Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Gianluca Tabella

Subsea Risk Management Based on Sensor Networks

Master's thesis in Reliability, Availability, Maintainability and Safety
Supervisor: Nicola Paltrinieri
August 2019

NTNU
Norwegian University of
Science and Technology

Abstract

This thesis consists of the evaluation of subsea risk management against oil spills based on the analysis of information received from sensors in an underwater distributed sensor network. The work starts by highlighting the importance of having a performing leak detection system either from an environmental or safety point of view. The Goliat FPSO is considered as a case study. This FPSO is located in the Barents Sea, which is a sensitive environment and has to meet requirements dictated by Norwegian authorities in order to prevent oil spills. An innovative technology is used on this platform to detect possible subsea oil spills: the use of passive acoustic sensors. A sensor network like the one present here is composed of different sensors which transmit information (a local decision) to a fusion center which takes a global decision on whether the leakage is occurring or not. This work will evaluate how the choice of different fusion rules (Counting Rule and Weighted Fusion Rule adapted for this work) can affect the performances of the leak detection system in its current configuration. Also, it will be discussed how different thresholds, selected for a specific fusion rule or sensor test, can change the final performance from a detection point of view. Methods used for detection are based on statistical signal processing and decision theory, often exploiting methodologies already in use in other fields (telecommunication engineering, medical sciences, military sciences), which have to be adapted to fit this application within the Oil&Gas industry. A further step is to develop a method able to localize the leakage point in a subsea template. This work proposes some methods which could be useful in order to localize the equipment responsible for the leakage. These proposed methods for leak localization are developed so that they can be coupled with the proposed methods for leak detection, giving a coherent set of operations that the sensors and the fusion center must perform. Performances of detection techniques are assessed balancing the need of having higher values of parameters like the True Positive Rate and the Precision, maintaining low values of False Positive Rate. Whereas, performances of localization techniques will be assessed according to their ability to localize leakage points in the shortest amount of time possible; if this is not possible, other parameters like the difference between the estimated position and the real leakage position will be considered. Some other simulations are carried out to test the proposed local and global thresholds coupled with specific fusion rules for detection and localization of the leakage. Performances of the different configurations will be graded according to global indexes necessary to gather together the above-mentioned detection performance parameters. These indexes can be based on either the ROC curve (like the Youden's Index) or the PR curve (like the F-scores).

Sammendrag

Denne masteroppgaven omhandler evalueringen av undervannsrisikostyring for å forhindre oljeutslipp basert på analyse av informasjon mottatt fra sensorer i et underwater distributed sensor network. Oppgaven begynner med en fremheving av viktigheten av et velfungerende lekkasjedeteksjonssystem både fra et miljø- og et sikkerhetsperspektiv. Goliat FPSO betraktes som en casestudie. Gjeldende FPSO er plassert i Barentshavet, hvilket er et sensitivt område og må oppfylle krav diktert av norske myndigheter for å forhindre oljeutslipp. En innovativ teknologi benyttes på denne plattformen for å detektere mulig oljeutslipp under vann: bruken av passive akustiske sensorer. Et sensornettverk slik som dette består av ulike sensorer som sender informasjon (lokal beslutning) til et fusjonssenter som tar en global beslutning vedrørende om lekkasjen pågår eller ikke. Denne oppgaven vil evaluere hvordan valget av ulike fusjonsregler (Counting Rule og Weighted Fusion Rule tilpasset denne oppgaven) kan påvirke yteevnen til lekkasjedeteksjonssystemet i dets gjeldende konfigurasjon. Det vil også bli diskutert hvordan forskjellige terskler, valgt for en spesifikk fusjonsregel eller sensortest, kan endre den endelige yteevnen sett fra et deteksjonsperspektiv. Deteksjonsmetoder baseres på statistisk signalprosessering og beslutningsteori, ofte ved utnyttelse av allerede eksisterende metoder brukt i andre felt (telekommunikasjonsteknologi, medisin, krigsvitenskap), som må tilpasses til dette bruksområdet innen olje -og gassindustrien. Et steg videre er å utvikle en metode som kan lokalisere lekkasjepunktet i en havbunnsramme. Denne oppgaven foreslår noen metoder som kan være nyttige for å lokalisere utstyret ansvarlig for lekkasjen. Disse foreslåtte metodene for lekkasjelokalisering er utviklet så de kan jobbe sammen med de foreslåtte metodene for lekkasjedeteksjon, hvilket vil gi et koherent sett av operasjoner som sensorene og fusjonssenteret må utføre. Yteevnen til deteksjonsteknikkene bestemmes ut ifra en balanse mellom behovet for høyere verdier av parametere som Sanne Positive Rate og Presisjon, og å beholde lave verdier av Falske Positive Rate. Yteevnen til lokaliseringsteknikkene vil bli evaluert ut ifra deres evne til å lokalisere lekkasjepunkter i løpet av kortest mulig tid. Hvis dette ikke er mulig vil andre parametere tas i betraktning, som for eksempel differansen mellom estimert posisjon og faktisk lekkasjeposisjon. Noen flere simuleringer utføres for å teste de foreslåtte lokale og globale tersklene brukt sammen med spesifikke fusjonsregler for deteksjon og lokalisering av lekkasjen. Yteevnen til de ulike konfigurasjonene vil bli rangert i henhold til globale indekser nødvendige for å samle ovennevnte deteksjonsevneparametere. Disse indeksene kan baseres enten på ROC-kurven (som Youdens indeks) eller på PR-kurven (som F-mål).

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

$\|\cdot\|$	Euclidean norm operator
~	Distributed as…

AAF	Amplitude Attenuation Function
AIS	Automatic Identification System
ALARP	As Low As Reasonably Possible
AUC	Area Under the Curve
AWGN	Additive White Gaussian Noise
BAT	Best Available Technique
BBM	Barycentric-Based Method
BMC	Binary Markov Chain
c	Sound speed
CAPEX	Capital Expenditure
CBA	Cost-Benefit Analysis
CBM	Centroid-Based Method
ccdf	Complementary cumulative distribution function
CR	Counting Rule
CSI	Channel State Information
CZ	Concordance Probability Objective Function
D	Depth
d^2	Squared Closest-to-(0,1)
DRMF	Dynamic Risk Management Framework
DtF	Decode-than-fuse
ET	Energy test
f	Sound frequency
FC	Fusion Center
FCM	Flow Control Module
FEED	Front-End Engineering Design
FLIR	Forward-Looking Infrared
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FPSO	Floating Production Storage and Offloading Unit
FR	Fusion Rule
GIS	Geographic Information Systems
HC	Hydrocarbon
HCM	HIPPS Control Module
HIPPS	High-Intensity Pressure Protection System
HLD	Hydrocarbon Leak Detector
HS	Hotspot
J	Youden's Index
JIP	Joint Industry Project
LDS	Leak Detection System
LLR	Log-Likelihood Ratio
$\ln(\cdot)$	Natural logarithm
LOC	Loss of Containment
$\log(\cdot)$	Common logarithm
LR	Likelihood Ratio
$N(\mu, \sigma^2)$	Gaussian pdf with mean μ and variance σ^2
NCS	Norwegian Continental Shelf
NCT	Non-cooperative target
NDIR	Nondispersive Infrared

OPEX	Operating Expense
OSD	Oil Spill Detection
P	Pressure
$P(\cdot \mid \cdot)$	Conditional pmf
$p(\cdot \mid \cdot)$	Conditional pdf
P_d	Probability of Detection
pdf	Probability density function
P_f	Probability of False Alarm
pmf	Probability mass function
PPV	Positive Predicted Value
PT	Pressure Transducer
PTT	Pressure-Temperature Transducer
$Q(\cdot)$	Q-function
Q_d	Global probability of detection
Q_f	Global probability of false alarm
QRA	Quantitative Risk Assessment
ROC	Receiver Operating Characteristic
ROV	Remotely Operated Vehicle
S	Salinity
SALD	Single Acoustic Leak Detector
SAR	Synthetic-Aperture radar
SCM	Subsea Control Module
SLAR	Side-Looking Airborne Radar
SNR	Signal-to-noise ratio
T	Temperature
TISP	Target Inside Sensors' Perimeter
TL	Transmission Loss
TN	True Negative
TNR	True Negative Rate
TOSP	Target Outside Sensors' Perimeter
TP	True Positive
TPR	True Negative Rate
TT	Temperature Transducer
UAWSN	Underwater Acoustic Wireless Sensor Network
UMP	Uniformly Most Powerful
WFR	Weighted Fusion Rule
WSN	Wireless Sensor Network
x_k	The kth component of the vector $\underline{x}$
XMT	Christmas Tree
$\underline{x_S}$	Sensor position
$\underline{x_T}$	Target position

1 Introduction

Over the years, the Oil&Gas industry has increased its interest in the exploitation of offshore reservoirs making the offshore production around one-third of the global crude oil output (U.S. Energy Information Administration, 2016). This interest comes from the need for unlocking new reserves as a consequence of the high demand for crude oil products which cannot be satisfied by only exploiting onshore (or shallow offshore) fields. There are several advantages related to the offshore O&G production (Shafer et al., 2013; Speight, 2014):

- Governments see an increase in revenues due to royalties;
- Offshore drilling and production provide jobs and stimulate the local economy;
- Less dependence on foreign oil;
- Oil market price may reduce as a consequence of domestic production.

Going offshore surely has some main disadvantages:

- The cost of an offshore project is much higher compared to onshore (A. Rasmussen and J. Piette, 1984);
- It is challenging from an engineering and management point of view (Speight, 2014);
- A loss of containment may result in severe damage to the environment (Hong and Yanjie, 2009).

The last issue has become increasingly important over time especially after events like the Deepwater Horizon oil spill (Beyer et al., 2016; Girard and Fisher, 2018; Hester et al., 2016).

Over the last years, a trend to explore areas which are more sensitive from an environmental point of view has developed. It is the case of the Barents Sea whose undiscovered resources have been estimated to be around 1165 million standard cubic meters of oil equivalent in Barents Sea South and 1370 million standard cubic meters of oil equivalent in Barents Sea North (Norwegian Petroleum Directorate, 2018). The concern about hydrocarbon exploration and exploitation in this area is due to the extreme value of its eco-system, also recognized by the WWF (WWF, 2004). Some characteristics of this area have been identified as:

- Naturalness: the lack of disturbance or degradation caused by human activities made species and biotopes be still in a very natural state;
- Representativity: some areas contain several species, ecological processes or other natural characteristics that are representative for the whole ecoregion or their specific sub-region;
- High natural biological diversity: some areas contain a naturally wide variety of species or include highly varied habitats and communities;
- Productivity: some area has naturally high productivity of species or features represented, contributing to sustain species or the ecosystem;
- Ecological significance for species: some areas are essential during fundamental phases of life of some species (breeding areas, nursery areas, feeding areas, resting areas, etc.);

- Dependency: some species depend on specific processes occurring in some specific areas;
- Presence of Source Areas: these areas contribute to the maintenance of essential processes or systems;
- Uniqueness: some areas present unique characteristics like the presence of endemic, rare or endangered species or because of outstanding ecological or evolutionary phenomena;
- Sensitivity: some areas contain a proportion of sensitive (or very sensitive) habitats or species.

This means this area could be severely damaged in case of a major oil spill (Hasselström et al., 2012; Loeng and Drinkwater, 2007). Oil production from fields located in the Barents Sea, in fact, faces the opposition of part of the Norwegian population as reported by newspapers and surveys (Milne, 2017; Wijnen, 2017).

Currently, there are only two producing fields in the Norwegian Barents Sea: Snøhvit (natural gas) and Goliat (containing both crude oil and natural gas) which started their production phase in 2007 and 2016, respectively (Norwegian Petroleum Directorate, 2019a). Figure 1.1 shows a map of their positions together with the new discoveries in that region:

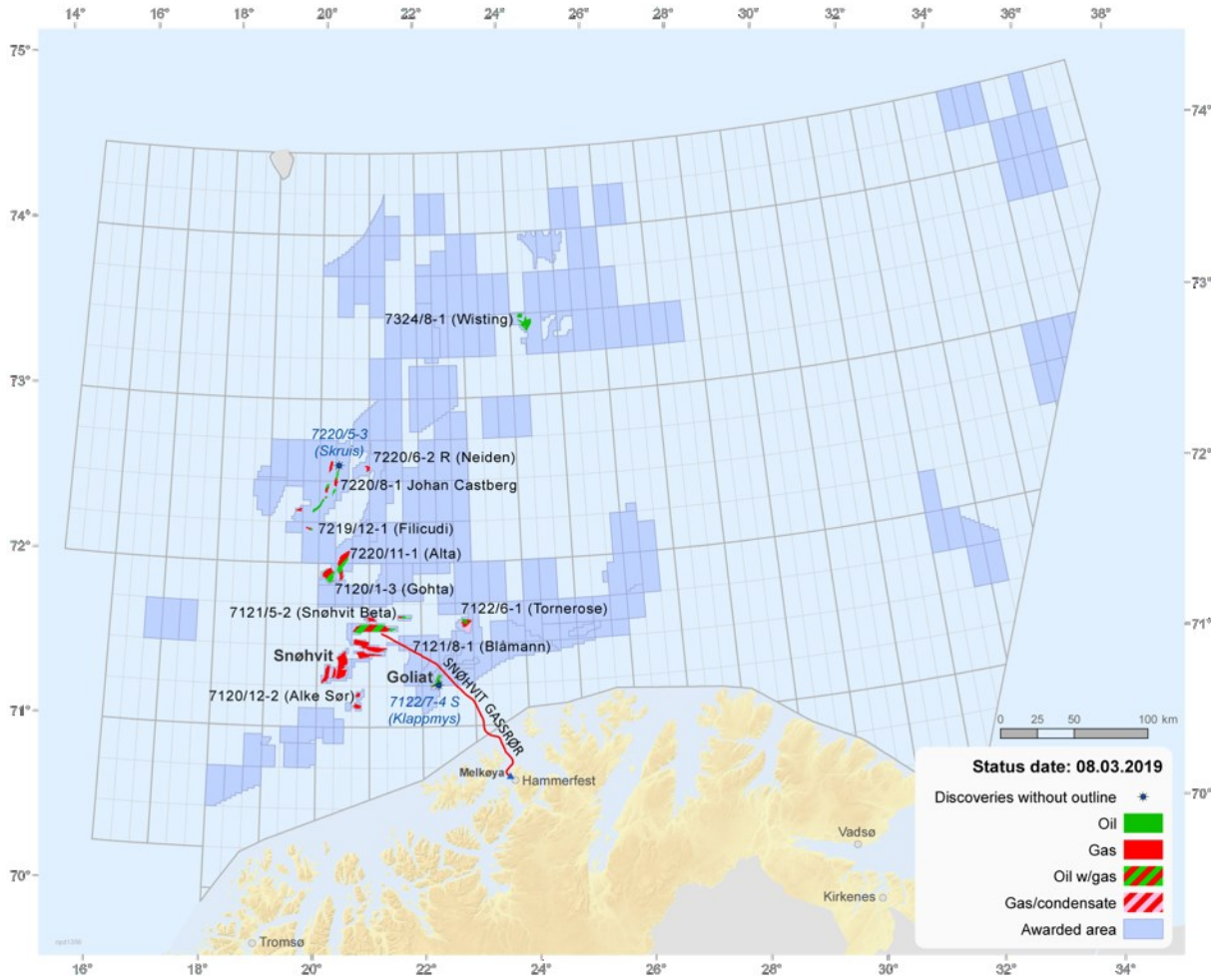


Figure 1.1 Fields and discoveries in the Barents Sea (Norwegian Petroleum Directorate, 2019a)

Companies operating in the Norwegian Continental Shelf (NCS) are required to perform continuous environmental monitoring during petroleum activities. This can be read from the guidelines of the Norwegian Environmental Agency, which summarize the duty of oil companies operating in the NCS concerning environmental monitoring (Norwegian Environment Agency, 2016). These guidelines make explicit reference to Section 49 of the Pollution Control Act which states that

"On orders from the pollution control authority, any person that possesses, does, or initiates anything that may generate pollution or result in waste problems has a duty, notwithstanding any duty of secrecy, to provide the pollution control authority or other public bodies with any information necessary to enable them to carry out their tasks pursuant to this Act […]" (Ministry of Climate and Environment, 1981).

