Path Tracking Control Framework for Autonomous Driving

An Outlier Robust approach

Shan Gao

Path Tracking Control Framework for Autonomous Driving

An Outlier Robust approach

by

Shan Gao

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday March 31, 2025 at 10:00 AM.

Student number:	5711002	
Project duration:	September, 2023 – March, 2025	
Thesis committee:	Prof. dr. Riccardo Ferrari,	TU Delft, supervisor
	Dr. Wolfram Martens,	TU Delft
	Prof. dr. Barys Shyrokau,	TU Delft

This thesis is confidential and cannot be made public until December 31, 2025.

Preface

Originally, from my discussion with my supervisors, this thesis project was supposed to be something related to platooning control. However, I was really struggle with the DCSC vehicle setup. All the challenges from the setup made it quite difficult to stick to the original goal. But I enjoyed my time in the DCSC lab in the first half of 2024, and I am grateful for the support provided by my supervisors and the DCSC staff.

With the problems I had, I realized that I could instead try to do a project that addresses the challenges of the DCSC vehicle platform. After collecting data and analyzing (like identifying the causes and consequences of failures mode), it becomes a safety engineering problem. Some theory and practices from the Control field do provide good solution to the robustness and safety of the system. The project was getting a lot promising, and I was fully absorbed in trying to achieving the goal of robustifying the system, and now here we are.

I really like a sentence from Agamennoni's work, "The robustness comes from the uncertainty over the uncertainty." Despite all the foreseeing difficulties and hidden challenges, just focusing on solving the problems at hand can be an effective point to start.

Shan Gao
Delft, March 2025

Abstract

Autonomous driving technology aims to improve road safety, reduce energy consumption, and increase traffic efficiency. Developing a full-scale autonomous vehicle is costly and subject to strict regulations, making model-scale autonomous vehicle platforms more accessible alternatives for research. In recent years, an autonomous vehicle platform has been developed as a testbed for autonomous driving at the Delft Center for Systems and Control (DCSC).

However, several challenges are encountered during the development phase due to sensor malfunctions and inaccuracies in vehicle system modeling. The external vision-based localization system (Motion Capture System) generates inaccurate measurements, and the communication between the central controller and the sensors can be unstable, leading to packet loss. Additionally, there is a Radio Control (RC) input layer between the controller and the DCSC vehicles, and the dynamics of the RC inputs are unknown. These challenges have limited the implementation of general control architectures in the system.

To address these challenges, an Outlier-Robust Path-Tracking Framework is proposed in this thesis to perform path-tracking tasks under simulated sensor malfunctions and unknown vehicle dynamics. Within the framework, the motion of the vehicle is simulated using an identified DCSC vehicle model. A first-order system is employed to model the relationship between the RC commands and the vehicle's rear wheel angular velocity and front wheel steering angle. Adaptive Outlier-Robust Kalman Filter is proposed to accommodate both the observational outliers and the model mismatch between the process model and the simulation model.

The performance of the framework is evaluated using synthetic datasets. Three types of observational disturbances—Gaussian noise, observational outliers, and packet loss—are introduced to the observations. The results show that, compared to the basic Kalman filter, the proposed filter provides more precise state estimation under observational disturbances. Additionally, compared to other non-adaptive outlier-robust Kalman filters, the proposed method can adjust the process noise covariance without requiring an accurate prior for the process noise covariance. The proposed framework demonstrated promising results for the path-tracking task under disturbances. Although the framework is tailored to our platform, the approach can be adapted to other systems with noisy measurements and model discrepancy.

Contents

1 Introduction

1.1. Motivation

The development of autonomous driving technology aims to improve road safety, reduce energy consumption, and increase traffic efficiency [34]. For full-scale autonomous vehicles, the DARPA Grand Challenge in 2005 [30] initiated a test of autonomous vehicles on off-road terrain to assess their ability to complete a course without human intervention. In 2007, the DARPA Urban Challenge took place at a California Air Force Base, designed to evaluate autonomous vehicles in a controlled urban traffic simulation. By 2024, some advanced full-scale vehicles achieved Level-4 automation, meaning they are mostly autonomous even in some safety-critical conditions [9]. However, these vehicles are limited to testing in constrained areas.

Manufacturing and testing full-scale vehicles can be very costly. For researchers, a more feasible alternative is the model-scale autonomous vehicle platform. Jacopo Tani *et al.* [29] proposed a model-scale vehicle with a full set of toy-urban scenarios at a very low implementation cost. Matthew *et al.* [21] focused on bridging the gap between model-scale vehicles and real automobiles, enabling control pipelines developed on model vehicles to be directly applied to full-scale vehicles. The CPM Lab [14] provided an open-source autonomous vehicle development environment, including a standardized environment for simulation and remote field testing.

In recent years, an autonomous ground vehicle platform has been developed as a testbed for autonomous driving at the Delft Center for Systems and Control (DCSC). However, the platform faced several challenges: inaccurate data from the external vision-based localization system, packet loss, and unknown radio-control dynamics. These challenges limit the implementation of general control architectures in the system. The goal of this thesis is to propose a control framework to address these challenges in a simulation environment and provide a feasible way to facilitate further applications of the DCSC vehicle platform.

1.2. Background

One of the main challenges in the DCSC autonomous vehicle system is the modeling of the DCSC vehicle. Overall, the dynamics of ground vehicles is a well-studied problem. For a ground vehicle with front-wheel steering and rear-wheel propulsion, under nominal driving conditions (when the road is not low-friction and the vehicle is not driving at high velocity), the vehicle dynamics can be described by a simple model—the kinematic bicycle model [15, 24]. On low-friction tracks or at higher velocities, the vehicle body dynamics [13, 26] must be taken into consideration. In this more complex model, external forces, moment, and the vehicle's inertia are considered. However, for the DCSC vehicle, there is an additional Radio-Control (RC) layer between the PWM input command and the vehicle driving system. This raises our first research question:

- **How can we model the dynamics of the DCSC vehicle?**

The other main challenge in the system is sensor malfunction. Observational outliers and packet loss often occur during operation, making state estimation more difficult. Observational outliers are numerically deviated from nominal observations [10] and have a non-negligible probability of occurrence. Thus, these outliers exhibit heavy-tailed properties and cannot be modeled by a single Gaussian distribution. Researchers have modeled outliers using heavy-tailed distributions [2, 23], while others have treated outliers as

useless information and rejected them [33]. Corresponding outlier-robust Kalman filters are developed for state estimation under noisy measurements. These filters can make used of both the prior knowledge of the system and the observations to give the estimation of the system states even under the observational outliers. However, the limitation of outlier-robust Kalman filters is that, when the parameters of the filters are poorly tuned, they may reject or down-weight nominal observations containing useful information, preventing these observations from being fully utilized to correct state estimation. This leads to our second research question:

- **How can we model and detect outliers in measurement data, and are existing outlier rejection methods sufficient for our system?**

Once the above research questions are addressed, it becomes possible to design a simulated control framework that can simulate the dynamics of the DCSC RC vehicle while performing basic autonomous driving applications under noisy observations. This raises our third research question:

- **How can we implement path-tracking algorithms for a vehicle operating under nominal conditions with sensor malfunctions?**

1.3. Main Contribution

To address these challenges, an Outlier-Robust Path-Tracking Framework is proposed in this thesis to perform path-tracking tasks under simulated sensor malfunctions and unknown vehicle dynamics. Within the framework, the motion of the vehicle is simulated using an identified DCSC vehicle model. An Adaptive Outlier-Robust Kalman Filter is employed to mitigate the effects of sensor malfunctions and estimate model mismatch. An efficient geometric path-tracking algorithm [11] is implemented to validate if the simulated vehicle is able to follow the designated track under disturbances.

A first-order system is employed to model the relationship between the RC commands and the vehicle's rear wheel angular velocity and front wheel steering angle. By combining this with the vehicle's kinematic relationships and force-moment body dynamics, a RC vehicle dynamics model is proposed in the thesis to describe the motion of the DCSC vehicle. Field testing data is used to identify the parameters of the model, improving its reliability.

For the Adaptive Outlier-Robust Kalman Filter, the process model is based on the kinematic bicycle model, which introduces a discrepancy between the process model and the identified DCSC vehicle model. To mitigate this model mismatch, the filter assumes the process noise is drawn from a Gaussian distribution with a varying process noise covariance. To accommodate sensor malfunctions, the filter assumes the observation noise is drawn from a bi-Gaussian model, where one of the Gaussian distributions has a large covariance. False observations are discarded by the filter. The occurrence of outliers is modeled by a Bernoulli distribution. During the filtering process, along with the dynamic states (position and orientation) of the vehicle, the process noise covariance and the likelihood of the Bernoulli distribution are also estimated. The expectation of the Bernoulli distribution's posterior serves as an observational outlier indicator.

The performance of the framework is evaluated using a synthetic dataset. Three types of observational disturbances—Gaussian noise, observational outliers, and packet loss—are introduced to the observations. The estimation error of the proposed filter and the tracking error of the simulated vehicle with respect to the path are compared. The results show that, compared to the basic Kalman filter, the proposed filter provides more precise state estimation under observational disturbances. Additionally, compared to other non-adaptive outlier-robust Kalman filters, the proposed method can adaptively adjust the process noise covariance without requiring an accurate prior for the process noise covariance. The proposed framework demonstrated promising results for the path-tracking task under disturbances. Although the framework is tailored to our platform, the approach can be adapted to other systems with noisy measurements and model discrepancy.

1.4. Thesis Organization

The organization of the thesis is as follows:

- In Chapter 2, the research background is introduced, including the DCSC autonomous vehicle platform, the dynamics of the vehicle, and existing outlier-robust Kalman filters.

- In Chapter 3, the outlier-robust path-tracking control framework is proposed. This includes the modeling of the Radio-Control vehicle and the adaptive outlier-robust Kalman filter, which adaptively adjusts the process noise covariance.
- In Chapter 4, the parameters of the RC vehicle model are identified using the real-time data collected during the field test.
- In Chapter 5, the performance of the observers are tested on the synthetic datasets.
- In Chapter 6, the performance of the path-tracking control framework is evaluated based on different combinations of observers and path-tracking controllers.
- In Chapter 7, the conclusions of the thesis is drawn.

2

Research Background

This chapter introduces the research background. First, we present the setup of the DCSC autonomous vehicle platform and discuss the challenges in the system that remain unresolved. Next, we analyze vehicle modeling, covering both the kinematic relationships and vehicle-body dynamics. Following this, we introduce the Kalman filter and its variants, with a particular focus on outlier-robust Kalman filters. Finally, we review and introduce two geometric path-tracking methods.

2.1. DCSC Autonomous Vehicle Platform Overview

An Autonomous Vehicle Platform is a collection of hardware and software designed to serve as a testbed for evaluating autonomous vehicle applications. Recently, such a platform has been developed at the Delft Center for Systems and Control (DCSC) [32]. In its current development phase, developers can remotely control the vehicle by sending commands within the DCSC Lab's Local Area Network (LAN) via Wi-Fi, and the vehicle's pose can be monitored using an external sensor suite. However, challenges such as sensor malfunctions and the modeling of the DCSC vehicle need to be resolved.

2.1.1. System and Subsystems

The nominal operation of the DCSC Autonomous Vehicle Platform relies on the coordination of several subsystems, including the Ground Control Station (GCS), Motion Capture System (MoCap), Remote-Control System (RC), Vehicle Control Unit (VCU), and Driving System.

Ground Control Station

The Ground Control Station serves as the centralized high-level controller of the platform, functioning as the "brain." It collects runtime information from the autonomous vehicle and sends control signals to the VCU. To handle this information efficiently, it requires enough computing power. High-level tasks such as perception, planning, and tracking are implemented here. In the DCSC platform, a laptop running *Ubuntu 16.04 OS* acts as the GCS.

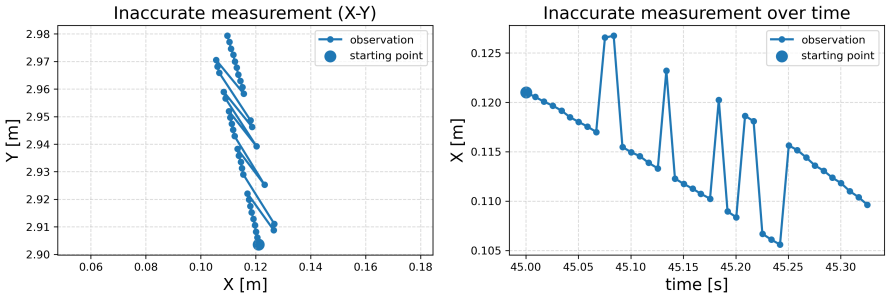
Motion Capture System

The Motion Capture System is an external sensor suite that measures the pose of objects. It consists of multiple cameras that track reflections within the experiment field from different perspectives, enabling 3D reconstruction and pose estimation. In the DCSC platform, the MoCap is an *OptiTrack* setup with eight ceiling-mounted cameras surrounding the experiment field. Three reflective markers are attached to the vehicle for tracking. The pose information is broadcast in real-time within the DCSC Lab LAN for the GCS to collect.

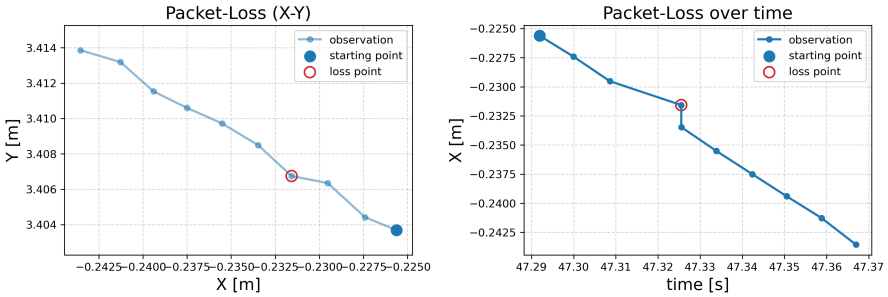
Radio-Control System

The Radio-Control System enables developers to send control signals to the VCU. It consists of a transmitter-receiver pair operating on a specific frequency band, with each receiver exclusively paired with one transmitter. For safety, the transmitter-receiver connection must be established before vehicle operation. Additionally, Wi-Fi within the DCSC Lab LAN facilitates autonomous control. The DCSC platform employs both the transmitter-receiver pair and Wi-Fi for remote control.

Vehicle Control Unit



(a) An example of inaccurate observations



(b) An example of packet-loss

Figure 2.1: The challenges in the observations from MoCap in DCSC Lab.

The Vehicle Control Unit comprises low-level controllers that execute control commands from developers. The autopilot manages information flow within the unit, with software optimized for specific development boards to ensure safety, efficiency, and maintainability. In the DCSC platform, the *ErleBrain3* development board running *Ubuntu 16.04 OS* is used, with *Ardupilot*[1] as the autopilot software. Communication with *Ardupilot* is facilitated by MAVROS[2] through ROS.

Driving System

The Driving System includes the steering and propulsion components that drive the vehicle. These components determine the vehicle's dynamics, and the modeling of vehicle motion and control strategies depend on them. In the DCSC platform, a steering servo adjusts the front wheels' angle, and a motor with an Electronic Speed Controller (ESC) provides propulsion.

2.1.2. Challenges

During the implementation of the vehicle platform, several realistic issues emerged that hinder the development of autonomous vehicle applications. These challenges include: (1) inaccurate measurements from the Motion Capture System; (2) packet loss during data collection by the Ground Control Station; (3) unknown dynamics of the servo and motor.

Inaccurate Observation

As shown in Figure 2.1a, the left side displays the observed vehicle position in the global frame, where some large deviations indicate clearly inaccurate measurements. On the right side shows the deviations over time. One potential cause of these inaccuracies is the sensitivity of the MoCap's tracking stability to lighting conditions, as strong environment light can degrade the performance of tracking algorithm.

Packet Loss

As shown in Figure 2.1b, the left side displays the observed vehicle position in the global frame, where packet loss is not apparent. However, upon examining the observations over time on the right side, one observation shares nearly the same timestamp as the following observation, while the X position differs significantly. This discrepancy is caused by packet loss. Due to the properties of the TCP communication protocol[3], failed data transmissions are buffered and sent once the connection is restored. However, the timestamps

[1] Ardupilot open-source documentation: `https://ardupilot.org/ardupilot/`
[2] MAVROS open-source documentation: `https://wiki.ros.org/mavros`
[3] UDPROS open-source documentation: `https://wiki.ros.org/ROS/UDPROS`

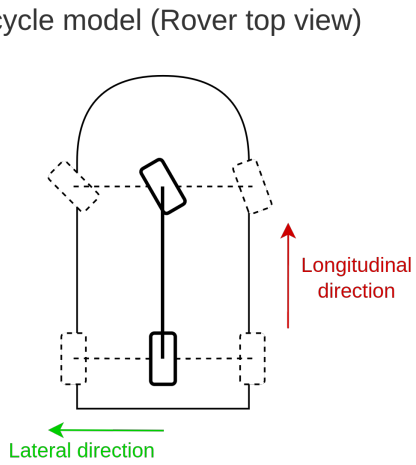


Figure 2.2: Bicycle model simplification

of the data are assigned upon being received by the GCS. Consequently, packet loss leads to incorrect timestamps for observations.

Unknown Actuator Dynamics

There is a relationship between the throttle control signal and the vehicle's velocity. When a fixed steering command is applied, the vehicle's velocity changes according to the throttle command, and the velocity stabilizes at a specific value after a while. However, the relationship of how fast and to which value the vehicle velocity converges under different throttle command is unknown. Similarly, the relationship between the steering command and the vehicle steering angle is also unclear.

2.2. Vehicle Dynamics

Vehicle dynamics describes the motion of vehicles, capturing the relationship between the vehicle's states and the inputs applied. Certain assumptions can be made to simplify the modeling of dynamics without losing generality. In this section, both the kinematic bicycle model and the vehicle-body dynamics are reviewed and studied to establish a fundamental understanding of vehicle dynamics.

2.2.1. Overview of Simplifications

To study the vehicle dynamics, it is necessary to specify the field of application, for example, the cross-country vehicle, the city-use vehicle and the formula-1 vehicle have very different operation requirements. In the DCSC Lab, the vehicle is operation on a smooth, plain ground with rather low velocity, accordingly, certain simplifications can be applied [13, 26]:

Kinematic Model

The vehicle motion is modeled using geometric relationships, ignoring the effects of external forces, moment, and vehicle inertia on the vehicle's linear and angular acceleration.

Bicycle Model

As shown in Figure 2.2, the lateral distance between the front and rear wheels is ignored. According to Ackermann Steering [4], the two front wheels must have different steering angles to avoid scrubbing while turning. In the bicycle model, the difference between the two front wheels is ignored, and the vehicle's steering angle is assumed to align with the front wheel steering angle.

Planar Motion

The (x-y) plane of the vehicle-body frame is parallel to the (X-Y) plane of the global frame. The vehicle motion is assumed to remain parallel to the ground plane, with both pitch (vehicle body rotation around the $x_{vehicle}$ axis) and roll (vehicle body rotation around the $y_{vehicle}$ axis) set to zero. The vehicle's position and yaw (vehicle body rotation around the $z_{vehicle}$ axis) can be projected onto a 2D plane without distortion.

2.2.2. Vehicle Dynamics with Only Kinematic Relationships

The kinematic bicycle model [4] simplifies the vehicle as a bicycle and considers only the kinematic relationships of its dynamics. This model captures most of the properties of vehicle motion under nominal operating conditions. Its simplicity and generality make it a popular choice for planning and control.

Assumptions

1. Kinematic Model: Only the geometric relationships are considered.
2. Planar Motion: The vehicle's motion is confined to a 2D plane.
3. Bicycle Model: The difference in orientation between the two front wheels is ignored.

Dynamics

The state of the model is $[X, Y, \Psi] \in \mathbb{R}^3$, (X, Y) is the position in the global coordinate frame, and Ψ is the orientation (Yaw angle). The input of the system is $[V_t, \delta_t] \in \mathbb{R}^2$, V_t is the longitudinal velocity of vehicle in the vehicle body frame and δ_t is the front wheel steering angle.

Given the wheel base length L_{wb}, the kinematic bicycle model dynamics can be expressed as [4]:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} V_t \cdot cos(\Psi) \\ V_t \cdot sin(\Psi) \\ \frac{V_t \cdot tan(\delta_k)}{L_{wb}} \end{bmatrix} \tag{2.1}$$

Limitations of the Kinematic Bicycle Model

According to study [24], the kinematic bicycle model can capture most of the vehicle dynamics when the lateral acceleration is below $0.5\mu g$ (where μ is the road friction coefficient and g is the gravitational acceleration). However, if the vehicle is driving on a low-friction, winding track, the lateral acceleration becomes highly sensitive to changes in longitudinal velocity. In such conditions, the vehicle's linear and angular acceleration must be considered to accurately model its motion.

2.2.3. Vehicle Dynamics Considering External Force Moment and Vehicle Inertial

The vehicle-body dynamics describe how the external force, moment and vehicle inertial govern the vehicle's linear and angular acceleration. This section takes tire force, resistance force into consideration to develop a more general and complex vehicle dynamics [13].

Assumptions

1. Planar Motion: The vehicle's motion is confined to a 2D plane.
2. Bicycle Model: The difference in orientation between the two front wheels is ignored.
3. Linear tire model: The lateral tire force is linear to the slip angle and the longitudinal tire force is linear to the slip ratio.
4. Uniform Longitudinal Tire Stiffness: The front and rear wheels share the same longitudinal tire stiffness.
5. Uniform Lateral Tire Stiffness: The front and rear wheels share the same lateral tire stiffness.
6. No Front Wheel Slip Ratio: The front wheel's slip ratio is assumed to be zero.

Tire Force Modeling

The main source of external force acting on the vehicle originates from the tires. The tires generate both longitudinal and lateral forces through interaction with the ground. The lateral force steers the vehicle, while the longitudinal force accelerates or decelerates it.

The lateral tire force is generated from the tire slip angle, as shown in Figure 2.3. The tire slip angle is defined as the difference between the tire's orientation and the direction of its velocity [8]. Given the vehicle's lateral velocity V_y, the direction of the front wheel velocity θ_{vf}, the rear wheel velocity θ_{vr}, and the front wheel steering angle δ, the tire slip angles at front wheels α_f and rear wheels α_r can be expressed as:

$$\begin{aligned} \alpha_f &= \delta - \theta_{vf} \approx \delta - \frac{V_y + l_f \Omega}{V_x} \\ \alpha_r &= -\theta_{vr} \approx -\frac{V_y - l_r \Omega}{V_x} \end{aligned} \tag{2.2}$$

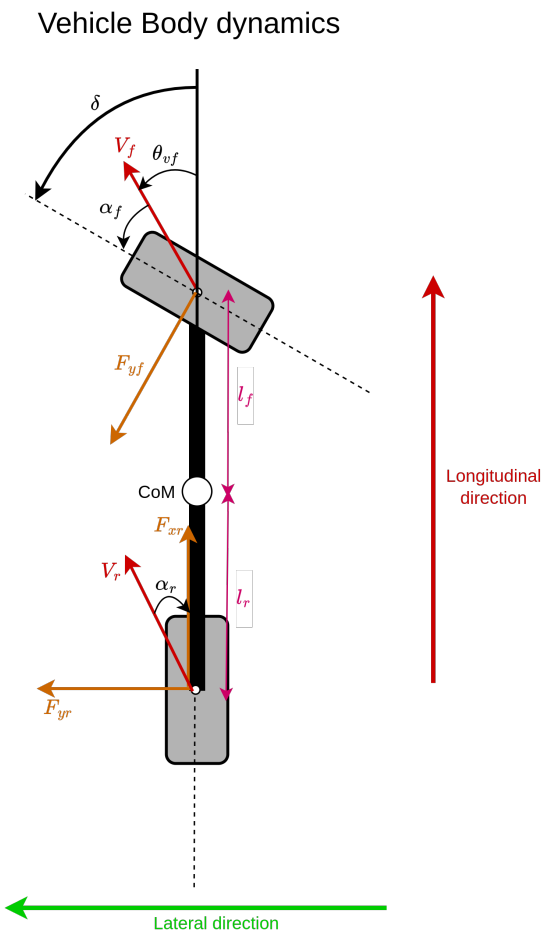


Figure 2.3: An illustration of the vehicle body's external forces, moment, and inertial modeling, V_f is the direction of front wheel velocity, V_r is the direction of rear wheel velocity.

where l_f and l_r are the distances from the Center of Mass (CoM) to the front and rear wheels, respectively.

The longitudinal tire force is generated from longitudinal slip, as shown in Figure 2.3. Longitudinal slip [26] is defined as the difference between the tire's longitudinal velocity V_x and its linear velocity $r_{eff} \cdot \omega$, where r_{eff} is the effective wheel radius and ω is the wheel angular velocity. The slip ratio σ_x is expressed as:

$$\sigma_x = \begin{cases} \frac{r_{eff} \cdot \omega - V_x}{r_{eff} \cdot \omega}, & \text{if accelerating} \\ \frac{r_{eff} \cdot \omega - V_x}{V_x}, & \text{if braking} \end{cases} \tag{2.3}$$

Assuming a linear tire model, given the tire model coefficients – longitudinal tire stiffness C_σ and lateral tire stiffness C_α—the longitudinal tire forces for the front wheel F_{xf} and rear wheel F_{xr}, as well as the lateral tire forces for the front wheel F_{yf} and rear wheel F_{yr}, are proportional to the slip ratio and slip angle. These forces can be expressed as:

$$F_{xf} = 0, \quad F_{xr} = C_\sigma \cdot \sigma_x, \quad F_{yf} = C_\alpha \cdot \alpha_f, \quad F_{yr} = C_\alpha \cdot \alpha_r \tag{2.4}$$

Force-Moment Vehicle Body Dynamics

Applying Newton's laws of motion, the vehicle body dynamics under external forces, moment, and vehicle inertia are analyzed. Given the mass of the vehicle m, the relationship between the total lateral force F_y, total longitudinal force F_x, lateral inertial acceleration a_y, and longitudinal inertial acceleration a_x is:

$$m \cdot a_y = F_y, \quad m \cdot a_x = F_x \tag{2.5}$$

From Figure 2.3, the total longitudinal force F_x is the sum of vector component of F_{xr} and F_{yf} in the longitudinal direction, while the total lateral force F_y is the sum of vector component of F_{yr} and F_{yf} in the lateral direction:

$$F_x = F_{xr} - F_{yf} \cdot \sin\delta, \quad F_y = F_{yr} + F_{yf} \cdot \cos\delta \tag{2.6}$$

Given the vehicle's longitudinal velocity V_x, lateral velocity V_y, and angular velocity Ω, the lateral inertial acceleration is a combination of the lateral acceleration $\dot{V}_y$ and the centripetal acceleration $V_x \cdot \Omega$. Similarly, the longitudinal inertial acceleration is a combination of the longitudinal acceleration $\dot{V}_x$ and the centripetal acceleration $V_y \cdot \Omega$:

$$a_y = \dot{V}_y + V_x \cdot \Omega, \quad a_x = \dot{V}_x - V_y \cdot \Omega \tag{2.7}$$

Substituting a_y and a_x from Equation (2.7) into Equation (2.5), the accelerations $\dot{V}_y$ and $\dot{V}_x$ are:

$$\dot{V}_y = \frac{F_y}{m} - V_x \cdot \dot{\psi} = \frac{F_{yr} + F_{yf} \cdot \cos\delta}{m} - V_x \cdot \dot{\psi} \tag{2.8}$$

$$\dot{V}_x = \frac{F_x}{m} + V_y \cdot \dot{\psi} = \frac{F_{xr} - F_{yf} \cdot \sin\delta}{m} + V_y \cdot \dot{\psi} \tag{2.9}$$

Applying Newton's Second Law for Rotation, as shown in Figure 2.3, given the distance from the center of gravity (CoG) to the rear wheel l_r and the distance from the CoG to the front wheel l_f, the relationship between the moment of inertia I_z, the vehicle's angular acceleration $\dot{\Omega}$, the tire forces F_{yf} and F_{yr}, and the vehicle's steering angle δ can be expressed as:

$$\dot{\Omega} = \frac{l_f \cdot F_{yf} \cdot \cos\delta - l_r \cdot F_{yr}}{I_z} \tag{2.10}$$

Dynamics

There are six states in the model: $[X, Y, \Psi, V_x, V_y, \Omega] \in \mathbb{R}^6$, which can be divided into two parts: vehicle body dynamics and kinematic relationships.

For the vehicle body dynamics, the state $[V_x, V_y, \Omega]$ represents the longitudinal velocity, lateral velocity, and angular velocity in the vehicle body frame. From Equations (2.9), (2.8), and (2.10), and by substituting the tire forces using Equation (2.4), the vehicle body dynamics can be expressed as:

$$\begin{bmatrix} \dot{V}_x \\ \dot{V}_y \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{m} \cdot (C_\sigma \cdot \sigma_x - C_\alpha \cdot \alpha_f \cdot \sin\delta) + \Omega \cdot V_y \\ \frac{1}{m} \cdot (C_\alpha \cdot \alpha_f \cdot \cos\delta + C_\alpha \cdot \alpha_r) - \Omega \cdot V_x \\ \frac{1}{I_z} \cdot (l_f \cdot C_\alpha \cdot \alpha_f \cdot \cos\delta - l_r \cdot C_\alpha \cdot \alpha_r) \end{bmatrix} \tag{2.11}$$

Here, C_σ is the longitudinal tire stiffness of the front and rear wheels, C_α is the lateral tire stiffness parameter of the front and rear wheels, respectively, and I_z is the moment of inertia at the vehicle's center of gravity (CoG).

For the kinematic relationships, the state $[X, Y, \Psi]$ represents the position and orientation of the vehicle in the global frame. The dynamics can be expressed as:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} V_x \cdot \cos(\Psi) - V_y \cdot \sin(\Psi) \\ V_x \cdot \sin(\Psi) + V_y \cdot \cos(\Psi) \\ \Omega \end{bmatrix} \tag{2.12}$$

Combining Equations (2.11) and (2.12) into a function $f(\cdot) : \mathbb{R}^6 \to \mathbb{R}^6$, and denoting the system state as $\mathscr{X} := [X, Y, \Psi, V_x, V_y, \Omega]$, the system dynamics can be written as:

$$\dot{\mathscr{X}} = f(\mathscr{X}) \tag{2.13}$$

2.3. Kalman Filter and its Variations

Under observational disturbances, controlling the vehicle becomes challenging, as the observations are not suitable for direct use in vehicle control. Therefore, the true state of the system must be inferred from noisy observations. The Kalman filter is an algorithm designed to recursively estimate the true states of a system. In this section, the formulation of Bayesian filtering problem, the basic Kalman filter [28] and its variations, which address different types of system dynamics, are reviewed [2, 3, 7, 28, 33].

2.3.1. Formulation of Bayesian Filtering Problem

Given the observations and the prior knowledge of the system, the problem that Bayesian filtering aims to resolve is how to use this information to estimate the true state of the system accurately and quickly during operation. In Bayesian filtering, the system is described by a Hidden Markov Model (HMM) [25].

Elements of Hidden Markov Model

The Hidden Markov Model is a probabilistic framework used to describe the evolving nature of time-varying processes [25]. It assumes that the exact full evolution of the system is not explicitly known, and only partial information about the system can be observed.

For example, consider a discrete-time system where the state of the system changes over time, and a sensor measures the states of the system. To describe the system using an HMM, four elements define the time-varying process: the dynamic states $\mathbf{x}_{1:t}$, the observations $\mathbf{y}_{1:t}$, the dynamics model $\mathscr{F}$, and the observation model $\mathscr{H}$.

Dynamic States

The non-observable part of the HMM is the latent dynamic states: $\mathbf{x}_{1:t} := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$, which carry information about the system dynamics. According to the *Markovian property* (Equation (2.14)) [25], the current state $\mathbf{x}_t$ is assumed to depend only on the current latent state $\mathbf{x}_{t-1}$. This relationship can be represented by the state transition probability distribution, as shown in Equation (2.15):

$$p(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \tag{2.14}$$

$$\mathbf{x}_t \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \tag{2.15}$$

Observations

The observable part of the HMM is the observations: $\mathbf{y}_{1:t} := \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t\}$, which reflect partial information about the latent states. According to the *conditional independence of the observation* (Equation (2.16)) [28], $\mathbf{y}_t$ depends only on the latent dynamic state $\mathbf{x}_t$. The relationship between the latent dynamic states and the observations can be represented by the observation probability distribution, as shown in Equation (2.20):

$$p(\mathbf{y}_t \mid \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1}) = p(\mathbf{y}_t \mid \mathbf{x}_t) \tag{2.16}$$

$$\mathbf{y}_t \sim p(\mathbf{y}_t \mid \mathbf{x}_t) \tag{2.17}$$

Initial Distribution, Dynamic Model, and Observation Model

The dynamic model (or process model) describes how the system states evolve over time, including their uncertainty. It is defined as a sequence of transition probability distributions $\mathscr{F} = \{p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \mid t = 1, \ldots, n\}$. The observation model describes how the observations depend on the latent states, including their uncertainty, and is denoted as $\mathscr{H} = \{p(\mathbf{y}_t \mid \mathbf{x}_t) \mid t = 1, \ldots, n\}$. The initial state probability distribution is assumed to be $p(\mathbf{x}_0)$.

Bayesian Filtering Problem and Solution

The goal of Bayesian filtering is to estimate the probability distribution of the latent states $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$, given the system's prior knowledge $\Theta := [\mathscr{F}, \mathscr{H}, p(\mathbf{x}_0)]$ and the observations $\mathbf{y}_{1:t}$.

