# EXAM NUMBER
# Y386445

## TABLE OF CONTENTS

# 1. HAZARD ANALYSIS

In this chapter we are going to define and justify a method which will be further used for hazard analysis, create the design of a Location Tracking System and design its components which have to be analyzed with a hazard analysis. Next, we will perform a hazard analysis for each of the defined earlier components of the Location Tracking System and, in conclusion, we will derive safety requirements for the Location Tracking System based on the results of the hazard analysis.

## 1.1 Description and justification of the method

Hazard and Operability study (HAZOP) [1] is used to identify any potential hazard and operational problems in existing or developing systems which could lead to considerable safety implications and major loses. Initially, this approach was used in chemistry industry to identify possible hazards in chemical plants [1]. According to McDermid, "a HAZOP study attempts to identify previously unconsidered failure modes by suggesting hypothetical faults for review" [2]. It means that applying HAZOP on a particular system could significantly assist in revealing vulnerabilities of that system and help to avoid potential hazards.

We are going to use HAZOP in the context of autonomous automobiles for a number of considerable advantages which this approach has:

- Firstly, HAZOP is the most widely-used technique for the hazard identification which is due to its well established rules, regulations and design notations. It means that while developing a new system like an autonomous car it is crucial to follow predefined instructions in order to consider all possible problems and eliminate them as early as possible.
- Secondly, HAZOP is based on using guidewords which "is used to prompt consideration of hypothetical failures, known as deviations, from the intended characteristics and behavior" [2]. In other words, a set of specific guidewords allow to consider a number of potential hazards and consequences which they might have. Thus, a considerable number of severe damages could be prevented in the phase of design process.
- Thirdly, HAZOP allows to interpret results of the hazard analysis in coherent and logical way and derive safety requirements for a system.

Traditional HAZOP technique uses a standard set of guidewords (such as no, more, less, as well as, part of, reverse, other than and many others) by coupling them with multiple parameters for process operation (temperature, flow, pressure etc.). However, this method is not suitable for software-based systems such as a location tracking system. Thus, we are going to use a set of guidewords proposed by DCSC [3] which can be split into three parts: service provision (omission, commission), service timing (early, late) and service value (undetectable and detectable) [2].

It should be noted what we understand by hazards in the context of autonomous vehicles. Below you can find the most catastrophic hazards which can occur: head-on collision, collision when overtaking, crashes happened on intersections, failure to comply with traffic regulations and traffic signs, getting off the road to a curb, stucking on the highway and many others. These hazardous events could lead to numerous people's deaths and damaged cars beyond repair.

In order to reveal all possible problems in the design stage, we need to undertake a deep analysis during HAZOP. To achieve that, we will record all identified issues into a table which will have six columns: guideword, deviation which explains what could go wrong, possible causes of the deviation, effects on the entire system which this deviation could have and comments.. Having undertaking this HAZOP approach, would significantly contribute to identifying possible hazards in the design stage of autonomous vehicle safety lifecycle.

## 1.2 Components

Now when we have an understanding of the process of hazard analysis and a well-defined work plan, we can think about the design of the system and its components which have to be analyzed with a HAZOP method.

The Location Tracking System (LTS) is the main component of autonomous automobiles which is responsible for location-awareness. It is a powerful computer which analyzes data collected from other different services, performs essential calculations such as distances towards surrounding objects, controls the execution of traffic laws according to the information from cameras, LIDAR, sensors and radars, plans its route and makes instant safety driving decisions, finds different way-around and many others functions. The LTS design is built on a principle of service-oriented architecture [4]: the LTS is split into microservices which are customized units of software (the LTS). You can find the design of the location tracking system and how it interacts with other services and environment in Fig. 1. Gray color defines the LTS itself. Blue color symbolizes various modules of the LTS which are integral parts of this system. Purple color symbolizes services which are internal parts of an self-driving car, but not the parts of the LTS. Yellow color signifies external services which are not parts of an autonomous vehicle.

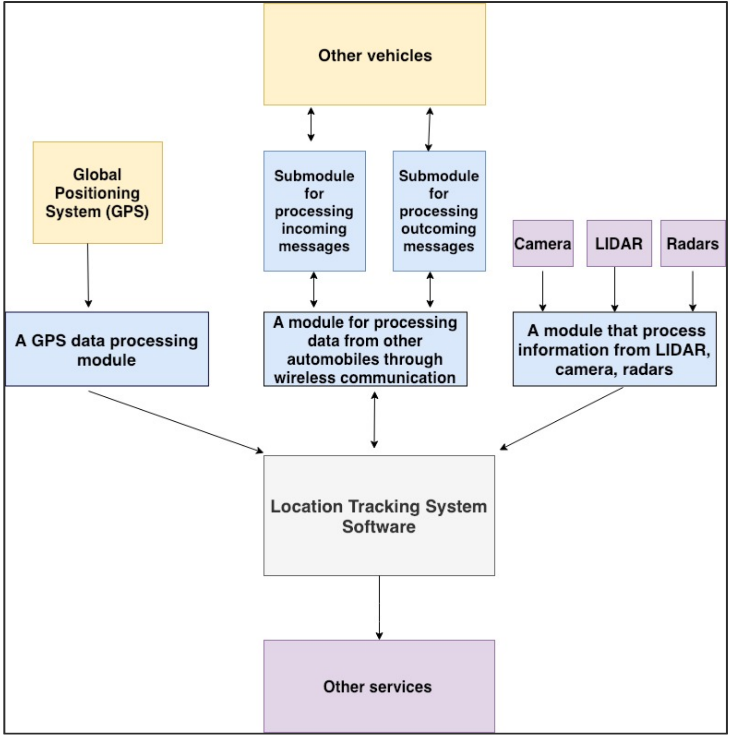
Other vehicles
Global Positioning System (GPS)
Submodule for processing incoming messages
Submodule for processing outcoming messages
Camera
LIDAR
Radars
A GPS data processing module
A module for processing data from other automobiles through wireless communication
A module that process information from LIDAR, camera, radars
Location Tracking System Software
Other services