Also, guidelines refer to Section 51 which states that

"The pollution control authority may order any person that possesses, does, or initiates anything that results in or that there is reason to believe may result in pollution to arrange or pay for any investigations or similar measures that may reasonably be required in order to: determine whether and to what extent the activity results in or may result in pollution, ascertain the cause of or impact of pollution that has occurred, ascertain how the pollution is to be combated. […]" (Ministry of Climate and Environment, 1981).

References are also made in Chapter 10 of "Regulations relating to conducting petroleum activities (the activities regulations)" (Petroleum Safety Authority Norway et al., 2016), which is about monitoring the external environment, and Section 34 of "Regulations relating to management and the duty to provide information in the petroleum activities and at certain onshore facilities (the management regulations)" (Petroleum Safety Authority Norway et al., 2017), which is about reporting information on monitoring, emission, discharges, and risk of pollution-related to offshore petroleum activities.

There is an explicit reference in Section 57 of "The Activities Regulations" about the detection of oil spills:

"Operators shall establish remote sensing systems to detect and map the position, area, quantity, and properties of acute pollution. The remote sensing system shall as independent of visibility, light and weather conditions as possible, provide sufficient information to ensure that acute pollution from the activity is detected and mapped as quickly as possible. Leak detection based on process monitoring, monitoring of the water column and benthic habitats, among other things, shall be assessed as part of the remote sensing system. […]" (Petroleum Safety Authority Norway et al., 2016).

It is clear how environmental monitoring is a necessary action required by the Norwegian Law in order to avoid oil spills. A Joint Industry Project (JIP) has been put in place by DNV-GL in collaboration with other industries: Biota Guard, BP, Contros, Eni Norge, Engie, FMC Technologies, ICD Industries, Kongsberg Maritime, KSAT, Lundin, Metas, Miros, Naxys, Norbit Subsea, Petrobras, Phaze Technologies, Sonardyne, Stinger, Vissim and others three observers (Norwegian Oil & Gas, Norwegian Ministry of Climate and Environment and the Petroleum Safety Authority Norway). This project was born to develop a best practice for designing and implementing offshore Leak Detection Systems (LDS) (DNV-GL, 2014). This JIP led to the development of a set of recommendations regarding offshore leak detection (DNV-GL, 2016), which is an update of the previously published set of recommendations (DNV-GL, 2010). The newer version recommends using the Best Available Technique (BAT) when it comes to the selection of the Leak

Detection System. Especially, Section 3.5 identifies critical parameters to improve in order to satisfy high-level functional requirements:

- Minimum leakage rate or volume to be detected;
- Ability to locate leakage source;
- Ability to declare the extent of the leaked fluid;
- Maximum detection range;
- Detection time for minimum leakage within the specified detection range;
- Ability to detect the type of fluid and its concentration;
- Ability to classify the leaking fluid;
- Availability of the detector.

Goliat field is the only field producing crude oil in the Barents Sea. A Subsea Production System (SPS) is used to exploit this field where different technologies have been combined to form the LDS. One technology will be taken into consideration and analyzed in this work: the use of passive acoustic sensors. The present work analyzes the behavior of this typology of sensors when it comes to detection and localization of leakage on a subsea template. The analysis will focus on how to manage the information transmitted by the sensors in order to increase performances; this will be done by testing different methods and developing new ones. For each method (for either detection or localization), different configurations and settings will be examined.

In particular:

- Chapter 2 is an overview of the oil spill risk from a management point of view, and it describes the state-of-the-art of subsea leak detection systems.
- Chapter 3 gives an introduction on the Goliat field, especially on oil spill preparedness, with a focus on the subsea oil detection system used to monitor the marine environment.
- Chapter 4 is a theoretical chapter whose aim is to explain how the underwater acoustic signal was modeled and the existing methods for an efficient signal processing in case of a possible leak detection either at single sensor level (local) or sensor network level (global).
- Chapter 5 proposes a method for tuning the passive acoustic sensors placed on the templates present in the Goliat field. Where tuning, in this case, means to fix the main input parameters needed for ensuring good performances. It also proposes two methodologies for fixing the main parameters necessary when fusing all the information received from the sensors. In this chapter, some new leak localization techniques are proposed which, together with detection techniques, can be used by an algorithm that either sensors or the fusion center must perform during operating conditions.
- Chapter 6 describes the simulations that were carried out by exploiting the results and the algorithms from Chapter 5. Results are shown too.
- Chapter 7 discusses the results present in Chapter 6.
- Chapter 8 is dedicated to the conclusions, and it addresses possible further works that could be carried out to complete and improve the results of this study.

2 Subsea Risk Management

2.1 Dynamic Risk Management

Subsea Risk Management can be used to identify hazardous scenarios and model their development. Leakage is not a concern only from an environmental point of view, but also from a safety perspective. It is known that many accidents in the chemical and process industry are caused by LOC (loss of containment), so it is for the offshore industry, either topside or subsea. The probabilities and the consequences of these events to happen can be lowered by paying more attention to early warnings (Konstantinidou et al., 2012). These early warnings can be either small leaks or near-misses. This can be done applying the Dynamic Risk Management Framework (DRMF), which is represented in Figure 2.1:

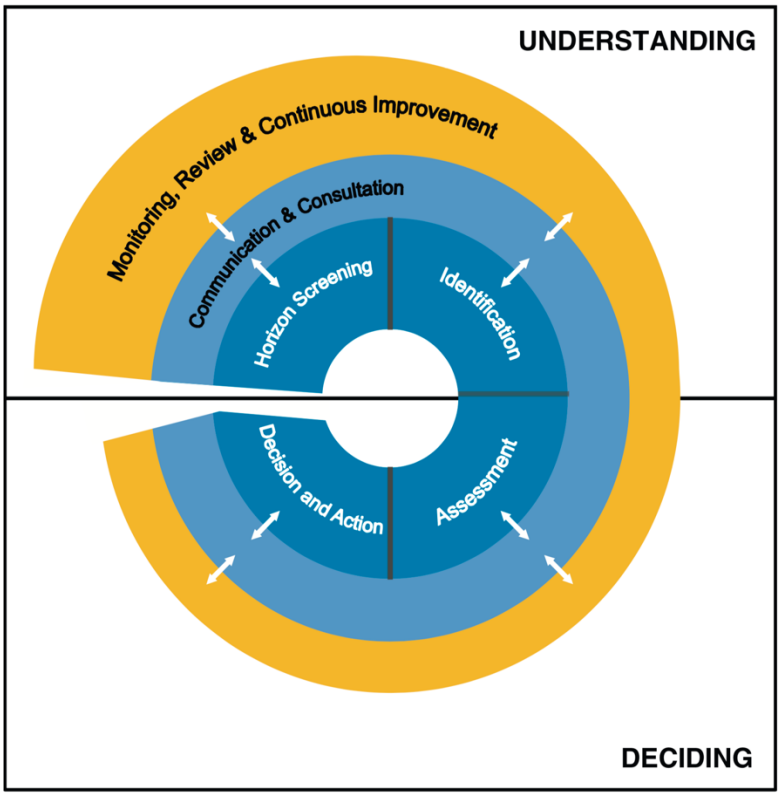


Figure 2.1 Dynamic Risk Management Framework (clockwise) (Paltrinieri et al., 2014)

Two general phases are present: Understanding and Deciding. Two sequential sub-phases compose each phase:

- Understanding: it is the phase regarding knowledge and information management. It is divided in Horizon Screening and Identification: in the first sub-phase risk issues are framed, and the system is studied and analyzed; the second one refers to the identification of hazard related to the process, the equipment, and the substances used, in order to define the potential accident scenarios.
- Deciding: it is the phase where the elaboration and judgment of produced information occur, and intervention is carried out. More specifically, Assessment is

the sub-phase where the previously identified scenarios are evaluated, and risk is estimated (Dynamic Risk Assessment can be used). The value of risk is later compared to thresholds dictated by risk matrices or other criteria. Decision and Action is the decision-making process and the consequent implementation of actions for non-acceptable risks.

Monitoring, Review & Continuous Improvement, together with Communication & Consultation, are two tasks that must be continuously carried out. The first one is particularly emphasized during the Understanding phase.

The DRMF is a continuous and circular process and needs regular reviews and updates to use as evidence. Moreover, it is open to new external experience and early warnings.

It is clear how the DRMF gets more effective as the knowledge of the system is higher (which is part of the Horizon Screening phase). Knowledge, in fact, can influence risk estimation. Aven and Krohn proposed the level of knowledge as a new variable in the definition of risk (Aven and Krohn, 2014):

$$R = f(s, p, c, k)$$

Where R is the risk estimation, s is the scenario, p is the probability of occurrence, and k is the level of knowledge. This is a change from the traditional definition of risk proposed by Kaplan and Garrick, often applied in the Quantitative Risk Assessments (QRA), which does not consider the level of knowledge (Kaplan and Garrick, 1981):

$$R = f(s, p, c)$$

It is now clear the importance of a reliable subsea leak detection system within the DRMF as part of the continuous monitoring phase. In fact, any early warning (in the form of a small oil spill) will contribute to increasing the level of knowledge which will directly affect the risk estimation (since new evidence is provided when these minor events occur), so that more effective actions can be taken.

2.2 State-of-the-art of Subsea Leak Detection

This section focuses on the external sensors that are used to monitor the external environment of the template (internal LDSs are only introduced).

The currently available technologies are the following (DNV-GL, 2016):

- Active acoustic sensors;
- Biosensors;
- Capacitive sensors;
- Fiber optic;
- Fluorescent sensors;
- Internal LDS / mass balance;
- Volumetric collection;
- Methane sniffers:
    - Semi-conductor,
    - Optical nondispersive infrared (NDIR),
    - Laser absorptiometry;
- Optical cameras;
- Passive acoustic sensors;
- Combination of multiple technologies can be implemented.

Table 2.1 describes the abovementioned techniques.

	Principle	HC	Coverage	Localiz.	Limitations
Active Acoustic Sensor	It is based on the same principles as sonars. An acoustic signal is emitted and is reflected between different media.	All	Local (single sensor) - Area (sensor network)	No (single sensor) - Yes (sensor network)	Sensitive to shadowing and background noise.
Biosensor	It uses a living organism. Biosensor response is measured by monitoring heart activity and frequency of opening/closing the clam.	Oil	Local	No	Seawater currents and buoyancy may move leaking oil away from the sensor. Additional sensors are needed.
Capacitive Sensor	It measures the change in the dielectric constant of the medium around the sensor.	All	Local	No	Biological growth can affect performances that are also depending upon the size and shape of the collector. If polymerized material is used, it can absorb water and sensitivity can be affected.
Fiber Optic Cable	It can be based on either temperature or acoustics measurement.	All	Area of installation	Yes	Fiber optic cable has a limited bend radius.
Fluorescent	It uses a light source of a specific wavelength for exciting molecules in the target fluid to a higher energy level. The molecules then relax to a lower state and light is emitted at a different wavelength which can be sensed by a detector.	Oil	Local	No	Marine growth can affect performances. Medium to be detected must naturally fluoresce, or a marker must be added.

Internal LDS/Mass Balance	It monitors pressure and flow rate using instrumentation installed in the SPS. Measured values are compared to expected pressure and flow given no leaks. The significant deviation between measured and predicted values indicate a possible leak in the system.	All	Area of installation	No, but some estimation can be carried out (Geiger, 2006)	Inaccurate when the production is unstable. Not able to detect small leaks.
Volumetric Collection	Leak detection is based on volumetric measurements. When a predetermined volume is collected, an action is initiated in the system and will give an alarm.	All	Local	Yes	Marine growth can affect performances. Trawls from fishing activity can be an issue.
Methane Sniffer	Dissolved methane diffuses over a membrane and into a sensor chamber. Three measurement principles can be used: Semi-conductor, Optical NDIR, and Laser absorptiometry.	All	Local	No	Quantification of the leak is difficult. Measurement depends on diffusion towards the sensor and seawater currents. The latter may move the leaking medium away from the sensor.
Optical Camera	It uses a video camera for the surveillance of the SPS.	All	Local	Yes	Line of sight sensor, depending on lightning. It is sensitive to marine growth, water turbidity, and pollution.
Passive Acoustic Sensor	A hydrophone listens for sounds resulting from leakage.	All	Area	Yes	Background noise can limit the sensitivity of the sensor.
Combination of technologies	Combination of the listed techniques.	It depends on which technologies have been combined.			It requires higher power consumption and increased data. It may result in additional complexity relating to the subsea control system.

Table 2.1 Summary of existing techniques for subsea leak detection (DNV-GL, 2016)

2.3 Technology Selection

The recommended practice of DNV-GL on offshore leak detection (DNV-GL, 2016) recommends using the BAT following the European directive on industrial emissions (European Union, 2010).

The BAT is, in this context, defined as the *"most effective and advanced stage in the development of activities and their methods of operation which indicates the practical suitability of particular techniques for providing the basis for emission limit values and other permit conditions designed to prevent and, where that is not practicable, to reduce emissions and the impact on the environment as a whole […]*" (DNV-GL, 2016).

A two-steps BAT process is suggested. The first step is where single techniques are assessed, and the second is where configurations of different techniques are compared.

Another driver for efficient leak detection (that can be used together with BAT) is the ALARP (As Low As Reasonably Possible) principle and refers to the continuous effort for minimizing hazards and risks also considering time and money needed (Health and Safety Executive, 2019). This approach is required for companies operating in the UK Continental Shelf (DNV-GL, 2016).

Either the BAT or the ALARP can be used simultaneously. This work, in fact, focuses on step 2 of the BAT where different techniques for subsea leak detection are compared. The comparison will be based on performances consideration using the ALARP as the primary driver.

2.4 Economic Aspect

So far, only the environmental and safety aspects related to the LDSs have been discussed. However, the economic side must be strongly taken into consideration.

The first issue is the economic damage that an oil company faces in case of an oil spill: production stops, maintenance must be carried out, remediation is required, possible compensation to third parties may be paid for, and the company experiences a loss of reputation (Gyo Lee et al., 2018).

A second issue is that LDSs are susceptible to false alarms. These events should be minimized because when topside operators receive an alarm from the LDS, an inspection is carried out using ROVs causing an unplanned and unnecessary shutdown which is critical for an oil and gas company (Bucelli et al., 2018).

