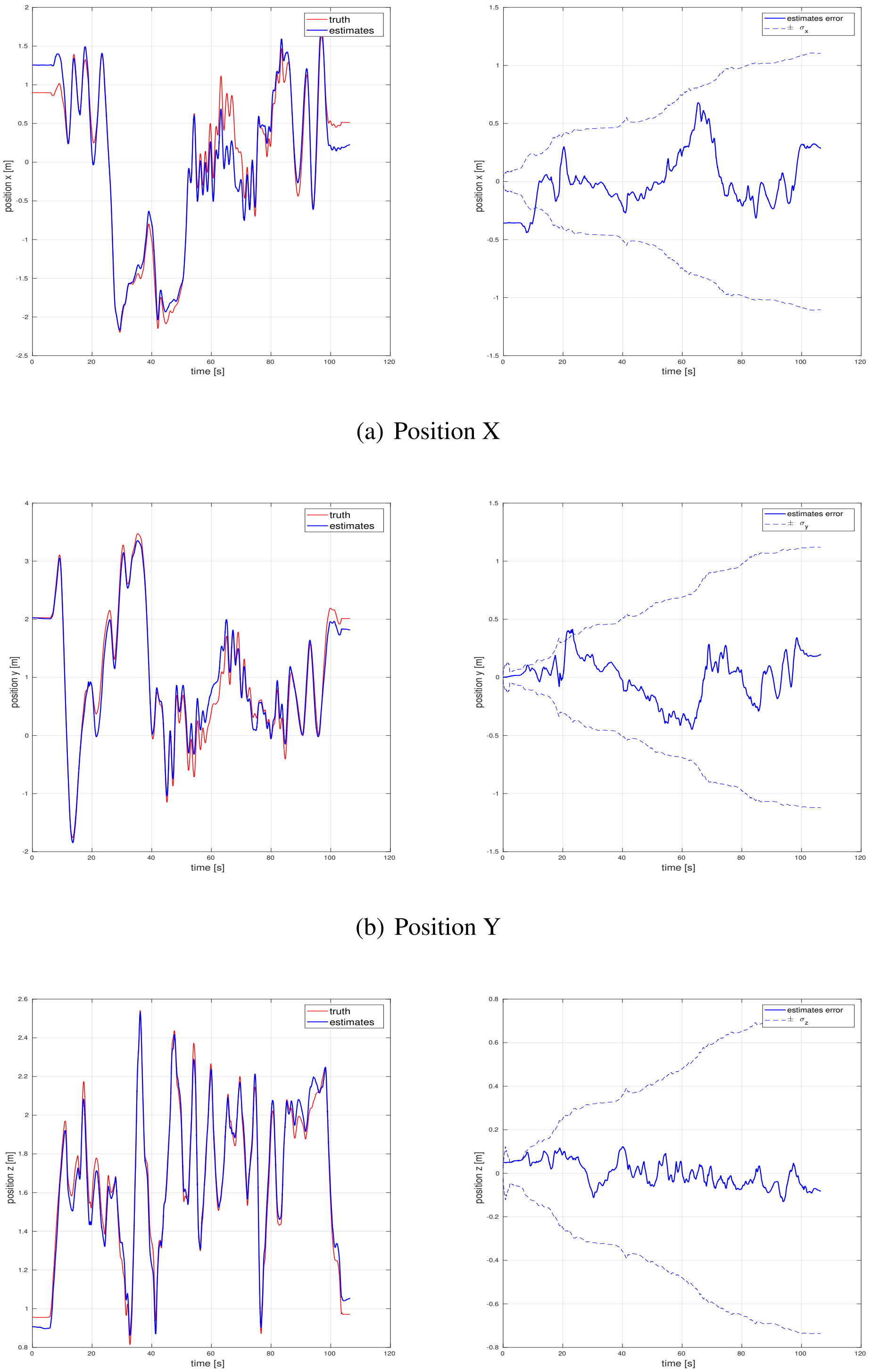


Figure 4.3: Top Down View of Flight Trajectory of the EuRoC V1 Difficult Dataset by the Noise-Adaptive Filer

```
    if new vision data packet arrival then
        for $j = 1 : \sharp$ of observed features do
            if new feature then
Feature Initialization:
                If any depth of $c_1$ or $c_2$ is negative, $j$-th feature is outlier
            else                                                              ▷ tracked feature
Outlier Gating Test:
                $r_j = y_j - h_j\left(\left(\hat{x}_{k-\hat{d}}\right)_{j-1}\right)$
                $S_j = (C_j)_{k-\hat{d}}\, P^{\text{dly}}_{j-1}\, (C_j)^{\mathrm{T}}_{k-\hat{d}} + R$        ▷ $(C_j)_{k-\hat{d}} = \frac{\partial h_j}{\partial x}\Big|_{\left(\hat{x}_{k-\hat{d}}\right)_{j-1}}$
                $\gamma = r_j^{\mathrm{T}} S_j^{-1} r_j \overset{?}{<} \chi_j^2$
                if outlier detected then
Measurement Updates:
                    $\tilde{x}_j \leftarrow \left(\hat{x}_{k-\hat{d}}\right)_{j-1}, \quad \tilde{P}_j \leftarrow P^{\text{dly}}_{j-1}$        ▷ $\left(\hat{x}_{k-\hat{d}}\right)_0 = \hat{x}_{k-\hat{d}}$, $P^{\text{dly}}_0 = P^{\text{dly}}$
                    while until converged do
                        Update measurement noise given the state
                        $\tilde{r}_j = y_j - h_j(\tilde{x}_j)$
                        $\tilde{C}_j = \frac{\partial h_j}{\partial x}\Big|_{\tilde{x}_j}$
                        $W_j = \tilde{r}_j\, \tilde{r}_j^{\mathrm{T}} + \tilde{C}_j\, \tilde{P}_j\, \tilde{C}_j^{\mathrm{T}}$
                        $\mathit{\Lambda}_j \leftarrow \dfrac{\nu_j\, R + W_j}{\nu_j + 1}$
                        Update the posteriori state given noise
                        $\tilde{S}_j = (C_j)_{k-\hat{d}}\, P^{\text{dly}}_{j-1}\, (C_j)^{\mathrm{T}}_{k-\hat{d}} + \mathit{\Lambda}_j$
                        $\tilde{K}_j = P^{\text{dly}}_{j-1}\, (C_j)^{\mathrm{T}}_{k-\hat{d}}\, \tilde{S}_j^{-1}$
                        $\tilde{x}_j \leftarrow \left(\hat{x}_{k-\hat{d}}\right)_{j-1} + \tilde{K}_j\, r_j$
                        $\tilde{P}_j \leftarrow P^{\text{dly}}_{j-1} - \tilde{K}_j\, (C_j)_{k-\hat{d}}\, P^{\text{dly}}_{j-1}$
                    end while
                    $\left(\hat{x}_{k-\hat{d}}\right)_j = \tilde{x}_j, \quad P^{\text{dly}}_j = \tilde{P}_j$
                else                                                          ▷ Chapter 3
                    $K^{\text{dly}}_j = P^{\text{dly}}_{j-1}\, (C_j)^{\mathrm{T}}_{k-\hat{d}}\, S_j^{-1}$
                    $\left(\hat{x}_{k-\hat{d}}\right)_j = \left(\hat{x}_{k-\hat{d}}\right)_{j-1} + K^{\text{dly}}_j\, r_j$
                    $P^{\text{dly}}_j = P^{\text{dly}}_{j-1} - K^{\text{dly}}_j\, (C_j)_{k-\hat{d}}\, P^{\text{dly}}_{j-1}$
                end if
                Correct using cross covariance                                ▷ Section 3.2.2
            end if
        end for
    end if
```