*Fig. 1. Location Tracking System design*

As it mentioned before, the Location Tracking System consists of multiple modules which need the hazard analysis is to be performed on (blue color). They are described below:

- A GPS data processing module (service) collects data from GPS sensor and defines the exact geo-referenced position of autonomous vehicle on the map.
- A module for processing data from other autonomous automobiles through wireless communication which are within X km radius. It is split into two submodules:
  - A submodule for processing outgoing messages. It has the ability to send other cars information about road repair work, the possibility/impossibility of overtaking (the road is clear or not), blind zones, cornering at intersections, accidents, traffic jams and others unpredictable events.
  - A submodule for processing incoming messages. It analyzes all incoming messages and rebuilds the route depending on the information provided (searches for workarounds), adjusts the speed in case of a sudden stop in front of the vehicle etc.

  Additionally, this module is essential when a car rides through urban canyons, tunnels and other "deaf" zones. If a car can't read data from GPS due to "weak signal" which is blocked by multiple skyscrapers, it can request data (its location, maps, routes etc) from other vehicles which are located in X km radius. According to Xu, "dedicated short range communications will support safety critical communications, such as collision warnings, as well as other valuable Intelligent Transportation System applications, such as Electronic Toll Collection (ETC), real-time traffic advisories, digital map update, etc" [5] .
- A module that process information from LIDAR, camera and radarsThis huge module (in here, head module) is also split into four submodules and each of them is responsible for collecting data from a one of the components:
  - A submodule which collects data from LIDAR which is a rotating sensor in the roof of the car which creates 3D map of its surrounding;
  - A submodule which receives data from camera which detects traffic lights, pedestrians, other vehicles, bicycles and other obstacles;
  - A submodule which analyzes data from radars which are located in the front and back sides of the car, they help to accurately measure the distance to other vehicles or objects (e.g., while parking);

These submodules only collect data, interpret it into a computer-read way and send to the main head module. The head module processes incoming information from each submodule and calculates the distance to surrounding objects, controls the execution of traffic laws, position the car on the road, follow the compliance with the markings, ensures that the car does not go beyond the edges of the road, does not hit pedestrians, and does not hit other vehicles.

## 1.3 Applying the hazard analysis

### *1.3.1 Hazard analysis of a GPS data processing module*

Let’s analyze each module defined in the previous section. In this section by hazards we understand all hazardous event which have already been described earlier. We are going to start with analyzing a GPS data processing module (Table 1).

*Table 1 A hazard analysis of a GPS data processing module*

| **A GPS data processing module** | | | | |
|---|---|---|---|---|
| **Guide word** | **Deviation** | **Possible causes** | **Effect** | **Comments** |
| Early | Information from GPS delivered earlier | 1. GPS sensor failure (loose connection with satellites)<br>2. GPS sensor was damaged | No harmful effects | Maps and car’s location will be updated earlier which wouldn’t affect a car’s route |
| Late | Information from GPS delivered later | | Moderate | It depends on how much is delay? Seconds? Milliseconds?<br>If the delay less than X milliseconds, than the effect for people is moderate: they can be delivered in a particular place later due to a car turn on a reduced speed mode, for example.<br>If the delay more than X milliseconds, than consequences can be the same as in the omission failure. |
| Omission | No information from GPS | 1. GPS sensor failure (loose connection with satellites)<br>2. GPS sensor was damaged<br>3. GPS sensor wasn’t turned on<br>4. A car is located in a “deaf” zone (urban canyons, tunnels) | Critical | Without knowing its location, a car cannot ride.<br>Effect to others cars: car stuck on the road could lead to a traffic jam;<br>Effect to people: people cannot be delivered in a particular place on time |
| Commission | Information from GPS delivered when it is not required | 1. GPS sensor failure (loose connection with satellites)<br>2. GPS sensor was damaged | No harmful effects | The same as in the early communication failure |
| Value, undetectable | Information from GPS is unreadable | 1. GPS sensor failure (loose connection with satellites)<br>2. GPS sensor was damaged<br>3. Self-driving car memory corruption | Critical | Unreadable maps could prevent a car from riding. Effects would be the same as in the omission failure |
| Value, detectable | Information from GPS was damaged but still readable | | Negligible | It could be more difficult to follow a route. Wireless communication could assist car to continue riding: a car could use these damaged maps and ask other cars about missing pieces of maps. |

### *1.3.2 Hazard analysis of a module for processing data from other autonomous automobiles through wireless communication*

Submodules for sending and receiving messages could lead to the same hazards and effects to people and other vehicles because they both focus on processing information from other automobiles. For this reason, we will analyse only the submodule for processing outgoing messages. The hazard analysis for the submodule for processing incoming messages would be identical. As messages are sent upon requests (when some events happen), there cannot be early failure. What is more, when we talk about incoming messages, commission doesn’t make sense because it is impossible to receive a message “accidentally”. You can find the hazard analysis for the submodule for processing outgoing messages in the Table 2.