However, during the implementation of a LDS, it is necessary to consider either CAPEX or OPEX associated with it. In order to achieve high performances, it would be enough to have a significant number of high-performance sensors. However, minimizing costs is one of the goals of any investment. In order to balance the need for having both a good dependability of the LDS and keeping costs under control, a Cost-Benefit Analysis (CBA) is often carried out.

CBA can be used to perform a judgment on whether to implement a specific configuration of LDS or not to implement it. Then results among different configurations can be compared. The following inequality (Paltrinieri and Khan, 2016) can be used for a preliminary evaluation. The left-hand side represents the Benefits of implementing a LDS, while the right-hand side represents the Costs of implementing a LDS. When this inequality is verified, it is suggested to implement a LDS:

$$\left[\left(Cost_{\text{wo\_LDS}} \cdot F_{\text{wo\_LDS}}\right) - \left(Cost_{\text{w\_LDS}} \cdot F_{\text{w\_LDS}}\right)\right] \cdot P_{\text{control}} > Cost_{\text{LDS}}$$

- $Cost_{\text{wo\_LDS}}$ is the present cost of an accident when no LDS is present;
- $Cost_{\text{w\_LDS}}$ is the present cost of an accident when a LDS is present (as seen, continuous monitoring can improve mitigation in case of an accident);
- $F_{\text{wo\_LDS}}$ is the frequency of the initiating event in case no LDS is present;
- $F_{\text{w\_LDS}}$ is the frequency of the initiating event in case a LDS is present;
- P_{control} is the probability that the LDS performs as required;
- $Cost_{\text{LDS}}$ is the present cost of the LDS (either CAPEX or OPEX can be accounted for when estimating this value).

If frequencies are not available, assume $F_{\text{wo\_LDS}} = F_{\text{w\_LDS}} = F_{\text{accident}}$; where F_{accident} is the frequency of the accident.

The outcome of this analysis has the limitation of being highly dependent on the frequencies of an accident, which may be extremely low and uncertain. To overcome this numerical issue, a Disproportion Value (X_{DV}) can be used to display an intentional bias in favor of the implementation of a LDS:

$$\left[\left(Cost_{\text{wo\_LDS}} \cdot F_{\text{wo\_LDS}}\right) - \left(Cost_{\text{w\_LDS}} \cdot F_{\text{w\_LDS}}\right)\right] \cdot P_{\text{control}} \cdot X_{\text{DV}} > Cost_{\text{LDS}}$$

$X_{\text{DV}} > 10$ only when risk is very high.

This CBA is strictly related to the principle of reducing risk ALARP (Rushton and Reston, 2006).

The CBA needs knowledge of many specific economic and financial data. A stand-alone study would be necessary for a correct use of this tool applied to this case study; for this reason, in this work, CBA is not used as an indicator for technology selection.

3 Goliat Field

3.1 Overview on the Goliat Field

As previously introduced, the Barents Sea is ground of new explorations carried out by oil companies. So far, two are the productive fields, but only one of them is producing crude oil: the Goliat Field.

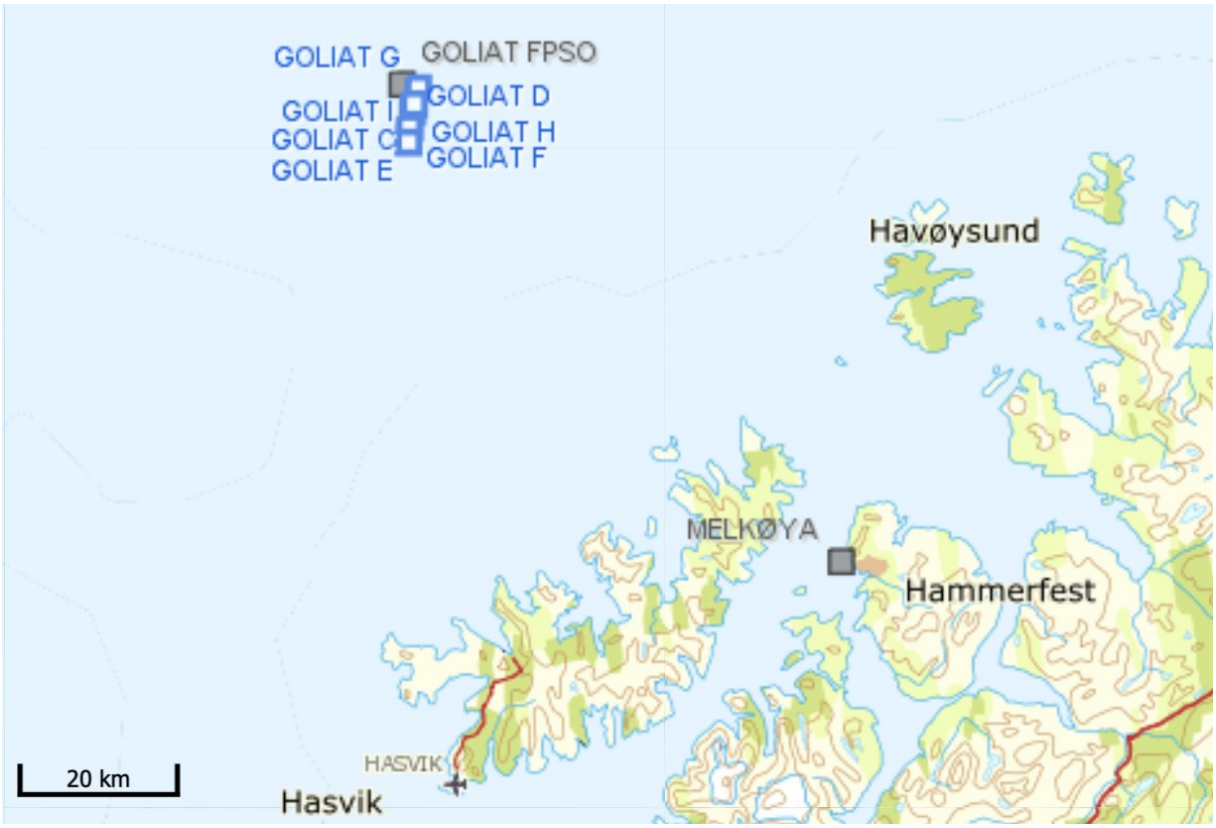


Figure 3.1 Location of Goliat FPSO and its SPS ("Mareano," 2019)

The Goliat Field was discovered in 2000 and, after the approval for production in 2009, started the production on 18th June 2019. It is operated by Vår Energi AS (formerly Eni Norge) which owns 65% of the license; the remaining 35% is owned by Equinor Energy AS (formerly Statoil Petroleum). This field is estimated to contain $93.10\ MSm^3$ of original oil-in-place, among which $31.45\ MSm^3$ are recoverable (Norwegian Petroleum Directorate, 2019b).

The field is developed by using the cylindrical-shaped Sevan 1000 FPSO designed by Sevan SSP and built by Hyundai Heavy Industries in South Korea. It is electrically powered by a $110\ km$ long AC cable which provides $60\ MW$ and by a $30\ MW$ turbine generator, while it is thermally powered by a waste heat recovery unit which provides $35\ MW$ and by $15\ MW$ from electrical heaters (Bjørnbom et al., 2016). Sevan 1000 FPSO has the following design capacities (Eni, 2015; Vårdal, 2010):

- Oil: $104\ kbopd$;
- Gas: $3.9\ MSm^3/d$;

- Additional spare capacities (space and weight) for possible future tie-ins.

Figure 3.2 Sevan 1000 FPSO during the transportation operation (Sevan SSP, 2019)

The reservoir is multiphase, but the associated gas is entirely re-injected, which means that only crude oil is produced. Production is made by using water injection as pressure support. Additional pressure support results from the gas reinjection (avoiding gas flaring).

The oil transportation is carried out by offloading it to shuttle tankers. The supply base is located in Hammerfest, 85 km southeast of the FPSO.

Eight templates are tied-back to the FPSO and placed in different locations around the field so that different reservoirs can be exploited (Figure 3.3) (Norwegian Petroleum Directorate, 2019b).

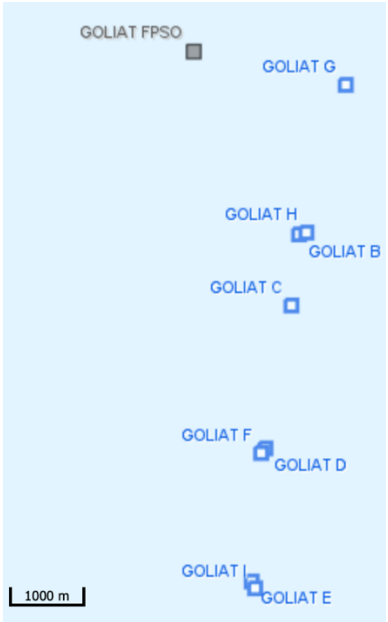


Figure 3.3 Close up of the template positions ("Mareano," 2019)

Each template has four well slots located between 360 and 420 meters of depth (from sea surface). 22 wells are currently present, and 11 of them take oil from different geological formations (Nordbø, 2010):

Realgrunnen main/central:

- Three production wells;
- Three water injection wells.

Realgrunnen south:

- One production well;
- One water injection well.

Kobbe:

- Seven production wells;
- Five water injection wells;
- Two gas injection wells.

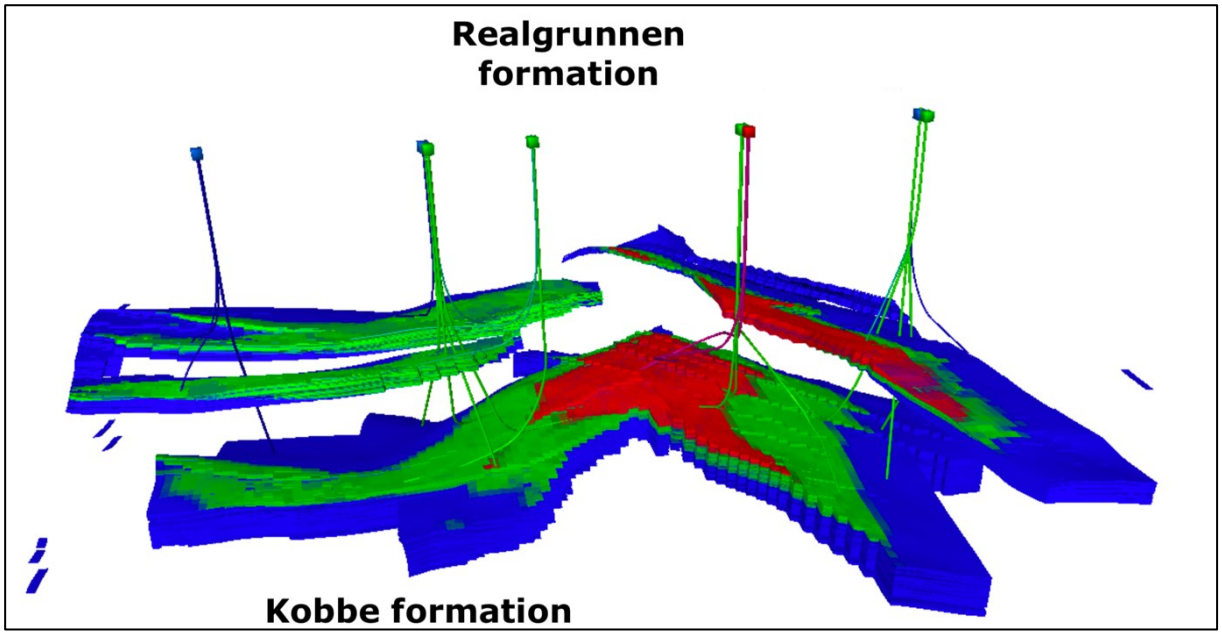


Figure 3.4 Underground representation of the reservoirs and their connections with the templates (Nordbø, 2010)

In Table 3.1, some of the specifications of the field are reported (Eni Norge, 2015; Vårdal, 2010):

	Realgrunnen	Kobbe
API gravity (@ $15°C$)	32.3 (medium oil)	43.3 (light oil)
Pressure	$120\ bar$	$180\ bar$
Temperature	$30 - 35°C$	$48°C$
Underground Depth	$\sim 1200\ m$	$\sim 1800\ m$

Table 3.1 Reservoir specifications

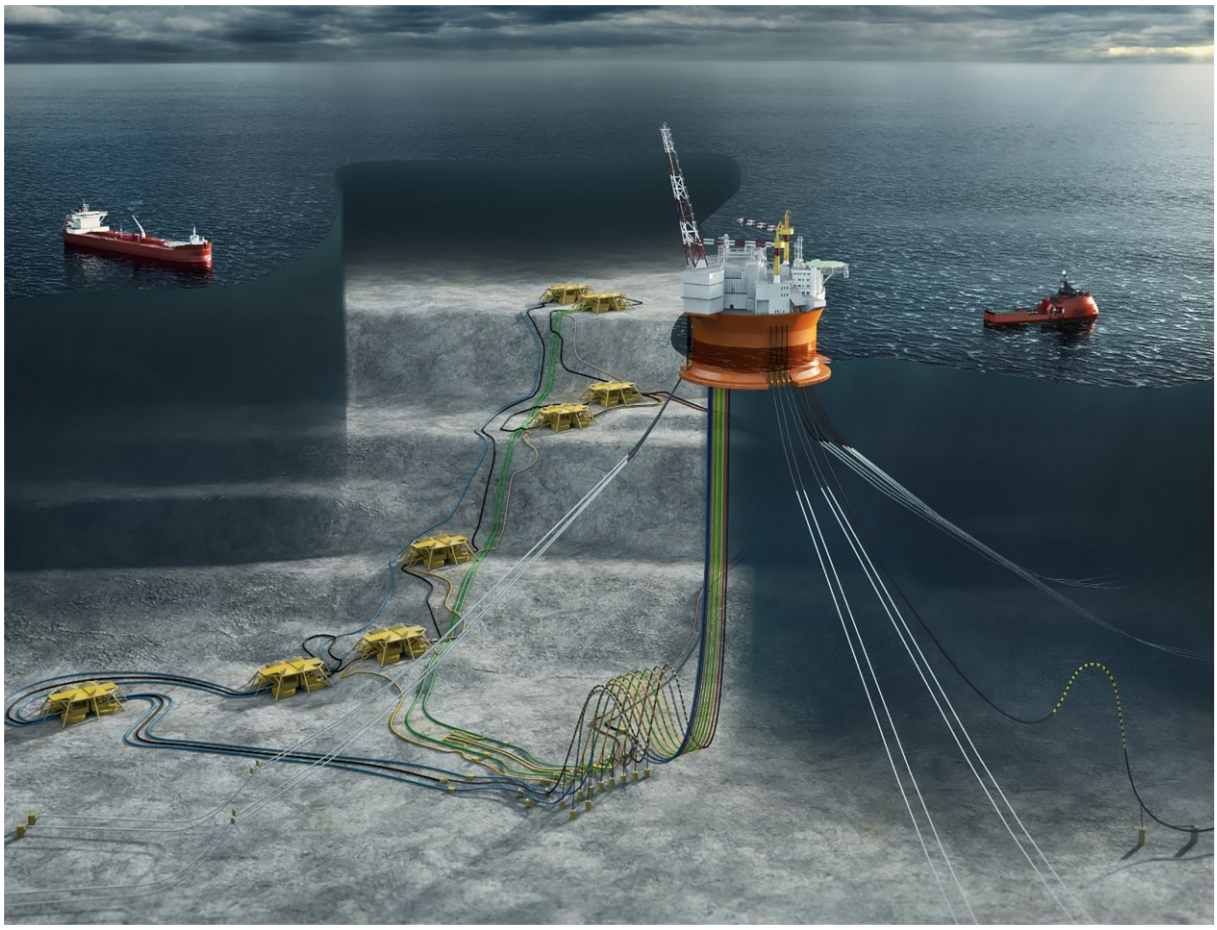


Figure 3.5 Goliat FPSO connected to the eight templates (Wulff, 2016)

3.2 Subsea Production System

About the SPS, the FEED phase was carried out by Eni Norge (now Vår Energi), while the Detail Design, the Procurement, and the Construction were performed by Aker Subsea (Vårdal, 2010).