Given the dynamic model $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \in \mathscr{F}$ and the historical observations $\mathbf{y}_{1:t-1}$, it is possible to infer the latent state at current time step, the predictive distribution of the latent state is obtained as follows [28]:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}) \, d\mathbf{x}_{t-1} \tag{2.18}$$

When the current observation $\mathbf{y}_t$ is obtained, the predictive estimate of the latent state is updated using Bayes' rule. The posterior distribution of the latent state $\mathbf{x}_t$ is then given by:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t) \cdot p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}{p(\mathbf{y}_{1:t})} \tag{2.19}$$

Given the historical observations $\mathbf{y}_{1:t-1}$, the distribution of the observations $\mathbf{y}_{1:t}$ in Equation (2.19) can be computed by marginalizing over the predictive distribution of latent state $\mathbf{x}_t$:

$$p(\mathbf{y}_{1:t}) = \int p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) \, d\mathbf{x}_t \tag{2.20}$$

2.3.2. Kalman Filter for Linear-Gaussian System

Consider a system with a linear dynamic model and a linear observation model, where the process noise and observation noise are zero-mean Gaussian distributed:

$$\mathbf{x}_t = \mathbf{F} \cdot \mathbf{x}_{t-1} + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}), \tag{2.21}$$

$$\mathbf{y}_t = \mathbf{H} \cdot \mathbf{x}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}), \tag{2.22}$$

where $\mathbf{F} \in \mathbb{R}^{N_x \times N_x}$ is the state transition matrix, $\mathbf{H} \in \mathbb{R}^{N_y \times N_x}$ is the observation matrix, $\mathbf{w}_t \in \mathbb{R}^{N_x}$ is the process noise, $\mathbf{v}_t \in \mathbb{R}^{N_y}$ is the observation noise.

Kalman Filtering Algorithm

A key property of the linear-Gaussian model is that a Gaussian distribution remains Gaussian after a linear transformation [28]. Based on Equations (2.21) and (2.22), the predictive distributions of the latent state and observation can be expressed as:

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t \mid \mathbf{F} \cdot \mathbf{x}_{t-1}, \mathbf{Q}), \tag{2.23}$$

$$p(\mathbf{y}_t \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t \mid \mathbf{H} \cdot \mathbf{x}_t, \mathbf{R}). \tag{2.24}$$

Here, both the dynamic model and the observation model are Gaussian distributed. The mean of each distribution is obtained by multiplying the state transition matrix $\mathbf{F}$ or observation matrix $\mathbf{H}$.

Given the likelihood of the previous latent state $\mathbf{x}_{t-1}$, with mean μ_{t-1} and covariance Σ_{t-1}:

$$\mathbf{x}_{t-1} \sim \mathcal{N}(\mu_{t-1}, \Sigma_{t-1}), \tag{2.25}$$

The predictive distribution of $\mathbf{x}_t$ (derived from Equation (2.18) in discrete time) is:

$$\begin{aligned} \mathbf{x}_{t|t-1} \sim p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) &= \mathcal{N}(\mu_{t|t-1}, \Sigma_{t|t-1}), \\ \mu_{t|t-1} &= \mathbf{F} \cdot \mu_{t-1}, \\ \Sigma_{t|t-1} &= \mathbf{F} \cdot \Sigma_{t-1} \cdot \mathbf{F}^T + \mathbf{Q}. \end{aligned} \tag{2.26}$$

The current measurement $\mathbf{y}_t$ is used by the Kalman filter to refine the estimate of the latent state. The posterior distribution of the state estimation with mean μ_t and covariance Σ_t can be written as[7]:

$$\begin{aligned} p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) &= \mathcal{N}(\mu_t, \Sigma_t), \\ \mu_t &= \mu_{t|t-1} + \mathbf{K}_t \cdot (\mathbf{y}_t - \mathbf{H} \cdot \mu_{t|t-1}), \\ \Sigma_t &= \Sigma_{t|t-1} - \mathbf{K}_t \cdot \mathbf{S}_t \cdot \mathbf{K}_t^T, \\ \mathbf{S}_t &= \mathbf{H} \cdot \Sigma_{t|t-1} \cdot \mathbf{H}^T + \mathbf{R}, \\ \mathbf{K}_t &= \Sigma_{t|t-1} \cdot \mathbf{H}^T \cdot \mathbf{S}_t^{-1}. \end{aligned} \tag{2.27}$$

Here, $\mathbf{K}_t$ is the Kalman gain, which determines the weight assigned to the residual (the difference between the observation and the prediction) to improve the posterior estimation.

2.3.3. Extended Kalman Filter for Nonlinear-Gaussian System

Consider a system with a nonlinear dynamics model and a nonlinear observation model, where the process noise and observation noise are zero-mean Gaussian distributed:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}), \tag{2.28}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}), \tag{2.29}$$

where $f(\cdot) : \mathbb{R}^{N_x} \to \mathbb{R}^{N_x}$ is the nonlinear state transition function, and $h(\cdot) : \mathbb{R}^{N_x} \to \mathbb{R}^{N_y}$ is the nonlinear observation function. Similar to the linear-Gaussian case (Section 2.3.2), the noise terms $\mathbf{w}_t$ and $\mathbf{v}_t$ are additive and follow zero-mean Gaussian distributions with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, respectively.

Approximating the Nonlinear System with Taylor-Series Expansion

A nonlinear system represents a more general form of dynamics. After a nonlinear transformation, the Gaussian distribution is not necessarily preserved [28]. To apply the Kalman filter algorithm for recursively estimating the posterior of the latent states, the nonlinear system is locally linearized near a fixed point. This linearization ensures that the Gaussian distribution is preserved, as the original nonlinear functions are approximated by linear functions. The linearization is achieved through a first-order Taylor-series expansion.[28].

Assuming the likelihood of the previous state $\mathbf{x}_{t-1}$ is the same as in Equation (2.25) and the predictive distribution $\mathbf{x}_{t|t-1}$ is as in Equation (2.26), within the region near the fix point $[\mathbf{x}-\delta\mathbf{x}, \mathbf{x}+\delta\mathbf{x}]$ (the region is supposed to be relatively small), the approximation can be written as:

$$f(\mathbf{x}_{t-1}) \approx f(\mu_{t-1}) + F_x(\mu_{t-1}) \cdot \delta\mathbf{x}, \tag{2.30}$$

$$h(\mathbf{x}_{t|t-1}) \approx h(\mu_{t|t-1}) + H_x(\mu_{t|t-1}) \cdot \delta\mathbf{x}, \tag{2.31}$$

$$F_x(\mu_{t-1}) = \frac{\partial f}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mu_{t-1}} \in \mathbb{R}^{N_x \times N_x}, \tag{2.32}$$

$$H_x(\mu_{t|t-1}) = \frac{\partial h}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mu_{t|t-1}} \in \mathbb{R}^{N_y \times N_x}, \tag{2.33}$$

where $F_x(\mu_{t-1})$ is the Jacobian of the state transition function evaluated at the point μ_{t-1}, and $H_x(\mu_{t|t-1})$ is the Jacobian of the observation function evaluated at the point $\mu_{t|t-1}$.

Extended Kalman Filtering Algorithm

The assumptions and steps taken so far for the nonlinear-Gaussian system allow us to apply the Kalman filter algorithm. The necessary conditions to apply the Kalman filter algorithm are:

- The additive noise is assumed to be zero-mean Gaussian.
- The dynamic model and the observation model are linear.

For the prediction step, the state transition matrix $\mathbf{F}$ in Equation (2.26) is replaced with the linearized state transition function $F_x(\mu_{t-1})$ (Equation (2.32)). For the update step, the observation matrix $\mathbf{H}$ in Equation (2.27) is replaced with the linearized observation function $H_x(\mu_{t|t-1})$ (Equation (2.33)).

2.3.4. Kalman Filter with Adaptive Process and Observation Noise Covariance

Consider a system with a nonlinear dynamic model and a nonlinear observation model, where the process noise and observation noise are both zero-mean Gaussian distributed, with time-varying process noise covariance $\mathbf{Q}_t$ and observation noise covariance $\mathbf{R}_t$. The system is described by the following equations:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t) \tag{2.34}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t) \tag{2.35}$$

where $\mathbf{w}_t$ is the process noise with time-varying covariance $\mathbf{Q}_t$, $\mathbf{v}_t$ is the observation noise with time-varying covariance $\mathbf{R}_t$, and the dynamic function $f(\cdot)$ and observation function $h(\cdot)$ have the same definitions as in Section 2.3.3.

Covariance Matching

The tuning of the process noise covariance $\mathbf{Q}_t$ and the observation noise covariance $\mathbf{R}_t$ can significantly affect the performance of Kalman filters. During the filtering process, the differences between the observations, predictions, estimations indicate the accuracy of the observations and the process model.

Covariance matching is an adaptive estimation approach that adjusts the covariance matrices based on the innovation and residual [3]. The innovation $\mathbf{d}_t$ is defined as the difference between the predicted observation and the actual observation, while the residual ϵ_t is defined as the difference between the posterior estimate and the observation. Given the mean of the predictive state $\mathrm{E}[\mathbf{x}_{t|t-1}] = \mu_{t|t-1}$ and the mean of the posterior state $\mathrm{E}[\mathbf{x}_t] = \mu_t$, the innovation and residual are defined as:

$$\mathbf{d}_t = \mathbf{y}_t - h(\mu_{t|t-1}) \tag{2.36}$$

$$\epsilon_t = \mathbf{y}_t - h(\mu_t) \tag{2.37}$$

During the filtering process, if the innovation increases, it suggests that the currently assumed process noise covariance $\mathbf{Q}_t$ is smaller than the ground truth, indicating that the assumed process model is less reliable. In this case, the process noise covariance should be increased. Similarly, if the residual increases, it suggests that the currently assumed observation noise covariance $\mathbf{R}_t$ is smaller than the ground truth, indicating that the sensor is less reliable, and the observation noise covariance should be increased. The adjustment rules are given by [3]:

$$\mathbf{Q}_t = \lambda \mathbf{Q}_{t-1} + (1-\lambda) \cdot (\mathbf{K}_t \cdot \mathbf{d}_t \mathbf{d}_t^T \cdot \mathbf{K}_t^T) \tag{2.38}$$

$$\mathbf{R}_t = \lambda \mathbf{R}_{t-1} + (1-\lambda) \cdot (\epsilon_t \epsilon_t^T + \mathbf{H}_x(\mu_{t|t-1}) \cdot \Sigma_{t|t-1} \cdot \mathbf{H}_x(\mu_{t|t-1})^T) \tag{2.39}$$

where the evolution of the process noise covariance $\mathbf{Q}_t$ and observation noise covariance $\mathbf{R}_t$ is modeled as a first-order system. The parameter $\lambda \in [0,1]$ controls the rate at which the covariance matrices adapt to changes in the innovation and residual. When $\lambda = 0$, the covariance matrices instantly adapt to the changes.

2.3.5. Robustify the Kalman Filter with Adaptive Observation Covariance

Consider a system with a nonlinear dynamic model and a nonlinear observation model, where the process noise and observation noise are both zero-mean Gaussian distributed, with time-varying observation noise covariance $\mathbf{R}_t$. The system is described by the following equations:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{2.40}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t), \quad \mathbf{R}_t \sim \mathcal{W}^{-1}(\nu_0 \Lambda_0, \nu_0) \tag{2.41}$$

Here, $\mathbf{R}_t$ follows an Inverse Wishart distribution with a inverse scale matrix $\Lambda_0 \in \mathbb{R}^{N_y \times N_y}$ ($\Lambda_0 > 0$) and degree-of-freedom $\nu_0 \in \mathbb{R}$ ($\nu_0 > N_y - 1$).

Gaussian-Inverse-Wishart Distribution

The observation noise of the system, as described in Equation (2.41), is Gaussian distributed, with the observation noise covariance drawn from an inverse-Wishart distribution. The covariance in the Gaussian distribution represents uncertainty, and the inverse-Wishart distribution introduces additional uncertainty to the observation noise. If the observation noise is Gaussian-Inverse-Wishart distributed, it follows a sub-exponential distribution with heavy-tail properties [1]. This assigns a non-negligible probability to the occurrence of observational outliers to model them.

The inverse-Wishart distribution is the conjugate prior for the covariance matrix of the Gaussian distribution [20]. This means that if the prior observation noise covariance is inverse-Wishart distributed, the posterior observation noise covariance will also follow an inverse-Wishart distribution. Given the state $\mathbf{x}_t$ and the prior observation noise covariance $\mathbf{R}_t$, the predictive distribution of the observation $\mathbf{y}_t$ can be expressed as:

$$\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{R}_t \sim \mathcal{N}(h(\mathbf{x}_t), \mathbf{R}_t), \quad \mathbf{R}_t \sim \mathcal{W}^{-1}(\nu_0 \Lambda_0, \nu_0) \tag{2.42}$$

Given the actual observation $\mathbf{y}_t$, the posterior distributions of the latent state $\mathbf{x}_t$ and the observation noise covariance $\mathbf{R}_t$ are:

$$\mathbf{x}_t \mid \mathbf{y}_{1:t} \sim \mathcal{N}(\mu_t, \Sigma_t), \quad \mathbf{R}_t \mid \mathbf{y}_{1:t} \sim \mathcal{W}^{-1}(\nu_t \Lambda_t, \nu_t) \tag{2.43}$$

Given the covariance of the residual $\mathbf{B}_t = \mathrm{E}[\epsilon_t \cdot \epsilon_t^T]$, the posterior parameters of the inverse-Wishart distribution are updated as follows [2]:

$$\nu_t = \nu_0 + 1 \tag{2.44}$$

$$\Lambda_t = \frac{\mathbf{B}_t + \nu_0 \cdot \Lambda_0}{\nu_0 + 1} \tag{2.45}$$

From Equation (2.45), it can be observed that a higher degree-of-freedom ν_0 results in the posterior scale matrix Λ_t being closer to Λ_0, indicating less uncertainty in the observation noise covariance.

Extended Kalman Filter Inverse Wishart Algorithm

Formulating the HMM based on the system, there are two latent states that need to be estimated: the dynamic state $\mathbf{x}_t$ and the observational noise covariance $\mathbf{R}_t$. Applying Bayes' rule, the posterior of the latent states can be expressed as:

$$p(\mathbf{x}_t, \mathbf{R}_t \mid \mathbf{y}_t) = \frac{p(\mathbf{x}_t, \mathbf{R}_t, \mathbf{y}_t)}{p(\mathbf{y}_t)}. \tag{2.46}$$

To avoid explicitly calculating the likelihood of the observation $\mathbf{y}_t$ and to decompose the interrelation between the two latent states, the variational Bayesian method is applied. During each filtering time step, the estimation of the latent states is updated iteratively (*e.g.*, if $\mathbf{x}_t$ is fixed, the optimal $\mathbf{R}_t$ is calculated). The Extended Kalman Filter Inverse Wishart Algorithm is provided in Algorithm 1.

Algorithm 1 Extended Kalman Filter with Inverse Wishart Update [2]:

Input: μ_{t-1}, Σ_{t-1}, $f(\cdot)$, $\mathbf{F}_x(\cdot)$, $h(\cdot)$, $\mathbf{H}_x(\cdot)$, $\mathbf{y}_t$, Λ_0, ν_0
Output: μ_t, Σ_t
Step 1: Prediction
$\mu_{t|t-1} \leftarrow f(\mu_{t-1})$
$\Sigma_{t|t-1} \leftarrow \mathbf{F}_x(\mu_{t-1}) \cdot \Sigma_{t-1} \cdot \mathbf{F}_x(\mu_{t-1})^T + \mathbf{Q}$
Step 2: Iterative Update
$\Lambda_t \leftarrow \Lambda_0$
$\mu_t \leftarrow \mu_{t|t-1}$
$\Sigma_t \leftarrow \Sigma_{t|t-1}$
repeat
  Step 2.1: Update Inverse Wishart parameters
  $\epsilon_t \leftarrow \mathbf{y}_t - h(\mu_t)$
  $\mathbf{B}_t \leftarrow \mathrm{E}[\epsilon_t \cdot \epsilon_t^T]$
  $\nu_t \leftarrow \nu_0 + 1$
  $\Lambda_t \leftarrow \frac{\mathbf{B}_t + \nu_0 \cdot \Lambda_0}{\nu_0 + 1}$
  Step 2.2: Update state mean and covariance
  $\mathbf{S}_t \leftarrow \mathbf{H}_x(\mu_{t|t-1}) \cdot \Sigma_{t|t-1} \cdot \mathbf{H}_x(\mu_{t|t-1})^T + \Lambda_t$
  $\mathbf{K}_t \leftarrow \Sigma_{t|t-1} \cdot \mathbf{H}_x(\mu_{t|t-1})^T \cdot \mathbf{S}_t^{-1}$
  $\mu_t \leftarrow \mu_{t|t-1} + \mathbf{K}_t \cdot \left[\mathbf{y}_t - h(\mu_{t|t-1})\right]$
  $\Sigma_t \leftarrow \Sigma_{t|t-1} - \mathbf{K}_t \cdot \mathbf{S}_t \cdot \mathbf{K}_t^T$
until Convergence of μ_t and Σ_t
Return: μ_t, Σ_t

2.3.6. Robustify the Kalman Filter with an Observational Outlier Detector

Assuming observations are occasionally contaminated by outliers, nominal observations are Gaussian distributed with a fixed covariance, while contaminated observations have a significantly larger covariance. The process noise is Gaussian distributed, and the system is defined as:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}) \tag{2.47}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + z_t \cdot \mathbf{v}_t + (1 - z_t) \cdot \mathbf{c}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}), \quad \mathbf{c}_t \sim \mathcal{N}(0, \mathbf{R}^c) \tag{2.48}$$

$$z_t \sim \text{Beta-Bernoulli}(\pi_t; \alpha, \beta) \tag{2.49}$$

where $z_t \in \{0, 1\}$ models outlier occurrence, π_t is the probability parameter of the Bernoulli distribution, assumed to follow a Beta distribution with shape parameters α and β. $\mathbf{R}^c$ is the covariance for contaminated observations, which is significantly larger than $\mathbf{R}$, other definitions align with the system described in Section 2.3.3.

Beta-Bernoulli Distribution

In the observation model (Equation (2.48)), the observation noise is assumed to be bi-Gaussian distributed. When $z_t = 1$, the observation is considered nominal, and when $z_t = 0$, the observation is potentially an outlier. The expectation of the posterior of z_t can be used as an observational outlier indicator. If the expectation of $z_t \mid \mathbf{y}_t$ is very close to 0 and lower than a threshold, the observation can be considered an observational outlier.

The variable z_t is assumed to follow a Beta-Bernoulli distribution, which is a special case of the Beta-Binomial distribution [18]. This model captures the statistical properties of observational outliers, which can be viewed as rare, accidental events that deviate significantly from nominal values [5].

Within the Beta-Bernoulli distribution, the Beta distribution serves as the conjugate prior to the Bernoulli distribution. This means that if the probability parameter π_t of the Bernoulli distribution is Beta distributed, the posterior of π_t will also be Beta distributed. Given the Bernoulli distribution with a Beta prior:

$$z_t \mid \pi_t \sim \text{Bernoulli}(\pi_t), \tag{2.50}$$

$$\pi_t \sim \text{Beta}(\alpha_0, \beta_0). \tag{2.51}$$

Once the observation $\mathbf{y}_t$ is obtained, and assuming the posterior of the Bernoulli variable $p(z_t \mid \mathbf{x}_t, \mathbf{y}_t)$ can

Algorithm 2 Extended Kalman Filter Bernoulli [33]:

Input: μ_{t-1}, Σ_{t-1}, $f(\cdot)$, $\mathbf{F}_x(\cdot)$, $h(\cdot)$, $\mathbf{H}_x(\cdot)$, $\mathbf{y}_t$, α_0, β_0, threshold
Output: μ_t, Σ_t
Step 1: Prediction
$\mu_{t|t-1} \leftarrow f(\mu_{t-1})$
$\Sigma_{t|t-1} \leftarrow \mathbf{F}_x(\mu_{t-1}) \cdot \Sigma_{t-1} \cdot \mathbf{F}_x(\mu_{t-1})^T + \mathbf{Q}$
Step 2: Iterative Update
$\alpha_t \leftarrow \alpha_0$
$\beta_t \leftarrow \beta_0$
$\pi_t \leftarrow 1$
$\mu_t \leftarrow \mu_{t|t-1}$
$\Sigma_t \leftarrow \Sigma_{t|t-1}$
repeat
  Step 2.1: Update Bernoulli variables
  $\epsilon_t \leftarrow \mathbf{y}_t - h(\mu_t)$
  $\mathbf{B}_t \leftarrow \mathrm{E}[\epsilon_t \cdot \epsilon_t^T]$
  $\mathrm{E}_{p(x_t)p(\pi_t)}[\ln p(\pi_t)] \leftarrow \psi(\alpha_t) - \psi(\alpha_t + \beta_t)$
  $\mathrm{E}_{p(x_t)p(\pi_t)}[\ln p(1-\pi_t)] \leftarrow \psi(\beta_t) - \psi(\alpha_t + \beta_t)$
  $q_z^*(z_t = 0) \leftarrow \exp\{-0.5 \cdot \mathrm{tr}[\mathbf{B}_t \mathbf{R}^{-1}] + \mathrm{E}_{p(x_t)p(\pi_t)}[\ln p(\pi_t)]\}$
  $q_z^*(z_t = 1) \leftarrow \exp\{\mathrm{E}_{p(x_t)p(\pi_t)}[\ln p(1-\pi_t)]\}$
  $\mathrm{E}_{p(\mathbf{x}_t)p(\pi_t)}[p(z_t)] \leftarrow \frac{q_z^*(z_t=0)}{q_z^*(z_t=0)+q_z^*(z_t=1)}$
  Step 2.2: Update Beta variables
  $\alpha_t \leftarrow \alpha_0 + \mathrm{E}_{p(\mathbf{x}_t)p(\pi_t)}[p(z_t)]$
  $\beta_t \leftarrow \beta_0 + 1 - \mathrm{E}_{p(\mathbf{x}_t)p(\pi_t)}[p(z_t)]$
  Step 2.3: Update state mean and covariance
  if $z_t \leq$ threshold **then**
    Outlier update
    $\mu_t \leftarrow \mu_{t|t-1}$
    $\Sigma_t \leftarrow \Sigma_{t|t-1}$
  else
    Inlier update
    $\mathbf{S}_t \leftarrow \mathbf{H}_x(\mu_{t|t-1}) \cdot \Sigma_{t|t-1} \cdot \mathbf{H}_x(\mu_{t|t-1})^T + \mathbf{R}$
    $\mathbf{K}_t \leftarrow \Sigma_{t|t-1} \cdot \mathbf{H}_x(\mu_{t|t-1})^T \cdot \mathbf{S}_t^{-1}$
    $\mu_t \leftarrow \mu_{t|t-1} + \mathbf{K}_t \cdot \left[\mathbf{y}_t - h(\mu_{t|t-1})\right]$
    $\Sigma_t \leftarrow \Sigma_{t|t-1} - \mathbf{K}_t \cdot \mathbf{S}_t \cdot \mathbf{K}_t^T$
  end if
until Convergence of μ_t and Σ_t
Return: μ_t, Σ_t

be calculated, the posterior of the Beta parameters π_t, α_t, and β_t can be expressed as [33]:

$$\alpha_t = \alpha_0 + \mathrm{E}[p(z_t)], \tag{2.52}$$

$$\beta_t = \beta_0 + 1 - \mathrm{E}[p(z_t)], \tag{2.53}$$

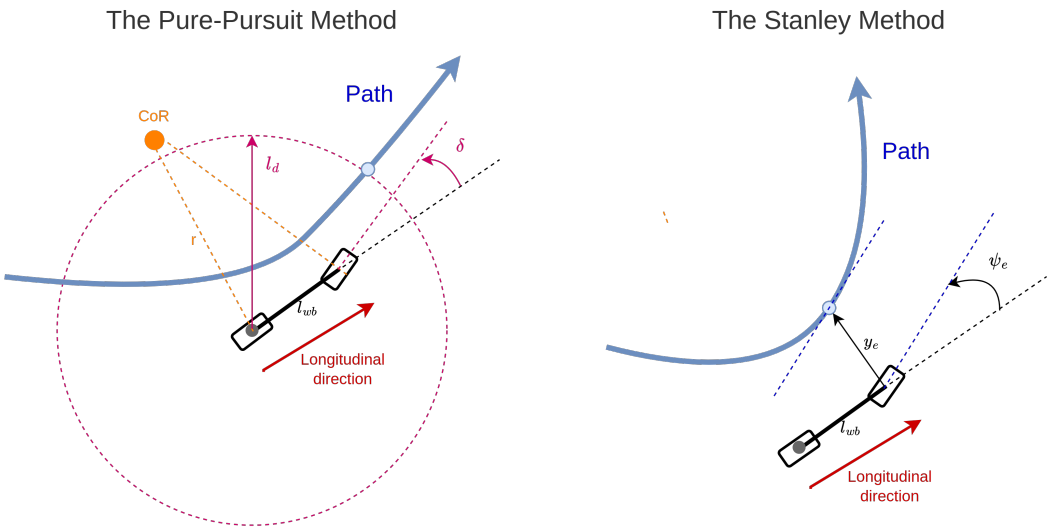
$$\pi_t \mid z_t, \mathbf{x}_t, \mathbf{y}_t \sim \mathrm{Beta}(\alpha_t, \beta_t). \tag{2.54}$$

Extended Kalman Filter Bernoulli Algorithm [33]

In the system, as described by Equations (2.48) and (2.47), there are three latent variables $[\mathbf{x}_t, z_t, \pi_t]$ that need to be estimated. Applying Bayes' rule, the posterior of the latent states can be expressed as:

$$p(\mathbf{x}_t, z_t, \pi_t \mid \mathbf{y}_t) = \frac{p(\mathbf{x}_t, z_t, \pi_t, \mathbf{y}_t)}{p(\mathbf{y}_t)} \tag{2.55}$$

where the three latent variables are interrelated, and the distribution of the observation $\mathbf{y}_t$ is extremely difficult to calculate. The Extended Kalman Filter Bernoulli Algorithm [33] applies variational Bayes to avoid the calculation of $p(\mathbf{y}_t)$ and provides a solution (not closed-form) to estimate the posterior of the latent states using the Kalman filtering algorithm. The algorithm is shown in Algorithm 2.



(a) An illustration of the Pure-Pursuit Method

(b) An illustration of the Stanley Method

Figure 2.4: Illustrations of the geometric path-tracking methods

2.4. Geometric Path-tracking control

Autonomous driving includes a wide range of applications, from planning to control. Path-tracking control is one of the fundamental tasks in autonomous driving. For example, when a feasible path is planned from location A to location B, the autonomous vehicle must be able to accurately follow the path. Geometric path-tracking methods primarily focus on the geometric relationships between the vehicle and the track. These methods are widely recognized for their computational efficiency and generality. In this section, the Pure-Pursuit Method [6] and the Stanley Method [11] are introduced.

2.4.1. Pure-pursuit method

The Pure-Pursuit Method (PPM) is a geometric path-tracking method that calculates the needed steering angle to drive the vehicle towards the selected points on the designated path (*i.e.* look-ahead point). It is a popular method due to it's robustness and simplicity, however, it can only handle the nominal driving condition, when the vehicle is operating under high velocity or on a low-friction path, the method cannot be applied[17, 22].

Searching for the Look-ahead point

While driving the vehicle, the PPM always searches for the look-ahead point on the designated path within the look-ahead distance. The choice of the point follows three rules:

- The point must lie within the look-ahead distance.
- The point should align with the vehicle's moving direction.
- The distance between the point and the vehicle should be close to the look-ahead distance.

Pure-Pursuit Method Control Law

Given the control point of the vehicle (the center of the rear axle) $[X_v, Y_v, \Psi_v]$ and the goal point (look-ahead point) on the track $[X_g, Y_g, \Psi_g]$, there exists a curve that connects the control point and the goal point.

Ackermann steering geometry describes the relationship between the radius of the curve and the wheel steering angle. Given the wheelbase length l_{wb}, the radius of the turn r, and the wheel steering angle δ, the relationship is expressed as:

$$\tan(\delta) = \frac{l_{wb}}{r} \tag{2.56}$$

As shown in Figure 2.4a, the vehicle's center of rotation (CoR), the control point (center of vehicle rear axle), and the goal point form an isosceles triangle. Given the fixed look-ahead distance l_d, the relationship

between the turn radius r and the lateral error y_e is expressed as:

$$r = \frac{l_d^2}{2 \cdot y_e} \tag{2.57}$$

The lateral error y_e is the distance from the vehicle to the path in the vehicle-body frame. Given the position and orientation of the points in the global frame, the relationship is described as:

$$y_e = -(X_v - X_g) \cdot sin(\Psi_v) + (Y_v - Y_g) \cdot cos(\Psi_v) \tag{2.58}$$

Combine Equation (2.56) and Equation (2.57), the steering control law can be given as:

$$\delta = \arctan\left(\frac{2 \cdot y_e \cdot l_{wb}}{l_d^2}\right) \tag{2.59}$$

During path-tracking, the vehicle does not necessarily need to reach the goal points: when the vehicle is driving on the track, the controller will constantly search for the next goal points. The look-ahead distance l_d is crucial for the controller's performance. A look-ahead distance that is too small leads to oscillations around the reference path, while one that is too large results in slow convergence of the lateral error [6].

2.4.2. Stanley Method

The Stanley method is a geometric path-tracking method developed by a research group at Stanford University for the DARPA Grand Challenge 2005 [11]. It penalizes both the position and orientation differences between the vehicle and the path.

Stanley Method Control Law

As shown in Figure 2.4b, the control law consists of two terms: the orientation error ψ_e and the cross-track error y_e. The orientation error ψ_e is defined as the difference between the orientation of the goal point and the control point in the vehicle-body frame. Given the orientation of the control point Ψ_v and the orientation of the goal point Ψ_g, the orientation error ψ_e is expressed as:

$$\psi_e = \Psi_g - \Psi_v. \tag{2.60}$$

The cross-track error is defined as the lateral error between the control point and the goal point in the vehicle-body frame, as described in Equation 2.58. Given the vehicle's longitudinal velocity V_x, and the weight of the cross-track error k_c, the control law is represented as:

$$\delta = \psi_e + \arctan\left(\frac{k_c \cdot y_e}{V_x}\right), \tag{2.61}$$

For the Stanley method, it is also important to choose an appropriate control point. The control point in this method is the center of the front axle, as the heading position should align closely with the orientation of the path.

3

Outlier-Robust Path-Tracking Framework

An Outlier Robust Path-Tracking Framework is proposed to simulate the dynamics of the DCSC vehicle and evaluate its path-tracking performance under disturbance. This chapter introduces a force-moment-based RC vehicle model that captures the dynamics of the DCSC vehicle. Additionally, an adaptive outlier-robust filter is proposed to provide state estimation in the presence of observational disturbances and model discrepancies.

3.1. Overview

In Section 2.1, several challenges faced by the DCSC autonomous vehicle platform are outlined. In this chapter, a framework is proposed to simulate these challenges in a controlled environment and address them through software.

The proposed outlier-robust path-tracking control framework is a control pipeline designed for the DCSC autonomous vehicle platform. The framework includes an authentic RC vehicle simulator to narrow the gap between simulation and field testing, simulating the occurrence of sensor malfunctions and packet-loss issues. Additionally, an Adaptive Outlier-Robust Kalman Filter is developed to estimate vehicle states in real-time, even in the presence of observational disturbances and model discrepancies.

3.1.1. Control Framework Pipeline

As shown in Figure 3.1, the framework consists of four modules: Map Generator, Path-Tracking Controller, Simulator, and State Observer. Each module has distinct functions and is interconnected.

Map Generator

The Map Generator creates waypoints based on a designated path. The path is partitioned by a sampling distance Δs, ensuring that neighboring waypoints are a fixed distance apart. Each waypoint $w := [X_w, Y_w, \Psi_w]$ is defined by its position and orientation in the global frame. The module generates waypoints $w_{1:N}$ based on the path configuration, with the first waypoint w_0 serving as the initial state for the simulation.

Path-Tracking Controller

The Path-Tracking Controller drives the vehicle to follow the path. The algorithm used is a geometric path-tracking method, as introduced in Section 2.4. It assumes the vehicle operates at moderate speeds, where geometric methods effectively handle steering control with low computational cost.

As shown in Figure 3.1, the module takes waypoints $w_{1:N}$ from the Map Generator and state estimates $\hat{\mathbf{x}}_t$ from the State Observer as inputs. It identifies the goal point $w_{goal,t}$ using a look-ahead distance and computes the steering angle δ_t based on the difference between $w_{goal,t}$ and $\hat{\mathbf{x}}_t$. The throttle input ω_t is set to a constant value to maintain a steady velocity.

Since the Simulator module cannot directly use δ_t and ω_t as inputs, a mapping function is applied to convert them into RC commands. The steering angle δ_t is linearly mapped to $\delta_{rc,t}$, and the throttle input ω_t is linearly mapped to $\omega_{rc,t}$, based on actuator modeling.