*Table 2 A hazard analysis of a submodule for processing outgoing messages*

| **A submodule for processing outgoing messages** | | | | |
|---|---|---|---|---|
| **Guide word** | **Deviation** | **Possible causes** | **Effect** | **Comments** |
| Late | A particular message has been sent later. | 1. Algorithmic failure<br>2. Problems in the sensor which is responsible for wireless communication<br>3. Weak signal<br>4. Data transmission failure<br>5. The storage of the messages has overflowed | Catastrophic | Delay in answering to other cars even less than 2000 milliseconds could lead to numerous hazard both for people within this car and other vehicles. For example, if a particular car asks whether it is possible to overtake, the answer from the asked car must be immediate because the answer is actual only at that time. |
| Omission | A particular message has not been sent. | | Catastrophic | The car that is not able to send messages about its intensions is a potential hazard on the road. Other vehicles wouldn't be notified about intensions of the car which could lead to severe accidents and, thus, peoples' deaths and cars crushes. |
| Commission | A particular message has been sent when it was not required. | 1. Algorithmic failure | Probably catastrophic | It depends on the type of the message. Wrong messages could provide misleading information for other vehicles which could lead to accidents. |
| Value, undetectable | A particular message which has been sent contains wrong value. | 1. Algorithmic failure<br>2. Problems in the sensor which is responsible for wireless communication<br>3. A particular message was corrupted while sending.<br>4. A particular message was intercepted and changed by hackers.<br>5. Data transmission failure<br>6. The storage of the messages has overflowed | Catastrophic | The same as in Commission<br>What is more, if a message was hacked and changed, it could lead to major hazards. |
| Value, detectable | A particular message which has been sent contains wrong value but still acceptable. | | Negligible | If a particular message was corrupted but still readable and makes sense, other vehicles will be able to process it and make right decisions. |

### *1.3.3 Hazard analysis of a module for processing data from different sensors*

We are going to start with analysing a submodule for processing data from LIDAR (Table 3), then a submodule for processing data from a camera (Table 4), then a submodule for processing data from radars (Table 5). As data from these sensors are received upon requests (for example, every X milliseconds), there cannot be early failure and commission failure.

*Table 3 A hazard analysis of a submodule for processing data from LIDAR*

| **A submodule for processing data from LIDAR** | | | | |
|---|---|---|---|---|
| **Guide word** | **Deviation** | **Possible causes** | **Effect** | **Comments** |
| Late | Information about car's surroundings have been passed later. | 1. LIDAR sensor was damaged<br>2. LIDAR wasn't turned on | Probably negligible | As a self-driving car is equipped with multiple sensors like a camera, radars, a delay in receiving the map of car's surrounding wouldn't have significant effects. In other words, essential calculation (e.g. distance to other objects) can be performed operating with data received from other sensors. However, it could consume more time, contain errors and be no so accurate. The severity of consequent hazards depends on the delay in receiving data from LIDAR. |
| Omission | Information about car's surroundings have not been passed. | 1. LIDAR sensor was damaged<br>2. LIDAR wasn't turned on | Catastrophic | This could lead to significant problems: without a map of car's surroundings it would be difficult to perform essential calculations and keep a car strictly on the road. Data received from other sensors is not enough |

| | | | | |
|---|---|---|---|---|
| | | | | for performing calculations such as object's size, distance, absorption characteristics etc. Various hazards can happen from collisions with other vehicles, crashing into objects (pedestrians, building etc) to getting off the road resulting in accident. |
| Value, undetectable | Information about car's surroundings is corrupted and, thus, is not readable. | 1. LIDAR sensor was damaged<br>2. LIDAR wasn't turned on<br>3. Some obstacles prevent to display surroundings in a proper way (tree's leaves, bugs, poor weather conditions and others) | Catastrophic | The same as in the omission failure |
| Value, detectable | Information about car's surroundings is corrupted but still readable. | | Negligible | If the information about car's surroundings is still readable and a car is able to detect object, other sensors will help a car drive safely |

*Table 4 A hazard analysis of a submodule for processing data from a camera*

| **A submodule for processing data from a camera** | | | | |
|---|---|---|---|---|
| **Guide word** | **Deviation** | **Possible causes** | **Effect** | **Comments** |
| Late | Information about traffic lights, pedestrians and other obstacles has been received late. | 1. Camera was damaged<br>2. Camera wasn't turned on | Catastrophic | In this case even a millisecond plays a significant role. For example, if a car receives information about traffic lights 1 millisecond later, it can result in numerous accidents and even people's deaths. What is more, delay could lead to failures to comply with traffic rules which could also have catastrophic consequences. |
| Omission | Information about traffic lights, pedestrians and other obstacles has not been received. | | Catastrophic | The same as in the late failure |
| Value, undetectable | Information about traffic lights, pedestrians and other obstacles is unreadable | 1. Camera was damaged<br>2. Camera wasn't turned on<br>3. Some obstacles prevent to display surroundings in a proper way (tree's leaves, bugs, poor weather conditions and others) | Catastrophic | The same as in the late failure |
| Value, detectable | Information about traffic lights, pedestrians and other obstacles is damaged but still unreadable | | Negligible | If the data is still readable and LTS is able to analyse it and distinguish traffic lights properly, hazards can be avoidable. Moreover, LIDAR and other sensors will help a car detect objects. |

*Table 5 A hazard analysis of a submodule for processing data from radars*

| A submodule for processing data from radars | | | | |
|---|---|---|---|---|
| **Guide word** | **Deviation** | **Possible causes** | **Effect** | **Comments** |
| Late | Information about objects in front and back of the car has been received late. | 1. One of the radars was damaged<br>2. One of the radars was not turned on | Probably negligible | Since a car has four radars (two in the front side and two in the back side), the problem with one radars wouldn't affect others. What it means that a car is still able to detect objects and measure the distances towards them. In the case if all radars stop work simultaneously, the car is still able to perform calculations and measure the distances towards objects with other sensors like LIDAR. However, in some circumstances like in traffic jams or while parking (where the distance towards other objects is very small and each inch matters) some crashes could happen. |
| Omission | Information has not been received. | | Probably negligible | The same as is in the late failure |
| Value undetectable | Information about objects in front and back of the car is unreadable. | 1. One of the radars was damaged<br>2. One of the radars was not turned on<br>3. One of the radars is dirty because of the dust, poor weather conditions etc. | Probably negligible | The same as is in the late failure |
| Value detectable | Information about objects in front and back of the car is damaged but still readable. | | Probably negligible | The same as is in the late failure |