These slots can be used for three different tasks:

- Production;
- Water Injection;
- Gas Injection.

The layout of a template can be seen in Figure 3.6, where the distinct positions of the four slots are visible.

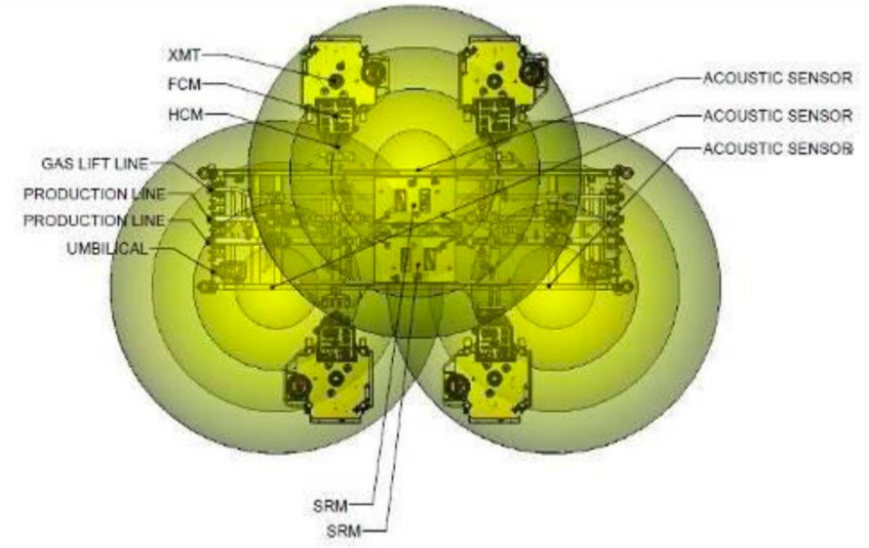


Figure 3.6 Main components of a template (Bjørnbom, 2011; Røsby, 2011)

In each template, some primary components are present:

- Manifold: well outlet and inlet streams are collected in the manifold present in the template. It aims to minimize the use of pipelines and risers and to optimize the flow of the different fluids in the system (Bai and Bai, 2010a);
- Subsea Trees (XMT): it is the set of valves, pipes, fittings, and connection on top of a wellbore. It is necessary to direct the crude oil to the manifold or to canalize the injection of water or gas inside the reservoir as well as other auxiliary chemicals. It can also be used to monitor process parameters and to safely stop the flow of fluids (Bai and Bai, 2010b);
- Subsea Control Module (SCM): it is used to provide well-control functions during production phase by the actuation of valves. It also monitors parameters like pressure (measured by pressure transducers PT), temperature (measured by temperature transducers TT); pressure-temperature transducers (PTT) can also be used for these measurements, and flow rate (measured by flowmeters). It receives electrical power, communication signal, and hydraulic power from the surface;
- Flow Control Module (FCM): the SCM actuates it. It consists of a choke valve whose task is to regulate the flow rate by regulating the diameter of its orifice (Bai and Bai, 2010c);
- The HIPPS Control Module (HCM): it monitors the transmitters and acts on the High-Integrity Pressure Protection System (HIPPS) by closing the valve in case of an excessive increase of pressure. This system protects downstream equipment (Bai and Bai, 2010c);
- Subsea Router Module (SRM): it is the communication link between the subsea equipment and the topside equipment (through umbilical);

Some other essential components of the subsea system are connected to the template:

- Subsea Umbilical System: it consists of electrical cables, fiber optic cables, steel tubes, and thermoplastic hoses. It may include two or three of these four components for executing specific functions like power, communication/signal transmissions, and fluid injections (Bai and Bai, 2010d).

- Subsea Production Riser System: it consists of pipes connected to floaters on the surface and the wellheads. It is the primary device of the production system to convey fluids to and from the vessel (Bai and Bai, 2010e).

3.3 Goliat Oil Spill Detection and Monitoring

Oil spill preparedness involves a large number of actors. Oil spills involve either public authorities or offshore workers. Each of these subjects has specific tasks and contributes to giving useful information about the oil spill according to its expertise.

Figure 3.7 Different actors are involved in the monitoring of the surrounding environment of the FPSO (Vårdal, 2010)

This remote sensing system can be summarized with the scheme in Figure 3.8:

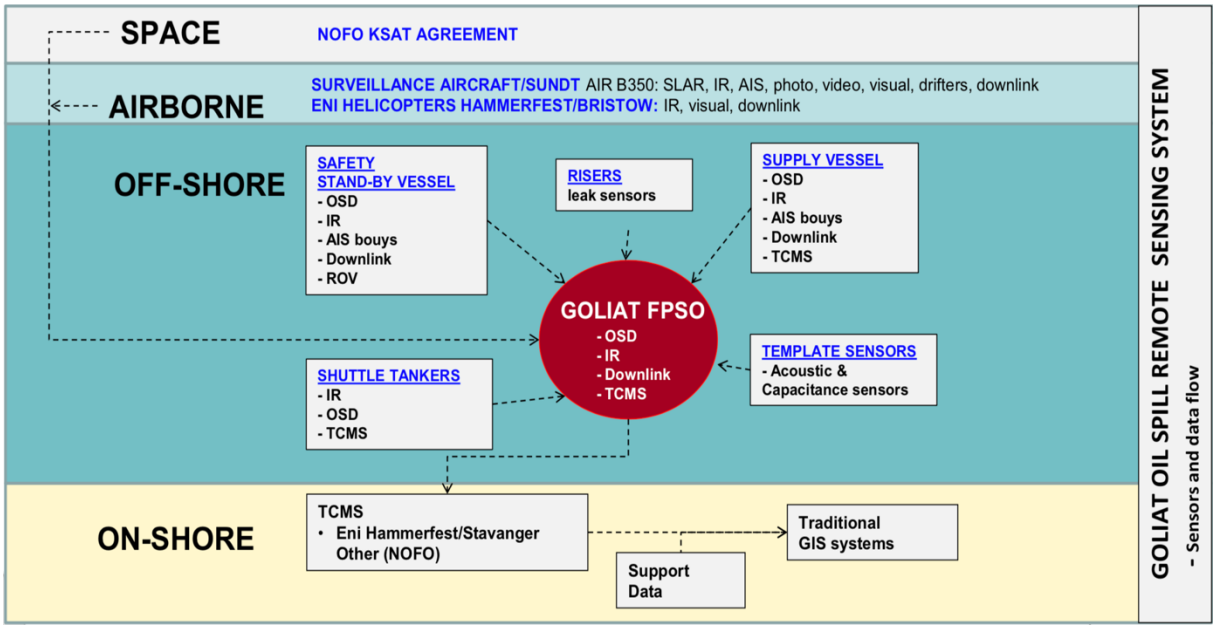


Figure 3.8 Oil spill remote sensing system of Goliat (Bjørnbom, 2017)

3.3.1 Onshore Level

When a possible oil spill is detected, an exchange of information begins between the personnel on the platform and the onshore departments. Main onshore departments are Vår Energi (either Hammerfest or Stavanger headquarters) and NOFO (Norsk Oljevernforening for Operatørselskaper - Norwegian Clean Seas Association for Operating Companies) which both give assistance and support during the mitigation phase.

3.3.2 Space Level

Goliat has access to satellite images via NOFO in collaboration with KSAT (Kongsberg Satellite Services). These two entities have signed an extended agreement for satellite-based remote sensing of the NCS for the detection of acute pollution from petroleum activity (Kongsberg, 2016). The satellite cannot directly detect oil. It only captures the effect of oil on the surface level. The real presence of oil must be confirmed with additional verifications (Bjørnbom, 2011).

3.3.3 Airborne Level

The Norwegian Coastal Administration is responsible for this service providing surveillance airplanes (Beechcraft Super King Air B350) with dedicated instrumentation finalized to the detection of oil spills. Instruments mounted on the aircraft consist of:

- Forward-Looking Infrared (FLIR) cameras (only on primary aircraft) which can sense infrared radiations (Richards, 2012);
- Side-Looking Airborne Radar (SLAR) which can produce images through active sensors mounted on the side of the aircraft. Oil has an absorbing effect on these waves, which reduces backscatter on the water surface. This means lower radar return will be generated when oil is present on the water surface (Gil and Alacid, 2018);
- Geographic Information Systems (GIS), this is useful because it can integrate different geographical data which improve the understanding of those data received by other technologies (Ivanov and Zatyagalova, 2008); this system is integrated with Automatic Identification System (AIS) used to identify ships and vessels present in the area (Norwegian Coastal Administration, 2011);
- Airplanes also have access to NOFO downlink system;
- Photo and video can be produced together with visual information.

Bristow Helicopters EC225 (transport helicopter and All-Weather Search and Rescue helicopter) are also used together with Airplanes. AWSAW helicopter can provide information through:

- Synthetic-Aperture Radar (SAR) is a technique similar to SLAR involving active sensors, and its measure is based on the backscattering produced by microwaves on the sea surface (Topouzelis, 2008);
- FLIR camera;
- Photo and video can be produced together with visual information.

Transport Helicopter can be used for:

- Transporting supporting items or personnel from the coast to the FPSO;
- Giving visual information.

Airborne collected information is also sent to the FPSO.

3.3.4 Marine Level

Goliat can take advantage of either Safety Stand-by Vessels (Esvagt Aurora) or Supply Vessels (Stril Barents) built under specific indications of Vår Energi. Both vessels are equipped with

- Oil Spill Detection (OSD) radars;
- IR cameras;
- Dispersion system;

The Safety Stand-by Vessel will also be provided with an observation Remotely Operated Vehicle (ROV).

During the offloading phase, the platform can also take advantage of the shuttle tanker, which is provided of some detection systems:

- 3 IR cameras;
- OSD radars.

AIS buoys are present in order to facilitate the identification and localization of vessels that, in case of a collision or dropped object, may cause significant oil spills.

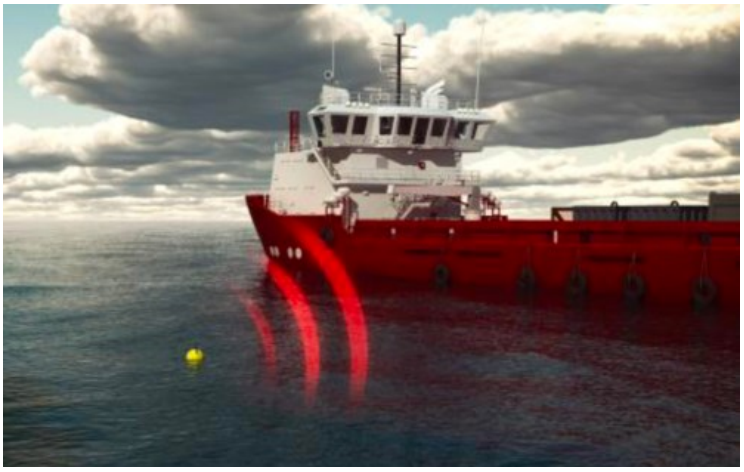


Figure 3.9 AIS buoy detecting the transit of a shuttle tanker (Vårdal, 2010)

3.3.5 FPSO Level

Goliat FPSO is provided with some sensors placed topside among which:

- OSD radars;
- IR cameras.

The FPSO can receive all the necessary information collected by the previous systems.

3.3.6 Subsea Level

Below the sea surface, there are several sensors which monitor the marine environment (and the process streams) looking for possible oil spills. Moreover, ROVs from safety stand-by vessels are available for ordinary (and extraordinary) inspections. Sensors are placed so that the main components of the subsea production system can be monitored.

Pipelines and risers are monitored using process sensors which measure parameters like pressure, temperature, and flow rate. Monitoring these parameters can be useful when the aim is to detect and locate a possible oil spill from a pipeline (Geiger, 2006).

The templates use five different technologies that may help detect anomalies that can be caused by an oil spill:

- Internal LDS (also used for pipelines and risers): they monitor the streams and measure temperature and pressure (using PTTs), and flow rate. When a difference in measured data between two near sensors is detected, a possible leak is present.
- External sensors: they monitor the surrounding environment. Goliat templates use two typologies of sensors:
    - Capacitive sensors;
    - Passive acoustic sensors.

Next section will focus on analyzing the use of sensors to monitor the subsea environment.

3.4 Goliat Subsea Leak Detection System

As seen in the introduction of the FPSO, Goliat has been using a combination of three technologies listed before:

- Internal LDS;
- Capacitive Sensors;
- Passive Acoustic Sensors.

According to Vår Energi (Bjørnbom, 2017), either BAT or ALARP principles were applied during the design phase of the LDS. Design took into account principles present in DNV-RP-302 guideline (Røsby, 2011):

- Integration of the LDS into the design of the SPS;
- Technology selection: use of risk assessment for hotspot selection;
- Combining technologies: area and point detectors.

3.4.1 Internal Leak Detection System

Internal LDS consists of PTTs and flowmeters having the function to monitor possible anomalies from normal conditions in the streams passing through the pipelines, valves and the risers. Some methods can be used to estimate the leakage position and to estimate the amount of spilled oil (Geiger, 2006). These sensors are not sufficiently accurate when it comes to leak detection since flow rate changes over time, making small leakages undetectable for these sensors (DNV-GL, 2016).

3.4.2 Capacitive Sensors

Hydrocarbon Leak Detectors (HLD) are capacitive sensors produced by Phaze (Røsby, 2011).

Figure 3.10 HLD by Phaze (Benestad, 2019)

They are based on the principle that oil (or gas), once leaking, tends to flow upwards because of its density (which is lower than water). While flowing upwards, the collector will trap it (the roof of the XMT forms the collector) making the oil cover the sensor window. As a consequence, the sensor measures a different dielectric constant. An exhaust port is present on the collector in order to reduce false alarms (Benestad, 2019; DNV-GL, 2016; Phaze, 2019).

HLDs are connected to the SCM. The HLDs are installed in the ceiling of the XMTs. Data are then transmitted topside via fiber optic cables. These sensors monitor the wellhead, the XMT, the FCM, and the connection point between the XMT and the manifold (Phaze,

2019; Røsby, 2011). Four sensors (one per slot) are present in case of a fully populated template (four producing wells).

