Marte Cecilie Fladmark Fylling

Master's thesis

2019

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Engineering
Department of Marine Technology

Marte Cecilie Fladmark Fylling

Development of Fault Detection Methods for Guidance and Navigation of Underwater Vehicles

June 2019

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Development of Fault Detection Methods for Guidance and Navigation of Underwater Vehicles

Marte Cecilie Fladmark Fylling

Marine Technology
Submission date: June 2019
Supervisor: Asgeir Johan Sørensen
Co-supervisor: Martin Ludvigsen

Norwegian University of Science and Technology
Department of Marine Technology

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER THESIS IN MARINE CYBERNETICS

SPRING 2019

FOR

Marte Cecilie Fladmark Fylling

Title: Development of fault detection methods for guidance and navigation of underwater vehicles

Work description (short description)
As of today, underwater vehicles are highly dependent on surface vessels for docking, and thus for recharging and data transfer. Relevant missions could last anywhere from a few hours to several days, introducing large weather dependencies and high operational costs. By making the underwater vehicles less human-dependent and creating subsea garages, these dependencies can be removed, and the costs can be significantly reduced.

As the underwater vehicles becomes more independent, the requirements for safety and reliability increases. To ensure safe operations, credible signal processing is crucial. This thesis should investigate how external conditions can be included in signal processing of a vehicle operating in an underwater environment with a varying quality/credibility.

Scope of work

1. Review necessary literature within the fields of marine control systems, underwater sensors, signal processing and fault detection.
2. Investigate typical failure modes (FMs) in sensor measurements; outliers, high noise/variance, signal freeze, signal bias and signal drift. In particular it should be studied how to model each relevant FM as a stochastic signal sequence .
3. Investigate the function and structure of different existing fault detection methods including classical and statistical methods.
4. Formulate a test methodology for validation of the studied fault detection methods.
5. Propose a method that integrates external factors concerning the current mission of the vehicle into the signal processing module.
6. Demonstrate the method by a case study using synthetic simulation of an ROV.
7. Document the findings of the study in a report.

The report shall be written in English and edited as a research report including literature survey, description of mathematical models, description of control algorithms, simulation results, model test results, discussion and a conclusion including a proposal for further work. Source code should be provided. It is supposed that Department of Marine Technology, NTNU, can use the results freely in its research work, unless otherwise agreed upon, by referring to the student's work.

The thesis should be submitted within 11. June.

Professor Asgeir J. Sørensen
Supervisor

Prof. Martin Ludvigsen
Co-supervisor

Asgeir J. Soren

Abstract

Underwater vehicles have a wide range of subsea applications where they play an increasingly important role. Such applications include inspection, monitoring, and maintenance of offshore and subsea installations, ocean exploration, search, and rescue. Their growing complexity and level of autonomy make them suitable for tasks conducted in deep water where human divers would perform poorly.

The research of this thesis is motivated by the desire to highlight the importance of safe and credible signal processing as underwater vehicles experience a significant change towards human independence and full autonomy. Safety and reliability are crucial factors if the use of autonomous vehicles is to be adapted and trusted by the public. Different operations, vehicles, and varying environmental conditions have different effects on the sensor measurements that a vessel's control system relies on for its performance. This thesis presents a method that incorporates these external operational conditions to determine the thresholds for fault detection.

A review of the function and structure of existing fault detection methods was conducted, and the algorithms were implemented in a joint test interface. By creating a separate *signal generation module*, different failure modes could be added to the test signals. With these synthetic signals, each implemented algorithm was tested, and their performance was evaluated. Common for all the tested algorithms is the lack of capability to include external conditions when determining the detection limits. Instead, they have to be tuned manually.

The proposed method uses the statistics of the measurements together with a quantified gain called the *vessel operational safety conditions (VOSC)* to determine the *dynamic thresholds*. The *VOSC* is determined by three attributes, namely the *vessel use mode*, the *environmental conditions*, and the *vessel conditions*. By quantifying these conditions, the thresholds can be dynamically changed as the operational conditions change. Thus, more customized and accurate thresholds can be obtained than by manual tuning.

An implementation approach based on the concept of hybrid control systems is discussed. Hybrid systems frameworks are well suited because the large variety in the dynamical behavior for various operational conditions can be captured using different sub-models merged into one hybrid system.

From the results, it can be concluded that the suggested method is a realistic proposal but requires further development and improvement before it can be implemented into a real system. A three-dimensional model, each with three separate categories, to determine the thresholds is too simple. An accurate division of these attributes is crucial for reliable performance. For further work, it is suggested that this three-dimensional model is expanded. It is also proposed to investigate the possibility of automatically quantifying the external conditions. Finally, it is recommended to identify all new risks, failure scenarios, and consequences of introducing the proposed algorithm.

Sammendrag

Undervannsfartøy har et bredt spekter av bruksområder under vann hvor de spiller en stadig viktigere rolle. Slike bruksområder inkluderer inspeksjoner, overvåkning og vedlikehold av offshore- og subsea-installasjoner, samt havforsking, søk og redning. Deres økende kompleksitet og selvstendighet gjør dem egnet for oppgaver som utføres i dypt vann der dykkere ville klart seg dårlig.

Forskningen i denne masteroppgaven er motivert av et ønske om å fremheve viktigheten av sikker og pålitelig signalbehandling nå som undervannsfartøyer opplever en stor forandring mot menneskelig uavhengighet og full autonomi. Sikkerhet og pålitelighet er avgjørende dersom bruken av autonome fartøy skal bli akseptert av offentligheten. Ulike operasjoner, fartøy og varierende miljøforhold har ulik effekt på sensormålingene som et fartøys kontrollsystem er avhengig av for god ytelse. Denne oppgaven presenterer en metode som inkorporerer disse eksterne forholdene for å bestemme grensene for feildetektering.

Det ble utført en studie av funksjonen og strukturen til eksisterende feildeteksjonsmetoder, og algoritmene ble implementert i et felles testgrensesnitt. Ved å opprette en separat *signalgenereringsmodul*, kunne ulike feilmoduser inkluderes i testsignalene. Ved å bruke disse syntetiske signalene ble hver av de implementerte algoritmene testet, og ytelsen ble evaluert. Felles for alle testede algoritmene er mangelen på evnen til å inkludere eksterne forhold for å bestemme deteksjonsgrensene. I stedet må de justeres manuelt.

Den foreslåtte metoden bruker statistikken til målingene sammen med en kvantifisert verdi kalt *fartøyets operasjonssikkerhetsforhold (VOSC)* for å bestemme *dynamiske grenser*. *VOSC* bestemmes ut fra tre bidrag, nemlig *fartøyets bruksmodus*, *miljøforholdene* og *fartøyets forhold*. Ved å kvantifisere disse forholdene kan grensene endres dynamisk når operasjonsforholdene endres. Dermed kan man oppnå mer tilpassede og nøyaktige grenser enn ved manuell justering.

En implementeringsmetode basert på begrepet hybride kontrollsystemer er diskutert. Hybridsystemer er godt egnet fordi den store variasjonen i dynamisk oppførsel for ulike operasjonsforhold kan tas hensyn til ved hjelp av forskjellige delmodeller slått sammen i et hybrid system.

Ut fra resultatene kan det konkluderes med at den foreslåtte metoden er et realistisk forslag, men krever videre utvikling og forbedring før den kan implementeres i et reelt system. En tredimensjonal modell, hver med tre separate kategorier, for å bestemme grensene er for enkel. En nøyaktig inndeling av disse bidragene er avgjørende for pålitelig ytelse. For videre arbeid foreslås det at denne tredimensjonale modellen utvides. Det foreslås også å undersøke muligheten til automatisk kvantifisering av de eksterne forholdene. Til slutt anbefales det å identifisere alle nye risikoer, feilscenarier og konsekvenser ved å introdusere den foreslåtte algoritmen.

Preface

This master's thesis is written at the Department of Marine Technology at the Norwegian University of Science and Technology during the spring of 2019. It is a continuation of the project thesis written during the autumn of 2018.

This thesis would not have been possible without my supervisor, Professor Asgeir Johan Sørensen. Your enthusiasm and expertise have been a great motivation and help for me through this semester. You have given me the freedom to form my own thesis and encouraged me throughout the entire process.

I would also like to thank my co-supervisor Martin Ludvigsen for useful inputs. Finally, I would like to thank my friends, my family, and all of my fellow students for all the support and good discussions.

Trondheim, June 8, 2019

Marte Cecilie Fladmark Fylling

Contents

List of Tables

List of Figures

Nomenclature

Acronyms

ADCP	Acoustic Doppler Current Profiler
AMOS	The Centre for Autonomous Marine Operations and Systems
AUV	Autonomous Underwater Vehicle
CTD	Conductivity Temperature Depth
CUSUM	Cumulative Sum
DP	Dynamic Positioning
DVL	Doppler Velocity Log
Env	Environmental Conditions
FDI	Fault Detection and Isolation
FM	Failure Mode
FMEA	Failure Mode and Effect Analysis
GNC	Guidance, Navigation and Control
GNSS	Global Navigation Satellite System
HIL	Hardware-In-The-Loop
HPR	Hydroacoustic Positioning Reference
HRI	Human-Robot Interaction
IMU	Inertial Measurement Unit
IQR	Interquartile Range
LBL	Long Base Line
LOA	Levels of Autonomy
LS	Least Squares
MUV	Manned Underwater Vehicles
NTNU	Norwegian University of Science and Technology
PDF	Probability Density Function

ROV	Remotely Operated Vehicle
RPN	Risk Priority Number
SBL	Short Base Line
SLAM	Simultaneous Localization and Mapping
USBL	Ultra Short Base Line
USV	Unmanned Surface Vessel
UUV	Unmanned Underwater Vehicle
VC	Vessel Conditions
VOC	Vessel Operational Condition
VOSC	Vessel Operational Safety Condition
VRU	Vertical Reference Unit
VUM	Vessel Use Mode

List of Symbols

$\bar{x}_k/\bar{y}_k$	Average signal value
$\hat{\theta}_t$	Estimate of deterministic signal component at time t
λ	Forgetting factor
μ	Mean value
$\nu/m(k)$	Drift term
σ	Standard deviation
σ^2	Variance
σ_0^2	Nominal variance
σ_{frozen}^2	Variance of frozen signal
$\sigma_{highvar}^2$	Variance of signal with increased noise
σ_i^2	Variance of signal i
$\sigma_{outlier}^2$	Variance of signal with outlier
θ_t	Deterministic signal component at time t
ε_t	Noise at time t
a	Variable that determines the tolerance of the dynamic thresholds
b	Bias
$d(t)$	Drift
$g_t/g(k)$	Averaged cumulative sum of distance measures
h	Threshold value
H^i	Hypothesis number i
$r(t)$	Residual

S_i	Sample set i
$s_t/s(k)$	Distance measure
v/w	Additive noise
$VOSC_{gain}$	Gain that quantifies the safety requirements given a set of operational conditions
$X(t)$	Random variable
X_n	The *nth* data point in the data set
y_{bias}	Biased signal
y_{drift}	Drifting signal
y_{frozen}	Frozen signal
$y_{highvar}$	Signal with increased noise
y_{noisy}	Noisy signal
$y_{nominal}$	Nominal sensor measurement
$y_{outlier}$	Signal with outlier
y_{real}	Undisturbed part of the measurement
A	Gain to determine tolerance of interquartile range limits
$\mathrm{f}_x(\mathrm{x})$	Probability density function of x
Q_i	Quartile number i, where i = 1,2,3
s_i	Weighting factor for signal i
T_f	Time period for filtering
w_i	Weight for signal i
x_w	Weighted signal
y_{fw}	Filtered and weighted signal
y_w	Weighted measurement

Chapter 1

Introduction

The following section presents the background and motivation behind the research of this thesis. It will briefly discuss the motivation for improving signal processing in marine control systems and what the gains of this solution may be. The main research question and objectives of this thesis will be defined and explained. Further, the main contributions will be summarized before describing the thesis outline.

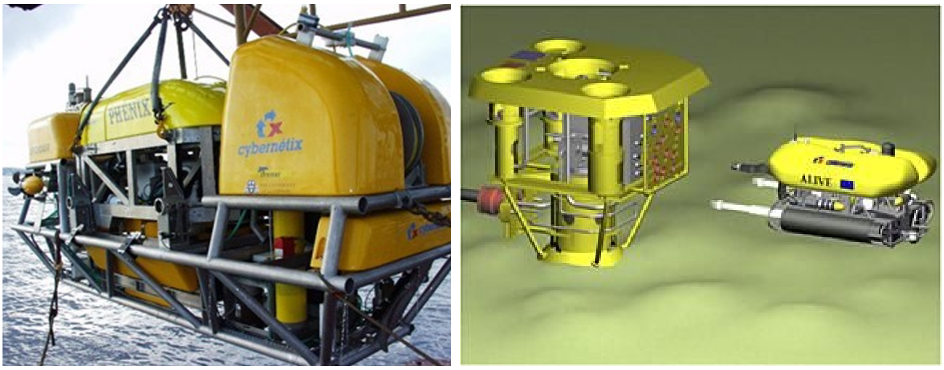
1.1 Background and Motivation

The ocean covers 71% of the Earth's surface, in which more than 80% is still unexplored and unmapped [1]. It is a source of vital resources such as food, energy, and medicine, and it accounts for two-thirds of the oxygen we breathe. The underwater world is dark, deep, and demanding, making it a difficult working environment for humans. During the 1960s, 70s, and 80s, 17 divers lost their lives, and several hundred people got permanent injuries in subsea operations in the Norwegian sector of the North Sea. In the 1990s, remotely operated vehicles (ROVs) began to take over some of the divers' tasks [2]. The use of ROVs improves safety and removes the need for exposing divers to high-risk operations. The future will have an increased demand for subsea inspection, maintenance, and repair operations carried out using underwater vehicles, as subsea installations reach into deeper waters and arctic areas.

As of today, underwater vehicles are highly dependent on the presence of expensive surface vessels during operation. The vehicles are programmed, launched, and recovered, the batteries are recharged, and new missions can be uploaded. This process is done repeatedly until a mission is finished or a survey area is covered. Relevant missions can last from hours, up to days or even weeks, and the needed surface vessels are both expensive and weather and human dependent. The ability to leave the underwater vehicles in situ would remove these dependencies, and they could be activated when needed, resulting in shorter response time in urgent situations. It would also contribute to increasing both the spatial range and the maximum possible duration of a mission without needing human intervention.

During the last years, several important steps have been taken with regards to resident underwater vehicles. Through a collaboration between NTNU AMOS and Equinor, a first of its kind charging station was installed in the Trondheim Fjord, May 2019, at 365 meters depth [3]. Here, technology companies from all over the world can test out their underwater robots and equipment. The station will serve as a subsea garage where underwater robots can live, charge, and pick up new tools. The first to test the station was Eelume with its flexible eel concept [4]. These snake robots are intended to work in the depths of the Åsgard field, after spending a few test weeks in the Trondheim Fjord.

Several companies have visited the concept of resident vehicles. The French company Cybernetix had back in 2001 developed and tested the concept called the Swimmer, where an AUV shuttle carried the ROV to a pre-installed subsea docking station at the seabed [5]. Later, Cybernetix developed the new concept ALIVE, where the shuttle was removed, and an AUV managed to autonomously dock to its pre-installed docking panel and perform some pre-programmed tasks [6]. Both concepts are displayed in Figure 1.1.



(a) Swimmer. Taken from [5]

(b) ALIVE. Taken from [7]

Figure 1.1: Cybernetix' docking concepts. The left figure shows the AUV shuttle intended to transport an ROV to its docking station. The left figure shows the fully autonomous ALIVE.

Other companies have also come up with different solutions, such as Hydroid with their REMUS 100 [8], Oceaneering with their E-ROV [9] and Saab Seaeye with their Sabertooth [10]. All concepts focus on the same goal, which is to remove surface dependencies and thus making the vehicles more autonomous.

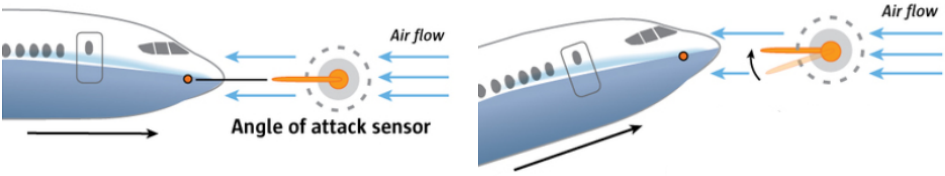
Unmanned and autonomous vehicles are relatively new concepts that challenge the way both ships, cars and airplanes are designed and used. New technology introduces new risks and challenges, and the need for well-defined rules and regulations for autonomous vehicles will keep growing. Safety and reliability will be crucial if the use of unmanned and autonomous vehicles is to be adapted, both by governments and large firms, but also trusted and accepted by the public.

This thesis focuses on the importance of reliable signal processing to ensure safe operations. Although the main focus lies on subsea operations, the principles are just as important for operations above the surface. Two fatal accidents caused by faulty sensor signals are the Lion Air Flight 610 crash and the Ethiopian Airlines Flight 302 crash. Both airplanes were of the brand-new type Boeing 737 MAX 8 and were only two and three months old.

The first crash happened on October 29, 2018, only 13 minutes after take-off from Jakarta, Indonesia. All 189 onboard died [11]. The second crash happened on March 10, 2019, six minutes after the Ethiopian Airlines Flight 302 took off from Addis Ababa, Ethiopia, killing all 157 onboard [12]. Even though the final reports are not expected to be released until August and September 2019, both the Federal Aviation Administration (FAA) and Boeing have explained the accidents as a consequence of the aircraft's new Manoeuvring Characteristics Augmentation System (MCAS) [13].

When Boeing set out to design this new aircraft model, engineers had to find a way to fit their new engine, which is larger and more fuel efficient than previously used engines, under the wing. The engine was moved slightly forward and higher up, causing an upward pitching moment. The MCAS was designed to address this problem by using an adjustable horizontal tail as an automatic stabilizer.

Figure 1.2 and 1.3 show how the new MCAS operates automatically to prevent a stall. The angle of attack sensor, shown in Figure 1.2, aligns itself with the incoming air flow. The measured angle between the wing and the airflow is fed into the flight computer. If the angle gets too high, indicating an approaching stall, the MCAS automatically activates. When activated, the MCAS adjusts the angle of the horizontal tail to move the aircraft nose downward. The crashes happened when the angle of attack sensor fed false information into the flight computer, and the aircraft crew failed to override the action due to lack of training at the time [14].



(a) Level flight

(b) Nose-up flight

Figure 1.2: Angle of attack sensor. Taken from [15]

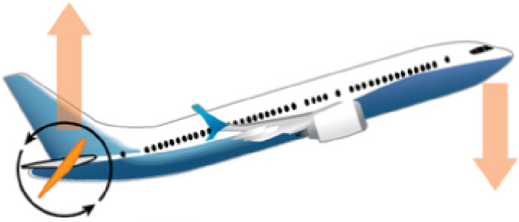


Figure 1.3: Horizontal tail. Taken from [15]

Another relevant example was lectured in the course *Marine Control Systems I*, to highlight the importance of integrating external conditions in signal processing. A drill ship was operating in a strict dynamic positioning (DP) control mode in harsh weather conditions. When a high amplitude set of waves hit the hull of the ship, it was thrown out of position. Due to the abrupt and large change in value, the control system reacted by discarding the position measurement as impossible. The operation ended with an emergency disconnect of the drilling string. This incident caused damage to equipment and costly delay of the operation.

These examples show that when operating with automatic or autonomous systems, a faulty signal may have fatal consequences. The cost of such an accident can be enormous. To obtain the safest and best possible performance of a control system, it must work with reliable signals. To ensure this, a dedicated *signal processing module* should be included to monitor sensor measurements continuously, and should be able to alarm the system or system operator if significant faults or abrupt changes in the signals are detected.

A challenge of processing signals in marine control systems is to include all available information about external conditions in the decision of whether a signal is valid or not. When a marine vessel is subjected to strong environmental forces, an abrupt change in mean may be a weaker indication of fault than if it were operating under calm conditions. How strict the signals are processed should be dependent on external factors to obtain the most accurate possible signal processing. This thesis presents a method that incorporates external operational conditions to determine the thresholds for fault detection.

1.2 Research Question and Objectives

The main research question addressed in this thesis is formulated as:

When studying signal processing for autonomous subsea operations, how can the limiting thresholds be dynamically set based on available information about the environmental, operational and vessel conditions?

The main objectives to achieve this include:

- Gain a complete overview of the necessary theoretical background within the fields of underwater vehicles, marine control systems, statistics, signal processing, and fault detection.
- Thoroughly investigate different failure modes and how they can be mathematically modeled.
- Research the function and structure of existing methods of fault detection. Built a joint test interface to study the performance of the researched methods.
- Create a module that implements the failure mode models, such that polluted test signals can be generated.
- Propose a method for dynamically determining the thresholds.
- Conduct a case study, involving a simulation based on a real scenario, to test the proposed method.

The results of the final case study are discussed concerning the research question to conclude if the proposed method is a realistic proposal and what the main challenges are. Lastly, a proposal for further work is presented.

1.3 Main Contributions

This thesis main contributions consist of four parts, followed by a general discussion of the outcomes and limitations of the proposed method. Further follows some concluding remarks and suggestions for further work.

The main contributions can be summarized as:

- Theoretical background covering the basic theory of marine systems, statistics, signal processing, and fault detection needed to gain an understanding of the methods used later in the thesis. This includes conducting a study of the different possible failure modes, how they behave, and how they can be modeled.
- Development of a testing methodology and interface for fault detection. This includes a thorough study of the structure and function of existing fault

detection algorithms. It also requires an approach to create polluted test signals, such that the algorithms can be tested and validated.

- A proposed method for dynamically determining the thresholds used in fault detection based on available information about the vessel operational conditions
- Case study with simulations of a real-life scenario to illustrate and test the proposed solution. The simulation signals are created using a generated signal to reveal weaknesses and/or strengths.

1.4 Thesis Outline

This thesis is divided into six chapters, including:

Chapter 2 - Background Material
This chapter consists of twelve sections. The first three sections cover the basics of underwater vehicles, autonomy, and guidance, navigation, and control of marine vessels. The fourth chapter describes commonly used underwater sensors, as these differ from those used above the surface. The last sections present relevant background theory about statistics, signal processing, and fault detection. Failure mode and effect analysis, hardware-in-the-loop simulation and functional barriers are presented and described in the context of marine vessels. It is also included a description of different failure modes and how they can be modeled.

Chapter 3 - Tools for Development of Fault Detection Methods
The basis for being able to propose a new method is a solid understanding of how existing methods are built up and how they function. It also requires a test platform to test different algorithms. This chapter presents a thorough study of some existing methods, trying to cover each of the presented failure modes. It also describes how these methods are implemented into a joint test interface, where different algorithms can be tested on appropriate test signals. To create these signals, a separate *signal generation module* is created, using the mathematical model of each failure mode to simulate faults.

Chapter 4 - Dynamic Thresholds
This chapter presents a proposed method for dynamically determining the thresholds for fault detection. The method focuses on how different external factors affect the safety requirements of the operation and how these effects can be quantified. The objective of the method is to be able to determine thresholds not only based on trial and error, but rather based on the available information about each operation.

Chapter 5 - Case Study: Simulation with Dynamic Thresholds
The case study tests the proposed fault detection method by simulating a real-life scenario. The scenario is explained in detail to be able to quantify all relevant

values of the operation. A synthetic signal is generated to simulate the chosen scenario, both in a fault-free and a faulty situation.

Chapter 6 - Discussion
This chapter presents a general discussion of the implemented methods in light of the research question. The performance of the proposed method is evaluated based on the results obtained in the case study. This chapter also discusses the potential limitations and challenges of the proposed method and investigates how it could be integrated into a real control system.

Chapter 7 - Conclusions and Further Work
The final chapter covers the concluding remarks of the thesis, together with a recommendation of challenges that should be addressed in further work.

Chapter 2

Background Material

Marine systems are designed to perform complex missions that require a high level of safety and reliability. The performance of these tasks relies on an accurate control system, which again depends on accurate and reliable signals. A single fault can have fatal consequences, such as collision, damage to subsea structures, or loss of vehicle. The following chapter covers the essential background theory of underwater vehicles, autonomy, guidance, navigation, and control systems, and the most commonly used underwater sensors. Further, it presents the relevant statistical concepts, including stochastic processes, random variables, the normal distribution, white noise, and hypothesis testing, which are prerequisites to understand the fundamentals of signal processing. It also introduces fault detection and diagnosis, where FMEA and HIL testing play an important role. In this context, it defines different failure modes and investigates how they can be modeled.

2.1 Underwater Vehicles

Underwater vehicles have undergone a phase of rapid increase in capability in recent years. They allow easy access to deeper oceans and can conduct inspections, maintenance, and repairs of subsea structures, in addition to ocean mapping of areas that are too demanding for human divers. Underwater vehicles can be divided into two main subcategories, namely *Manned Underwater Vehicles* (MUVs) and *Unmanned Underwater Vehicles* (UUVs) depending on if the vehicle is designed to have a human occupant or not. The term UUV is a generic expression that includes both *Remotely Operated Vehicles* (ROVs) and *Autonomously Underwater Vehicles* (AUVs). ROVs can further be divided into different categories, mainly based on weight, size, and intended operations. The main division of underwater vehicles is illustrated in Figure 2.1.

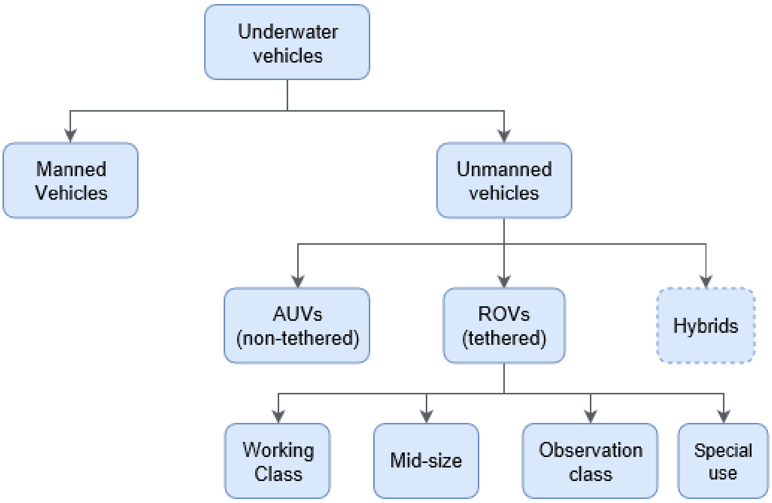


Figure 2.1: Classification of underwater vehicles. Based on Figure 7 in [16].