Simulator

The Simulator models the vehicle's motion using a realistic vehicle dynamics model, as shown in Equation (3.8). Due to the highly nonlinear nature of the dynamics, an ordinary differential equation (ODE) solver [1] is

[1] Scipy:solve_ivp

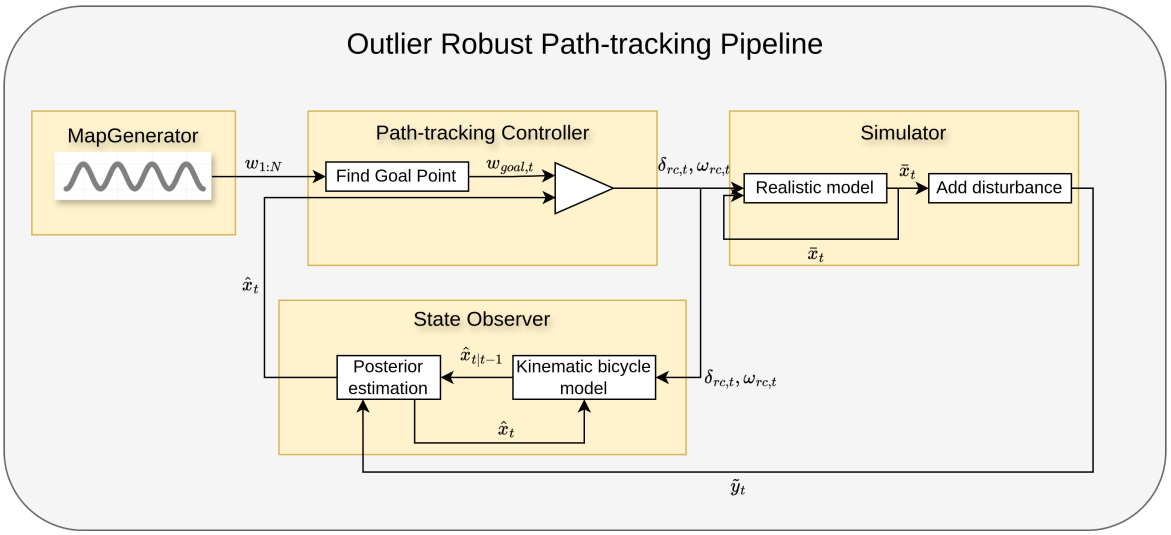


Figure 3.1: A flow chart of the outlier-robust path-tracking control pipeline: $w_{1:N}$ is all the waypoints of the designated path, $\bar{x}$ is the ground-truth state, $\tilde{y}_t$ is the noisy observation, $\hat{x}_t$ is the state estimation.

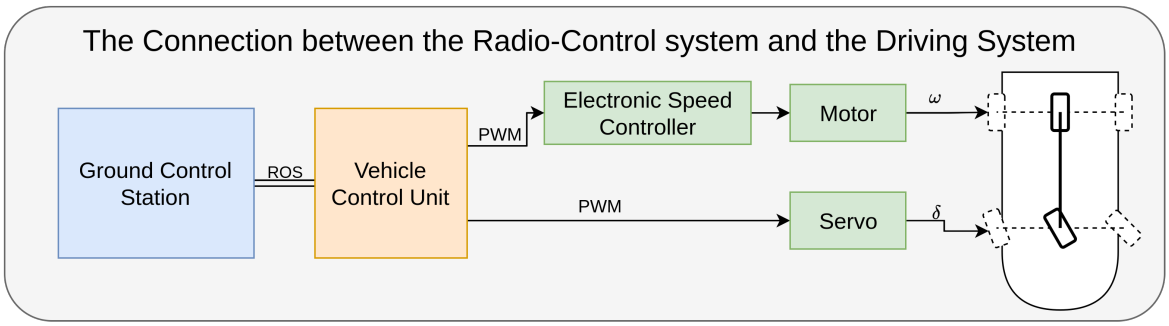


Figure 3.2: The connection between the Vehicle Control Unit and the Rover driving system

used to simulate vehicle motion within a specified sampling interval.

As shown in Figure 3.1, the module takes RC commands $\delta_{rc,t}$ and $\omega_{rc,t}$ as inputs. Using the previous ground-truth state $\bar{\mathbf{x}}_{t-1}$, it calculates the ground-truth state at the next sampling time $\bar{\mathbf{x}}_t$. The module outputs two components: the ground-truth state $\bar{\mathbf{x}}_t$, which is saved for the next iteration, and the noisy observation $\tilde{\mathbf{y}}_t$.

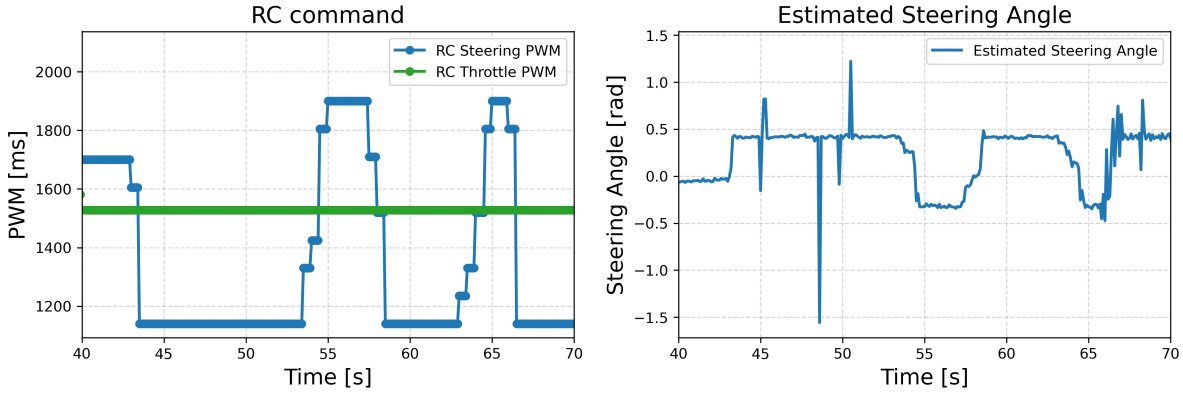
State Observer

The State Observer estimates the vehicle's state using prior information and observations. Kalman filter and its variations are implemented in this module. The kinematic bicycle model is used as the process model (prior knowledge of the system) due to its simplicity and generality.

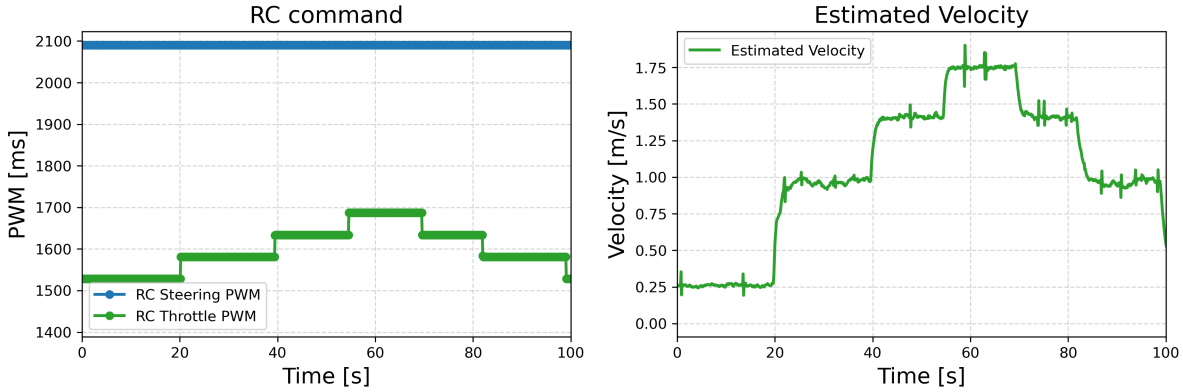
Given inputs $[\delta_{rc,t}, \omega_{rc,t}]$ and the previous state estimate $\hat{x}_{t-1}$, the module predicts $\hat{x}_{t|t-1}$ using the kinematic bicycle model. The prediction $\hat{x}_{t|t-1}$ and observation $\tilde{\mathbf{y}}$ are combined to produce the posterior state estimate $\hat{x}_t$. The module outputs the state estimate $\hat{x}_t$, which is recursively used for filtering and by the Path-Tracking Controller to compute control signals.

3.2. Radio-Control Vehicle Model

To simulate the motion of the Radio-Control (RC) vehicle in the DCSC autonomous vehicle platform, its dynamics must be studied. This section develops a RC vehicle model based on vehicle modeling principles and the RC system.



(a) The vehicle is controlled with a fixed throttle command and varying steering command. The left side shows the Radio-Control inputs, and the right side shows the estimated vehicle steering angle.



(b) The vehicle is controlled with a fixed steering command and varying throttle command. The left side shows the Radio-Control inputs, and the right side shows the estimated vehicle velocity.

Figure 3.3: The Radio-Control inputs and the estimated vehicle steering angle and velocity.

3.2.1. Radio-Control Input Modeling

The connection between the Radio-Control system and the driving system is illustrated in Figure 3.2. The Ground Control Station (GCS) communicates with the Vehicle Control Unit (VCU) through ROS. The GCS publishes RC commands to the VCU, which then relays them to the driving system to control the vehicle's steering angle and throttle.

For steering control, the steering command δ_{rc} actuates the servo to adjust the front wheels' steering angle δ. For throttle control, the throttle command ω_{rc} is processed by the Electronic Speed Controller and Motor to generate torque on the rear wheel, influencing the wheel's angular velocity ω. This section models the relationship between these inputs and outputs.

Analysis of Field Testing Data

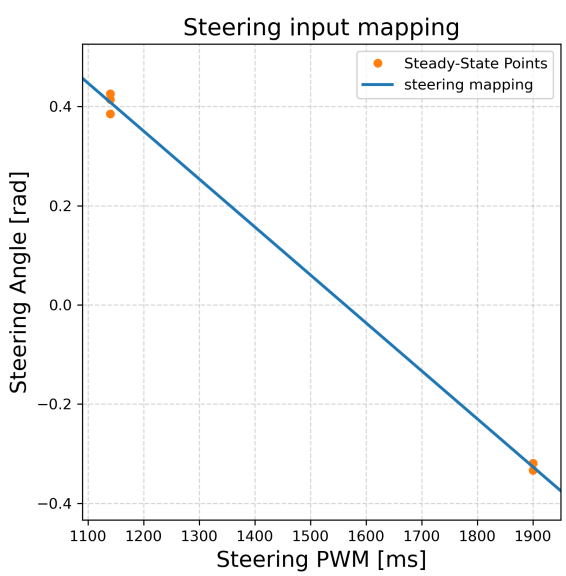
Prior information on the Radio-Control input dynamics is limited. To gain insight into these dynamics, we conducted field tests by actuating the vehicle in various ways and collected runtime data. The data is shown in Figure 3.3. The data collection and processing methods are detailed in Chapter 4.

The MoCap in the setup can only measure the position and orientation $[X_t, Y_t, \Psi_t]$ in the global frame. However, the vehicle's velocity and steering angle cannot be directly observed. Using these available measurements, $\hat{V}_{x,t}$ and $\hat{\delta}_t$ are estimated as follows:

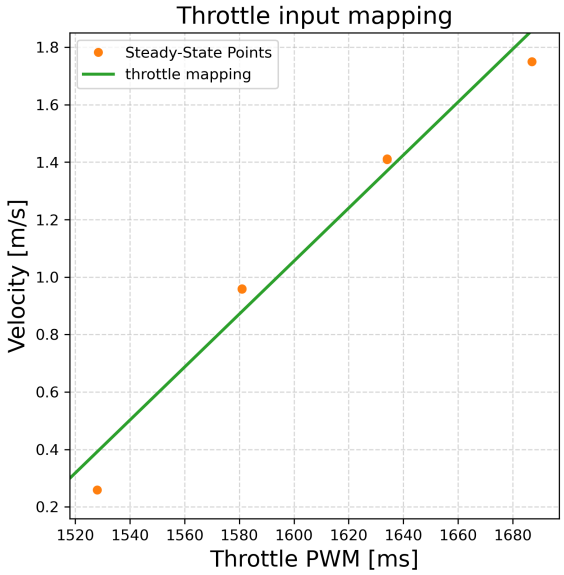
$$\hat{\delta}_t = \arctan \frac{(\Psi_t - \Psi_{t-1}) \cdot l_{wb}}{V_{x,t} \cdot \Delta t} \tag{3.1}$$

$$\hat{V}_{x,t} = \frac{\sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2}}{\Delta t} \tag{3.2}$$

As shown in Figure 3.3a, the throttle command σ_{rc} is held constant while the steering command δ_{rc} varies over time. This setup maintains a low, constant vehicle velocity, allowing us to focus on the relationship



(a) The relationship between the steady-state steering angle and the RC steering command.



(b) The relationship between the steady-state velocity and the RC throttle command.

Figure 3.4: The relationship between the steady-state vehicle states and the RC input commands.

between the steering command and the vehicle's steering angle. When the steering command changes, the steering angle adjusts accordingly, stabilizing after a brief transition period.

As shown in Figure 3.3b, the steering command δ_{rc} is held constant while the throttle command σ_{rc} varies over time. This setup keeps the wheel steering angle constant, enabling us to examine the relationship between the throttle command and the vehicle's velocity. The figure shows that when the throttle command changes, the velocity stabilizes after a transition period.

At this stage, we can assume that the steering angle δ is related to the steering command δ_{rc}, and the velocity V_x is related to the throttle command σ_{rc}. When the RC command changes, the corresponding vehicle state requires a transition period to converge to a new value.

Approximating the Radio-Control Input Dynamics with a First-Order System

Based on our assumption of the RC input dynamics, the process resembles the time response of a first-order system to a reference signal. As shown in Figure 3.3, the response exhibits almost no overshooting, similar to the step response of a first-order system [19].

The transfer function of a first-order system is:

$$\frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1} \tag{3.3}$$

where $Y(s)$ is the output in the Laplace domain, $U(s)$ is the input in the Laplace domain, K is the gain, and τ is the time constant. Applying the inverse Laplace transform, the dynamics in the time domain can be expressed as:

$$\dot{y} = -\frac{1}{\tau} \cdot y + \frac{K}{\tau} \cdot u \tag{3.4}$$

where y is the output in the time domain, and u is the input in the time domain. Replacing the output y with the wheel angular velocity ω and steering angle δ, and the input u with the RC inputs ω_{rc} and δ_{rc}, the approximated RC input dynamics are:

$$\begin{bmatrix} \dot{\omega} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_\omega} \cdot \omega + \frac{K_\omega}{\tau_\omega} \cdot (\omega_{rc} - \omega_{rc,0}) \\ -\frac{1}{\tau_\delta} \cdot \delta + \frac{K_\delta}{\tau_\delta} \cdot (\delta_{rc} - \delta_{rc,0}) \end{bmatrix} \tag{3.5}$$

where K_ω is the throttle mapping gain, K_δ is the steering mapping gain, $\omega_{rc,0}$ is the throttle mapping offset, $\delta_{rc,0}$ is the steering mapping offset, τ_ω is the throttle mapping time constant, and τ_δ is the steering mapping time constant.

From Figure 3.3, there is a relationship between the steady-state values of $\hat{\delta}$ and $\hat{V}_x$ with the RC inputs δ_{rc} and σ_{rc}. The values of the inputs and steady states at the same timestamp are collected, and a linear relationship is assumed, as shown in Figure 3.4:

$$\hat{V}_x = k_v \cdot \omega_{rc} + v_0 \tag{3.6}$$

$$\hat{\delta} = k_\delta \cdot \delta_{rc} + \omega_0 \tag{3.7}$$

k_v, k_δ are gains and v_0, ω_0 are offsets. The relationship between the longitudinal velocity V_x and the wheel angular velocity ω is discussed in Section 3.2.2.

3.2.2. Radio-Control Vehicle Dynamics

Building on the vehicle dynamics discussed in Section 2.2 and the RC input dynamics developed earlier, the combined dynamics form the RC vehicle dynamics developed in this thesis.

Assumptions

1. Planar Motion: The vehicle's motion is confined to a 2D plane.
2. Bicycle Model: The difference in orientation between the two front wheels is ignored.
3. First-Order Actuator Model: The input dynamics is modeled as first-order system.
4. Uniform Longitudinal Tire Stiffness: The front and rear wheels share the same longitudinal tire stiffness.
5. Uniform Lateral Tire Stiffness: The front and rear wheels share the same lateral tire stiffness.
6. No Front Wheel Slip Ratio: The front wheel's slip ratio is assumed to be zero.

Radio-Control Vehicle Dynamics

The RC vehicle dynamics are defined by the state vector $[X, Y, \Psi, V_x, V_y, \Omega, \omega, \delta] \in \mathbb{R}^8$. Here: $[X, Y, \Psi]$ represent the vehicle's position and orientation (yaw) in the global frame. $[V_x, V_y, \Omega]$ represent the longitudinal velocity, lateral velocity, and angular velocity (yaw rate) in the vehicle-body frame. $[\omega, \delta]$ represent the rear wheel's angular velocity and the front wheel's steering angle.

The kinematic relationship and vehicle body dynamics are expressed in Equation (2.13), and the RC input dynamics are expressed in Equation (3.5). Combining these, the RC vehicle dynamics are:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\Psi} \\ \dot{V}_x \\ \dot{V}_y \\ \dot{\Omega} \\ \dot{\omega} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} V_x \cdot \cos(\Psi) \\ V_x \cdot \sin(\Psi) \\ \Omega \\ \frac{1}{m} \cdot (C_\sigma \cdot \sigma_{xr} - C_\alpha \cdot \alpha_f \cdot \sin\delta - \mu \cdot V_x^2) + \Omega \cdot V_y \\ \frac{1}{m} \cdot (C_\alpha \cdot \alpha_f \cdot \cos\delta + C_\alpha \cdot \alpha_r) - \Omega \cdot V_x \\ \frac{1}{I_z} \cdot (l_f \cdot C_\alpha \cdot \alpha_f \cdot \cos\delta - l_r \cdot C_\alpha \cdot \alpha_r) \\ -\frac{1}{\tau_\omega} \cdot \omega + \frac{K_\omega}{\tau_\omega} \cdot (\omega_{rc} - \omega_{rc,0}) \\ -\frac{1}{\tau_\delta} \cdot \delta + \frac{K_\delta}{\tau_\delta} \cdot (\delta_{rc} - \delta_{rc,0}) \end{bmatrix} \tag{3.8}$$

All parameters are explained in previous sections.

Kinematic Bicycle Model Simplification

Based on the RC input mapping in Equations (3.6) and (3.7), the RC input commands δ_{rc} and σ_{rc} can serve as inputs to the system. Replacing the velocity and steering angle in Equation (2.1), the RC kinematic bicycle model is:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} (k_v \cdot \omega_{rc} + v_0) \cdot \cos(\Psi) \\ (k_v \cdot \omega_{rc} + v_0) \cdot \sin(\Psi) \\ \frac{(k_v \cdot \omega_{rc} + v_0) \cdot \tan(k_\delta \cdot \delta_{rc} + \delta_0)}{L_{wb}} \end{bmatrix} \tag{3.9}$$

This model extends the kinematic bicycle model with an additional input mapping layer. It can serve as the process model in observer design or as a basis for controller design.

Discussion on Wheel Angular Velocity and Longitudinal Velocity

In Section 3.2, the relationship between the RC input command and the vehicle velocity was observed during field testing, as shown in Figure 3.3b. However, in modeling the RC input dynamics, the RC throttle

command is related to the wheel angular velocity. To understand how changes in wheel angular velocity affect longitudinal velocity, consider Equation (3.8):

$$\dot{V}_x = \frac{1}{m} \cdot (C_\sigma \cdot \sigma_{xr} - C_\alpha \cdot \alpha_f \cdot \sin\delta - \mu \cdot V_x^2) + \Omega \cdot V_y \tag{3.10}$$

Assuming the vehicle is driving straight ($\delta = 0$) under nominal conditions, the steady-state longitudinal velocity V_x is determined by:

$$\mu V_x^2 = \frac{r_{eff} \cdot \omega - V_x}{r_{eff} \cdot \omega} \tag{3.11}$$

During acceleration, V_x cannot equal $r_{eff} \cdot \omega$ because part of the longitudinal force overcomes resistance. During turning ($\delta \neq 0$), the longitudinal force also partially overcomes lateral tire forces, further reducing V_x for the same throttle command.

In order to simplify the relationship between the RC command, wheel angular velocity, and longitudinal velocity, the RC kinematic model applies the direct relationship between the throttle command and longitudinal velocity (Equation (3.6)).

3.3. Adaptive Outlier-Robust Kalman Filter

The model discrepancy arises from the difference between the process model of the state observer and the simulation model of the simulator, while the observational disturbances result from sensor malfunctions. In this section, an Adaptive Outlier-Robust Kalman Filter is proposed, which can adaptively adjust to process noise and mitigate the influence of observational outliers.

3.3.1. Discrepancy between Simulation Model and Process Model

In the proposed framework, the simulator is based on the RC vehicle model (Equation (3.8)), while the process model assumed by the state observer is based on the kinematic bicycle model (Equation (3.9)). This model mismatch can result in significant process noise, indicating that the process model of the observer cannot accurately capture the dynamics of the system.

To accommodate the model discrepancy, two critical design parameters of the filter are the process noise covariance $\mathbf{Q} \in \mathbb{R}^{N_x \times N_x}$ and the observation noise covariance $\mathbf{R} \in \mathbb{R}^{N_y \times N_y}$. Intuitively, the matrix $\mathbf{Q}$ reflects the reliability of the process model, while the matrix $\mathbf{R}$ reflects the reliability of the observations.

However, for the outlier-robust Kalman filter, observations with large innovation (*i.e.*, those that deviate significantly from the prediction) are either down-weighted [2] or rejected [33]. In these cases, if the prior process noise covariance $\mathbf{Q}$ is set too small for the system, the outlier-robust Kalman filter will assign greater weight to the predictions, ignoring corrections from nominal observations (which contain information about the true state). Consequently, as the difference between the predictions and observations grows, the observations can be entirely ignored, and the posterior will align with the predictions, which is deviate significantly from the ground-true states.

As mentioned in Section 2.3.4, the innovation and the residual provide information about the reliability of the process model and the observations [3]. Inspired by Agamennoni *et al.*, we assume the process noise is drawn from a zero-mean Gaussian distribution with varying covariance. The process noise covariance is drawn from an Inverse-Wishart distribution, resulting in the following dynamic model:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q_t}), \quad \mathbf{Q}_t \sim \mathcal{W}^{-1}(\Lambda_0, \nu_0) \tag{3.12}$$

where the positive-definite matrix $\Lambda_0 \in \mathbb{R}^{N_x \times N_x}$ and the degree-of-freedom $\nu_0 \in (N_x - 1, \infty)$ are the parameters determining the shape of the Inverse-Wishart distribution, as mentioned in Section 2.3.5.

As shown in the upper part of Figure 3.5, in the proposed HMM, the latent dynamic state $\mathbf{x}_t$ is related to the previous dynamic state $\mathbf{x}_{t-1}$ and the process noise covariance $\mathbf{Q}_t$. In each filtering step, $\mathbf{Q}_t$ is drawn from an Inverse-Wishart distribution with the same shape $\mathcal{W}^{-1}(\Lambda_0, \nu_0)$. The predictive distribution can be expressed as:

$$\mathbf{x}_{t|t-1} \sim \mathcal{N}(f(\mathbf{x}_{t-1}), \mathbf{Q}_t) \tag{3.13}$$

$$\mathbf{Q}_t \sim \mathcal{W}^{-1}(\Lambda_0, \nu_0) \tag{3.14}$$

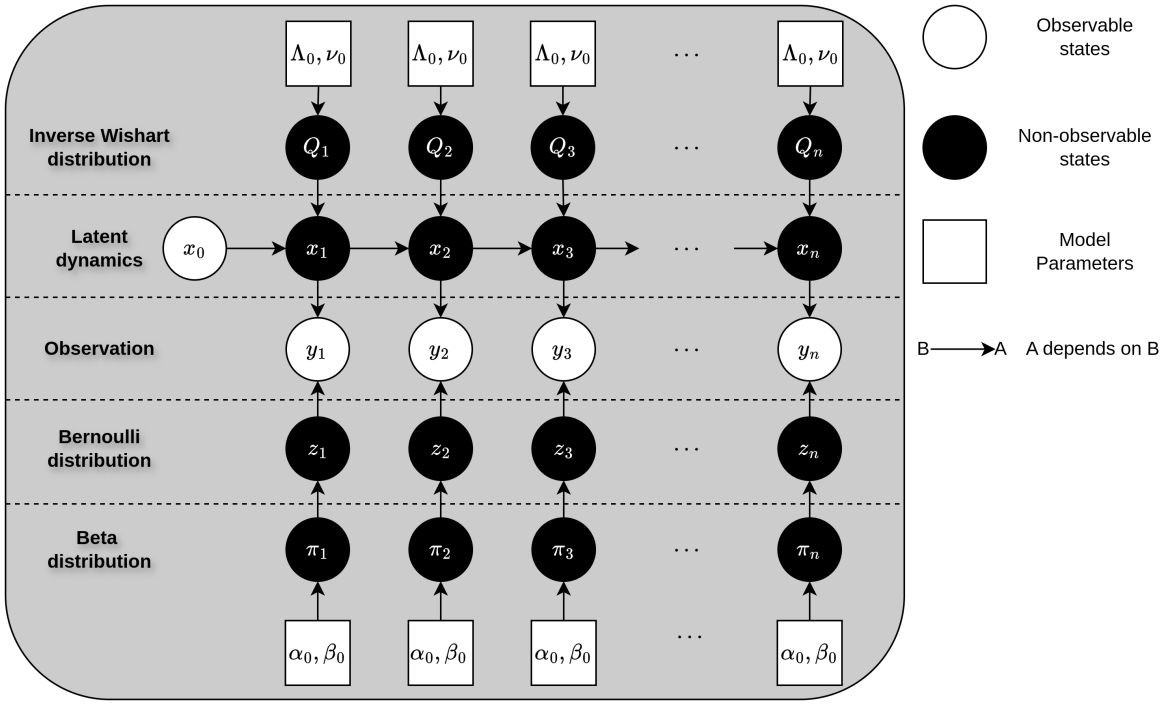


Figure 3.5: The graphic model of the system with adaptive process noise covariance and observational outlier detector

Once the observation $\mathbf{y}_t$ is obtained, and assuming the posterior distribution of the dynamic state $\mathbf{x}_t$ is Gaussian distributed with mean μ_t and covariance Σ_t, the updated parameters of the Inverse-Wishart distribution Λ_t and ν_t are given by the conjugate relationship between the Gaussian and Inverse-Wishart distributions, as described in Equations (2.45) and (2.44):

$$\begin{aligned} &\mathbf{Q}_t \sim \mathcal{W}^{-1}(\Lambda_t, \nu_t) \\ &\nu_t = \nu_0 + 1 \\ &\Lambda_t = \mathbf{B}_t + \nu_0 \cdot \Lambda_0 \\ &\mathbf{B}_t = \mathrm{E}[(\mathbf{y}_t - h(\mu_t)) \cdot (\mathbf{y}_t - h(\mu_t))^T] \end{aligned} \tag{3.15}$$

3.3.2. System Description

Within the proposed framework, different types of observational disturbances are introduced. In the filter design, the primary focus is on observational outliers. As introduced in Section 2.3.6, observational outliers are observations that are numerically distant from nominal observations, with a non-negligible probability of occurrence.

Combining Equation (3.12) and Equation (2.48), the graphical model of the HMM is shown in Figure 3.5. The dynamic model and the observation model can be expressed as:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q_t}), \quad \mathbf{Q}_t \sim \mathcal{W}^{-1}(\Lambda_0, \nu_0) \tag{3.16}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + z_t \cdot \mathbf{v}_t + (1 - z_t) \cdot \mathbf{c}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}), \quad \mathbf{c}_t \sim \mathcal{N}(0, \mathbf{R}^c) \tag{3.17}$$

$$z_t \sim \text{Beta-Bernoulli}(\pi_t; \alpha_0, \beta_0) \tag{3.18}$$

3.3.3. Adaptive Outlier-Robust Kalman Filtering Algorithm

To perform the posterior estimation of the latent states, the filtering problem and its solution are derived in this section.

Latent States and Observations

In the Hidden Markov Model (HMM), at each filtering time step, after obtaining the observation $\mathbf{y}_t$, there are four latent variables $[\mathbf{Q}_t, \mathbf{x}_t, z_t, \pi_t]$ that need to be estimated, and four parameters $\lambda_t, \nu_t, \alpha_t, \beta_t$ that need to be updated accordingly. Our goal is to perform posterior estimation of all these variables and parameters.

Joint Distribution, Predictive Distribution and Posterior Distribution

Given the dynamic model in Equation (3.16), the observation model in Equation (3.17), and the graphical model in Figure 3.5, applying the Bayesian rule and the Markovian property (2.14), the joint distribution of all the latent states $[\mathbf{Q}_t, \mathbf{x}_t, z_t, \pi_t]$ and the observations $\mathbf{y}_{1:t}$ as:

$$\begin{aligned} p(\mathbf{x}_t, \mathbf{Q}_t, z_t, \pi_t, \mathbf{y}_{1:t}) &= p(\mathbf{y}_t \mid \mathbf{x}_t, z_t) \cdot p(\mathbf{x}_t, \mathbf{Q}_t, z_t, \pi_t, \mathbf{y}_{1:t-1}) \\ &= p(\mathbf{y}_t \mid \mathbf{x}_t, z_t) \cdot p(\mathbf{x}_t \mid \mathbf{Q}_t) \cdot p(\mathbf{Q}_t, z_t, \pi_t, \mathbf{y}_{1:t-1}) \\ &= p(\mathbf{y}_t \mid \mathbf{x}_t, z_t) \cdot p(\mathbf{x}_t \mid \mathbf{Q}_t) \cdot p(\mathbf{Q}_t) \cdot p(\mathbf{y}_{1:t-1}) \cdot p(z_t, \pi_t) \\ &= p(\mathbf{y}_t \mid \mathbf{x}_t, z_t) \cdot p(\mathbf{x}_t \mid \mathbf{Q}_t) \cdot p(\mathbf{Q}_t) \cdot p(\mathbf{y}_{1:t-1}) \cdot p(z_t \mid \pi_t) \cdot p(\pi_t) \end{aligned} \tag{3.19}$$

The assumed distribution of each items in Equation (3.19) according to Equation (3.16) and (3.17) can be given as:

$$\mathbf{y}_t \mid \mathbf{x}_t, z_t \sim \mathcal{N}(h(\mathbf{x}_t), \mathbf{R})^{z_t} \tag{3.20}$$

$$\mathbf{x}_t \mid \mathbf{Q}_t \sim \mathcal{N}(f(\mathbf{x}_{t-1}), \mathbf{Q}_t) \tag{3.21}$$

$$\mathbf{Q}_t \sim \mathcal{W}^{-1}(\Lambda_0, \nu_0) \tag{3.22}$$

$$\mathbf{z}_t \mid \pi_t \sim \text{Bernoulli}(\pi_t) \tag{3.23}$$

$$\pi_t \sim \text{Beta}(\alpha_0, \beta_0) \tag{3.24}$$

Applying the Bayesian rule, the posterior distribution of the latent states can be expressed as:

$$p(\mathbf{x}_t, \mathbf{Q}_t, z_t, \pi_t \mid \mathbf{y}_{1:t}) = \frac{p(\mathbf{x}_t, \mathbf{Q}_t, z_t, \pi_t, \mathbf{y}_{1:t})}{p(\mathbf{y}_{1:t})} \tag{3.25}$$

There are two main challenges in explicitly calculate posterior (Equation (3.25)):

- The marginal distribution of $p(\mathbf{y}_{1:t})$ cannot be computed explicitly.
- From the graphic model in Figure 3.5, z_t depends on π_t, π_t depends on α_0, β_0.
- From the graphic model in Figure 3.5, $\mathbf{x}_t$ depends on $\mathbf{Q}_t$, $\mathbf{Q}_t$ depends on Λ_0, ν_0.

Therefore, the variational Bayesian can be applied to calculate the approximation of the posterior.

Variation Bayesian Approximation

Applying the logarithm to both sides of the posterior (Equation (3.25)) and using Bayes' rule, the log-likelihood can be expressed as:

$$\begin{aligned} \ln p(\mathbf{x}_t, \mathbf{Q}_t, z_t, \pi_t \mid \mathbf{y}_{1:t}) &= \ln p(\mathbf{x}_t, \mathbf{Q}_t, z_t, \pi_t, \mathbf{y}_{1:t}) - \ln(\mathbf{y}_{1:t}) \\ &= \ln p(\mathbf{y}_{1:t} \mid \mathbf{x}_t, \mathbf{Q}_t, z_t, \pi_t) + \ln p(\mathbf{x}_t, \mathbf{Q}_t, z_t, \pi_t) - \ln(\mathbf{y}_{1:t}) \end{aligned} \tag{3.26}$$

According to "Alternative View of the EM Algorithm" [31], assuming the likelihood $q(\cdot)$ is the approximated posterior of the latent states $\Theta := [\mathbf{Q}_t, \mathbf{x}_t, z_t, \pi_t]$, the first term on the right-hand side in Equation (3.26) is lower bounded by:

$$\ln p(\mathbf{y}_{1:t} \mid \Theta) \geq L(q(\Theta)) = \int q(\Theta) \cdot [\ln p(\Theta, \mathbf{y}_{1:t}) - \ln p(\Theta)] \cdot d\Theta \tag{3.27}$$

When the likelihood q is exactly the posterior distribution, the equality holds.

