Master's Thesis of Engineering

Fail-Safe Algorithm for Sensors of Autonomous Vehicle

자율주행 자동차 센서의 고장 진단 알고리즘

August 2018

Graduate School of Engineering
Seoul National University
Mechanical Engineer Major

Choi, Seung-rhi

Abstract

This paper presents a fail-safe algorithm for extetoceptive sensors of autonomous vehicle. The proposed fault diagnosis mechanism consists of three parts: 1) fault detecting by duplication-comparison method, 2) fault isolating by possible area prediction and 3) in-vehicle sensor fail-safe. The main ideas are the usage of redundant external sensor pairs, which estimates the same target, whose results were compared to detect the fault. The faults are detected by modified duplication-comparison method and the novel fault isolation method using target predictions. By comparing the estimations of surrounding vehicles and the raw measurement data, the location of faults can be determined whether it is from sensors themselves or a software error. In addition, faults were isolated by defining Possible Area that measurements could exist on sensor coordinate, which can be predicted by using previous estimation results. The performance of the algorithm has been tested by using offline vehicle data analysis via Matlab. Various fault injection experiment were conducted and the performance of the suggested algorithm was evaluated based on the time interval between injection and the detection of faults.

Keyword : Autonomous Vehicle, Automotive Radar Sensor, LIDAR sensor, Fail-detection, Fault-Diagnosis, Fault Injection Test

Student Number : 2016-26830

Table of Contents

Chapter 1. Introduction

1.1. Study Background

Health monitoring or fault-tolerant control systems is one of the most essential elements for control systems. When an automated system operates, there is always a chance that system failure occurs. It can be broken hardware (e.g. crack in the bolt cutting it), deficiency in software or an error in sensors. It is critical because these somewhat trivial breakdowns can cause dangerous situations. Therefore, there have been many studies on fault-detection and fault-tolerant control system.

Health monitoring and fault-tolerant control systems can be divided into three main processes; fault detection, fault diagnosis and fault tolerant control. The overall processes are illustrated in Figure 1. First of all, fault should be detected. However, it is not an easy task because forecasting every possible fault and their effect is almost impossible. Nevertheless, there exist several methods to analyze fault and their effects. One of these methods is Fault Propagation Analysis (FPA) using FMEA matrix[1]. It divides system into the parts that performs the same function and analyzes

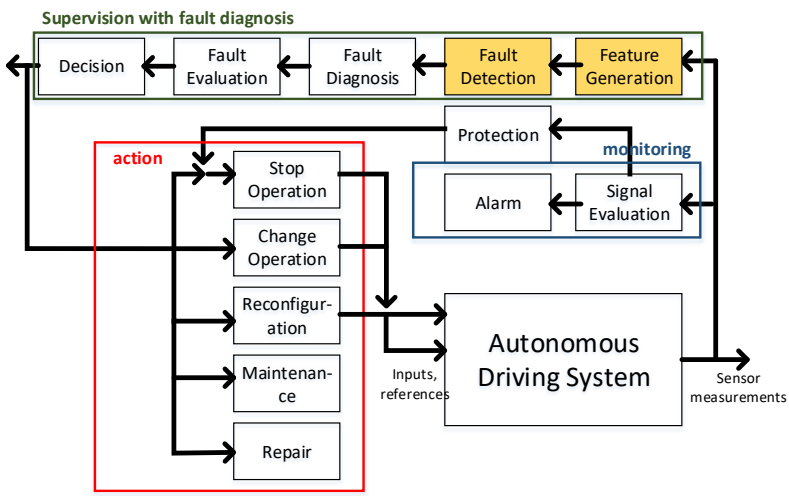


Figure 1 designing fault-tolerance control system process

how failure of one elements can affect the entire system. Blanke[1] applied the method to the 3-level valve control system and it is usually applied to hardware system. In this paper, the method was applied to the software of autonomous driving system in order to see how one of sensors' failure affects the entire autonomous driving system. The example was described in Figure 2. However, pioneers in this field have figure out two effective ways which are using limit checking[2] and using redundancy relations[3]. Both generating features from output signals of control system, but the differences lie between whether it uses direct signal properties or redundant relations. Second, faults are isolated and their failure effects are classified into hazard classes in fault-diagnosis process. There are also a lot of method to classify fault effects, but they were not investigated deeply in this paper. Lastly, after faults are detected and evaluated, fault-tolerant control should be designed.

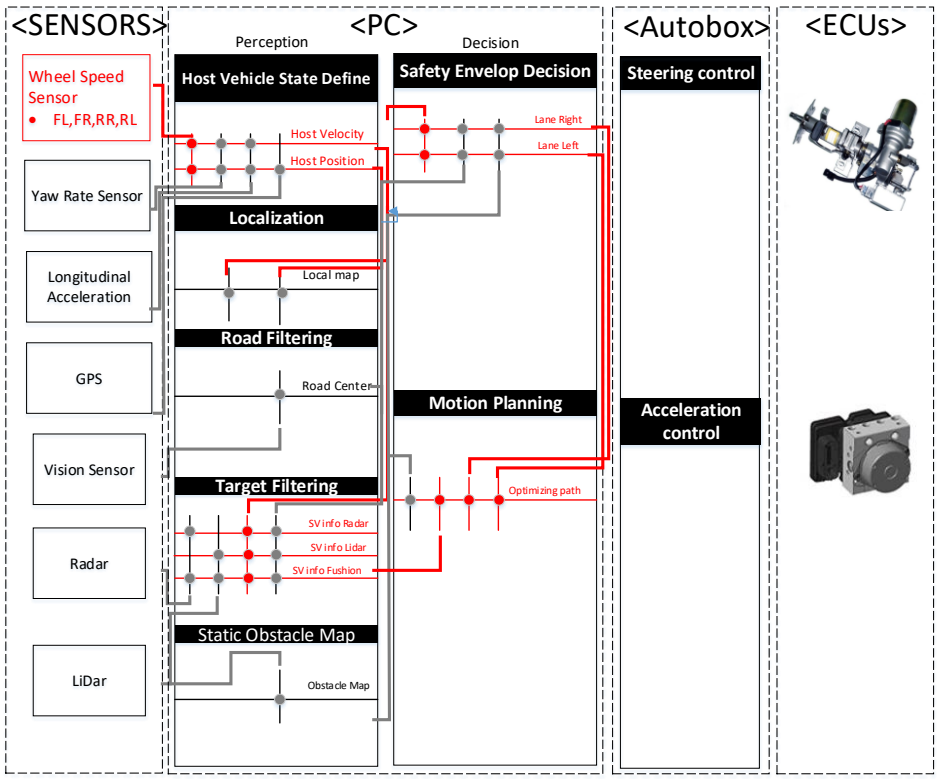


Figure. 2 autonomous driving system FPA analysis

Several fault-tolerant control designs were introduced in [3].

1.2. Purpose of Research

Sensors have been the most vulnerable elements of control system. Therefore, their failure effect should be investigated when our developing reliable autonomous driving system. As many driver assistant systems like Adaptive Cruise Control (ACC), Lane Keeping Assistant System (LKAS), Autonomous Emergency Braking (AEB) were already on the market, some studies on fail-safe algorithm have conducted. For instance, Y. Jeong et al[4] vehicle sensors and actuator fault detection algorithm for automated vehicles. He used various sensor pairs and three different vehicle models to estimate vehicle's yaw rate. He compared the results of estimation with yaw rate measurement from yaw rate sensor and figured out the abnormal behavior in sensors. He also uses adaptive threshold to mitigate the sensitivity of the algorithm.

To guarantee the reliability of perception system, there are also many researches on fault detection of Exteroceptive sensors. For example, Bader[5] proposed a fault tolerant architecture for data fusion. The proposed architecture consists of two parts; error detection and error isolation. In error detection module, an error in fusion system detected by comparing estimation results from two different kalman filter branches. Then, the error is isolated using kalman filter residuals and sensor measurements. Through this approach, he detected abnormal behaviors in GPS sensor. In addition, R. Chandail[6] designed fault-tolerant vision odometry system. In the paper, he suggested an architecture which consists of several nodes for fault detection and isolation. With results of these fault-diagnosis nodes, he magnified corresponding measurement noise variance in order to gain robustness in EKF estimation. Lastly, K. Oh[7] suggested probabilistic fault detection and diagnosis algorithm for measuring preceding vehicle with Radar sensor. His main idea is using sliding mode observer with a longitudinal kinematic model to find preceding and ego vehicles'

acceleration. After finding accelerations, he checked whether they lie in appropriate range of acceleration or not.

However, there are few studies on radar sensor's failure and their detection algorithms although they play an important role in autonomous driving system. Therefore, this paper aims to design fault-detection and diagnosis algorithm for automotive radar sensor of autonomous vehicles. Sensors used in automated car were presented and explained in Chapter 2. In Chapter 3, fail-safe algorithm proposed in this paper was described. Chapter 4 explain the concept of functional safety and reliability evaluation. All simulation and simulation results were described in Chapter 5. Finally, Chapter 6 concluded paper and discussed about future work.

Chapter 2. Sensors of Autonomous Vehicle

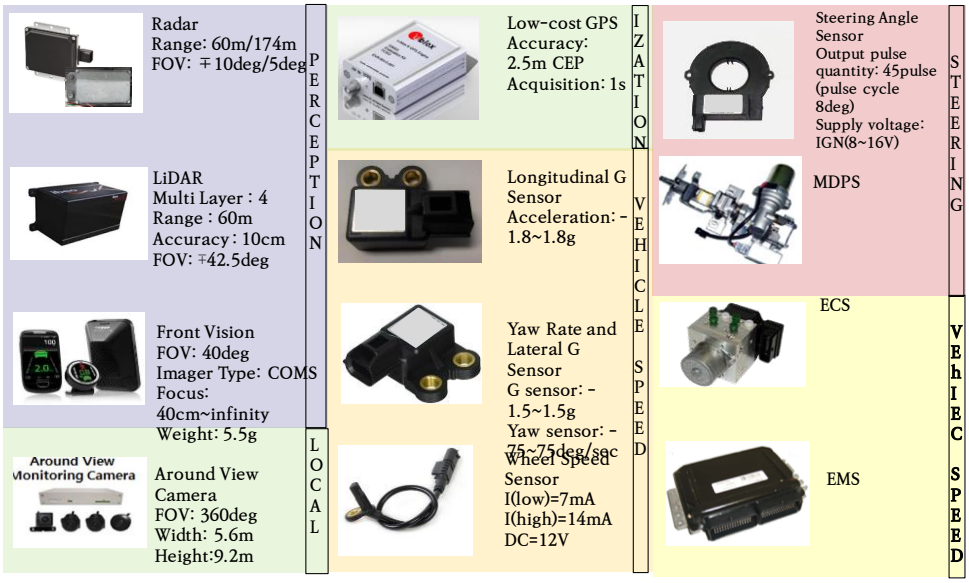


Figure. 3 Sensors of autonomous driving system

One of the most important parts of autonomous driving technology is perception. In order for a car to drive by itself, it is essential to know where it is and what there are around it. Therefore, various sensors have been used for automated cars as described in Figure 3. Sensors were classified according to their usage. Range sensors (i.e. automotive radar sensor and LiDAR sensor) and vision sensor provide a wealth of information about object and situation around the vehicle. AVM camera and low-cost GPS also give information about the environment where the vehicle drive, its usually used for locating vehicle's position. There are some sensors for estimating the state of vehicle. Four wheel speed sensors, an acceleration sensor, and yaw rate sensor are usually the main measurements to estimate vehicle's velocity, acceleration and yaw rate. Lastly, there are sensors and actuators for steering control and vehicle speed control.