An ROV denotes an underwater vehicle with a physical link, a tether or umbilical, to a remote human operator located on a surface vessel or in a submarine. This link gives unlimited power, data transfer, and high bandwidth communication, but it also limits the spatial coverage and maneuvering capacity due to drag forces on the cable. It does, however, have high capabilities for doing light intervention and sampling.

AUVs can operate entirely independent of direct human input, thus relying on an onboard power system and intelligence. AUVs are mostly used in ocean mapping due to its good hydrodynamic and maneuvering capabilities. AUVs with manipulator capabilities are under development, and several attempts have been made, such as the ALIVE and Swimmer concepts by Cybernetix, which are mentioned in Section 1.1.

The development of AUVs with manipulator capabilities and ROVs independent of a physical connection has led to a third category, namely hybrids. Typical ROVs may be slow due to its size and shape, while AUVs are fast with its torpedo shape. However, ROVs usually have better stability and intervention capabilities than AUVs. By combining the best features from both, the result is a vehicle with greater capabilities than each one has individually. Eelume's snake robot, presented in Section 1.1 is a good example of a hybrid. Its flexible shape can act both as a torpedo-shaped AUV and as an intervention arm. It can operate both with and without an attached cable and can change tools depending on the mission.

2.2 Autonomy

As this thesis considers methods of signal processing in the context of autonomous subsea vehicles, it is relevant to define what autonomy is. Autonomy is a broad subject and can be defined in several different ways. One of these definitions states that an autonomous system is a self-governing and independent system [17]. Considering an AUV, this means that it can operate without human intervention for extended periods of time. When assigned a mission, it can plan, re-plan, and make its own decisions based on sensor input and observations of the environment. It is common to distinguish between the terms *automatic* and *autonomous.* While automatic systems can perform well-defined tasks automatically without human intervention, autonomous systems are designed to perform complex tasks that may contain significant uncertainties in an unstructured environment [18].

Autonomy is commonly divided into different levels of autonomy (LOA), characterized by the level of human-robot interaction (HRI), mission complexity, and environmental complexity. According to [19], the four levels of autonomy can be defined as:

1. **Automatic operation (remote control):** Even though the system operates automatically, the human operator directs and controls all high-level mission-related functions. These functions are often pre-programmed. Also referred to as *human-in-the-loop.*
2. **Management by consent (tele-operation):** The system automatically recommends actions for selected mission-related functions and prompts the operator at key points for specific information and decisions. At this level, the system may have limited communication bandwidth, including time delays. The system can perform many of the functions independently of human control when assigned to do so. Also called *human-delegated.*
3. **Management by exception (semi-autonomous):** The system automatically executes mission-related functions in situations where the response time is too short for operator intervention. The operator is alerted of the progress and may override or change parameters and cancel or redirect actions. Exceptions may be brought to the operator's attention for decisions. Also called *human-supervisory control.*
4. **Highly autonomous:** The system automatically executes mission-related functions in an unstructured environment with the ability to both plan and re-plan the mission. The system is intelligent and independent. Also referred to as *human-out-of-the-loop.*

For autonomous systems, an important control and safety function is related to situation awareness. The integrity of situation awareness is dependent on a set of reliable signals.

2.3 Guidance, Navigation and Motion Control

The accuracy and safety of a vehicle's mission rely on a reliable control system, which is responsible for automatic control of position, attitude, velocity, and acceleration. A control system is usually constructed as three separate blocks denoted as the guidance, navigation and motion Control (GNC) systems. These three blocks interact with each other through data and signal communication and can either be fully independent of each other or coupled to different degrees [20].

The guidance system provides the desired positions and trajectories that the vessel should maintain or follow according to the requirements of the operator or the mission. The navigation system is responsible for providing reliable feedback signals with information about the vessel motion such as position, attitude, and velocity. This information is further passed as feedback to the guidance and control systems. The motion control system generates adequate control commands to the vessel's actuators in order to maintain the desired position or follow the generated path.

A DP system is a simple example of a GNC system, as illustrated in Figure 2.2. The path planner block makes up the guidance system and creates a feasible path for the vehicle to follow. The sensors and the observer form the navigation system, keeping track of the vehicle's position, heading, velocity, and attitude. Finally, the controller tries to minimize the difference between the actual motion of the vessel, given by the navigation system, and the desired motion of the vessel as given by the guidance system.

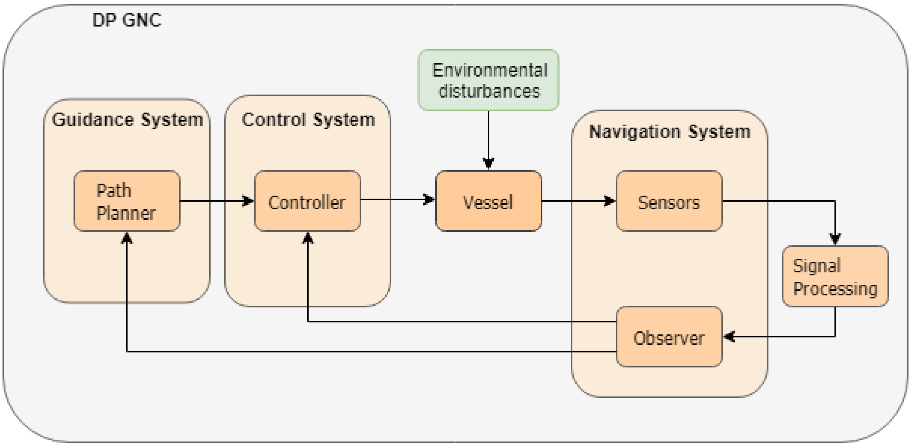


Figure 2.2: DP GNC system. Based on Figure 9.3 in [20].

A more detailed description is presented below as they are explained in [20].

2.3.1 Guidance

Guidance is the process of determining the path of a vessel towards a given setpoint, which can be either constant or moving. The system continuously computes the reference position, velocity, and acceleration of the vessel to be used by the motion control system. In its simplest form, guidance systems are used to generate a reference trajectory for time-varying target tracking or a planned path for time-invariant path following. In control literature, the different scenarios are classified as:

- **Setpoint regulation:** The most basic guidance system is where the input is constant or provided by a human operator. The corresponding controller will in this case be a regulator.
- **Trajectory tracking:** The objective is to make the vehicle's position and velocity follow a desired time-varying reference value. This desired output can be generated using reference models made of low-pass filters.
- **Path following:** The goal is to follow a predefined time-invariant path. No restrictions are placed concerning the temporal propagation along the path.

The guidance system can use external inputs, such as wind, current and wave measurements, earth topological data, obstacle, and collision avoidance data, and the state vector measurements to create a feasible trajectory for motion control. Feasible means to create a trajectory that is consistent with the vessel dynamics.

2.3.2 Navigation

Navigation is the technique of directing the vessel by determining its position, attitude, and course, as well as velocity and acceleration in some cases. This is usually done by using a combination of a global navigation satellite system (GNSS) or underwater acoustics and motion sensors such as accelerometers and gyros. A further explanation of different commonly used sensor systems can be found in Section 2.4.

Control systems for ships and underwater vehicles are commonly implemented with a state estimator to process the sensor and navigation data. Raw measurements usually have to be monitored and handled by its own signal processing unit for validation and quality checking. These measurements are further used as input to the navigation system. This system contains a state estimator, such as a Kalman filter, which can filter noise, predicate, estimate, and reconstruct unmeasured states.

2.3.3 Motion Control

Motion control is the action of determining the control forces and moments necessary in order to satisfy a given control objective, such as path following. The output from the navigation system is used for feedback control, while the signals available from the guidance system and external sensors are used in feedforward control. Commonly used control systems are based on techniques such as conventional Proportional-Integral-Derivative (PID) control, Linear Quadratic Regulators (LQR) or nonlinear control.

2.4 Underwater Sensors

Accurate positioning of the vehicle is essential in all phases of both surface and underwater navigation. Accurate position measurements are crucial for both operational and safety reasons. Underwater navigation introduces several problems compared to navigation above the surface. The main problem is the rapid attenuation of higher frequency signals. Underwater signals are only able to travel a short distance, and thus excludes the use of GNSS, which is commonly used for navigation above the surface.

It is common to divide underwater sensors into two main groups, namely *payload sensors* and *navigation sensors*. The payload sensors are the measurement units that collect the data either by remote sensing or by direct measurements [18]. As these measurements are often exposed to noise and disturbances, signal quality control is important. The navigation sensors measure the state of the vehicle, which is further used by an internal control system to position the vehicle correctly. Considering *Simultaneous Localization and Mapping (SLAM)* [21], payload sensors will also be integrated into navigation. Even if these will not be addressed in detail in this thesis, signal processing methods used in navigation are also applicable here. The most commonly used payload and navigation sensors will be presented with a brief description below.

2.4.1 Payload Sensors

Acoustic Doppler Current Profiler
Acoustic Doppler Current Profilers (ADCPs) are hydroacoustic current meters that provide velocity profiles of the water columns in front of the instrument. The ADCP transmits an acoustic signal and measures backscatter intensity and Doppler shift of the reflected signal. The signals are usually transmitted from four piezoelectric transducers that are angled in different directions. Based on these measurements, the instrument can provide a three-dimensional current profile of the measured water column [18]. An ADCP can also work as an acoustic Doppler Velocity Log or as a bottom-track by measuring its relative velocity to the seabed instead of the water.

Conductivity Temperature Depth Sensors
A Conductivity Temperature Depth (CTD) sensor is an oceanographic instrument that measures the conductivity, temperature, and pressure of seawater. These measurements are used to calculate the water's salinity, speed of sound, depth, and density. The conductivity is a measure of how well a solution can conduct electricity and is directly related to the solution's salinity. Combining the temperature and salinity measurements can determine the density of the water. Density and salinity are important properties in oceanography, while the speed of sound is crucial for sonar applications such as seabed mapping or acoustic navigation [18].

Vision Systems
Optical imaging of the seabed can provide high-resolution data about the color, shape, and texture of the seabed. This is still the most reliable method for identifying objects underwater [18]. However, there are many challenges when it comes to obtaining quantitative data from optical imaging. Light absorption and backscatter limit the range for cameras.

Active Sonars
Active sonars are used to measure reflected acoustic signals documenting objects on the seabed or in the water column [18]. By using receiver and transmission beams with known directions, each impulse can establish (x,y,z)-points, and thus create a three-dimensional model of the seabed.

2.4.2 Navigation Sensors

Hydroacoustic Position Reference Systems
Hydroacoustic Position Reference (HPR) systems determine position using the time of flight of transmitted sound waves. HPR systems use one or several markers located on a fixed position on the seabed (or on a surface vessel) plus one or several transducers mounted on the vessel to be positioned. Underwater acoustic navigation and position systems may use various types of markers. These markers can be used as either transmitters or receivers, or as both. The markers can be divided into five main classes as defined in [22]. These classes are as follows:

- *Transducer:* A transmitter (or receiver) often mounted on the hull which sends out an interrogation signal with a given frequency and receives a reply on a second frequency.
- *Transponder:* The receiver (or transmitter) is installed on the seabed or on an underwater vehicle which receives an interrogation signal on one given frequency and sends out a response signal on a second frequency.
- *Beacon/Pinger:* A beacon/pinger is a transmitter attached to the seabed or to a submersible which continuously sends out a pulse on a particular frequency.
- *Hydrophone:* A hydrophone is a directional/omnidirectional receiver that is mounted on the hull and can receive a reply from a transponder or a

beacon/pinger.

- *Responder:* A responder is a transmitter mounted on the seabed or on a submersible, which can receive a control signal to transmit an interrogation signal for a transducer or a hydrophone.

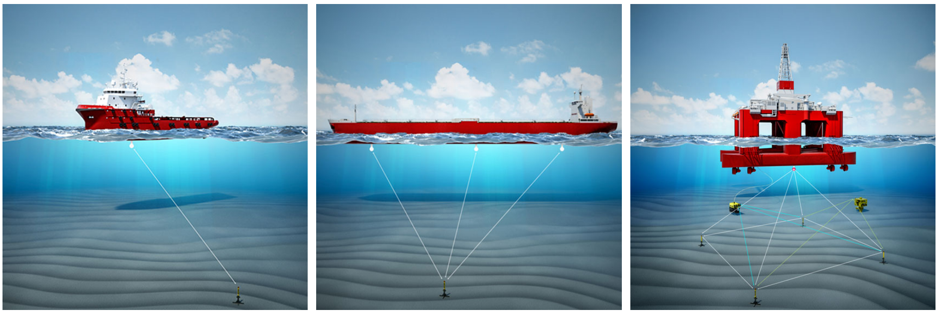
Hydroacoustics
The theory of sound is similar to that of light as both consist of waves propagating through a medium. The sound waves are subject to scattering, reflection, and absorption. As water is an imperfect acoustic medium, energy may be removed from the signal and converted into heat by physical absorption [16]. The propagation of a sound wave is associated with acoustic energy. The propagation velocity is dependent on both salinity, density, and temperature of the medium.

Short Baseline (SBL)
The name *short* refers to the distance between the transducers, which are located on the hull of the vessel. The baselines range from 5 to 20 meters. A minimum of three transducers are required, but four is commonly used for redundancy in the system. The transducers on the hull of the vessel receives acoustic signals from a transponder/beacon mounted on the underwater vehicle. The travel time of these signals are used to calculate the heading and distance relative to each transducer. This gives a three-dimensional position of the underwater vehicle.

Ultra-Short Baseline (USBL)
In a USBL system, the three separate hydrophones on the vessel's hull are replaced by one single hydrophone with three sensors. This simplifies the installation, as the calibration only must be done for one transducer instead of three. The position estimate of the underwater vehicle is based on the range and on the vertical and horizontal angle measurements from the transducer. The ship- or seabed-mounted transducer sends out an acoustic signal that is picked up by the transponder mounted on the underwater vehicle. By using the response time, range and bearing can be calculated. The bearing angle is found by using the phase difference of the acoustic signal.

Long Baseline (LBL)
The term *long* baseline comes the spacing between the baseline transponders, which can be several kilometers long. This type of acoustic system generally gives more accurate measurements than both SBL and USBL. The transponders mounted on the seabed each replies on different frequencies, enabling their signals to be distinguished from each other. The minimum number of transponders needed for unambiguous navigation in three dimensions is three, but four are usually used for redundancy.

The vehicle's position is determined by acoustically measuring the distance from the interrogator on the vehicle to the transponders on the seabed. This is done by recording the time of the interrogation together with the time of reply. The total signal run time is then used to calculate the range. This process is done simultaneously for all the transponders, giving an estimated position of the vessel relative to the array of transponders.



(a) USBL **(b)** SBL **(c)** LBL

Figure 2.3: Hydroacoustic position reference systems. Taken from [23]

Figure 2.3a shows a USBL system for positioning of a surface vessel. The USBL system only requires one transducer mounted on the vessel and one transponder mounted on the seabed. Figure 2.3b shows an SBL system used to position a surface vessel relative to a transponder mounted on the seabed. The SBL system consists of four transducers mounted on the hull and one transponder mounted on the seabed. Figure 2.3c shows a case where an LBL system is used to position two ROVs relative to a rig. The LBL system has four transponders mounted on the seabed, one transducer on each of the ROVs and one on the rig.

Gyrocompass

A gyrocompass is a non-magnetic compass using a fast-spinning disc and the Earth's rotation to find the geographical direction. The system will converge to a steady state of minimum potential energy, where its axis is aligned with the Earth's north-south axis. Gyrocompasses are widely used for ship navigation, replacing the magnetic compasses. They have the advantage that they find the *true* north as determined by Earth's rotation, which is more useful in the context of navigation than the *magnetic* north. Also, they are not affected by magnetic disturbances, such as the vessel's steel hull. The drawback of gyrocompasses is that it requires time to converge. This makes it more suitable for a ships than planes, which have faster dynamics and therefore require other techniques to handle the rapid changes [24].

Inertial Measurement Unit

An Inertial Measurement Unit (IMU) uses accelerometers, gyrocompasses and in some cases magnetometers to keep track of linear acceleration and angular velocity. IMUs are commonly used in Inertial Navigation Systems (INS), which utilizes these measurements to calculate the position and heading of an object relative to a given starting point, heading and velocity [25].

Vertical Reference Unit

A Vertical Reference Unit (VRU) measures heave, roll, and pitch motions of the vessel. In some cases, it also measures the angular velocities. The main function of a VRU is to adjust the position measurements provided by the position reference system for roll and pitch motions [26].

Doppler Velocity Log
A Doppler Velocity Log (DVL) is an acoustic sensor that estimates the velocity relative to the seabed. It contains piezoelectric oscillators to transmit and receive sound signals. The DVL measures Doppler shift, i.e., the frequency shift of the echo in the incoming signal reflected off the seabed or water columns. Velocities of three dimensions are obtained by having several transducers pointing in different directions.

2.5 Random Signals and Noise

The concept of random variables is suitable to deal with unpredictable fluctuations in measurements. It enables the ability to come up with a probabilistic characterization of a random quantity, such as noise. This resulting theory is called the theory of random or stochastic processes. As noise is present in most real systems, modeling of noise plays an important part in signal processing and fault detection. Noise is usually modeled as a random variable with certain properties. This section will therefore describe what a random variable is and describe its most important properties.

2.5.1 Stochastic Processes

A stochastic process is a mathematical object that is widely used to model quantities for which there is no way to predict an exact value at a future instant of time [27]. It is defined as a family of random variables, and can be denoted as:

$$\{X(t)\}_{t \,\epsilon\, T}\,, \tag{2.1}$$

where t refers to the time. This means that at every time t in the set T, a random number $X(t)$ can be observed.

2.5.2 Random Variables

As mentioned, most real systems are subjected to noise [28]. Noise is an input variable which is not measurable, and which has an unknown behavior. These noise variables usually represent a wide range of different effects that are acting on the studied physical plant, its measurements, and its control system. Noise variables are represented as random functions of time that can be characterized by their average properties and probability density functions.

Probability Density Function
A random variable can be defined as a real function that maps each element of its sample space S into points of the real axis. A random variable is usually represented by a capital letter $(X,\ Y,\ Z)$, where a lowercase letter $(x,\ y,\ z)$ denotes a specific real value of the random variable. Considering a continuous random variable X, its distribution function $F_X(x)$ may be written as:

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{X} f_X(u)du, \tag{2.2}$$

where $f_X(x)$ is called the probability density function (PDF). This function describes the relative likelihood for a random variable to take on a given value. The integral of the PDF between two limits gives the probability of the random variable to take on a value between those limits:

$$p(\theta_1 < X < \theta_2) = \int_{\theta_1}^{\theta_2} f_X(\theta)d\theta. \tag{2.3}$$

By definition, a PDF must satisfy:

$$f_X(x) \geq 0 \ \text{ for all x}, \tag{2.4}$$

and

$$\int_{-\infty}^{\infty} f_X(\theta)d\theta = 1. \tag{2.5}$$

Mean
The *mean* or the *expectation*, *E(X)*, of the random variable X is defined as:

$$E(X) = \mu = \int_{-\infty}^{\infty} x f_X(x)dx. \tag{2.6}$$

Variance and Standard Deviation
The variance σ_X^2 is a measure of the dispersion of a random variable about its expected value and is expressed by:

$$\sigma_X^2 = E[(X - E(X))^2)] = E(X^2 - (E(X))^2, \tag{2.7}$$

where σ_X is called the standard deviation.

2.5.3 The Normal Distribution

The normal (Gaussian) distribution plays an important role in the analysis of random signals. It is often used to represent random variables whose distributions are unknown, as this distribution is a fairly accurate characterization of many random physical processes. The central limit theorem [28], states that the behavior of variables representing the combined effect of a large number of phenomena tends to converge towards the normal distribution.

The normal distribution is often called the bell curve due to its shape, as shown in Figure 2.4. The distribution is symmetric about the mean value, and values closer to the mean are more likely to occur than values further away.

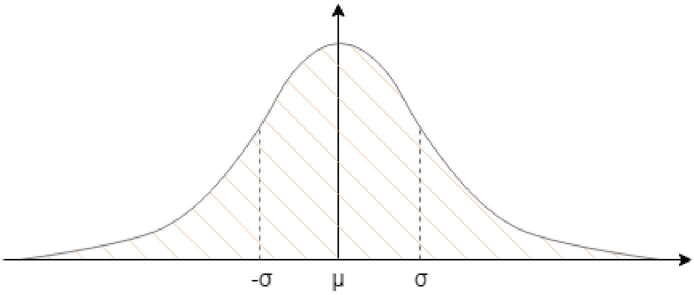


Figure 2.4: The normal distribution. Based on Figure 4.12 in [29].

The PDF of a normally distributed variable can be expressed as:

$$f_X = \frac{1}{\sqrt{2\pi}\sigma} exp\left[-\frac{1}{2\sigma^2}(x-\mu)^2\right], \tag{2.8}$$

where μ is the mean, σ is the standard deviation and σ^2 is the variance. The distribution only depends on the mean and variance, and the random variable x can be described using the following notation:

$$x \sim \mathcal{N}(\mu, \sigma^2). \tag{2.9}$$

2.5.4 Additive White Gaussian Noise

Additive white Gaussian noise is a basic noise model that is widely used to mimic the effect of many random processes that occur in nature. White noise is defined as a stationary random process with equal intensity at different frequencies, giving it a constant spectral density function. The term *white* refers to white light, which has uniform power across the whole frequency band. The term *additive* specifies that the noise is added to the original signal, such that the noisy signal can be modeled as:

$$y_{noisy} = y_0 + w. \tag{2.10}$$

The term *Gaussian* refers to the Gaussian probability distribution of the noise samples.

2.6 Hypothesis Testing

Before moving on to failure modes and signal processing, it is convenient to introduce the theory of hypothesis testing. Hypothesis testing is one of the central subjects of mathematical statistics and represents the basic logic and reasoning behind fault detection [28]. The main objective is to make a decision based on the statistical properties of the time series from which the observations originate.

The decision problem is usually presented as a choice between two or more hypotheses. If it involves only two hypotheses, the decision problem is called *binary*. Otherwise, it is called *multiple*. One of the hypotheses is referred to as the *null hypothesis* and is denoted as H^0. The other hypotheses are referred to as the *alternative hypotheses*, denoted as $H^j,\ j = 1, ..., k$. In fault detection, the decision problem can be stated as the choice between two hypotheses, namely a fault-free mode or a failure mode. Similarly, it can also be formulated as a *multiple* decision problem, where each hypothesis represents each possible failure mode. In this case, the fault-free mode can be set as the null hypothesis, while each of the failure modes is the alternative hypotheses.

A typical example to illustrate a binary decision problem is the decision whether the mean of a time-series is zero or another value. That is:

$$H^0 : \mu = 0 \tag{2.11}$$

$$H^1 : \mu \neq 0. \tag{2.12}$$

2.7 Failure Modes

This section presents different failure modes and their behavior. The objective is to find a mathematical model that describes each of them, to be able to simulate their behavior. Given the information about stochastic processes in Section 2.5, additive white Gaussian noise can be used to model sensor noise. Thus, the sensor measurements can be modeled as the sum of two parts: a signal part y_{real} and an additive noise part v:

$$y_{nominal} = y_{real} + v, \qquad v \sim \mathcal{N}(0, \sigma^2), \tag{2.13}$$

which can be expressed as:

$$y_{nominal} \sim \mathcal{N}(y_{real}, \sigma_0^2). \tag{2.14}$$

Eq. (2.14) represents a nominal fault-free measurement, with a mean that is equal to the mean of the original signal and variance that includes the additive noise. If a signal fault occurs, the distribution equation must be modified to account for the failure. The most common failures that might occur, which will be explained further in this section, are:

- Outliers
- Signal freeze
- High variance
- Sensor bias
- Sensor drift

Outliers

Outliers, commonly called wild points, are measurements that deviate significantly from the expected value, resulting in a low signal to noise ratio. Such a low ratio indicates that the measurement has low credibility and gives no useful information. As can be interpreted from the name, outliers generally occur only for a single or few measurements before the noise level returns to nominal values. A graphical illustration showing the appearance of an outlier is given in Figure 2.5.

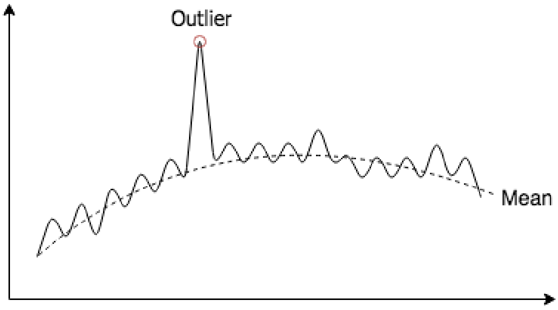


Figure 2.5: Signal sequence with an outlier

An outlier can be modelled as:

$$y_{outlier} = y_{real} + v_{outlier}, \tag{2.15}$$

where $v_{outlier}$ is assumed to be a set of independent random variables, which can be modelled as variables under a normal distribution, expressed as:

$$y_{outlier} \sim \mathcal{N}(y_{real}, \sigma^2_{outlier}), \qquad \sigma^2_{outlier} \gg \sigma^2_0. \tag{2.16}$$

A measurement is usually considered an outlier if it is located outside an accepted interval that can be defined as:

$$y_k \in [\bar{y}_k - \alpha\sigma_0, \bar{y}_k + \alpha\sigma_0], \tag{2.17}$$

where α often is set to be in the interval 3-9 [26]. σ_0 denotes the expected standard deviation in the nominal case, and $\bar{y_k}$ is the signal mean at a given time.

Signal Freeze

A frozen signal is when a measurement is exactly equal to its previous value for a significant number of consecutive measurements. Thus, when a signal freezes, its variance is zero if compared to its previous measurement. A frozen signal can accordingly be modeled as:

$$y_{frozen} \sim \mathcal{N}(y_0, 0), \tag{2.18}$$

where y_0 denotes the signal value of the frozen signal. An illustration of this phenomenon is shown in Figure 2.6.