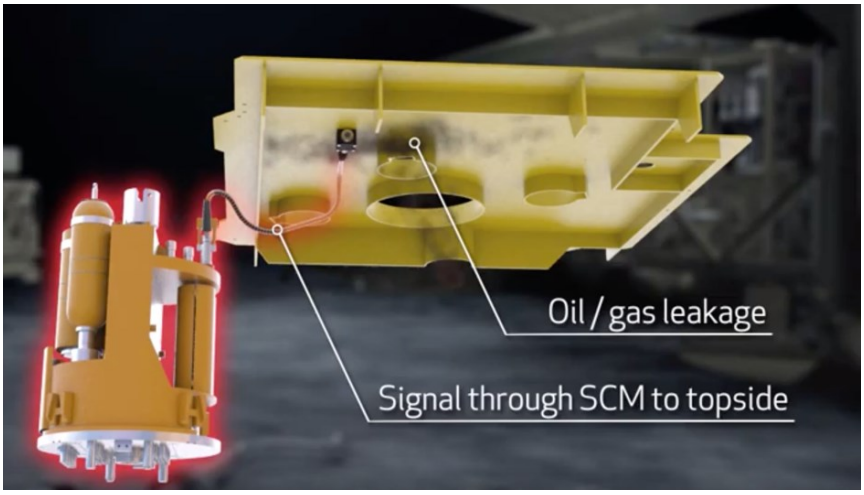


Figure 3.11 Leak detection operated by a HLD connected to the SCM (Benestad, 2019)

3.4.3 Passive Acoustic Sensors

Single Acoustic Leak Detectors (SALD) are passive acoustic sensors produced by Naxys (Naxys, 2019; Røsby, 2011).

Figure 3.12 SALD by Naxys (Naxys, 2011)

A spill creates an acoustic signal as the fluid is passing through a hole. When a leak occurs, a low-frequency acoustic signal is detected by a hydrophone and analyzed. Deviations from what the SALD classifies as noise will produce a local alarm (Austine, 2017).

This typology of sensors should be placed close to the hotspots in order to be able to improve its detection performance as this is strictly related to the distance between leakage source and sensor.

The function of a SALD is to monitor the template, especially those areas not covered by the capacitive sensors. They are mounted on the manifold and connected to the SRM which will transmit data topside via optical fiber cables. A total number of three SALDs is present for any template.

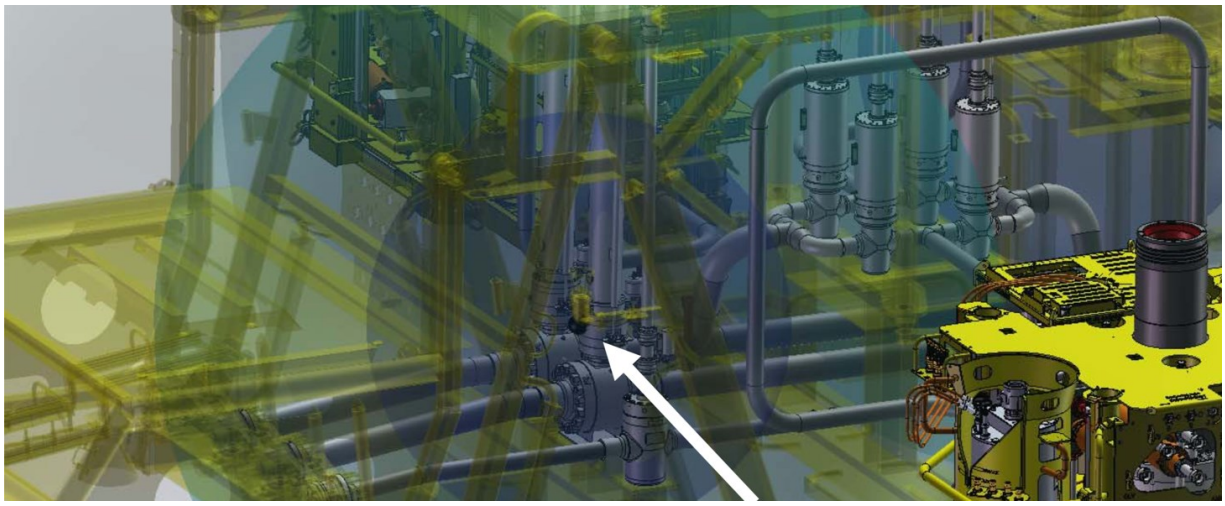


Figure 3.13 Positioning of a SALD on the manifold (Røsby, 2011)

3.5 Hotspots Identification and Sensors Positioning

3.5.1 Christmas Trees and HLDs Positioning

As reported in Table 2.1, which summarizes different technologies, HLDs are sensors which provide point coverage, which means they are specifically useful in those spots corresponding to specific critical equipment (like the XMTs). Positioning these sensors in non-critical spots would not give effective results, as the leak detection only occurs in case the HLD is placed over the leakage point. The XMTs are suitable locations for HLDs positioning as the following critical equipment are present (Bjørnbom, 2017):

- The Spool connection;
- The Annulus Wing Block;
- The Production Block;
- The FCM.

HLDs cannot localize the leakage; this means that it will not be possible to know which exact component is experiencing a leakage, but only that the leakage is located below the area of the collectors.

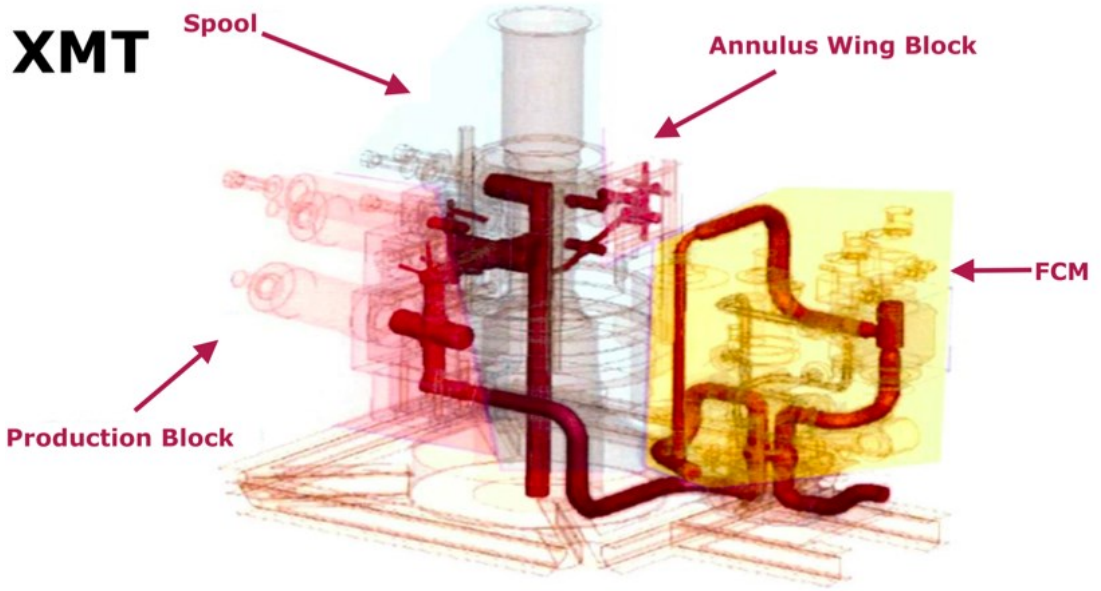


Figure 3.14 Identification of hotspots in a XMT (Bjørnbom, 2017)

3.5.2 Manifold and SALDs Positioning

Multiple SALDs create a network of sensors able to extend the coverage from a single local point to an entire area. This characteristic makes it reasonable to install these sensors in a way that they can monitor critical hotspots that, in case of leakage, would release fluid that HLD collectors could not trap. These hotspots correspond to a set of valves and connections:

- Header Isolation Valves;
- Branch Valves;
- Hub connections.

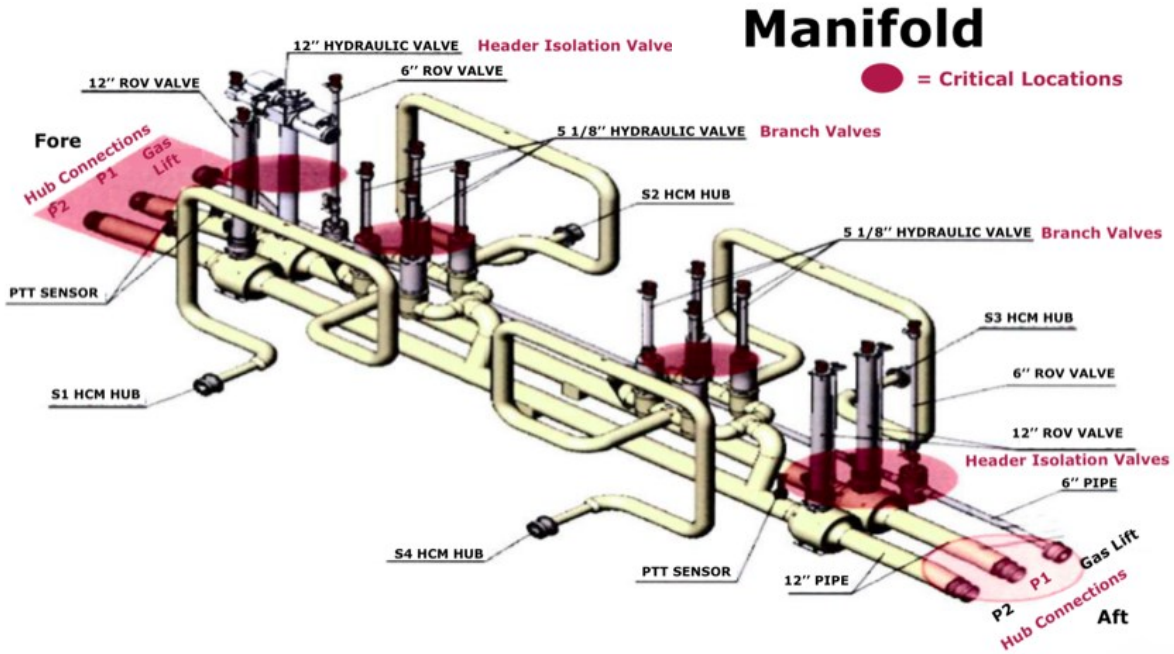


Figure 3.15 Identification of hotspots in the manifold (Bjørnbom, 2017)

The use of passive acoustic sensors working in a network for either detection or localization of oil spills in a subsea template is a technique whose accumulated field experience is limited.

3.5.3 Global Layout

In case of a fully populated template (four production wells), seven sensors are present (four HLDs and three SALDs).

In Figure 3.16 and Figure 3.17, it is possible to see the actual position of the SALDs around the template and the collection areas above the XMTs:

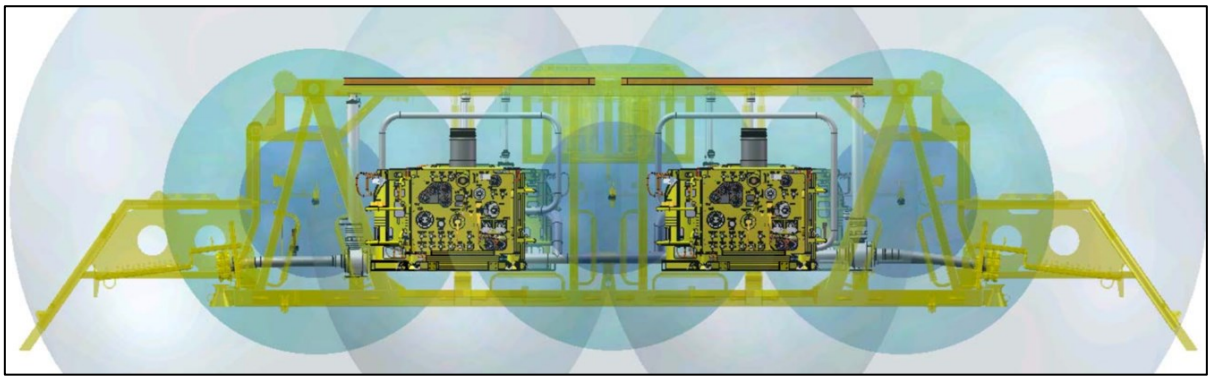


Figure 3.16 Side view of a template (Røsby, 2011)

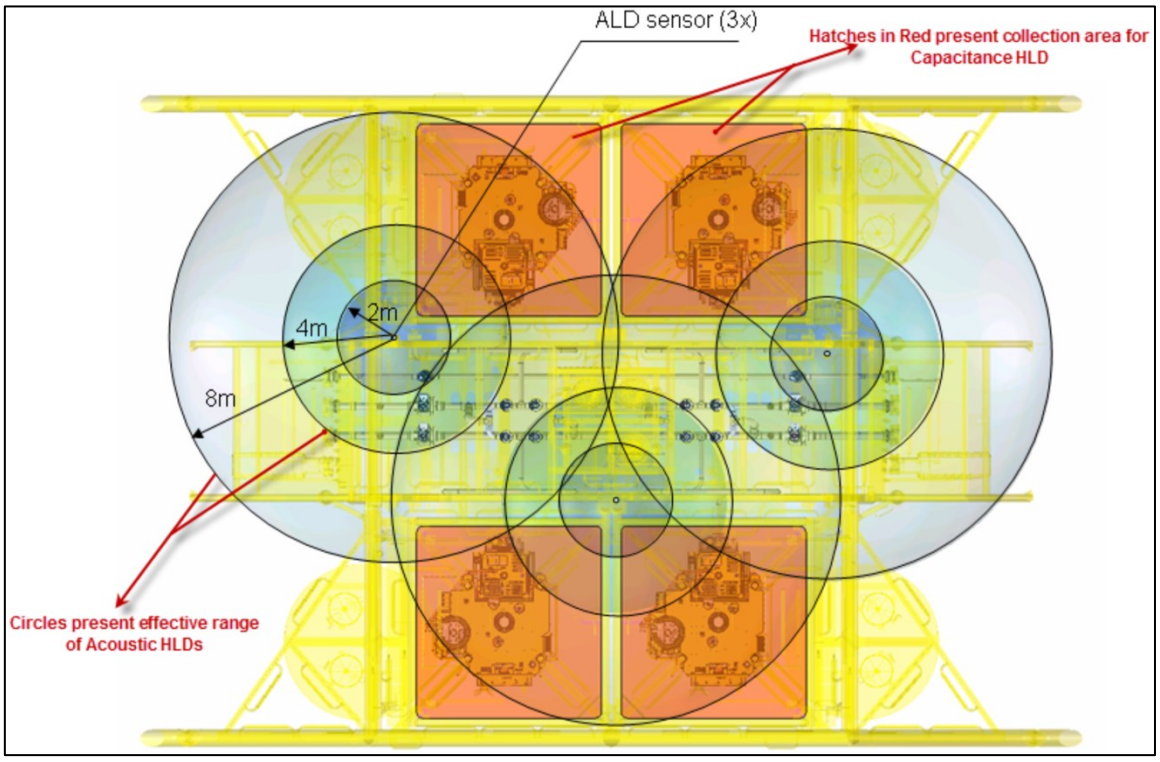


Figure 3.17 Top view of a template (Bjørnbom, 2017; Røsby, 2011)

4 Underwater Acoustic Signal Modelling and Processing

This chapter has the aim to introduce the mathematical and physical models and theories used in order to develop this study.