2.1. Automotive Radar Sensor and LiDAR Sensor

Radar sensor is a sensor originally used to detect unidentified

flying object at the border or for airplane to avoid dangerous crashes. However, it started to be in automotive industry. Especially it is used for driving assistant systems such as Adaptive Cruise Control (ACC) and Autonomous Emergency Braking (AEB) because it is the most appropriate sensor to sense around objects. Radar sensor is highly technology-intensive. It consists of two parts; transmitter which radiates electromagnetic waves and receiver which receives reflected signals and processes them. It measures distance to objects by calculating round-trip duration of radio wave and relative velocity of objects by analyzing frequency difference between emitted and reflected signal. In addition, various data processing technology are used. For example, radar sensors mitigate clutter effect by analyzing power of reflected signals and classify them by their characteristic. Radar sensors are also able to track specific object using their unique signal processing method. Lastly but not least, it can not only operate independent on the weather condition but also penetrate most materials. Therefore, they have been verified to be compatible sensors for autonomous vehicle and widely used until now.

LIDAR is laser scanner which operates using laser instead of radio wave. Although LIDAR sensor is more susceptible to weather condition, it has a higher accuracy and finer range resolution. Also, environments can be represented in 3D map using LIDAR sensor. Thus, LIDAR sensor is being considered useful for autonomous driving. Especially, it is useful when autonomous cars drive in more complicated environment. There are various types of LiDAR on the market. IBEO' s four-layer LIDAR was used in this paper.

2.2. In-Vehicle Sensors

Wheel speed sensors, acceleration sensor, and yaw rate sensor were used to estimate vehicle's current state. In our car, four-wheel speed sensors are mounted at each wheel. Wheel speed sensor uses Magneto elasticity. To be specific, magnet and coil are installed inside the sensor while ferromagnetic saw-toothed wheel

is equipped inside wheel frame. When car's driving, the ferromagnetic saw-toothed wheel rotates with the wheel. Then, the rotation of the saw-toothed wheel induces magnetoelectricity in the coil. Since the saw-toothed wheel has a regular interval between each tooth, the induced electricity has specific frequency. This frequency can be analyzed to get the speed of the wheel.

The acceleration sensor, also called accelerometer, measures the force produced by acceleration. And, there are various ways to measure the acceleration.

Yaw rate senses the Coriolis acceleration. When the vehicle has yaw velocity, the acceleration takes place in the y-axis direction because of Coriolis Effect. Thus, yaw sensor senses longitudinal acceleration.

2.3. Other Sensors and Actuators

In addition to the above sensors, GPS sensors, AVM camera, steering angle sensor, and etc. are used in autonomous driving system. For example, GPS is a system which allows us to know our location, speed and time using satellite. Three satellites are require to locate our current position. In addition, AVM camera, it stands for Around View camera Module, consists of 4 camera which mounted various parts of vehicle. AVM camera provides us a top view of vehicle so that the lanes and road marks can be easily seen. Lastly, steering angle sensor measures relative angle using sensor's magnetic field. When the steering column rotates, a multipole magnet generates a square signal (Hall Effect).

2.4. Sensor Spec

The automotive radar sensor is the product of Delphi Technologies. Its operating frequency is 77 Hz. it consists of long-range radar and mid-range radar. Long-range radar can detect object within up to 174 m. However, long-range radar has narrower Field of View (FOV) while short-range radar has wider

FOV.

LIDAR (Light Detection and Ranging) sensor is the product of ibeo automotive system. It is a 2D/4 layer laser scanner, which has 3.2 azimuth detection range. And, it has +/−42.5 degree FOV.

For acceleration sensors, it is possible to measure from −1.8 g to +1.8 g.

And, yaw rate sensor are able to measure from −1.5 g to +1.5 g.

Wheel speed sensor operates at a rated voltage of 12V. When it measures the speed of wheel, current is between 7mA and 14 mA.

Lastly, steering wheel sensors has 45 output pulse quantity, which equals to pulse cycle 8 degree.

Chapter 3. Fail-Safe Algorithm for Sensors of Autonomous Vehicle

The fail-safe algorithm for automotive radar sensor of autonomous vehicle is represented in this chapter. The algorithm is designed to detect and isolate faults so as to help the autonomous driving system drive in a steady manner even in case of trouble.

3.1. Main Strategy for Detecting Sensor Failure

Many fault-detection systems utilize redundant relations in the system. However, for range sensors like automotive radar sensors, it is hard to find redundant relation. It's because not only range sensors senses arbitrary objects around the vehicle but also it's hard to predict their faults and faults effects. It would be difficult if we try to deal with every possible fault when designing fault-detection system. Thus, what needs to be considered is whether or not it causes an important failure in the system. Range sensors senses surrounding environment. They distinguish the stopping obstacles and secure the path for the vehicle to be driven. They also determine where to go by detecting vehicles and pedestrians. Since automotive radar sensors and LIDAR sensors detect the same objects, the redundancy relation can be found as describe in Figure

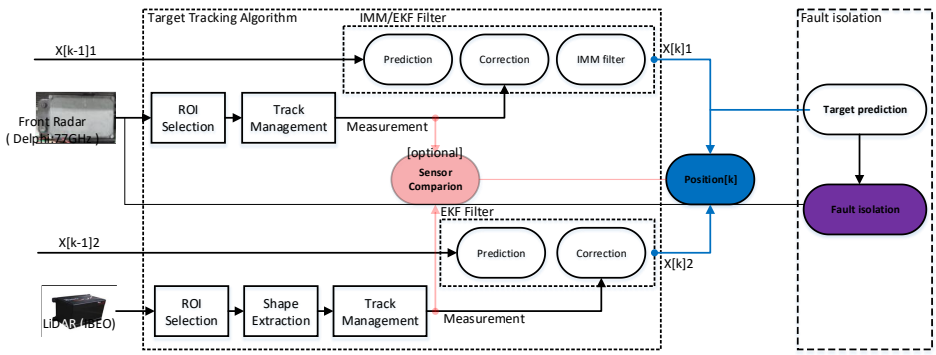


Figure 4 fault-detection architecture

4. There are some objects like vehicle which are detected using both automotive radar sensor and LIDAR sensor. These objects provide redundancy in the system.

The main idea for fault detection is comparing position difference between estimated surrounding vehicles by automotive radar sensor and those by LIDAR sensor. Each sensor has its own tracking mechanism as described in Figure 4. By comparing these results, the algorithm detects system faults caused by either of these sensors. The algorithm also allows you to compare sensor readings to determine if the fault was caused by the software[5]. After detection, the faults should be isolated. The main idea for fault isolation is predicting the future behavior of surrounding vehicles and checking if the sensors has detected the vehicles or not. Although this is kind of ad-hoc procedure, it is an effective way to confirm the sensor failure because there is no reliable information other than the past when the sensor fails.

3.2. Automotive Radar Sensor Fail-Safe Algorithm

The entire architecture of the algorithm is descried in Figure 5. As descried in the figure, the fail-safe algorithm receives and processes surrounding vehicle information from tracking algorithm. First, the algorithm identifies and numbers the surrounding vehicle, which is called "Target Selection". The vehicles are classified by their relative distance from the ego vehicle as illustrated in Figure 6. The 12 blue squares on the host vehicle in Figure 6 indicate the assumed measurement model[8]. With this assumption, the

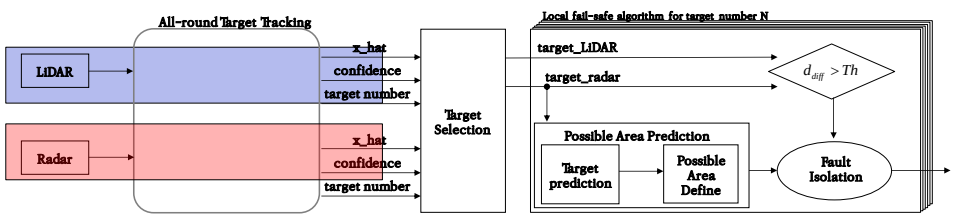


Figure 5 structure of fault detection algorithm

algorithm calculates Euclidian distance to each measurement point from the host vehicle and determines the nearest point. Based on this point and lane information, surrounding vehicles are classified into different areas. Finally, the algorithm selects the nearest vehicle to check sensor' s fail-safety. The pink square in Figure 6 indicates the selected vehicle at each area.

Then, the fail-safe algorithm operates for each target. The algorithm starts after the target is confirmed as a real vehicle, which means both sensors recognize the same vehicle during some amount of time. And, the algorithm is terminated when a fault is detected or a new target near at each area is detected and does not restart until another vehicle is recognized as a real vehicle.