### *1.3.4 Hazard analysis of a module for processing data from LTS (controller)*

*Table 6 A hazard analysis of the Location Tracking System*

| Location Tracking System (Controller) | | | | |
|---|---|---|---|---|
| **Guideword** | **Deviation** | **Possible causes** | **Effect** | **Comments** |
| Late | Analysis of the data collected from all sensors started later. | A particular sensor has sent the data with a delay | Critical | It could result in slow motion of the car, stuck on the road which could lead to a traffic jam. |
| Omission | Analysis of the data hasn't started at all. | 1. Algorithmic failure<br>2. A car was hacked which has resulted in service denial<br>3. The storage of the information has overflowed | Catastrophic | Complete failure to update the state of the car could lead to catastrophic consequences. |

## 1.4 Derived Safety Requirements (DSRs)

For all DSRs, a suitable message describing a particular issue should be immediately sent to the server in order to notify maintenance services that something happened and needs to be fixed immediately. What is more, for each component the Worst Case Execution ("Response" - in the context of autonomous vehicles) Time should be calculated in milliseconds. The Derived Safety Requirements are shown in Table 7.

*Table 7 Derived Safety Requirements*

| **№** | **Source of problem** | **DSR** |
|---|---|---|
| 1. | A GPS data processing module *– late* | Maps for a certain distance (e.g. radius X km) should be downloaded in advance, so that a car can use it until new maps will be downloaded. What is more, a reduced speed mode should be turned on before new maps is downloaded. |
| 2. | A GPS data processing module ***– omission, undetectable value*** | Firstly, a car should request map or route from cars which are located in the radius X km (through wireless communication). If it doesn't help, a car should turn on an emergency mode, keep the side of the road and stop. Engineers should arrive as soon as possible to fix the car and provide people with a new car. |

| | | |
|---|---|---|
| 3. | A GPS data processing module – ***detectable value*** | Wireless communication could assist car to continue riding: a car could use the damaged maps and ask other cars to provide missing pieces of maps before new maps will be downloaded. A car should turn on a reduced speed mode. |
| 4. | A submodule for processing outgoing messages – ***late*** | Delay in answering to other vehicles should not exceed X milliseconds. Additionally, other vehicles should be notified about car's intensions in X milliseconds. |
| 5. | A submodule for processing outgoing messages –***omission*** | Each sent message should have a status (sent, delivered). If the message has been sent but has not been delivered, a car should try to send the message again. If it doesn't help, a car should keep the side of the road and stop. |
| 6. | A submodule for processing outgoing messages – ***commission, undetectable value*** | Each message should have its type (important, minor). Before sending a particular message, the module should check whether it is important or not and if it is so, the information in this message should be verified again. Thus, each message would be genuine. What is more, to prevent cyber security problems, each message should be encrypted properly. |
| 8. | A submodule for processing data from LIDAR – ***omission, undetectable value*** | A car should turn on an emergency mode, keep the side of the road and stop. To be able to go to the side of the road, information for essential calculations should be received from other sensors, i.e. the system should be fault tolerant |
| 9. | A submodule for processing data from a camera – ***late, omission, value undetectable*** | As only camera can distinguish traffic lights and other sensors cannot do that, a car should turn on an emergency mode, keep the side of the road and stop, waiting for a repairman service. |
| 10. | Location Tracking System (Controller) – ***late, omission*** | Data from sensors should be collected every X milliseconds. A failure to update the state of the car and perform essential calculations should be detected within X seconds. Then emergency mode should be turned on, the car should keep the side of the road and stop, waiting for a repairman service. |

# 2. FAILURE ANALYSIS

In this chapter we are going to come up with the method that will be used to perform the failure analysis and then we will apply this method to the most significant hazard identified before – when a car hits an object.

## 2.1 Description of the method

Since Failure Modes and Effects Analysis (FMEA) is quite similar to bottom-up technique HAZOP analysis with only a "difference is that an FMEA considers failure modes of components of a system, while a HAZOP analysis considers abnormalities in a process" [6], we are going to use another technique which is called Fault Tree Analysis (FTA). FTA is a top-down process which is used to decompose an undesired event into possible causes in increasing detail to determine the causes or combinations of causes of the top event. Now it is the most common diagrammatic safety analysis technique due its well-defined semantics and clear structure of diagrams. According to Paige, "the fault tree qualitatively illustrates how the undesired event results from the primary events, via zero or more intermediate events." [7].

FTA is a graph which is consist of two types of nodes: events which are "the failure of a subsystem down to an individual component" and gates which represent "how failures in subsystems can combine to cause a system failure" [6]. We are going to decompose a failure to the level of the components.

## 2.2 Application of identified method

From HAZOP we can conclude that there is a great number of potential hazards which "might endanger human life, lead to substantial economic loss, or cause extensive environmental damage" [8]. One of the most catastrophic hazards is a collision. A car can hit numerous objects from other vehicles to pedestrians which could lead to multiple deaths. For this reason, we will apply the failure analysis method (FTA) to this hazard. After having undertaken FTA, we will be able to conclude what could cause a car to hit objects. You can find the demonstration of Fault Tree Analysis in Fig. 2.