(a) Position X

(b) Position Y

(c) Position Z

Figure 4.4: Position and Estimation Error of the EuRoC V1 Difficult Dataset by the Noise-Adaptive Filter

Table 4.1: Indication that the Noise-Adaptive Filter is Well-Tuned for EuRoC V1 Difficult Dataset

Multiplier on R	/10	/3	1	x3	x10
RMS error [m]	0.9240	0.3801	0.1700	0.5153	0.5610

filtering by comparing with a baseline, the latency-adaptive filtering. Since Table 3.2 already shows that the latency-adaptive filtering is the promising combination, we choose the method as a baseline here. The baseline rejects outliers whenever chi-squared test fails, and although the iteration in the noise-adaptive filtering might increase computational resource, it significantly improves the accuracy of estimation. Fortunately, the iteration in Algorithm 3 rapidly converges to the optimal noise covariance by twice or three times iterations. For sensitivity analaysis, RMS position errors resulting from four filters—the baseline correction, the baseline correction plus outlier adaptive filtering, the latency-adaptive filtering, and a combination of all proposed adaptive approaches—are compiled in Table 4.2. Motion blur datasets are more sensitive to outliers since the improvement is larger when applied to those datasets. Thus, depending on computation margin and cost,

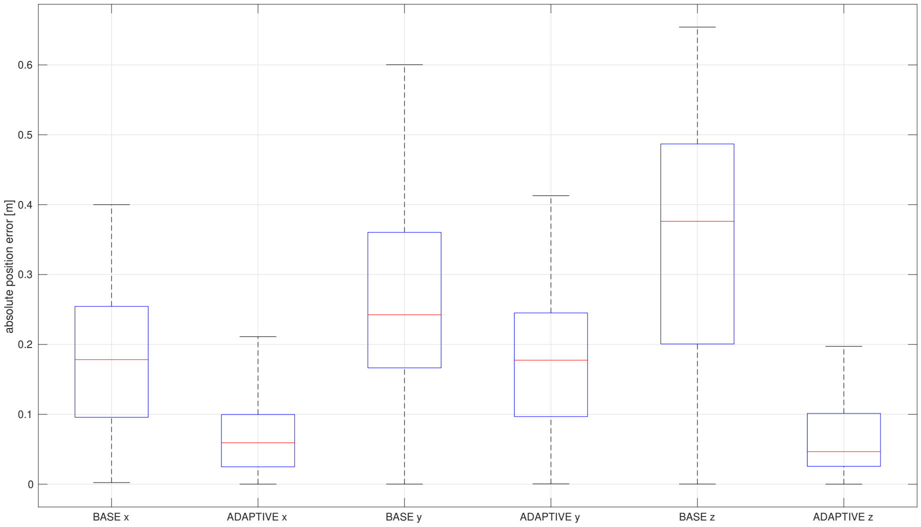


Figure 4.5: Box Plot of Absolute Estimation Error of Position of of the EuRoC V1 Difficult Dataset by the Noise-Adaptive Filer

Table 4.2: Sensitivity Analysis in RMS Position Error [m] of Noise-Adaptive Filtering

Dataset	EuRoC V1 Easy Slow Motion 0.41 m/s, 16.0 deg/s **Bright Scene**		EuRoC V1 Difficult Fast Motion 0.75 m/s, 35.5 deg/s **Motion Blur**	
Method	Latency-Adaptive OFF	Latency-Adaptive ON	Latency-Adaptive OFF	Latency-Adaptive ON
Baseline	0.2558	0.1461	0.3656	0.2663
Noise-Adaptive	0.2237	**0.1427**	0.2264	**0.1700**

we can select adequate mode.

Although a number of researchers have investigated visual inertial odometry of the EuRoC datasets, few of them thoroughly focus on measurements with outliers. Table 3.3 reveals that the proposed estimator, the noise-adaptive filter, outperforms other impressive methods, called ”S-MSCKF” and ”SVO+MSF” in which stereo is available. Since SVO+MSF is loosely coupled, its algorithm actually diverges.