If a system fault is detected, the algorithm isolates a fault using target prediction[9]. Predictions of the vehicles show the location of nearby vehicles that would otherwise properly estimated when the sensor had not failed. Thus, there should be a measurement on the edge of predicted vehicles if the sensor is operating normally, which is called Possible Area. Possible Areas defined at each area are described in Figure 7. Each Possible Area is defined as an ellipse. The algorithm inspects the area using the following equation.

$$\frac{(x-x_c)^2}{L_x}+\frac{(y-y_c)^2}{L_y}\le 1 \qquad (1.1)$$

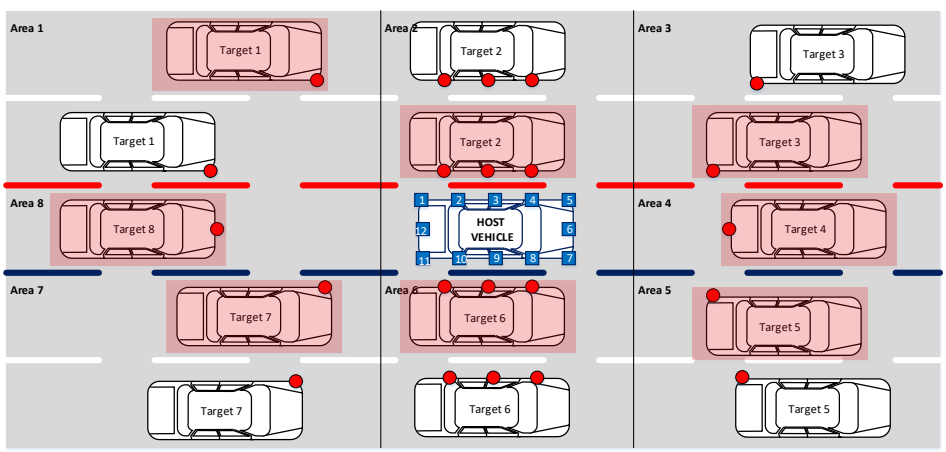


Figure 6 target selection

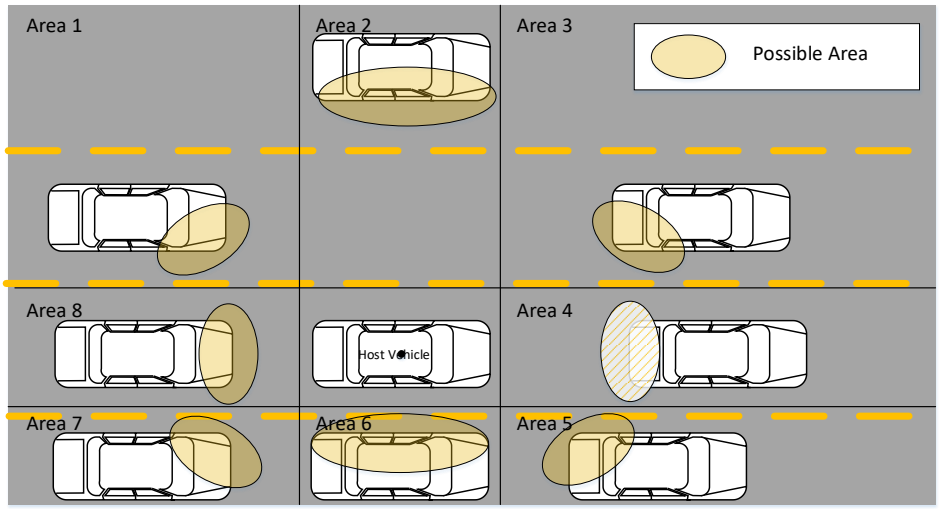


Figure 7 Possible Area

Where, (x, y) denotes the measurement points and (x_c, y_c) denotes the center of Possible Area.

Basically, the algorithm ensures sensor fail safety because it works every time step which is normally within 100 millisecond. In the other word, the algorithm constantly checks whether the system properly detect the surrounding vehicles in a short time once the system has verified that the vehicle is actually around.

However, other sensors are also used to track the surrounding vehicle as described in Figure 8. For example, vision sensors provide lane information in order for the tracking algorithm can select measurement of interest. In-vehicle sensors are used in prediction step to compensate for the distance the host vehicle has

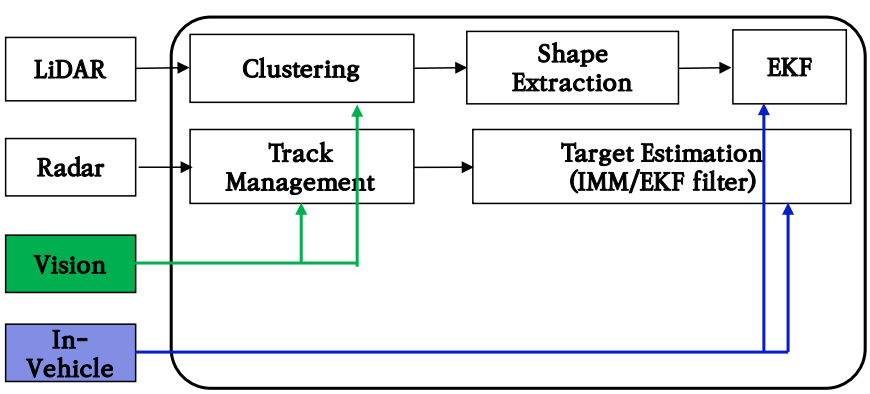


Figure 8 target tracking block diagram

made. Therefore, fault safety algorithm for these two sensors must be preceded.

3.3. In-Vehicle Sensor Fail-Safe Algorithm

The fail-safe algorithm for two rear wheel speed sensors, yaw rate sensor, and acceleration sensor was proposed in this paper. The main idea for fault detection is using Kalman filter and filter's residual. The Kalman filter used in this paper estimates current vehicle states, which represents mainly vehicle's velocity. The augmented state were used in order to mitigate effects of bias in sensors. The augmented state is described as below.

$$x = \begin{bmatrix} v_x & \gamma & a & \dot{\gamma} & b_a & b_\gamma \end{bmatrix}^T \tag{1.2}$$

Where, v_x denotes the longitudinal velocity, γ denotes the yaw rate, a denotes the longitudinal acceleration, $\dot{\gamma}$ denotes the yaw acceleration, b_a denotes the bias in the longitudinal acceleration sensor, and b_γ denotes the bias in the yaw rate sensor. The measurements are represented as below.

$$z = \begin{bmatrix} v_{RL} & v_{RR} & a & \gamma \end{bmatrix}^T \tag{1.3}$$

Where, v_{RL} denotes the rear left wheel speed, v_{RR} denotes the rear right wheel speed, a denotes the longitudinal acceleration, and γ denotes the yaw rate. Also, the process model and the measurement model are described below.

$$x[k+1] = F[k] \cdot x[k] + q[k],\ q[k] \sim (0, Q)$$

$$F = \begin{bmatrix} 1 & 0 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & dt & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.4}$$

$$z[k] = H[k] \cdot x[k] + v[k],\ \ v[k] \sim (0, R)$$
$$H = \begin{bmatrix} 1 & -0.8 & 0 & 0 & 0 & 0 \\ 1 & +0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

Lastly, the Kalman filter is described as below.

$$\begin{aligned} &\hat{x}^{-}[k+1] = F[k] \cdot \hat{x}^{+}[k] \\ &M[k+1] = F[k] \cdot P[k] \cdot F[k]^T + Q \\ &K[k+1] = M[k+1] \cdot H[k+1]^T \cdot (H[k+1] \cdot M[k+1] \cdot H[k+1]^T + R[k+1])^{-1} \\ &\hat{x}^{+}[k+1] = \hat{x}^{-}[k+1] + K[k+1] \cdot (z[k+1] - H[k+1] \cdot \hat{x}^{-}[k+1]) \\ &P[k+1] = (I - K[k+1]H[k+1])M[k+1] \end{aligned} \quad (1.6)$$

By estimating noise measurement variance, a fault is detected and reconstructed. The measurement noise variance estimation is described below.

$$\begin{aligned} R[k] = &\frac{1}{k} r[k] \cdot r[k]^T \\ &+ \frac{k-1}{k} (H[k-1] \cdot M[k-1] \cdot H[k-1] + R[k-1]) \\ &- H[k] \cdot M[k] \cdot H[k] \end{aligned} \quad (1.7)$$

Where, $r[k] = z[k] - H[k] \cdot x[k]$ denotes the residual of the kalman filter.

For the fault detection, there should be a change detection algorithm since the reconstructed fault has stochastic property. CUMSUM algorithm[10] is proposed in this paper. The summary of CUMSUM algorithm is described below.

$$\begin{aligned} &S = \sum_{i=0}^{t_a} \ln \frac{p_{\theta_1}(y_i)}{p_{\theta_2}(y_i)} \\ &g = \max(0, S) \\ &g > threshold \end{aligned} \quad (1.8)$$

Where θ_2 represents a normal state, θ_1 represents a fault state, and p_{θ_1} , p_{θ_2} represent the probability of a state, respectively.

Chapter 4. Safety Evaluation and System Dependability

When designing a control system, it is important to guarantee its functional safety and dependability. One of important roles of fault-detection and fault-tolerant control system is improving system's functional safety and dependability. Therefore, in order for better understanding of the fail-safe algorithms for sensors of autonomous vehicle, this chapter described theoretic background on dependability and safety evaluation.

4.1. Dependability and Safety Evaluation Methods

Reliability is defined as component or system's ability to function correction under certain period and conditions. Safety is the ability that a system does not cause danger to person or environment. Lastly, integrity or safety integrity refers to a system's ability to properly operate safety-related system or cover faults which may cause dangerous situation. If a system satisfies all of these properties, it can be said that the system is

Table 1 safety-related concepts

property	definition	measure
Reliability	Ability of a system to perform a required function under stated conditions, within a given scope, during a given period	MTTF
Safety	Ability of a system not to cause danger to persons or equipment or the environment	
Dependability	A property of a system that justifies placing one's reliance on it.	
Availability	Probability that a system or equipment will operate satisfactorily and effectively at any period of time	MTTF /(MTTF + MTTR)
Integrity	Ability to detect faults in its own operation and to inform a human operator	
Safety Integrity	Probability of a safety-related system satisfactorily performing the required safety functions under all the stated conditions within a period of time	

dependable. To summarize, dependability means that a system operates satisfactorily under any condition causing no dangerous situation. The above concepts were summarized in Table. 1[2]. These properties are tested and evaluated at the early design phase of a system through V-cycle[11].

At early design phase, possible faults and their effects are investigated. There are many ways to investigate the effects of fault on system's dependability. Some of these methods widely used are Failure Mode and Effect Analysis (FMEA) and Fault Tree Analysis (FTA). They are formalized method to guess all possible faults and their effects. M. B. Swarup and M. S. Rao[12] have conducted FMEA and FTA of radar sensor, brake sensor and speed sensor failure. In addition to their work, more detailed investigation on radar sensor failure considering its architecture was conducted in this paper and described in Figure 9 and Table 2.