Assuming $q_x(\mathbf{x}_t)$ is the approximate posterior of $\mathbf{x}_t$, $q_Q(\mathbf{Q}_t)$ is the approximate posterior of $\mathbf{Q}_t$, $q_z(z_t)$ is the approximate posterior of z_t, and $q_\pi(\pi_t)$ is the approximate posterior of π_t, the Mean-field approximation [31] can be applied to express the approximate posterior of the latent states:

$$\begin{aligned} q(\Theta) &= q(\mathbf{x}_t, \mathbf{Q}_t, z_t, \pi_t) \\ &= q_x(\mathbf{x}_t) \cdot q_Q(\mathbf{Q}_t) \cdot q_z(z_t) \cdot q_\pi(\pi_t) \end{aligned} \tag{3.28}$$

Applying the approximated likelihood (Equation (3.28)) to the lower bound (Equation (3.27)), the approximated lower bound for any one of the latent states (here, for simplicity, the example latent state is $\mathbf{x}$) can be expressed as [31]:

$$L(q_x) = \int q_x \cdot [\ln p(\mathbf{y_{1:t}}, \Theta) \cdot \prod_{\theta \in \{Q, z, \pi\}} q_\theta \cdot \theta] \cdot d\mathbf{x} - \int q_x \cdot \ln(q_x) \cdot d\mathbf{x} - \cdots \tag{3.29}$$

where terms irrelevant to the state $\mathbf{x}$ are ignored. Equation (3.29) is maximized when:

$$\ln(q_x) = \int \ln p(\mathbf{y_{1:t}}, \Theta) \cdot \prod_{\theta \in \{Q,z,\pi\}} q_\theta \cdot \theta \tag{3.30}$$

Applying the same procedure for the other latent states and removing the non-related variables from the joint distribution, the optimal approximated posterior under variational Bayes can be expressed as:

$$\ln q_x^*(\mathbf{x}_t) \propto \mathrm{E}_{p(Q_t)p(z_t)p(\pi_t)} \left[\ln p(\mathbf{y}_t \mid \mathbf{x}_t, z_t) + \ln p(\mathbf{x}_t \mid \mathbf{Q}_t, \mathbf{x}_{t-1})\right] \tag{3.31}$$

$$\ln q_Q^*(\mathbf{Q}_t) \propto \mathrm{E}_{p(x_t)p(z_t)p(\pi_t)} \left[\ln p(\mathbf{x}_t \mid \mathbf{Q}_t, \mathbf{x}_{t-1}) + \ln p(\mathbf{Q}_t)\right] \tag{3.32}$$

$$\ln q_z^*(z_t) \propto \mathrm{E}_{p(x_t)p(Q_t)p(\pi_t)} \left[\ln p(\mathbf{y}_t \mid \mathbf{x}_t, z_t) + \ln p(z_t \mid \pi_t)\right] \tag{3.33}$$

$$\ln q_\pi^*(\pi_t) \propto \mathrm{E}_{p(x_t)p(Q_t)p(z_t)} \left[\ln p(z_t \mid \pi_t) + \ln p(\pi_t)\right] \tag{3.34}$$

The update rule of the latent states

1. From Equation (3.31), the approximated log-likelihood of $\mathbf{x}_t$ is proportional to the expectation over the combination of two Gaussian density functions. Since the product of Gaussian probability density functions is still Gaussian, $q_x^*(\mathbf{x}_t)$ has a Gaussian distribution density function:

$$q_x^*(\mathbf{x}_t) \propto \exp(-0.5 \cdot \mathrm{E}p[z_t] \cdot ||\mathbf{y}_t - h(\mathbf{x}_t)||^2_{\mathbf{R_t^{-1}}} - 0.5 \cdot ||\mathbf{x_t} - f(\mathbf{x}_{t|t-1})||^2_{\mathbf{Q_t^{-1}}}) \tag{3.35}$$

Given the mean μ_{t-1} and covariance Σ_{t-1} of the previous state $\mathbf{x}_{t-1}$, Equation 3.16 can be used to calculate the predictive distribution $\mathbf{x}_{t|t-1}$. Once the observation is obtained and the expectation of the Bernoulli variable $\mathrm{E}p[z_t]$ is estimated, the Kalman Filtering algorithm, as shown in Algorithm 3 (lines 25 to 36), can be applied to calculate the posterior of $\mathbf{x}_t$.

2. From Equation (3.32), the approximated likelihood of $\mathbf{Q}_t$ is proportional to the product of the probability density functions of a Gaussian distribution and an Inverse-Wishart distribution. According to the conjugate relationship between them, the posterior of $\mathbf{Q}_t$ and its shape parameters Λ_t, ν_t can be calculated as in Equation (3.15). The entire updating procedure is shown in Algorithm 3 (lines 37 to 41).

3. From Equation (3.33), the approximated likelihood to z_t is proportional to the product of the probability of a Gaussian distribution and a Bernoulli distribution, it can be written as:

$$q_z^*(z_t) \propto \exp(-0.5 \cdot z_t \cdot ||\mathbf{y}_t - h(\mathbf{x}_t)||^2_{\mathbf{R_t^{-1}}} + z_t \cdot \mathrm{E}[\ln p(\pi_t)] + (1 - z_t) \cdot \mathrm{E}[\ln p((1 - \pi_t)]), \quad z_t \in \{0, 1\} \tag{3.36}$$

here, π_t is Beta distributed with shape parameters α_t and β_t. Given the expression of the digamma function $\psi(\cdot)$ [27], the expectation of $\ln p(\pi_t)$ and $\ln p(1 - \pi_t)$ can be expressed as [27]:

$$\mathrm{E}[\ln p(\pi_t)] = \psi(\alpha_t) - \psi(\alpha_t + \beta_t) \tag{3.37}$$

$$\mathrm{E}[\ln p(1 - \pi_t)] = \psi(\beta_t) - \psi(\alpha_t + \beta_t) \tag{3.38}$$

Since the variable z_t can only be 0 or 1, the expectation of z_t can be calculated as:

$$\mathrm{E}[p(z_t)] = \frac{q_z^*(z_t = 1)}{q_z^*(z_t = 0) + q_z^*(z_t = 1)} \tag{3.39}$$

4. From Equation (3.34), the approximated posterior of the Beta variable is the product of the probability density functions of a Beta distribution and a Bernoulli distribution. Using the conjugate relationship between them, the posterior can be calculated as in Equations (2.52), (2.53), and (2.54). The entire update process for both the Beta variable π_t and the Bernoulli variable z_t is shown in Algorithm 3 (lines 14 to 24).

Filtering process

Combining the process noise covariance estimation with the Extended Kalman Filter Bernoulli [33], the algorithm of Adaptive Extended Kalman Filter Bernoulli is given in Algorithm 3. For comparison, the process noise covariance estimation process is also added to the Extended Kalman Filter Inverse-Wishart [2] for comparison, the resulting Adaptive Extended Kalman Filter Inverse-Wishart is given in Algorithm 4.

Algorithm 3 Adaptive Extended Kalman Filter with Beta-Bernoulli Distribution:

Input: μ_{t-1}, Σ_{t-1}, $f(\cdot)$, $\mathbf{F}_x(\cdot)$, $h(\cdot)$, $\mathbf{H}_x(\cdot)$, $\mathbf{y}_t$, α_0, β_0, Λ_0, threshold
Output: μ_t, Σ_t
Initialize
$\Lambda_t \leftarrow \Lambda_0$
$\alpha_t \leftarrow \alpha_0$
$\beta_t \leftarrow \beta_0$
$\pi_t \leftarrow 1$
$\mu_t \leftarrow \mu_{t|t-1}$
$\Sigma_t \leftarrow \Sigma_{t|t-1}$
repeat
  Prediction
  $\mu_{t|t-1} \leftarrow f(\mu_{t-1})$
  $\Sigma_{t|t-1} \leftarrow \mathbf{F}_x(\mu_{t-1}) \cdot \Sigma_{t-1} \cdot \mathbf{F}_x(\mu_{t-1})^T + \Lambda_t$
  Update Bernoulli variables
  $\epsilon_t \leftarrow \mathbf{y}_t - h(\mu_t)$
  $\mathbf{B}_t \leftarrow E[\epsilon_t \cdot \epsilon_t^T]$
  $\mathrm{E}_{p(x_t)p(\pi_t)}[\ln p(\pi_t)] \leftarrow \psi(\alpha_t) - \psi(\alpha_t + \beta_t)$
  $\mathrm{E}_{p(x_t)p(\pi_t)}[\ln p(1-\pi_t)] \leftarrow \psi(\beta_t) - \psi(\alpha_t + \beta_t)$
  $q_z^*(z_t = 0) \leftarrow \exp\{-0.5 \cdot \mathrm{tr}[\mathbf{B}_t \mathbf{R}^{-1}] + \mathrm{E}_{p(x_t)p(\pi_t)}[\ln p(\pi_t)]\}$
  $q_z^*(z_t = 1) \leftarrow \exp\{\mathrm{E}_{p(x_t)p(\pi_t)}[\ln p(1-\pi_t)]\}$
  $\mathrm{E}_{p(\mathbf{x}_t)p(\pi_t)}[p(z_t)] \leftarrow \frac{q_z^*(z_t=1)}{q_z^*(z_t=0)+q_z^*(z_t=1)}$
  Update Beta variables
  $\alpha_t \leftarrow \alpha_0 + \mathrm{E}_{p(\mathbf{x}_t)p(\pi_t)}[p(z_t)]$
  $\beta_t \leftarrow \beta_0 + 1 - \mathrm{E}_{p(\mathbf{x}_t)p(\pi_t)}[p(z_t)]$
  Update state mean and covariance
  if $z_t \leq$ threshold **then**
    Outlier update
    $\mu_t \leftarrow \mu_{t|t-1}$
    $\Sigma_t \leftarrow \Sigma_{t|t-1}$
  else
    Inlier update
    $\mathbf{S}_t \leftarrow \mathbf{H}_x(\mu_{t|t-1}) \cdot \Sigma_{t|t-1} \cdot \mathbf{H}_x(\mu_{t|t-1})^T + \mathbf{R}$
    $\mathbf{K}_t \leftarrow \Sigma_{t|t-1} \cdot \mathbf{H}_x(\mu_{t|t-1})^T \cdot \mathbf{S}_t^{-1}$
    $\mu_t \leftarrow \mu_{t|t-1} + \mathbf{K}_t \cdot \left[\mathbf{y}_t - h(\mu_{t|t-1})\right]$
    $\Sigma_t \leftarrow \Sigma_{t|t-1} - \mathbf{K}_t \cdot \mathbf{S}_t \cdot \mathbf{K}_t^T$
  end if
  Update process noise covariance
  $\epsilon_t \leftarrow \mathbf{y}_t - h(\mu_t)$
  $\mathbf{B}_t \leftarrow \mathrm{E}[\epsilon_t \cdot \epsilon_t^T]$
  $\nu_t \leftarrow \nu_0 + 1$
  $\Lambda_t \leftarrow \frac{\mathbf{B}_t + \nu_0 \cdot \Lambda_0}{\nu_0 + 1}$
until Convergence of μ_t and Σ_t
Return: μ_t, Σ_t

4

System Parameters Identification

A Radio-Control vehicle model is developed in Section 3.2, however, the parameters of the vehicle model still needed to be specified. In order to improve the fidelity of the model, the gray-box system identification [12] is carried out to determine the parameters of the model in this section.

4.1. Vehicle data collection during the field testing

As introduced in Section 2.1, the Motion Capture System (MoCap) tracks the vehicle's position and orientation in real-time. The MoCap communicates these observations to the Ground Control Station (GCS), while the Radio-Control (RC) system sends throttle and steering commands through the GCS. Upon receiving state and input data, the GCS assigns timestamps to them. This section describes the collected vehicle state data and the necessary pre-processing steps.

4.1.1. Dataset

The dataset $[X, Y, \Psi, \delta_{rc}, \sigma_{rc}, t]$ can be divided into three key components: vehicle state data, RC input data, and timestamp.

- Vehicle State Data: The MoCap reconstructs the vehicle's state in 3D space within the "MOCAP" coordinate frame, including its position and orientation. Since the vehicle operates on a 2D plane, the state data is transformed from the MOCAP frame to the global frame. In the global frame, only the (X-Y) position $[X, Y] \in \mathbb{R}^2$ and the yaw angle $\Psi \in \mathbb{R}$ are relevant.
- RC Input Command: The RC transmitter or the GCS sent RC input commands to the vehicle, including the throttle command σ_{rc} and the steering command δ_{rc}. For communication with the vehicle's driving system, the RC input commands are represented as Pulse Width Modulation (PWM) values.
- Timestamp: When the GCS receives the data, it assigned a timestamp to each entry.

4.1.2. Dataset Pre-Processing

Due to sensor and communication malfunctions, the collected data contained observation noise, packet loss, and incorrect timestamps. To mitigate the impact of these issues, data smoothing and interpolation methods are applied to remove noise, eliminate outliers, and fill in missing data.

To address noise and outliers, a moving-average-based smoothing method is employed. Moving-average methods apply a sliding window over a subset of data and compute statistics for that subset. For example, the "Gaussian" moving average assumes the subset follows a Gaussian distribution and weights the noise accordingly. Several smoothing methods are compared, including "Moving Median", "Gaussian", "Quadratic Regression", and "Robust Quadratic Regression". As shown in Figure 4.1, "Quadratic Regression" and "Robust Quadratic Regression" are significantly affected by outliers, whereas "Moving Median" and "Gaussian" produced more reliable results by effectively smoothing outliers. Based on these findings, the "Moving Median" smoother[1] is selected for data smoothing.

[1] `smoothdata`: `https://nl.mathworks.com/help/matlab/ref/smoothdata.html`

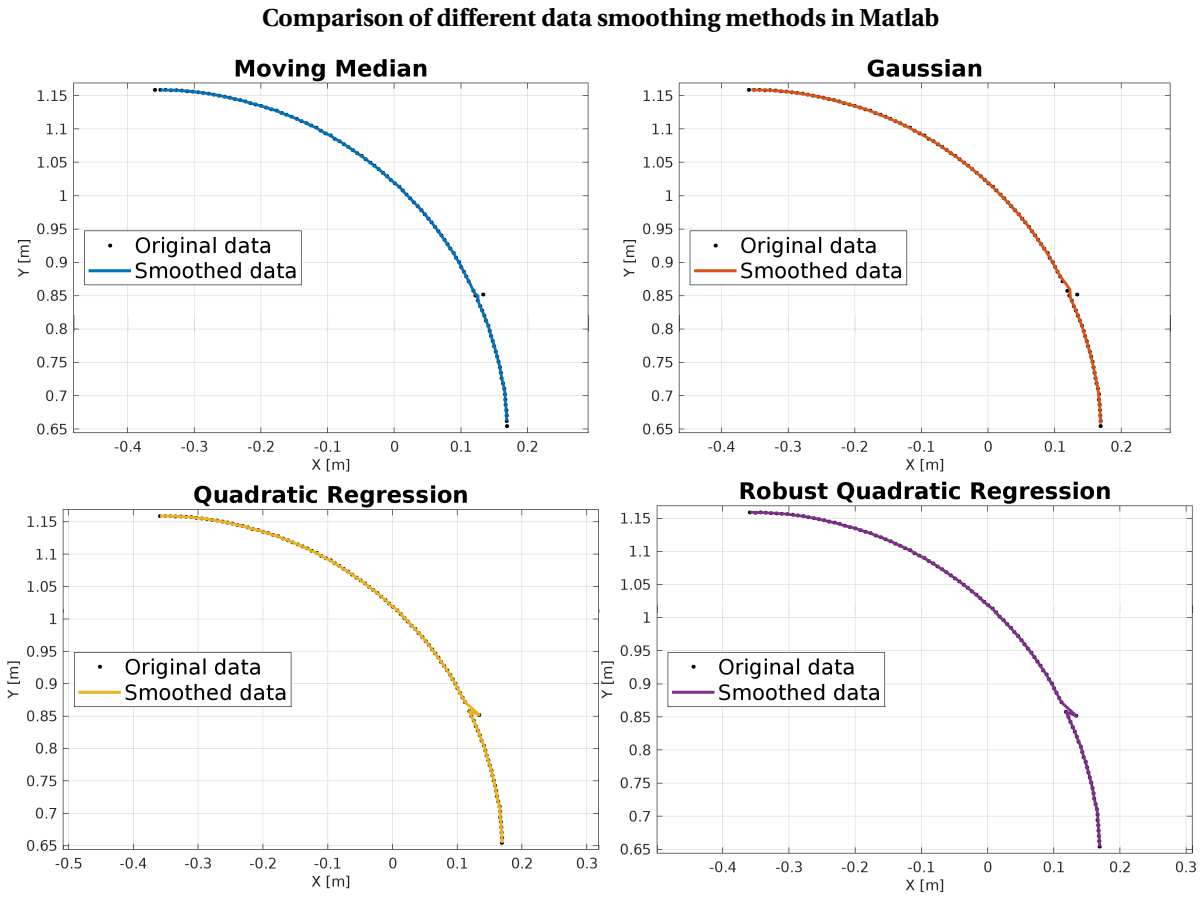


Figure 4.1: A comparison of different data smoothing methods on the collected runtime data.

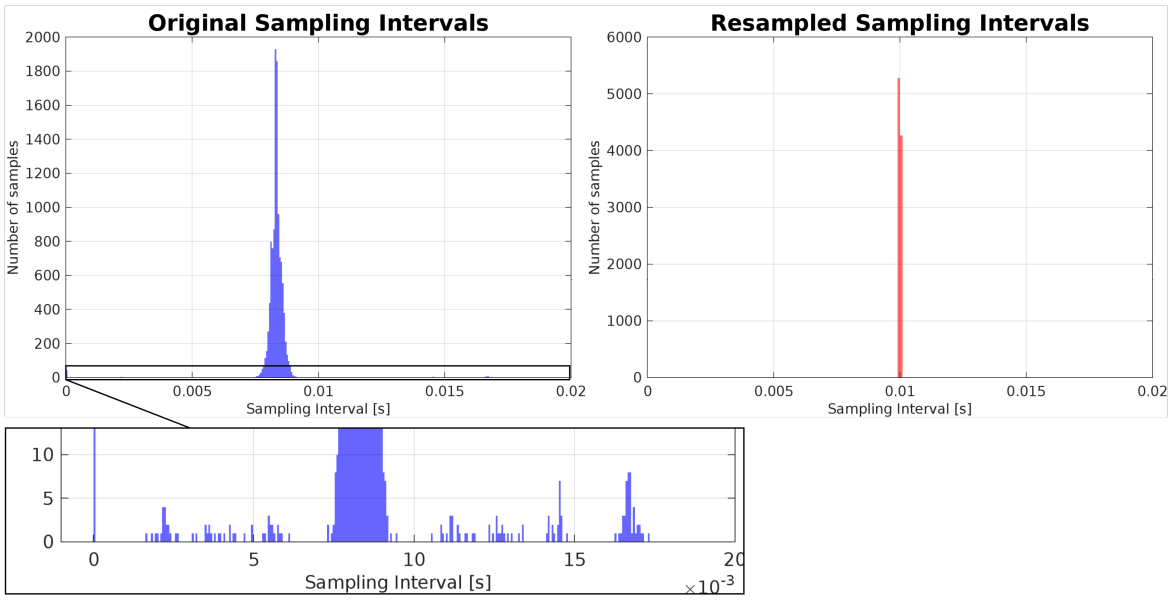


Figure 4.2: The distribution of sampling interval Δt before and after the resampling

Packet loss resulted in missing data points and incorrect timestamps assigned by the GCS. To address this, a linear interpolation method[2] is applied. This method approximates missing data points by interpolating between adjacent data points. Figure 4.2 compares the sampling intervals before and after resampling. Initially, the original sampling intervals varied between 0 [s] and 0.18 [s] due to packet loss. Such variability and missing data points can lead to numerical issues during steering angle and velocity estimation. The resampling process ensured a fixed sampling interval, which is essential for data analysis.

Before interpolation, the orientation data Ψ in the global frame is unwrapped. Typically, angle data are defined within the range $[-\pi, \pi]$. However, this can cause numerical issues during interpolation. To avoid this, the data is unwrapped to the range $(-\infty, \infty)$, ensuring that angles are not normalized to $[-\pi, \pi]$ during processing.

4.2. Apply Gray-box System Identification

The parameters of the vehicle model are identified using gray-box system identification [12]. To apply this method, the following components must be selected:

- The model requiring identification.
- A list of parameters and their initial guesses.
- The system identification data.
- The simulation method: The RC vehicle system is highly nonlinear, so continuous-time simulation is applied.
- The optimization algorithm: The Trust-Region Reflective Newton method of nonlinear least-squares [12] is used to solve the nonlinear optimization problem.

4.2.1. Model and Initial Parameter Estimation

The RC vehicle model introduced in Section 3.2 is used for system identification. This model is developed to capture the dynamics of the DCSC autonomous vehicle, and system identification is employed to increase its fidelity.

A list of parameters requiring identification, along with their initial guesses, is provided in Table 4.1. The initial guesses are either derived from the vehicle manual or empirically estimated. If the "Range" column in the table contains only one value, the parameter is deterministic and does not require estimation.

Table 4.1: The identified parameters of the RC vehicle model with the steering ID data

Parameter	Meaning	Range	Initial Guess	Result
m	Vehicle mass [kg]	[0,4]	3.32962	3.70492
C_α	Lateral tire stiffness	[0, + inf]	27.7163	5.10405
C_σ	Longitudinal tire stiffness	[0, + inf]	4.29394	2.3854
l_f	Distance from front axle to CoG [m]	02214	0.2214	0.2214
r	Rear wheel radius [m]	0.056	0.056	0.056
μ	Resistance coefficient	[0,10]	10	9.96977
K_δ	Steering coefficient	[0, + inf]	0.000835728	0.000812959
K_σ	Throttle coefficient	[0, + inf]	0.1416	0.164756
τ_δ	Steering Tau [s]	[0, + inf]	0.00249688	0.00128383
τ_σ	Throttle Tau [s]	[0, + inf]	1.81983	0.928954

4.2.2. System Identification Data

For system identification, a dataset containing input-output pairs must be provided. The system is activated by the input data, and the output data corresponds to the vehicle's state. A dataset with a fixed throttle command and varying steering command is used, as shown in Figure 4.3. The throttle command, shown in the bottom left, is fixed, indicating that the longitudinal velocity of the vehicle does not change significantly during operation. The steering command, shown in the bottom right, varies over time to drive the vehicle around

[2] `retime`: `https://nl.mathworks.com/help/matlab/ref/timetable.retime.html`

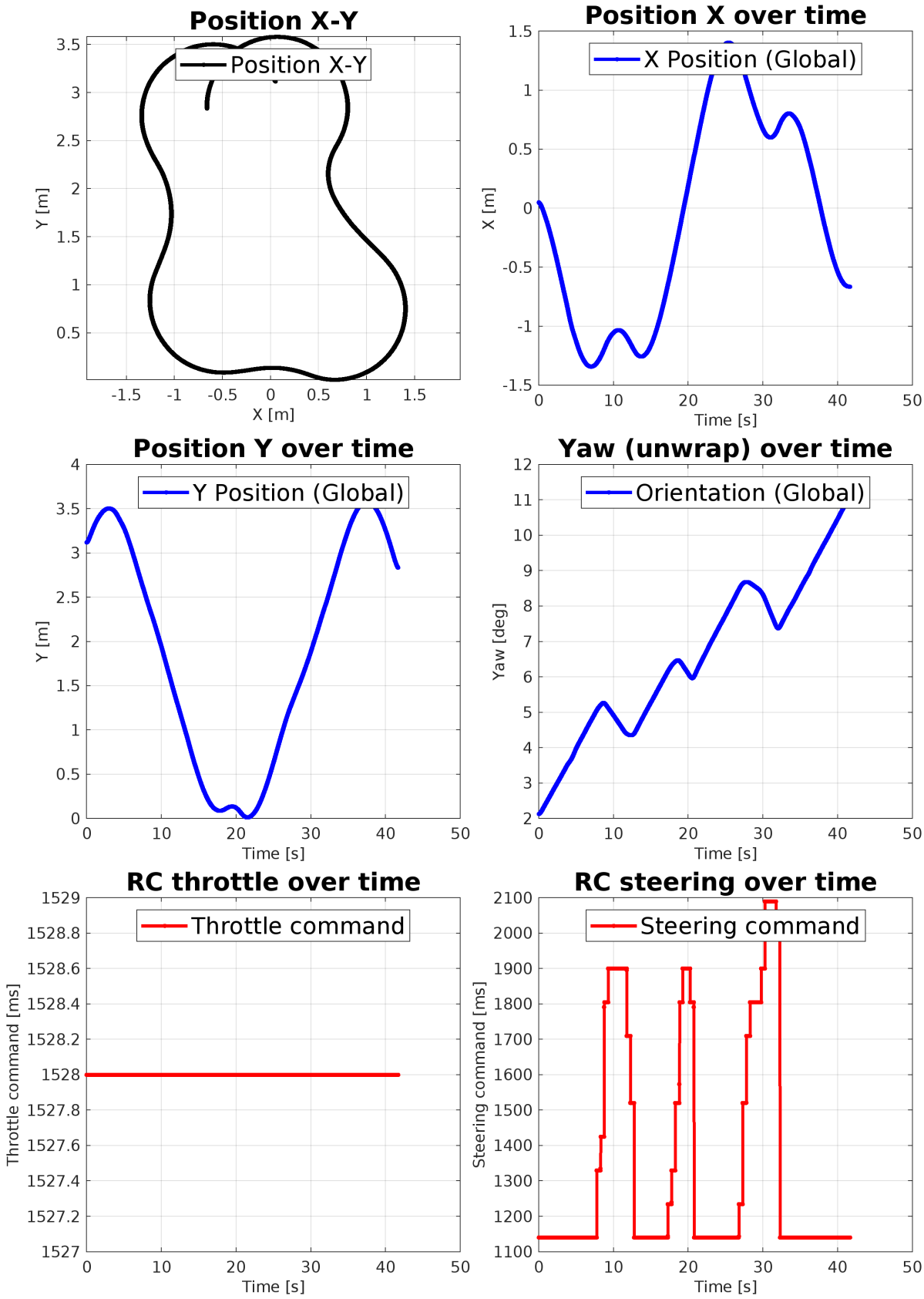


Figure 4.3: Data collected during the field testing, the vehicle is driven with fix throttle command and changing steering command.

the testing field in the lab. The state of the vehicle is shown in the top and middle plots. The orientation of the vehicle, shown in the middle right, is unwrapped to avoid numerical issues. Since the path-tracking application emphasizes steering control and steering dynamics, this dataset is selected for identification.

4.3. Result of identification

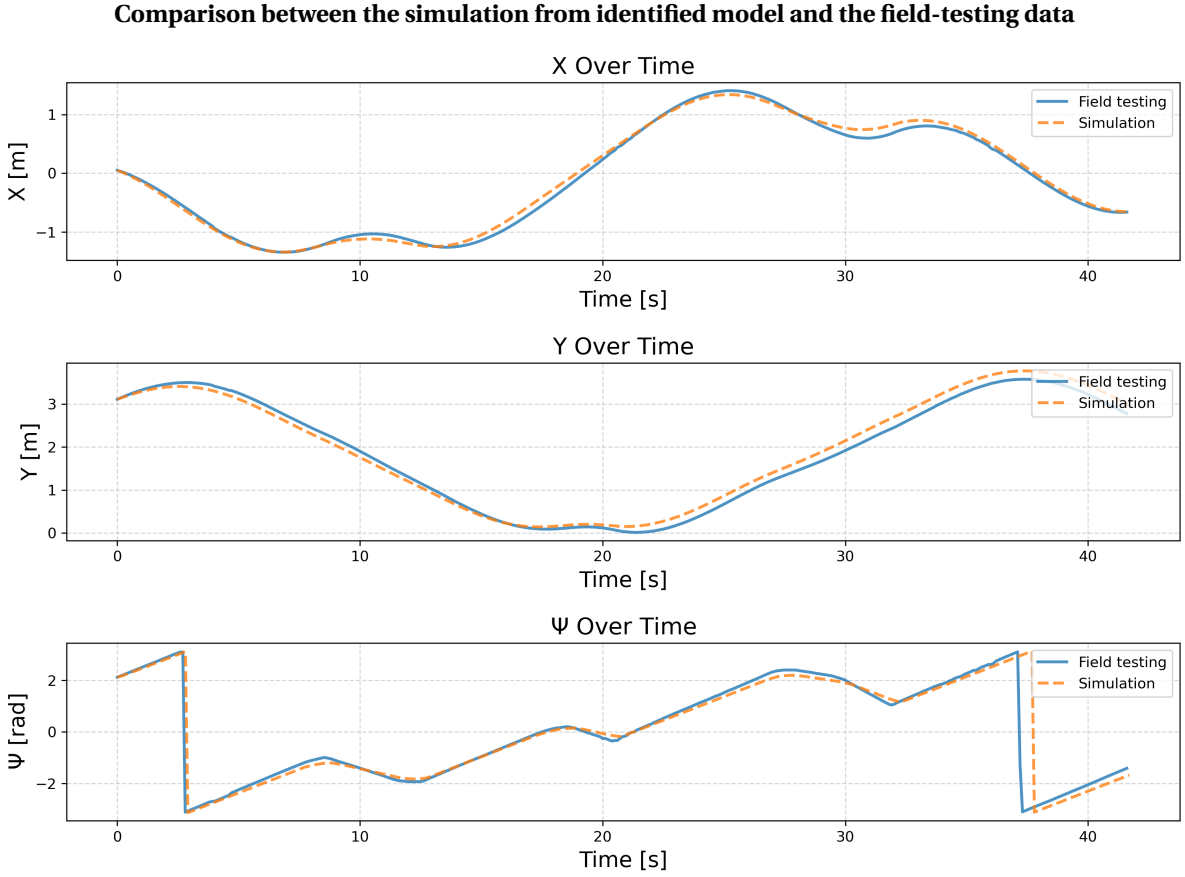


Figure 4.4: Comparison between the simulation and the field testing data on each state, the simulation is based on the RC vehicle model with identified parameters.

Based on the system identification setup described above, the identification results are presented in Table 4.1. Using the identified parameters, an open-loop simulation is performed with the RC vehicle model. The input for the simulation is the RC command shown in Figure 4.3, and the output of the identified model is compared with the runtime data collected during field testing. The goal of this comparison is to qualitatively evaluate how well the identified model captures the motion of the DCSC autonomous vehicle.

To perform the open-loop simulation with the identified RC vehicle model, an ordinary differential equation (ODE) solver is employed to handle the model's nonlinearity. The simulation is conducted in periods of 0.1 [s]. For example, at time step k, the state $\mathbf{x}_k$ served as the initial state, and the input $\mathbf{u}_k$ is applied. A simulation with a duration of 0.1 [s] is then performed to predict the state at time step $k+1$. If the RC vehicle dynamics is discretize by the Euler method, it would introduce numerical issues and is therefore avoided.

As shown in Figure 4.4, the position and orientation from the simulation are compared with the field testing data. The results indicate that the three states did not fully match, suggesting limitations in the current model. Figure 4.5 compares the trajectories, there are deviations between the open-loop simulation and the field testing data. While the identified model captured some dynamics of the real system, certain discrepancies remain.

4.3.1. Discussion on the identified model

After analysis, several factors may account for the observed deviations:

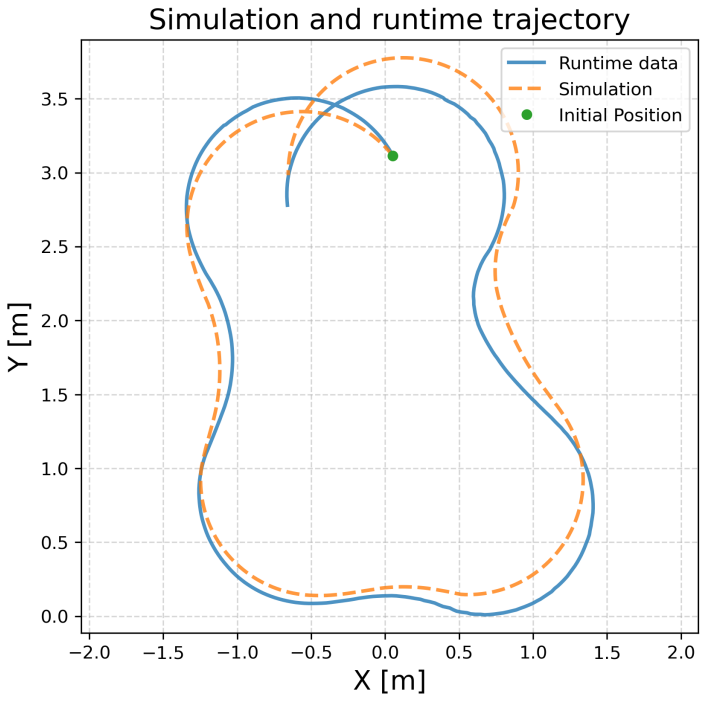


Figure 4.5: Comparison between the simulation and the field testing data on the (X-Y) driving trajectory, the simulation is based on the RC vehicle model with identified parameters.

1. Since latent states are not fully observable, only the position and orientation can be measured. Other states must be estimated from observations, which may introduce errors.

2. The relationship between the steering command and the steering angle is nonlinear, which the model may not fully capture.

3. The RC vehicle model is simplified and may not account for certain dynamics.

4. The dataset used for system identification may be inadequate, additional runtime data from field testing could improve the identification results.

5

Adaptive Outlier Robust Kalman Filter

The Kalman filter and its variants are designed to estimate the latent state of a system with the prior knowledge of the system and the observations. The proposed Adaptive Outlier-Robust Kalman Filter is specifically developed for systems with observational outliers and model discrepancies. To evaluate the performance of the proposed observer, synthetic datasets incorporating different types of disturbances are generated. A quantitative evaluation is conducted to compare the proposed method with several existing approaches.

5.1. Overview

This section provides an overview of the experiments, where different observers are applied to various synthetic datasets to evaluate and compare their estimation accuracy and efficiency.

5.1.1. Observers

The comparison is made between five different Kalman filtering algorithms including the basic method (EKF), the non-adaptive outlier-robust methods (EKF-B, EKF-IW) and the adaptive outlier-robust methods (EKF-B-adp, EKF-IW-adp):

- The Extended Kalman Filter (EKF),
- The Extended Kalman Filter Inverse-Wishart (EKF-IW, [2], Algorithm 1),
- The Extended Kalman Filter Bernoulli (EKF-B, [33], Algorithm 2),
- The proposed Adaptive Extended Kalman Filter Bernoulli (EKF-B-adp, Algorithm 3),
- The proposed Adaptive Extended Kalman Filter Inverse-Wishart (EKF-IW-adp, Algorithm 4).