Table 4.3: Comparison with Other Methods in RMS Position Error [m] of Noise-Adaptive Filtering

Dataset Method	EuRoC V1 Easy **Bright Scene**	EuRoC V1 Difficult **Motion Blur**
Noise-Adaptive Filter	0.1472	0.1700
SVO+MSF (loosely coupled)	0.40	×
S-MSCKF (stereo-filter)	0.34	0.67

CHAPTER 5
CONCLUSION

5.1 Contributions

This thesis has presented two adaptive filtering for V-INS and evaluated their performances with flight datasets testing. Two unknown parameters—sensor-related delays and outliers—arise in various realistic conditions, so compensating for uncertainties of the parameters improved accuracy and robustness of V-INS.

In particular, the following contributions were made:

i) *Development of a practical EKF-based V-INS accounting for vehicle-feature correlations. Development of tightly coupled visual inertial odometry (VIO) for autonomous flight of UAVs.* EKF-based V-INS is capable of solving more broad scopes of navigation problems than the recent state-of-the-art VIO algorithms created for solving the only IMU and vision fusion problem. Correlations between features and vehicle state were fully considered which improves the consistency of the filter.

ii) *Development of a reliable and accurate filtering formulation for measurements with unknown time delays.* We define time delays of vision data measurements in V-INS. For compensating delayed measurements and estimating unknown delay values, this thesis presented latency-adaptive filtering that includes state augmentation, interpolation, and residual, Jacobian, covariance corrections. The optimality of the three corrections and the observability of the state augmentation were validated and the resulting algorithm is identical to the standard EKF.

iii) *Development of a robust and adaptive state estimation framework for V-INS under frequent outliers occurrence.* We utilize adequate outlier removal techniques in image processing front end. For estimating the states of V-INS with measurement outliers, this document presented a novel implementation of the outlier robust EKF to VIO, for which we derive iterative update steps for computing the precision noise matrices of vision outliers when the Mahalanobis gating test detects remaining outliers in filtering front end.

iv) *Test of the performance of V-INS employing the adaptive filtering algorithms in the benchmark flight datasets for comparison to other state-of-the-art VIO algorithms.* We used realistic and widely used flight datasets for comparison to other state-of-the-art VIO algorithms. In particular, to show more improvements of our method the over others' approaches, we tested in the fast motion and motion blur flight datasets.

v) *Validation of improved accuracy of V-INS employing the latency-adaptive filtering in the fast motion flight dataset. Validation of improved robustness of V-INS employing the noise-adaptive filtering in the motion blur flight dataset.*Results from flight datasets testing show that the novel navigation approach improves the accuracy and reliability of state estimation with unknown time delays and frequent outliers in V-INS. With the adaptive filtering, RMS errors of estimation were decreased. In particular, latency-adaptive filtering improved reliability of estimation for the fast motion datasets. Moreover, the noise-adaptive filtering accelerates the robustness of estimation for motion blur datasets.

Furthermore, the overall approach in this document can be easily employed in other filter-based V-INS frameworks and suitable to monocular VIO although this study used a stereo camera to showcase the methods.

5.2 Future Work

i) *Investigation of color noise in V-INS.* A required assumption of the Kalman filter is the whiteness of measurement noise. As an illustration, during sampling and transmission in image processing, colored noise that may originate from a multiplicity of sources often degrades the quality of images [94]. The vibrational effects of camera sensors might also produce colored measurement noise [95]. Thus, we can test whiteness of noise by computing an autocorrelation matrix in offline. If the residuals of vision data are correlated with themselves at different timestamp, then colored measurement noise occurs in V-INS. Since modeling noise without additional prior knowledge of the noise statistics is typically difficult, the machine-learning techniques-based state estimator for colored noise [96, 97] may handle the unknown correlations in V-INS.

ii) *Extension of the noise-adaptive filtering for process outliers.* Since an IMU is also a sensor, it could generate outliers in V-INS. With accounting for the process outliers, the accuracy and robustness of the estimator would be improved. If we distinguish process outliers from IMU sensors with measurement outliers from vision data, the extended outlier robust EKF [64] may be an impressive approach for this case.

iii) *Evaluation with other flight datasets or by real-time flight tests.* Although the reliability and robustness of this thesis were validated by testing benchmark flight datasets, validating with other flight datasets would be beneficial to prove robustness. Moreover, UAVs stacked the navigation algorithms in this study can be operated with a controller in the loop. The use of a controller in the loop is more important validation criteria due to the potential for controller-navigation coupling.

The research goals presented here will significantly advance state-of-the-art state estimation in vision-aided inertial navigation for UAVs.

Appendices

APPENDIX A

EXPERIMENTAL EQUIPMENT AND ENVIRONMENTS

Burri et al. [84] provide benchmark datasets of UAV flying, and Table A.1 illustrates the sensor specifications of the datasets. They obtain the noise model parameters from the IMU at rest and provide them; that is, σ_a, σ_ω, σ_{b_a}, and σ_{b_ω} are known. The intrinsic and extrinsic parameters of both cameras are also given; that is, f_u, f_v, $\mathcal{T}_{c/b}$, and ${}^{b}p_{c/b}$ are known. The visual-inertial sensor unit is calibrated with *Kalibr* [98] prior to dataset collection. Furthermore, IMU and cameras are hardware time-synchronized such that the middle of the exposure aligned with the IMU measurements. The visual-inertial sensor employs an automatic exposure control that is independent for both cameras. This results in different shutter times and in turn in different image brightnesses. Since the mid-exposure times of both cameras are temporally aligned, synchronization is not affected by different shutter times.

Table A.1: Sensors of EuRoC Datasets

Sensor	Rate	Characteristics
Cameras	2 × 20 Hz	Global Shutter
IMU	200 Hz	Instrumentally Calibrated

Visual and inertial data is logged and timestamped on-board the MAV, while ground truth is logged on the base station. The accuracy of the synchronization between the ground truth and the sensor data is limited by the fact that both sources are recorded on different machines and that the timestamps of the devices are unavailable for the ground-truth system. A maximum likelihood (ML) estimator [46] aligns the data temporally and calibrates the position of the ground-truth coordinate with respect to the body sensor unit. In fact, the ML estimator synchronizes the time-varying temporal offset between the ground-truth and the sensor system. Additionally, it determines the unknown transform between the ground-truth reference frame and the body frame. To obtain the full ML solution, they employ a batch estimator in an offline procedure. Finally, as ground truth, they provide the ML solutions instead of raw data.