4.1 Acoustic Signal Modelling

The generic acoustic signal received by the kth sensor is modeled using the following equation (Ciuonzo and Salvo Rossi, 2017):

$$\begin{cases} H_0: & y_k = w_k \\ H_1: & y_k = \xi_k \cdot AAF\left(\underline{x_T}, \underline{x_S}_k\right) + w_k \end{cases}$$

- y_k is the normalized received acoustic amplitude at the kth sensor;
- w_k is a Gaussian distributed random variable representing the amplitude of the thermal noise (modeled as Additive White Gaussian Noise (AWGN)) due to the molecular agitation at the kth sensor (Kularia et al., 2016). More specifically $w_k \sim \mathcal{N}(0, \sigma_{w,k}^2)$ (Kosta, 2002). Noise variance is assumed as equal for any sensor (only one model of SALD is used). Noise variance is normalized to a unitary value: $\sigma_{w,k}^2 = 1$;
- ξ_k is a Gaussian distributed random variable representing the fluctuation in the received signal amplitude (fading coefficient); more specifically $\xi_k \sim \mathcal{N}(0, \sigma_\xi^2)$ (Kosta, 2002). Defining the sensing signal-to-noise ratio as $SNR = \sigma_\xi^2 / \sigma_{w,k}^2$, due to the previous simplification, $\sigma_\xi^2 = SNR \cdot \sigma_{w,k}^2 = SNR \cdot 1 = SNR$.
- $AAF\left(\underline{x_T}, \underline{x_S}_k\right)$ is the distance-dependent amplitude attenuation function which, in this case, is uniquely determined by the distance between the target (whose position is $\underline{x_T}$) and the kth sensor (whose position is $\underline{x_S}_k$). In the case of this work, all distances are calculated neglecting differences in height between two points assuming a 2D model. AAF is retrieved based on either absorption by seawater or spreading loss using physical models (a specific chapter is dedicated to how the AAF is obtained in this study). It is fixed as $AAF = 1$ in the case where the distance coincides with a reference distance l_{ref}.

Being the components of the signal two Gaussian distributed random variables, the signal itself is also a Gaussian distributed random variable (Ciuonzo and Salvo Rossi, 2017; Salvo Rossi, 2019):

$$\begin{cases} H_0: & y_k \sim \mathcal{N}(0, \sigma_{w,k}^2) \\ H_1: & y_k \sim \mathcal{N}\left(0, \sigma_\xi^2 \cdot AAF^2\left(\underline{x_T}, \underline{x_S}_k\right) + \sigma_{w,k}^2\right) \end{cases} \Rightarrow \begin{cases} p(y_k|H_0) = \dfrac{e^{-\frac{y_k^2}{2\sigma_{w,k}^2}}}{\sqrt{2\pi\sigma_{w,k}^2}} \\ p(y_k|H_1) = \dfrac{e^{-\frac{y_k^2}{2\left(\sigma_\xi^2 \cdot AAF^2\left(\underline{x_T}, \underline{x_S}_k\right) + \sigma_{w,k}^2\right)}}}{\sqrt{2\pi\left(\sigma_\xi^2 \cdot AAF^2\left(\underline{x_T}, \underline{x_S}_k\right) + \sigma_{w,k}^2\right)}} \end{cases}$$

which, considering the previous simplifications can be re-written as:

$$\begin{cases} H_0: & y_k \sim \mathcal{N}(0,1) \\ H_1: & y_k \sim \mathcal{N}\left(0, SNR^2 \cdot AAF^2\left(\underline{x_T}, \underline{x_S}_k\right) + 1\right) \end{cases} \Rightarrow \begin{cases} p(y_k|H_0) = \dfrac{e^{-y_k^2/2}}{\sqrt{2\pi}} \\ p(y_k|H_1) = \dfrac{e^{-\frac{y_k^2}{2\left(SNR^2 \cdot AAF^2\left(\underline{x_T}, \underline{x_S}_k\right)+1\right)}}}{\sqrt{2\pi\left(SNR^2 \cdot AAF^2\left(\underline{x_T}, \underline{x_S}_k\right)+1\right)}} \end{cases}$$

Figure 4.1 is a plot showing an example of the distribution probabilities of y_k. It can be seen the broader distribution of the value of y_k in the case of hypothesis H_1 due to its higher variance with respect to the hypothesis H_0. In this example, for the hypothesis H_1: $y_k \sim \mathcal{N}(0,5)$.

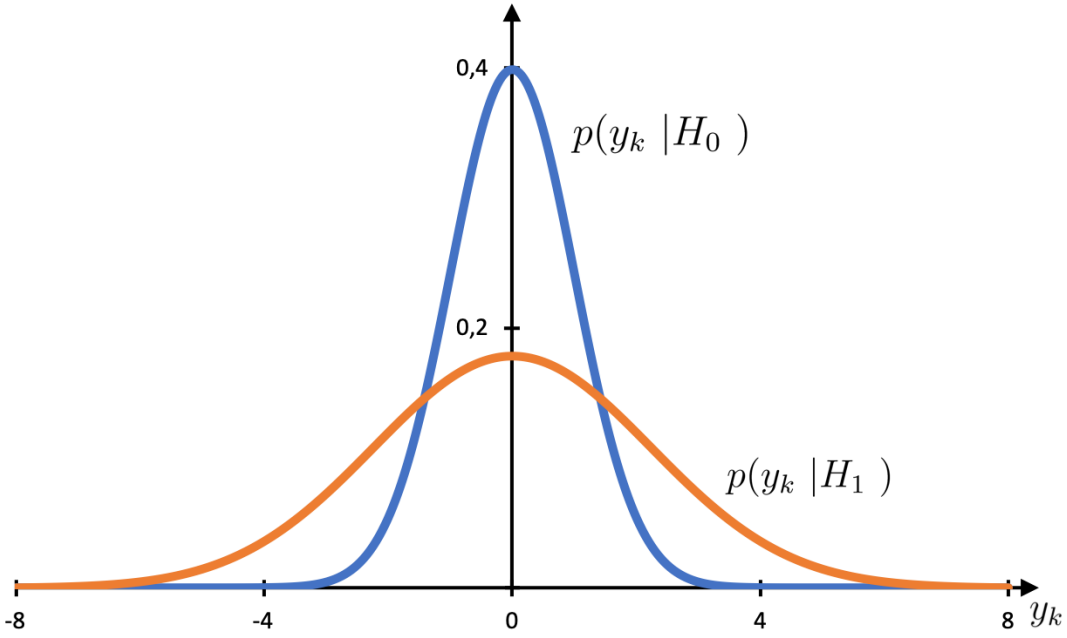


Figure 4.1 Example of distribution of the value of the normalized signal amplitude

4.1.1 Amplitude Attenuation Function

The Amplitude Attenuation Function (AAF) is, as previously introduced, a function that describes the decrease (attenuation) of a wave amplitude over space. This phenomenon, in case of an underwater acoustic wave (where the amplitude represents the sound pressure), is a consequence of multiple factors (Brennan, 2017):

- Geometric Spreading Loss;
- Absorption;
- Reverberation;
- Scattering.

AAF of a signal in a medium typically obeys an exponential law that can be written in the following general form (Chen and Holm, 2003):

$$AAF\left(\underline{x_2}, \underline{x_1}\right) = \frac{Ampl\left(\underline{x_2}\right)}{Ampl\left(\underline{x_1}\right)} = e^{-TL\left(\underline{x_2}, \underline{x_1}\right)}$$

Therefore AAF is the ratio between the amplitude $Ampl$ of a signal in two different points in space. In the previous equation $AAF\left(\underline{x_2},\underline{x_1}\right)$ is the ratio of the amplitude of a signal traveling from the point represented by the vector $\underline{x_1}$ to the point represented by the vector $\underline{x_2}$. What determines the value of AAF is the value of Transmission Loss (TL), which must be reported in Nepers (Np).

TL can be obtained summing all the previous listed contributions. In this work, only two contributions are considered: Spreading Loss and Absorption by seawater:

$$TL\left(\underline{x_2},\underline{x_1}\right) = TL_{\text{seawater\_absorption}}\left(\underline{x_2},\underline{x_1}\right) + TL_{\text{spreading\_loss}}\left(\underline{x_2},\underline{x_1}\right)$$

This simplification is necessary since this model is used to simulate all the signals in this work. In order to account for reverberation and scattering and contribution of sediments, more specific information would be necessary (such as the precise positions of both leak source and sensor, their respective distances from the sea bottom and the sea surface, the exact tridimensional configuration of the subsea template to account possible obstacles). Moreover, to account for sediment contribution to TL, specific morphological information about the seafloor would be necessary.

4.1.1.1 Seawater Absorption

Transmission Loss per unit length due to seawater absorption is called absorption coefficient (indicated as α and often measured in dB/km or Np/km). It usually depends on either the environmental conditions (later represented by the factor α_0) or the sound frequency (f usually in Hz or kHz). It obeys to the following simplified equation (Szabo, 1994):

$$\alpha = \alpha_0 \cdot f^n$$

From this expression, it is immediate that:

$$TL_{\text{seawater\_absorption}}\left(\underline{x_2},\underline{x_1}\right) = \alpha \cdot \left\|\underline{x_2} - \underline{x_1}\right\|$$

There are several ways to estimate the absorption coefficient for underwater sound; here, the most known and used are reported. Thorp proposed the most straightforward equation by carrying out field measurements in the Bahamas (Thorp, 1967). It has no other dependencies than f. It does not catch variations in the absorption as a consequence of the presence of Boric Acid (H_3BO_3) and Magnesium Sulphate ($MgSO_4$) and other specific properties of the seawater like acidity (pH), salinity (S), temperature (T) and depth (D) leading to poorly accurate results (Kularia et al., 2016). Lower errors can be obtained at $T \approx 4°C$ and $D \approx 1000\,m$ (Sehgal et al., 2010). It is meant to be used at low sound frequencies of $100\,Hz \leq f \leq 10\,kHz$ (Al-Dharrab et al., 2013; Domingo, 2008).

A further method is proposed by Schulkin & Marsh (Schulkin and Marsh, 1962) based on measurements in the North Atlantic Ocean. It depends on f, T, S and D. It has a suggested range of validity of $2\,kHz \leq f \leq 25\,kHz$ (Al-Dharrab et al., 2013).

A more accurate method is the one proposed by Fisher & Simmons (Fisher and Simmons, 1977). This method, unlike the Thorp Equation, takes into account the sound absorption caused by H_3BO_3 and $MgSO_4$ together with other parameters like T, pressure (P) (which can be easily obtained knowing D and the average density of the seawater). This model should not be used at $D \geq 8000\,m$ and was obtained assuming fixed values of salinity ($S =$

$35\ ppt$) and acidity ($pH = 8$) (Sehgal et al., 2010). It has a suggested range of validity of $10\ kHz \leq f \leq 400\ kHz$ (Al-Dharrab et al., 2013).

Ainslie & McColm is probably the best method in terms of accuracy with respect to its simplicity. The absorption coefficient in this model, in fact, also depends on S and pH (Ainslie and McColm, 1998). It increases the chances of obtaining more accurate results (Sehgal et al., 2010).

The method used in this work is the one proposed by Francois & Garrison (Francois and Garrison, 1982a, 1982b). This method is the most complex in terms of calculations, but because of its complexity and, since it is based on ocean measurements taken in the Arctic, Northeast Pacific Ocean, Atlantic Ocean, Mediterranean, Red Sea, and the Gulf of Aden, it is considered one of the most accurate models. Moreover, it introduces the dependency on the sound speed (c) which has never been explicitly considered as a variable in the previous methods. f, c, S, D, T and pH are the explicit variables on which this method is based. Its optimal range of validity is the following (Etter, 2018):

- $0.16\ kHz \leq f \leq 650\ kHz$;
- $7.69 \leq pH \leq 8.18$;
- $8\ ppt \leq S \leq 40.5\ ppt$;
- $-17.5°C \leq T \leq 22°C$;
- $0.013\ km \leq D \leq 3.35\ km$.

Francois & Garrison's method is what it is used in this research because its optimal range of value is vast making this method very versatile and usable even in case of significant changes in seawater conditions or in case a different reference f is used. It also has the highest number of variables among all the methods, and it is considered a reliable and accurate method. Moreover, it is based on actual field measurements in different seas and oceans among which the Arctic Area.

Francois & Garrison's method is based on the following expression which can be separated into three different additive contributions given by H_3BO_3 and $MgSO_4$ and pure water:

$$\alpha = \frac{A_1 P_1^* f_1 f^2}{f_1^2 + f^2} + \frac{A_2 P_2^* f_2 f^2}{f_1^2 + f^2} + A_3 P_3^* f^2$$

- α is the seawater absorption ($\frac{dB}{km}$);
- f is the sound frequency (kHz);
- A_1, A_2, A_3 are parameters depending on various variables ($\frac{dB}{km\ kHz}$):
  $A_1 = \frac{8.86}{c} \cdot 10^{(0.78 \cdot pH - 5)}$
  $A_2 = 21.44 \frac{S}{c}(1 + 0.025 \cdot T)$
  $A_3 = \begin{cases} 3.964 \cdot 10^{-4} - 1.146 \cdot 10^{-5} T + 1.45 \cdot 10^{-7} T^2 - 6.5 \cdot 10^{-10} T^3, & T > 20°C \\ 4.937 \cdot 10^{-4} - 2.59 \cdot 10^{-5} T + 9.11 \cdot 10^{-7} T^2 - 1.50 \cdot 10^{-8} T^3, & T \leq 20°C \end{cases}$
- P_1^*, P_2^*, P_3^* are nondimensional pressure correlation factors:
  $P_1^* = 1$
  $P_2^* = 1 - 1.37 \cdot 10^{-4} D + 6.2 \cdot 10^{-9} D^2$
  $P_3^* = 1 - 3.83 \cdot 10^{-5} D + 4.9 \cdot 10^{-10} D^2$
- f_1, f_2 are relaxation frequencies of H_3BO_3 and $MgSO_4$ (kHz):
  $f_1 = 2.8 \left(\frac{S}{35}\right)^{0.5} \cdot 10^{\left(4 - \frac{1245}{273 + T}\right)}$

$$f_2 = \frac{8.17 \cdot 10^{\left(8-\frac{1990}{273+T}\right)}}{1 + 0.0018(S-35)}$$

Where c is in m/s, T is in $°C$, S is in ppt and D is in m.

Francois & Garrison's method needs knowledge of c. In order to calculate this parameter in the underwater condition, a few methods exist. Those reported here are among the most popular methods to obtain sound speed (Talib et al., 2011).

Medwin's equation (Medwin, 1975), which is the simplest method that can be implemented, shows how sound speed depends on T, S and D. Medwin proposed a range of validity of $-17.5°C \leq T \leq 22°C$; $0\,ppt \leq S \leq 45\,ppt$; $0\,m \leq D \leq 1000\,m$.

Mackenzie's equation (Mackenzie, 1981) is a very popular nine-terms equation. It depends on T, S and D. Its range of validity is $-2°C \leq T \leq 30°C$; $25\,ppt \leq S \leq 40\,ppt$; $0\,m \leq D \leq 8000\,m$. It makes results more reliable than Medwin's equations when deep water conditions apply.

Del Grosso's equation (Del Grosso, 1974) is among the most reliable methods to obtain c. It depends on T, S and P (which can be calculated knowing D and the average density of seawater). Its range of validity is $0°C \leq T \leq 30°C$; $30\,ppt \leq S \leq 40\,ppt$; $0\,m \leq D \leq 1000\,m$.