5.1.2. Datasets

The performance of these methods is evaluated on synthetic datasets. Two different synthetic datasets are used to compare the observers' performance, either with or without model discrepancy. For each dataset, three types of observational disturbances are added to the observations.

- Dataset without discrepancy: the dataset generated using the kinematic bicycle model (Equation (2.1)) with Gaussian noise, observational outliers and packet-loss.
- Dataset with discrepancy: The dataset generated using the identified model (Equation (3.8)) with Gaussian noise, observational outliers and packet-loss.

5.1.3. Evaluation Metrics

The Root-Mean Square Error (RMSE) and Maximum Error (EMax) are used to quantify the difference between the estimated and ground-truth values in the datasets. The RMSE reflects the average performance, while the EMax reflects the worst-case performance. The computation time per step is used to quantify the operational efficiency of the algorithm in each time step.

Given the ground-truth $\bar{\mathbf{x}}_t$ and the estimate $\hat{\mathbf{x}}_t$, the RMSE and EMax are defined as:

$$\text{RMSE} := \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\bar{\mathbf{x}}_t - \hat{\mathbf{x}}_t)^T(\bar{\mathbf{x}}_t - \hat{\mathbf{x}}_t)} \tag{5.1}$$

$$\text{EMax} := \max_{1\le t\le N}\sqrt{(\bar{\mathbf{x}}_t - \hat{\mathbf{x}}_t)^T(\bar{\mathbf{x}}_t - \hat{\mathbf{x}}_t)} \tag{5.2}$$

5.2. Modeling Observational Disturbance

Synthetic datasets are an effective way to evaluate the performance of observers, as they allow full control over the disturbances introduced into the system and provide ground-truth data. To generate these datasets, several models are developed to simulate system dynamics and observations under different types of disturbances. Duran-Martin *et al.* [7] used a linear model to generate observations with observational noise and outliers. In this thesis, the linear model is extended to a nonlinear model, three types of observational disturbances are added to the observations:

- Additive observational Gaussian noise.
- Observational outliers.
- Packet loss.

5.2.1. Additive Observational Gaussian Noise

Observational noise refers to signals that perturb measurements, introducing uncertainty into the system. This noise is commonly assumed to follow a Gaussian distribution due to its generality, as justified by the Central Limit Theorem [16]. Given the latent state $\mathbf{x}_t \in \mathbb{R}^{N_x}$, the observation function $h(\cdot)$, and the noise $\mathbf{v}_t \in \mathbb{R}^{N_y}$, the observation $\mathbf{y}_t$ is defined as:

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}).$$

5.2.2. Additive Observational Outliers

Observational outliers are data points that deviate significantly from the overall distribution, appearing numerically distant from other observations [10]. In the synthetic dataset, outliers are drawn from a Gaussian distribution with a large covariance $\mathbf{R}^c$. A Bernoulli variable z_t determines whether an outlier occurs ($z_t = 0$). The probability of outliers is controlled by π, similar to the model introduced in Section 2.3.6:

$$\begin{aligned}&\mathbf{y}_t = h(\mathbf{x}_t) + z_t \cdot \mathbf{v}_t + (1 - z_t)\cdot \mathbf{c}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}), \quad \mathbf{c}_t \sim \mathcal{N}(0, \mathbf{R}^c),\\ &z_t \sim \text{Bernoulli}(\pi), \quad z_t \in \{0, 1\}.\end{aligned}$$

5.2.3. Packet Loss

Packet loss occurs when data packets fail to reach their destination. In ROS-TCP, during packet loss, no data is published, and the failed observations are stacked for retransmission. Once the connection is re-established, the stacked data is resent. However, since the data is timestamped only at the time of collection, delayed observations are assigned incorrect timestamps. Additionally, no data is collected during the loss period. An illustration of packet loss is provided in Figure 5.1.

In order to model the packet-loss, a Finite State Machine (FSM) is applied, it includes 3 states, and the state transition matrix is given in Table 5.1:

$$\text{State} := \{\text{Connect}, \text{Reconnect}, \text{Loss}\}$$

When the connection is corrupted, the observation at the time step when packet loss occurs is stored and resent upon reconnection. During the loss period (lasting k time steps), observations are replaced with NaN. At the exact moment of reconnection, the stacked observation is sent. Once the connection is restored, observations return to their nominal state. The observations under different states are summarized as follows:

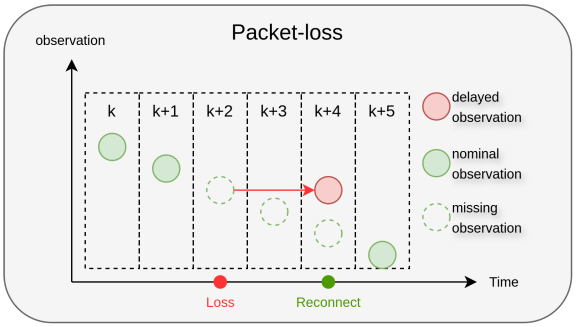


Figure 5.1: An illustration of packet-loss, during the loss period, there is no observations collected, the delayed observation is held until the system reconnects.

Table 5.1: The state-transition table of the packet loss

Current State	Condition	Probability	Next State	Action
Connect		Probability p	Connect	
Connect		Probability $1-p$	Loss	$\mathbf{y}_{corrupt} = \mathbf{y}_t$, $t_{corrupt} = t$
Loss	$t < t_{corrupt} + k$	1	Loss	
Loss	$t = t_{corrupt} + k$	1	Reconnect	$\mathbf{y}_t = \mathbf{y}_{corrupt}$
Reconnect	$t > t_{corrupt} + k$	1	Connect	

$$\mathbf{y}_t = \begin{cases} h(\mathbf{x}_t) + \mathbf{v}_t, & \text{if Connect} \\ \text{NaN}, & \text{if Loss} \\ h(\mathbf{x}_{t-k}) + \mathbf{v}_{t-k}, & \text{if Reconnect} \end{cases}$$

5.3. Without Model Discrepancy

Assuming there is no discrepancy between the process model of the observers and the model used by the simulator, both the process model and the simulation model are based on the kinematic bicycle model (Equation (2.1)). In this experiment, the performance of the observers is evaluated under different observational disturbances but without model discrepancy.

- The process model for the observers is based on the kinematic bicycle model.
- The datasets are generated using the kinematic bicycle model with Gaussian noise, observational outliers, and packet loss.

5.3.1. Dataset

The datasets are generated based on the dynamics of the kinematic bicycle model, as described in Equation (2.1). The simulation parameters are summarized in Table 5.2.

As shown in Figure 5.2, one trial of the entire dataset is illustrated. Within each trial, the longitudinal velocity V_x and the steering angle δ are both fixed, resulting in a circular trajectory for the true latent state. At the same time, three types of observational disturbances are added:

- The left subplot shows observations under Gaussian noise, where the observations are nominal.
- The middle subplot shows observations under outliers, where some observations deviate significantly from the true latent states (observational outliers).
- The right subplot shows observations under packet loss, where some periods have no observations.

However, as shown in Figure B.1, across 100 different trials, all trials share the same longitudinal velocity, while the steering angles are randomly selected within $[-0.3, 0.3]$ [rad]. As a result, the circular trajectories have different radius.

Table 5.2: The parameters for generating the synthetic data using kinematic bicycle model

Property	Value
Wheel-base length	0.33[m]
Sampling interval	0.1[s]
Sampling period	100[s]
Trails	100
Process noise covariance	$10^{-6} \cdot \mathbf{I}$
Observation noise covariance	$10^{-3} \cdot \mathbf{I}$
Probability of outliers	5%
Outlier covariance	$10^{1} \cdot \mathbf{I}$
Probability of packet-loss	1%
Duration of packet-loss	1.5[s]

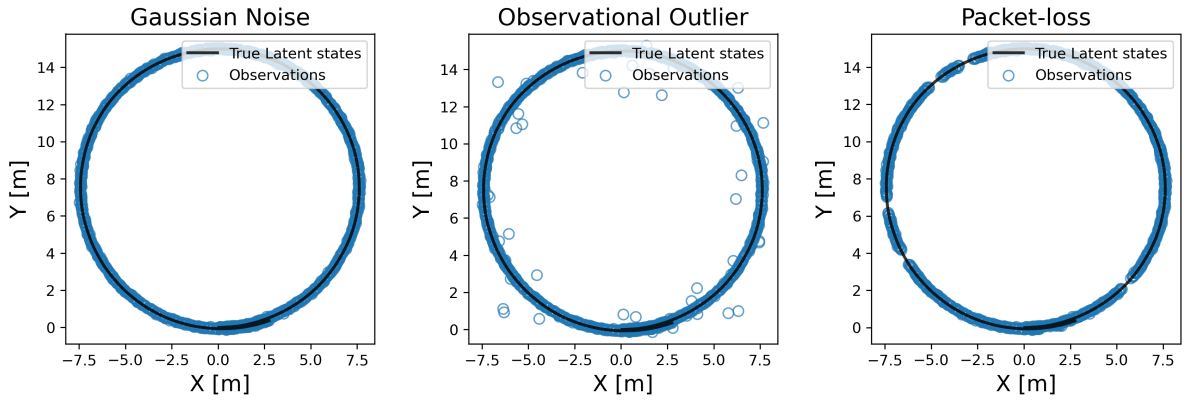


Figure 5.2: One example trail of the synthetic dataset generated by the kinematic bicycle model with different types of disturbance, the full dataset is demonstrated in Figure B.1.

5.3.2. Performance of the Observers

To select the hyperparameter for the observers, Bayesian optimization [7] is employed, following the setup described in [7][1]. The resulting parameters are listed in Table 5.3:

- i is the number of the iterations for the variational Bayesian inference: i (only the integer part is used), if i is higher, the estimation are more likely to converge to fix value, but can take a lot of time.
- $\nu_0, \nu_{Q,0}, \nu_{R,0}$ are the degree-of-freedom that determines the shape of the Inverse-Wishart distribution; a higher value corresponds to lower uncertainty in the covariance.
- α_0, β_0 are the parameters that determine the shape of the Beta distribution.

From Figure 5.3, the RMSE and EMax of the position (X-Y) estimation are compared across all 100 trials. In the box plot, a higher position indicates a larger estimation error and lower accuracy, while a wider box indicates that more samples fall within that error range.

As shown in Table 5.4, under Gaussian noise, the adaptive methods (EKF-B-adp and EKF-IW-adp) perform worse than the non-adaptive methods, while the non-adaptive methods exhibit similar performance. As shown in Figure 5.4, the fourth subplot compares the estimation error of the EKF-B-adp and EKF-B. The estimation error of EKF-B-adp is larger and less smooth. The second subplot shows that the EKF-B-adp attempts to estimate the process noise covariance. However, since there is no discrepancy between the process model and the simulation model, estimating the covariance from observations leads to an overestimation of the process noise covariance. This overestimation causes the adaptive methods to be more affected by noise, resulting in inferior performance compared to the non-adaptive methods.

As shown in Table 5.4, in the presence of observational outliers, the EKF-IW demonstrates the best performance, while the EKF-B has the worst performance on average. The adaptive methods show promising

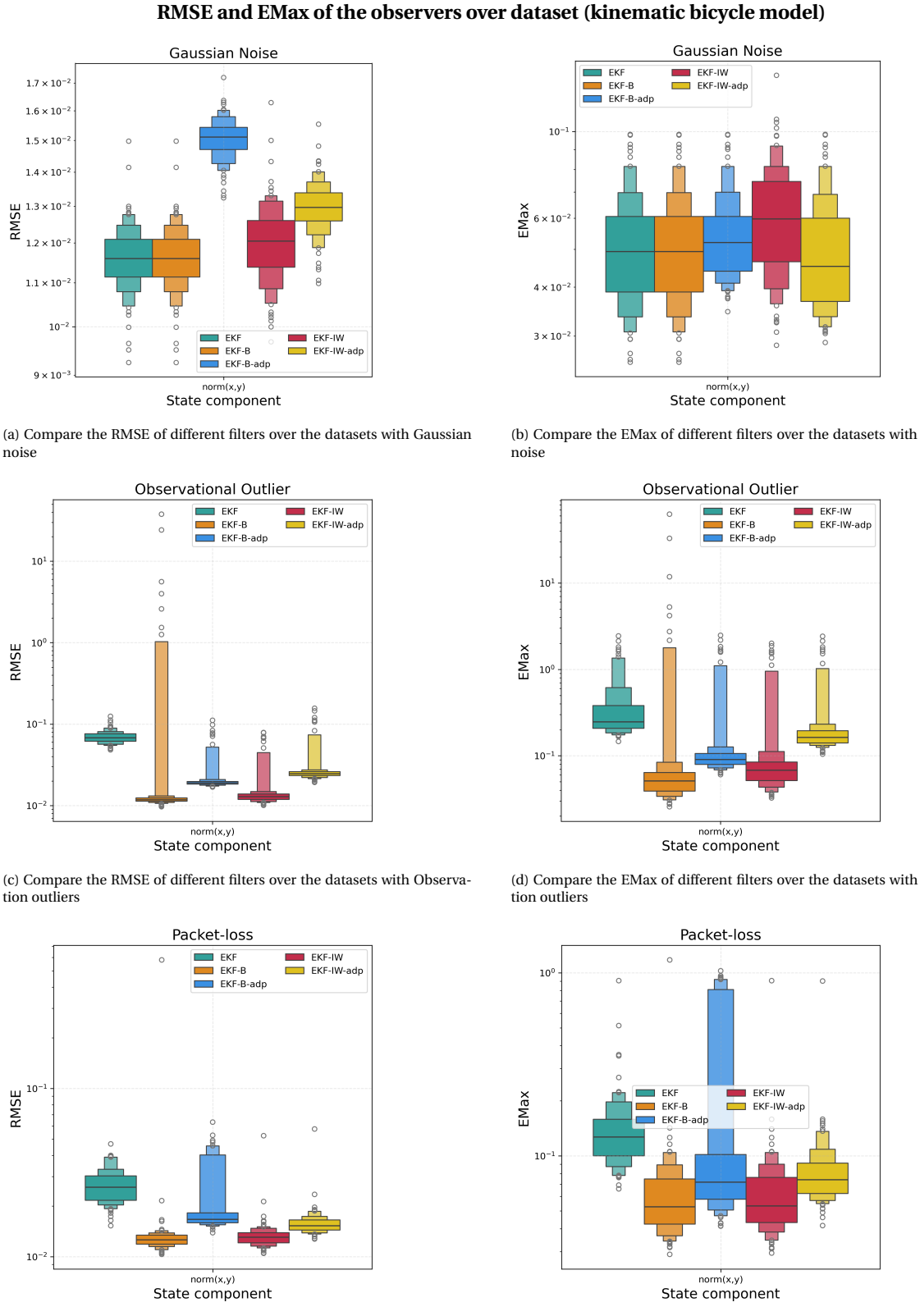
[1] Bayesian Optimization: `https://github.com/bayesian-optimization/BayesianOptimization`

Table 5.3: Parameters of observers under different types of observational disturbance on the dataset (kinematic bicycle model)

Noise Type	Method	Parameters
Gaussian Noise	EKF	NaN
Gaussian Noise	EKF-IW	$i = 2.0$, $\nu_0 = 11.25$
Gaussian Noise	EKF-B	$\alpha_0 = 9.17$, $\beta_0 = 5.89$, $i = 4.12$
Gaussian Noise	EKF-B-adp	$\alpha_0 = 6.6$, $\beta_0 = 10.0$, $i = 7.03$, $\nu_0 = 100.0$
Gaussian Noise	EKF-IW-adp	$\nu_{Q,0} = 100.0$, $\nu_{R,0} = 3.0$, $i = 10.0$
Observational Outlier	EKF	NaN
Observational Outlier	EKF-IW	$i = 2.85$, $\nu_0 = 6.04$
Observational Outlier	EKF-B	$\alpha_0 = 6.98$, $\beta_0 = 0.0$, $i = 3.95$
Observational Outlier	EKF-B-adp	$\alpha_0 = 6.54$, $\beta_0 = 10.0$, $i = 7.09$, $\nu_0 = 100.0$
Observational Outlier	EKF-IW-adp	$\nu_{Q,0} = 100.0$, $\nu_{R,0} = 3.0$, $i = 4.79$
Packet-loss	EKF	NaN
Packet-loss	EKF-IW	$i = 2.0$, $\nu_0 = 100.0$
Packet-loss	EKF-B	$\alpha_0 = 9.17$, $\beta_0 = 5.89$, $i = 4.12$
Packet-loss	EKF-B-adp	$\alpha_0 = 6.42$, $\beta_0 = 10.0$, $i = 7.27$, $\nu_0 = 100.0$
Packet-loss	EKF-IW-adp	$\nu_{Q,0} = 100.0$, $\nu_{R,0} = 3.0$, $i = 2.0$

Table 5.4: performance of observers under different types of observational disturbance on the dataset (kinematic bicycle model)

Noise Type	Method	RMSE |X,Y| [m]	EMax |X,Y| [m]	Computation time [s]
Gaussian Noise	EKF	0.011636	0.051601	0.015476
Gaussian Noise	EKF-IW	0.012028	0.061682	0.017239
Gaussian Noise	EKF-B	0.011636	0.051601	0.026903
Gaussian Noise	EKF-B-adp	0.015061	0.054684	0.046092
Gaussian Noise	EKF-IW-adp	0.012973	0.049840	0.051654
Observational Outlier	EKF	0.070004	0.411163	0.007794
Observational Outlier	EKF-IW	0.016638	0.176676	0.011559
Observational Outlier	EKF-B	0.798132	1.237874	0.022745
Observational Outlier	EKF-B-adp	0.023556	0.220525	0.048677
Observational Outlier	EKF-IW-adp	0.031049	0.279771	0.021568
Packet-loss	EKF	0.026828	0.147029	0.010883
Packet-loss	EKF-IW	0.013523	0.070152	0.011750
Packet-loss	EKF-B	0.018422	0.072054	0.024483
Packet-loss	EKF-B-adp	0.021680	0.219865	0.041583
Packet-loss	EKF-IW-adp	0.016045	0.089382	0.011250



(a) Compare the RMSE of different filters over the datasets with Gaussian noise

(b) Compare the EMax of different filters over the datasets with Gaussian noise

(c) Compare the RMSE of different filters over the datasets with Observation outliers

(d) Compare the EMax of different filters over the datasets with Observation outliers

(e) Compare the RMSE of different filters over the datasets with Packet-loss

(f) Compare the EMax of different filters over the datasets with Packet-loss

Figure 5.3: The comparison of the RMSE and EMax of different filters under different types of disturbance, when the process model is perfectly aligned with the simulator dynamics.

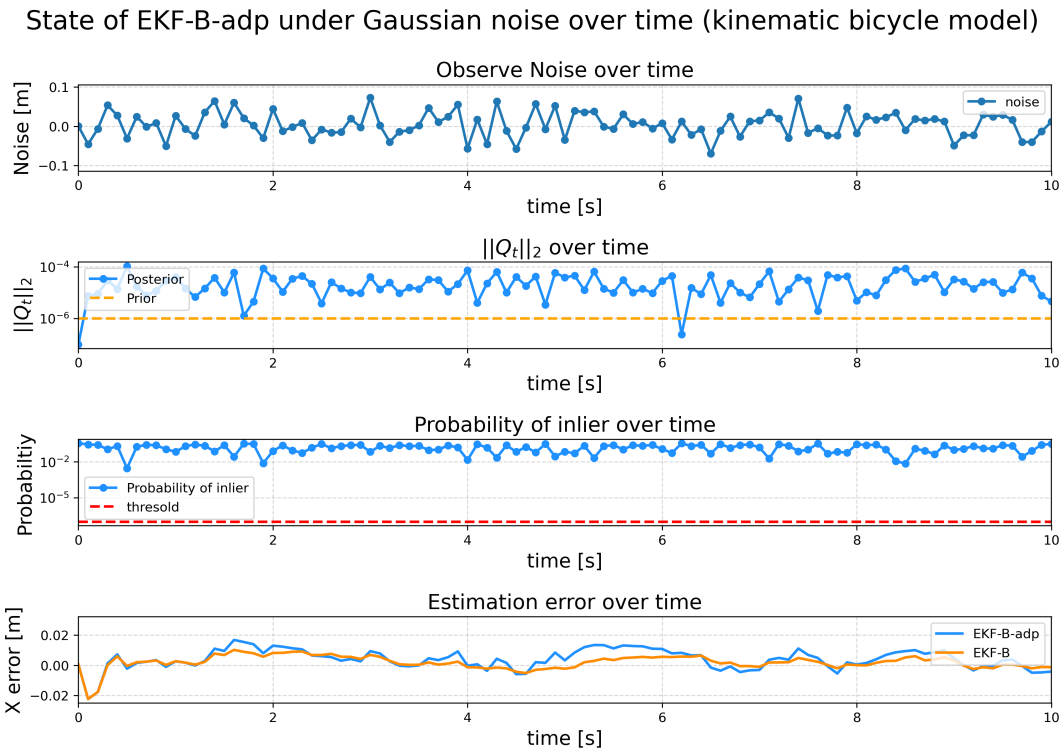


Figure 5.4: The estimation of latent states by Adaptive EKF-Bernoulli on the on the dataset generated by the kinematic bicycle model with Gaussian noise. From upper side: (a) the first figure shows the observational noise over time; (b) the second figure shows the l2-norm of the estimated process noise over time; (c) the third figure shows the estimated probability of inlier over time, and the classification threshold; (d) the forth figure shows the estimation error of state X over time of EKF and EKF-B-adp.

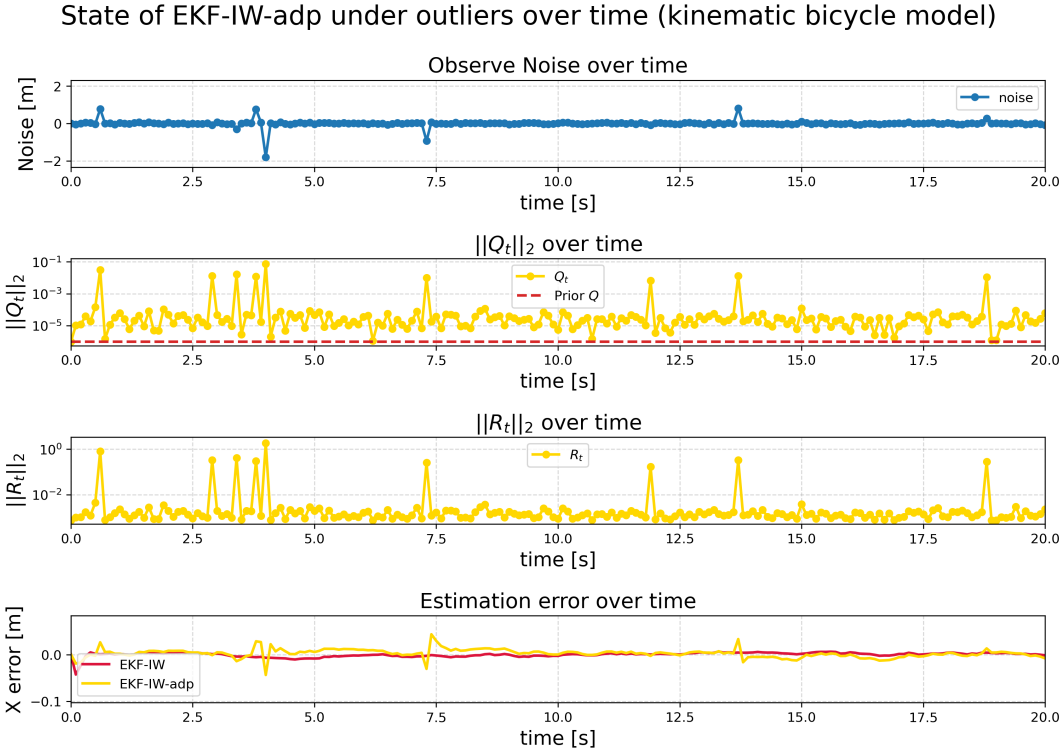


Figure 5.5: The estimation of latent states by EKF-IW-adp on the under observational outliers on the dataset (kinematic bicycle model). From top: (a) the first figure shows the observational noise over time; (b) the second figure shows the l2-norm of the estimated process noise covariance $||\mathbf{Q}_t||_2$ over time; (c) the second figure shows the l2-norm of the estimated observation noise covariance $||\mathbf{R}_t||_2$ over time; (d) the forth figure shows the estimation error of state X over time of EKF-IW and EKF-IW-adp.

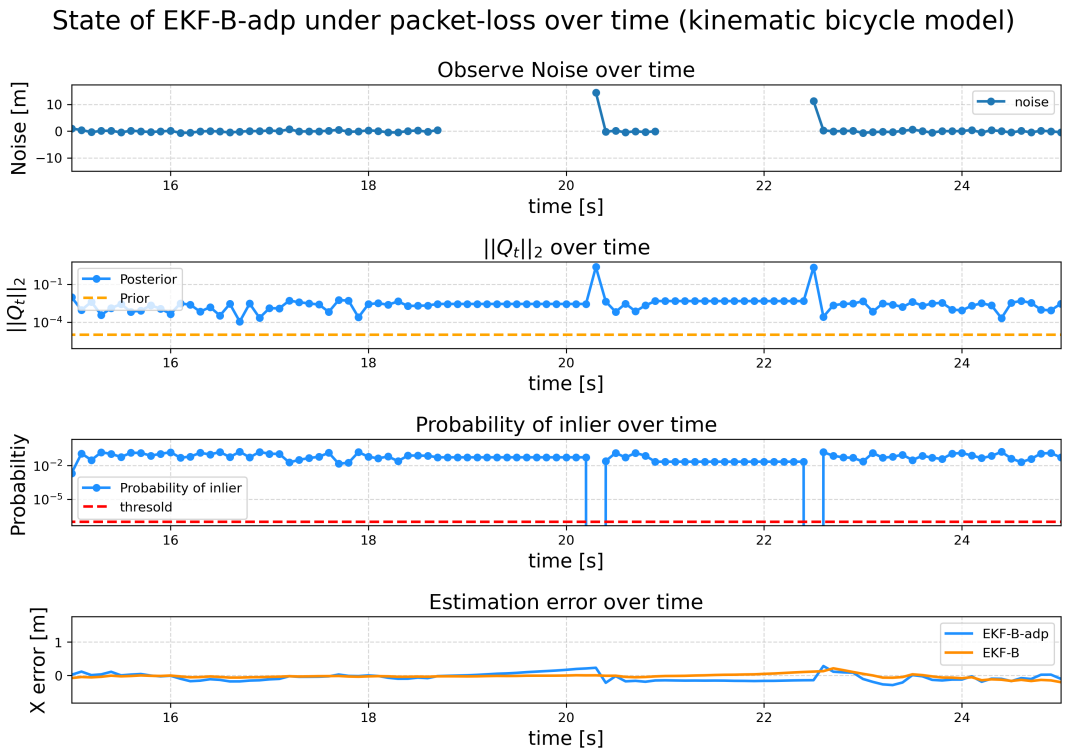


Figure 5.6: The estimation of latent states by Adaptive EKF-Bernoulli on the on the dataset generated by the kinematic bicycle model with Packet-loss. From upper side: (a) the first figure shows the observational noise over time; (b) the second figure shows the l2-norm of the estimated process noise over time; (c) the third figure shows the estimated probability of inlier over time, and the classification threshold; (d) the forth figure shows the estimation error of state X over time of EKF-B and EKF-B-adp.

performance compared to the basic EKF. The basic EKF is heavily influenced by the outliers. Under the Gaussian noise assumption, observational outliers skew the posterior distribution toward the observations, and this influence persists for several time steps. For EKF-IW, large residuals increase the observation covariance, causing the filter to rely more on the process model predictions, as the process noise covariance is much smaller. However, for the EKF-B, as shown in Figures 5.3c and 5.3d, in most cases, the EKF-B performs well in rejecting outliers and providing accurate estimations. However, in some cases, the estimation is significantly poor. The EKF-B has a limitation: it continues to trust the model and rejects nominal observations that contain useful state information. For EKF-IW-adp, as shown in Figure 5.5, the simultaneous adjustment of the process noise covariance and observation noise covariance causes them to cancel each other out, resulting in poor outlier robustness.

As shown in Table 5.4 and Figures 5.3e and 5.3f, under packet loss, the adaptive methods (EKF-B-adp and EKF-IW-adp) exhibit worse performance compared to the non-adaptive methods. Delayed observations are numerically distant from the nominal state at the same timestamp, as illustrated in the upper subplot of Figure 5.6. During the loss period, no observations are available, and the filters rely solely on predictions from the process model. Since the process model is assumed to perfectly align with the simulation model, the system operates in an open-loop manner. If the estimation deviates before the loss period, the deviation increases during the loss period until observations are used for posterior estimation, as shown in the fourth subplot of Figure 5.6.

In terms of computational efficiency, as shown in Table 5.4, the EKF has the lowest computational cost, while the outlier-robust filters incur higher costs. The computational cost primarily depends on the hyperparameter i, where a higher i indicates more inner iterations for the latent states to converge.

5.3.3. Discussion

To summarize the results when there is no model discrepancy, the key takeaways are:

- The EKF consistently has the lowest computational cost, while the computational cost of the outlier-robust methods depends on the number of inner loop iterations i.

- The outlier-robust filters effectively remove the influence of observational outliers and provide accurate estimations. However, for the EKF-B, a good selection of parameters is crucial for performance, as it may reject useful information from the observations.
- The adaptive methods exhibit worse performance due to the absence of discrepancy between the process model and the simulation model. These methods overestimate the process noise covariance based on observations.
- EKF-IW-adp updates the process noise covariance and the observation noise covariance simultaneously, resulting in poor robustness to observational outliers.

5.4. Model Discrepancy and Small Assumed Process Noise Covariance

Assuming there is model discrepancy between the process model for observers and the simulation model, however, the assumed process noise covariance $\mathbf{Q}$ is set to be a relatively small value, the experiment aims to evaluate the performance of the observers when they are not foreseeing the model discrepancy.

- The process model for the observers is based on kinematic bicycle model.
- The datasets are generated using identified model (Equation (3.8)) with Gaussian noise, observational outliers and packet-loss.
- The assumed process noise covariance for the observers is set relatively small $||\mathbf{Q}||_2 = 10^{-5}$.

5.4.1. Dataset

The synthetic dataset generated using the identified model as Equation (3.8). The full synthetic dataset is shown in Figure B.2. The parameters for generating the dataset are listed in Table 5.5.

Table 5.5: The parameters for generating the synthetic data using the identified model

Property	Value
Vehicle mess	3.7 [kg]
Lateral stiffness	5.1 [N/m]
Longitudinal stiffness	2.38 [N/m]
Front axle to CoG	0.22 [m]
Rear axle to CoG	0.11 [m]
Wheel radius	0.056 [m]
Resistance coefficient	9.96
Steering coefficient	0.0008
Throttle coefficient	0.16
Steering time constant	0.0013 [s]
Throttle time constant	0.93 [s]
Sampling interval	0.1[s]
Sampling period	100[s]
Trails	100
Process noise covariance	$10^{-5} \cdot \mathbf{I}$
Observation noise covariance	$10^{-3} \cdot \mathbf{I}$
Probability of outliers	5%
Outlier covariance	$10^{-1} \cdot \mathbf{I}$
Probability of packet-loss	1%
Duration of packet-loss	1.5[s]

As shown in Figure 5.7, the observations are subject to different types of observational disturbances. Compared to the trajectory generated by the kinematic bicycle model in Figure 5.2, the trajectory of the identified model deviates from a perfect circle in each lap. This deviation is primarily due to the longitudinal velocity of the identified model being subject to the RC input dynamics, which requires time to stabilize at the reference velocity. As shown in Figure 5.8, when the same RC commands are applied to both the process model and the identified model, there is a major difference between their trajectories. This difference represents the model discrepancy that can affect the estimation performance of the filters.

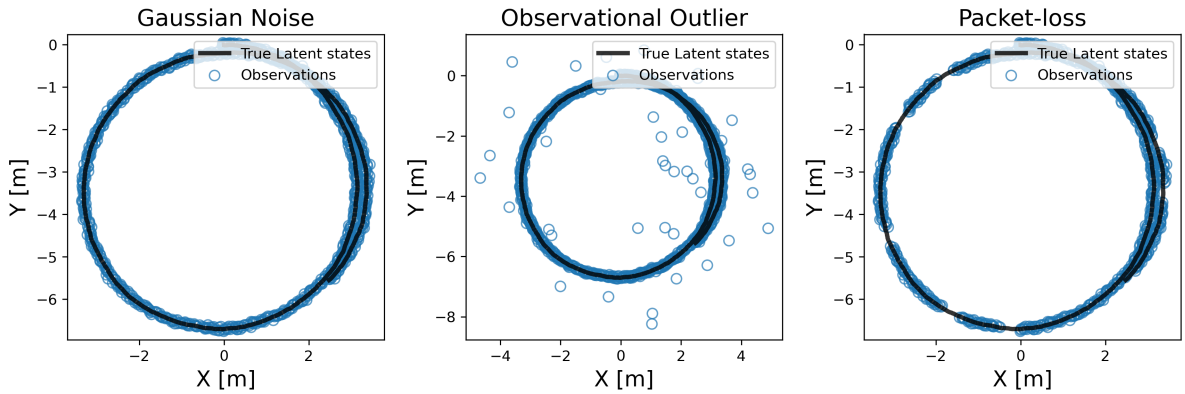


Figure 5.7: One example trail of the synthetic dataset generated by the realistic vehicle model with different types of disturbance.