Table A.2: A Comparative description of EuRoC Datasets

Name	Scene	Motion	Average Velocity	Average Angular Velocity
V1_01_easy	Bright	**Slow**	0.41 m/s	16.0 deg/s
V1_02_medium	Bright	**Fast**	0.91 m/s	32.1 deg/s
V1_01_easy	**Bright**	Slow	0.41 m/s	16.0 deg/s
V1_03_difficult	**Motion Blur**	Fast	0.75 m/s	35.5 deg/s

APPENDIX B
EVALUATION ERROR METRIC

Sturm et al. [86] provide a set of tools that can be used to pre-process the datasets and to evaluate the tracking results. To validate estimation results, we need to evaluate the errors in the estimated trajectory by comparing it with the ground-truth. Among various error metrics, two prominent methods are the absolute trajectory error (ATE) and the relative pose error (RPE). In this thesis, to evaluate the overall performance of V-INS employing the adaptive filtering, the ATE measure is selected.

B.1 Absolute Trajectory Error (ATE)

The absolute trajectory error directly measures the difference between points of the true and the estimated trajectory. As a pre-processing step, we associate the estimated poses with ground truth poses using the timestamps. Based on this association, we align the true and the estimated trajectory using the Horn et al. [99]'s closed-form method based on singular value decomposition. Finally, we compute the differences between each pair of poses, and output the mean, median, and standard deviation of these differences.

APPENDIX C

STOCHASTIC CLONING (OR THE SCHMIDT-KALMAN FILTER)

For shorthand expressions in this Appendix, we denote state x and covariance P without the augmented state by the feature initialization. First, we prove Equations (3.22) – (3.25). During the delay period, cross-covariance term $P^{\,\text{crs}}$ is propagated from time $k - \hat{d}$ to time $k - s$

$$\begin{bmatrix} P_{k-\hat{d}} & {P^{\,\text{crs}}_{(k-s)\,|\,(k-\hat{d}\,)}}^{\text{T}} \\ P^{\,\text{crs}}_{(k-s)\,|\,(k-\hat{d}\,)} & P_{k-s} \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ 0 & \Phi_{k-\hat{d}} \end{bmatrix} \begin{bmatrix} P_{k-\hat{d}} & P_{k-\hat{d}} \\ P_{k-\hat{d}} & P_{k-\hat{d}} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \Phi_{k-\hat{d}} \end{bmatrix}^{\text{T}} + \begin{bmatrix} 0 & 0 \\ 0 & Q\,\Delta t_{\text{IMU}} \end{bmatrix}$$

$$= \begin{bmatrix} P_{k-\hat{d}} & P_{k-\hat{d}}\,\Phi^{\text{T}}_{k-\hat{d}} \\ \Phi_{k-\hat{d}}\,P_{k-\hat{d}} & \Phi_{k-\hat{d}}\,P_{k-\hat{d}}\,\Phi^{\text{T}}_{k-\hat{d}} + Q\,\Delta t_{\text{IMU}} \end{bmatrix},$$

where $P_{k-\hat{d}} \approx P^{\,\text{dly}}$ and Φ denotes the state transition matrix. After s time steps, at time k, the final cross-covariance term computed during the delay period is

$$\begin{bmatrix} P_{k-\hat{d}} & {P^{\,\text{crs}}_{k\,|\,(k-\hat{d})}}^{\text{T}} \\ P^{\,\text{crs}}_{k\,|\,(k-\hat{d})} & P^{-}_{k} \end{bmatrix} = \begin{bmatrix} P_{k-\hat{d}} & P_{k-\hat{d}} \left(\prod_{i=k-\hat{d}}^{k-1} \Phi_i^{\text{T}}\right) \\ \left(\prod_{i=k-1}^{k-\hat{d}} \Phi_i\right) P_{k-\hat{d}} & P^{-}_{k} \end{bmatrix}. \qquad \text{(C.1)}$$

Next, we prove Equations (3.16) – (3.21). The modified Kalman gain is computed as follows:

$$\begin{aligned}\begin{bmatrix} K^s \\ K^{\text{crs}} \end{bmatrix} &= \begin{bmatrix} P_{k-\hat{d}} & {P^{\text{crs}}_{k\,|\,(k-\hat{d})}}^{\text{T}} \\ P^{\text{crs}}_{k\,|\,(k-\hat{d})} & P^-_k \end{bmatrix} \begin{bmatrix} C_{k-\hat{d}} & 0 \end{bmatrix}^{\text{T}} (\, C_{k-\hat{d}} \, P_{k-\hat{d}} \, C^{\text{T}}_{k-\hat{d}} + R \,)^{-1} \\ &= \begin{bmatrix} * \\ P^{\text{crs}}_{k\,|\,(k-\hat{d})} \, C^{\text{T}}_{k-\hat{d}} \, (\, C_{k-\hat{d}} \, P_{k-\hat{d}} \, C^{\text{T}}_{k-\hat{d}} + R \,)^{-1} \end{bmatrix}, \end{aligned} \tag{C.2}$$

where K^s denotes the stationary Kalman gain and

$$C_{k-\hat{d}} \, P_{k-\hat{d}} \, C^{\text{T}}_{k-\hat{d}} = \begin{bmatrix} C_{k-\hat{d}} & 0 \end{bmatrix} \begin{bmatrix} P_{k-\hat{d}} & {P^{\text{crs}}_{k\,|\,(k-\hat{d})}}^{\text{T}} \\ P^{\text{crs}}_{k\,|\,(k-\hat{d})} & P^-_k \end{bmatrix} \begin{bmatrix} C_{k-\hat{d}} & 0 \end{bmatrix}^{\text{T}}.$$