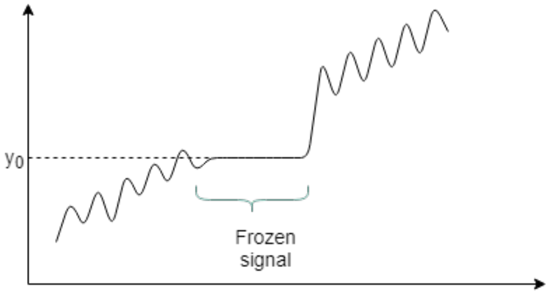


Figure 2.6: Signal sequence with a frozen signal

High Variance

The phenomenon of high variance is similar to outliers, but where outliers only occur for a single or few measurements, a high variance can occur over a longer period before returning to nominal values. High noise can be modeled as a signal with the nominal mean, but with increased variance, as expressed in (2.19).

$$y_{highvar} \sim \mathcal{N}(y_{real}, \sigma^2_{highvar}), \qquad \sigma^2_{highvar} \gg \sigma^2_0. \tag{2.19}$$

Sensor Bias

A sensor bias appears as a sudden change in the signal mean, whereas the noise level remains the same. The biased measurement can be modeled as:

$$y_{bias} \sim \mathcal{N}(y_{real} + b, \sigma^2_0), \tag{2.20}$$

where b denotes the deviation from the nominal mean.

Sensor Drift

Sensor drift is described by a change in the signal mean, while the variance is kept constant. As opposed to the constant bias mentioned above, a sensor drift varies with time. A sensor drift model can be defined as:

$$y_{drift} \sim \mathcal{N}(y_{real} + d(t), \sigma_0^2), \tag{2.21}$$

where *d(t)* denotes the time-varying deviation from the nominal mean.

Note that both sensor drift and sensor bias are difficult to detect and handle without redundancy in the measurements. If a system has two sensors measuring the same state, an increasing deviation between the two measurements is an indication of drifting. In the case where the system has three or more sensors for the same value, it can also determine which measurement to discard. Figure 2.7 shows an example of two sensors measuring the same value. The deviation starts increasing, and the system must choose to trust either sensor 1, sensor 2, or the average.

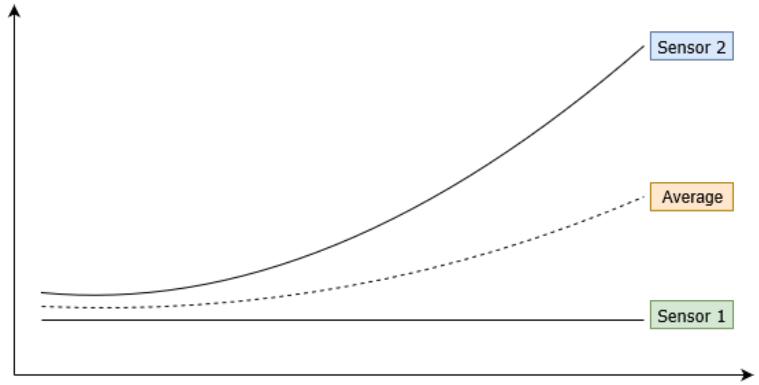


Figure 2.7: Drifting of sensor signals

Summary

By modeling the different failure modes as above, fault detection is reduced to the problem of determining which distribution a signal likely belongs to. This means that the failure modes can be detected by studying either a change in the mean value or a change in the variance, given the nominal values y_{real} and σ_0^2. Table 2.1 below summarizes the stochastic models of different sensor failure modes.

Table 2.1: Stochastic modelling of sensor failure modes

Failure mode	**Stochastic model**	**Note**
Nominal	$y_{nominal} \sim \mathcal{N}(y_{real}, \sigma_0^2)$	$y_{nominal} = y_{real} + v$
Outlier	$y_{outlier} \sim \mathcal{N}(y_{real}, \sigma_0^2)$	$\sigma_{outlier}^2 \gg \sigma_0^2$
Signal freeze	$y_{frozen} \sim \mathcal{N}(y_0, 0)$	$\sigma_{frozen}^2 = 0$
High variance	$y_{highvar} \sim \mathcal{N}(y_{real}, \sigma_{highvar}^2)$	$\sigma_{highvar}^2 \gg \sigma_0^2$
Sensor bias	$y_{bias} \sim \mathcal{N}(y_{real} + b, \sigma_0^2)$	$b = constant$
Sensor drift	$y_{drift} \sim \mathcal{N}(y_{real} + d(t), \sigma_0^2)$	$d(t) = varying$

Failure modes that only concern a change in variance can be detected without any estimation of y_{real}, making it possible to detect these failures using a simple signal processing module. Such failure modes include frozen signals, high variance, and outliers. A frozen signal can be detected by comparing the new measurement to the previous measurement. High variance and outliers can be detected by requiring the variance or measurements to lie within a defined range of validity, as defined in (2.17). Signal bias can be identified as a significant and abrupt change in signal mean. The detection of signal drift requires either an estimated value of y_{real} or redundancy in the measured state.

2.8 Failure Mode and Effect Analysis

Failure Mode and Effect Analysis (FMEA) is a qualitative reliability technique for analyzing and systematically identifying each failure mode and the effects they may have on the system [26]. To successfully develop an FMEA, it must include all the significant failure modes for each contributing element or part of the system. An FMEA can be performed both at system, subsystem or part level. The failure modes studied in this thesis are sensor faults, which do not affect the properties of the plant but creates substantial errors in sensor readings. Figure 2.8 shows the main steps of an FMEA, with an explanation of the terminology in Table 2.2.

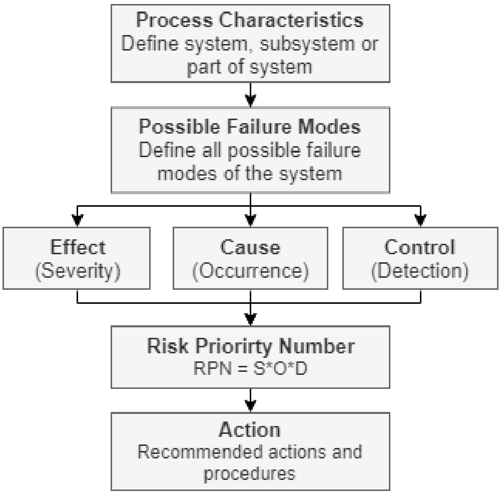


Figure 2.8: Steps of an FMEA

Table 2.2: FMEA terminology [27]

Definitions	
Process characteristics	The purpose and function of the defined system
Failure mode	Every possible failure mode for each contributing element in the system
Effect (Severity)	Which effects are most severe in case of each failure mode?
Cause (Occurrence)	What are the causes of each failure mode and what are the likelihoods of occurrence?
Control (Detection)	Comprises the method by which a failure is detected and isolated, and the time it takes. Often referred to as the effectiveness of the fault detection method
Risk priority number (RPN)	A measure for assessing risk and identifying critical failure modes associated with the system
Action	Recommended actions and procedures in case of failures

FMEA is a well-defined method for analyzing the redundancy design intent of marine control systems and defining the worst-case failure. Considering a DP system, the objective of an FMEA is to develop a fault-tolerant system that can not only handle adverse circumstances but also allow faults to be corrected as they occur, without jeopardizing the operation at hand. Following are some examples of potential consequences of failures in the control system of a marine vessel [27]:

- **Fault in heading measurements:** In heading control mode, this will cause the vessel to keep a wrong course. This will result in drift-off from the desired path that is increasing with time. In track control mode, there will be a permanent error between the desired and actual track.
- **Fault in turn rate measurements:** In heading control mode, this will result in a transient heading error, but will then be counteracted by the controller. Similar behavior will be seen in track mode.
- **Fault in measurements of the deviation from desired track:** This will not cause an effect in heading control but will create an offset equal to the size of the fault in tracking mode.
- **Fault in the track controller:** Causes the heading demand output from this controller to remain at the value it had when the fault occurred. This will cause the heading controller to steer the vessel away from the actual desired track, resulting in drive-off from the desired position or heading.
- **Loss of position and/or heading**: The vessel is no longer capable of maintaining its desired position, which in dynamic positioning mode is considered the most serious safety breach.

2.9 Hardware-in-the-loop Testing

While the FMEA focuses on physical layout and hardware, *Hardware-in-the-loop* (HIL) testing focuses on the software part of the control systems. HIL and FMEA are complementary activities, which both are needed for new-builds [30]. A HIL simulation set-up involves a real-time dynamic simulator that simulates all signals to and receives all command signals from the studied control system. For a DP system, this means that the HIL simulator will simulate the vessel dynamics, environmental forces, and thruster forces commanded by the control system. Using HIL testing in the offshore and maritime industry has a huge potential for increasing safety and reducing costs. It opens for both earlier, deeper and broader testing of the systems, which leads to faster commissioning time with less risk [26].

Some functionalities are not tested using regular HIL simulator technology. These functions may, however, be tested using full-scale testing with the control system operational. For these tests, a different type of HIL simulator known as an FMEA simulator is applied. The FMEA simulator provides the possibility to manipulate control signals flowing between the control system and the process plant. Both the plant and the control system operate as normal, but the FMEA simulator makes it possible to manipulate signals and simulate the effect of failures [31]. The concept of an FMEA simulator inspires the testing methodology in this thesis. When testing implemented algorithms, the input signals are polluted with faults to study the effects.

2.10 Signal Processing

Accurate control performance is highly dependent on the measurements received from the sensor system. Unreliable sensor measurements will have a negative influence on the control system's capability for safe and accurate performance. The measurements received from the sensors introduce a list of challenges such as noise, time delays, and failures. Another challenge is the presence of multiple signals, where synchronization, sensor drifting, and credibility must be considered. Each signal used in the control system should therefore be checked and processed for faults and bad quality in its own *signal processing unit*, as illustrated in Figure 2.9.

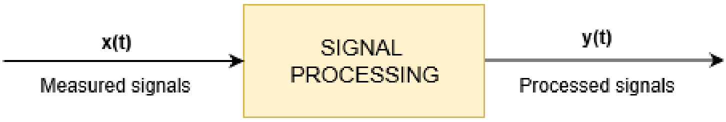


Figure 2.9: Signal processing module

The purpose of the signal processing module is to continuously monitor measurements of all external sensor signals, positioning reference signals, and thrust feedback signals. The module should perform a quality check of the signals. The quality check consists of checking both the integrity and continuity of the signal and fix format errors. It is also responsible for monitoring the signal for each of the faults described in Section 2.7. For simplicity, the signal processing tests can be divided into two main categories, namely *individual signal testing* and *multiple signal testing* [26]. This thesis will focus on individual signal testing.

2.10.1 Individual Signal Testing

Individual signal testing consists of several different tests done on each incoming signal. Different failure modes can be detected using different approaches, which are further explained below.

Signal Range Testing
Some signals have a defined range of validity. A gyrocompass measures the vessel's heading, whose output values should be within the interval 0-360°. If the signal processing unit receives a measurement outside this interval, it should immediately

classify the signal as faulty and discard it. This method is a simple way of rejecting invalid measurements. Signal range testing is accomplished by defining a maximum and minimum allowable value for the relevant measurements. The measurement should be discarded if it is located outside the allowable range. This interval can be expressed as:

$$x\,[k] \in [\,x_{min}, x_{max}\,]. \tag{2.22}$$

Variance Testing

The variance of a signal can mainly disclose two faults, namely temporal high variance or a frozen signal. A sudden peak in signal variance may indicate a sensor failure, inaccurate measurements, or an unknown disturbance acting on the relevant sensor. Zero variance may indicate a faulty sensor, transmission error, or loss of signal. Thus, both an upper and a lower boundary should be considered in a variance test.

Consider the sequence $\{x\,[k]\}$, at $t = k$ consisting of $n-1$ historical signal values in addition to itself. The sequence can be expressed as:

$$\{x[\,i\,] : i = k-(n-1), ..., k-1, k\}. \tag{2.23}$$

The average value $\bar{x}_k$ can be calculated by dividing the total sum of the sequence, by the number of entries n, as shown below.

$$\bar{x}_k = \frac{1}{n} \sum_{i=k-(n-1)}^{k} x\,[i]. \tag{2.24}$$

The corresponding variance of the signal sequence can be found by:

$$\sigma_k^2 = \frac{1}{n-1} \left(\sum_{i=k-(n-1)}^{k} x\,[i]^2 - n\bar{x}_k^2 \right). \tag{2.25}$$

The expression for signal variance can be rewritten on recursive form as:

$$\sigma_{k+1}^2 = \frac{n}{n-1}(y\,[k+1] - (\bar{x}\,[k+1])^2), \tag{2.26}$$

where *y[k]* is defined as:

$$y\,[k] = \frac{1}{n} \sum_{i=k+1-(n-1)}^{k} x\,[i]^2, \tag{2.27}$$

and

$$y\,[k+1] = y\,[k] + \frac{1}{n}((x\,[k+1]^2 - (x[k-(n-1)])^2). \tag{2.28}$$

Other statistical methods may also be used for variance testing, such as the interquartile range method, which will be discussed later in the thesis.

Outlier Testing
An outlier, as described in Section 2.7, is a signal value that is located outside a band of defined width around the estimated signal means. In other words, it is a signal that deviates significantly from previous measurements. If an outlier is detected, the signal value should be rejected for one sample. The outlier is then replaced with a calculated value, such as the previous signal value or the mean value. Outlier testing is conducted by defining an interval around the mean where the values are acceptable. This interval can be expressed as:

$$x\,[k] \in [\,\bar{x}_k - a\sigma, \bar{x}_k + a\sigma], \tag{2.29}$$

where a usually lies in the interval $a \in [3, 9]$. This method is similar to the signal range test but uses the statistics of the signal to expand or constrict the thresholds.

2.10.2 Multiple Signal Testing

For many operations, redundancy is both important and imposed by the authorities. This means that a system is required to have two or more sensors measuring the same state. The signal processing unit must therefore be able to handle multiple signals for each state. In this context, it must be able to determine which signal that should be trusted the most and detect if a sensor fails or suddenly shows a large deviation compared to the other measurements.

Weighting
For configurations with two or more sensors available, the signal processing module can perform a weighting of the individual signals in order to calculate the optimal average signal. The weighting factors for each signal can either be set manually or be calculated automatically based on the variance of the signals.

Considering a system with n independent signals. The estimated weighted signal can be calculated as:

$$x_\omega = \sum_{i=1}^{n} s_i x_i, \quad \sum_{i=1}^{n} = 1, \tag{2.30}$$

where each value s_i is calculated based on the either manually or automatically set weighting factors as:

$$s_i = \frac{w_i}{\sum_{i=1}^{n} w_k}. \tag{2.31}$$

Assuming a system with three available sensor measurements for the state x, namely x_1, x_2 and x_3. Using the manual method, the weighting factors w_1, w_2 and w_3 are set by the operator, and the final weighted signal x_w becomes:

$$x_w = \frac{w_1 x_1 + w_2 x_2 + w_3 x_3}{w_1 + w_2 + w_3}. \tag{2.32}$$

Using automatic weighting, the weights are calculated based on the principle of minimum variance. Considering a similar situation as above, but with only two sensor measurements for the state, the weights can be calculated as:

$$s_1 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}, \quad s_2 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}. \tag{2.33}$$

A general automatic weighting function for n signals can be written as:

$$s_i = \frac{\prod_{j \neq i} \sigma_j^2}{\sum_n^{k=i} \prod_{j \neq k} \sigma_j^2}. \tag{2.34}$$

In the automatic case, a signal with a high variance relative to the other signals will accordingly have a low weighting factor. A frozen signal with zero variance will be discarded from the weighting.

Handling Loss of Signals

The signal processing unit should be able to handle both the enabling and disabling of sensors. Abrupt disabling or enabling of a sensor can result in an oscillating response. Thus, the signal average should be filtered for a specific period after the loss or gain of a sensor signal. Let y_w be the newly weighted signal after a failure situation. The filtered and weighted signal y_{fw} can be found using the low-pass equation given as:

$$\dot{y}_{fw} = -\frac{1}{T_f} y_{fw} + \frac{1}{T_f} y_w. \tag{2.35}$$

The filter should be activated subject to an event that indicates the loss of a sensor signal. The time T_f depends on the signal difference and a maximum allowable rate of change. When enabling a sensor, it will remain smooth with no filtering of the signal. Figure 2.10 shows an example where two sensor signals are weighted to an optimal average until one of the sensors is disabled. The filter is then activated, ensuring a smooth transition.

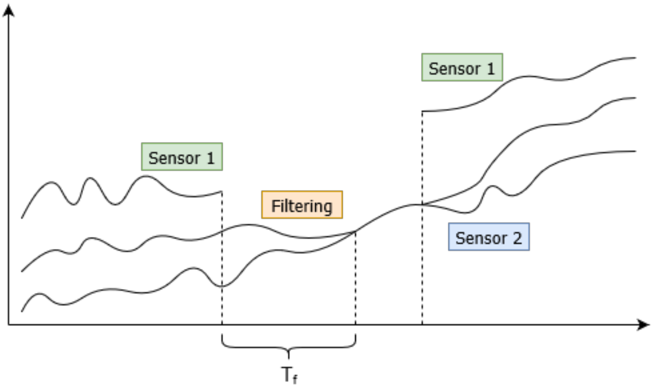


Figure 2.10: Disabling and enabling of sensor signals

Handling Drift of Signals
Signal drifting is the phenomenon where one of several signals measuring the same state suddenly starts to deviate from the other, as described in Section 2.7. A control system may use the average of several signals measuring the same state as the true value. The increasing difference indicates that one of the sensor signals is drifting, providing a faulty measurement to the system. However, it gives no information about which signal the system should discard.

One method to detect drifting is called voting. If two or more sensors or positioning reference systems are available, the voting method can detect drifting between the different sources. In the case where three or more sensors are available, it can also determine which of the sensors that are drifting by rejecting the one sensor with the highest deviation relative to the other two.

2.11 Approaches to Fault Detection and Diagnosis

Technological systems are highly vulnerable to faults. Faults can occur in the system's actuators, sensor readings, and control systems. One single fault in a highly automated system with complex interactions between different components may cause the failure of the whole system. As a result, fault detection and diagnosis systems are becoming increasingly important. Fault detection and diagnosis systems implement the following tasks [27]:

1. **Fault detection:** The identification that something is wrong in the monitored system.
2. **Fault isolation:** The determination of the exact location or component of which the fault has occurred.
3. **Fault identification:** The determination of the magnitude of the fault.

Together, the isolation and identification tasks are referred to as the fault diagnosis task. While fault detection and fault isolation are equally important and a must in all practical systems, the fault identification task is often omitted. Therefore, most systems only contain the fault detection and isolation tasks and are commonly referred to as FDI (fault detection and isolation) systems. The tasks of fault detection and diagnosis systems are illustrated in Figure 2.11.

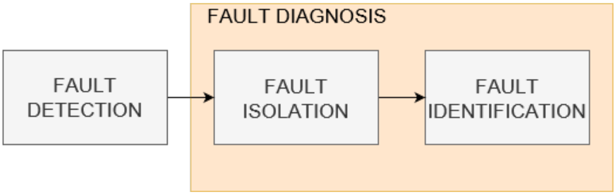


Figure 2.11: Tasks of fault detection and diagnosis. Based on Figure 1.1 in [28].

The detection performance of the system is characterized by several quantifiable values, including:

- Fault sensitivity: The technique's ability to detect faults of reasonable small size
- Reaction speed: The technique's ability to detect faults with a reasonable short time delay after their occurrence
- Robustness: The technique's ability to operate in the presence of disturbances, noise and modeling errors without triggering a large number of false alarms

The approaches of fault detection and diagnosis may be classified into two main groups [28]. The model-free methods are those who do not utilize a mathematical model of the studied plant, whereas the model-based methods use a mathematical model. Both main groups are briefly explained below.

2.11.1 Model-free Methods

Model-free fault detection and diagnosis methods do not utilize mathematical models of the monitored plant. There exist several different approaches within this group.

Physical redundancy
This approach uses multiple sensors to measure the same physical value. Any significant discrepancies between the different measurements indicate a sensor fault. Fault isolation is only possible with three sensors measuring the same value since two divergent sensor measurements do not contain any information about which sensor that contains an error. This approach requires extra hardware, including both extra costs and weight.

Special sensors
Special sensors may be installed purely for detection and diagnosis. These sensors may include limit sensors, performing limit checking, or other sensors that may measure some physical values indicating faults, such as sound and vibration.

Limit checking
This approach compares plant measurements to pre-set limits. If the measurements exceed the threshold, a fault is indicated. Many systems include two levels of limits. The first level serves as a pre-warning, while the second level triggers an emergency alarm. This approach has two significant drawbacks. First, since the plant variables may vary widely as a result of normal input variations, the test thresholds must be set quite conservatively. Second, the effect of a single component fault can propagate to several plant variables, setting off a large number of alarms, which makes isolation difficult.

Spectrum analysis
Spectrum analysis of the plant measurements may be used for fault detection and

diagnosis. Plant variables usually exhibit a typical frequency spectrum under normal conditions. Any deviations from this spectrum can indicate a fault. Some faults may even be possible to isolate using the characteristics of the spectrum.

Logic reasoning
These methods are aimed at evaluating the symptoms obtained by detection techniques. The simplest methods consist of logical rules, i.e., IF, AND, and THEN statements, to conclude. Each conclusion can further serve as a symptom in the next if-statement until a final conclusion is reached.

2.11.2 Model-based Methods

In contrast to the model-free methods, model-based fault detection and diagnosis methods require access to a mathematical model of the studied plant. The monitored plants are dynamical systems described by a continuous-time operation. Most model-based fault detection and diagnosis approaches utilize the concept of *analytical redundancy*. In contrast to physical redundancy where multiple sensor measurements are compared to each other, this method is based on comparing sensor measurements to analytically computed values to determine the difference. Mathematically, this can be written as:

$$r(t) = y(t) - \hat{y}(t), \tag{2.36}$$

where *r(t)* is called a *residual*. In the case of no faults, the residual should be close to zero or vanish entirely. A non-vanishing residual would indicate the existence of a fault in the system. Figure 2.12 shows a scheme where the model is used to determine an analytically estimated value $\hat{y}$ for every input u. The system is exposed to disturbances and faults denoted as d and *f*, respectively. The residuals are calculated at each sample time as the difference between the actual measured value, y, and the analytically computed value, $\hat{y}$.

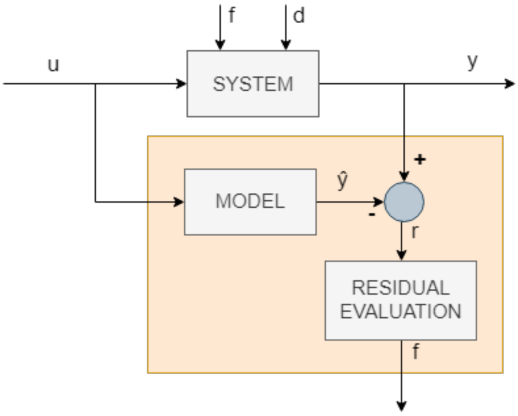


Figure 2.12: Diagnosis of continuous systems. Based on Figure 1.10 in [27].

Following this method, the residuals are indications of the presence or absence of faults in the system. The diagnosis algorithms for continuous-variable systems can generally be divided into two steps, represented in Figure 2.13.

1. **Residual generation:** Sensor measurements and analytical values are compared to determine the residuals to be evaluated in the next step.
2. **Residual evaluation:** The calculated residuals are evaluated in order to arrive at detection and isolation decisions. The presence of noise, disturbances, and modeling errors causes the residuals to never be identically equal to zero, even with no faults. Therefore, the evaluation should test the residuals against different thresholds to unveil faults.

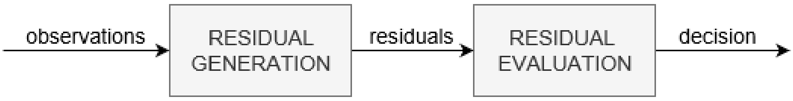


Figure 2.13: Stages of model-based fault detection and diagnosis. Based on Figure 1.2 in [28]

Residual generation is commonly done by applying either a Kalman filter, diagnostic observers, parity relations or parameter estimation, which can be read more about in [28].

2.11.3 On-line and Off-line Methods

The on-line fault detection problem is to detect the occurrence of a fault as soon as possible while avoiding a large number of false alarms. On-line algorithms run concurrently with the process they are monitoring, processing each data point as it becomes available, with the real-time constraint that processing should be completed before the next data point arrives.

Off-line algorithms consider the entire data set at once, and there is no real-time constraint on the run time. The advantage of this approach is that the complete set of observations is available when the analysis is performed. This advantage makes the off-line approach very convenient for method development, as it puts no constraints on time or what data sets that can be used. The drawback is that it cannot be used to prevent faults during operations.

2.12 Functional Barriers

Most systems have protection equipment or other features to protect people and assets against harm in case failures or dangerous deviations occur in the system. The equipment or features that are installed for this purpose are called *safety barriers* or *barriers* [32]. Barriers can be described as both physical and nonphysical multi-layered structures with the purpose of preventing unwanted events or accidents [33]. This is highly relevant in control systems to prevent faults such as loss of position, where signal processing and fault detection can be relevant barriers.

2.12.1 Barrier Properties

The reliability of a barrier should be evaluated based on several criteria, as presented in [34]. The criteria can briefly be summarized as:

- *Specificity:* The barrier should be able to detect and prevent the consequences of a *specified* dangerous event.
- *Adequacy:* The adequacy of a barrier can be based on its capacity and ability to prevent accidents within the design basis and meet the requirements of relevant standards and norms.
- *Independence:* A barrier should be independent of all other barriers related to the specified hazardous event. This means that the performance should not be affected by the failure of another barrier or the conditions causing another barrier to fail.
- *Dependability:* The barrier should reduce the identified risk by a known and specified amount. That is, the barrier must be dependable in the sense that it can be counted on to do what it is supposed to do.
- *Robustness:* The barrier should be able to handle extreme events and should not be disabled if another barrier is enabled.
- *Auditability:* The barrier should be designed to allow regular periodic validation of the function.

2.12.2 Barrier Classification

Barriers can be classified in several different ways, two of which are introduced briefly in the following as done in [32].

Proactive and Reactive Barriers
It can be distinguished between proactive and reactive barriers:

- Proactive barrier: A barrier that is installed to prevent or reduce the probability of unwanted events or accidents. Often called a *frequency-reducing* barrier. Examples of proactive barriers are anti-lock braking systems (ABS) and electronic stability control (ESP) systems.
- Reactive barrier: A barrier that is installed to prevent or reduce the consequences in case of an unwanted event or accident. Often called a *consequence-*

reducing barrier. Examples of reactive barriers are seat belts, airbag systems, and alarming systems.