On the top of the three is an initial undesired event. Then it is followed by one of the three events: basic event (circle) which symbolizes an initiating fault requiring no further development, undeveloped event (diamond) which is not developed because it is considered unnecessary, intermediate event which arises from the combination of other. From the variety of gates we use only one which is "OR" (the occurrence of one or more input events will cause the output to occur).

Let's look at the FTA in more details. Car can hit an object due to the problems with brakes: they didn't work (mechanical or electronical issues), they haven't been pressed (software issues) or they have been pressed too late (software issue) which lead to the hazard occurred. We are not going to analyse why the brakes didn't work because it is out of control of the Location Tracking System. Instead, we are going to analyse two other causes of the hazard.

LTS could fail due to a number of reasons. One of them is denial of service which is caused by safety security problem. Enemies could find the vulnerabilities in the LTS and hack it which has resulted in denial of service and, as a result, to the loss of the control and hazard occurred. As our goal is to look at how the components of the LTS could lead to the hazard, we are not going to focus on security problem.

Another significant problem which could result in the car's collision is an algorithmic failure in performing calculations. Poor choice of programming language, types of variables, weak calculations algorithms – all these issues could be the potential reason of the hazard.

Finally, LTS controller could fail because one of the services hasn't provided essential data (map of surroundings, speed, distances to other vehicles, detection of the traffic lights and others). This could be caused by one of the LTS submodules failed to collect data from sensors or failed to properly analyze revived data. Such components as LIDAR, camera, radars could be damaged, turned off etc. which is resulted in the hazard occurred. Additionally, the submodule for processing data from and to other vehicles could fail because other automobiles haven't notified the car about their intentions or they have notified but the car failed to process these messages.

As we can see, at the bottom of the fault tree are basic initiating fault events. This technique allows us to look at the system from other hand which would lead to valuable design improvements. As Paige emphasizes, "the major benefit of this approach is that it explicitly identifies the relationship of events to each other, including the root fault events" [7].

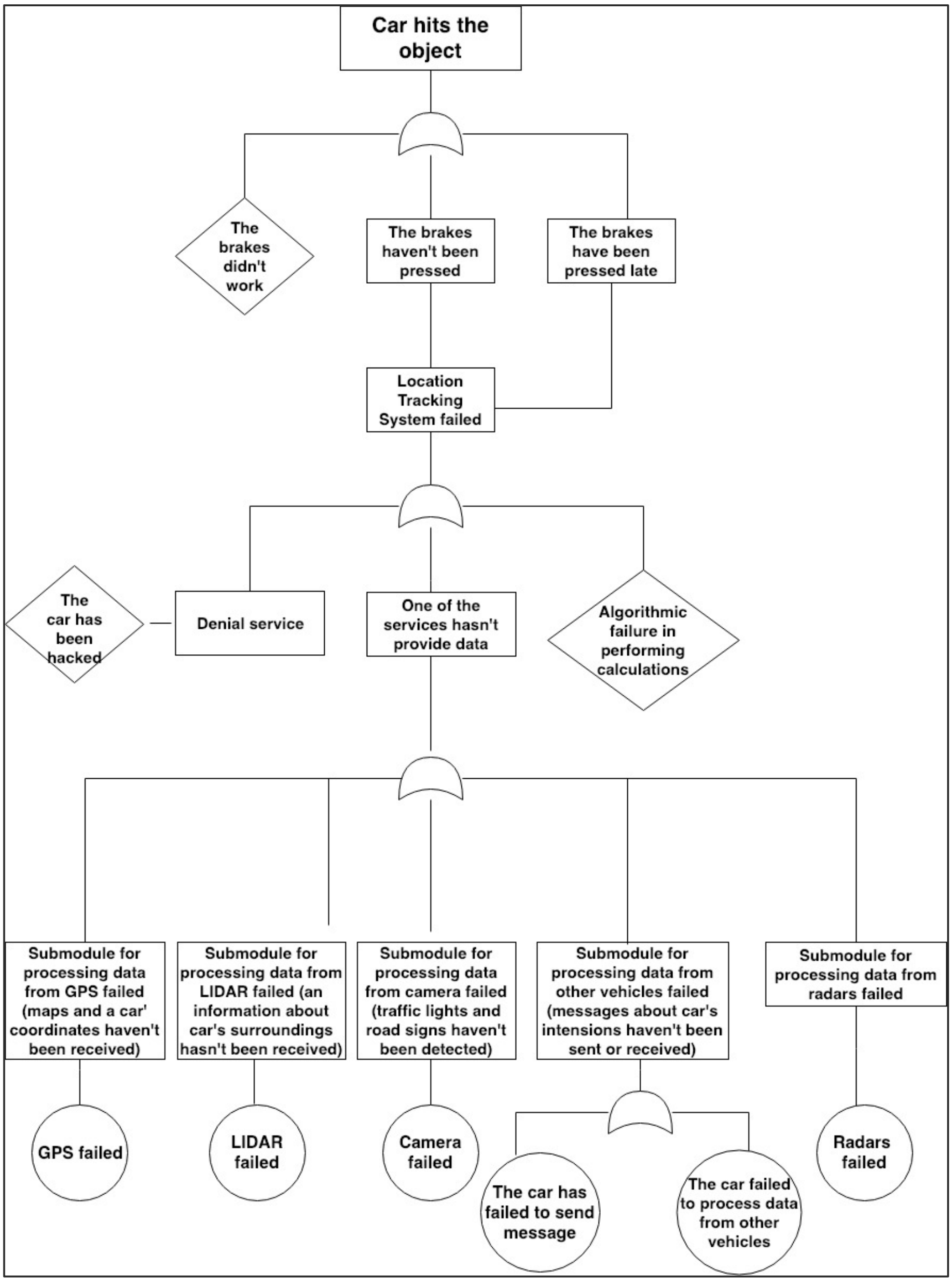
Car hits the object
The brakes didn't work
The brakes haven't been pressed
The brakes have been pressed late
Location Tracking System failed
The car has been hacked
Denial service
One of the services hasn't provide data
Algorithmic failure in performing calculations
Submodule for processing data from GPS failed (maps and a car' coordinates haven't been received)
Submodule for processing data from LIDAR failed (an information about car's surroundings hasn't been received)
Submodule for processing data from camera failed (traffic lights and road signs haven't been detected)
Submodule for processing data from other vehicles failed (messages about car's intensions haven't been sent or received)
Submodule for processing data from radars failed
GPS failed
LIDAR failed
Camera failed
The car has failed to send message
The car failed to process data from other vehicles
Radars failed