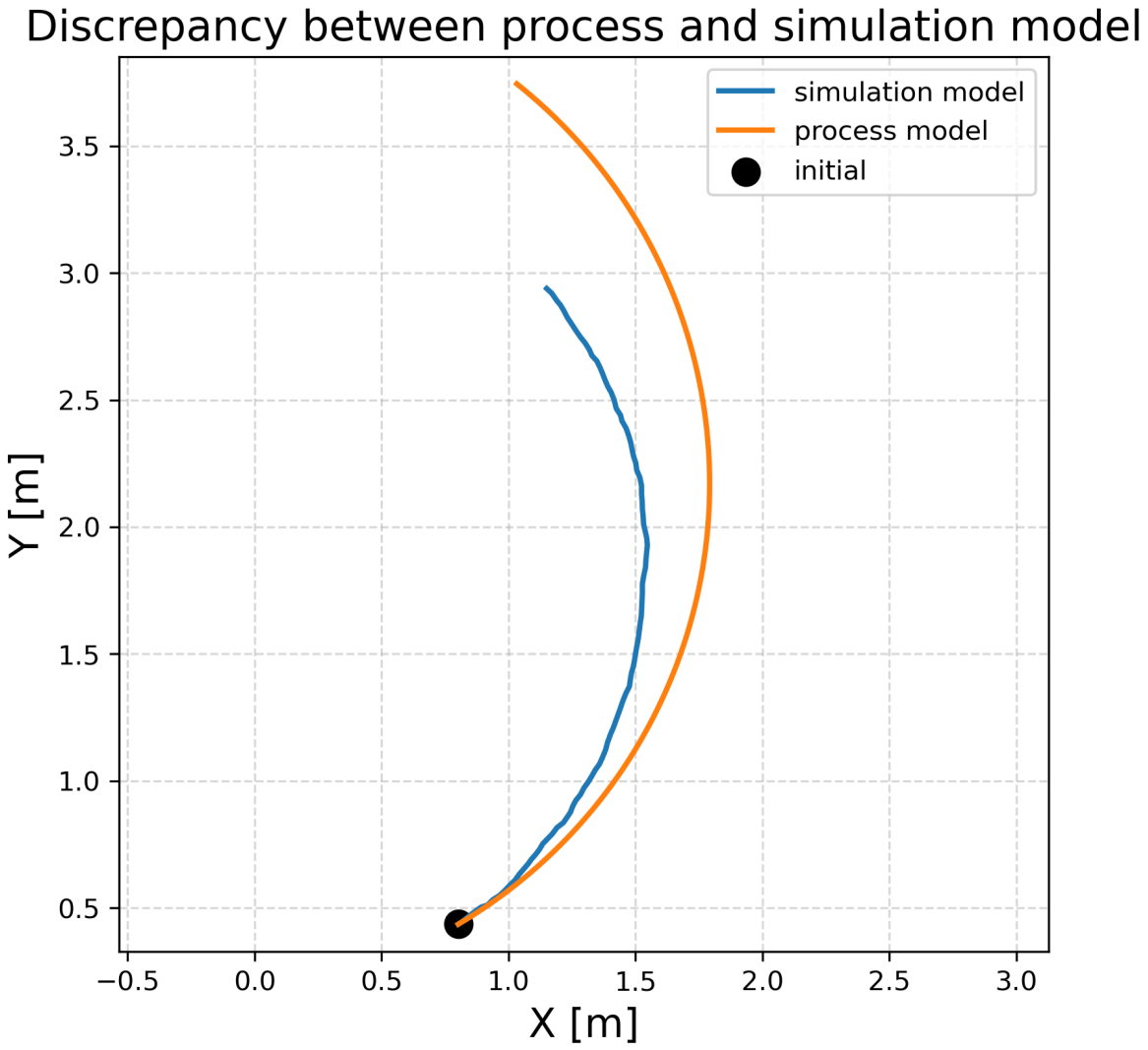


Figure 5.8: The discrepancy between the process model and the simulation model

5.4.2. Performance of the Observers

Assuming a discrepancy between the process model and the simulation model, the performance of the observers is compared under both observational disturbances and model discrepancies. In the experiment, the hyper-parameters are tuned using Bayesian Optimization. When the prior process noise covariance is set to a small value $\mathbf{Q} = 10^{-5} \cdot \mathbf{I}$, the hyper-parameters are listed in Table 5.6, and the results are shown in Figures 5.9 and B.4, as well as Table 5.7.

Table 5.6: When $||\mathbf{Q}||_2 = 10^{-5}$, parameters of observers under different types of observational disturbance on the dataset (identified model)

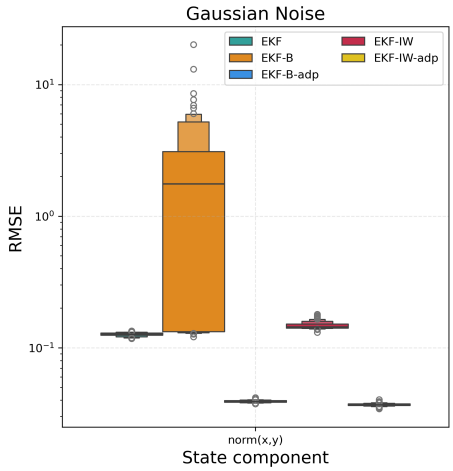
Noise Type	Method	Parameters
Gaussian Noise	EKF	NaN
Gaussian Noise	EKF-IW	$i = 10.0$, $\nu_0 = 100.0$
Gaussian Noise	EKF-B	$\alpha_0 = 7.16$, $\beta_0 = 0.0$, $i = 7.1$
Gaussian Noise	EKF-B-adp	$\alpha_0 = 10.0$, $\beta_0 = 4.2$, $i = 2.0$, $\nu_0 = 3.0$
Gaussian Noise	EKF-IW-adp	$\nu_{Q,0} = 8.0$, $\nu_{R,0} = 100.0$, $i = 2.0$
Observational Outlier	EKF	NaN
Observational Outlier	EKF-IW	$i = 10.0$, $\nu_0 = 100.0$
Observational Outlier	EKF-B	$\alpha_0 = 10.0$, $\beta_0 = 0.1$, $i = 7.49$
Observational Outlier	EKF-B-adp	$\alpha_0 = 6.28$, $\beta_0 = 0.1$, $i = 2.0$, $\nu_0 = 3.0$
Observational Outlier	EKF-IW-adp	$\nu_{Q,0} = 28.49$, $\nu_{R,0} = 3.0$, $i = 6.22$
Packet-loss	EKF	NaN
Packet-loss	EKF-IW	$i = 10.0$, $\nu_0 = 100.0$
Packet-loss	EKF-B	$\alpha_0 = 10.0$, $\beta_0 = 0.1$, $i = 10.0$
Packet-loss	EKF-B-adp	$\alpha_0 = 10.0$, $\beta_0 = 4.18$, $i = 2.0$, $\nu_0 = 3.0$
Packet-loss	EKF-IW-adp	$\nu_{Q,0} = 8.0$, $\nu_{R,0} = 20.59$, $i = 2.0$

Table 5.7: When $||\mathbf{Q}||_2 = 10^{-5}$, performance of observers under different types of observational disturbance on the dataset (identified model)

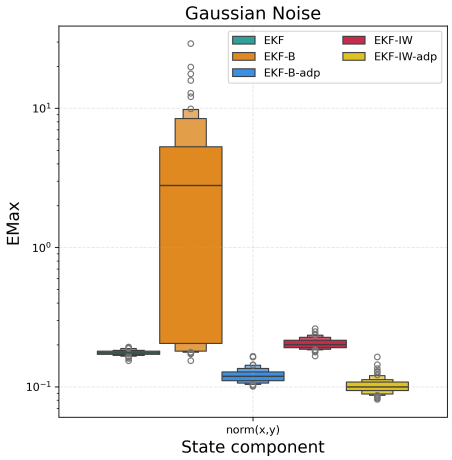
Noise Type	Method	RMSE |X,Y| [m]	EMax |X,Y| [m]	Computation time [s]
Gaussian Noise	EKF	0.126386	0.175566	0.010060
Gaussian Noise	EKF-IW	0.147443	0.204363	0.050901
Gaussian Noise	EKF-B	2.395385	4.003070	0.025893
Gaussian Noise	EKF-B-adp	0.039059	0.121088	0.020694
Gaussian Noise	EKF-IW-adp	0.036930	0.101920	0.010582
Observational Outlier	EKF	0.160129	0.648895	0.010060
Observational Outlier	EKF-IW	0.172320	0.369092	0.050901
Observational Outlier	EKF-B	2.815395	4.623943	0.025893
Observational Outlier	EKF-B-adp	0.046094	0.334490	0.020694
Observational Outlier	EKF-IW-adp	0.079729	0.593392	0.010582
Packet-loss	EKF	0.151368	0.677562	0.010060
Packet-loss	EKF-IW	1.028829	1.787883	0.050901
Packet-loss	EKF-B	5.338150	9.269091	0.025893
Packet-loss	EKF-B-adp	0.078110	0.618666	0.020694
Packet-loss	EKF-IW-adp	0.074056	0.600812	0.010582

As shown in Table 5.7, and Figure 5.9a and 5.9b, under Gaussian noise, the adaptive methods (EKF-B-adp and EKF-IW-adp) outperform the non-adaptive methods (EKF, EKF-B, and EKF-IW). The EKF also performs better than the non-adaptive outlier-robust methods (EKF-B and EKF-IW). As shown in Figure B.4a, the adaptive methods are able to estimate the system state, although the estimates are affected by noise, resulting in less smooth trajectories. In contrast, the EKF produces smoother trajectories by effectively denoising the observations, but its overall estimation accuracy is lower than that of the adaptive methods due to poorly set process noise covariance. The non-adaptive outlier-robust methods (EKF-B and EKF-IW) perform poorly. For EKF-B, observations distant from the process model predictions are rejected. With a small process noise

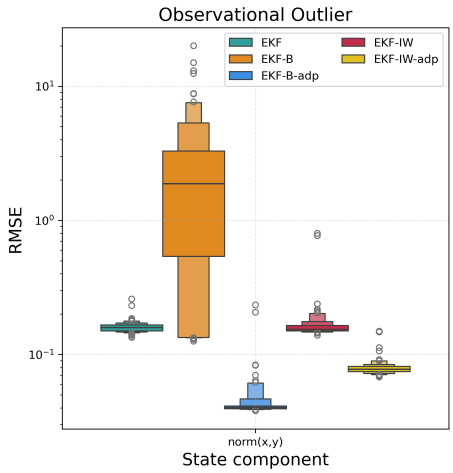
RMSE and EMax of the observers over dataset (identified model) with $||\mathbf{Q}||_2 = 10^{-5}$



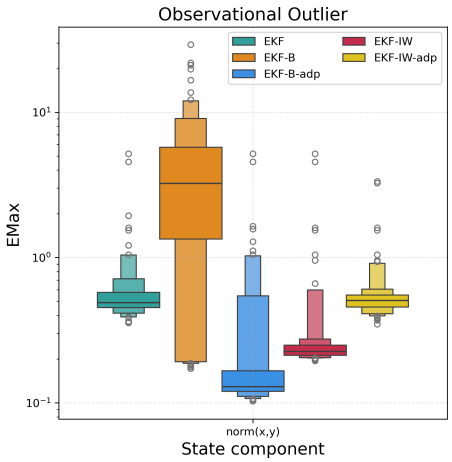
(a) Compare the RMSE of different filters over the datasets with Gaussian noise



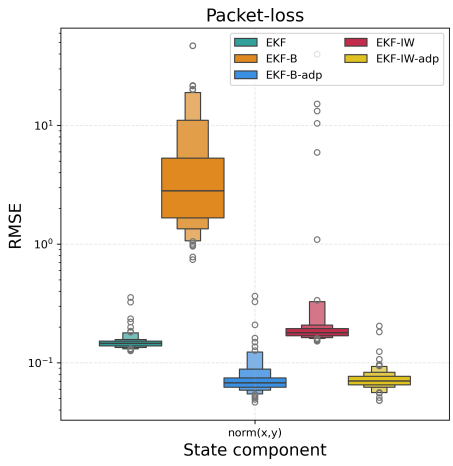
(b) Compare the EMax of different filters over the datasets with Gaussian noise



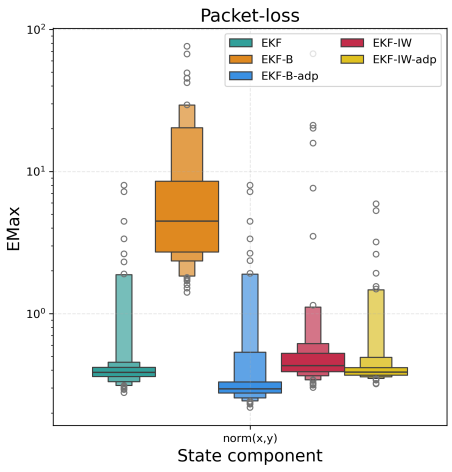
(c) Compare the RMSE of different filters over the datasets with Observation outliers



(d) Compare the EMax of different filters over the datasets with Observation outliers



(e) Compare the RMSE of different filters over the datasets with Packet-loss



(f) Compare the EMax of different filters over the datasets with Packet-loss

Figure 5.9: The comparison of the RMSE and EMax of observers under different types of disturbance over the dataset (identified model) when the prior process noise covariance is low $||\mathbf{Q}||_2 = 10^{-5}$.

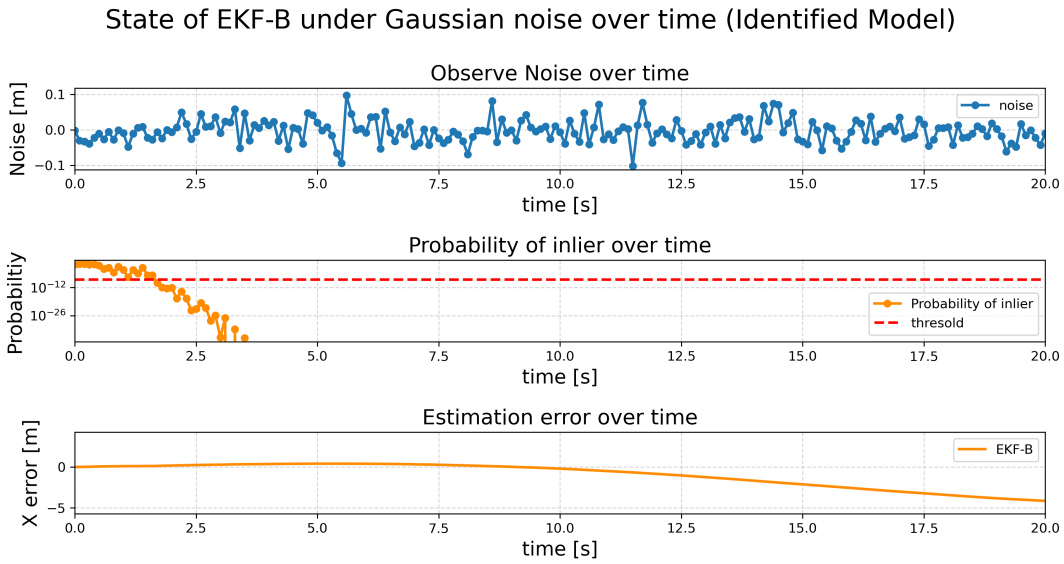


Figure 5.10: The estimation of latent states by EKF-B over time under Gaussian noise on the dataset (identified model) when the prior process noise covariance is low $||\mathbf{Q}||_2 = 10^{-5}$. From top: (a) the first figure shows the observational noise over time; (b) the second figure shows the estimated probability of inlier over time, and the classification threshold; (d) the third figure shows the estimation error of state X over time of EKF-B.

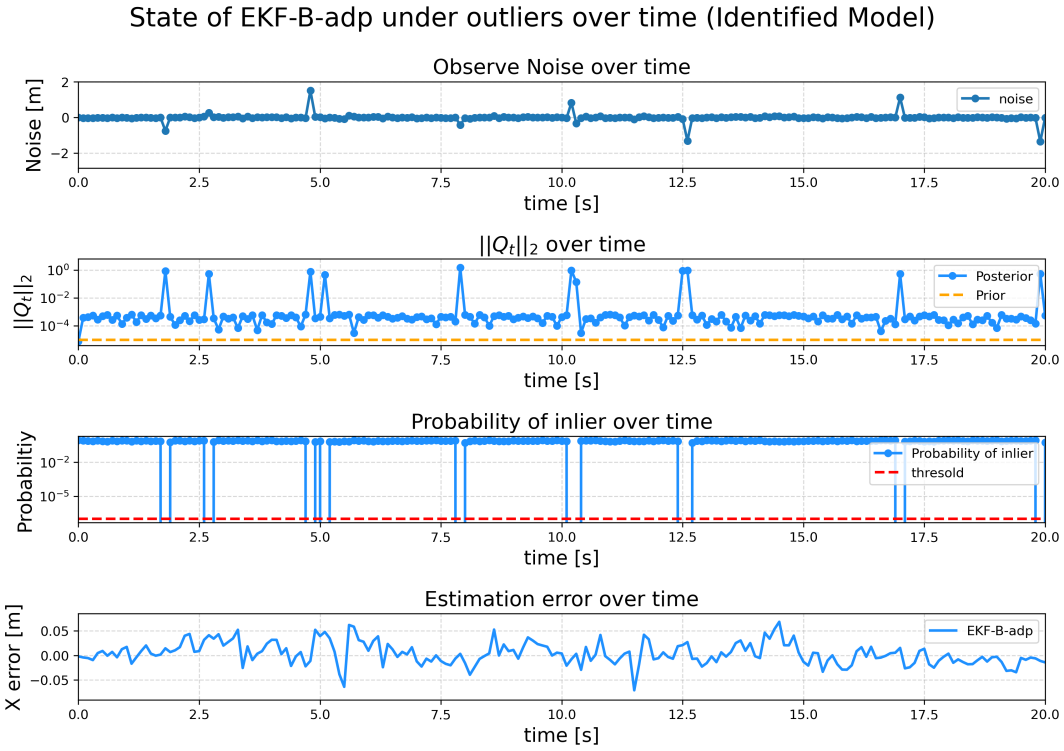


Figure 5.11: The estimation of latent states by EKF-B-adp over time under observational outliers on the dataset (identified model) when the prior process noise covariance is low $||\mathbf{Q}||_2 = 10^{-5}$. From top: (a) the first figure shows the observational noise over time; (b) the second figure shows the estimated process covariance $||\mathbf{Q}_t||_2$ over time; (c) the third figure shows the estimated probability of inlier over time, and the classification threshold; (d) the forth figure shows the estimation error of state X over time of EKF-B-adp.

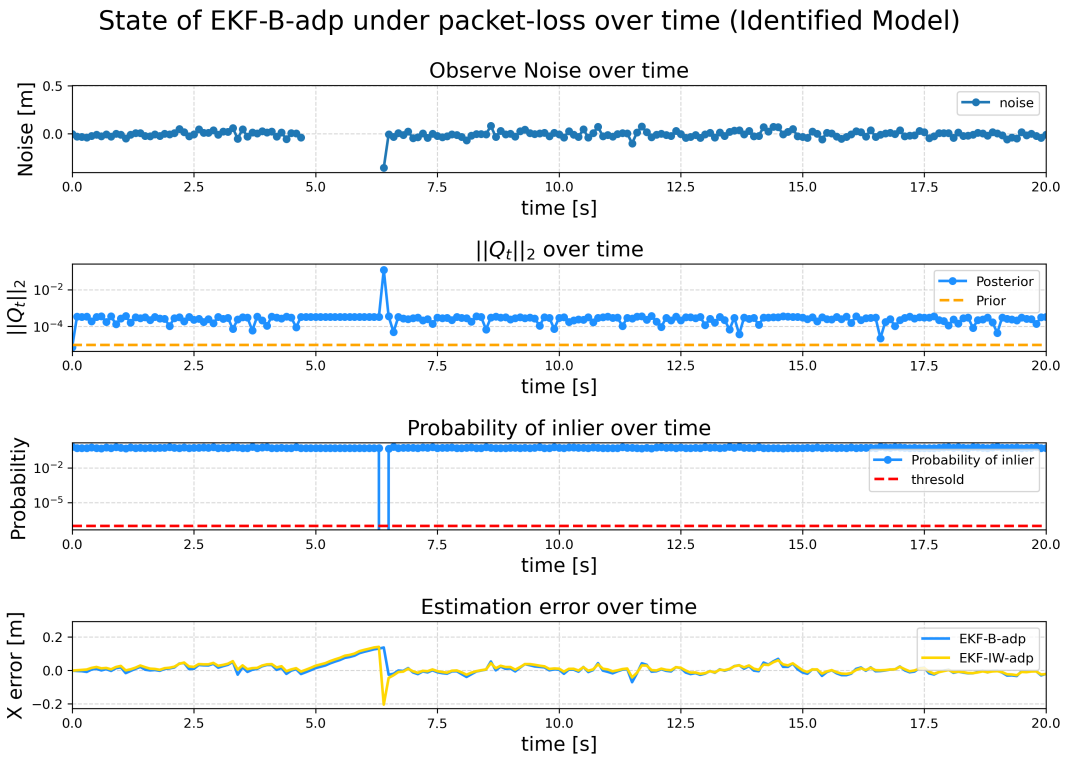


Figure 5.12: The estimation of latent states by EKF-B-adp over time under packet-loss on the dataset (identified model) when the prior process noise covariance is low $||\mathbf{Q}||_2 = 10^{-5}$. From top: (a) the first figure shows the observational noise over time; (b) the second figure shows the estimated process covariance $||\mathbf{Q}_t||_2$ over time; (c) the third figure shows the estimated probability of inlier over time, and the classification threshold; (d) the forth figure shows the estimation error of state X over time of EKF-B-adp.

covariance, EKF-B eventually trusts no observations and relies solely on predictions from the process model. Similarly, EKF-IW neither fully trusts its process model nor recognizes that the observations contain useful information, leading to poor estimations. However, EKF-IW still incorporates some information from the observations, resulting in better estimation compared to EKF-B. The state of EKF-B is illustrated in Figure 5.10, where the inlier probability drops below the threshold after several time steps, causing all subsequent observations to be rejected.

As shown in Table 5.7, and Figure 5.9c and 5.9d in the presence of observational outliers, the adaptive methods (EKF-B-adp and EKF-IW-adp) outperform the non-adaptive methods, with EKF-B-adp achieving better performance than EKF-IW-adp. However, as shown in Figure 5.9d, the performance variance of EKF-B-adp is relatively large. Figure B.4b demonstrates that the adaptive methods are still affected by outliers, resulting in non-smooth estimates, with EKF-IW-adp being more significantly impacted. The non-adaptive outlier-robust methods continue to perform poorly in this scenario. As shown in Figure 5.11, EKF-B-adp successfully detects observational outliers, as the inlier probability drops below the threshold when outliers appear. Additionally, the estimated process noise covariance adapts over time in response to significant noise or outliers.

As shown in Table 5.7, when packet loss occurs, as shown in Figures 5.9e and 5.9f, the adaptive outlier-robust methods demonstrate superior performance. The estimation results are illustrated in Figure B.4c. As shown in the fourth plot of Figure 5.12, the error between the estimation and the ground truth increases during the loss period. The delayed observation is identified as an outlier and rejected, while the nominal observations are used for posterior estimation. At the end of the loss period, the observations are used to correct the discrepancy accumulated during the loss period. In contrast, in Figure B.4c, the non-adaptive outlier-robust methods produce highly deviated estimations.

5.5. Model Discrepancy and Proper Assumed Process Noise Covariance

The process noise covariance $\mathbf{Q}$ is a key design parameter for observers, reflecting the reliability of the process model. Given the model discrepancy, the experiment evaluates the performance of the observers when the

process noise covariance is set to a larger value, $||\mathbf{Q}||_2 = 10^{-3}$. The result of the experiment is shown in Figure 5.13, Figure B.5, Table 5.9, and the parameters of the observers are in Table 5.8.

- The process model for the observers is based on kinematic bicycle model.
- The datasets are generated using identified model (Equation (3.8)) with Gaussian noise, observational outliers and packet-loss.
- The assumed process noise covariance for the observers is set relatively small $||\mathbf{Q}||_2 = 10^{-3}$.

5.5.1. Performance of the Observers

Table 5.8: When $||\mathbf{Q}||_2 = 10^{-3}$, parameters of observers under different types of observational disturbance on the dataset (identified model)

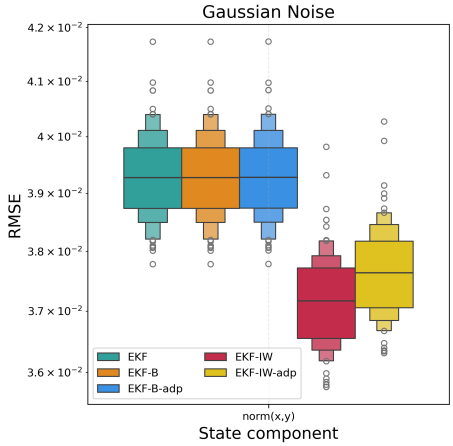
Noise Type	Method	Parameters
Gaussian Noise	EKF	NaN
Gaussian Noise	EKF-IW	$i = 2.42$, $\nu_0 = 5.73$
Gaussian Noise	EKF-B	$\alpha_0 = 9.17$, $\beta_0 = 5.89$, $i = 4.12$
Gaussian Noise	EKF-B-adp	$\alpha_0 = 0.16$, $\beta_0 = 0.4$, $i = 7.95$, $\nu_0 = 99.93$
Gaussian Noise	EKF-IW-adp	$\nu_{Q,0} = 73.93$, $\nu_{R,0} = 5.01$, $i = 2.0$
Observational Outlier	EKF	NaN
Observational Outlier	EKF-IW	$i = 6.96$, $\nu_0 = 4.29$
Observational Outlier	EKF-B	$\alpha_0 = 0.22$, $\beta_0 = 9.11$, $i = 5.5$
Observational Outlier	EKF-B-adp	$\alpha_0 = 10.0$, $\beta_0 = 5.59$, $i = 2.0$, $\nu_0 = 100.0$
Observational Outlier	EKF-IW-adp	$\nu_{Q,0} = 100.0$, $\nu_{R,0} = 3.0$, $i = 4.92$
Packet-loss	EKF	NaN
Packet-loss	EKF-IW	$i = 2.0$, $\nu_0 = 17.97$
Packet-loss	EKF-B	$\alpha_0 = 9.18$, $\beta_0 = 5.93$, $i = 4.12$
Packet-loss	EKF-B-adp	$\alpha_0 = 0.8$, $\beta_0 = 9.82$, $i = 2.33$, $\nu_0 = 12.63$
Packet-loss	EKF-IW-adp	$\nu_{Q,0} = 88.8$, $\nu_{R,0} = 15.41$, $i = 2.0$

Table 5.9: When $||\mathbf{Q}||_2 = 10^{-3}$, metrics of observers under different types of observational disturbance on the dataset (identified model)

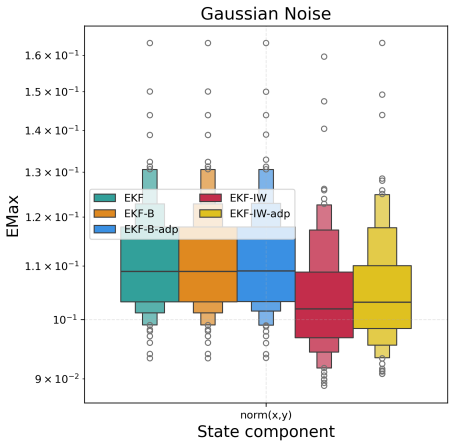
Noise Type	Method	RMSE |X,Y| [m]	EMax |X,Y| [m]	Computation time [s]
Gaussian Noise	EKF	0.039319	0.111732	0.007537
Gaussian Noise	EKF-IW	0.037166	0.104784	0.013029
Gaussian Noise	EKF-B	0.039319	0.111732	0.029546
Gaussian Noise	EKF-B-adp	0.039323	0.111806	0.071405
Gaussian Noise	EKF-IW-adp	0.037662	0.106346	0.011966
Observational Outlier	EKF	0.275430	2.494505	0.007537
Observational Outlier	EKF-IW	0.047451	0.231400	0.013029
Observational Outlier	EKF-B	0.055910	0.299032	0.029546
Observational Outlier	EKF-B-adp	0.051674	0.503403	0.071405
Observational Outlier	EKF-IW-adp	0.045599	0.281840	0.011966
Packet-loss	EKF	0.086933	0.876265	0.007537
Packet-loss	EKF-IW	0.065877	0.364489	0.013029
Packet-loss	EKF-B	0.628589	1.835428	0.029546
Packet-loss	EKF-B-adp	0.140016	1.023225	0.071405
Packet-loss	EKF-IW-adp	0.066260	0.385650	0.011966

As shown in Table 5.9 and Figures 5.13a and 5.13b, under Gaussian noise, the performance of all observers is nearly identical. The adaptive methods (EKF-B-adp and EKF-IW-adp) are able to identify the reliability of the process model from the observations and adjust the process noise covariance accordingly. However, when the process noise covariance is set to an appropriate value and no observational outliers are present,

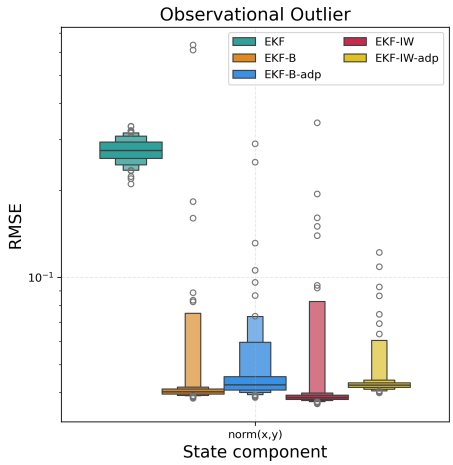
RMSE and EMax of the observers over dataset (identified model) with $||\mathbf{Q}||_2 = 10^{-3}$



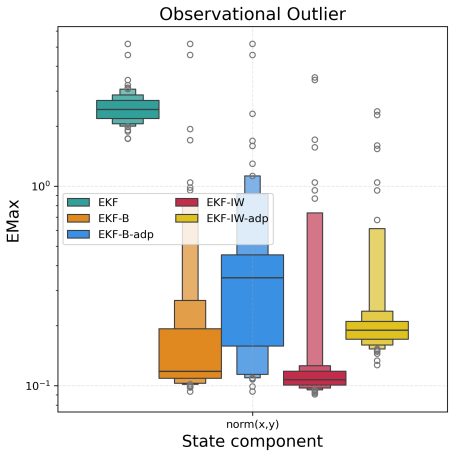
(a) RMSE of observers under Gaussian noise



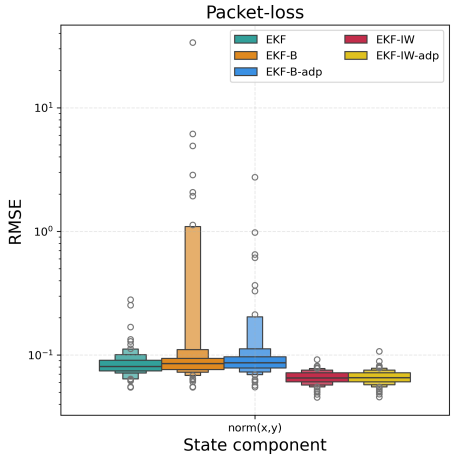
(b) EMax of observers under Gaussian noise



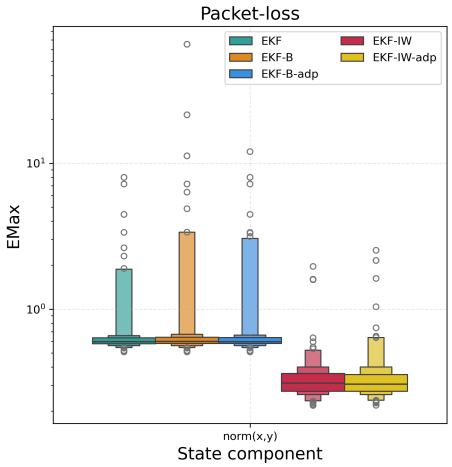
(c) RMSE of observers under observational outliers



(d) EMax of observers under observational outliers



(e) RMSE of observers under packet-loss



(f) EMax of observers under packet-loss

Figure 5.13: The comparison of the RMSE and EMax of observers under different types of disturbance over the dataset (identified model) when the prior process noise covariance is high $||\mathbf{Q}||_2 = 10^{-3}$.