Active and Passive Barriers
Barriers may also be divided into active or passive:

- Active barrier: A barrier that is dependent on actions from an operator, control system, and/or energy systems to perform its function. Examples of active barriers are fire alarm systems and signal processing systems.
- Passive barrier: A barrier that is integrated into the design of the workplace and does not require any human actions, energy sources, or information sources to perform its function. Examples of passive barriers are firewalls and mid-road barriers.

This theory can be linked to marine control systems by considering an example of functional barriers to loss of position. Figure 2.14 shows an example of the multi-layered structure that makes up the functional barriers to loss of position. These are barriers that are dedicated to preventing the undesired event of which the vehicle loses its position. This is done by employing redundancy and signal processing in several steps.

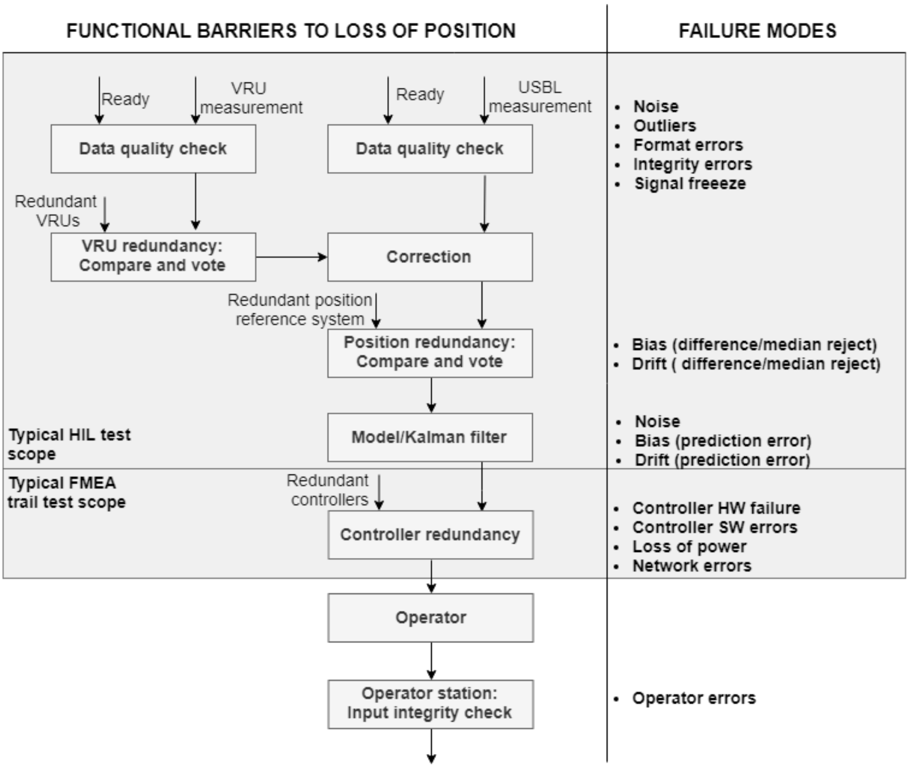


Figure 2.14: Functional barriers to loss of position. Based on [35].

Chapter 3

Tools for Development of Fault Detection Methods

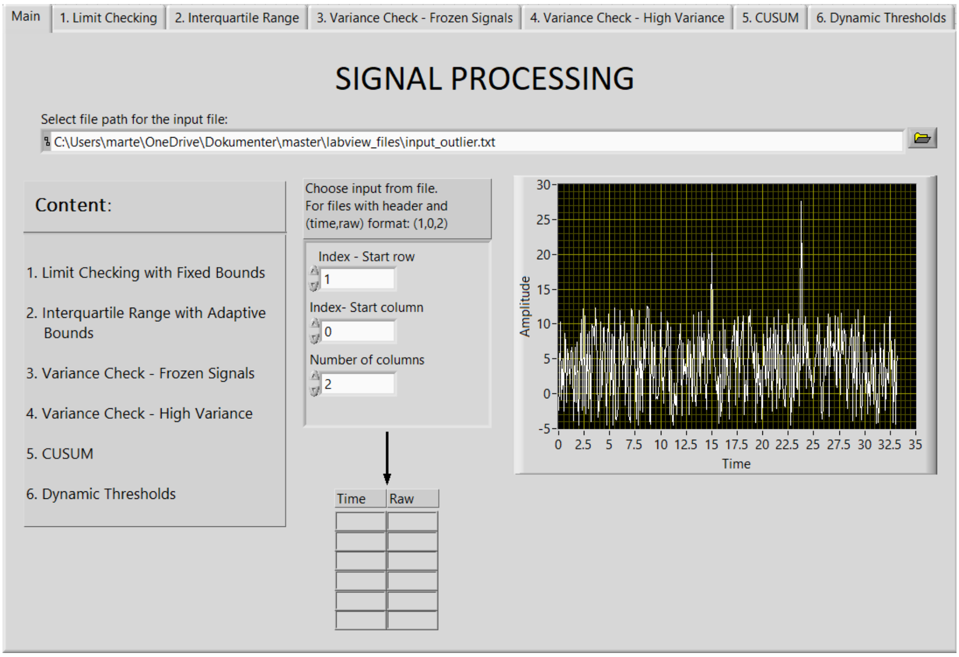
This chapter studies existing tools and methods for fault detection and implements them in a joint interface using the program LabVIEW. The theories discussed are mainly based on [36], [37] and [27], with some modifications. The implementation of the algorithms has a focus on usability and a simple interface. For this reason, a picture of the front panel for each tested algorithm is attached in Appendix C along with its respective implementation description. All the chosen methods are model-free and use the statistics or behavior of the data series to detect faults. To be able to test the algorithms, a signal generation module is created and described. Finally, each of the implemented algorithms is tested and validated using generated test signals.

3.1 LabVIEW

The signal processing module developed in this thesis is created in the program Laboratory Virtual Instrument Engineering Workbench (LabVIEW) [38]. LabVIEW is a development environment developed by National Instruments. The program uses a graphical programming approach that helps visualize the code during development.

The program has two prominent advantages for this task. The first advantage is that MTS' ROV simulator is developed using the same program, making it easy to integrate the module into the control system for further work. The second advantage is the possibility to create a user-friendly interface, called the front panel. This comes of high importance, as the point of the module is to be able to notify the system operator the moment a fault occurs. The front panel gives the user full overview and control of the executing program.

Figure 3.1 shows the main page of the final fault detection program. This panel is an example of how an interface might look. The interface can show any values relevant to the user and can contain plots showing the input signal and signal properties. The interfaces let the user choose and adjust variables that are relevant to the method and study the results consecutively.



SIGNAL PROCESSING

Select file path for the input file:
C:\Users\marte\OneDrive\Dokumenter\master\labview_files\input_outlier.txt

Content:

1. Limit Checking with Fixed Bounds
2. Interquartile Range with Adaptive Bounds
3. Variance Check - Frozen Signals
4. Variance Check - High Variance
5. CUSUM
6. Dynamic Thresholds

Choose input from file.
For files with header and
(time,raw) format: (1,0,2)

Index - Start row: 1
Index- Start column: 0
Number of columns: 2

Figure 3.1: Example of a front panel

3.2 Limit Checking - Fixed Bounds

Limit checking is a simple algorithm based on setting an upper and lower bound for each measurement. If the measured value is within these limits, the measurement is considered valid. If the value is outside this interval, it should be rejected, and an alarm should be triggered. This method requires prior knowledge about the signals in order to determine the limits. In practice, this method is useful for measurements where the range is clearly defined, such as for compasses or vehicle velocity.

3.3 Interquartile Range - Adaptive Bounds

The method of limit checking using fixed bounds requires some prior information about the signals to be studied. To be able to determine the upper and lower limits, it must be possible to establish an acceptable interval where the measurements should be found at each instant of time. It also requires precisely selected limits to avoid false alarms or undetected signal faults. In practice, this is not always possible.

In cases where no such limits can be predefined, adaptive bounds must be applied. Adaptive bounds can be set by utilizing the method of *interquartile range* (IQR). This method uses the distribution of the given data set to calculate the interquartile range, which is further used to set an upper and lower threshold and detect outliers. This method is based on the assumption that the measurements are randomly distributed samples, which can be represented by a normal distribution.

Figure 3.2 shows an illustration of how the complete distribution is divided into quartiles and how the limits are set. Table 3.1 contains names and definitions explaining the symbols in the figure.

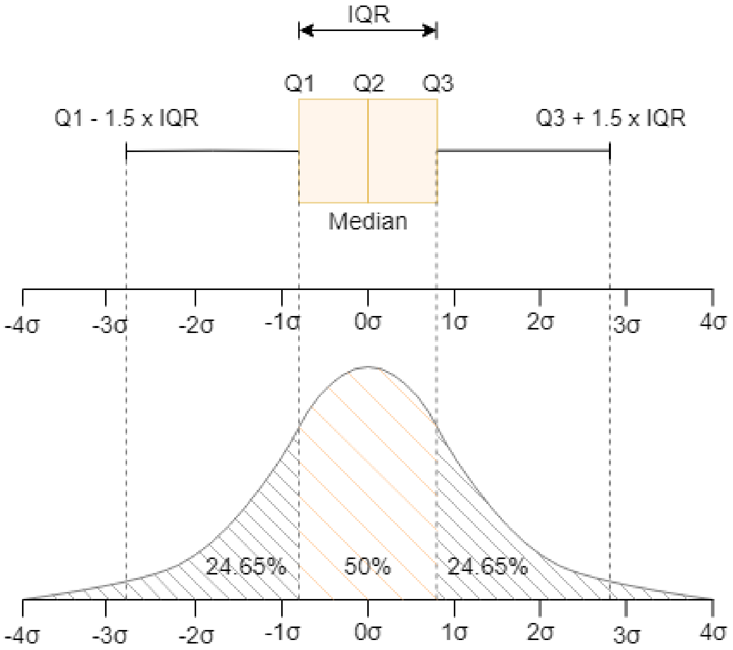


Figure 3.2: Interquartile range. Based on Figure 6.13 in [29].

Table 3.1: Interquartile range - Definitions

Symbol	Name	Definition
Q1	First quartile	Splits off the lowest 25% from the highest 75%
Q2	Median	Splits the data set in half
Q3	Third quartile	Splits off the highest 25% from the lowest 75%

There are mainly three different methods for computing the quartile values. These methods are presented in Table 3.2, both for an even and odd number of data points.

Table 3.2: Methods for calculating the quartile values

Method 1:

- ○ Use the median to divide the data set into two equal halves:
  - If an odd number of data points: **do not include** the median in either half
  - If an even number of data points: split the data set in two equal halves
- ○ The value for the first quartile is the median of the lower half of the data
- ○ The value for the third quartile is the median of the upper half of the data

Method 2:

- ○ Use the median to divide the data set into two equal halves:
  - If an odd number of data points: **include** the median in both halves
  - If an even number of data points: split the data set in two equal halves
- ○ The value for the first quartile is the median of the lower half of the data
- ○ The value for the third quartile is the median of the upper half of the data

Method 3:

- ○ If there are an even number of data points, Method 3 is equal to Method 1 & 2
  - If there are (4n+1) data points:
    - $\mathrm{Q1} = \mathrm{X}_n \cdot 0.25 + \mathrm{X}_{n+1} \cdot 0.75$
    - $\mathrm{Q3} = \mathrm{X}_n \cdot 0.25 + \mathrm{X}_{n+1} \cdot 0.75$
  - If there are (4n+3) data points:
    - $\mathrm{Q1} = \mathrm{X}_{n+1} \cdot 0.75 + \mathrm{X}_{n+2} \cdot 0.25$
    - $\mathrm{Q3} = \mathrm{X}_{3n+2} \cdot 0.25 + \mathrm{X}_{3n+3} \cdot 0.75$

where X_n is read as the nth data point in the data set.

When the quartiles are calculated, they can be used to determine the upper and lower bounds as follows:

$$\begin{aligned} \text{IQR} &= \text{Q3} - \text{Q1} \\ \text{Lower bound} &= \text{Q1} - 1.5 \cdot \text{IQR} \\ \text{Upper bound} &= \text{Q3} + 1.5 \cdot \text{IQR} \end{aligned} \tag{3.1}$$

3.3.1 Moving Window

The IQR algorithm uses the statistics of the data set to determine the quartiles, which are further used to determine the upper and lower bounds. The strength of this method is that the bounds can adapt to the signal without any prior knowledge about the system. However, an obvious weakness with this method is that the limits are not adaptive with regards to time changes in the signal. By using the whole data set to calculate the statistics of a drifting signal, some outliers may not be detected. It also means that a faulty signal early in the data set will ruin the credibility of the entire test if this value is included in the calculations. This can easily be fixed by implementing a moving window or a loss function.

One strategy for making the algorithm time adaptive is by introducing a moving window. This means that only a fixed number of samples are used to calculate the quartiles that determine the boundaries. By implementing this, a detected outlier will only influence n number of samples, where n is the window size.

Considering a data set S with N samples and a window size n, the moving window method is described in Figure 3.3. For each iteration, the first sample in the current set S_i is deleted, and a new sample is added.

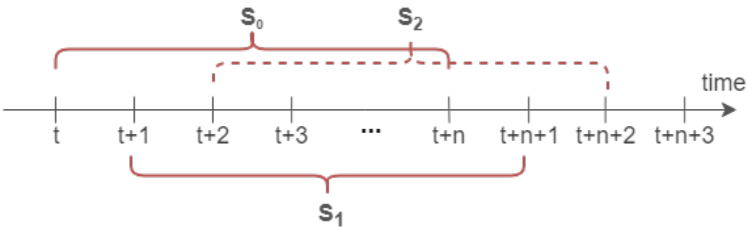


Figure 3.3: Moving window with window size n

This method is implemented with the IQR algorithm with adjustable window size and will be used for further testing.

3.3.2 Loss Function

A second way of making the algorithm time adaptive is presented in [36]. In general, this approach is based on forgetting old measurements by introducing a loss function. This function can be chosen as any desired function that fulfills the purpose of fading out old measurements. An example of a loss function is the exponential function:

$$L = e^{-\lambda t}, \tag{3.2}$$

where t is the time and λ is a forgetting factor that determines the decay rate of the loss function with time. Figure 3.4 shows four similar loss functions, each with different forgetting factors.

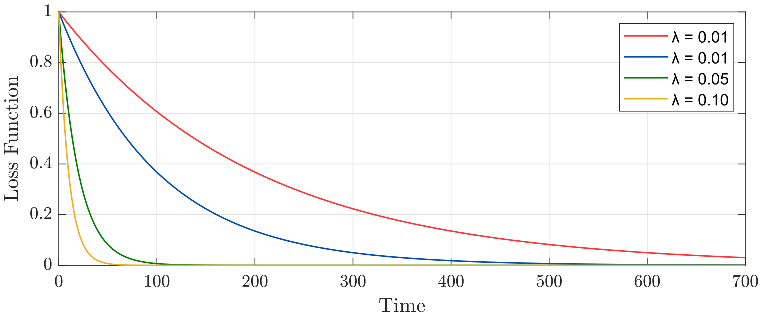


Figure 3.4: Loss functions with different forgetting factors

3.4 Variance Check

The variance of a signal can contain important information about the measurements. There are especially two interesting cases, namely a zero variance and a sudden peak in variance. These two cases can indicate a frozen signal, maybe from a faulty sensor, or a disturbance causing increased noise to the system. Changes in the variance are simple to detect if the algorithm is able to calculate the variance of the signal continuously.

The variance of the signal can be calculated by using the formula:

$$\sigma^2 = \frac{(\sum_{i=1}^{\mathrm{n}} x_i - \bar{x})^2}{\mathrm{n}-1}. \tag{3.3}$$

where n is the sample length, x_i is the current signal value, and $\bar{x}$ is the current mean. The sample length determines how many samples of the data set that will be considered in the calculations. By using a variable sample size for the mean and variance, the algorithm is made more time adaptive. This follows the same logic as a moving window: The signal mean and variance is calculated based on n number of samples. By choosing a sample size that is equal to the total number of samples, each value is included in the calculation of the variance. The resulting variance is constant for all values of the data set. By decreasing the sample size, the variance becomes more accurate for each studied value but exhibits more fluctuations. The sample size should be balanced such that a sudden change in variance will be detected relatively fast but should include enough data to avoid false alarms.

3.4.1 Frozen Signals

As described in Section 2.7, a frozen signal is defined as a fault where the variance is equal to zero. A frozen signal can indicate a lost connection or the loss of a sensor and may lead to the termination of the current operation.

The variance check intended to detect frozen signals is implemented with two detection methods. One method is to check whether the variance of any point is equal to zero. The second method checks if a signal value is exactly equal to its previous value for a significant number of consecutive measurements. The second method is included to account for those cases where the sample size is set such that even though the variance at a point is zero, it is not detected. This case is shown in Figure 3.5, where the variance using two different sample sizes are compared. When the sample size is set to 50, the algorithm never detects that the variance is zero at several points during the test. To account for this, the algorithm also continuously checks if the current sample is exactly equal to the previous sample, and if this occurs for a significant number of data points in a row.

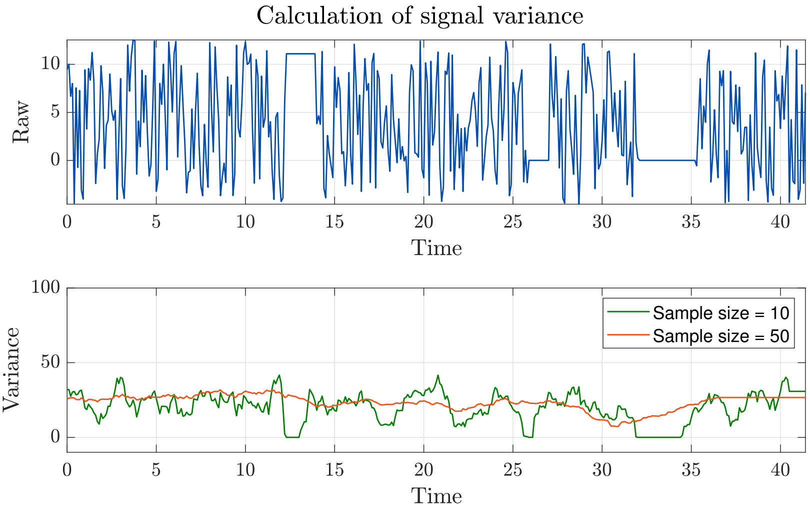


Figure 3.5: Variance using sample sizes of 10 and 50

3.4.2 High Variance

High variance is defined as a sudden increase in the variance of the signal, where $\sigma^2_{highvar} \gg \sigma^2_0$. This may indicate a temporary disturbance acting on the sensor, creating additional noise. To detect high variance, the method of interquartile range can be applied. This method is described in Section 3.3, where it is used to determine the upper and lower allowable limit for a measurement. By setting the variance as input to the algorithm, an upper bound can be set for the variance as well, while the lower bound can be set to zero.

3.5 Cumulative Sum

The cumulative sum (CUSUM) algorithm is a sequential analysis technique first developed by E. S. Page [39] but is here derived as in [27] and implemented as in [36]. The algorithm is used to detect known changes by hypothesis testing between a fault-free condition H^0 and a condition with faults H^1. The CUSUM test compares two probability density functions (PDFs) and their means to determine which of the PDFs the random variable is most likely to belong to.

Consider a sequence of independent random variables $z(i)$, where i denotes the time instant, and with a PDF, $p_\theta(z)$, that depends upon only one scalar parameter θ. Before an unknown time change, $\theta = \theta_0$. After the time change, it is equal to $\theta = \theta_1 \neq \theta_0$. The goal is to detect and estimate the change of this parameter. It is assumed no prior knowledge of the distribution of the time change. θ_0 is known by hypothesis, while θ_1 can be either known or unknown. The first case yields the CUSUM algorithm, while the second the generalized likelihood ratio (GLR) algorithm.

A fundamental concept for both algorithms is the log-likelihood ratio, which given the random variable z is defined as:

$$s(z) = \ln \frac{p_{\theta_1}(z)}{p_{\theta_0}(z)}. \tag{3.4}$$

The name comes from the fact that the likelihood function of an observation z is by definition equal to the PDF, $p_\theta(z)$. The key statistical property of this ratio is that:

$$E_{\theta_0}(s) = \int_{-\infty}^{\infty} s(z) p_{\theta_0}(z) dz < 0, \tag{3.5}$$

and

$$E_{\theta_1}(s) = \int_{-\infty}^{\infty} s(z) p_{\theta_1}(z) dz > 0, \tag{3.6}$$

where E_{θ_0} and E_{θ_1} denote the expectations of the random variables under the two distributions $p_{\theta_0}(z)$ and $p_{\theta_1}(z)$, respectively. This means that a change in the parameter θ is reflected as a change in the sign of the mean value of the log-likelihood ratio.

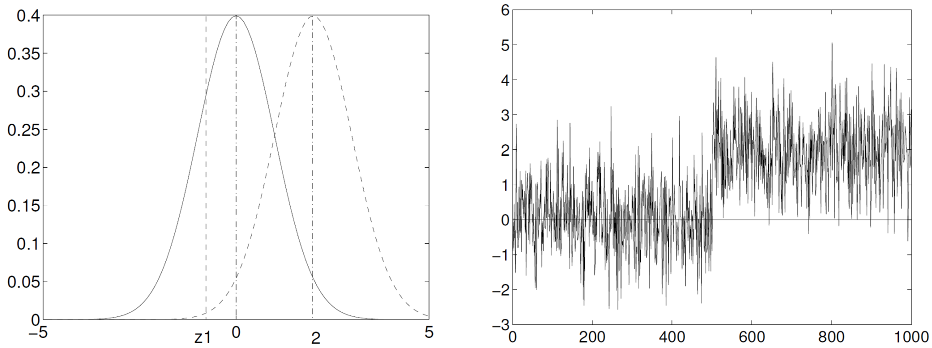
Considering the cumulative sum:

$$S(k) = \sum_{i=1}^{k} s(z(i)) = \sum_{i=1}^{k} \ln \frac{p_{\theta_1}(z(i))}{p_{\theta_0}(z(i))}, \tag{3.7}$$

where k denotes the present time instant. *S(k)* is expected to exhibit a negative drift before change, and a positive drift after change.

This theory can easily be understood by considering the following example. Assume that $p_{\theta(z)}$ is a Gaussian distribution, where the parameter θ represents the mean. Figure 3.6a shows two distributions of the random variable z. The leftmost distribution, $p_{\mu_0(z)}$ has a mean of $\mu_0 = 0$ and a variance of $\sigma^2 = 1$. The distribution to the right, $p_{\mu_1(z)}$, has a mean of $\mu_1 = 2$ and the same variance as $p_{\mu_0(z)}$. Consider the realization z_1. As z_1 is most probably obtained when the random variable z has $p_{\mu_0(z)}$ as its PDF, $\frac{p_{\mu_1}(z_1)}{p_{\mu_0}(z_1)} < 1$. This illustrates that the log-likelihood ratio is on the average negative when z has $p_{\mu_0(z)}$ as its PDF and positive when it has $p_{\mu_1(z)}$ as its PDF.

Figure 3.6b shows a realization of a sequence of independent random variables with distribution $p_{\mu_0(z)}$ before the time $k = 500$ and distribution $p_{\mu_1(z)}$ after. Given this, the cumulative sum $S(k)$ is expected to exhibit a negative drift before change, and positive drift after change. This behavior is shown in Figure 3.7.



(a) Two Gaussian probability density functions with equal variance $\sigma^2 = 1$ and means of $\mu_0 = 0$ and $\mu_1 = 2$. Taken from Figure 6.7 in [27].

(b) Sequence of independent random variables with distributions as shown in Figure 3.6a. The time on the x-axis is expressed in number of samples. Taken from Figure 6.8 in [27].

Figure 3.6: Gaussian distributed sequence of random variables

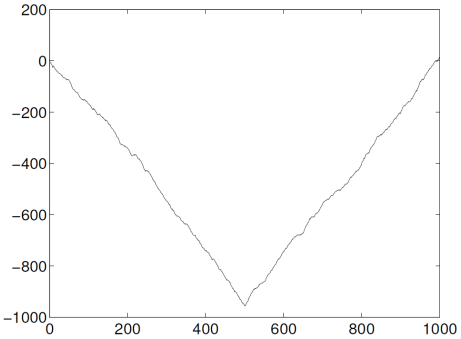


Figure 3.7: Change of *S(k)* for the sequence shown in Figure 3.6b. Time is expressed in the number of samples. Taken from Figure 6.9 in [27].

To study the change, the relevant information lies in the difference between the value of the log-likelihood ratio and its current minimum value. The corresponding decision rule at each time instant is:

$$g(k) = S(k) - m(k) \geq h, \tag{3.8}$$

where h is e user defined positive threshold and:

$$m(k) = \min_{1 \leq j \leq k} S(j). \tag{3.9}$$

The detection rule is a comparison between the cumulative sum $S(k)$ and the adaptive threshold $m(k) + h$. The value $m(k)$, called the drift term, makes sure the threshold is modified on-line and keeps full memory of the information contained in past observations.

The Least Squares Cumulative Sum Filter

To detect sudden biases in the signal mean, a least squares cumulative sum (LS CUSUM) filter can be applied. The basic assumption for this method is that the measurements consist of the deterministic component θ_t and an additive white noise ε_t, such that the measurement can be written as:

$$y_t = \theta_t + \varepsilon_t, \tag{3.10}$$

where y_t denotes the measurement. The parameter θ_t is estimated from y_t and is used to calculate the residual ε_t. This residual is further used to calculate a distance measure, s_t. The distance measure is averaged to get g_t, which is further used as input to the stopping rule. The different steps with their inputs and outputs are illustrated in Figure 3.8.

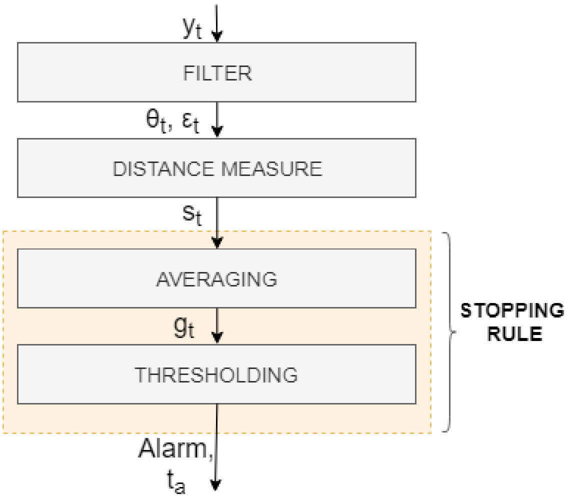


Figure 3.8: Fault detection using a LS CUSUM filter. Based on Figure 3.1 in [36].