*Fig. 2. Fault Tree Analysis*

# 3. MITIGATION STRATEGIES

In this chapter we will identify the sensitivity points in the design, describe and justify appropriate architectural approaches that could be used to enhance the dependability of the system and, lastly, we will update the failure analysis based on the architecture approach.

## 3.1. Sensitivity points

As it was concluded earlier, one of the most important tasks of a self-driving vehicle is to precisely estimate its position relative to its surroundings based on the data collected from various sensors measurement. It means that "localization must be reliable and sufficiently accurate in the presence of potential sensor" [9]. To achieve "centimeter-level accuracy that is robust against sensor failures and missing information" [9], the Location Tracking System should follow fault tolerance logic which means that "the failed sensor of one type, can be replaced by the functionality of a sensor of another type" [10]. However, before introducing fault tolerance strategies and enhancing the dependability of the LTS, the system should be analysed for the occurrence of sensitivity points.

With the help of the Fault Tree Analysis applied before we can identify the most vulnerable places in the LTS design. Sensitivity points are listed below:

1. **GPS failed.** A car should functionate in the range of environments from open roads to urban canyons. But what the system should do in the case if the signal is lost and, thus, GPS fails? Without maps and precise car's location a vehicle wouldn't be able to continue its route and deliver people to their desirable destination. Thus, the system should be able to detect the loss of the GPS signal in X milliseconds and navigate even without GPS data.
2. **LIDAR failed.** A number of reasons could prevent LIDAR to operate it properly. An internal map of surroundings is a key factor in estimating car's location. Thus, loss of LIDAR can't prevent a vehicle in estimating its position on the road.
3. **Camera failed.** Without pictures of the car's surroundings a car cannot obey driving rules and detect traffic lights and signs which could lead to exceeding the established speed limit and, thus, numerous collisions with other objects. For this reason, a failure in the camera shouldn't affect the ability of the car to distinguish entities.
4. **Submodule for processing information from other vehicles failed.** Without knowing the intentions of other vehicles, a car can't make decisions about turns and other important factors. Thus, even if the car failed to receive a message from other vehicles, it should still be able to ride.
5. **LTS controller failed.** An error in calculations, memory storage corruption, hack of the software could result in numerous hazard and make irrecoverable and irremediable damage to people. To prevent these numerous potential dangers, LTS failure should be detected immediately.
6. Another significant sensitivity point is the **denial of service in the case of accident.** If the car was considerably damaged, its sensors responsible for its location on the map still would work. Having the car's exact position, emergency services can easily detect its location and arrive immediately to save people's lives.

## 3.2. Architectural approach

Now when we identified the sensitivity point in the LTS design, we can define appropriate architectural approaches that could be used to enhance the dependability of the system. For each of the sensitivity point described above we will suggest the solution which assists in preventing hazards.

1. **GPS failed.** In the case of GPS failure, the system should still ride. To achieve this, in the car can be implemented offline map-building technique which uses cameras or laser rangefinder components followed by online localization. What is more, "utilizing additional sensors on board the vehicle, such as radars, to improve the fault tolerant nature of the system will help provide an accurate estimate in many different situations" [9]. In other words, in the architecture design of the car we should add laser rangefinder components which would be launched in the case of GPS failure. In combination with a camera and radars, this laser rangefinder components would build the map, estimate the car position and, thus, a car would be able to continue its route.
2. **LIDAR failed.** Similarly, in the case of LIDAR failure, cameras, radars and laser rangefinder components can assist in building an internal map of the car's surroundings.
3. **Camera failed.** However, as the camera alone is responsible for detecting traffic lights and road signs, we need another component which could be launched in the case if the camera fails. In order to achieve fail-safe riding, we should introduce another additional camera which would work and assist in obeying road rules if the main camera failed.

4. **Submodule for processing information from other vehicles failed.** All messages intended for other cars should be also send to the server which places them in a shared database. Thus, the car would read a message from two different resources (through wireless communication and from the server) which decreases the likelihood of the hazard occurred.
5. **LTS controller failed.** The failure in the controller and other components should be detected within X milliseconds. To achieve that, watchdog needed to check expected tasks performed during a period of time, i.e. the data from submodules is received every X milliseconds. If appropriate task execution pattern not achieved, turn on the emergency mode, keep the side of the road and wait for maintenance service. The failure in the particular submodule should be detected within X milliseconds and appropriate replacement sensors and equipment should be launched immediately. At the same time, the failure should be passed to the server.
6. To be able to identify **car's position in the case of accident**, we need to introduce an emergency unit which works even without power and, thus, allows to easily find the car's coordinates on the map.