The update of covariance matrix using the cross covariance term is

$$\begin{aligned}\begin{bmatrix} * & * \\ * & P^+_k \end{bmatrix} &= \begin{bmatrix} P_{k-\hat{d}} & {P^{\text{crs}}_{k\,|\,(k-\hat{d})}}^{\text{T}} \\ P^{\text{crs}}_{k\,|\,(k-\hat{d})} & P^-_k \end{bmatrix} - \begin{bmatrix} K^s \\ K^{\text{crs}} \end{bmatrix} \begin{bmatrix} C_{k-\hat{d}} & 0 \end{bmatrix} \begin{bmatrix} P_{k-\hat{d}} & {P^{\text{crs}}_{k\,|\,(k-\hat{d})}}^{\text{T}} \\ P^{\text{crs}}_{k\,|\,(k-\hat{d})} & P^-_k \end{bmatrix} \\ &= \begin{bmatrix} * & * \\ * & P^-_k - K^{\text{crs}} \, C_{k-\hat{d}} \, {P^{\text{crs}}_{k\,|\,(k-\hat{d})}}^{\text{T}} \end{bmatrix}. \end{aligned} \tag{C.3}$$

We finally prove the optimality of the latency-adaptive filtering in Chapter 3. Since the standard EKF is an optimal estimator, if we prove that the latency-adaptive filter is identical to the standard EKF, then the latency-adaptive filtering approach becomes also optimal estimation. Let us recall Equations (3.16) – (3.18).

$$\begin{aligned}
K^{\,\mathrm{crs}} &= P^{\,\mathrm{crs}}_{k\,|\,(k-\hat{d})}\, C^{\mathrm{T}}_{k-\hat{d}}\,(\,C_{k-\hat{d}}\,P_{k-\hat{d}}\,C^{\mathrm{T}}_{k-\hat{d}} + R\,)^{-1} \\
&= \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}P_{k-\hat{d}}\,C^{\mathrm{T}}_{k-\hat{d}}\,(\,C_{k-\hat{d}}\,P_{k-\hat{d}}\,C^{\mathrm{T}}_{k-\hat{d}} + R\,)^{-1} \qquad \text{(C.4)} \\
\hat{x}^{+}_{k} &= \hat{x}^{-}_{k} + K^{\,\mathrm{crs}}\, r_{k-\hat{d}} \qquad \text{(C.5)} \\
P^{+}_{k} &= P^{-}_{k} - K^{\,\mathrm{crs}}\, C_{k-\hat{d}}\, {P^{\,\mathrm{crs}}_{k\,|\,(k-\hat{d})}}^{\mathrm{T}} \\
&= P^{-}_{k} - K^{\,\mathrm{crs}}\, C_{k-\hat{d}}\, P_{k-\hat{d}}\, \Phi^{\mathrm{T}}_{k-\hat{d}}\Phi^{\mathrm{T}}_{k-s}\cdots\Phi^{\mathrm{T}}_{k-1}\,, \qquad \text{(C.6)}
\end{aligned}$$

Next, we assume that delayed measurement y is available immediately without delays. In other words, for this assumed case, measurement update is first performed and then propagation steps are processed. At time $[\,k - \hat{d}]$, given $\hat{x}^{-}_{k-\hat{d}}$ and $P^{-}_{k-\hat{d}}$, the standard measurement update performs as follows:

$$\begin{aligned}
K_{k-\hat{d}} &= P^{-}_{k-\hat{d}}C^{\mathrm{T}}_{k-\hat{d}}\,(\,C_{k-\hat{d}}P^{-}_{k-\hat{d}}C^{\mathrm{T}}_{k-\hat{d}} + R\,)^{-1} \\
\hat{x'}^{+}_{k-\hat{d}} &= \hat{x}^{-}_{k-\hat{d}} + K_{k-\hat{d}}\, r_{k-\hat{d}} \\
P'^{\,+}_{k-\hat{d}} &= P^{-}_{k-\hat{d}} - K_{k-\hat{d}}C_{k-\hat{d}}P^{-}_{k-\hat{d}},
\end{aligned}$$

where $\hat{x'}^{+}_{k-\hat{d}} \neq \hat{x}_{k-\hat{d}}$ and $P'^{\,+}_{k-\hat{d}} \neq P_{k-\hat{d}}$ since $\hat{x'}^{+}_{k-\hat{d}}$ and $P'^{\,+}_{k-\hat{d}}$ are values after the corrections by the measurement update, shown in Figure C.1. That is, red lines illustrates the original processes of the latency-adaptive filtering and blue lines presents the processes of the assumed case. From time $(k - s)$ to time k during the delay period, the assumed case propagates state estimates and covariance recursively. At time $(k - s)$,

$$\begin{aligned}
\hat{x'}^{-}_{k-s} &\approx \Phi_{k-\hat{d}}\,\hat{x'}^{+}_{k-\hat{d}} \\
&= \Phi_{k-\hat{d}}\,\hat{x}^{-}_{k-\hat{d}} + \Phi_{k-\hat{d}}\,K_{k-\hat{d}}\,r_{k-\hat{d}} \\
P'^{\,-}_{k-\hat{d}+1} &= \Phi_{k-\hat{d}}\,P'^{\,+}_{k-\hat{d}}\,\Phi^{\mathrm{T}}_{k-\hat{d}} + Q\,\Delta t_{\mathrm{IMU}} \\
&= \Phi_{k-\hat{d}}\,P^{-}_{k-\hat{d}}\,\Phi^{\mathrm{T}}_{k-\hat{d}} + Q\,\Delta t_{\mathrm{IMU}} - \Phi_{k-\hat{d}}\,K_{k-\hat{d}}C_{k-\hat{d}}P^{-}_{k-\hat{d}}\,\Phi^{\mathrm{T}}_{k-\hat{d}}.
\end{aligned}$$