The distance measure can be calculated using different approaches:

- Use the residuals directly:

$$s_t = \varepsilon_t = y_t - \hat{\theta}_{t-1}, \tag{3.11}$$

  where $\hat{\theta}_{t-1}$ is the estimate of the signal based on measurements up to time $t-1$. This approach is suitable for the change in the mean problem.

- Use the square of the residuals:

$$s_t = \varepsilon_t^2. \tag{3.12}$$

  This approach is useful for detection of both variance and parameter changes.

A *stopping rule* is used for giving an alarm when θ_t has exceeded a given threshold, and a fault is detected. To make the alarm decisions, an additional test statistic g_t is introduced. The test statistic g_t sums up its input s_t and should give an alarm if it exceeds the predefined threshold h. Using white noise input, g_t tends to drift similar to a random walk, which may result in false alarms or detection delay. Two additional mechanisms can be added to the algorithm to prevent this. The drift term ν is subtracted at each time instant to prevent positive drifting. To prevent negative drifting, g_t is set to zero each time it becomes less than zero.

The equations below summarize the two-sided LS CUSUM filter, which assumes θ_t can be both positive and negative. Here, an adaptive filter is combined with the CUSUM test as the change detector. The signal is assumed to be piecewise constant. When an alarm is triggered, the algorithm is restarted by setting $g_t^{(1)} =$

$g_t^{(2)} = 0$.

$$\hat{\theta}_t = \frac{1}{t-t_0} \sum_{k=t_0+1}^{t} y_k \tag{3.13a}$$

$$\varepsilon_t = y_t - \hat{\theta}_{t-1} \tag{3.13b}$$

$$s_t^{(1)} = \varepsilon_t \tag{3.13c}$$

$$s_t^{(2)} = -\varepsilon_t \tag{3.13d}$$

$$g_t^{(1)} = \max(g_{t-1}^{(1)} + s_t^{(1)} - \nu, 0) \tag{3.13e}$$

$$g_t^{(2)} = \max(g_{t-1}^{(2)} + s_t^{(2)} - \nu, 0) \tag{3.13f}$$

Tuning of the LS CUSUM Filter Algorithm
The CUSUM filter requires that the variables h and ν are set before running the algorithm. The tuning of the filter can be done by following some simple steps:

1. Start with a high value for the threshold h and choose ν as half of the expected change.
2. ν can be adjusted such that g_t is equal to zero more than 50% of the time.
3. Adjust h such that the number of false alarms or delay for detection is satisfactory.
4. For faster detection, ν can be decreased, while h can be increased to reduce the number of false alarms.

3.6 Testing and Validation

The studied methods for fault detection must be tested and validated. This means that after implementation, which is described in Appendix C, they should be tested to see that they fulfill their intended purpose and give the correct solutions. Each algorithm is tested by running them on a valid test signal. If a fault is detected, a simulated alarm is triggered. The methodology for conducting these tests are inspired by FMEA simulation. The algorithms are tested on signals that have been modified to contain different failure modes. This is done by dedicating a separate module to create polluted signals.

3.6.1 Signal Generation

The first step of validating the fault detection algorithms is to create a suitable test signal. In this project, this is done by constructing a separate *signal generation module*. The implementation of this model in LabVIEW is described in Appendix B. This module is responsible for creating the test signals for the algorithms and should be able to run both in real-time and saved to file. By creating the test signals, the number of test cases is not limited by the signal source. A simulated signal can contain all kinds of both likely and unlikely failure modes.

The test methodology is shown in Figure 3.9, inspired by the concept of FMEA simulation [31]. A clean signal is used as a basis but is manipulated to contain faults before entering the signal processing module. The objective of this manipulation is to simulate the effects different failures have on the system.

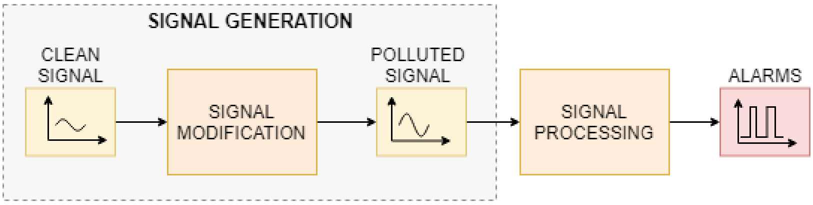


Figure 3.9: Stages of simulation

The signal generation module is created by using a uniform white noise with an adjustable noise amplitude and offset as a basis. This is chosen to represent a random signal subjected to noise. Section 2.7 defined and studied the mathematical modeling of different failure modes. These models are implemented in the signal generator, such that each failure can be added to the clean signal. While the program generates the signal, it simultaneously writes all data to an output file. This file contains both the raw random signal and the polluted signal together with its frequency and calculated mean and variance. An example of an output file is included in Appendix B.3.

3.6.2 Limit Checking

The limit checking algorithm is implemented such that the upper and lower bounds must be set manually based on prior knowledge of the data set. The algorithm goes through each of the data points and compare them to the predefined upper and lower bound. If the current value is outside the allowed interval, it is set as the maximum or minimum allowable value, and an alarm is triggered.

Figure 3.10 shows the result of the test. The limits were set by adjusting them until a satisfying number of alarms were triggered. The alarm is simulated by a Boolean variable, where one means that the alarm is triggered. The results show that the algorithm behaves as expected and is able to detect and correct values outside the predefined range of validity.

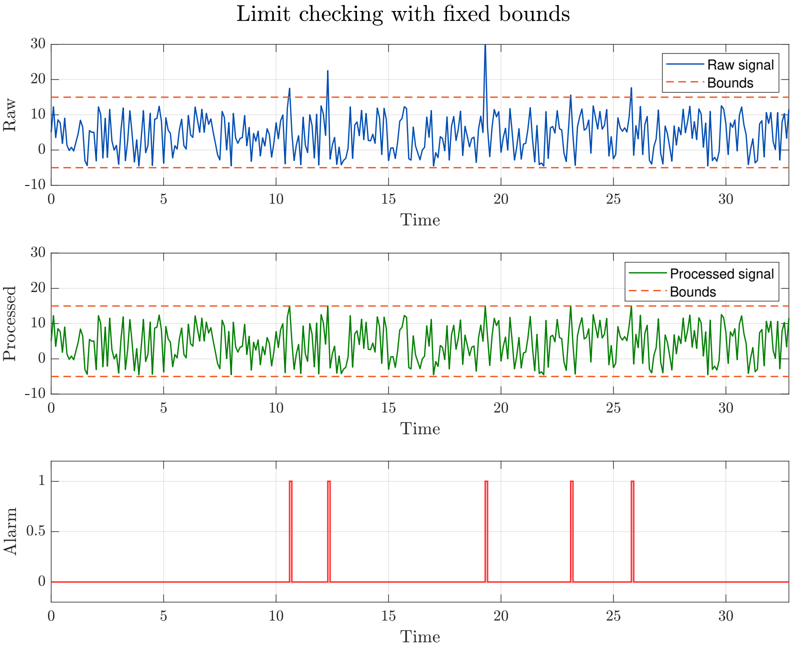


Figure 3.10: Testing - Limit checking

3.6.3 Interquartile Range

The method of limit checking requires both prior knowledge about the measurements and measurements that can be defined and limited by an upper and lower bound. The method of IQR uses the statistics of the studied data to determine the limits, making it a more convenient alternative for more practical measurements where no prior information is available.

The algorithm is implemented with an adjustable window size. This is done in order to make it more time adaptive, meaning that one fault only affects the statistic of n number of samples, where n denotes the window size. In addition, this means that it can handle a changing measurement without either having to trigger false alarms or not being able to detect faults.

This example is shown in Figure 3.11 and 3.12, where the red dotted lines represent the upper and lower bound. Figure 3.11 shows how the algorithm worked before a moving window was implemented. This result is similar to the result the method of limit checking would have given, except the limits are set automatically. Since the algorithm uses the statistics of the entire data set, it chooses limits that fit both start and end value. Due to this, no faults occurring early in the data set are detected.

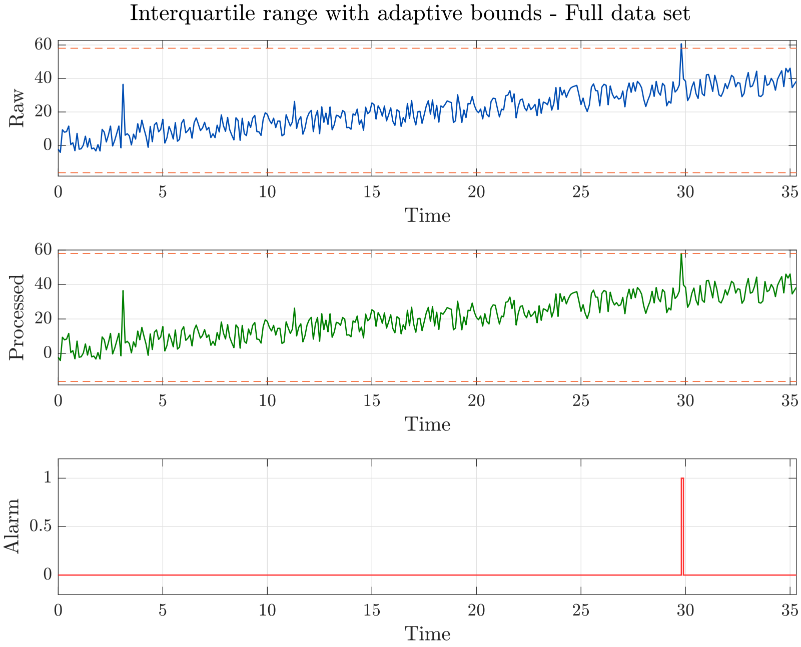


Figure 3.11: Testing - IQR with no moving window

Figure 3.12 shows the result after the moving window is implemented, and the window size is set to 20. The limits are able to adapt to the statistics of the current data more precisely, and the algorithm manages to find both added outliers.

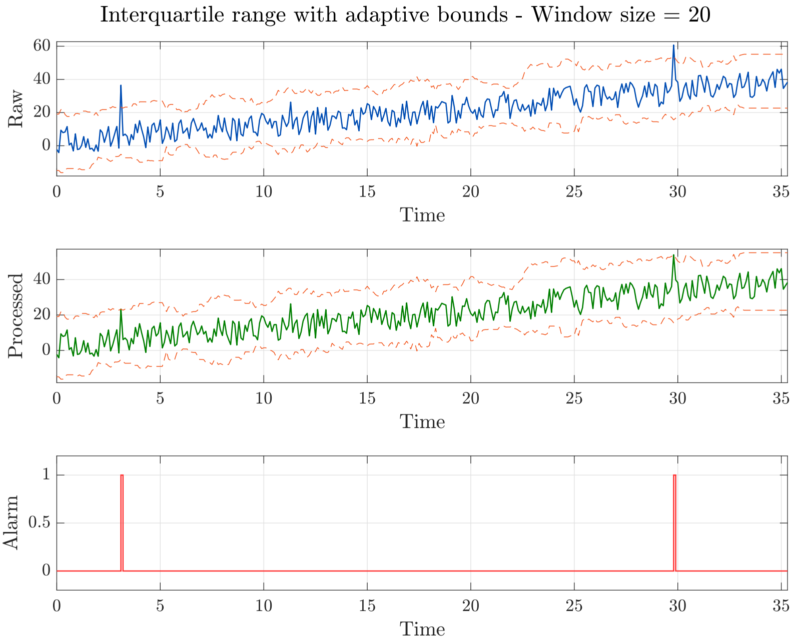


Figure 3.12: Testing - IQR with window size 20

3.6.4 Variance Check - Frozen Signals

The variance check intended to detect frozen signals is implemented with two detection methods. The first method checks if the variance of any point is equal to zero, and the second checks if a signal value is exactly equal to its previous value for a significant number of consecutive measurements. By implementing the second method, the algorithm should be able to detect frozen signals independent on the variance sample size. To test the performance, the algorithm was tested with a sample size equal to the size of the data set, giving a constant variance for all values. This is shown in Figure 3.13. Even though the algorithm never discovers the occurrences of zero variance, the frozen signals are detected, and the alarm is triggered.

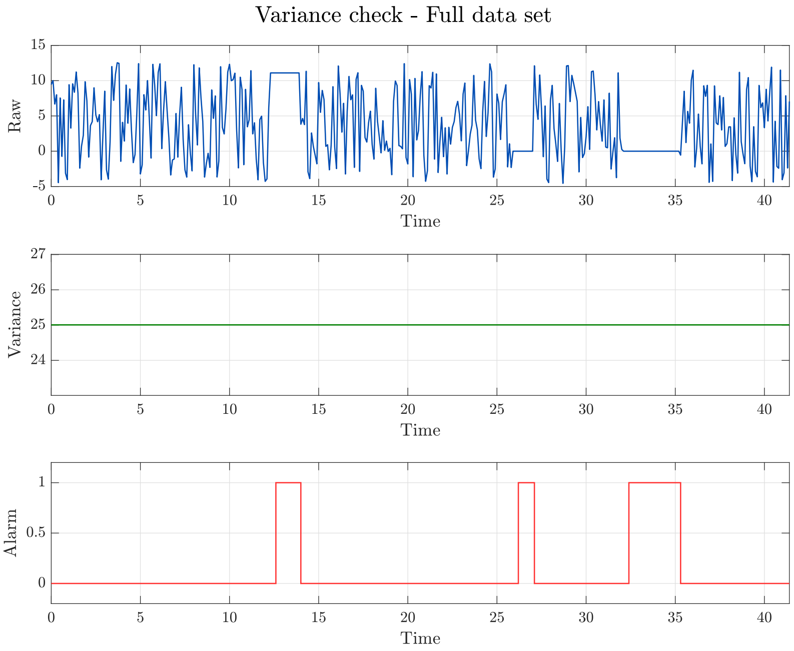


Figure 3.13: Testing - Frozen signal check with all data points

Figure 3.14 illustrates how a lower sample size affects the variance. With a sample size of 10, the variance fluctuates with time, and the zero variance is detectable using both methods.

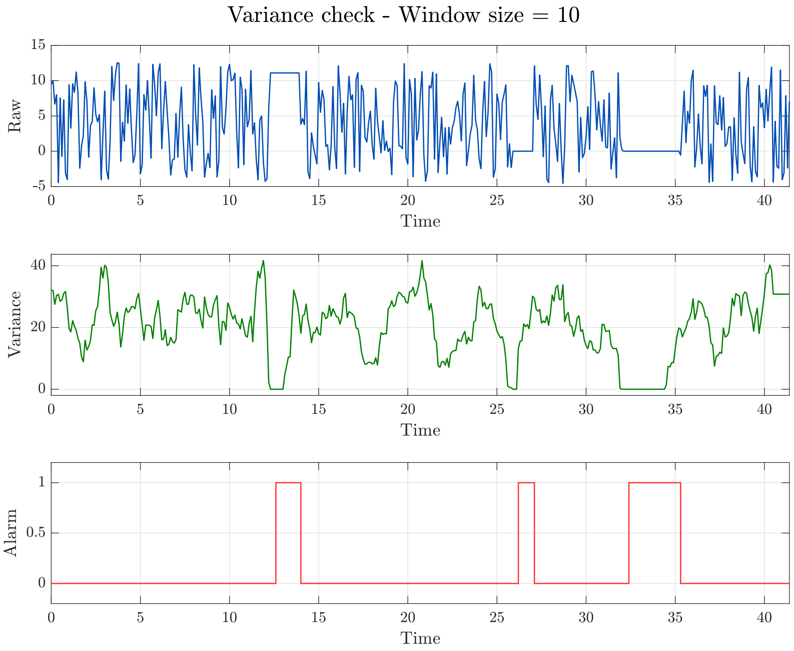


Figure 3.14: Testing - Frozen signal check with sample size 10

3.6.5 Variance Check - High Variance

To detect the occurrence of high variance, the method of interquartile range is applied using the variance as input. The sample size for the variance is adjustable such that it can handle data sets with different frequencies of sampling. The sample size tuned to get a satisfactory balance between detection delay and fluctuations. The variable that determines the tolerance of the thresholds is set as the default value of 1.5 in this analysis.

The window size of the IQR algorithm is set to equal the size of the data set. This is possible since the variance is supposed to remain approximately constant, as opposed to a changing signal as tested in Section 3.6.3. The sample size is set to 50. The test signal could, for instance, represent the surge movement of an ROV, where the signal is subjected to disturbances creating a temporary increased variance. The result presented in Figure 3.15 shows that the algorithm is able to detect the periods of high variance and toggle the alarm.

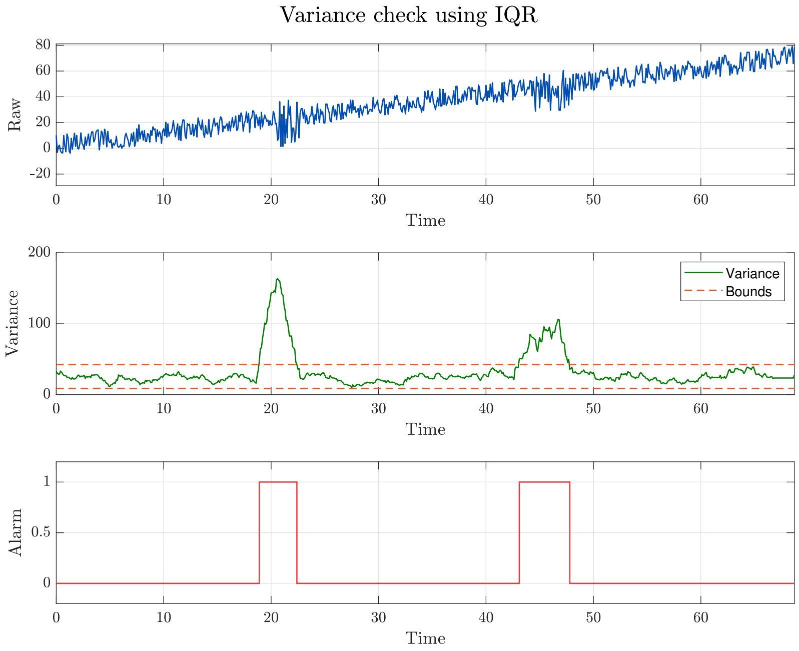


Figure 3.15: Testing - High variance check with all data points

3.6.6 CUSUM

The LS CUSUM filter is implemented to detect abrupt changes in the signal. It requires a predefined threshold h and a drift term ν. When conducting the test, these values were tuned based on the tuning method proposed in Section 3.5. ν was set to half of the expected change, which equals half of the noise amplitude, and h was decreased until a satisfying number of alarms were triggered.

The upper plot of Figure 3.16 shows the signal value together with its estimate. Halfway through the data set the signal value shows a sudden change in the mean value, which is due to a bias added to the signal. The middle plot shows the variables $g_t^{(1)}$ and $g_t^{(2)}$ together with the predefined thresholds. $g_t^{(1)}$ and $g_t^{(2)}$ sum up the errors between the actual signal value and the estimated value. When the bias occurs, $g_t^{(1)}$ exceeds the upper threshold and triggers the alarm. When the alarm is triggered, the algorithm is reset such that the biased mean is accepted as the true mean until the next abrupt change is detected.

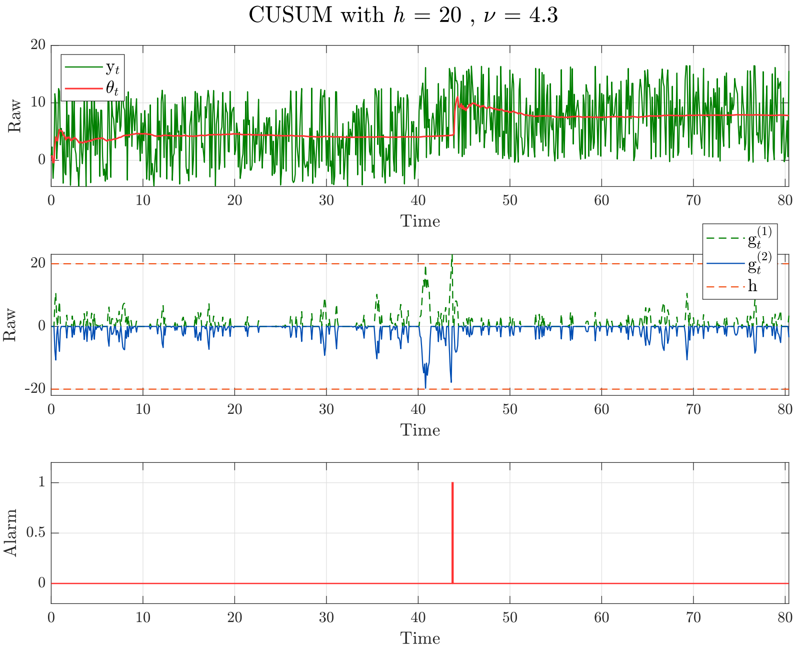


Figure 3.16: Testing - CUSUM with $h = 20$ and $\nu = 4.3$

Chapter 4

Dynamic Thresholds

A common feature of fault detection algorithms is the need for predefined threshold values. These thresholds determine how much slack the signal is given before an alarm is triggered. The LS CUSUM filter, presented in Section 3.5, needs a threshold value h for the stopping rule, and the method of interquartile range, presented in Section 3.3, contains a variable (with default value 1.5) that can be adjusted according to the requirements of the system. The currently used method for determining these thresholds is to tune the values to obtain a balance between the number of false alarms and the time delay of detection. This method requires trial and error and may be difficult in real-time systems where no prior information about the measurements is available.

Chapter 3 presented two methods of making the bounds adaptive with respect to time, including the use of a moving window and a loss function. This ensures that one faulty signal does not affect the statistics of the entire dataset, but only for a specified number of samples given by the window size or by the forgetting factor. The objective of introducing dynamic thresholds is to not only make the fault detection time adaptive but also operation adaptive.

The term *dynamic thresholds* means that the limits are adjusted dynamically based on several factors. This includes factors such as environmental conditions, vessel conditions, and vessel use mode, where all factors are studied with a focus on how they affect the safety requirements of the current operation. This chapter starts by defining and discussing the relevant conditions affecting the optimal threshold. Further, a simplified method for including these when determining the threshold values is proposed.

4.1 Vessel Operational Conditions

Following the method in [40], the *Vessel Operational Conditions (VOC)* can be defined as a triplet of attributes, namely:

$$VOC := \langle VUM,\ Env,\ VC \rangle\,, \tag{4.1}$$

where *VUM* refers to the *Vessel Use Mode*, which describes the current task of the vessel, *Env* refers to the state of the environment and *VC* refers to the current conditions of the vessel. Figure 4.1 shows some examples of vessels operating in different *VOC*.

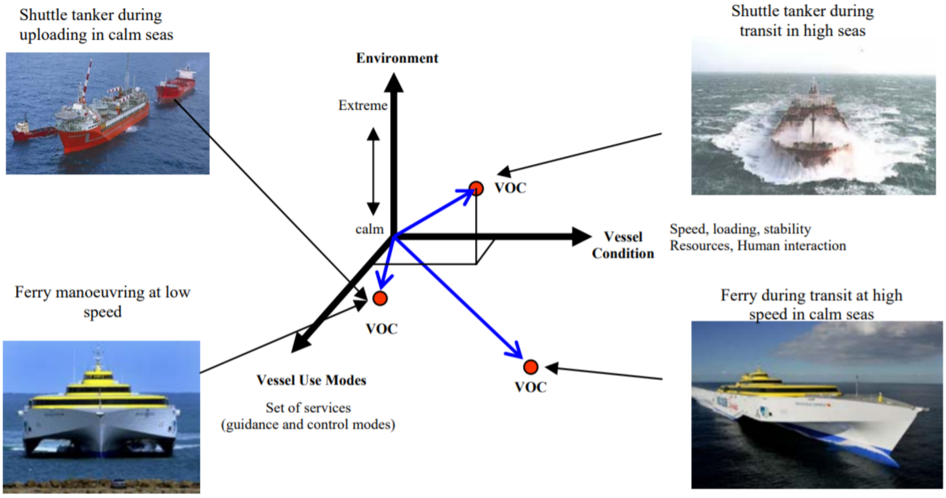

Shuttle tanker during uploading in calm seas
Shuttle tanker during transit in high seas
Environment
Extreme
calm
VOC
Vessel Condition
Speed, loading, stability
Resources, Human interaction
Ferry manoeuvring at low speed
Ferry during transit at high speed in calm seas
VOC
Vessel Use Modes
Set of services
(guidance and control modes)
VOC

Figure 4.1: Examples of vessels with different *Vessel Operational Conditions (VOC)*. Taken from Figure 4 in [41].

The vessel operational conditions are used to classify different parametric models, which are further used for control analysis and design. In this thesis, this property will be modified such that it classifies different safety classes instead of control modes. This new property is called the *Vessel Operational Safety Conditions* to highlight how the focus is shifted towards safety and risk.

4.2 Vessel Operational Safety Conditions

Marine vessels perform complex tasks and operations under varying environmental conditions. Due to the variety of both operational and environmental conditions under which these tasks must be performed, various degrees of accuracy may be required in order to avoid accidents and ensure safety. This introduces the necessity of being able to quantify the security requirement of an operation based on current external factors. This quantification can be done by presenting the *Vessel Operational Safety Conditions (VOSC)*, based on the *VOC*. The *VOSC* is a combination of the same three attributes as the *VOC*, which will be further explained in the following subsections.

4.2.1 Environmental Conditions

Marine systems are designed to carry out complex operations in harsh and changing environmental conditions. These conditions have a great impact on the precision needed to complete the given task safely. Strong gusts or a high sea may cause larger variations in measurements than in conditions with no wind and calm sea. This means that an abrupt change in heading may indicate a fault under some conditions, while it can be expected under tougher conditions. The *environmental conditions*, *Env*, incorporates the environmental conditions in the *VOSC*. This includes both wind, waves and current, and can roughly be classified into calm, moderate and rough environmental conditions.

4.2.2 Vessel Use Mode

Vessels perform a large variety of tasks, referred to as missions or operations. Each of these missions may require the vessel to operate in a different *Vessel Use Mode (VUM)*. Some examples of *VUMs*, as presented in [41], are:

- Transit
- ROV support
- Surveying
- Drilling
- Pipe lying

Each of these *VUMs* may require different types of GNC systems, which can be referred to as a *GNC service*. These are the services that are required by the *VUM* to conduct a given mission or operation. Each of these services requires a different level of accuracy. Examples of these services are:

- Station keeping
- Course keeping
- Manoeuvring (trajectory tracking)
- Waypoint tracking
- Stopping
- Docking (homing)

4.2.3 Vessel Conditions

The attribute *vessel conditions (VC)* refers to the current conditions of the vessel. This includes the vessel speed, loading, and resources. Resources refer to factors such that human interaction, available power, and range. Relevant factors in the context of signal credibility include sensor type, signal strength, and redundancy.