You can find the autonomous car's refined architectural design in the Fig. 3. Emergency (replacement) components (on the picture they have a red border) will be launched in the case of problems with main components. Watchdog controls the performance of tasks and responses time. A server with a shared database stores all messages from vehicles and serves as a back up storage. All these introduced components enhance the dependability, reliability of the system and minimize the risk of hazards occurrence.

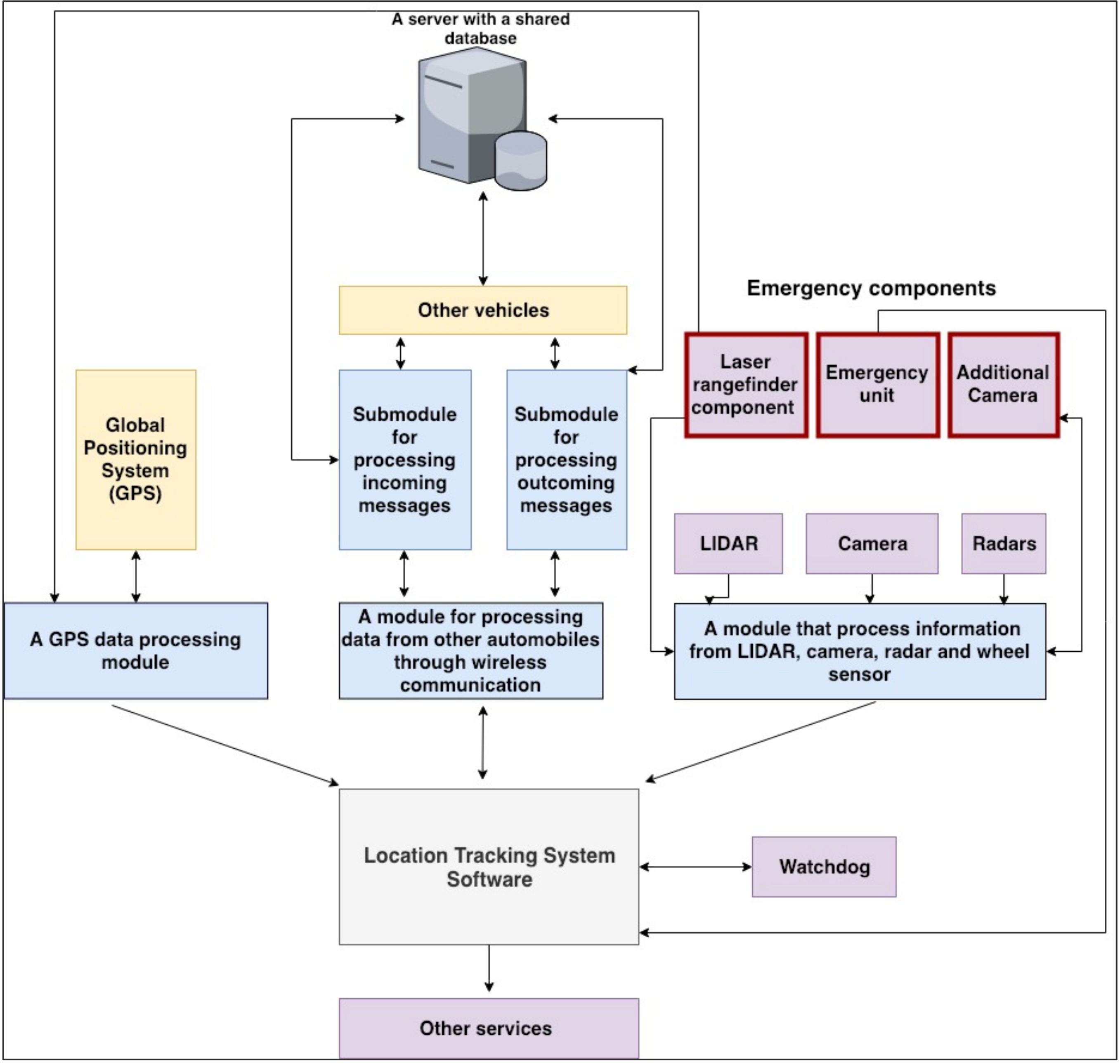
A server with a shared database
Emergency components
Other vehicles
Laser rangefinder component
Emergency unit
Additional Camera
Global Positioning System (GPS)
Submodule for processing incoming messages
Submodule for processing outcoming messages
LIDAR
Camera
Radars
A GPS data processing module
A module for processing data from other automobiles through wireless communication
A module that process information from LIDAR, camera, radar and wheel sensor
Location Tracking System Software
Watchdog
Other services

*Fig. 3. Refined architectural design of an autonomous vehicle*

## 3.3. Updated failure analysis

In the previous section we came up with a new architectural design which would minimize the level of rick and enhance the dependability of the system. However, as new components such as laser rangefinder components, emergency unit, additional camera were introduced, we need to undertake a failure analysis again to identify all possible deviation in the system which could lead to numerous hazards. You can find the updated failure analysis in Fig. 4.