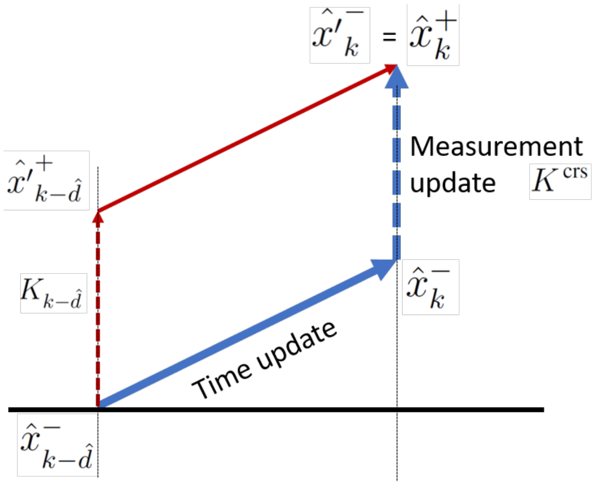

$\hat{x}'^{-}_{k}$ = $\hat{x}^{+}_{k}$
$\hat{x}'^{+}_{k-\hat{d}}$
Measurement update K^{crs}
$K_{k-\hat{d}}$
$\hat{x}^{-}_{k}$
Time update
$\hat{x}^{-}_{k-\hat{d}}$

Figure C.1: Optimality of the Latency-Adaptive Filter

Likely, at time k, only time update is processed since the measurement was already used to update.

$$\begin{aligned}
\hat{x}^{+}_{k} = \hat{x}'^{-}_{k} &\approx \Phi_{k-1}\hat{x}'^{+}_{k-1} \\
&= \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,\hat{x}^{-}_{k-\hat{d}} + \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,K_{k-\hat{d}}\,r_{k-\hat{d}} \\
&= \hat{x}^{-}_{k} + \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,K_{k-\hat{d}}\,r_{k-\hat{d}} \\
&= \hat{x}^{-}_{k} + K^{\text{crs}}\,r_{k-\hat{d}} \qquad \text{(C.7)} \\
P^{+}_{k} = P'^{-}_{k} &= \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,P^{-}_{k-\hat{d}}\,\Phi^{\text{T}}_{k-\hat{d}}\Phi^{\text{T}}_{k-s}\cdots\Phi^{\text{T}}_{k-1} \\
&\quad + Q\,\Delta t_{\text{IMU}} + \Phi_{k-1}Q\,\Delta t_{\text{IMU}}\,\Phi^{\text{T}}_{k-1} + \cdots \\
&\quad - \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,K_{k-\hat{d}}\,C_{k-\hat{d}}P^{-}_{k-\hat{d}}\,\Phi^{\text{T}}_{k-\hat{d}}\Phi^{\text{T}}_{k-s}\cdots\Phi^{\text{T}}_{k-1} \\
&= P^{-}_{k} - \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,K_{k-\hat{d}}\,C_{k-\hat{d}}P^{-}_{k-\hat{d}}\,\Phi^{\text{T}}_{k-\hat{d}}\Phi^{\text{T}}_{k-s}\cdots\Phi^{\text{T}}_{k-1} \\
&= P^{-}_{k} - K^{\text{crs}}\,C_{k-\hat{d}}P^{-}_{k-\hat{d}}\,\Phi^{\text{T}}_{k-\hat{d}}\Phi^{\text{T}}_{k-s}\cdots\Phi^{\text{T}}_{k-1}. \qquad \text{(C.8)}
\end{aligned}$$

Since Equations (C.5) and (C.6) are identical to Equations (C.7) and (C.8), respectively, the hypothesis was completely proved. In other words, the latency-adaptive filter for V-INS acts as if the delayed vision data from an image are available at the right time when the image was captured.

REFERENCES

[1] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Basic Engineering*, vol. 83, no. 3, pp. 95–108, 1961.

[3] B. D. Anderson and J. B. Moore, *Optimal filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979.

[4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[5] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *International Symposium on Aerospace/Defense, Sensing, Simulation and Controls*, vol. 3, 1997, pp. 182–193.

[6] S. F. Schmidt, "The Kalman filter: Its recognition and development for aerospace applications," *Journal of Guidance, Control, and Dynamics*, vol. 4, no. 1, pp. 4–7, 1981.

[7] D. Gaylor and E. G. Lightsey, "GPS/INS Kalman filter design for spacecraft operating in the proximity of the international space station," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003, p. 5445.

[8] M. J. Holzinger, K. T. Alfriend, C. J. Wetterer, K. K. Luu, C. Sabol, and K. Hamada, "Photometric attitude estimation for agile space objects with shape uncertainty," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 921–932, 2014.

[9] J. Le Ny, E. Feron, and M. A. Dahleh, "Scheduling continuous-time Kalman filters," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1381–1394, 2011.

[10] D. M. Hawkins, *Identification of outliers*. Springer, 1980.

[11] A. D. Wu, E. N. Johnson, M. Kaess, F. Dellaert, and G. Chowdhary, "Autonomous flight in GPS-denied environments using monocular vision and inertial sensors," *Journal of Aerospace Information Systems*, vol. 10, no. 4, pp. 172–186, 2013.

[12] G. Chowdhary, E. N. Johnson, D. Magree, A. Wu, and A. Shein, "GPS-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft," *Journal of Field Robotics*, vol. 30, no. 3, pp. 415–438, 2013.

[13] D. Magree and E. N. Johnson, “Factored extended Kalman filter for monocular vision-aided inertial navigation,” *Journal of Aerospace Information Systems*, vol. 13, no. 12, pp. 475–490, 2016.

[14] J Delmerico and D Scaramuzza, “A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[15] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, “Robust visual inertial odometry using a direct EKF-based approach,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 298–304.

[16] T. Qin, P. Li, and S. Shen, “VINS-MONO: A robust and versatile monocular visual-inertial state estimator,” *arXiv preprint arXiv:1708.03852*, 2017.

[17] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, “Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle,” *Journal of Field Robotics*, vol. 33, no. 4, pp. 431–450, 2016.

[18] M. K. Paul and S. I. Roumeliotis, “Alternating-Stereo VINS: Observability analysis and performance evaluation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4729–4737.