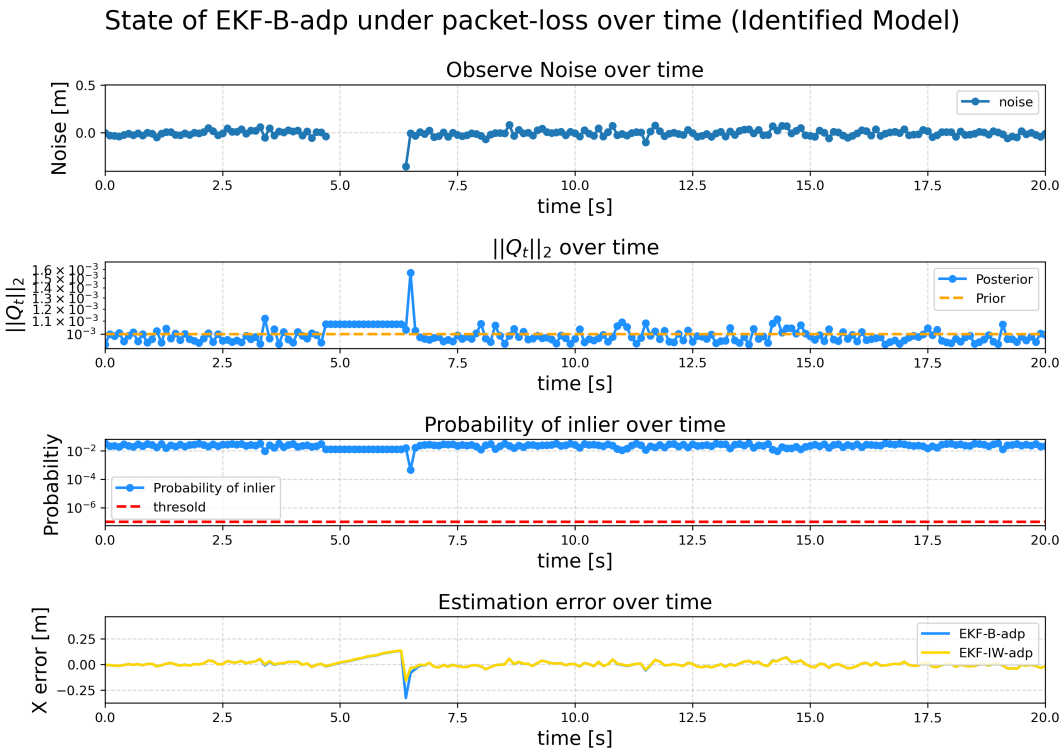


Figure 5.14: The estimation of latent states by EKF-B-adp over time under packet-loss on the dataset (identified model) when the prior process noise covariance is low $||\mathbf{Q}||_2 = 10^{-3}$. From top: (a) the first figure shows the observational noise over time; (b) the second figure shows the estimated process covariance $||\mathbf{Q}_t||_2$ over time; (c) the third figure shows the estimated probability of inlier over time, and the classification threshold; (d) the forth figure shows the estimation error of state X over time of EKF-B-adp.

the adaptive outlier-robust methods are not outperform the other methods. As shown in Figure B.5, the state estimation results are less smooth. When the assumed process noise covariance is set to a larger value, the estimation results for all methods became more susceptible to noise.

As shown in Table 5.9 and Figures 5.13e and 5.13f, when packet loss occurs, the EKF-B and EKF-B-adp exhibit worse performance. Due to the increased process noise covariance, the update rule for EKF-B-adp, $\Lambda_t = \frac{\nu_0\Lambda_0+S_t}{\nu_0+1}$, implies that the estimated process noise covariance $\mathbf{Q}_t$ is lower-bounded by $\frac{\nu_0\Lambda_0}{\nu_0+1}$. If the prior noise covariance is set too large, the estimation of $\mathbf{Q}_t$ is affected. As shown in Figure 5.14, the delayed observation after the loss period is not rejected, indicating that EKF-B-adp is less sensitive to less distant outliers. Similarly, EKF-IW-adp also exhibits reduced sensitivity to outliers. However, the robustness of EKF-B-adp relies heavily on its ability to successfully detect observational outliers, whereas EKF-IW-adp can down-weight outliers based on residuals at all times. If outliers are incorporated into the filtering process, their impact is more significant for EKF-B-adp, leading to worse performance.

5.5.2. Discussion

To summarize the results on the dataset with discrepancy, the key takeaways are:

- The adaptive outlier-robust methods can estimate the discrepancy between the process model and the simulation model, adjusting the process noise covariance accordingly. This property enables the adaptive methods to adapt to model discrepancy while rejecting observational outliers.
- When there is a discrepancy between the process model and the simulation model, the non-adaptive outlier-robust methods tend to over-trust the process model and neglect information from observations if the prior process noise covariance is set too small. When the prior process noise covariance is appropriately tuned, these methods perform well but are still affected by observational noise, resulting in less smooth estimates.
- EKF-B and EKF-B-adp struggle to fully detect and reject less distant observational outliers, leading to worse worst-case performance. Their performance heavily depends on the accurate identification of

outliers.

- For the adaptive methods, the estimated process noise covariance is lower-bounded by the prior process noise covariance. If the adaptive methods cannot accurately estimate the process noise covariance during filtering, their performance degrades.

6

The Performance of the Path-Tracking Framework

In the previous chapters, the identified vehicle model, the observers are introduced and evaluated. In this chapter, the performance of the outlier-robust path-tracking framework is evaluated based on integration of different path-tracking controllers and observers. A winding track is designated as a testing scenario for the framework, the results of the experiment are presented to identify the most suitable setup.

6.1. Overview

In Section 2.4, two geometric path-tracking controllers, the Pure-Pursuit Method (PPM) and the Stanley Method, are introduced. These controllers require the specification of two design hyperparameters. The performance of the path-tracking controllers is evaluated on a winding track without any state disturbances.

- Specify the look-ahead distance for the PPM and the lateral gain for the Stanley Method.
- The controllers use the ground-truth state for feedback control.

Next, the integration of different path-tracking controllers and the observers tested in Chapter 5 is evaluated on the winding track under observational disturbances and model discrepancy.

- The framework is evaluated under observational disturbances, including Gaussian noise, observational outliers, and packet loss.
- The simulation model is based on the identified model (RC force-moment model, as in Equation (3.8)), while the process model of the observers is based on the kinematic bicycle model (Equation (2.1)).

6.2. Designated Path and Path-Tracking Controller

In this section, the winding path is introduced as a testing scenario. The PPM and the Stanley method are evaluated on this scenario to specify their hyperparameters, and the tracking error is introduced as the evaluation metric.

6.2.1. Designated Path

The winding path is generated using the Map Generator, as introduced in Section 3.1.1. The configuration of the winding path is determined by the magnitude and frequency of a sinusoidal wave. The path is shown in Figure 6.1. The sinusoidal wave combines both sharp turns and relatively straight parts between the turns. A higher frequency results in a more winding path, while a higher magnitude leads to longer straight parts but sharper turns.

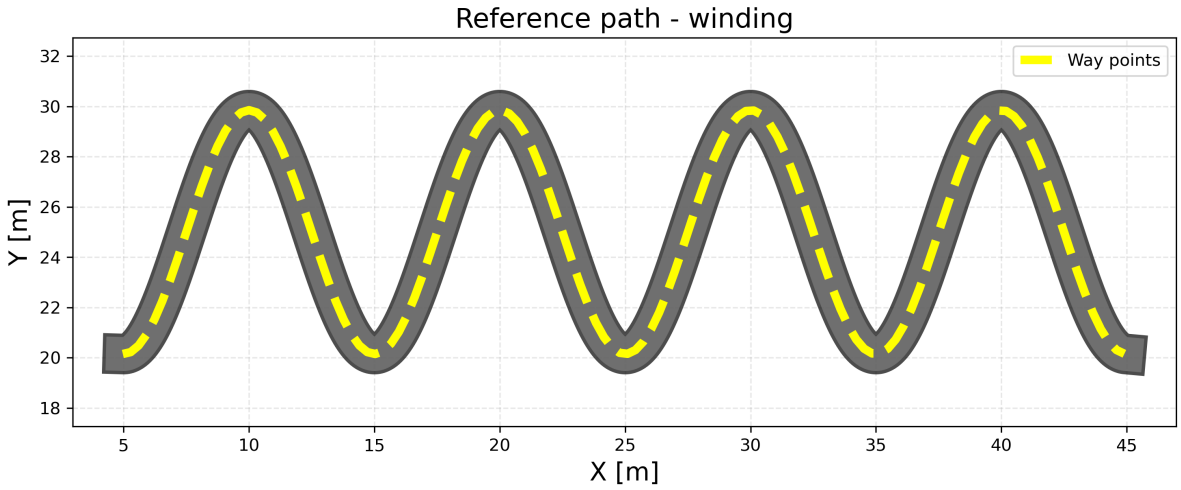


Figure 6.1: The designated path generated from sinusoid wave.

6.2.2. Path-tracking Error

For a reference path, the primary objective of path-tracking control is to minimize the tracking error. Given the vehicle's position and orientation $[X_v, Y_v, \Psi_v]$ in the global frame and the nearest waypoint relative to the vehicle position on the track $[X_{np}, Y_{np}, \Psi_{np}]$ in the global frame, the tracking error $[x_e, y_e, \psi_e]$ in the vehicle-body frame is defined as:

$$\begin{bmatrix} x_e \\ y_e \\ \psi_e \end{bmatrix} = \begin{bmatrix} \cos(\Psi_v) & \sin(\Psi_v) & 0 \\ -\sin(\Psi_v) & \cos(\Psi_v) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{np} - X_v \\ Y_{np} - Y_v \\ \Psi_{np} - \Psi_v \end{bmatrix} \tag{6.1}$$

Where x_e represents the lateral distance between the vehicle and the designated path, and ψ_e represents the orientation error between the vehicle and the designated path. To evaluate the average tracking performance over the entire simulation duration, the RMSE of the tracking error is used.

6.2.3. Parameters of path-tracking controller

Table 6.1: RMSE for Orientation and Lateral Errors on Winding Routes

controller type	LHD/gain	RMSE Orientation [rad]	RMSE Lateral [m]
pure_pursuit	0.1000 [m]	0.0959	0.1054
pure_pursuit	0.2000 [m]	0.0893	0.1050
pure_pursuit	**0.5000 [m]**	**0.0814**	**0.1044**
pure_pursuit	1.0000 [m]	0.1131	0.1048
pure_pursuit	1.5000 [m]	0.1912	0.1181
pure_pursuit	2.0000 [m]	0.2638	0.1559
pure_pursuit	3.0000 [m]	0.4245	0.2876
Stanley	0.1000	0.0691	0.1052
Stanley	**0.5000**	**0.0699**	**0.1038**
Stanley	1.0000	0.0755	0.1040
Stanley	2.0000	0.0789	0.1044
Stanley	5.0000	0.0827	0.1048
Stanley	10.0000	0.0845	0.1048
Stanley	15.0000	0.0855	0.1048

As introduced in Section 2.4, for the PPM, the design hyperparameter is the look-ahead distance (LHD), which determines the region where the controller searches for the goal point. For the Stanley method, the

design hyperparameter is the lateral gain, which determines how much the controller penalizes the lateral error y_e during feedback control.

The controllers are applied to the simulator, and the RMSE of the tracking error (Equation (6.1)) is compared to determine these hyperparameters. The simulation has a duration of 200 [s] with a fixed sampling interval of 0.1 [s]. The design parameters of the simulation model are listed in Table 5.5, and the reference velocity is set to 0.5 [m/s].

The RMSE of the lateral error and the orientation error are used to compare the performance of the path-tracking controllers, as shown in Table 6.1. From the results, since the target velocity of the vehicle is set to a very small value, the geometric methods perform well in path-tracking. It is observed that there is no significant difference in the orientation error and lateral error across the given parameter list; all configurations perform well. Therefore, the parameters resulting in the least orientation error are selected.

6.3. Evaluation of the Framework with Pure-Pursuit Method

In this section, the performance of the framework is evaluated, with five different observers and the PPM as the path-tracking controller.

- The framework is evaluated under three types of observational disturbances including Gaussian noise, observational outliers, packet-loss.
- The observers include EKF, the non-adaptive outlier-robust methods (EKF-B, EKF-IW), and the proposed adaptive outlier-robust method (EKF-B-adp, EKF-IW-adp).
- The path-tracking controller is the Pure-pursuit Method.

6.3.1. Parameters

The hyper-parameters of the simulation, observers, and controllers are specified in Table 6.2. For the simulation model, the hyper-parameters are identical to those in Table 4.1. For the adaptive observers (EKF-B-adp and EKF-IW-adp), the parameters from Table 5.6 for observational outliers are used, as the prior process noise covariance should be set lower for these methods. For the non-adaptive methods (EKF-B and EKF-IW), the parameters from Table 5.8 for observational outliers are used.

Table 6.2: The parameters for the outlier-robust path-tracking framework

Property	Value
Sampling interval	0.1[s]
Sampling period	100[s]
Process noise covariance	$10^{-5} \cdot \mathbf{I}$
Observation noise covariance	$10^{-3} \cdot \mathbf{I}$
Prior Process noise covariance	$10^{-3} \cdot \mathbf{I}$
Pure pursuit (Look ahead distance)	0.1 [m]
Stanley (Lateral gain)	1.0
EKF-B (α_0)	1.31
EKF-B (β_0)	3.92
EKF-B (number of inner loop i)	8.0
EKF-B-adp (α_0)	10.0
EKF-B-adp (β_0)	4.83
EKF-B-adp (process noise scaling ν_0)	3.0
EKF-B-adp (prior Process noise covarianceΛ_0)	$10^{-5} \cdot \mathbf{I}$
EKF-B-adp (number of inner loop i)	2.0
EKF-IW (observation noise scaling ν_0)	14.9
EKF-IW (number of inner loop i)	10.0
EKF-IW-adp (process noise scaling $\nu_{Q,0}$)	17.6
EKF-IW-adp (observation noise scaling $\nu_{R,0}$)	3.0
EKF-IW-adp (number of inner loop i)	10.0

6.3.2. Results

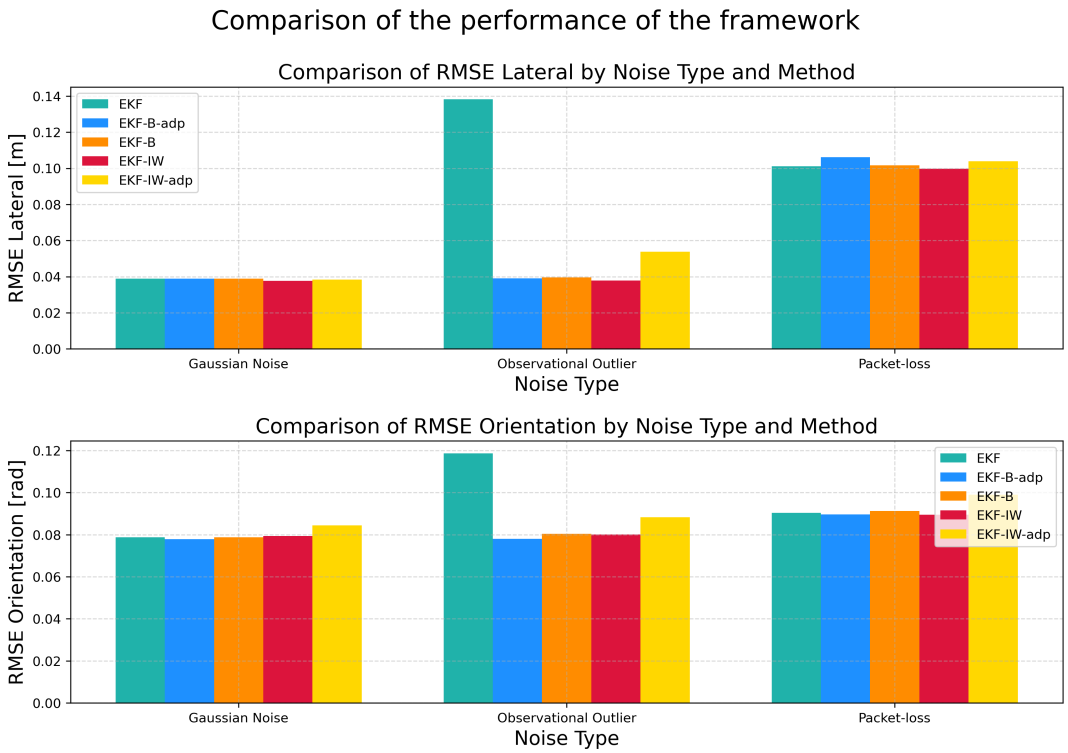
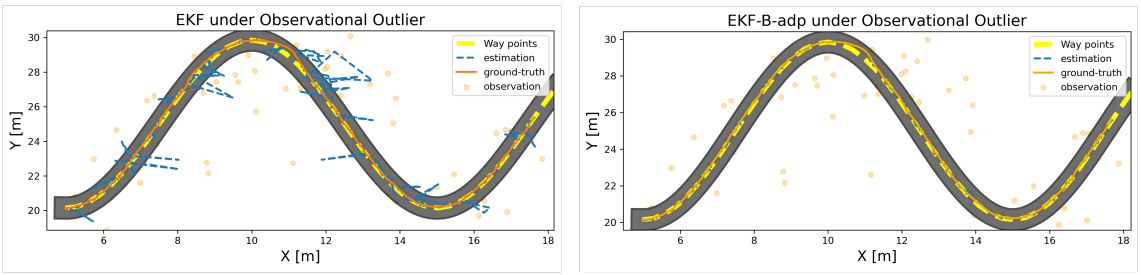


Figure 6.2: The comparison of the tracking error of the observers under different types of disturbance using Pure Pursuit method: (a) the first figure is the RMSE of the lateral error; (b) the second figure is the RMSE of the orientation error.

As shown in Figure 6.2, the RMSE of the lateral error and orientation error. Under Gaussian noise, the framework has almost similar performance with different observers, thus the both the lateral error and orientation error are quite low. In the presence of observational outliers, the performance of the EKF gets bad, this is due to the fact that EKF is not able to mitigate the influence of the observational outliers as shown in Figure 6.3a, some state estimations are disturbed by the observational outliers, resulting in some big spikes, and trajectory of the simulated vehicle appear to be non-smooth. In comparison, when the EKF-B-adp is applied as the observer, under the observational outliers, the state estimation is not be significantly affected, the estimation is very similar to the ground-truth (latent) state. When packet loss occurs, it poses a significant challenge for the framework. As shown in Figure 6.4, since the observers rely on open-loop simulation during the loss period, and due to the discrepancy between the process model and the simulation model, the vehicle's trajectory deviates from the desired path. The observer does not identify the delayed observation as an outlier, and the estimation is affected by it, resulting in poor tracking performance. The reason EKF-B-adp fails to identify the observational outlier is that the parameters it uses are not optimized for the packet-loss dataset.



(a) Path-tracking under observational outliers using EKF

(b) Path-tracking under observational outliers using EKF-B-adp

Figure 6.3: The trajectory and the state estimation performance of the path-tracking framework under observational outliers using PPM.

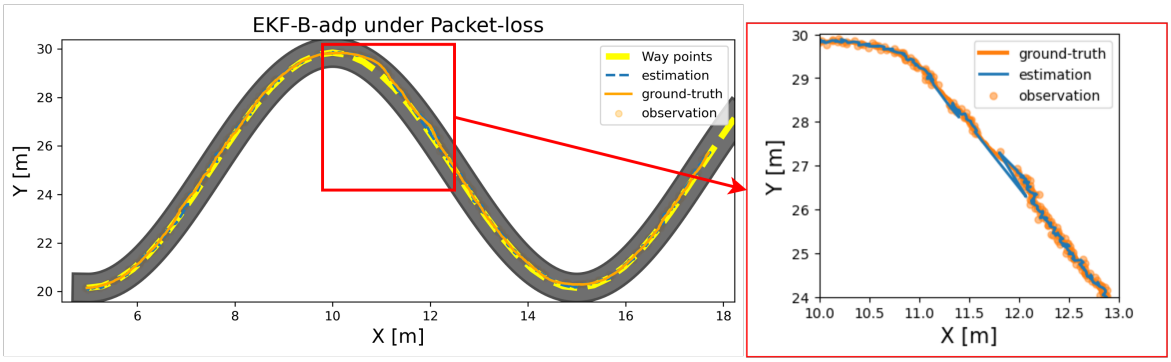


Figure 6.4: The trajectory and the state estimation performance of the path-tracking framework under packet-loss using PPM

6.4. Evaluation of the Framework with Stanley Method

In this section, the performance of the framework is evaluated, with five different observers and the Stanley method as the path-tracking controller. The parameters for the observers and controllers are in Table 6.2.

- The framework is evaluated under three types of observational disturbances including Gaussian noise, observational outliers, packet-loss.
- The observers include EKF, the non-adaptive outlier-robust methods (EKF-B, EKF-IW), and the proposed adaptive outlier-robust method (EKF-B-adp, EKF-IW-adp).
- The path-tracking controller is the Stanley Method.

6.4.1. Results

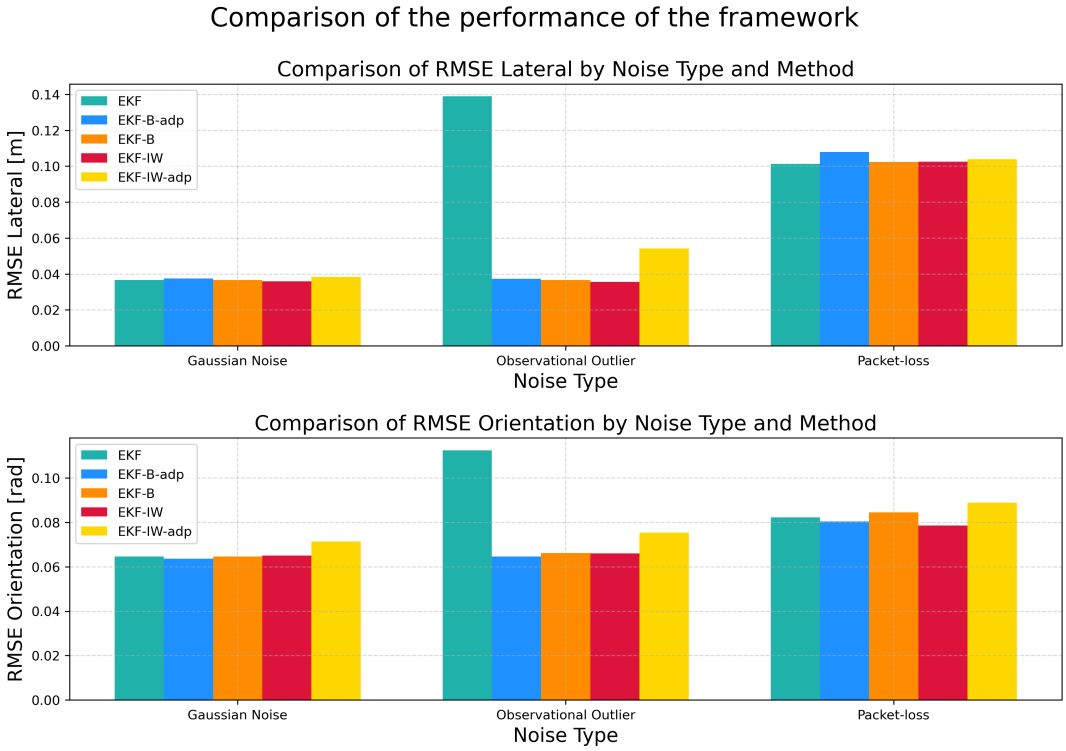
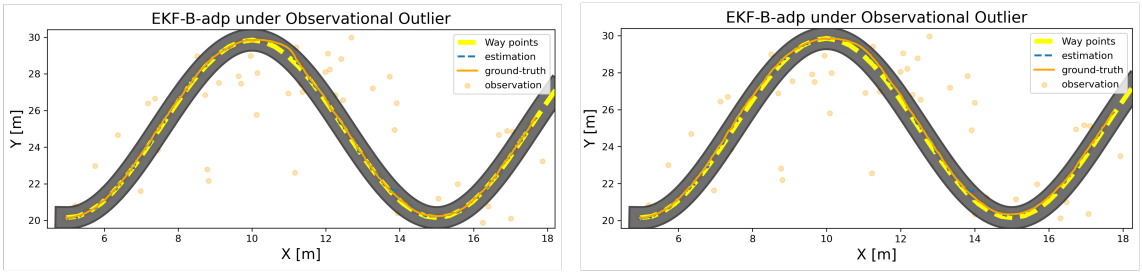


Figure 6.5: The comparison of the tracking error of the observers under different types of disturbance using Stanley: (a) the first figure is the RMSE of the lateral error; (b) the second figure is the RMSE of the orientation error.

As shown in Figure 6.5, in compared to Figure 6.2, the overall distribution of the performance over different observers are similar, while Stanley method exhibits better performance in minimizing orientation error compared to the Pure Pursuit method. As shown in Figure 6.6, comparing the trajectory of the vehicle after the sharp turn, on the left side, the trajectory using Stanley method is slightly smoother than the right side. This improvement stems from the Stanley method's control law, which explicitly regulates orientation error.



(a) Pure-Pursuit Method

(b) Stanley Method

Figure 6.6: Comparison between the PPM and Stanley method using the same observer under observational outliers

6.5. Improper Assumed Process Noise Covariance

As shown in Figure 6.7, the performance of the non-adaptive outlier-robust Kalman filters is significantly worse than that of the other methods. As shown in Figure 6.8, when the assumed process noise covariance for the observer is improperly set, the non-adaptive outlier-robust observers fail to perform state estimation. They ignore all observations as the residuals grow larger, rendering the framework unable to perform path-tracking with these observers.

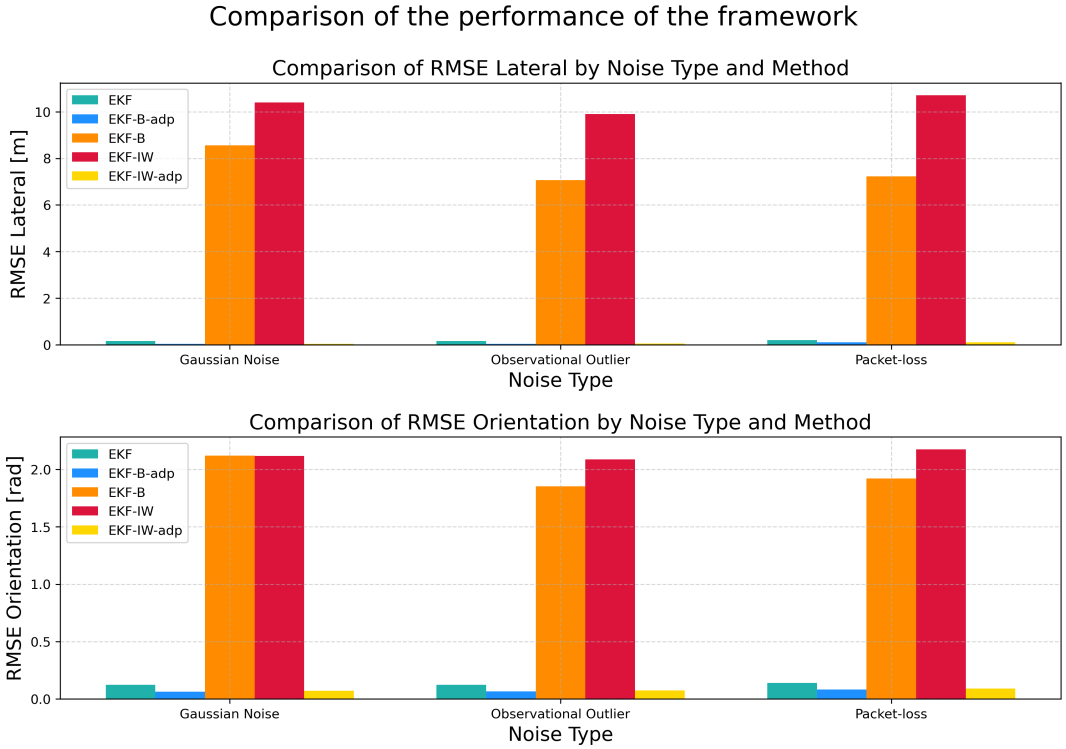


Figure 6.7: The comparison of the tracking error of the observers when the process noise covariance is improperly set: (a) the first figure is the RMSE of the lateral error; (b) the second figure is the RMSE of the orientation error.

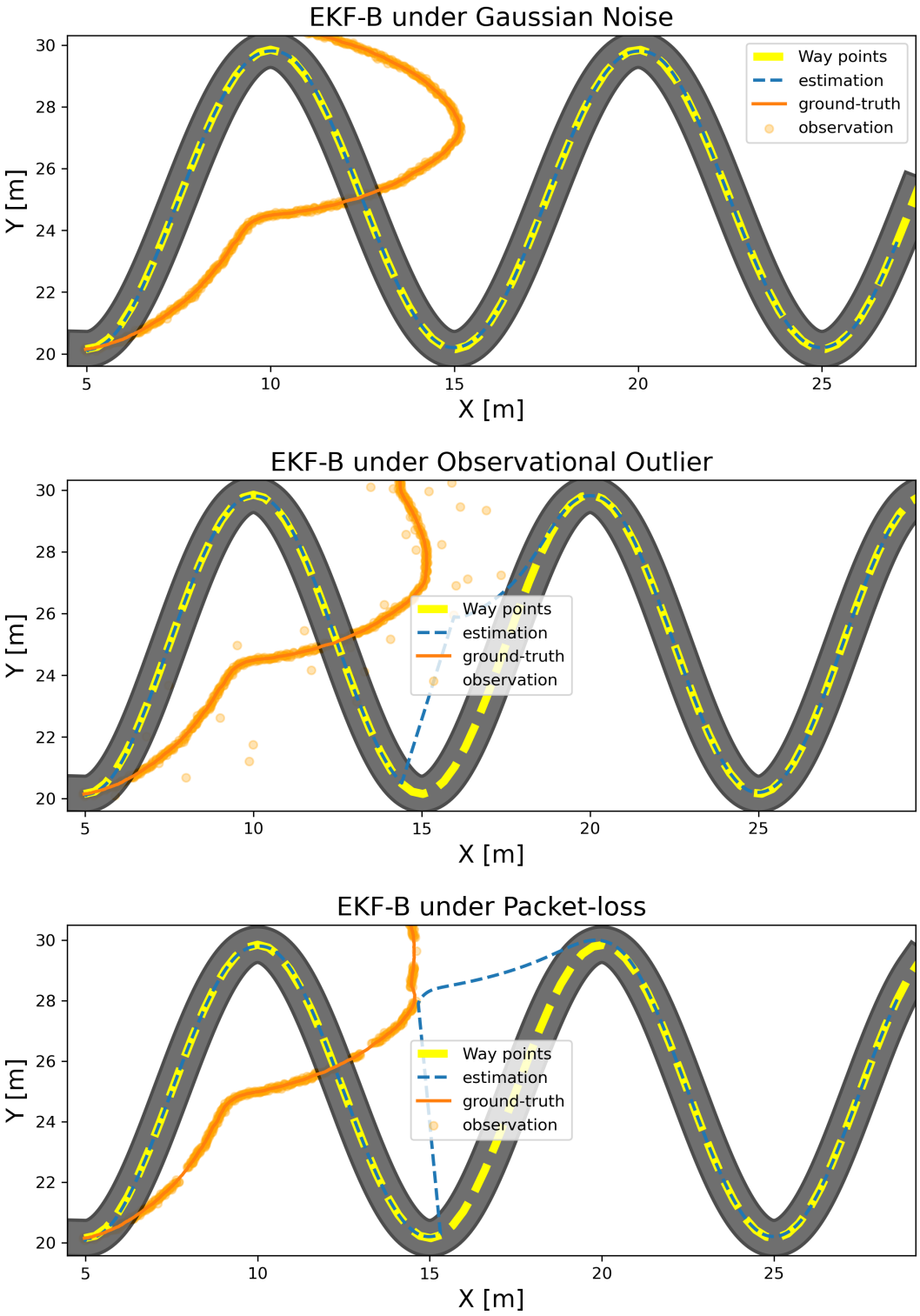


Figure 6.8: The simulation result of the framework with the EKF-B and Stanley method however the process noise covariance is set too small, the observer ignored all the observations, the framework cannot perform path-tracking.

7

Conclusion

In this thesis, an outlier-robust path-tracking control framework was proposed to address the challenges in the autonomous vehicle platform at DCSC. The proposed framework consists of three main modules: a simulator that models the motion of the Radio-Control (RC) vehicle, an adaptive outlier-robust Kalman filter, and a path-tracking controller.

For the RC vehicle dynamics, a first-order system was used to describe the relationship between the RC input command and the front wheel steering angle and rear wheel angular velocity. Combined with the kinematic relationship of the vehicle motion and the vehicle-body dynamics, the RC vehicle dynamics can capture most of the dynamics of the DCSC vehicle. State and input data were collected during field testing to identify some parameters of the RC vehicle dynamics, improving its reliability. However, discrepancies between the model and the real dynamics remain. More complex dynamics and additional field-testing data are needed for future improvements.

For the adaptive outlier-robust Kalman filter, the process noise covariance is assumed to be time-varying and is estimated during the filtering process. A Beta-Bernoulli distribution was used to model the occurrence of observational outliers. The expectation of the Bernoulli variable is estimated during the filtering process and serves as an observational outlier indicator. The proposed filter was tested under three types of observational disturbances—Gaussian noise, observational outliers, and packet loss—as well as under discrepancies between the process model and the simulation model to evaluate its performance. Compared to non-adaptive outlier-robust Kalman filters, the proposed filter can estimate the reliability of the process model relative to the observations and adjust the process noise covariance during the filtering process. However, since the process noise covariance of the proposed adaptive filters is influenced by the observations, their state estimation may not be as smooth as that of the non-adaptive methods.

During testing of the proposed path-tracking framework, the path-tracking performance was evaluated for different combinations of filters and controllers. The path-tracking performance is highly dependent on the accuracy of the state estimation provided by the filter. In this context, for non-adaptive outlier-robust filters, poorly chosen design parameters can lead to the rejection of all nominal observations, causing the filter to rely solely on predictions from the process model, which results in poor control performance.

In conclusion, the proposed framework is capable of performing path-tracking under model discrepancies and various types of observational disturbances. The proposed filter demonstrates better performance in the simulated environment of the DCSC vehicle platform. The filter is generally applicable to noisy systems and exhibits good robustness. Future development of the proposed filter should focus on improving its efficiency and robustness to Gaussian noise.

A

Adaptive Extended Kalman Filter Inverse Wishart

In this section, the algorithm of the adaptive extended Kalman filter with Inverse Wishart (EKF-IW-adp) is introduced. It assumes that both the process noise covariance and the observation noise covariance are time-varying.