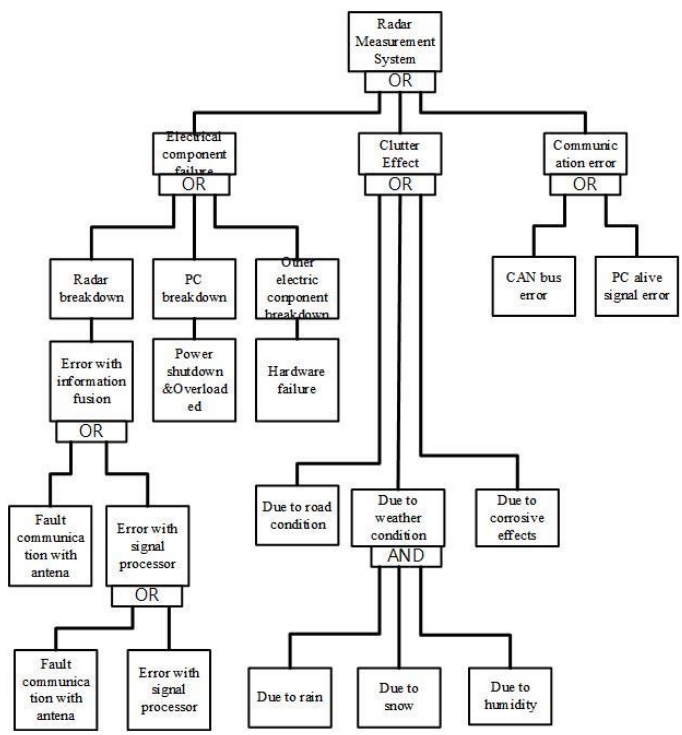


Figure 9 Fault Tree Analysis

Table 2 Failure Mode and Effect Analysis

Component	Failure mode	Failure cause	Failure effect on unit	Failure effect on system	Counter action
Radar measurement System	Electrical components failure	Radar Breakdown, PC(ECU) shout down, physically damaged electrical components	Sensor output equal to the maximum or minimum of that sensor	Causing serious malfunction of system	Override
	Induced noise	Random variations superimposed on the desired echo signal received in the radar receiver	Random variations superimposed on the desired echo signal received in the radar receiver	Aggravating system's performance	fail-tolerance actions
	Clutter Effect	Serious performance issues with radar system	Caused by a long radar waveguide between the radar transceiver and the antenna	Aggravating system's performance, may cause unexpected system behavior	Override or fail-tolerance actions
	Delay	Too much computation load or lack of computation capability of PC	Time gap between position of a real object and that of measured one	May cause unexpected system behavior	fail-tolerance actions
	Communication error	CAN bus error, PC alive signal error	Sensor output equal to the maximum or minimum of that sensor	Causing serious malfunction of system	Override

4.2. Fault Propagation Mechanism

When developing fault-detection and fault-tolerant control system, it is useful to know how hazard situation evolves from a fault. As described in Figure 10[13, 14], when a fault occur in a system, it propagate to the system and cause system error. At the end, system error causes system failure and hazard situation if it were not for appropriate action. Furthermore, G. J. Uriagereka et al[13] defined the time between when a fault occurs and when a hazardous situation happens as Fault Tolerant Time Interval (FTTI).

In addition, parts of J. Alrat et al[14]'s work explained that there were several type of faults and classified them into 7 categories. First, some fault become system error as depicted number 1. The associated time is the fault dormancy, which indicates the duration that a fault in system doesn't cause system error. Second, some faults described by number 2 could cause no considerable influence on the system. Third, as described by number 3, fault can be detected before system failure occurs. Associated time is referred as the latency of error detection. Fourth, number 4 depicted fault which cause error but tolerated without detection. Fifth, number 5 describes failure of detection. And, number 6 represent the situation that a fault causes a system error but the error is detected and tolerated by safety-related algorithms. Lastly, number 7 describes the situation where tolerance

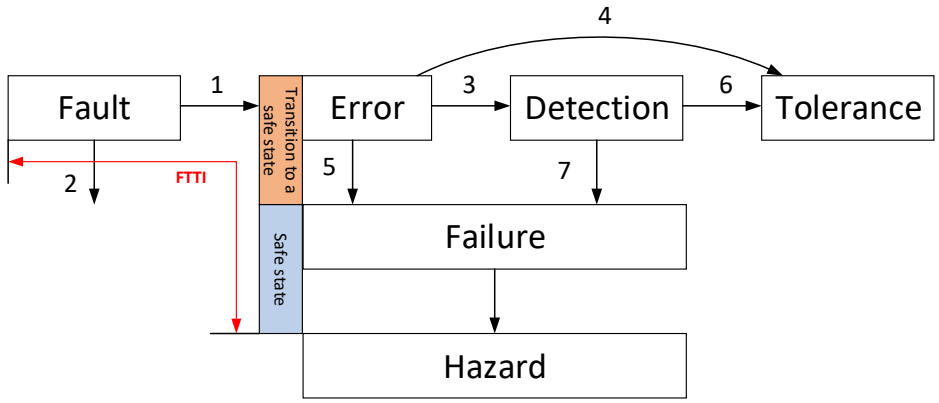


Figure. 10 fault propagation chain

mechanisms fail.

4.3. Fault Injection

Fault injection method is one of safety evaluation methods, which investigates whether the system returns to safe state or not after injecting virtual failure of some elements. Its main advantages were illustrated as follows; 1) it is easily shown how system behaves under a certain fault. 2) it is one of the most effective ways to evaluate fault-diagnosis and fault-tolerant mechanisms. 3) it makes fault forecasting or fault removal possible and it also allows us to identify the coverage of fault-tolerant systems. Fault injection method is classified into three types; hardware-based fault injection, software-based fault injection, and simulation-based fault injection. Hardware-based fault injection uses the prototype of the product. Thus, while the experiment results are reliable, it costs a lot. In case of software-based fault injection, there should be a software whose development is completed. However, simulation-based fault injection only uses system model and simulator it can be performed at lower prices and at early design phase. Fault injection method consists of 3 stages. First of all, fault models are models. Based on models faults, their location and simulation scenarios are determined. Then, those scenarios are injected to the system model. Finally, the results are collected at the end of simulation.

In this paper, a simulation-based fault injection has conducted in order to evaluate the validity of the suggested fault detection algorithm and suggested an important time interval between fault detection and happening of hazard event and defined it as Fault Tolerable Interval After Detection (FTIAD). The system model used in the experiment was an autonomous driving system equipped with path following model and adaptive cruise control system. The structure of autonomous driving system is illustrated in Figure 11. For the simulation experiment, PreScan[15] and Carsim[16] simulator were used via Matlab/Simulink.

The path following model is the model which extracts vehicle's

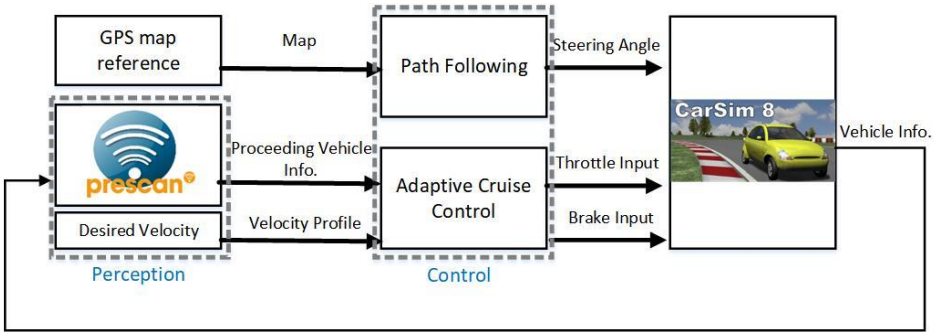
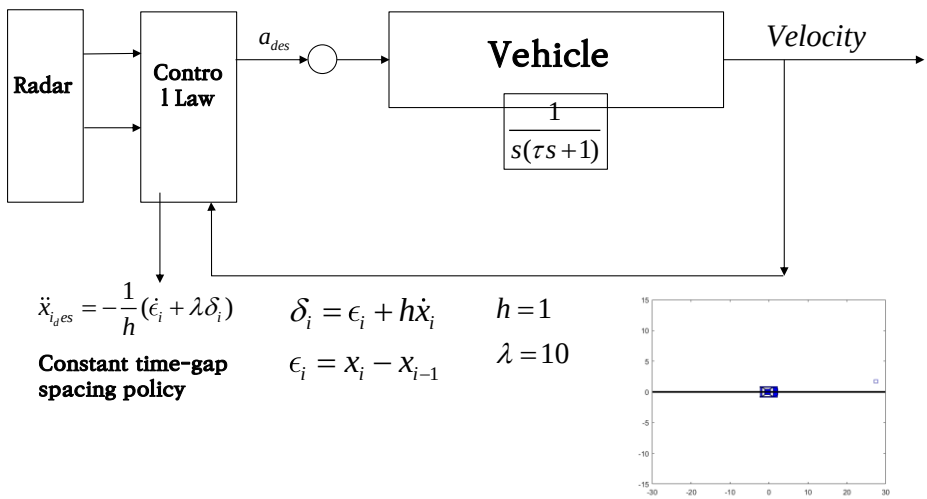


Figure 11 autonomous driving system

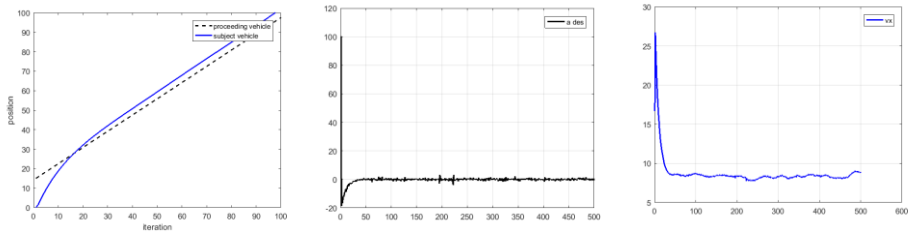
desired yaw rate based on road curvature and control vehicles using adaptive yaw rate gain steering controller[17]. Also, desired acceleration for adaptive cruise control was calculated abiding by constant time gap policy[18] as described in Figure 12. Figure 13(a) represents predetermined position, velocity and acceleration of preceding vehicle. Other figures in Figure 13 described position, velocity and calculated desired acceleration of vehicle with adaptive cruise controller (b) when radar sensor stuck (c) under radar sensor failure for detecting preceding vehicle (d) when radar sensor senses vehicle which doesn't exist (e) under sudden radar sensor deviation. Lastly, the target fault detection algorithm is the algorithm which detect abnormal behavior in radar sensor using



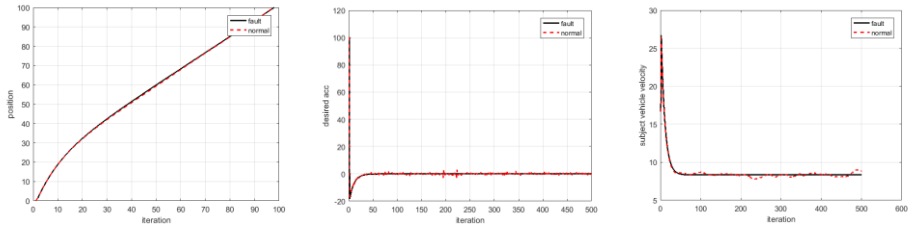
(a)

Figure 12 constant time gap policy

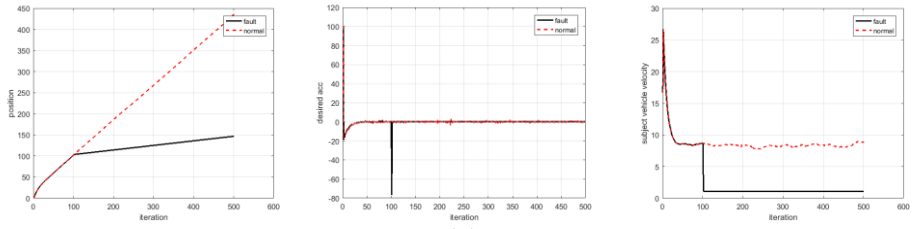
estimated measurement noise. The relative distance (clearance)