[19] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, “Robust stereo visual inertial odometry for fast autonomous flight,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018.

[20] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual–inertial odometry using nonlinear optimization,” *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[21] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, “Autonomous aerial navigation using monocular visual-inertial fusion,” *Journal of Field Robotics*, vol. 35, no. 1, pp. 23–51, 2018.

[22] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint Kalman filter for vision-aided inertial navigation,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 3565–3572.

[23] M. Li and A. I. Mourikis, “High-precision, consistent EKF-based visual-inertial odometry,” *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[24] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast semi-direct monocular visual odometry,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 15–22.

[25] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “SVO: Semidirect visual odometry for monocular and multicamera systems,” *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[26] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, “A robust and modular multi-sensor fusion approach applied to mav navigation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2013, pp. 3923–3929.

[27] M. K. Paul, K. Wu, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis, “A comparative analysis of tightly-coupled monocular, binocular, and stereo vins,” in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 165–172.

[28] H. L. Alexander, “State estimation for distributed systems with sensing delay,” in *Proceedings of SPIE Data Structures and Target Classification*, vol. 1470, 1991, pp. 103–111.

[29] T. D. Larsen, N. A. Andersen, O. Ravn, and N. K. Poulsen, “Incorporation of time delayed measurements in a discrete-time Kalman filter,” in *Proceedings of the 37th IEEE Conference on Decision and Control (CDC)*, vol. 4, 1998, pp. 3972–3977.

[30] R. Van Der Merwe, E. A. Wan, S. Julier, *et al.*, “Sigma-point Kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation,” in *AIAA Guidance, Navigation, and Control Conference*, 2004, pp. 16–19.

[31] R. Van Der Merwe, “Sigma-point Kalman filters for probabilistic inference in dynamic state-space models,” PhD thesis, Oregon Health & Science University, 2004.

[32] S. F. Schmidt, “Applications of state space methods to navigation problems,” *Advances in Control Systems*, vol. 3, pp. 293–340, 1966.

[33] S. I. Roumeliotis and J. W. Burdick, “Stochastic cloning: A generalized framework for processing relative state measurements,” in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2002, pp. 1788–1795.

[34] A. Gopalakrishnan, N. S. Kaisare, and S. Narasimhan, “Incorporating delayed and infrequent measurements in extended Kalman filter based nonlinear state estimation,” *Journal of Process Control*, vol. 21, no. 1, pp. 119–129, 2011.

[35] S. J. Julier and J. K. Uhlmann, “Fusion of time delayed measurements with uncertain time delays,” in *Proceedings of American Control Conference (ACC)*, IEEE, 2005, pp. 4028–4033.

[36] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, “Kalman filtering with intermittent observations,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[37] M. Choi, J. Choi, J. Park, and W. K. Chung, “State estimation with delayed measurements considering uncertainty of time delay,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 3987–3992.

[38] H. Yoon, D. C. Sternberg, and K. Cahoy, “Interpolation method for update with out-of-sequence measurements: The augmented fixed-lag smoother,” *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 11, pp. 2546–2553, 2016.

[39] K. Lee and E. N. Johnson, “Multiple-model adaptive estimation for measurements with unknown time delay,” in *AIAA Guidance, Navigation, and Control Conference*, 2017, p. 1260.

[40] J.-O. Nilsson, I. Skog, and P. Händel, “Joint state and measurement time-delay estimation of nonlinear state space systems,” in *10th IEEE International Conference on Information Sciences, Signal Processing and their Applications (ISSPA)*, 2010, pp. 324–328.

[41] M. Li and A. I. Mourikis, “Online temporal calibration for camera–imu systems: Theory and algorithms,” *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 947–964, 2014.

[42] T. Qin and S. Shen, “Online temporal calibration for monocular visual-inertial systems,” *arXiv preprint arXiv:1808.00692*, 2018.

[43] K. Lee and E. N. Johnson, “State and parameter estimation using measurements with unknown time delay,” in *IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 1402–1407.

[44] ——, “Latency compensated visual inertial odometry for autonomous flight,” *Journal of Guidance, Control, and Dynamics*, 2019, IN PROGRESS.

[45] R. K. Mehra, “On the identification of variances and adaptive kalman filtering,” *IEEE Transactions on Automatic Control*, vol. 15, no. 2, pp. 175–184, 1970.

[46] P. S. Maybeck, *Stochastic models, estimation, and control*. Academic press, 1982.

[47] R. F. Stengel, *Optimal control and estimation*. Courier Corporation, 2012.

[48] J.-A. Ting, E. Theodorou, and S. Schaal, “A Kalman filter for robust outlier detection,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 1514–1519.

[49] S. Särkkä and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 596–600, 2009.

[50] R. Piché, S. Särkkä, and J. Hartikainen, "Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate Student-*t* distribution," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2012, pp. 1–6.

[51] M. Roth, E. Özkan, and F. Gustafsson, "A Student's *t* filter for heavy tailed process and measurement noise," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 5770–5774.

[52] A. Solin and S. Särkkä, "State space methods for efficient inference in Student-*t* process regression," in *Artificial Intelligence and Statistics*, 2015, pp. 885–893.

[53] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "An outlier-robust Kalman filter," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1551–1558.

[54] ——, "Approximate inference in state-space models with heavy-tailed noise," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5024–5037, 2012.

[55] M. J. Beal, *Variational algorithms for approximate Bayesian inference*. University of London United Kingdom, 2003.

[56] M. C. Graham, J. P. How, and D. E. Gustafson, "Robust state estimation with sparse outliers," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 7, pp. 1229–1240, 2015.

[57] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[58] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[59] C. Troiani, A. Martinelli, C. Laugier, and D. Scaramuzza, "2-point-based outlier rejection for camera-imu systems with applications to micro aerial vehicles," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 5530–5536.

[60] H.-P. Kriegel, P. Kröger, and A. Zimek, "Outlier detection techniques," *Tutorial at KDD*, vol. 10, 2010.

[61] P. Angelov, "Anomaly detection based on eccentricity analysis," in *IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)*, 2014, pp. 1–8.