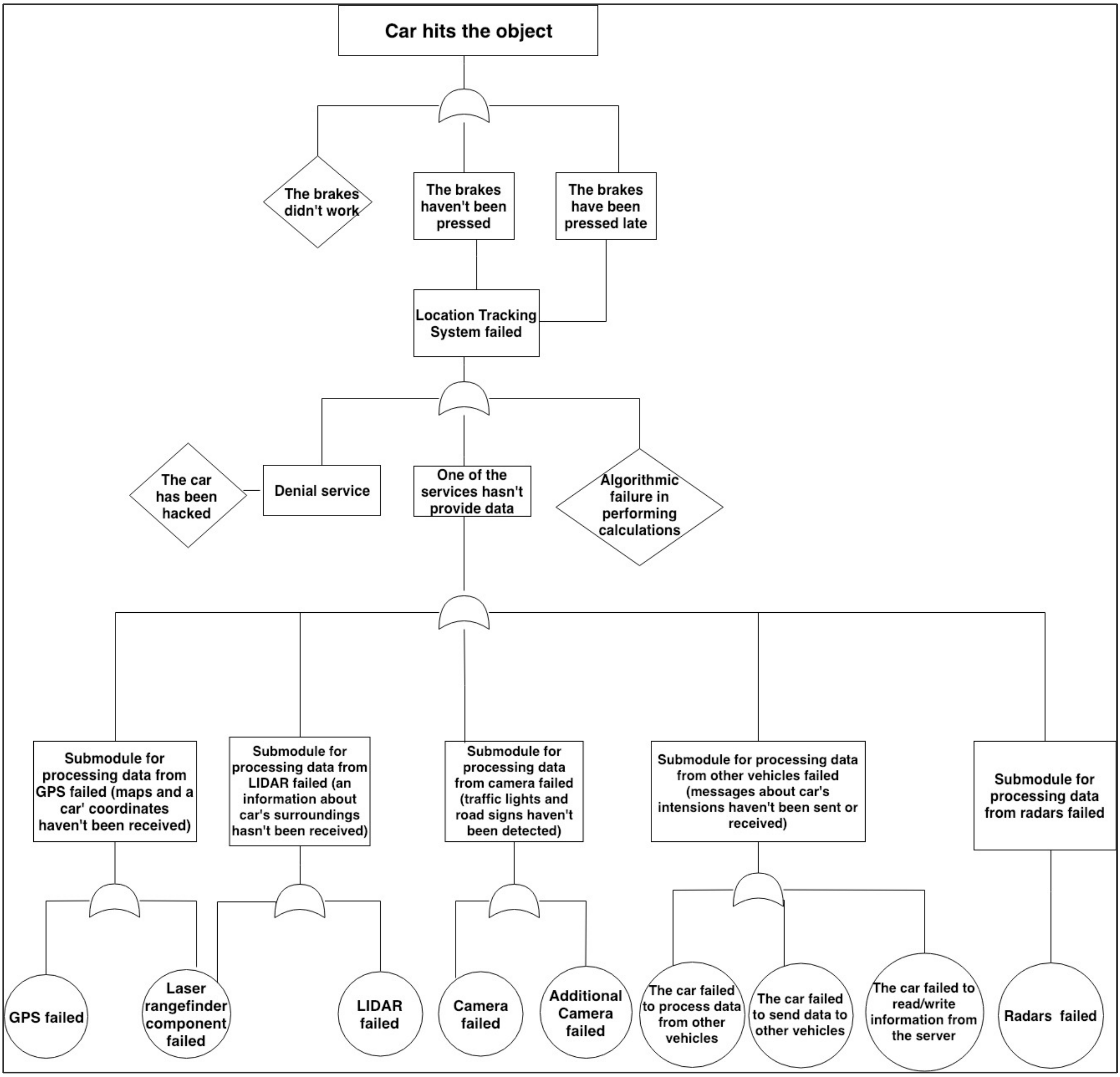
Car hits the object
The brakes didn't work
The brakes haven't been pressed
The brakes have been pressed late
Location Tracking System failed
The car has been hacked
Denial service
One of the services hasn't provide data
Algorithmic failure in performing calculations
Submodule for processing data from GPS failed (maps and a car' coordinates haven't been received)
Submodule for processing data from LIDAR failed (an information about car's surroundings hasn't been received)
Submodule for processing data from camera failed (traffic lights and road signs haven't been detected)
Submodule for processing data from other vehicles failed (messages about car's intensions haven't been sent or received)
Submodule for processing data from radars failed
GPS failed
Laser rangefinder component failed
LIDAR failed
Camera failed
Additional Camera failed
The car failed to process data from other vehicles
The car failed to send data to other vehicles
The car failed to read/write information from the server
Radars failed

*Fig. 4. Updated failure analysis*

# 4. LIST OF RESOURCES

[1] D. Thesis, “A Modular Method for Hazard and Operability Studies of Process Plant By Matthew Jefferson Esq ., MA ( Cantab ), MEng .,” no. December, 1999.

[2] J. A. Mcdermid, D. J. Pumfrey, D. Computing, and S. Centre, “A Development of Hazard Analysis To Aid Software Design.”

[3] J. A. McDermid, M. Nicholson, D. J. Pumfrey, and P. Fenelon, “Experience with the application of HAZOP to computer-based systems,” *COMPASS ’95 Proc. Tenth Annu. Conf. Comput. Assur. Syst. Integrity, Softw. Saf. Process Secur.*, pp. 37–48.

[4] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. 2005.

[5] Q. Xu, T. Mak, and R. Sengupta, “Vehicle-to-Vehicle Safety Messaging in DSRC Qing.”

[6] E. Ruijters and M. Stoelinga, “Fault Tree Analysis: A survey of the state-of-the-art in modeling, analysis and tools.”

[7] P. J. Brooke and R. F. Paige, “Fault trees for security system design and analysis,” *Twort’s Water Supply*, vol. 22, no. 3, pp. 256–264, 2003.

[8] J. C. Knight, “Safety Critical Systems: Challenges and Directions,” 2002.

[9] B. Shin and C. Sciences, “Fault Tolerant Control and Localization for Autonomous Driving : Systems and Architecture,” 2016.

[10] C. A. Jerlin, “Fault Tolerance in Wireless Sensor,” vol. 2, no. 2, pp. 142–146, 2015.