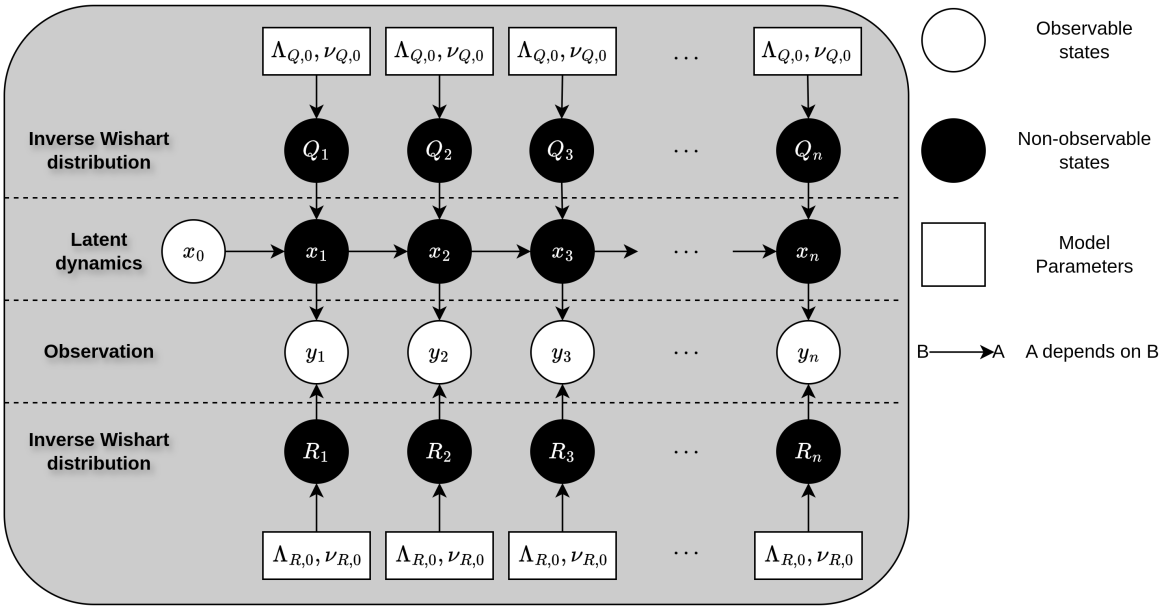


Figure A.1: The graphic model of the system with adaptive process noise covariance and adaptive observation noise covariance

System Description

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t), \quad \mathbf{Q}_t \sim \mathcal{W}^{-1}(\nu_{Q,0}\Lambda_{Q,0}, \nu_{Q,0}) \tag{A.1}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t), \quad \mathbf{R}_t \sim \mathcal{W}^{-1}(\nu_{R,0}\Lambda_{R,0}, \nu_{R,0}) \tag{A.2}$$

Here, $\mathbf{Q}_t$ is drawn from an inverse Wishart distribution with a inverse scale matrix $\Lambda_{Q,0}$ and degree-of-freedom $\nu_{Q,0}$, and $\mathbf{R}_t$ follows an Inverse Wishart distribution with a inverse scale matrix $\Lambda_{R,0} \in \mathbb{R}^{N_y \times N_y}$ and degree-of-freedom $\nu_{R,0} \in \mathbb{R}$.

Joint Distribution and Update Rule

According to the graphic model in Figure A.1, the hidden state of the system is $\Theta := [\mathbf{x}_t, \mathbf{Q}_t, \mathbf{R}_t]$, there are four hyperparameter of the system. Given the Equation (A.1), (A.2), the joint distribution of the latent state

and the observations $\mathbf{y}_{1:t}$ is:

$$p(\mathbf{x}_t, \mathbf{Q}_t, \mathbf{R}_t, \mathbf{y}_{1:t}) = p(\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{R}_t) \cdot p(\mathbf{x}_t \mid \mathbf{Q}_t, \mathbf{x}_{t-1}) \cdot p(\mathbf{Q}_t) \cdot p(\mathbf{R}_t) \tag{A.3}$$

The variational Bayesian is applied here to approximate the posterior of the states, the optimal approximation of the posterior of the states can be expressed as:

$$\ln q_x^*(\mathbf{x}_t) \propto \mathrm{E}_{p(Q_t)p(R_t)} \left[\ln p(\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{R}_t) + \ln p(\mathbf{x}_t \mid \mathbf{Q}_t, \mathbf{x}_{t-1}) \right] \tag{A.4}$$

$$\ln q_Q^*(\mathbf{Q}_t) \propto \mathrm{E}_{p(x_t)p(R_t)} \left[\ln p(\mathbf{x}_t \mid \mathbf{Q}_t, \mathbf{x}_{t-1}) + \ln p(\mathbf{Q}_t) \right] \tag{A.5}$$

$$\ln q_z^*(\mathbf{R}_t) \propto \mathrm{E}_{p(x_t)p(Q_t)} \left[\ln p(\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{R}_t) + \ln p(\mathbf{R}_t) \right] \tag{A.6}$$

The update rule as shown in Algorithm 4.

Algorithm 4 Adaptive Extended Kalman Filter with Inverse-Wishart Distribution:

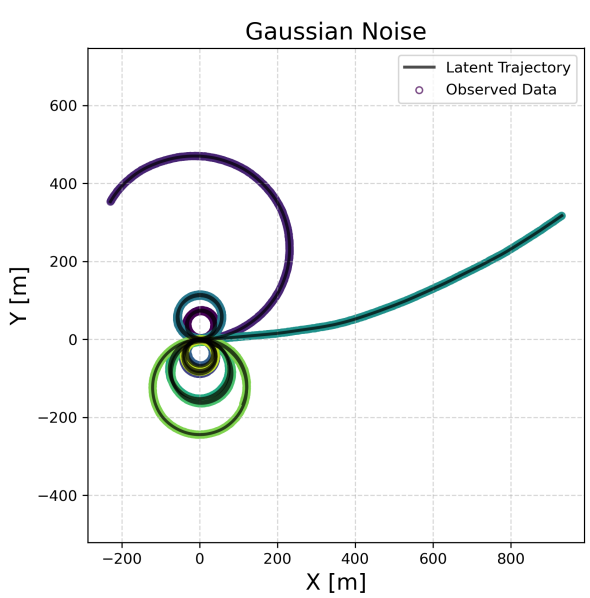
Input: μ_{t-1}, Σ_{t-1}, $f(\cdot)$, $\mathbf{F}_x(\cdot)$, $h(\cdot)$, $\mathbf{H}_x(\cdot)$, $\mathbf{y}_t$, $\Lambda_{Q,0}$, $\Lambda_{R,0}$, $\nu_{Q,0}$, $\nu_{R,0}$
Output: μ_t, Σ_t
Initialize
$\Lambda_{Q,t} \leftarrow \Lambda_{Q,0}$
$\Lambda_{R,t} \leftarrow \Lambda_{R,0}$
$\nu_{Q,t} \leftarrow \nu_{Q,0}$
$\nu_{R,t} \leftarrow \nu_{R,0}$
repeat
  Prediction
  $\mu_{t|t-1} \leftarrow f(\mu_{t-1})$
  $\Sigma_{t|t-1} \leftarrow \mathbf{F}_x(\mu_{t-1}) \cdot \Sigma_{t-1} \cdot \mathbf{F}_x(\mu_{t-1})^T + \Lambda_{Q,t}$
  Update Inverse Wishart parameters
  $\epsilon_t \leftarrow \mathbf{y}_t - h(\mu_t)$
  $\mathbf{B}_t \leftarrow \mathrm{E}[\epsilon_t \cdot \epsilon_t^T]$
  $\nu_{R,t} \leftarrow \nu_{R,0} + 1$
  $\Lambda_{R,t} \leftarrow \frac{\mathbf{B}_t + \nu_{R,0} \cdot \Lambda_{R,0}}{\nu_{R,0}+1}$
  Step 2.2: Update state mean and covariance
  $\mathbf{S}_t \leftarrow \mathbf{H}_x(\mu_{t|t-1}) \cdot \Sigma_{t|t-1} \cdot \mathbf{H}_x(\mu_{t|t-1})^T + \Lambda_{R,t}$
  $\mathbf{K}_t \leftarrow \Sigma_{t|t-1} \cdot \mathbf{H}_x(\mu_{t|t-1})^T \cdot \mathbf{S}_t^{-1}$
  $\mu_t \leftarrow \mu_{t|t-1} + \mathbf{K}_t \cdot \left[\mathbf{y}_t - h(\mu_{t|t-1}) \right]$
  $\Sigma_t \leftarrow \Sigma_{t|t-1} - \mathbf{K}_t \cdot \mathbf{S}_t \cdot \mathbf{K}_t^T$
  Update process noise covariance
  $\epsilon_t \leftarrow \mathbf{y}_t - h(\mu_t)$
  $\mathbf{B}_t \leftarrow \mathrm{E}[\epsilon_t \cdot \epsilon_t^T]$
  $\nu_{Q,t} \leftarrow \nu_{Q,0} + 1$
  $\Lambda_{Q,t} \leftarrow \frac{\mathbf{B}_t + \nu_{Q,0} \cdot \Lambda_{Q,0}}{\nu_{Q,0}+1}$
until Convergence of μ_t and Σ_t
Return: μ_t, Σ_t

B

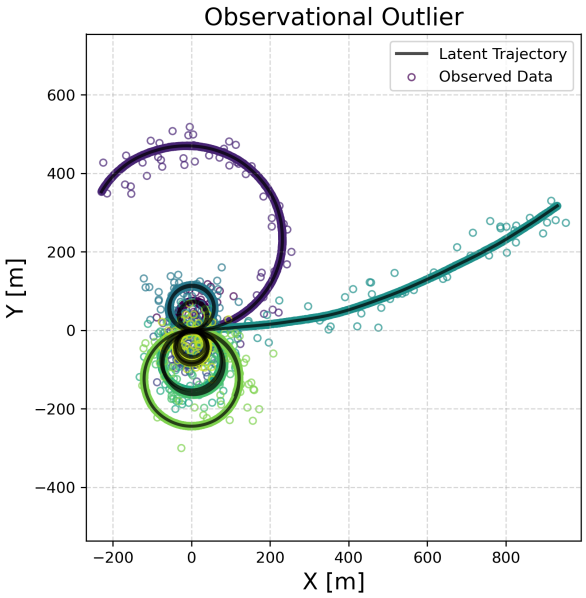
Synthetic Datasets and the Performance of Observers

In Chapter 5, part of the experimental results are presented. Some results, which can occupy an entire page, have been moved to the Appendix to ensure consistency and readability. These include:

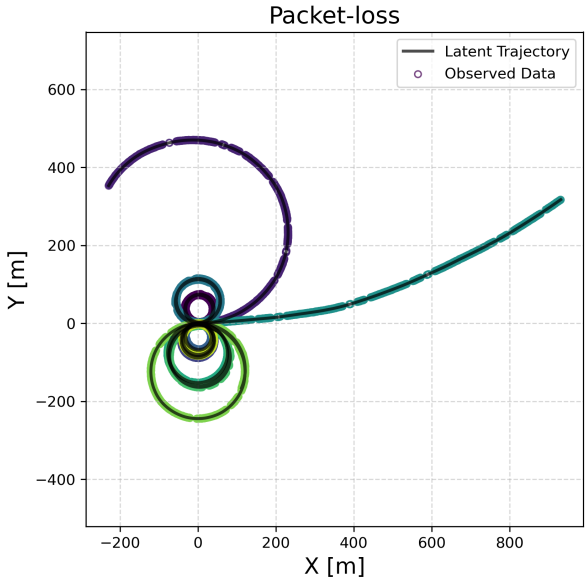
- The full trials of the synthetic datasets generated by the kinematic bicycle model (Figure B.1).
- The full trials of the synthetic datasets generated by the Identified model (Figure B.2).
- The estimation results from the observers over the dataset (kinematic bicycle model) (Figure B.3).
- The estimation results from the observers over the dataset (Identified model) (Figures B.4 and B.5).



(a) Synthetic dataset with Gaussian noise

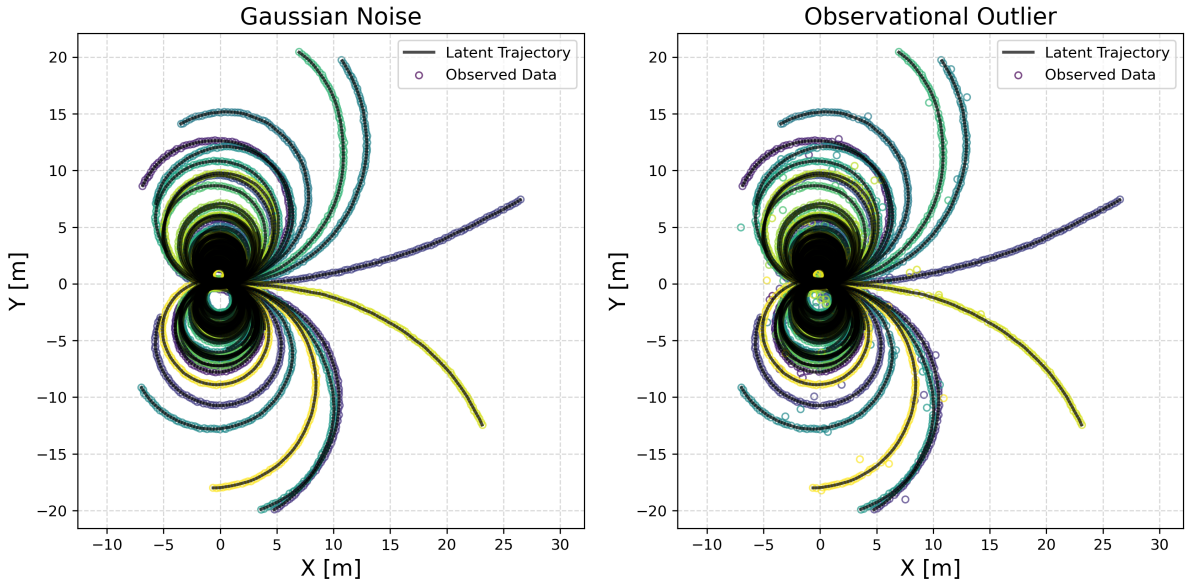


(b) Synthetic dataset with Observational outliers



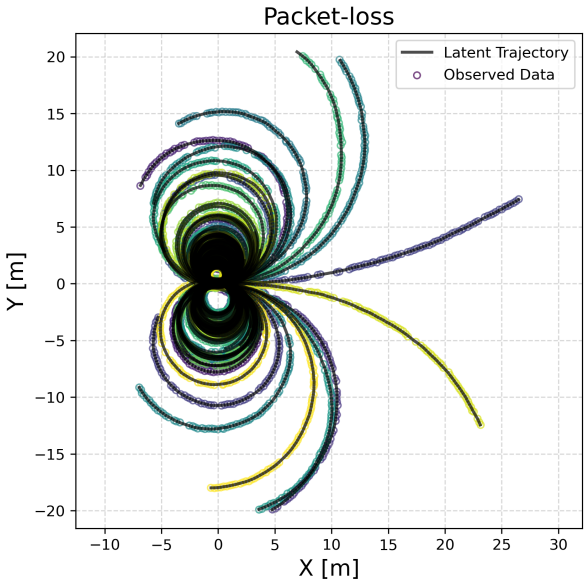
(c) Synthetic dataset with Packet-loss

Figure B.1: The synthetic dataset generated by the kinematic bicycle model with different types of disturbance. Each dataset contains 100 trails, and there are 1000 time steps in each trail.



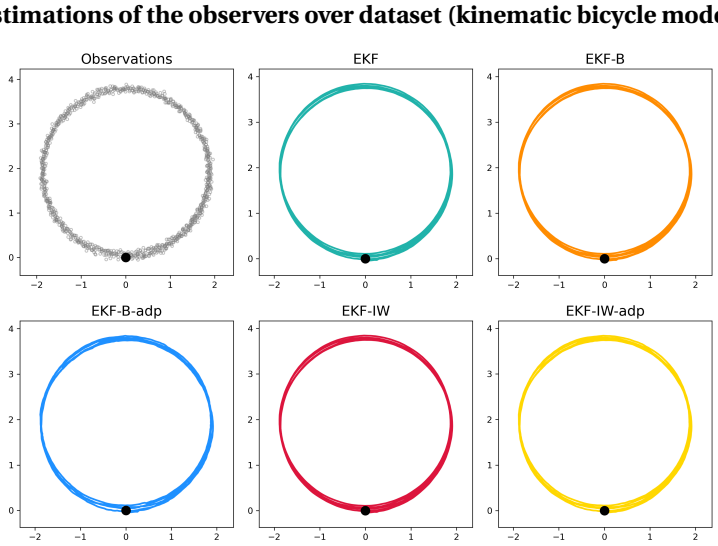
(a) Synthetic dataset with Gaussian noise

(b) Synthetic dataset with Observational outliers

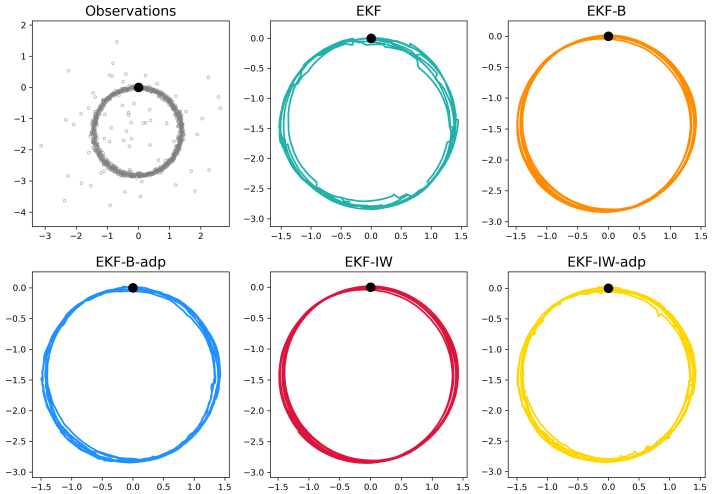


(c) Synthetic dataset with Packet-loss

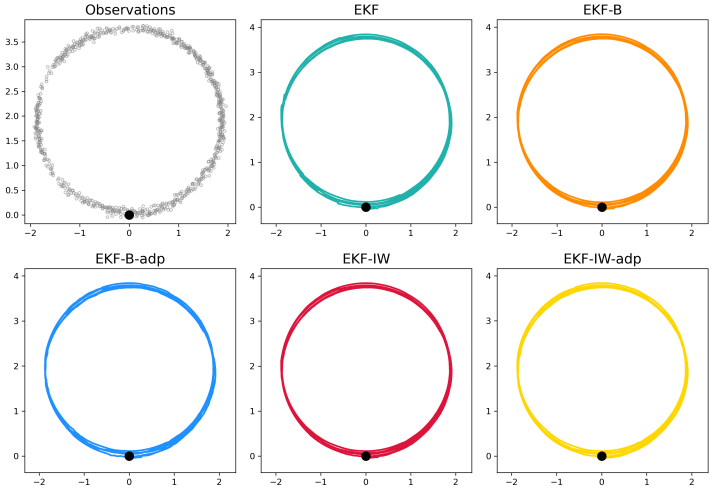
Figure B.2: The synthetic dataset generated by the realistic vehicle model with different types of disturbance. Each dataset contains 100 trails, and there are 1000 samples in each trail.



(a) The state estimation of different filters on a single trail of the dataset with Gaussian noise



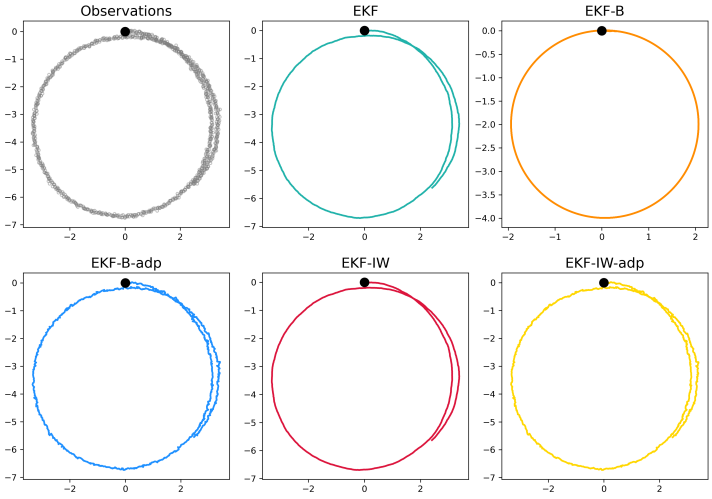
(b) The state estimation of different filters on a single trail of the dataset with Observational outliers



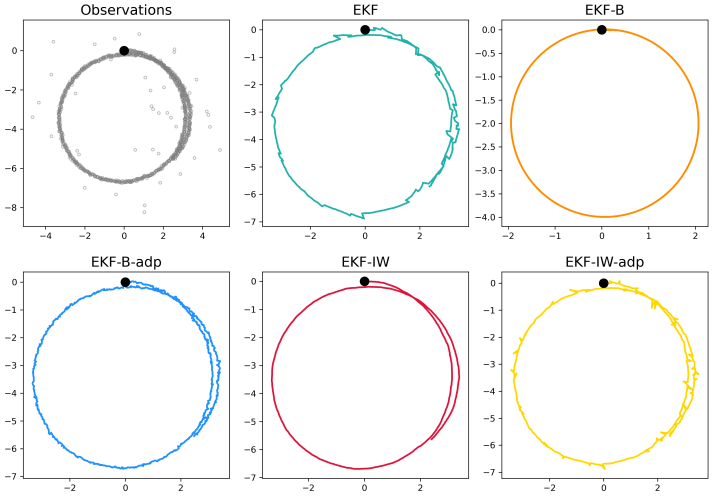
(c) The state estimation of different filters on a single trail of the dataset with Packet-loss

Figure B.3: The state estimation of different filters with under different types of disturbance, when the process model is perfectly aligned with the simulator model

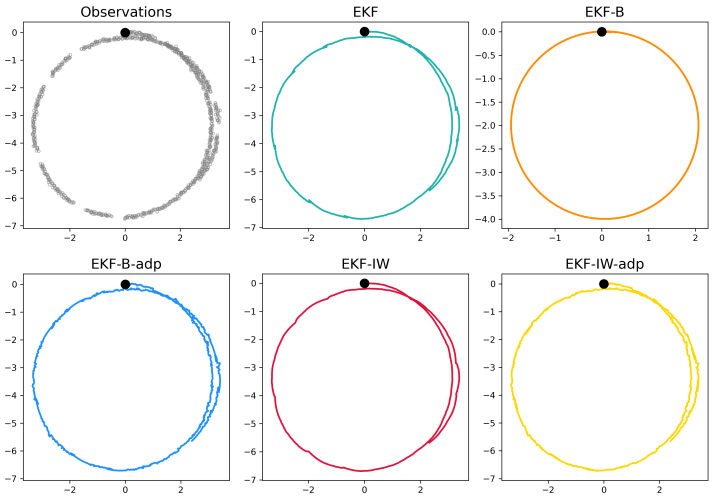
Estimations of the observers over dataset (Identified model) with $||\mathbf{Q}||_2 = 10^{-5}$



(a) The state estimation of different filters on a single trail of the dataset with Gaussian noise

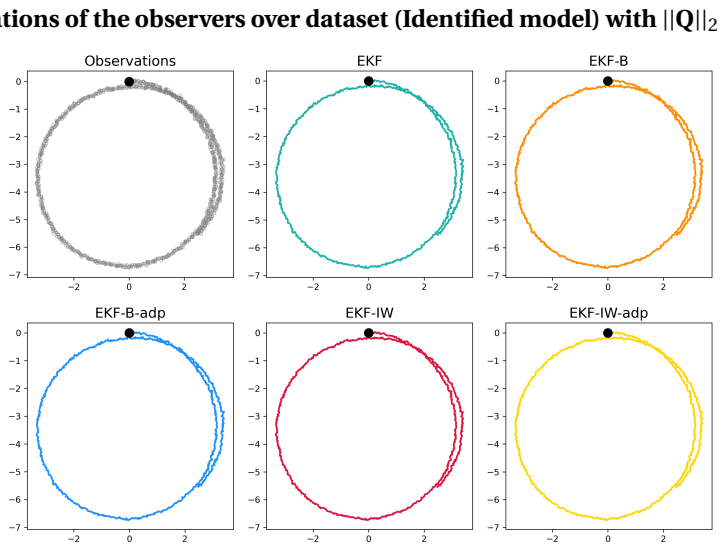


(b) The state estimation of different filters on a single trail of the dataset with Observational outliers

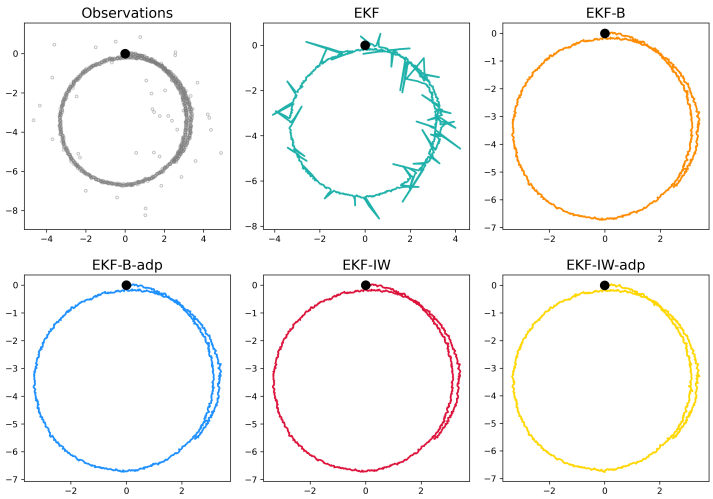


(c) The state estimation of different filters on a single trail of the dataset with Packet-loss

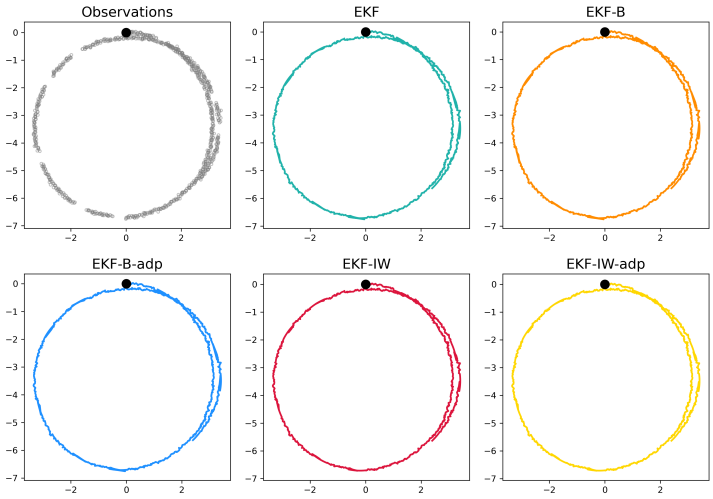
Figure B.4: The state estimation of observers under different types of disturbance over the dataset (Identified model) when the prior process noise covariance is low $||\mathbf{Q}||_2 = 10^{-5}$.



(a) The state estimation of different filters on a single trail of the dataset with Gaussian noise



(b) The state estimation of different filters on a single trail of the dataset with Observational outliers



(c) The state estimation of different filters on a single trail of the dataset with Packet-loss

Figure B.5: The state estimation of observers under different types of disturbance over the dataset (Identified model) when the prior process noise covariance is large $||\mathbf{Q}||_2 = 10^{-3}$.

Bibliography

[1] Gabriel Agamennoni, Juan I. Nieto, and Eduardo M. Nebot. An outlier-robust kalman filter. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2011. doi: 10.1109/ICRA.2011.5979605.

[2] Gabriel Agamennoni, Juan I. Nieto, and Eduardo M. Nebot. Approximate inference in state-space models with heavy-tailed noise. *IEEE Transactions on Signal Processing*, 60, 2012. ISSN 1053587X. doi: 10.1109/TSP.2012.2208106.

[3] Shahrokh Akhlaghi, Ning Zhou, and Zhenyu Huang. Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation. In *2017 IEEE Power Energy Society General Meeting*, volume 2018-January, pages 1–5, 2017. doi: 10.1109/PESGM.2017.8273755.

[4] Noor Hafizah Amer, Hairi Zamzuri, Khisbullah Hudha, and Zulkiffli Abdul Kadir. Modelling and control strategies in path tracking control for autonomous ground vehicles: A review of state of the art and challenges. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 86, 2017. ISSN 15730409. doi: 10.1007/s10846-016-0442-0.

[5] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A. Lozano. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys*, 54, 4 2021. ISSN 15577341. doi: 10.1145/3444690. URL `https://doi.org/10.1145/3444690`.

[6] R. Craig Coulter. Implementation of the pure pursuit path tracking algorithm, 1992. URL `https://api.semanticscholar.org/CorpusID:62550799`.

[7] Gerardo Duran-Martin, Matias Altamirano, Alexander Y. Shestopaloff, Leandro Sánchez-Betancourt, Jeremias Knoblauch, Matt Jones, François-Xavier Briol, and Kevin Murphy. Outlier-robust kalman filtering through generalised bayes. 5 2024. URL `http://arxiv.org/abs/2405.05646`.

[8] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Object detection and tracking for autonomous navigation in dynamic environments. *International Journal of Robotics Research*, 29, 2010. ISSN 02783649. doi: 10.1177/0278364910365417.

[9] Center for Sustainable Systems. Autonomous vehicles factsheet. Technical report, University of Michigan, 2024.

[10] Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han. Outlier detection for temporal data: A survey, 9 2014. ISSN 10414347.

[11] Gabriel M. Hoffmann, Claire J. Tomlin, Michael Montemerlo, and Sebastian Thrun. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In *Proceedings of the American Control Conference*, 2007. doi: 10.1109/ACC.2007.4282788.

[12] The MathWorks Inc. System identification toolbox (r2024b), 2024. URL `https://www.mathworks.com`.

[13] Reza N. Jazar. *Vehicle dynamics: Theory and applications*. 2008. doi: 10.1007/978-0-387-74244-1.

[14] Maximilian Kloock, Patrick Scheffe, Janis Maczijewski, Alexandru Kampmann, Armin Mokhtarian, Stefan Kowalewski, and Bassam Alrifaee. Cyber-physical mobility lab: An open-source platform for networked and autonomous vehicles. In *2021 European Control Conference, ECC 2021*, 2021. doi: 10.23919/ECC54610.2021.9654986.

[15] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *IEEE Intelligent Vehicles Symposium, Proceedings*, volume 2015-August, 2015. doi: 10.1109/IVS.2015.7225830.

[16] Sang Gyu Kwak and Jong Hae Kim. Central limit theorem: The cornerstone of modern statistics. *Korean Journal of Anesthesiology*, 70, 2017. ISSN 20057563. doi: 10.4097/kjae.2017.70.2.144.

[17] Steve Macenski, Shrijit Singh, Francisco Martín, and Jonatan Ginés. Regulated pure pursuit for robot path tracking. *Autonomous Robots*, 47, 2023. ISSN 15737527. doi: 10.1007/s10514-023-10097-6.

[18] Dan Navarro and Amy Perfors. An introduction to the beta-binomial model, 2014.

[19] Norman S. Nise. Control system engineering (8th edtion). *Wiley*, 2019.

[20] Steven W Nydick. The wishart and inverse wishart distributions. *Electronic Journal of Statistics*, 6:1–19, 2012.

[21] Matthew O'Kelly, Varundev Sukhil, Houssam Abbas, Jack Harkins, Chris Kao, Yash Vardhan Pant, Rahul Mangharam, Dipshil Agarwal, Madhur Behl, Paolo Burgio, and Marko Bertogna. F1/10: An open-source autonomous cyber-physical platform. 1 2019. URL `http://arxiv.org/abs/1901.08567`.

[22] Myung Wook Park, Sang Woo Lee, and Woo Yong Han. Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm. In *International Conference on Control, Automation and Systems*, 2014. doi: 10.1109/ICCAS.2014.6987787.

[23] Robert Piché, Simo Särkkä, and Jouni 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. doi: 10.1109/MLSP.2012.6349794.

[24] Philip Polack, Florent Altche, Brigitte DAndrea-Novel, and Arnaud De La Fortelle. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *IEEE Intelligent Vehicles Symposium, Proceedings*, 2017. doi: 10.1109/IVS.2017.7995816.

[25] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3, 1986. ISSN 07407467. doi: 10.1109/MASSP.1986.1165342.

[26] Rajesh Rajamani. *Vehicle Dynamics and Control (Mechanical Engineering Series)*. 2012.

[27] Joram Soch. The book of statistical proofs, 2024. URL `StatProofBook/StatProofBook.github.io`.

[28] Simo Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013. ISBN 9781139344203. doi: 10.1017/CBO9781139344203.

[29] Jacopo Tani, Liam Paull, Maria T. Zuber, Daniela Rus, Jonathan How, John Leonard, and Andrea Censi. Duckietown: An innovative way to teach autonomy. *Advances in Intelligent Systems and Computing*, 560, 2017. ISSN 21945357. doi: 10.1007/978-3-319-55553-9_8.

[30] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23, 2006. ISSN 15564959. doi: 10.1002/rob.20147.

[31] Dimitris G. Tzikas, Aristidis C. Likas, and Nickolaos P. Galatsanos. The variational approximation for bayesian inference: Life after the em algorithm. *IEEE Signal Processing Magazine*, 25:131–146, 2008. ISSN 10535888. doi: 10.1109/MSP.2008.929620.

[32] Marjolein van Lierop. *Model Predictive Control for Vehicle Platooning A Practical Comparison against Traditional Methods*. PhD thesis, 2022. URL `https://resolver.tudelft.nl/uuid:520ba179-2293-4049-a14f-40db52aa7e54`.

[33] Hongwei Wang, Hongbin Li, Jun Fang, and Heping Wang. Robust gaussian kalman filter with outlier detection. *IEEE Signal Processing Letters*, 25, 2018. ISSN 10709908. doi: 10.1109/LSP.2018.2851156.

[34] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8, 2020. ISSN 21693536. doi: 10.1109/ACCESS.2020.2983149.