Chen & Millero's equation (Chen and Millero, 1977) is a method which is commonly known as the UNESCO equation since it was endorsed by UNESCO itself (Fofonoff and Millard Jr., 1983). It depends on T, S and P and has a range of validity of $0°C \leq T \leq 40°C$; $0\,ppt \leq S \leq 40\,ppt$; $0\,bar_g \leq P \leq 1000\,bar_g$ even though there exists a correction to make it more accurate at lower temperatures ($0°C \leq T \leq 15°C$) and higher pressures ($300\,bar_g \leq P \leq 1000\,bar_g$) (Millero and Li, 1994). This wide range of validity makes it the state-of-the-art in sound speed calculations in underwater conditions. The equation is the following:

$$c = C_w(T,P) + A(T,P)\cdot S + B(T,P)\cdot S^{3/2} + D(T,P)\cdot S^2$$

- $C_w(T,P) = (C_{00} + C_{01}\cdot T + C_{02}\cdot T^2 + C_{03}\cdot T^3 + C_{04}\cdot T^4 + C_{05}\cdot T^5)$
  $+(C_{10} + C_{11}\cdot T + C_{12}\cdot T^2 + C_{13}\cdot T^3 + C_{14}\cdot T^4)\cdot P$
  $+(C_{20} + C_{21}\cdot T + C_{22}\cdot T^2 + C_{23}\cdot T^3 + C_{24}\cdot T^4)\cdot P^2$
  $+(C_{30} + C_{31}\cdot T + C_{32}\cdot T^2)\cdot P^3$
- $A(T,P) = (A_{00} + A_{01}\cdot T + A_{02}\cdot T^2 + A_{03}\cdot T^3 + A_{04}\cdot T^4)$
  $+(A_{10} + A_{11}\cdot T + A_{12}\cdot T^2 + A_{13}\cdot T^3 + A_{14}\cdot T^4)\cdot P$
  $+(A_{20} + A_{21}\cdot T + A_{22}\cdot T^2 + A_{23}\cdot T^3)\cdot P^2$
  $+(A_{30} + A_{31}\cdot T + A_{32}\cdot T^2)\cdot P^3$
- $B(T,P) = B_{00} + B_{01}\cdot T + (B_{10} + B_{11}\cdot T)\cdot P$
- $D(T,P) = D_{00} + D_{10}\cdot P$

Where c is in m/s, T is in $°C$, S is in ppt while P is in kPa. This equation experienced some changes of coefficients once the International Temperature Scale has been adopted. The new coefficients are those used in this work. Updated coefficients A_{xx}, B_{xx}, C_{xx}, D_{xx} can be found in the specific article (Wong and Zhu, 1995).

Other equations that have not been introduced here exist (Etter, 2018). They can be used as alternatives but may not be as accurate and versatile as Chen & Millero's equation. These methods are those of Wilson (Wilson, 1960), Leroy (Leroy, 1969), Frye & Pugh (Leroy, 1969), Lovett (Lovett, 1978), Coppens (Coppens, 1981), Leroy et al. (Leroy et al., 2008).

4.1.1.2 Geometric Spreading Loss

Spreading in space must be considered when calculating transmission loss. The simplest case involves the assumption of an infinite medium with a point-source radiating in all directions. In this case, the energy transmitted is conserved, but it is spread over spheres having a larger and larger radius (Lurton, 2010). The increase of transmission loss (in dB) as the radius increases is:

$$TL_{\text{spreading\_loss}}(R_2, R_1) = k_{\text{spr}} \cdot 10\log\left(\frac{R_2}{R_1}\right)$$

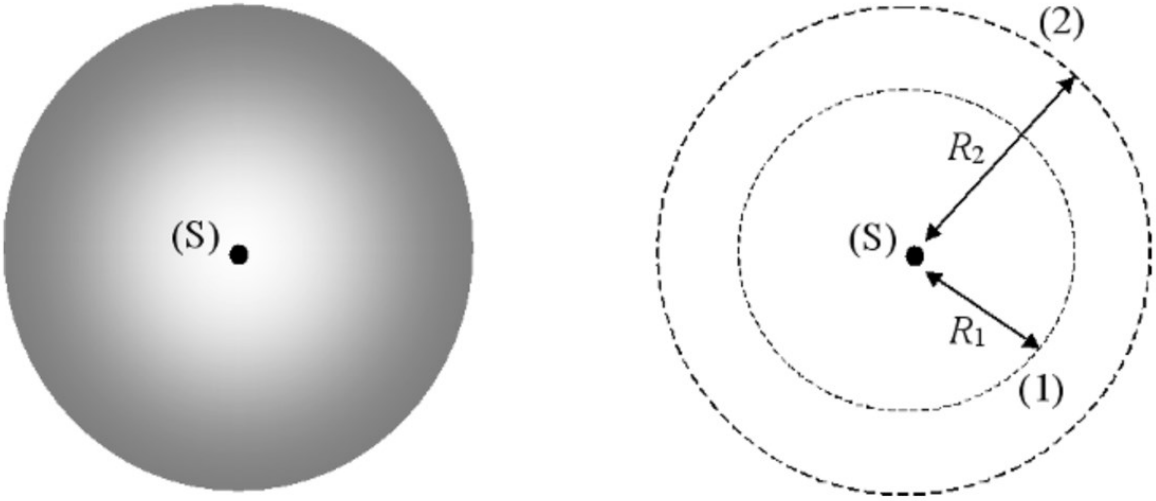


Figure 4.2 Spherical Spreading

In the specific case of spherical spreading $k_{\text{spr}} = 2$ and it can be used to model spreading in deep water condition. For cylindrical spreading instead $k_{\text{spr}} = 1$ which is used to model spreading in shallow water condition. A common practice is to set an intermediate value $k_{\text{spr}} = 1.5$. This is called "practical spreading" (Al-Dharrab et al., 2013; Joshy, 2010; Sehgal et al., 2010; Xiao, 2010).

4.1.1.3 Overall Transmission Loss

Adding the two just examined contributions, the following equation for TL (in dB) can be written down:

$$TL_{dB}(l) = \alpha \cdot (l - l_{\text{ref}}) \cdot 10^{-3} + k_{\text{spr}} \cdot 10\log\left(\frac{l}{l_{\text{ref}}}\right)$$

This equation is valid assuming that l (which is the distances of the sensor from the target) and l_{ref} (which is a reference distance from the target making sure $l_{\text{ref}} \leq l$) are in m and α is in dB/km. Transmission Loss must then be converted from dB to Np so it can be used to obtain the AAF:

$$AAF(l) = e^{-TL_{Np}(l)}$$

It is clear how $AAF = 1$ in the case in which the distance between the sensor and the target is equal to the reference distance: $l = l_{\text{ref}}$.

4.2 Wireless Sensor Network

The present work uses the model of an underwater acoustic wireless sensor network (UAWSN). A Wireless Sensor Network (WSN) is an expression describing a group of spatially dispersed and dedicated sensors for monitoring and recording the physical conditions of the environment and organizing the collected data at a central location (Jawarkar et al., 2013). Those used in a template are not wirelessly connected (cables are used instead). This will not result in any change as the communication channel is assumed as perfect with no communication errors.

WSN can be classified in different typologies. The scheme in Figure 4.3 represents the most straightforward classification for a WSN (Carlos-Mancilla et al., 2016):

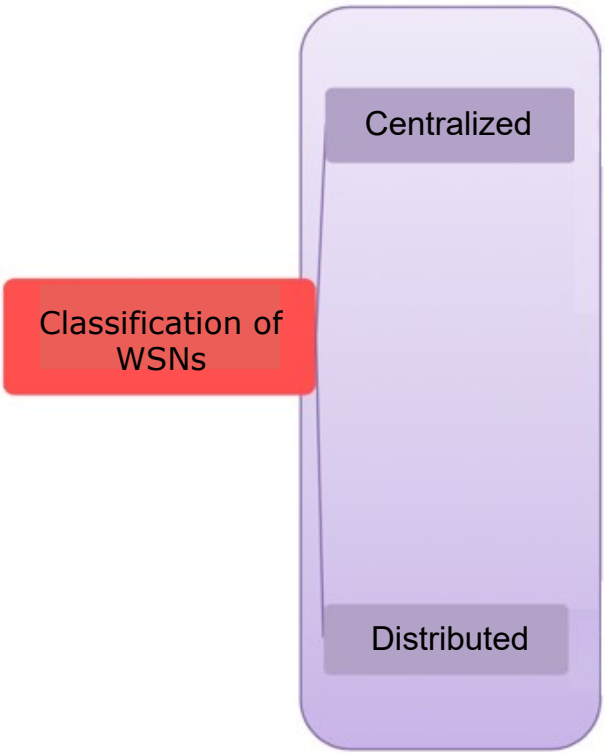


Figure 4.3 Classification of WSNs

These two architectures can be defined as follows (Ling et al., 2009):

- Centralized WSNs are infrastructures in which sensors send a raw measurement data to a Processing Center which has the task to perform a final decision;
- Distributed WSNs are infrastructures in which sensors transmit a local decision obtained from the analysis of raw data to a Fusion Center (FC) which will take a final global decision.

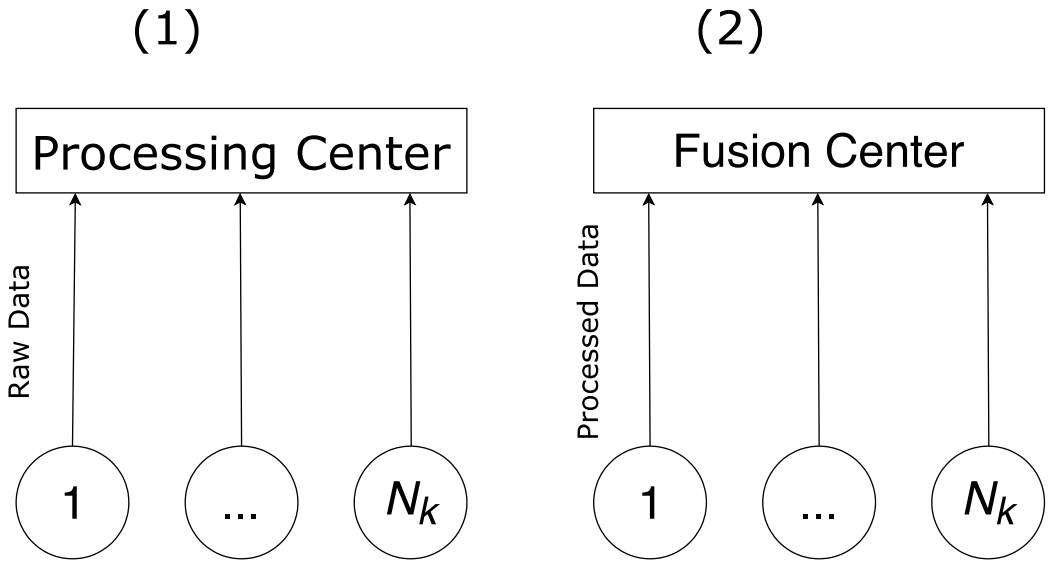


Figure 4.4 Scheme of a centralized (1) and a distributed (2) WSN with N_K sensors

A Distributed WSN is used in the present work. According to the previous definition, this means that each sensor is responsible for a local decision which is transmitted to the FC (Salvo Rossi, 2019):

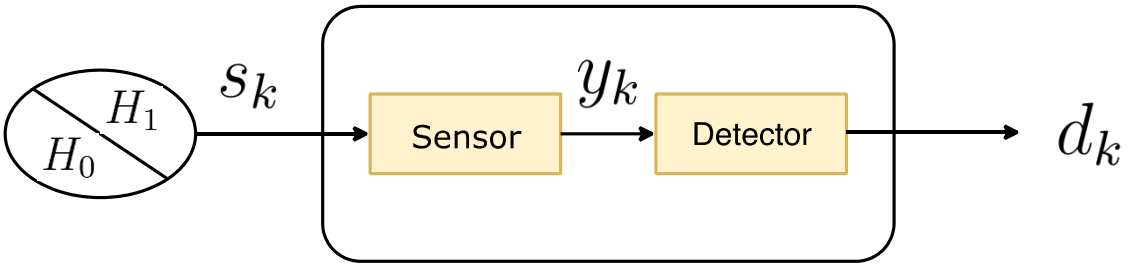


Figure 4.5 Sensor model in the WSN

The target generates a signal s_k, so each sensor performs a test based on the received signal y_k, and a local decision d_k is taken.

The target follows under the category of the non-cooperative targets (NCT), which refers to those targets whose state and attributes are unknown (Jing et al., 2018). The architecture of a WSN is shown in Figure 4.6 (Yan et al., 2015):

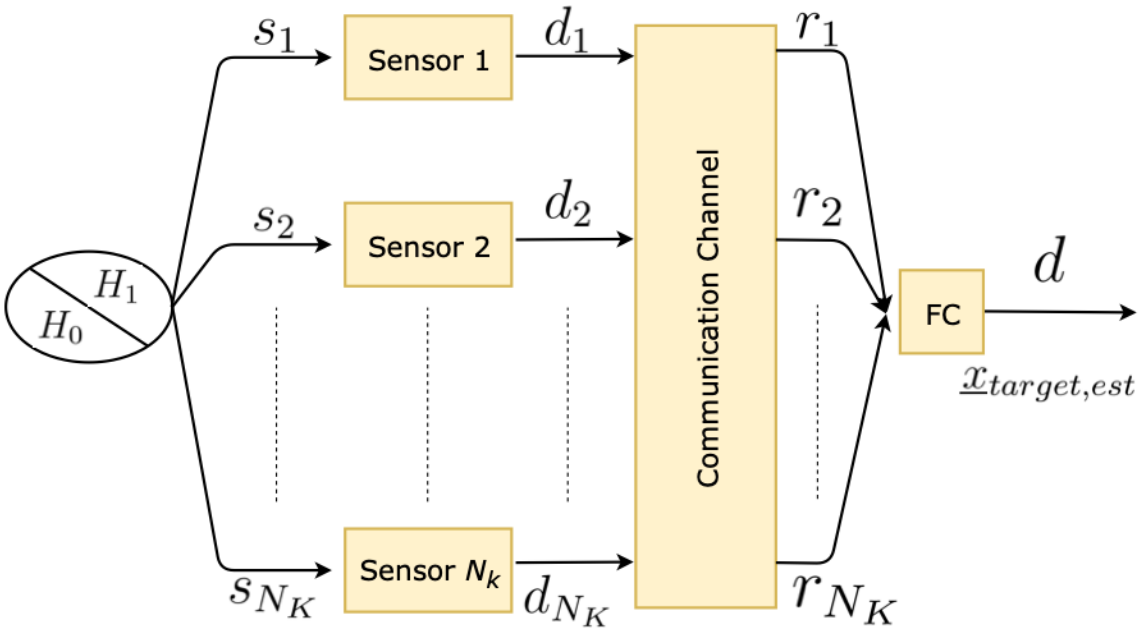


Figure 4.6 Model of the WSN

From the scheme in Figure 4.6, it is possible to see that the considered distributed WSN has only one sink node with sensors placed in a star configuration with no hierarchical- or cluster-based structures. It is also necessary to account the virtual presence of a communication channel between each sensor and the FC. The FC gives the overall decision after applying a fusion rule (FR) whose result is compared with a threshold. The decision is indicated as d and, in case of a positive detection, an estimated target position is calculated.