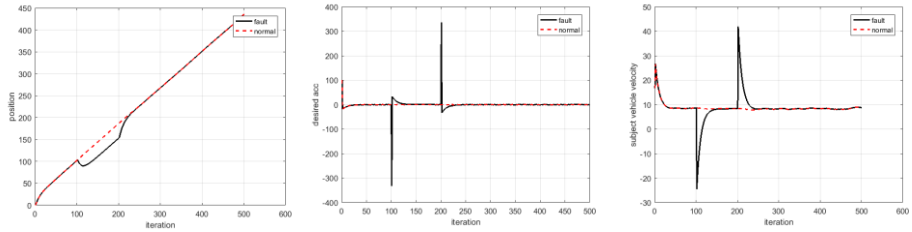
(a)



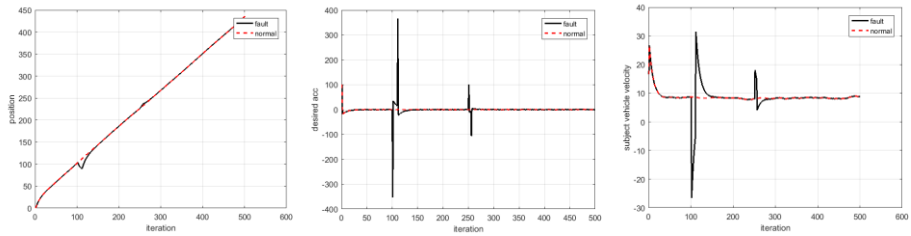
(b)



(c)



(d)



(e)

Figure 13 results of fault injection test

was measured by camera and radar sensors and these measurements were used for the algorithm as described in Figure 14.

Two cars drives on the elliptical two lane road in the simulation environment. The velocity profile of preceding vehicle is predetermined and host vehicle is autonomous vehicle as described in Figure 15. Several types of fault such as drift and noise in longitudinal and lateral direction were injected. The results (a) under longitudinal drift, (b) longitudinal noise and (c) lateral drift are plotted in Figure 16. The solid black line in graph represent desired clearance when system operates normally. The blue dot lines indicate upper and lower clearance boundary if the clearance exceeds this boundary, cars may cause hazardous accident like crashes. In addition, the solid blue horizontal lines indicates detected time. To clarify, the red dots represent the point where hazard accident happens. Through these results, FTIAD can be defined as described in Figure 17.

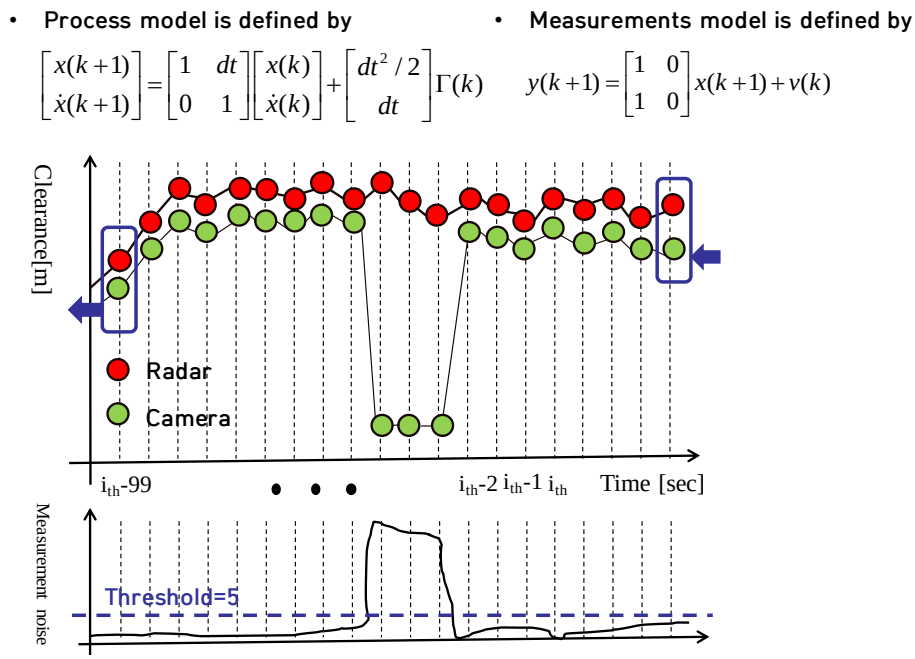


Figure 14 description of fault-detection algorithm

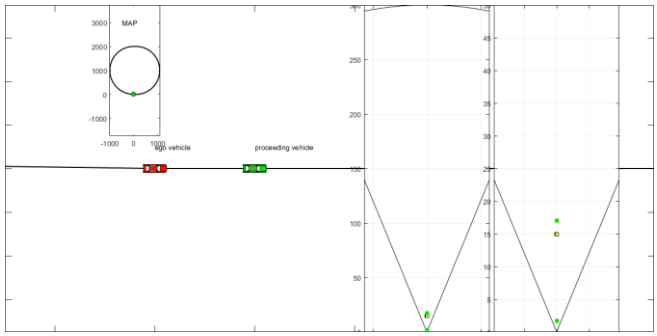
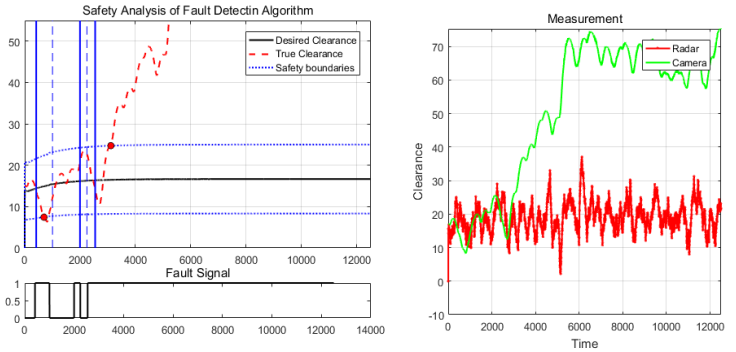
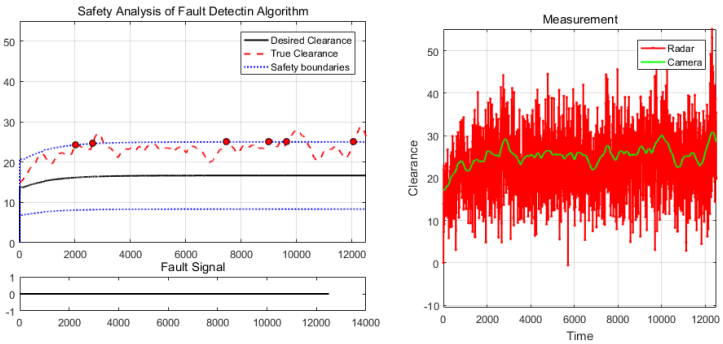


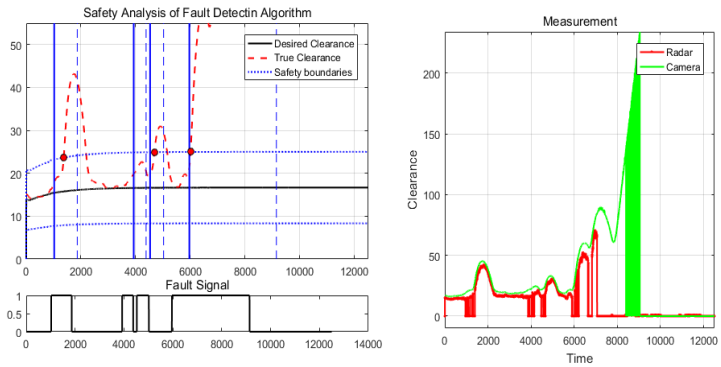
Figure 15 simulation environment



(a)



(b)



(c)

Figure 16 results of radar sensor fault injection

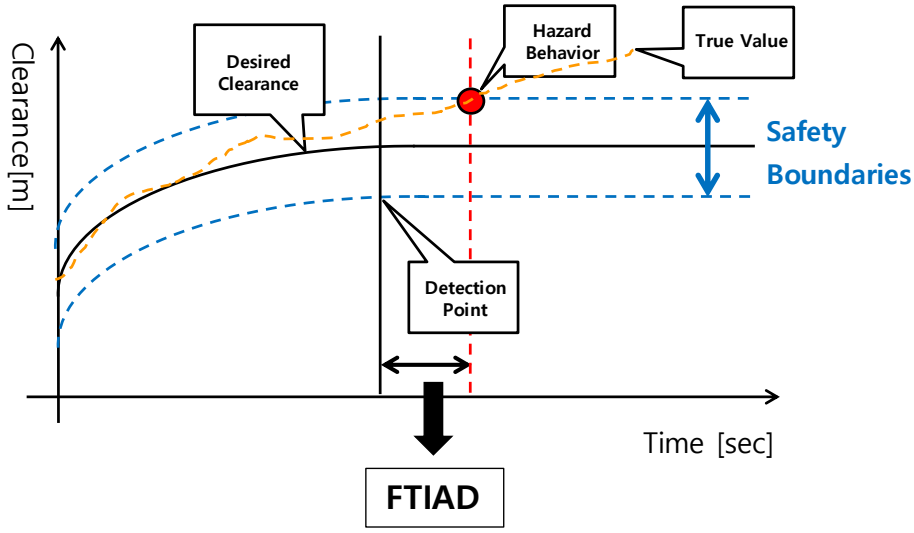


Figure 17 Fault Tolerant Interval After Detection

Chapter 5. Simulation and Result

5.1. Description of Driving Data

In this paper, a fault injection test was performed using actual driving data, which was collected when the test vehicle was driving on the Seoul National University (SNU)' s on-campus road. The entire driving path is shown in Figure 18. From some point in the road, there is a preceding vehicle. The road is a common two-lane urban street where parked cars, trees, poles and street ramps are on the roadside. The road and preceding vehicle is described in Figure 19. Another feature is that the road is inclined and has a high curvature. The car drove 1.1 kilometers along the road for about three minutes.

The test vehicle is described in Figure 20. It is a Hyundai Motors' K7 and equipped with various sensors. It is a typical sedan equipped with autonomous driving system. All the sensors mounted on the car was described in Chapter 2.