4.3 Threshold Setting

The goal of this chapter is to develop a strategy for dynamically determining the threshold values needed for fault detection. The thresholds are crucial in order to balance the number of false alarms while ensuring safe operations. Figure 4.2 illustrates the set-up of the proposed strategy. The dynamic threshold setting block takes the current *VOSC* as input, together with relevant information about the signal. Based on this, it sends the calculated thresholds h to the signal processing unit, which is responsible for detecting faults in the signal.

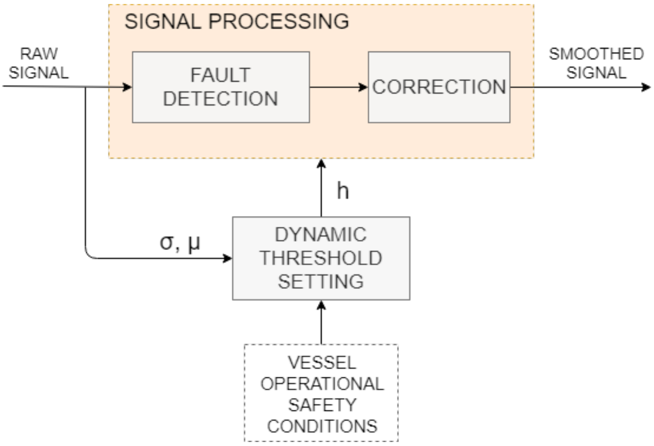


Figure 4.2: Dynamic threshold setting

A simplified model of the thresholds can be written as:

$$\text{Upper} = h_{upper} = \mu + a \cdot \sigma \tag{4.2}$$

$$\text{Lower} = h_{lower} = \mu - a \cdot \sigma, \tag{4.3}$$

where a is to be determined based on the current *VOSC*. To be able to do this, all the attributes of the *VOSC* must be quantified for the studied operation, with respect to how they affect the requirements of the alarming system. Table 4.1

shows an example of how this can be done, where the gains for each attribute are in the range of one to three.

Let the value one correspond to a scenario requiring a strict threshold, i.e., high accuracy or conditions where a small deviation may indicate a fault, and three correspond to a scenario where maximum tolerance is allowed. Under *VC*, the different gains are modeled by considering the conditions of an AUV, ROV, and ship. This is based on the assumption that an AUV requires stricter gains as a critical fault may lead to loss of vehicle. A ship under transit in rough weather conditions may allow a larger tolerance in its thresholds, as abrupt deviations may be due to large wave forces and should not be discarded.

It can be questioned whether it would be better to set the thresholds as maximum strict regardless of the external conditions, and thus guarantee safe operations. The answer to this is that some circumstances may lead to abrupt changes that would not be physically possible under different conditions, such as with calm weather. To discard a signal as a fault when it is valid can be just as damaging as an undetected fault.

Table 4.1: Classification of VOSC attributes

Gain	1	2	3
VUM	Docking	Manoeuvring	Transit
Env	Calm	Moderate	Rough
VC	AUV conditions	ROV conditions	Ship conditions

When the gains for each attribute are defined, the $VOSC_{gain}$ can be calculated as the magnitude of the three-dimensional vector, as shown in Figure 4.3. This is easily done by using the formula:

$$M = \sqrt{x^2 + y^2 + z^2} = \sqrt{VUM^2 + Env^2 + VC^2} = VOSC_{gain}. \tag{4.4}$$

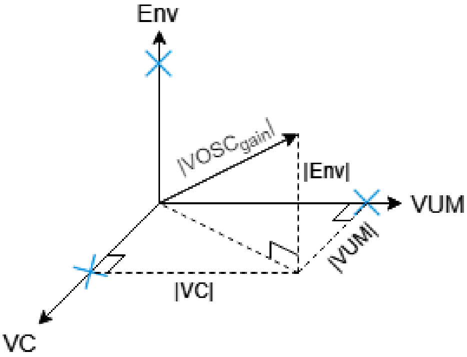


Figure 4.3: Magnitude of $VOSC_{gain} = |\langle VUM,\ Env,\ VC \rangle|$

The calculated $VOSC_{gain}$ can be combined with the statistics of the signal by utilizing the properties of the normal distribution. As shown in Figure 4.4, only 0.3% of a normally distributed data set can be found more than three standard deviations from the signal mean. This means that a strict threshold can be set to equal $4 \cdot \sigma$. As a starting point, the maximum allowable threshold can be set to $9 \cdot \sigma$ [26]. The $VOSC_{gain}$ is responsible for determining which value in this interval that should be used in a given scenario.

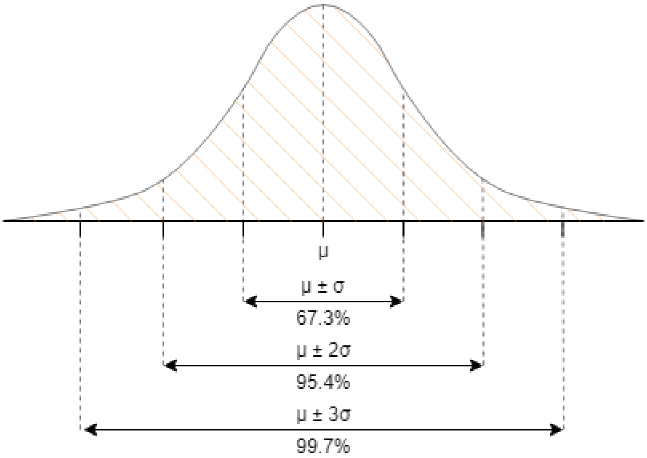


Figure 4.4: Probabilities associated with a normal distribution. Based on Figure 4.12 in [29].

The final part of the method is based on coupling each $VOSC_{gain}$ to an a-value, which further determines the upper and lower bounds using the signal statistics. A finite number of possible $VOSC_{gains}$ can be determined, depending on the number of possible states for each of the attributes $\langle VUM,\ Env,\ VC \rangle$. These are sorted and normalized by dividing by the smallest value. Each of these values can be paired to an a-value by interpolation. This is shown in Figure 4.5.

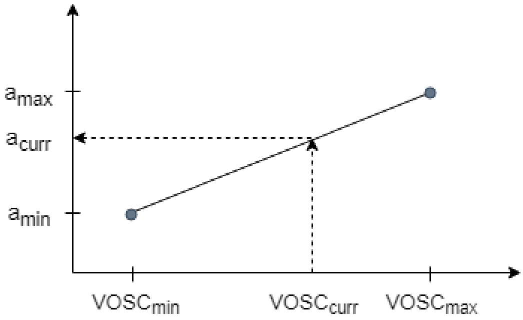


Figure 4.5: Interpolation of the threshold parameter a based on current $VOSC_{gain}$

A simplified pseudocode for dynamically setting the upper and lower thresholds is as follows:

```
Input: $\mu$, $\sigma$, $a_{min}$, $a_{max}$, $\langle VUM, Env, VC \rangle$
Output: $h_{lower}$, $h_{upper}$


   for each possible VUM:
      for each possible VC:
         for each possible Env:
            Calculate $VOSC_{gain}$ for all combinations;
            Save current gain as $VOSC_{curr}$;


   Normalize every VOSC by dividing each VOSC with min(VOSC);

   Calculate change in $a$ as:    $\Delta a = a_{max} - a_{min}$;
   Calculate change in VOSC as:    $\Delta VOSC = VOSC_{max} - VOSC_{min}$;

   Interpolate to find a: a $= a_{min} + \frac{\Delta a}{\Delta VOSC} \cdot (VOSC_{curr} - VOSC_{min})$;

   Upper bound: $h_{upper} = \mu + a \cdot \sigma$;
   Lower bound: $h_{lower} = \mu$ - $a \cdot \sigma$;
```

A description of the implementation of the algorithm in LabVIEW is attached in Appendix C.7.

Chapter 5

Case Study: Simulation with Dynamic Thresholds

This chapter will present a practical example to illustrate the concept of dynamical thresholds in fault detection. The developed algorithm is generic and can be applied for many types of systems and operations but will here be tested in the context of subsea operations. The operation will be simulated using a generated signal. The use of synthetic test signals makes it possible to include faults without being dependent on faults occurring in a real simulation.

This chapter starts by describing the physical aspects and challenges of the studied scenario before it discusses and quantifies the attributes that together determine the final $VOSC_{gain}$ of the system. Further, this gain is used to dynamically determine the thresholds used in fault detection. The method will be tested on a signal with both a constant and varying mean.

5.1 Case Description

The case study aims to present the method of dynamic thresholds from a more practical point of view. In order to do so, a practical scenario must be considered, which includes information about all the factors affecting the vehicle as it completes its mission. Such factors cover the vehicle, the environmental conditions, and the operational conditions.

The scenario to be studied is an ROV that has finished its intervention mission and starts the return back to its seabed mounted docking station. The return can be divided into three separate phases of operation, where the vessel use mode, the vessel conditions, and the environmental conditions vary depending on the current phase.

5.1.1 Phases of Operation

Consider an ROV that has finished its mission, such as inspection of a seabed mounted construction or cable. The ROV must autonomously return to its docking station to recharge, upload collected data, and download new missions. During this approach the main sensor system changes, as well as the vehicle speed and the necessary level of accuracy in measurements. The environmental and operational conditions may vary with the position of the ROV. By defining separate intervals where these factors are approximately constant, the approach can be divided into three separate phases. The thresholds are set based on many assumptions made about the surroundings of the ROV through these phases.

Phase 1 - Transit
The first phase is called the transit phase. The goal of this phase is to navigate from the ended mission to a location in proximity of the docking station. It is assumed that a track of USBL transponders connects the location of the ended mission and the location of the docking station, as graphically illustrated in Figure 5.1. This distance can be several kilometers long, where the ROV can keep a constant and relatively high speed without needing very accurate measurements. The ROV can be subjected to strong underwater currents through this phase but does not need a strict threshold as some temporary deviations in this phase are allowable.

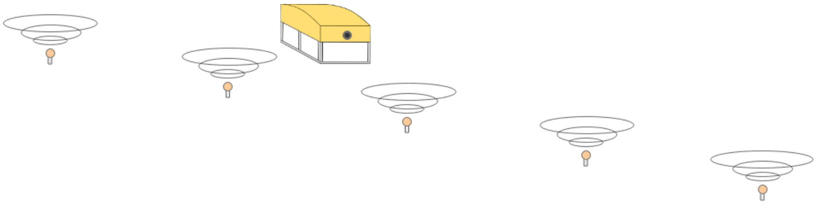


Figure 5.1: Phase 1 - Transit: Following a seabed mounted USBL track

Phase 2 - Manoeuvring
Moving into the low-speed maneuvering phase, it is assumed that the docking station is placed in the middle of an LBL network consisting of four transponders placed in a square. The assumed set-up is shown in Figure 5.2, where the grey box illustrates the dock. This system can provide a higher accuracy towards the station, and therefore requires stricter thresholds. The goal of the maneuvering phase is to position the ROV relative to the docking station and lower the speed to prepare for the homing phase. Assuming the docking station is located in a more secluded environment, deviations due to environmental conditions are expected to be small.

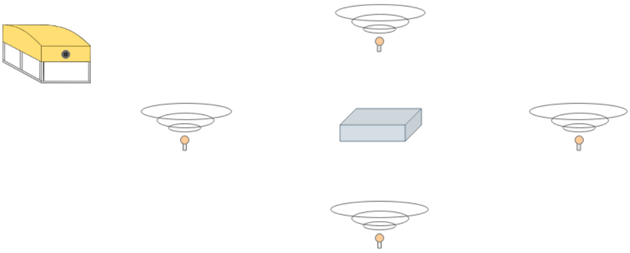


Figure 5.2: Phase 2 - Manoeuvring: Receiving signals from an LBL network in proximity of the docking station

Phase 3 - Homing

The final phase starts when the visual instruments of the ROV can detect the docking station. The goal of the homing phase is to connect the ROV to the docking station successively. Through this phase, low speed and reliable measurements are crucial. This phase has the highest requirement for accuracy

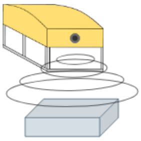


Figure 5.3: Phase 3 - Homing: Using the ROV's visual instruments to connect to the docking station

All the main properties of each phase can be divided into the three categories relevant in the context of threshold setting, namely the *vessel use mode, environmental conditions*, and *vessel conditions*. These properties are summarized in Table 5.1

Table 5.1: Main properties of the three phases of operation

Phase	**1: Transit**	**2: Manoeuvring**	**3: Homing**
Vessel Use Mode (VUM)	Course keeping over a large distance, from ended mission to docking station	Lowering of speed and manoeuvring to prepare for the homing phase	Slowly and accurately connect to the docking station
Environmental Conditions (Env)	May be exposed to strong currents	Assuming less current forces	Assuming calm conditions in proximity of dock
Vessel Conditions (VC)	Uses a USBL track with relatively high speed	Uses a LBL system with increased accuracy and lowered speed	Uses camera vision to slowly and accurately position itself

5.1.2 VOSC

Considering the main properties of the three phases described in Table 5.1, the attributes to determine each $VOSC_{gain}$ can be quantified. The vessel use mode is different for each phase, as the objective of the phase changes. The vessel conditions vary with the different sensors and speeds used. The environmental conditions are challenging to determine, as they may vary from hour to hour and do not necessarily depend on the phase of operation. As a simplification, they are set as constant within each phase, with decreasing strength as the ROV approaches the docking station. This is done under the assumption that the docking station is located in a more sheltered location than the site of the ended mission.

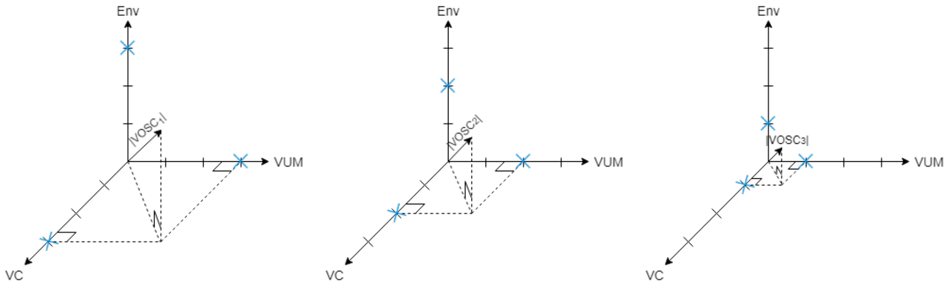
It is important to note that the gains are determined by focusing on how each factor affects the need for strict thresholds. When being exposed to rough environmental conditions, a more tolerant threshold may be necessary, as more significant deviations are expected. A strict threshold under these conditions may result in a number of false alarms and valid signals getting discarded as faults.

Each gain is quantified with a number between one and three for this simulation. The numbers should be interpreted such that one means that the corresponding factor causes the need for a strict limit, while three means that a higher tolerance is allowable. Following this logic, all gains are set and presented in Table 5.2.

Table 5.2: Determination of VOSC attributes

Phase	1: Transit	2: Manoeuvring	3: Homing
Vessel Use Mode (VUM)	3 (Tolerant)	2 (Moderate)	1 (Strict)
Environmental Conditions (Env)	3 (Tolerant)	2 (Moderate)	1 (Strict)
Vessel Conditions (VC)	3 (Tolerant)	2 (Moderate)	1 (Strict)

The final $VOSC_{gain}$ is determined as the magnitude of the three-dimensional vector when all the attributes are determined. Figure 5.4 shows a spatial representation of each gain.



(a) Phase 1: Transit **(b)** Phase 2: Manoeuvring **(c)** Phase 3: Homing

Figure 5.4: $VOSC_{gain}$ for the three phases of operation

5.1.3 Simulation Signal

The test signal simulates the approach toward the docking station. During the approach, the ROV uses three different sensor systems, namely USBL, LBL, and camera vision. A change of the main sensor system can be used as an indication to when a new phase begins, and the thresholds should be recalculated using a new $VOSC_{gain}$. To simulate this change, the frequency of the input signal is changed as the ROV moves through the phases. By continuously identifying the incoming frequency, the system should be able to distinguish between each of the three phases. An abrupt change in frequency indicates a new phase, which means that the system should retrieve the new corresponding $VOSC_{gain}$, its corresponding a-value, and recalculate the limits based on the current mean and variance.

Figure 5.5 shows an example of a test signal. The upper plot shows the raw signal as a function of time, and the two dotted lines represent the time instants where the frequency changes and the vehicle moves into a new phase. The lower plot shows how the frequency varies with time, starting at 10 Hz in phase 1, changing to 35 Hz in phase 2 and reaching 100 Hz in phase 3.

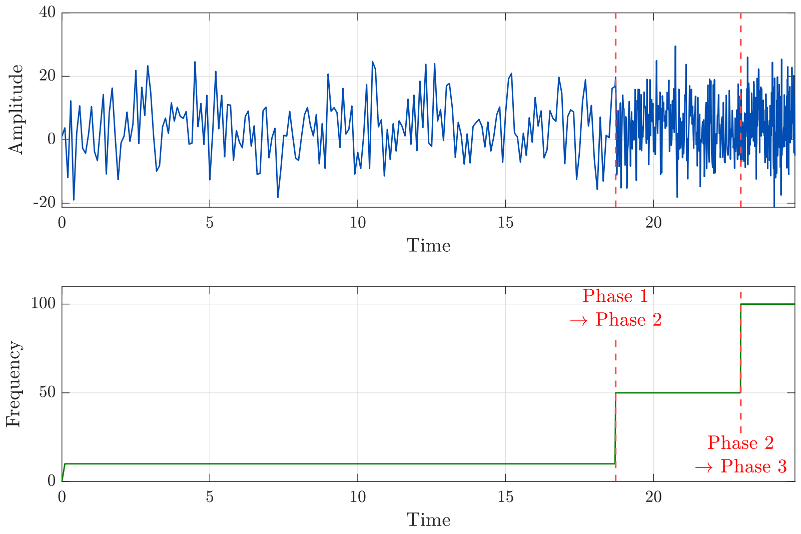


Figure 5.5: Example of a test signal

An estimate for appropriate frequencies is based on information from [42]. This presentation about hydroacoustics includes a table with information about the typical position accuracy that can be expected at different frequencies. Table 5.3 shows these frequencies, their typical accuracies, and to which phase this bandwidth is assigned.

Table 5.3: Frequency bandwidth for the three phases of operation

Frequency Band	Bandwidth	Position Accuracy	Phase
Low Frequency	7.5-15 kHz	0.5-2.5 m	1
Medium Frequency	19-36 kHz	0.25-1 m	2
Extra High Frequency	50-110 kHz	<0.05 m	3

The algorithm is tested on two different simulation cases, each with both a clean and a polluted signal. The frequency is changed equally in both cases based on the values from Table 5.3. With regards to computational time, all frequencies are scaled down by a factor of one thousand compared to the table values. The objective is only to emphasize the algorithm's ability to dynamically change the thresholds, which is evident enough in short and simplified simulations. The signal has a frequency of 10 Hz in phase 1, 35 Hz in phase 2, and 100 Hz in phase 3. The two test cases can be briefly described as:

1. **Case 1 - Yaw movement:** A generated signal simulating the yaw movement, i.e., the heading, of an ROV as it approaches the docking station. The heading is assumed to be constant throughout the approach. When the ROV enters a new phase, a new frequency is detected from the sensor system, and the limits should be recalculated.
2. **Case 2 - Surge movement:** A generated signal simulating the surge movement, i.e., movement in the x-direction, of an ROV as it approaches the docking station. The movement is simulated as a constant increase in position with time. The change of frequency indicates that the ROV receives signals from a new sensor, and the system should react by updating the $VOSC_{gain}$ based on the new conditions.

For both cases, two versions of the signal are tested, namely:

(a) *Clean signal:* A generated signal with no added faults, except noise. The objective of this test is to assure that the algorithm works and that in a fault-free situation, no false alarms are triggered.

(b) *Polluted signal:* A signal polluted with outliers in every phase by using the signal generation module. The outliers added have a constant deviation from the mean value in every phase. The objective is to test that equal deviations have varying severity depending on the phase in which it occurs.

5.2 Results

The first four columns of Table 5.4 show the variables that are predefined by the user. The last two columns show the calculated values that determine the dynamic thresholds.

Table 5.4: Resulting threshold values

Phase	VUM	Env	VC	VOSC	a
1	3	3	3	5.1961	9
2	2	2	2	3.4641	6.5
3	1	1	1	1.7320	4

5.2.1 Case 1 - Yaw Movement

The first test case simulates the yaw movement of the ROV. The heading is assumed to be constant throughout the approach. The algorithm is tested on both a clean and polluted signal.

1a - Clean signal

The first test is conducted using a clean signal without any faults present. The results are shown in Figure 5.6. The upper plot shows that the algorithm behaves as desired. In the first phase of operation, the interval of allowed signal values is wide. As the frequency increases, the limits get narrower. The limits are most conservative in the final phase of operation, where small deviations from the nominal behavior would have triggered an alarm.

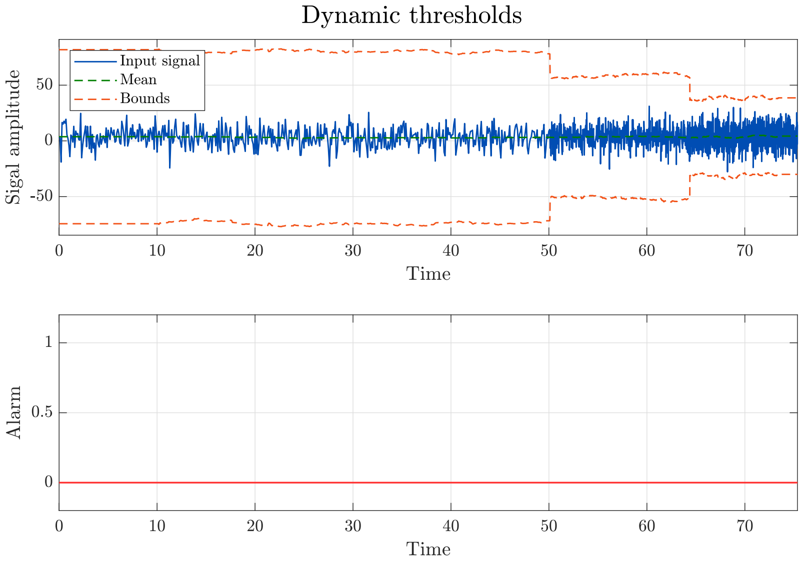


Figure 5.6: 1a: Yaw movement - Clean signal

The probability density function shows how the thresholds are set in relation to the signal values. Figure 5.7 displays the calculated thresholds relative to the distribution for each phase. The thresholds of the final phase show that any deviation outside this distribution would trigger an alarm, while the first two phases would accept significant deviations before discarding the value as a fault.

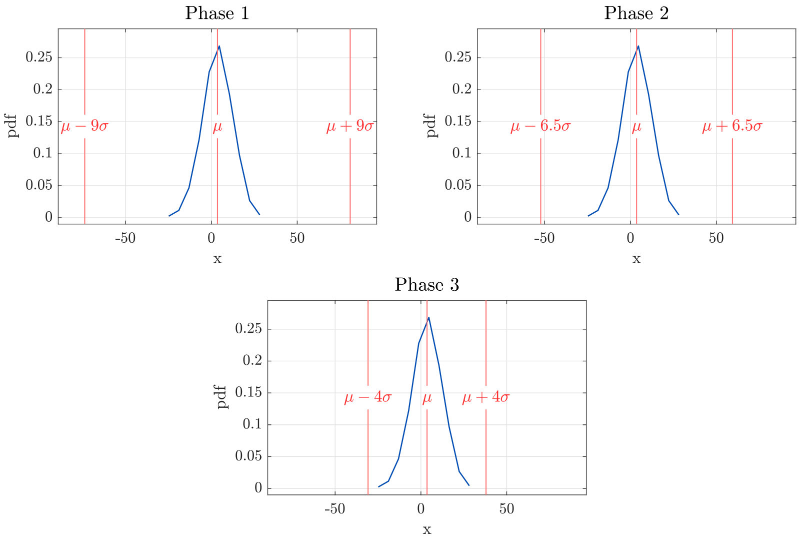


Figure 5.7: 1a: Signal probability density distribution - Clean signal

1b - Polluted signal

The second test is performed on a polluted signal to see how the algorithm handles deviations of equal magnitude occurring in different phases. The signal has a constant mean throughout the simulation.

The results are presented in Figure 5.8 and 5.9. The first figure shows the signal with its mean and the calculated limits. The outliers occur in each phase, with approximately equal deviations from the mean. The bottom plot shows that the faults only trigger alarms in the second and third phase of operation, where the $VOSC_{gain}$ indicates the need for stricter limits. The second figure shows the PDF of the polluted signal. The tail of the distribution is longer in this case than in Figure 5.7, due to the outliers. In the last two phases, the limits are set such that a small percentage of the distribution is located outside the allowed interval.