[62] B. S. J. Costa, C. G. Bezerra, L. A. Guedes, and P. P. Angelov, "Online fault detection based on typicality and eccentricity data analytics," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2015, pp. 1–6.

[63] C. G. Bezerra, B. S. J. Costa, L. A. Guedes, and P. P. Angelov, "A comparative study of autonomous learning outlier detection methods applied to fault detection," in *IEEE International Conference on Fuzzy Systems*, 2015, pp. 1–7.

[64] K. Lee and E. N. Johnson, "Robust state estimation and online outlier detection using eccentricity analysis," in *IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 1350–1355.

[65] G. Chang, "Robust Kalman filtering based on Mahalanobis distance as outlier judging criterion," *Journal of Geodesy*, vol. 88, no. 4, pp. 391–401, 2014.

[66] P. C. Mahalanobis, "On the generalized distance in statistics," National Institute of Science of India, 1936.

[67] K. E. Atkinson, *An introduction to numerical analysis*. John Wiley & Sons, 2008.

[68] A. Gelb, *Applied optimal estimation*. MIT press, 1974.

[69] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[70] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*. CRC press, 2011.

[71] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

[72] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[73] F. L. Markley *et al.*, "Attitude error representations for Kalman filtering," *Journal of guidance control and dynamics*, vol. 26, no. 2, pp. 311–317, 2003.

[74] A. D. Wu, "Vision-based navigation and mapping for flight in GPS-denied environments," PhD thesis, Georgia Institute of Technology, 2010.

[75] A. Bjorck, *Numerical methods for least squares problems*. SIAM, 1996, vol. 51.

[76] J. M. Montiel, J. Civera, and A. J. Davison, “Unified inverse depth parametrization for monocular SLAM,” Robotics: Science and Systems, 2006.

[77] E. Meijering, “A chronology of interpolation: From ancient astronomy to modern signal and image processing,” *Proceedings of the IEEE*, vol. 90, no. 3, pp. 319–342, 2002.

[78] K. Shoemake, “Animating rotation with quaternion curves,” in *ACM SIGGRAPH computer graphics*, ACM, vol. 19, 1985, pp. 245–254.

[79] E. B. Dam, M. Koch, and M. Lillholm, *Quaternions, interpolation and animation*. Citeseer, 1998, vol. 2.

[80] R. E. Kopp and R. J. Orford, “Linear regression applied to system identification for adaptive control systems,” *AIAA Journal*, vol. 1, no. 10, pp. 2300–2306, 1963.

[81] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre, “Nonlinear observability and observer design through state augmentation,” in *IEEE 53rd Annual Conference on Decision and Control (CDC)*, IEEE, 2014, pp. 133–138.

[82] D Del Vecchio and R. Murray, “Observability and local observer construction for unknown parameters in linearly and nonlinearly parameterized systems,” in *Proceedings of the American Control Conference (ACC)*, IEEE, vol. 6, 2003, pp. 4748–4753.

[83] A. Carrassi and S. Vannitsem, “State and parameter estimation with the extended Kalman filter: An alternative formulation of the model error dynamics,” *Quarterly Journal of the Royal Meteorological Society*, vol. 137, no. 655, pp. 435–451, 2011.

[84] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The EuRoC micro aerial vehicle datasets,” *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[85] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: An open-source robot operating system,” in *ICRA workshop on open source software*, Kobe, Japan, vol. 3, 2009, p. 5.

[86] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2012, pp. 573–580.

[87] M. Trajković and M. Hedley, “Fast corner detection,” *Image and vision computing*, vol. 16, no. 2, pp. 75–87, 1998.

[88] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *European conference on computer vision*, Springer, 2006, pp. 430–443.

[89] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2*, Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[90] B. Kitt, A. Geiger, and H. Lategahn, “Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme,” in *Intelligent Vehicles Symposium (IV)*, IEEE, 2010, pp. 486–492.

[91] *Open CV*, `https://opencv.org`.

[92] J. L. Devore, *Probability and Statistics for Engineering and the Sciences*. Cengage Learning, 2015.

[93] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[94] X. Shenshu, Z. Zhaoying, Z. Limin, X. Changsheng, and Z. Wendong, “Adaptive filtering of color noise using the Kalman filter algorithm,” in *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference (IMTC)*, vol. 2, 2000, pp. 1009–1012.

[95] A. Kumar and J. L. Crassidis, “Colored-noise Kalman filter for vibration mitigation of position/attitude estimation systems,” in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007, p. 6516.

[96] K. Lee and E. N. Johnson, “State estimation using Gaussian process regression for colored noise systems,” in *IEEE Aerospace Conference*, 2017, pp. 1–8.

[97] K. Lee, Y. Choi, and E. N. Johnson, “Kernel embedding-based state estimation for colored noise systems,” in *IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, 2017, pp. 1–8.

[98] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2013, pp. 1280–1286.

[99] B. K. Horn, H. M. Hilden, and S. Negahdaripour, “Closed-form solution of absolute orientation using orthonormal matrices,” *JOSA A*, vol. 5, no. 7, pp. 1127–1135, 1988.

VITA

Kyuman Lee received his B.S. degree in aerospace engineering from the Korea Advanced Institute of Science and Technology and two M.S. degrees in aerospace engineering and mathematics from the Georgia Institute of Technology. He is currently a Ph.D. candidate in aerospace engineering at Georgia Tech, where he is conducting research pertaining to adaptive state estimation for vision-aided inertial navigation systems of UAVs. His other research at Georgia Tech, field test-based guideline development for the integration of UAVs in GDOT operations, is funded by the Georgia Department of Transportation (GDOT). He also worked as a teaching assistant in a control system design laboratory for two years. After earning his doctorate, Kyuman plans to enter academia or a research institute as a post-doctoral scholar. He has published six conference papers and one of them was a winner of the American Institute of Aeronautics and Astronautics (AIAA) GNC best paper award. He is a member of the AIAA and the IEEE Control Systems Society.