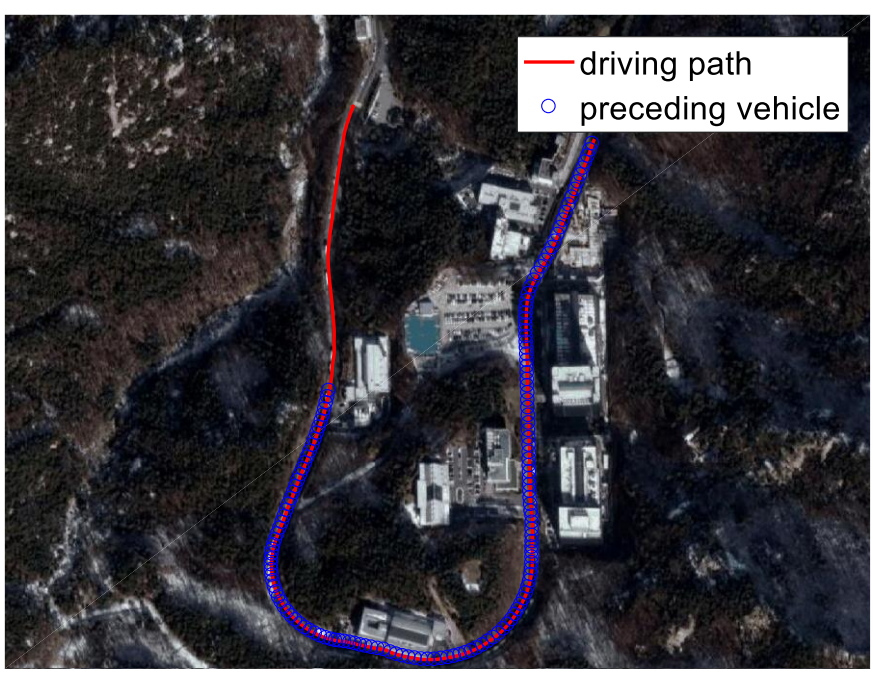


Figure 18 driving path

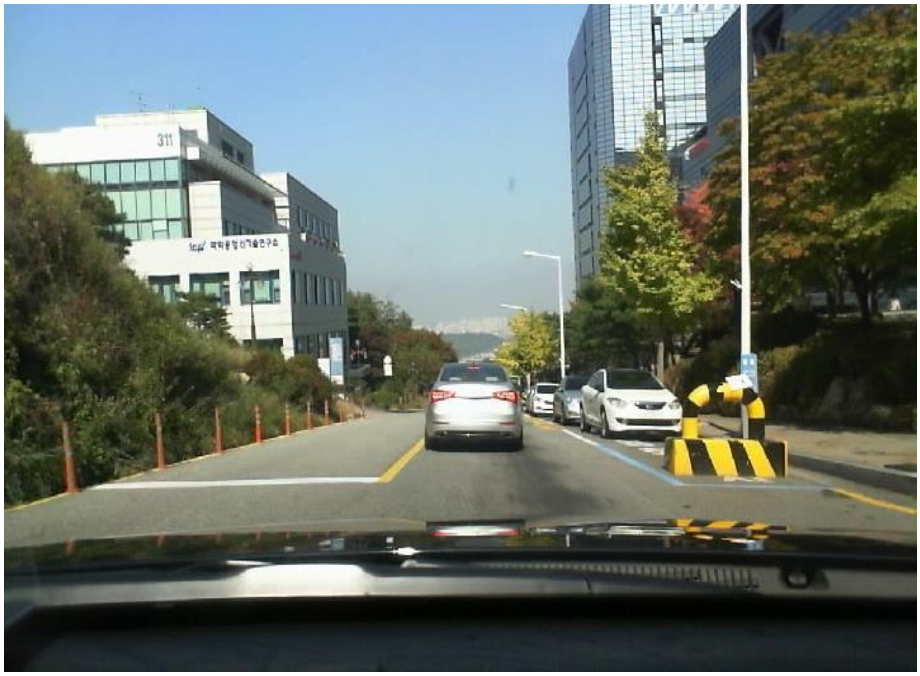
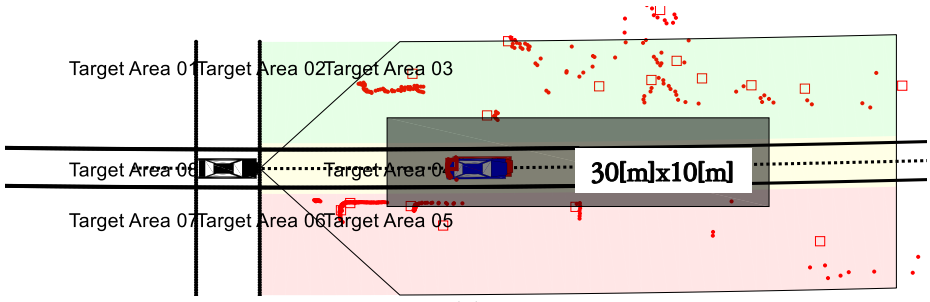
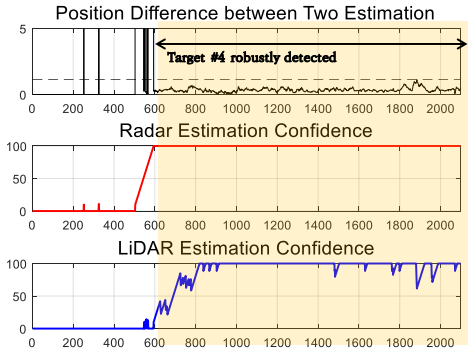


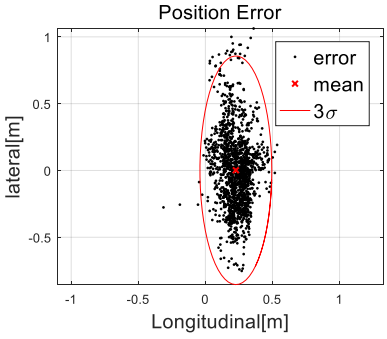
Figure 19 image of preceding vehicle

Figure 20 test vehicle



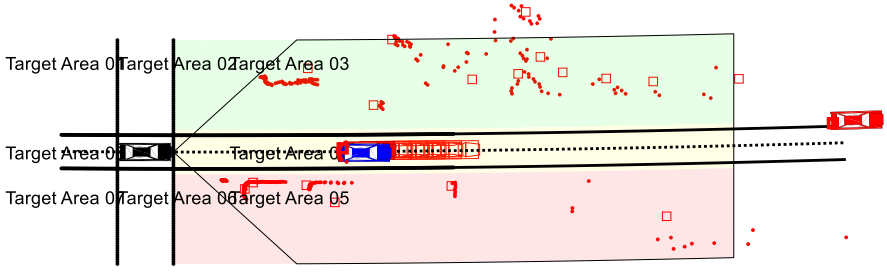
(a)



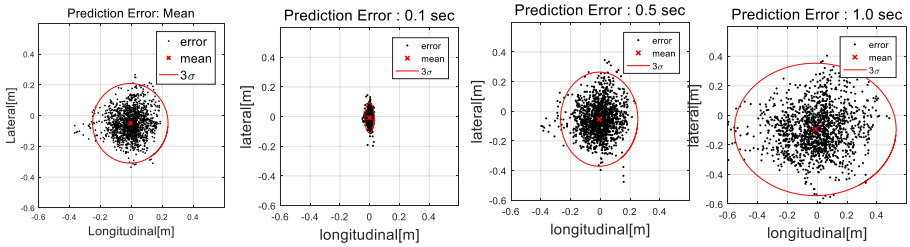


(b) (c)

Figure 21 target tracking accuracy



(a)



(b)

Figure 22 target prediction accuracy

The target tracking performance was analyzed using the driving data. During the driving time, the preceding vehicle is detected within a certain range as described in Figure 21(a). The blue vehicle is a preceding vehicle detected by LIDAR sensor and the red vehicle is a preceding vehicle detected by radar sensor. Red squares and red points are the measurement of radar sensor and LIDAR sensor, respectively. The longitudinal position error is less than 50 centimeters while the lateral position error is less than 1 meter. Considering that the lane is about 3 meters wide, the error is small enough.

The target prediction performance was also analyzed using the same driving data. The prediction using the estimation result before k^{th} step was compared with the estimation at k^{th} step. The predictions are accurate and their errors lie within the range of 50 centimeters as described in Figure 22.

5.2. Radar Sensor Fault Model and Fault Injection Test

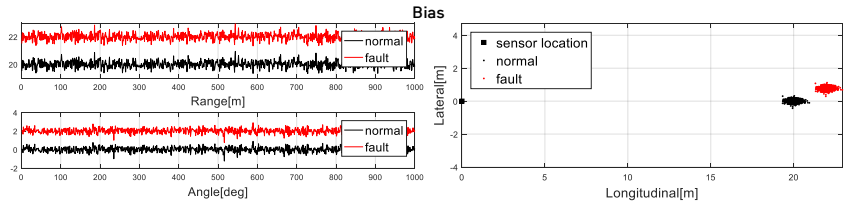
It is hard to design radar sensor fault model because not only it consists of complex component but also sophisticated data processing techniques are used for radar sensor as explained in Chapter 2. Nevertheless, radar sensor may break down in some circumstances. Possible cases of radar sensor failure from literatures or FMEA and FTA results in Chapter 4 are tabulated in Table 3. They can be classified into internal failures and external failures. However, it is still hard to find proper fault models because too complex to find the cause of the failure. Thus, the sensor failures have been modelled using general failure models. E. Balaban et al[19] organized a six common sensor failure. Six fault models are tabulated in Table 4. In Table 4, n denotes normal sensor noise, which correspond to $n \sim N(0, 0.25)$, X denotes a normal signal, and Y_f denotes a modeled fault signal. Parameters have been chosen at random. The designed fault are plotted in Figure 22. The radar sensor at (0, 0) detected the object at (20, 0).