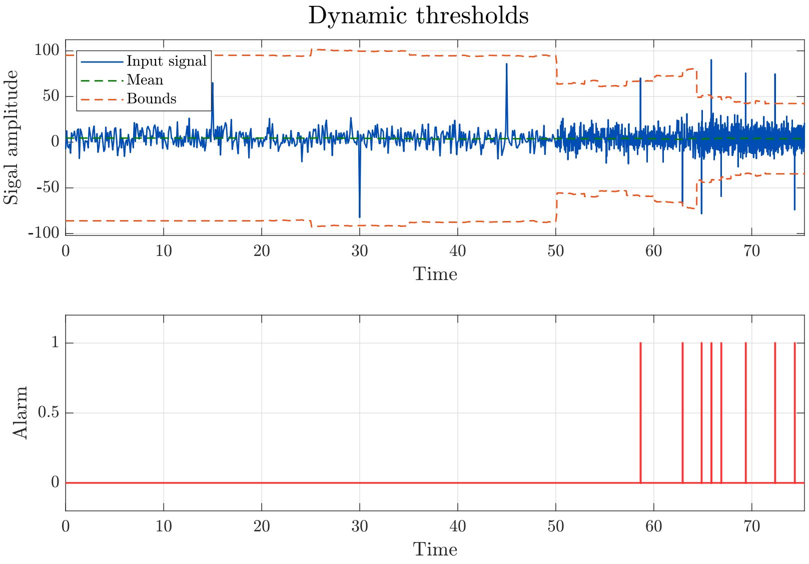


Figure 5.8: 1b: Yaw movement - Polluted signal

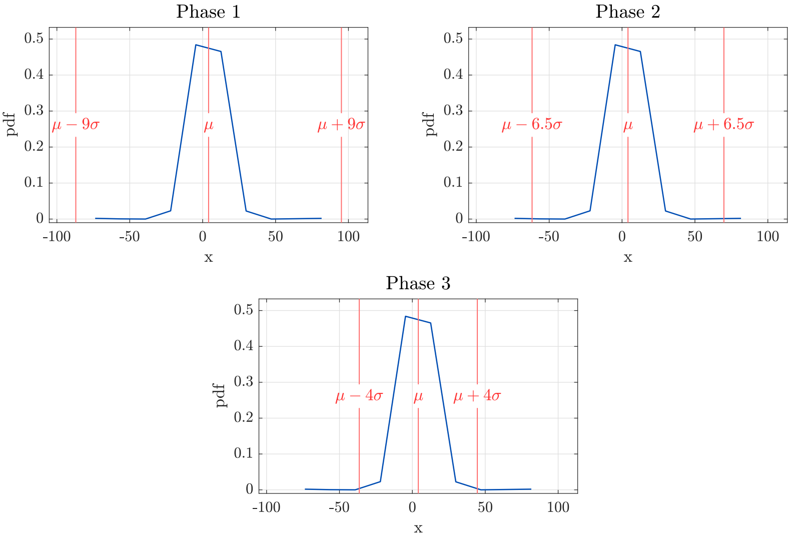


Figure 5.9: 1b: Signal probability density distribution - Polluted signal

5.2.2 Case 2 - Surge Movement

The second test scenario simulates the surge movement of the ROV. This case is included to test how the algorithm reacts when the signal is not normally distributed and cannot be represented as in Figure 5.7 and 5.9. The constantly changing mean makes the sample size used to calculate the mean and variance more crucial. The following simulations use a sample size of 100.

1a - Clean signal

The surge movement of the ROV is simulated by a steady drifting signal with constant variance and no added faults. This is based on a simplifying assumption that the ROV keeps constant speed. In reality, the speed should slow down as the distance between the ROV and the docking stations decreases. Figure 5.10 shows the results of the simulation. The thresholds clearly show where a new frequency is detected, and the algorithm reacts by constricting the thresholds. No false alarms are triggered.

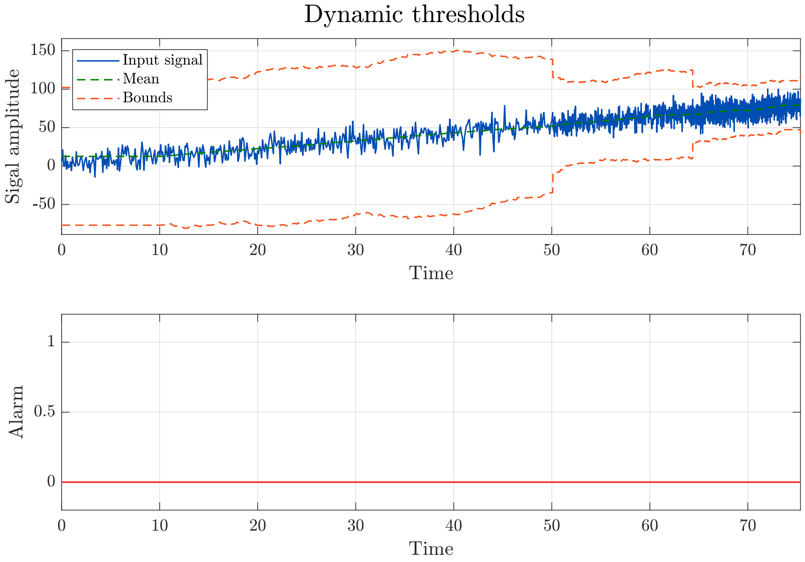


Figure 5.10: 2a: Surge movement - Clean signal

1b - Polluted signal

The polluted version of the simulation signal contains equal outliers in each phase. The resulting plots, shown in Figure 5.11, display the behavior of the thresholds. As the frequency increases, a new $VOSC_{gain}$ is calculated, and the thresholds are changed accordingly. Thus, the algorithm classifies the outliers as faulty signals only in the two most critical phases of the operation.

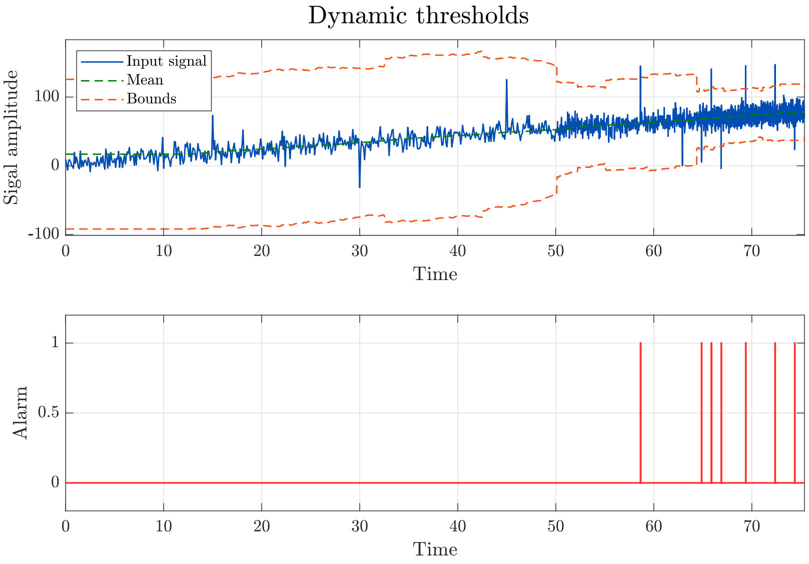


Figure 5.11: 2b: Surge movement - Polluted signal

5.3 Summary

Several methods have been implemented and tested throughout this thesis. Table 5.5 summarizes the advantages and limitations of each method.

Table 5.5: Summary of investigated fault detection methods

Method	Advantages	Limitations
Limit checking	Accurate limits for measurements with a clearly defined range of validity.	Requires prior knowledge about the measurements and the existence of a defined range of validity.
Interquartile range	Requires no prior information about the system and is able to adapt to time changes in the signal.	Does not include external factors. Still requires tuning of one variable.
Variance check: Frozen signals	Accurately detects if a signal is frozen using two different detection approaches.	Is not able to provide an estimated value while the signal is frozen.
Variance check: High variance	As the variance is assumed to be approximately constant, the entire data set can be used to detect faults, giving an accurate description of the variance.	Does not include any function to handle a sudden peak in variance.
CUSUM	Effective to detect abrupt changes in the signal mean.	Requires a lot of tuning to get accurate results, both of the threshold value and the drift term.
Dynamic thresholds	Allows dynamic thresholds that includes external factors and that is customized for the current operation. Requires less tuning.	Can be difficult to set minimum and maximum threshold value. Needs an improved division of attributes.

Chapter 6

Discussion

This chapter provides a discussion of the proposed solution to the research question, together with a comparison of the existing methods that have been investigated in this thesis. The covered topics include the main outcome, the limitations of the suggested method, and finally, a discussion of potential risks arising from implementing the algorithm in a real control system.

6.1 Main Outcome

The main outcome of this thesis is a proposed solution to integrate external factors, including operational, vehicle, and environmental conditions, to determine the thresholds used in fault detection. The algorithm was implemented and tested in a case study to investigate the performance of the method. The results are satisfactory and show that the method has potential if further developed. Note that the implemented algorithm is based on several simplifications concerning real-life operations and conditions, but is only meant to serve as a framework for further work.

This thesis has tested several existing methods for fault detection. The performance of each is summarized in Table 5.5. They all show their strengths and weaknesses but have the common drawback of not including any external conditions. The method of limit checking with fixed bounds has a great performance with measurements that have a predefined interval of validity. However, the area of use is very narrow, as most measurements do not have this property.

Considering the method of interquartile range, this algorithm is able to adapt to an unknown and varying signal. It is based solely on the statistics of the data set and can be modified such that it contains one variable. This variable can be set to determine how tolerant or strict the thresholds should be. This means that this method is also compatible with the method of dynamic thresholds. Instead of using a dynamically set number of standard deviations from the mean, a defined number of interquartile ranges from the quartiles can be applied.

The implemented methods for detecting faults in the variance fulfilled their purpose. These are the algorithms that require the smallest amount of tuning. By implementing two methods to detect a frozen signal, the performance is not dependent on the used sample size. To detect high variance, the interquartile range method was modified with the variance as input. The results were satisfactory for both methods.

The final implemented method was the LS CUSUM filter. This method requires more tuning as it contains two variables, which could be very time-consuming in a real system. This algorithm could also be compatible with dynamic thresholds. The challenge with combining these two methods is to determine an upper and lower boundary for the limits. The variable h, which determines when and how often an alarm is triggered, is not based on the measurement's statistics, making it challenging to set universal limits that work for different measurements.

6.2 Limitations and Challenges

The proposed method requires several improvements before it could be implemented in a real system. The most prominent challenge is to develop a better approach for quantifying the attributes that determine the *VOSC*. In the case study, the $VOSCgain$ is determined by three different attributes, each with three levels of severity each. The level of severity is set based on how strict the given attribute indicates that the limits should be. Considering two cases of the same vehicle, conducting the same operation, using the same sensor systems and speed, rough environmental conditions would induce more tolerant thresholds than calm conditions. Moreover, different operations also require different degrees of accuracy in the measurements. A sudden deviation under transit may not be as critical as the same deviation under a strict DP operation.

The ideal way to quantify all external conditions would have been automatically. For instance, information about wind, waves, and currents should automatically determine the gain for this condition. The same may apply for attributes such as signal strength, vessel speed, and current mission.

One possible solution could be to implement the algorithm as part of a hybrid system. Hybrid system frameworks are especially suited for describing marine vessel dynamics because the variations in dynamical behavior for various vessel operational conditions can be captured using different sub-models merged into one hybrid system. In the same manner, the dynamical behavior for various vessel operational safety conditions can be accounted for using different sub-models in the signal processing module. Figure 6.1 illustrates this concept. The figure is based on a hybrid control system presented in [26], but in addition to a bank of observers, controllers, and allocations, a bank of signal processing units is included.

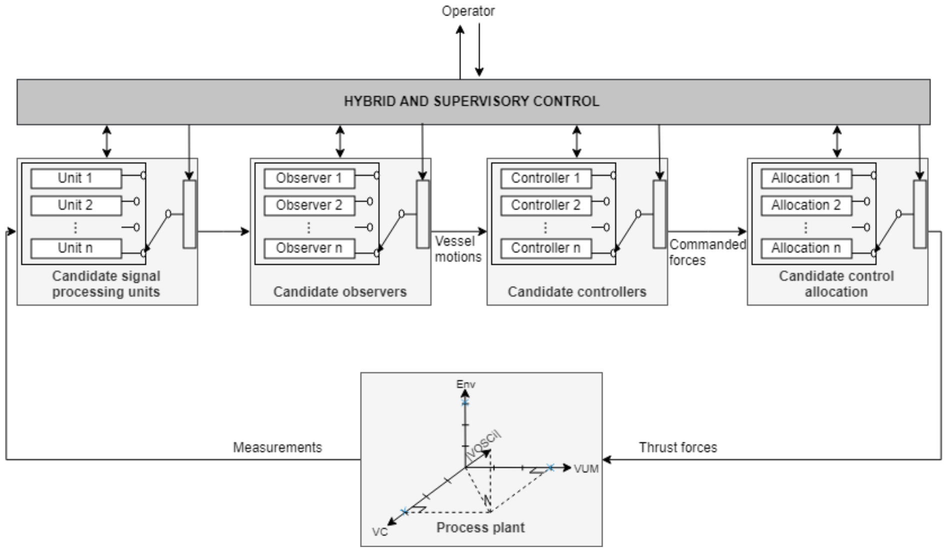


Figure 6.1: Illustration of a hybrid control system for a marine vessel in different vessel operational safety conditions. In addition to the "traditional" set-up, this also includes a bank of signal processing units corresponding to different conditions. Based on Figure 9.2 in [26]

6.3 Induced Risks

When implementing new functions in a system, it induces new risks and consequences. Identifying and assessing these risks is crucial such that the algorithm does not cause more damage than it prevents. Dynamic thresholds introduce the risks for the wrong threshold setting, which may have severe consequences. In a situation where the algorithm sets too strict thresholds, it may result in several false alarms. A high number of subsequent false alarms could result in an operation being interrupted, causing costly delays. On the other hand, if the algorithm sets too tolerant limits, faults can go on undetected, causing errors further in the control system, such as significant fluctuations in commanded thrust.

Chapter 7

Conclusions and Further Work

The research question of this thesis has been:

When studying signal processing for autonomous subsea operations, how can the limiting thresholds be dynamically set based on available information about the environmental, operational and vessel conditions?

The following section concludes the thesis and gives suggestions for further work.

7.1 Concluding Remarks

The research of this thesis is motivated by the desire to highlight the importance of safe and credible signal processing, as both underwater and surface vessels experience a great change towards human independence and full autonomy. Different operations, vehicles, and varying environmental conditions may affect the sensor measurements that a vessel's control system relies on for its performance.

All the tested algorithms have flaws and strengths. In light of the research question of this thesis, they all miss the common feature of including external factors in the threshold calculations. The method of limit checking using fixed bounds is stable and reliable for measurements with a clearly defined interval of validity. The two tests dealing with low and high variance are both easy to implement and well-functioning. These have some variables that have to be set manually, but these are mostly about how long to accept a high or low variance before an alarm should be triggered.

The method of interquartile range is the method most similar to the proposed method, as it is solely based on the statistics of the measurements. It showed satisfactory performance and has the advantage of being adaptive with regards to time changes in the signal. This algorithm contains a variable that can be adjusted based on the tolerance needed for operation. This implies that this method could also have been implemented using dynamic thresholds.

The least squares cumulative sum filter was implemented purely to detect sudden biases. Its performance was good, but the tuning was time-consuming even though the algorithm comes with some simple steps to tune it. This algorithm could have the potential of being combined with dynamic thresholds to make it adaptive with regards to external conditions. However, combining them would introduce the challenge of determining the minimum and maximum threshold values without prior knowledge about the signal's properties.

Considering the proposed method of dynamic thresholds, it can be concluded that it is a realistic proposal but requires a lot of further development before it can be implemented in a real system. Introducing a new function into signal processing also introduces new risks and possible consequences. A three-dimensional model, each with three separate categories, to determine the $VOSC_{gain}$ is too simple for reality. An accurate and describing division of these attributes is crucial for the performance of the method.

This thesis has focused on the necessity of fault detection and how external conditions can be included to detect faults. The main outcome is a proposal for such a method, which in an orderly manner can adjust the limiting thresholds based on several external factors. Thus, it can finally be concluded that the research question has been answered, and this thesis lays the groundwork for such a solution.

7.2 Further Work

During the development process of this thesis, several topics for further investigation have emerged. The list below describes the main recommendations and suggestions for further work.

1. Exploration and comparison of the performance of the dynamic threshold algorithm when implemented together with the interquartile range, versus the results obtained in the case study.

2. Expansion of the three-dimensional model for determining the $VOSC_{gain}$ that determines the final thresholds. The number of necessary attributes should be investigated. Each attribute should also have a finer division.

3. Improvement of the algorithm to automatically quantify the environmental conditions based on wind, wave, and current information. Ideally, the system should be able to identify environmental loads acting on the vessel. It could be implemented by using sub-models in a hybrid control system.

4. Implementation of the algorithm in a HIL simulator to investigate the functionality together with the vessel's response.
5. Identification of all new risks, failure scenarios, and consequences of introducing the proposed algorithm.

Bibliography

[1] NOAA, *How much of the ocean have we explored?*, Online, 2018. Available from: https://oceanservice.noaa.gov/facts/exploration.html.

[2] Haugan, I., *Snake robot ready to go on watch in the deep seas*, *Gemini*, 2019. Available from: https://geminiresearchnews.com/2019/01/snake-robot-ready-to-go-on-watch-in-the-deep-seas/.

[3] ——, *Ntnus dypeste laboratorium åpnes i trondheimsfjorden*, *Gemini*, 2019. Available from: https://gemini.no/kortnytt/ntnus-dypeste-laboratorium-apnes-i-trondheimsfjorden/.

[4] Liljebäck, P. and Mills, R., *Eelume: A flexible and subsea resident IMR vehicle*, in *OCEANS 2017-Aberdeen*, IEEE, 2017, pp. 1–4.

[5] Fidani, A., *SWIMMER: Hybrid AUV/ROV concept*, Online, 2009. Available from: https://d26pw6xcesd4up.cloudfront.net/1433418810/alain_fidani_-_swimmer.pdf.

[6] Beckman, J., *Prototype auvs prove capability for subsea inspection and intervention*, Online. Available from: https://www.offshore-mag.com/articles/print/volume-64/issue-10/subsea/prototype-auvs-prove-capability-for-subsea-inspection-and-intervention.html.

[7] Marty, P., *Alive: An autonomous light intervention vehicle*, *Scandinavian Oil-Gas Magazine*, vol. 32, pp. 10–11, 5/6 2004.

[8] Raspante, F., *Underwater mobile docking of autonomous underwater vehicles*, in *OCEANS 2012 Conference*, 2012, pp. 1–15.

[9] Oceaneering, *E-rov system*, Online, 2018. Available from: https://www.oceaneering.com/datasheets/ROV-E-ROV.pdf.

[10] Saab Seaeye, *Sabertooth*, Online, 2016. Available from: http://www.saabseaeye.com/solutions/underwater-vehicles/sabertooth-single-hull.

[11] BBC, *Lion air crash: Boeing 737 plane crashes in sea off jakarta*, Online, 2018. Available from: https://www.bbc.com/news/world-asia-46014260.

[12] ——, *Ethiopian airlines: 'no survivors' on crashed boeing 737*, Online, 2019. Available from: https://www.bbc.com/news/world-africa-47513508.

[13] Stewart, E., *The boeing 737 max 8 crashes and controversy, explained*, Online, 2019. Available from: https://www.vox.com/2019/3/12/18262359/boeing-737-max-controversy-faa-trump.

[14] Ostrower, J., *What is the boeing 737 max manevering characteristics augumentation system?*, Online, 2018. Available from: https://theaircurrent.com/aviation-safety/what-is-the-boeing-737-max-maneuvering-characteristics-augmentation-system-mcas-jt610/.

[15] Gates, D., *Faa evaluates a potential design flaw on boeing's 737 max after lion air crash*, Online, 2018. Available from: https://www.seattletimes.com/business/boeing-aerospace/faa-evaluates-a-potential-design-flaw-on-boeings-737-max-after-lion-air-crash/.

[16] Ludvigsen, M., *Lecture notes - Arctic Marine Measurements Techniques, Operations and Transport*, 2016.

[17] J. Albus and P.J. Antsaklis, *Setting the stage: Some autonomous thoughts on autonomy*, in *IEEE ISIC/CIRA/ISAS Joint Conference*, IEEE, 1998.

[18] Sørensen, A. J. and Ludvigsen, M., *Underwater technology platforms*, *Encyclopedia of Maritime and Offshore Engineering*, pp. 1–11, 2017.

[19] Ludvigsen, M. and Sørensen, A. J., *Towards integrated autonomous underwater operations for ocean mapping and monitoring*, *Annual Reviews in Control*, vol. 42, pp. 145–157, 2016.

[20] Fossen, T. I., *Handbook of Marine Craft Hydrodynamics and Motion Control*, 4th ed. Wiley, 2011.

[21] Durrant-Whyte, H. and Bailey, T., *Simultaneous localization and mapping: Part i*, *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[22] Milne, P. H., *Underwater Acoustic Positioning Systems.* Gulf Publishing Co., 1983.

[23] Kongsberg Maritime, *HiPAP - Acoustic underwater positioning and navigation systems*, Online. Available from: https://www.kongsberg.com/maritime/products/Acoustics-Positioning-and-Communication/acoustic-positioning-systems/hipap-models/hipap-352?OpenDocument.

[24] Gade, K., *The seven ways to find heading*, *The Journal of Navigation*, vol. 69, no. 5, pp. 955–970, 2016.

[25] Antonelli, G., *Underwater Robots*, 3rd ed. Springer, 2014.

[26] Sørensen, A. J., *Marine Control Systems, Propulsion and Motion Control of Ships and Ocean Structures.* Akademika Forlag, 2013.

[27] Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M., *Diagnosis and Fault-Tolerant Control*, 3rd ed. Springer, 2016.

[28] Gertler, J. J., *Fault Detection and Diagnosis in Engineering Systems*, 2nd ed. CRC Press, 1998.

[29] Montgomery, D. C. and Runger, G. C., *Applied Statistics & Probability for Engineers*. Wiley, 2002.

[30] Skogdalen, J. E. and Smogeli, Ø., *Looking forward - reliability of safety critical control systems on offshore drilling vessels*, *Deepwater Horizon Study Group*, 2010.

[31] Johansen, T. A., Sørensen, A. J., Nordahl, O. J., Mo, O., and Fossen, T. I., *Experiences from hardware-in-the-loop (hil) testing of dynamic positioning and power management systems*, *OSV Conference Singapore*, pp. 24–25, 2007.

[32] Rausand, M., *Risk Assessment*, 1st ed. Wiley, 2011.

[33] Sklet, S., *Safety barriers: Definition, classification, and performance*, *Journal of loss prevention in the process industries*, vol. 19, no. 5, pp. 494–506, 2006.

[34] Hollnagel, E., *Barriers and accident prevention*. Routledge, 2016.

[35] Nguyen, D., *Signal processing: Signal quality checking and fault detection*, 2018.

[36] Gustafsson, F., *Adaptive Filtering and Change Detection*. Wiley, 2000.

[37] Basseville, M. and Nikiforov, I. V., *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, 1993.

[38] National Instruments, *What is LabVIEW?*, Online, 2019. Available from: http://www.ni.com/en-no/shop/labview.html.

[39] Page, E. S., *Continuous inspection scheme*, *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[40] Perez, T., Sørensen, A. J., and Blanke, M., *Marine vessel models in changing operational conditions - a tutorial*, *IFAC Proceedings Volumes*, vol. 39, 2006.

[41] ——, *Vessel operational conditions and formal specification of ship guidance navigation and control system requirements*, *Journal of Maritime Research*, 2006.

[42] Thomson, D., *Acoustic positioning systems*, Online, 2005. Available from: https://www.ths.org.uk/documents/ths.org.uk/downloads/2005-04-05-hydrofest-3_acoustics.pdf.

Appendix A

Electronic Attachments

The files in this appendix are included in electronically submitted versions.

A.1 SignalGenerator.vi

This file contains the LabVIEW program for generating random signals with the option of adding different failure modes.

A.2 SignalProcessing.vi

This file is the LabVIEW program containing all the fault detection algorithms that have been implemented and tested in this thesis. This program also includes a list of subVIs (subfunctions):

- header_limitcheck.vi
- header_iqr.vi
- header_frozen.vi
- header_highvar.vi
- header_cusum.vi
- header_dynthres.vi
- iqr.vi
- iqr_variance.vi

The upper six subVIs create headers in the output files for each method. The bottom two calculate the interquartile range values.

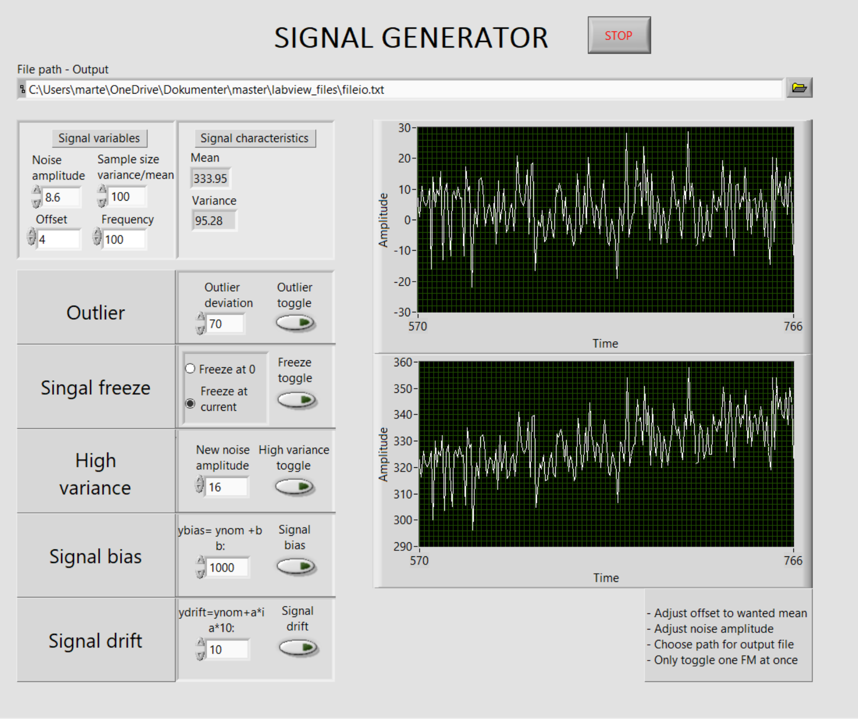
Appendix B

Signal Generator

This appendix describes the implementation of the signal generation module. This includes a brief description of the interface and how the different failure modes are integrated into the program. Lastly, it shows an example of the output file the program writes.

B.1 Front Panel

The signal generation module, displayed in Figure B.1, is designed to create valid test signals. The test signals are created by using a uniform white noise with an adjustable noise amplitude and offset as a basis. This basis represents a random signal subjected to noise. Section 2.7 defined and studied the mathematical modeling of different failure modes. These models have been implemented in the signal generator, such that each failure mode can be added to the clean signal. The right-hand side of the interface shows the original signal versus the polluted signal. The left side shows all the optional failures that can be added to the signal. It also shows the signal characteristics and lets the user change the signal variables.