Table 3 failure modes of automotive radar sensor

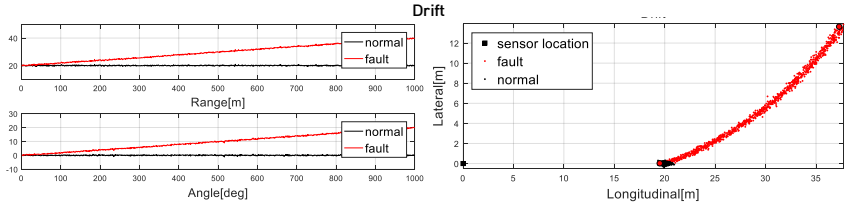
Internal	External
Internal communication error[20]	Road condition like low overpass causing clutter effect[12]
Radiation failure[20]	
Sensor status failure[20]	
Blockage[20]	
Overtemperature shut down[20]	Adverse weather condition causing clutter effect[12]
Partial blockage[20]	
Partial blockage at side lobe[20]	
Failure causing side lobe[20]	
Internal tracking failure[20]	Failure in CAN
Communication error[12]	
Deficient communication error[12]	External objects blocking the radio signal
Receiver failure[12]	
Noise failure caused by electronic component[12]	
Noise failure caused by increasing distance[12]	Excessive sensor tilting by external force
Clutter effect caused by corrosion[12]	

Table 4 six fault models

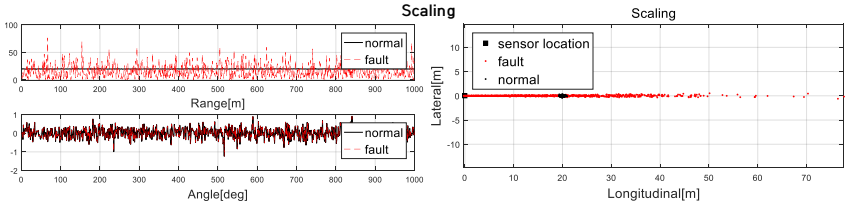
Failure	Model equation	Parameter description	Parameter
Bias	$Y_f = X + \beta + n$	β : constant offset	β_r = 2[m] β_θ =2[°]
Drift	$Y_f = X + \delta(t) + n$	$\delta(t)$:time-varying offset	$\delta(t) = a \cdot t$ $a_r, a_\theta \sim N(0, 0.25)$
Scaling	$Y_f = \alpha \cdot X + n$	α : random constant $0 < \alpha < \infty$	$\alpha \sim N(0,1)$
Noise	$Y_f = X + n_f + n$	n_f :noise	$n_{f,r} \sim N(0, 2[m])$ $n_{f,\theta} \sim N(0, 2[°])$
Loss of signal	$Y_f = C$	C =0	
Stuck sensor	$Y_f = C$	C : a constant	C_r =20[m] C_θ =0[°]



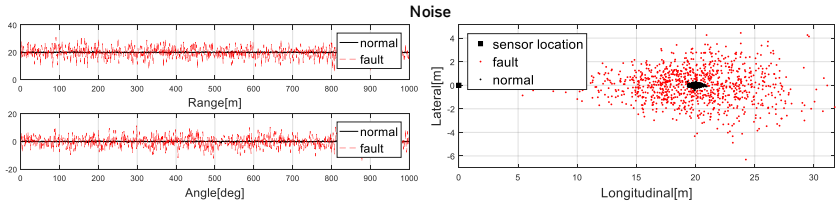
(a)



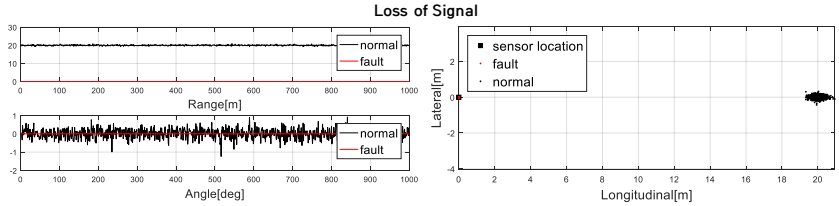
(b)



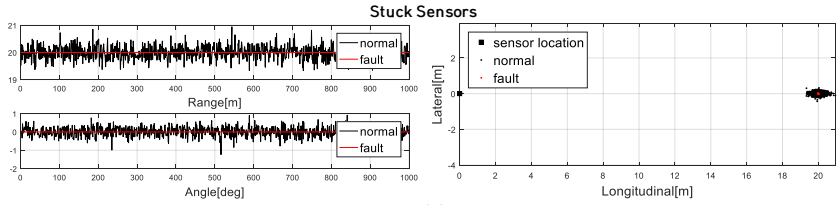
(c)



(d)

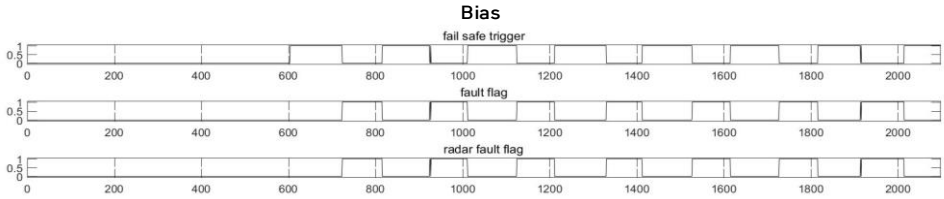


(e)

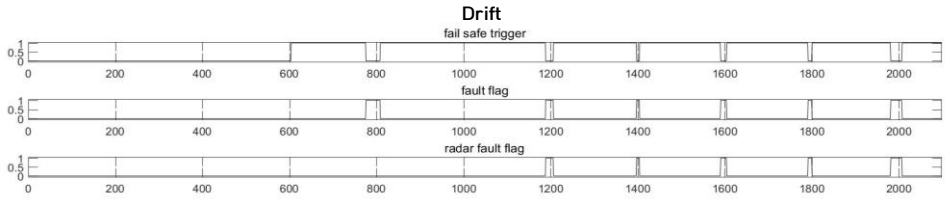


(f)

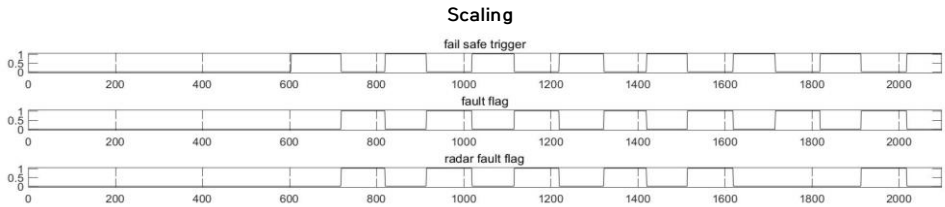
Figure 23 six fault models applied to radar sensor



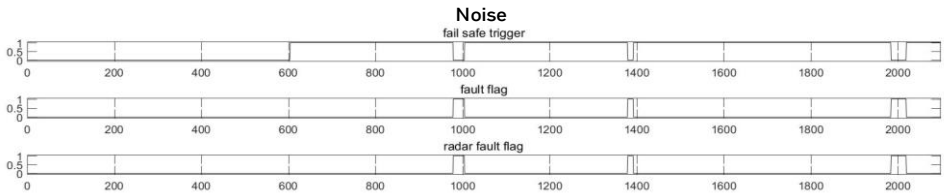
(a)



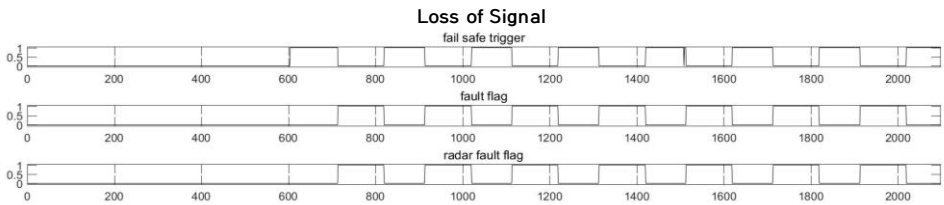
(b)



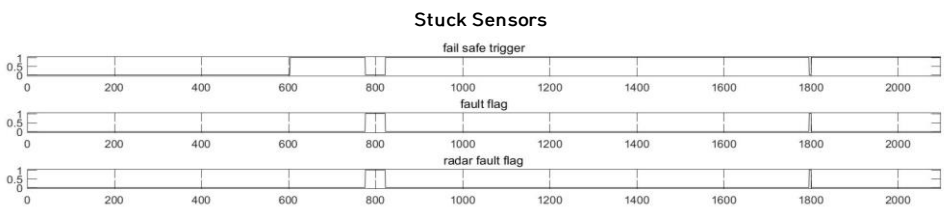
(c)



(d)



(e)



(f)

Figure 24 fault injection test result

Black dots in Figure 23 indicate normal radar measurement results while red dots in Figure 23 indicate radar measurement results when radar sensor fails. The horizontal axis in Figure 23 represents time and all fault models were measured for 1000 time step, which equals to 100 seconds.

All six fault models are injected into the driving data. To verify the fail-safe algorithm's performance, the faults were activated for 10 seconds at intervals of 10 seconds, starting from 70 seconds. The simulation results of fail-safety algorithm in Area 4 were plotted in Figure 24. The fail-safe trigger is set to be one when the fail-safe algorithm is activating and the fault flag is set to be one when the algorithm detects a system fault. The radar fault flag is set to be one when a fault is isolated as a radar sensor failure.

5.3. Fault Injection Test for In-Vehicle Sensors

The estimated velocity of the host vehicle with the driving data is plotted in Figure 25. The black solid line indicates vehicle velocity without faults in sensors and the red solid line indicates vehicle velocity when the rear right wheel speed sensor has failed.

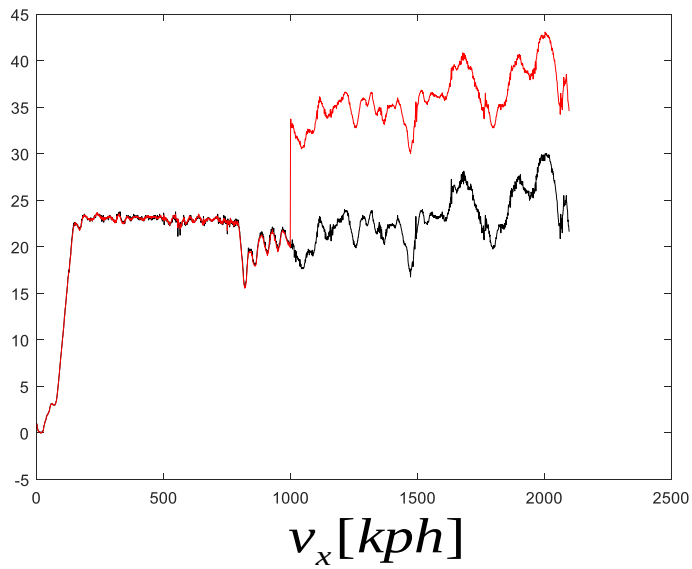


Figure 25 velocity of host vehicle without fault-detection algorithm

The 7.5 m/s additive fault was injected to the simulation at 100 seconds. When the kalman filter equips with the fail-safe algorithm, there is no difference between the estimated velocity without faults and that with faults as described in Figure26. Figure 27 represents kalman filter residuals, which is equal to reconstructed faults.

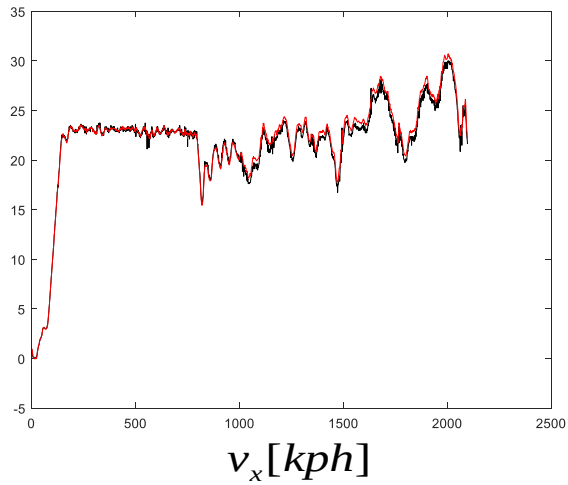


Figure 26 velocity of host vehicle with fault-detection algorithm

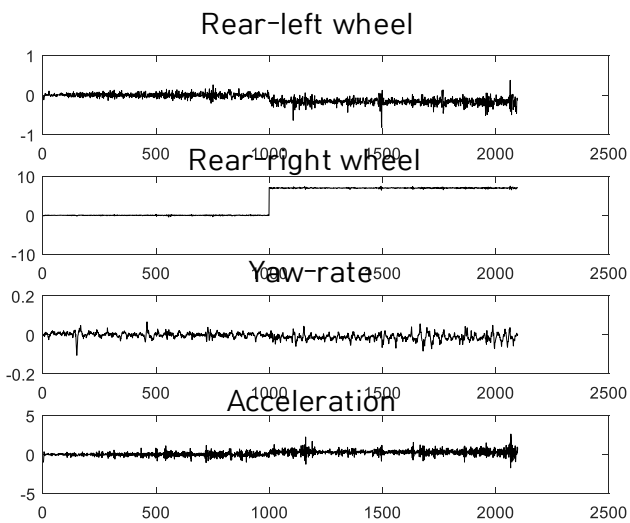


Figure 27 residuals

Chapter 6. Conclusion

Previously, Bader et al[5] designed fault tolerant architecture for data fusion. In his work, he proposed an architecture detecting failure in data fusion algorithm. His work focuses on finding the root of failure, weather it is from software, or either of the kalman filter branches, in order to mitigate fault effects when fusing data. He proved his work using GPS sensors and various in-vehicle sensors with process model representing kinematic motion of car-like robot. Also, R. Chandail[6] have suggested a fault-tolerant vision odometry system. His architecture consists of several safety-related nodes. For example, there is the Integrity Checking node, where the node inspects that the entered input signal corresponds with each fault model. After dividing faults, the algorithm imposed a large weight on the measurement covariance so as to mitigate the effect of abnormal measurement signal. However, their studies have focus on range sensor such as radar sensor or LIDAR sensor. In this study, the fail-safe algorithm for radar sensor of autonomous vehicle is designed and tested.

The proposed algorithm utilize a redundant relation in perception system of autonomous driving. Since both radar sensors and LIDAR sensors are used to detect surrounding objects, there are some objects detected by both sensors. The algorithm detects the faults by comparing distance between them. In order to compare the distance between them, surrounding vehicles should be classified. In this paper, the algorithm identifies the surrounding vehicles into eight different areas using relative distance from the host vehicle to measurement points. Finally, the algorithm selects the nearest vehicles in each area so as for both sensors to recognize the same target. After confirming they are tracking the same target, the algorithm checks if the distance between them deviates from the other. Also, detected faults are isolated by the algorithm. For the fault isolation, predictions of the previous surrounding vehicles' estimation were used. They show the location of nearby vehicles that would otherwise properly estimated when

the sensor had not failed. There should be a measurement on the edge of predicted vehicles if the sensor is operating normally, which is called Possible Area. Thus, the algorithm inspects the Possible Area for fault isolation. Also, the proposed algorithm was verified using real driving data. The real driving data which was recorded when the test vehicle was driving on the on-campus road of Seoul National University. To test the algorithm, six radar sensor faults were modeled based on the FMEA analysis and injected into the simulation. They are the most common sensor failures; bias, drift, scaling, loss of signal and stuck sensors. The simulation results showed that the proposed algorithm successfully detecting the radar sensor fault models. This study therefore suggests an effective way to detect faults in automotive radar sensors. Most notably, this is the first study to our knowledge to design fault-detection algorithm for range sensors of autonomous vehicle. In the other word, the novelty of this study lies on the fact that the proposed fault-detection algorithm is for range sensors. Not only is there no study on fault-detection systems for range sensors but also it is difficult to design a fault-detection system for range sensors of autonomous vehicle. Also, this study provides compelling evidence that shows the effectiveness of the proposed algorithm, which means this approach can be utilized for autonomous driving technologies to detect the abnormal behavior of range sensors.

However, there are some limitations. First, although the proposed algorithm is an effective way to detect the failure of an automotive radar sensor, it checks the fail-safety of radar sensor only when there are surrounding vehicles around the vehicle. Especially, the algorithm operates when the surrounding vehicles are stably detected for over 2 seconds, which means there should be surrounding vehicles driving almost the same speed with the host vehicle. Second, there is no evaluation index in this paper. In this paper, the performance of the algorithm was verified with real-driving data. And, the test results were only represented by fault flags in Figure 23. Thus, the future work should therefore include another fail-safe strategy for the case when there is no

surrounding vehicle and design the evaluation indexes for the fail-safe algorithm.

Also, dependability and safety evaluation have conducted in this paper. Previously, M. B. Swarup and M. S. Rao[12] analyzed adaptive cruise control algorithm using FMEA and FTA analysis. They investigated radar sensor system failure, brake sensor failure and speed sensor failure. Also, G. J. Uriagereka et al[13] defined controllability and safety evaluation method. They introduced fault injection method and one of the most important safety-related indexes, FTTI. However, M. B. Swarup and M. S. Rao only enumerated the cause of sensor failure and G. J. Uriagereka at al only conducted fault injection test for lateral controller for autonomous vehicle. Therefore, fault injection test for radar sensor was conducted in this paper. To investigate the effect of radar sensor failure on autonomous driving system, the radar sensor faults were modeled and fault injection test were conducted.

PreScan and Simulink/Matlab were used for this test. Also, adaptive cruise control abiding by constant time gap policy and path-tracking algorithm were equipped as the target autonomous driving syste. The result showed that the failure of radar sensor can cause dangerous situation and we can derive a safety-related index, FTIAD. It is worth noting that to our knowledge it is the first safety related indexes about fault detection algorithm.

However, there is a limitation that the test was only conducted with the simplified autonomous driving system, which means it is not realistic. Thus, we need to extend the test target to the more complex and realistic autonomous driving system in order to predict the accurate effect of radar sensor failure on autonomous driving system and to verify effectiveness of proposed safety-related index, FTIAD.

Bibliography

[1] M. Blanke, "Consistent design of dependable control systems," *Control Engineering Practice,* vol. 4, no. 9, pp. 1305-1312, 1996.

[2] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

[3] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, and J. Schröder, *Diagnosis and fault-tolerant control*. Springer, 2006.

[4] Y. Jeong *et al.*, "Vehicle sensor and actuator fault detection algorithm for automated vehicles," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*, 2015, pp. 927-932: IEEE.

[5] K. Bader, B. Lussier, and W. Schön, "A fault tolerant architecture for data fusion: A real application of Kalman filters for mobile robot localization," *Robotics and Autonomous Systems,* vol. 88, pp. 11-23, 2017.

[6] R. Chandail, "Vision Augmented State Estimation with Fault Tolerance," University of Waterloo, 2018.

[7] K. Oh, S. Park, J. Lee, and K. Yi, "Functional perspective-based probabilistic fault detection and diagnostic algorithm for autonomous vehicle using longitudinal kinematic model," *Microsystem Technologies,* pp. 1-11, 2018.

[8] B. Kim, K. Yi, H.-J. Yoo, H.-J. Chong, and B. Ko, "An IMM/EKF approach for enhanced multitarget state estimation for application to integrated risk management system," *IEEE Transactions on Vehicular Technology,* vol. 64, no. 3, pp. 876-889, 2015.

[9] B. Kim and K. Yi, "Probabilistic states prediction algorithm using multi-sensor fusion and application to Smart Cruise Control systems," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013, pp. 888-895: IEEE.

[10] M. Basseville and I. V. Nikiforov, *Detection of abrupt changes: theory and application*. Prentice Hall Englewood Cliffs, 1993.

[11] H.-L. Ross, *Functional Safety for Road Vehicles-New Challenges and Solutions for E-mobility and Automated Driving*. Springer, 2016, p. 269.

[12] M. B. Swarup and M. S. Rao, "Safety Analysis of Adaptive Cruise Control System Using FMEA and FTA," *International Journal of Advanced Research in Computer Science and Software Engineering Research Paper,* vol. 4, no. 6, 2014.

[13] G. J. Uriagereka, R. Lattarulo, J. P. Rastelli, E. A. Calonge, A. R. Lopez, and H. E. Ortiz, "Fault injection method for safety and controllability evaluation of automated driving," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*, 2017, pp. 1867-1872: IEEE.

[14] J. Arlat, A. Costes, Y. Crouzet, J.-C. Laprie, and D. Powell, "Fault injection and dependability evaluation of fault-tolerant systems," *IEEE Transactions on Computers,* vol. 42, no. 8, pp. 913-923, 1993.

[15] *PreScan-TASS International*. Available: https://tass.plm.automation.siemens.com/prescan
[16] *CarSim Overview-Mechanical Simulation*. Available: https://www.carsim.com/products/carsim/index.php
[17] C. Jung, H. Kim, Y. Son, K. Lee, and K. Yi, "Parameter adaptive steering control for autonomous driving," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, 2014, pp. 1462-1467: IEEE.
[18] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.
[19] E. Balaban, A. Saxena, P. Bansal, K. F. Goebel, and S. Curran, "Modeling, detection, and disambiguation of sensor faults for aerospace applications," *IEEE Sensors Journal,* vol. 9, no. 12, pp. 1907-1917, 2009.
[20] "Delphi ESR Startup Guide," 20th October,2015, vol. Version 2.1.

Abstract

본 논문에서는 자율 주행 자동차의 외부 센서 고장 진단 알고리즘을 개발하였다. 개발된 알고리즘은 세 부분으로 이루어진다. 1) 듀플리케이션-컴페이즌 방법을 통한 고장 진단 2) 가능 영역을 이용한 고장 분류 3) 차량 센서 고장 진단 알고리즘이다. 제안된 알고리즘의 주요 전략은 같은 물체를 감지하는 외부 센서의 리던던시를 이용하는 것이다. 시스템의 고장은 듀플리케이션-컴페리즌 방법을 통해 검출되고 고장 분류는 타겟의 예측을 통해 이루어진다. 또한, 주변 차량의 추정 결과와 센서 측정치를 비교하여 고장이 소프트웨어 고장인지 센서의 고장인지 분류할 수 있다. 마지막으로 고장은 앞으로 센서 측정치 존재할 것으로 예상된 지점은 가능 영역을 통해 이루어진다. 알고리즘의 성능은 주행데이터와 Matlab을 통해 수행되었다. 여러 번의 고장 주입 시험을 수행하였고 제안된 알고리즘의 성능은 시간 간격 개념을 제시하였다.