SIGNAL GENERATOR

File path - Output
C:\Users\marte\OneDrive\Dokumenter\master\labview_files\fileio.txt

Signal variables
Noise amplitude: 8.6
Sample size variance/mean: 100
Offset: 4
Frequency: 100

Signal characteristics
Mean: 333.95
Variance: 95.28

Outlier
Outlier deviation: 70

Singal freeze
○ Freeze at 0
● Freeze at current

High variance
New noise amplitude: 16

Signal bias
ybias= ynom +b
b: 1000

Signal drift
ydrift=ynom+a*i
a*10: 10

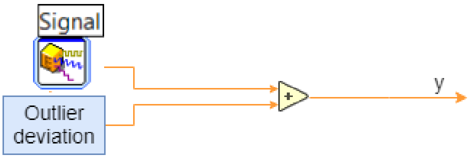
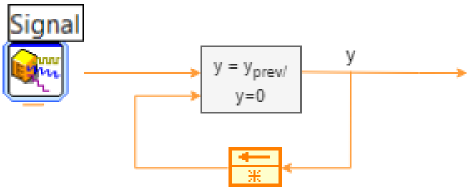
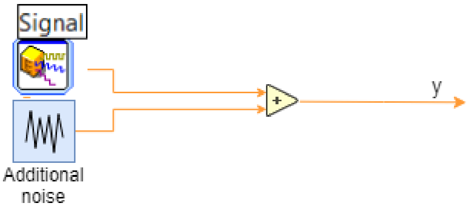
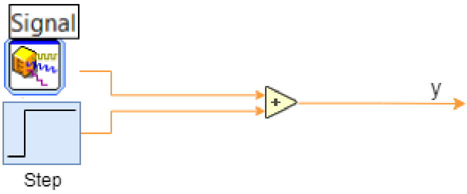
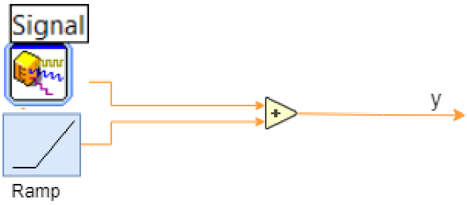
- Adjust offset to wanted mean
- Adjust noise amplitude
- Choose path for output file
- Only toggle one FM at once

Figure B.1: Front panel - Signal generation module

B.2 Failure Modes

Table B.1 shows the approach for modelling each of the studied failure modes in LabVIEW.

Table B.1: Approach for modelling each failure mode in LabVIEW

Failure mode	Modelling approach
Outliers: An outlier is simulated by adding a deviation to one sample of the signal. The magnitude of the deviation is set by the user.	
Signal freeze: A signal freeze is simulated in two different approaches, chosen by the user: 1. Forcing the signal to equal zero from the moment the switch is toggled. 2. Forcing the signal to remain at its current value from the moment the switch is toggled.	
High variance: High variance is simulated by increasing the noise amplitude as long as the switch is turned on. When the switch is toggled off, the amplitude returns to normal. The new noise amplitude can be determined by the user.	
Bias: A signal bias is simulated by adding a constant value to each of the signal samples. The magnitude of the bias is determined by the user.	
Drift: Signal drift is simulated by creating a ramp function that can be added to the signal. The slope of the ramp is determined by the user.	

B.3 Output File

While the program generates the signal, it simultaneously writes all data to an output file. This file contains both the raw random signal and the polluted signal together with its frequency, mean, and variance. The table below shows a sample of a written output file.

Table B.2: Example of an output file

Time	Raw	Processed	Mean	Variance	Frequency
0.000000	9.133980	9.133980	9.133980	0.000000	10000
0.100000	2.108496	2.008496	5.621238	12.339357	10000
0.200000	-6.135356	-6.335356	1.702374	38.941238	10000
0.300000	8.066158	7.766158	3.293320	36.799258	10000
0.400000	8.791422	8.391422	4.392940	34.276067	10000
0.500000	5.821492	5.321492	4.631032	28.846828	10000
0.600000	1.034577	0.434577	4.117253	26.309667	10000
0.700000	-1.313642	-2.013642	3.438391	26.246932	10000
0.800000	-3.731428	-4.531428	2.641744	28.407773	10000
0.900000	19.485724	18.585724	4.326142	51.101765	10000
1.000000	3.522085	2.522085	4.253046	46.509580	10000
1.100000	12.572870	11.472870	4.946365	47.921380	10000
1.200000	6.828531	5.628531	5.091147	44.486662	10000
1.300000	-2.331158	-3.631158	4.560982	44.963012	10000
1.400000	-4.294617	-5.694617	3.970609	46.845047	10000
1.500000	21.483140	19.983140	5.065142	61.887275	10000
1.600000	-1.774957	-3.374957	4.662783	60.837129	10000
1.700000	7.487888	5.787888	4.819734	57.876056	10000
1.800000	-0.679811	-2.479811	4.530284	56.338008	10000
1.900000	6.680767	4.780767	4.637808	53.740775	10000
2.000000	8.467339	6.467339	4.820167	51.846784	10000
2.100000	3.991620	1.891620	4.782505	49.519897	10000
2.200000	0.449975	-1.750025	4.594135	48.147497	10000
2.300000	11.845061	9.545061	4.896256	48.240738	10000
2.400000	6.758925	4.358925	4.970763	46.444338	10000
2.500000	-8.605827	-11.105827	4.448587	51.474726	10000
2.600000	4.900170	2.300170	4.465312	49.575528	10000
2.700000	2.780958	0.080958	4.405156	47.902678	10000
2.800000	1.255265	-1.544735	4.296540	46.581195	10000
2.900000	5.278976	2.378976	4.329287	45.059589	10000
3.000000	3.721375	0.721375	4.309677	43.617591	10000
⋮	⋮	⋮	⋮	⋮	

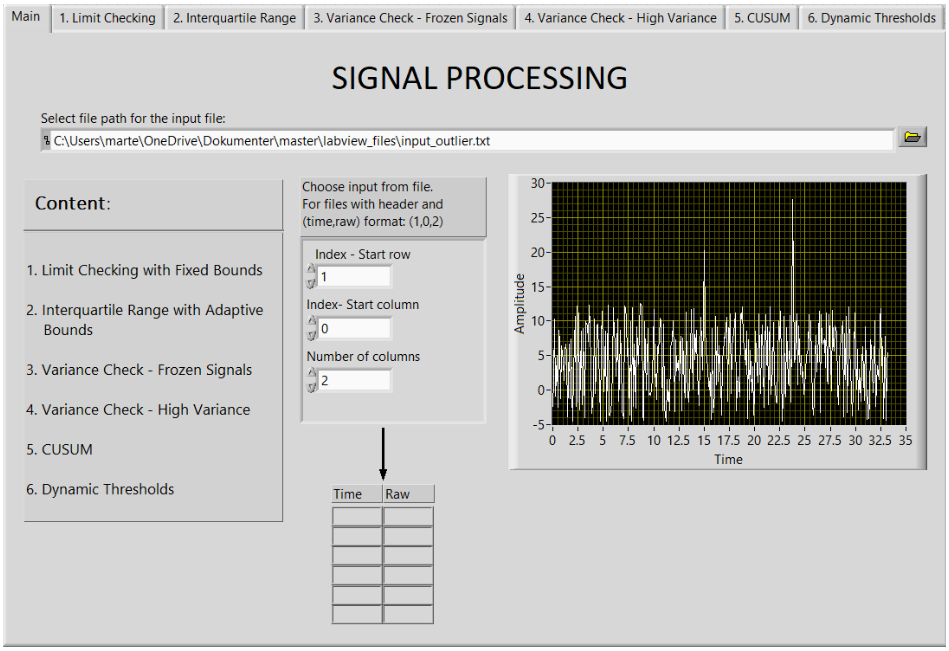
Appendix C

Implementation of Fault Detection Methods

This appendix contains a brief description of how each studied method was implemented. With each description, a picture of the corresponding front panel is attached.

C.1 Main

Figure C.1 shows the main page of the final fault detection program. This page shows an overview of the content of the program, including all the implemented algorithms. On this page, the user must determine the file path for the input file and establish the rows and columns where the relevant signal data is located. The plot to the right shows the data accessed from the input file. By selecting one of the tabs at the top of the panel, the various methods can be tested on the chosen input file.



SIGNAL PROCESSING

Select file path for the input file:
C:\Users\marte\OneDrive\Dokumenter\master\labview_files\input_outlier.txt

Content:

1. Limit Checking with Fixed Bounds
2. Interquartile Range with Adaptive Bounds
3. Variance Check - Frozen Signals
4. Variance Check - High Variance
5. CUSUM
6. Dynamic Thresholds

Choose input from file.
For files with header and (time,raw) format: (1,0,2)

Index - Start row: 1

Index- Start column: 0

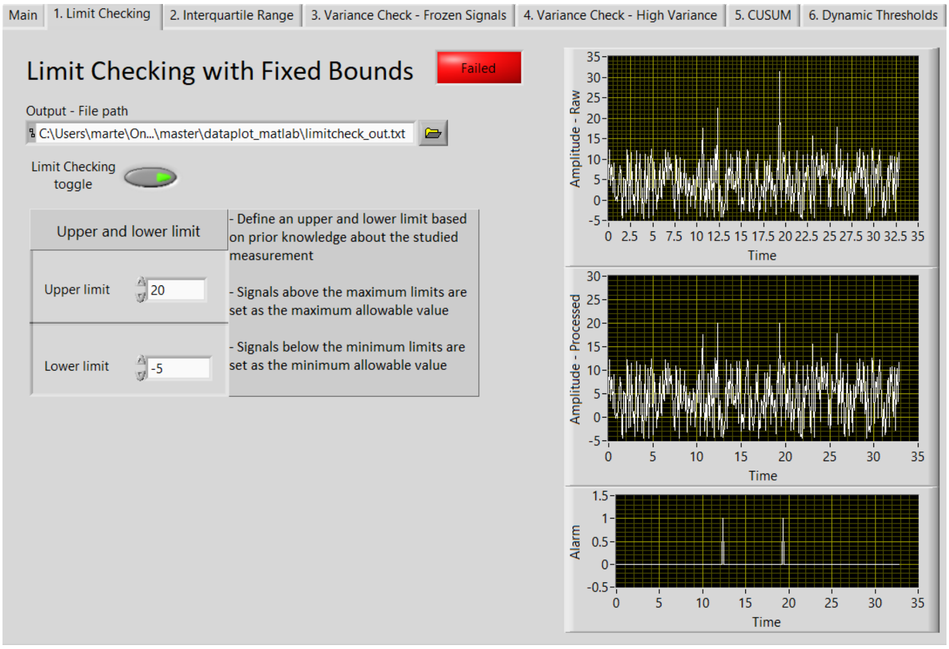
Number of columns: 2

Time	Raw

Figure C.1: Front panel - Main

C.2 Limit Checking

The front panel for the limit checking algorithm is shown in Figure C.2. The upper plot shows the input signal, while the middle plot shows the output after using limit checking. The algorithm was implemented such that the upper and lower bounds have to be set manually and should be adjusted to fit the data set. The algorithm goes through each of the data samples and compare them to the predefined upper and lower bound. If the current value is outside the allowed interval, an alarm is triggered, and the value is set as the maximum or minimum allowable value. The bottom plot shows the alarm as a function of time. When a fault is detected, the value is equal to 1, while it remains 0 as long as the signal is fault-free. After running a test, the LED next to the title shows whether or not the data set passed the test. If it is green, it means that every single value in the data set was within the allowed interval.



Limit Checking with Fixed Bounds

Output - File path: C:\Users\marte\On...\master\dataplot_matlab\limitcheck_out.txt

Limit Checking toggle

Upper and lower limit

Upper limit: 20

Lower limit: -5

- Define an upper and lower limit based on prior knowledge about the studied measurement

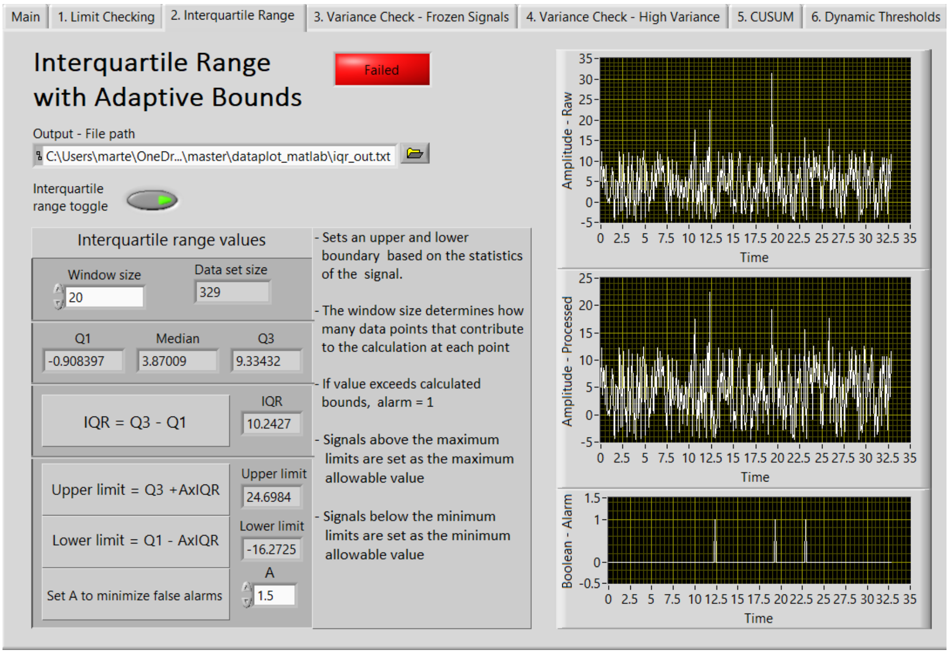
- Signals above the maximum limits are set as the maximum allowable value

- Signals below the minimum limits are set as the minimum allowable value

Figure C.2: Front panel - Limit checking

C.3 Interquartile Range

Figure C.3 shows the resulting front panel from implementing the method of interquartile range. The three plots on the right show the original signal, the processed signal, and the alarm function, respectively. The window size is adjustable and determines how many samples that are included when calculating the limits for each signal value. The total number of samples is displayed to make it easier for the user to choose a window size. All important values, such as the quartiles and the final limits, are displayed in the panel. One modification has been done compared to the original algorithm. The tolerance of the upper and lower bounds can be adjusted by changing the variable A, which in the original algorithm is set as a constant equal to 1.5.



Interquartile Range with Adaptive Bounds

Output - File path: C:\Users\marte\OneDr...\master\dataplot_matlab\iqr_out.txt

Interquartile range toggle

Interquartile range values

Window size	Data set size
20	329

Q1	Median	Q3
-0.908397	3.87009	9.33432

IQR = Q3 - Q1 — IQR: 10.2427

Upper limit = Q3 +AxIQR — Upper limit: 24.6984

Lower limit = Q1 - AxIQR — Lower limit: -16.2725

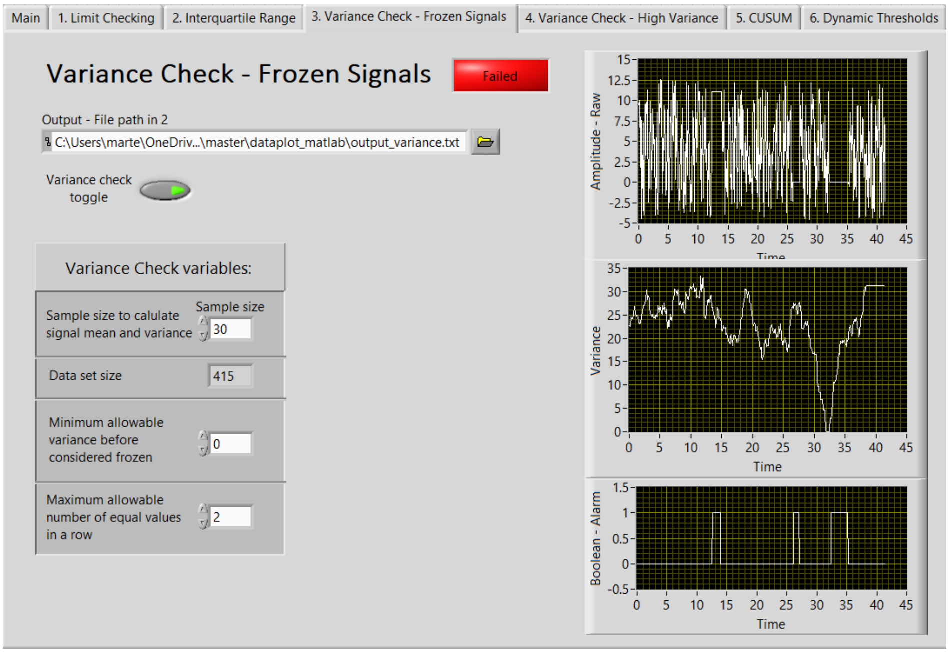
Set A to minimize false alarms — A: 1.5

- Sets an upper and lower boundary based on the statistics of the signal.
- The window size determines how many data points that contribute to the calculation at each point
- If value exceeds calculated bounds, alarm = 1
- Signals above the maximum limits are set as the maximum allowable value
- Signals below the minimum limits are set as the minimum allowable value

Figure C.3: Front panel - Interquartile range

C.4 Variance Check: Frozen Signals

The implemented front panel to detect frozen signals is shown in Figure C.4. The interface has three adjustable values: The sample size for calculating the variance, the minimum acceptable variance before it is considered frozen, and the maximum allowable number of equal consecutive measurements. The three plots show the input signal, the variance, and the alarm function, respectively.



Variance Check - Frozen Signals

Failed

Output - File path in 2: C:\Users\marte\OneDriv...\master\dataplot_matlab\output_variance.txt

Variance check toggle

Variance Check variables:

Sample size to calulate signal mean and variance — Sample size: 30

Data set size: 415

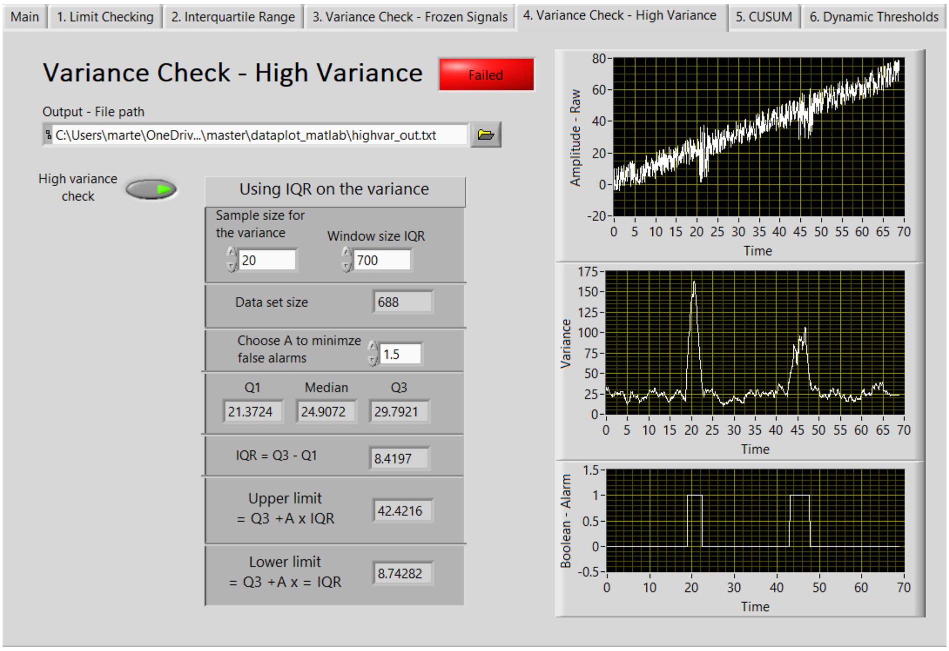
Minimum allowable variance before considered frozen: 0

Maximum allowable number of equal values in a row: 2

Figure C.4: Front panel - Frozen signals

C.5 Variance Check: High Variance

The resulting front panel to detect high variance is shown in Figure C.5. The algorithm has three adjustable variables, namely the IQR window size, the variance sample size, and the variable determining the tolerance of the limits. The option to modify the window size and sample size was added to be able to test how the result is affected by the amount of data available for the calculations.



Variance Check - High Variance

Output - File path
C:\Users\marte\OneDriv...\master\dataplot_matlab\highvar_out.txt

High variance check

Using IQR on the variance

Sample size for the variance: 20

Window size IQR: 700

Data set size: 688

Choose A to minimze false alarms: 1.5

Q1	Median	Q3
21.3724	24.9072	29.7921

IQR = Q3 - Q1: 8.4197

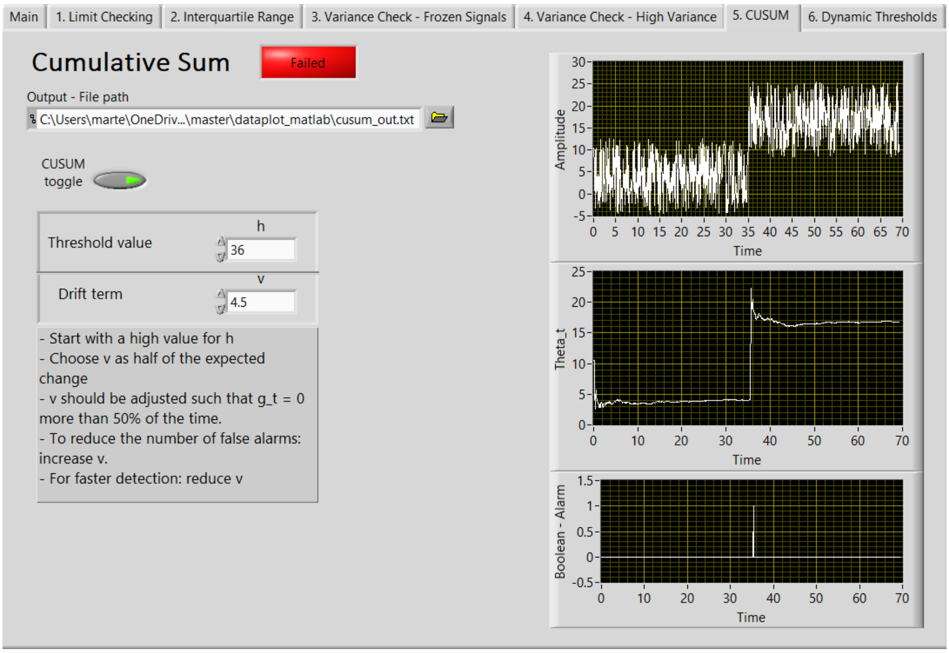
Upper limit = Q3 +A x IQR: 42.4216

Lower limit = Q3 +A x = IQR: 8.74282

Figure C.5: Front panel - High variance

C.6 Cumulative Sum

Figure C.6 shows the front panel of the implemented CUSUM algorithm. The two variables h and ν, corresponding to the threshold value and the drift term, can be set by the user. Some simple steps to tune these variables are included in the panel. The upper plot shows the input signal, the middle plot shows the signal estimate θ_t, and the bottom plot shows the alarm function.



Cumulative Sum

Output - File path

C:\Users\marte\OneDriv...\master\dataplot_matlab\cusum_out.txt

CUSUM toggle

Threshold value h: 36

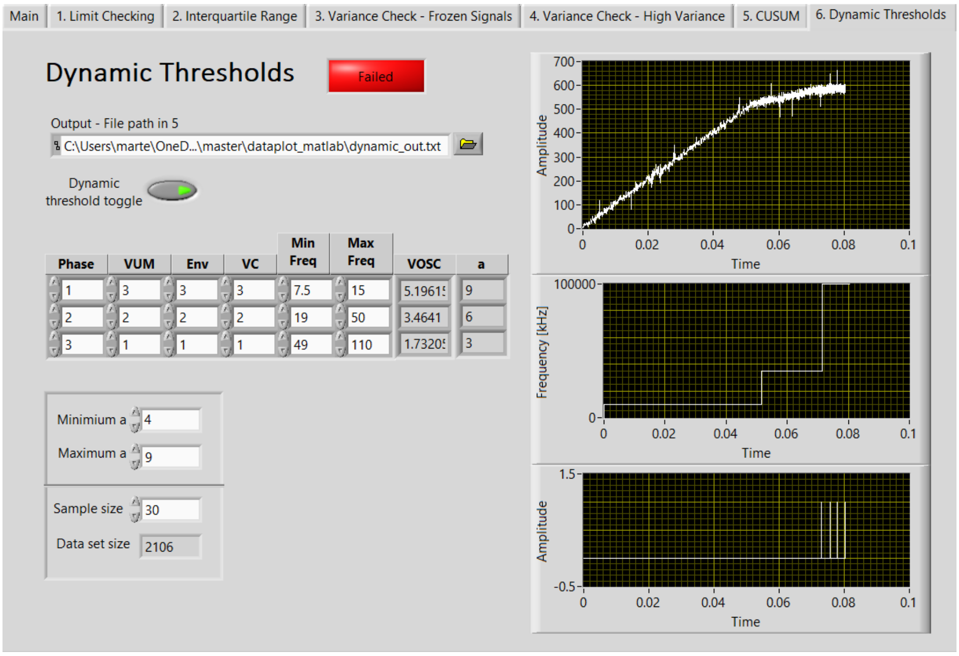
Drift term v: 4.5

- Start with a high value for h
- Choose v as half of the expected change
- v should be adjusted such that g_t = 0 more than 50% of the time.
- To reduce the number of false alarms: increase v.
- For faster detection: reduce v

Figure C.6: Front panel - CUSUM

C.7 Dynamic Thresholds

The resulting front panel of the implemented algorithm, displayed in Figure C.7, shows all relevant values together with plots of the input signal, frequency change, and the alarm function. The algorithm is implemented for a case with three separate phases but can easily be modified to include more. The interface requires the user to determine all necessary variables, represented as a matrix in the front panel. For each phase, values for *VUM*, *Env* and *VC* has to be determined, based on the relevant operations. Each phase must also be assigned a frequency band. A minimum and maximum *a*-value must be set to determine the final thresholds. This value determines the tolerance of the final limits. A more detailed explanation of these variables is given in Chapter 4. After running the algorithm, the interface will show the calculated $VOSC_{gain}$ for each phase together with its corresponding *a*-value.



Dynamic Thresholds

Output - File path in 5
C:\Users\marte\OneD...\master\dataplot_matlab\dynamic_out.txt

Dynamic threshold toggle

				Min Freq	Max Freq		
Phase	VUM	Env	VC			VOSC	a
1	3	3	3	7.5	15	5.19615	9
2	2	2	2	19	50	3.4641	6
3	1	1	1	49	110	1.73205	3

Minimium a 4
Maximum a 9

Sample size 30
Data set size 2106

Figure C.7: Front panel - Dynamic thresholds