To conclude, it is possible to summarize all the characteristics and assumptions regarding either the acoustic signals or the WSN modeled in this work:

- The system is a distributed WSN working with a decode-then-fuse (DtF) approach;
- Either acoustic signals due to the leak or thermal noise are assumed as having amplitudes normally distributed over time with zero mean and are statistically independent;
- Target is non-cooperative (unknown position and state), but its position is fixed;
- Sensors are in a fixed and known position;
- Sensors are omnidirectional (they can receive the signal from any direction);
- Amplitudes have been normalized assuming noise variance equal to one for any sensor and assuming AAF equal to 1 (no attenuation of amplitude) when the distance between from the target is equal to a reference distance l_{ref};
- Sensors do not collaborate, and their decisions are taken independently;
- Sensors send a local binary hard decision: $d_k = \{0,1\}$ where $k \in \{1,2,\dots,N_K\}$ to limit energy consumption;
- The communication channel is assumed perfect: $r_k = d_k$ where $k \in \{1,2,\dots,N_K\}$.

4.3 Statistical Signal Processing

Signal processing at sensor level and decision fusion at FC level are both based on binary statistical classification. Classification is the action of dividing a set of data into different classes, and it is carried out by a classifier. In this specific case, classification is said to be binary since the set of hypothesis has two elements $H = \{H_0, H_1\}$, where H_0 is the null hypothesis and H_1 is the alternative hypothesis. Confusion matrixes (Table 4.1) are used to summarize the results of a classification process (Fawcett, 2006):

			Assigned Class	
			Positive (H_1)	Negative (H_0)
Actual Class	Positive (H_1)	P	TP	FN
	Negative (H_0)	N	FP	TN

Table 4.1 Confusion Matrix for Binary Classification

Where:

- True Positive (TP) is the number of positive events correctly classified as positive.
- False Positive (FP) is the number of negative events wrongly classified as positive.
- False Negative (FN) is the number of positive events wrongly classified as positive.
- True Negative (TN) is the number of negative events correctly classified as negative.

If P and N are the numbers of positive events and negative events, respectively, it is possible to introduce some useful metrics:

$$\begin{cases} \text{True Positive Rate} \Longrightarrow TPR = \dfrac{TP}{P} = \dfrac{TP}{TP + FN} \\ \text{False Negative Rate} \Longrightarrow FNR = \dfrac{FN}{P} = \dfrac{FN}{TP + FN} \end{cases} \Longrightarrow TPR + FNR = 1$$

$$\begin{cases} \text{False Positive Rate} \Longrightarrow FPR = \dfrac{FP}{N} = \dfrac{FP}{FP + TN} \\ \text{True Negative Rate} \Longrightarrow TNR = \dfrac{TN}{N} = \dfrac{TN}{FP + TN} \end{cases} \Longrightarrow FPR + TNR = 1$$

These parameters can be seen through a probabilistic approach. Let λ be the result of a test performed by the classifier on a set of data and γ the value of a threshold such that, if $\lambda \geq \gamma$, the classifier declares the event as positive, otherwise as negative. That said, TPR and FPR can be re-written as the following conditional probability mass functions (pmf) (Kay, 1998):

$$\begin{cases} \text{Probability of Detection} \Longrightarrow P_d = P(\lambda \geq \gamma | H_1) \\ \text{Probability of False Alarm} \Longrightarrow P_f = P(\lambda \geq \gamma | H_0) \end{cases}$$

What a binary classifier does is to perform a test in order to assign a class. A test is defined as Uniformly Most Powerful (UMP) when, once P_f is fixed, it has the highest value of P_d among all possible tests. A useful theorem can be used to highlight which test is the UMP (if present): this theorem is known as Neyman-Pearson Lemma (Neyman and Pearson, 1933) which can be written in the form (Kay, 1998) where it says that for a hypothesis test between two simple hypotheses (H_0 as the null hypothesis and H_1 as the alternative hypothesis), in order to maximize P_d for a fixed P_f, decide H_1 if

$$\lambda_{\text{LR}} = \frac{p(x|H_1)}{p(x|H_0)} \geq \gamma$$

Where x is the data measured by the test, λ_{LR} is the result of the test called Likelihood Ratio (LR) and γ is a threshold found from the following integral (after fixing p_f):

$$\int_{\{x:\lambda_{\text{LR}} \geq \gamma\}} p(x|H_0)dx = p_f$$

With these premises, LR is the uniformly most powerful test possible. This approach will be recalled later to evaluate different decision rules. It is often a common practice to perform a Log-Likelihood Ratio (LLR) instead of a LR. There are some advantages in doing a LLR instead of a LR (Jurafsky and Martin, 2009):

- It increases the numerical stability since the logarithms of probabilities will less likely cause an underflow with respect to probabilities (which would result in smaller numbers);
- It is faster since the summation of the logarithms of probabilities is performed instead of a multiplication of those (sum is faster than multiplication);
- Many pdfs have an exponential form; such form is analytically eliminated applying a natural logarithm.

This means the test present in the Neyman-Pearson Lemma can be written using the LLR:

$$\lambda_{\text{LLR}} = \ln\left[\frac{p(x|H_1)}{p(x|H_0)}\right] = \ln[p(x|H_1)] - \ln[p(x|H_0)] \geq \gamma$$

4.3.1 Sensor Level

Each sensor has to continuously perform a hypothesis test in order to declare whether the measured signal corresponds to a positive event or negative event. The goal of a passive acoustic sensor, in fact, is to be able to successfully distinguish between a received signal due to a positive event (like the sound produced by an oil leak) and a received signal generated by the molecular vibrations at the receiver (thermal noise). In order to decide which test to perform, it is useful to apply a local LLR to look for the most powerful test. According to the pdfs used to represent the signal amplitude, the result of the test may change. This is the scenario evaluated (as reported in the section dedicated to the signal modeling):

$$\begin{cases} p(y_k|H_0) = \dfrac{e^{-\frac{y_k^2}{2\sigma_{w,k}^2}}}{\sqrt{2\pi\sigma_{w,k}^2}} \\ p(y_k|H_1) = \dfrac{e^{-\frac{y_k^2}{2\left(\sigma_\xi^2 \cdot AAF_k^2 + \sigma_{w,l}^2\right)}}}{\sqrt{2\pi\left(\sigma_\xi^2 \cdot AAF_k^2 + \sigma_{w,k}^2\right)}} \end{cases}$$

The following is the outcome of the LLR (Ciuonzo and Salvo Rossi, 2017):

$$\lambda_{\text{LLR},k} = \ln\left[\frac{p(y_k|H_1)}{p(y_k|H_0)}\right] = \frac{1}{2}\ln\left[\frac{\sigma_{w,k}^2}{\sigma_\xi^2 \cdot AAF_k^2 + \sigma_{w,k}^2}\right] + \frac{\sigma_\xi^2 \cdot AAF_k^2}{\sigma_{w,k}^2\left[\sigma_\xi^2 \cdot AAF_k^2 + \sigma_{w,k}^2\right]} y_k^2$$

The result shows how this LLR is an increasing function of y_k^2. This result means that the following energy test (ET) is UMP:

$$\lambda_{\mathrm{ET},k} = y_k^2 \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_k$$

With this test, it is possible to exploit the difference in the variance of the two pdfs. The diagram in Figure 4.7 shows how the threshold γ_k interacts with the pdfs (less information is shown compared to that in Figure 4.1 for illustrative reasons):

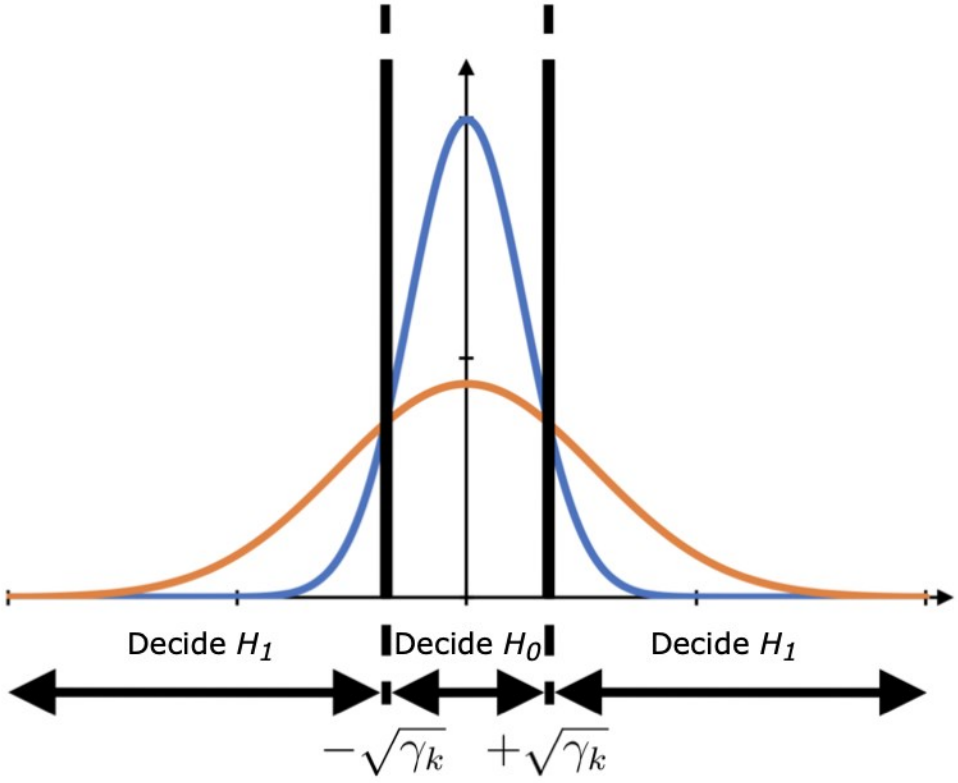


Figure 4.7 Example of Energy Test

It is clear that each possible value of γ_k leads to a non-perfect classification as there is always an overlap between the two pdfs. Different methods are available in order to decide a suitable threshold for each sensor (this will be discussed later). Knowing γ_k and the pdfs, it is possible to obtain two analytical equations which calculate $P_{d,k}$ and $P_{f,k}$ exploiting the knowledge of the complementary cumulative distribution function (ccdf). In this case (Kay, 1998):

$$\begin{cases} P_{d,k} = P\left(\lambda_{\mathrm{ET},k} \geq \gamma_k | H_1\right) = 2\mathcal{Q}\left(\sqrt{\dfrac{\gamma_k}{\sigma_\xi^2 \cdot AAF_k^2 + \sigma_{w,k}^2}}\right) \\ P_{f,k} = P\left(\lambda_{\mathrm{ET},k} \geq \gamma_k | H_0\right) = 2\mathcal{Q}\left(\sqrt{\dfrac{\gamma_k}{\sigma_{w,k}^2}}\right) \end{cases}$$

Where $\mathcal{Q}(\cdot)$ is the Q-function, which is defined as (Kay, 1998):

$$\mathcal{Q}(x) = \frac{1}{2\pi}\int_x^{+\infty} e^{\left(-\frac{t^2}{2}\right)} dt = \frac{1}{2}\mathrm{erfc}\left(\frac{x}{\sqrt{2}}\right).$$

As previously introduced, each sensor takes a binary decision that will be sent to the FC. This local decision is based upon the ET, more specifically:

$$d_k = \begin{cases} 1, & y_k^2 \geq \gamma_k \\ 0, & y_k^2 < \gamma_k \end{cases}$$

4.3.2 Fusion Center Level

The FC has the aim to collect all the data received from each sensor and elaborate a more reliable decision through increasing the global probability of detection and decreasing the global probability of false alarm. In analogy with the test taken at Sensor Level, the FC applies a Fusion Rule (FR). At sensor level, the processed information was y_k which was the variable of the ET. At FC level, instead, the variable of the test is the collection of decisions received from the sensors. This variable can be written using the following notation: $\underline{d} = \left[d_1, d_2, \dots, d_{N_K}\right]^T$.

4.3.2.1 Weighted Fusion Rule

It is possible to apply the Neyman-Pearson Lemma in order to be able to retrieve the UMP FR:

$$\Lambda_{\text{LLR}} = \ln\left[\frac{P(\underline{d}|H_1)}{P(\underline{d}|H_0)}\right] = \ln\left[\frac{\prod_{k=1}^{N_K} P(d_k|H_1)}{\prod_{k=1}^{N_K} P(d_k|H_0)}\right] = \ln\left[\frac{\prod_{k=1}^{N_K} \left(P_{d,k}\right)^{d_k}\left(1-P_{d,k}\right)^{(1-d_k)}}{\prod_{k=1}^{N_K} \left(P_{f,k}\right)^{d_k}\left(1-P_{f,k}\right)^{(1-d_k)}}\right]$$

$$= \sum_{k=1}^{N_K} \ln\left[\frac{\left(P_{d,k}\right)^{d_k}\left(1-P_{d,k}\right)^{(1-d_k)}}{\left(P_{f,k}\right)^{d_k}\left(1-P_{f,k}\right)^{(1-d_k)}}\right] = \sum_{k=1}^{N_K} \left\{ d_k \ln\left[\frac{P_{d,k}}{P_{f,k}}\right] + (1-d_k)\ln\left[\frac{1-P_{d,k}}{1-P_{f,k}}\right]\right\}$$

This FR is commonly known as Chair-Varshney Rule (Chair and Varshney, 1986) or Weighted Fusion Rule (WFR) since each decision is weighted considering the performances of the sensor (obtained in the previous section):

$$\Lambda_{\text{WFR}} = \sum_{k=1}^{N_K} \left\{ d_k \ln\left[\frac{P_{d,k}}{P_{f,k}}\right] + (1-d_k)\ln\left[\frac{1-P_{d,k}}{1-P_{f,k}}\right]\right\} \underset{H_0}{\overset{H_1}{\gtrless}} \overline{\gamma}$$

where Λ is the symbol used for the result of the FR used in the FC and $\overline{\gamma}$ is the applied global threshold. The problem of this FR is its applicability: $P_{d,k}$ and $P_{f,k}$ are parameters that can be calculated only by knowing the constituent parameters of the amplitude pdfs. Many of them may be assumed as typical (statistical CSI, SNR level). The one that cannot be assumed is the target position because it is a NCT. In fact, the position is a variable in the AAF, which is necessary for local performance calculations. For this reason, WFR cannot be exploited in all its effectiveness when dealing with NCTs. In this work, $P_{d,k}$ and $P_{f,k}$ have been estimated a priori (they may be updated).

The global probability of detection (Q_d) and the global probability of false alarm (Q_f) for a given threshold can be obtained through numerical methods.

4.3.2.2 Counting Rule

A way to avoid the problem of having the target position unknown is by applying the Counting Rule (CR) (Ruixin Niu and Varshney, 2005). In this FR, decisions are not weighted:

$$\Lambda_{\text{CR}} = \sum_{k=1}^{N_K} d_k \begin{cases} \geq \overline{\gamma} \Rightarrow H_1 \\ < \overline{\gamma} \Rightarrow H_0 \end{cases}$$

Because of how d_k is calculated, it is clear how this FR works: the number of sensors that declared the event as positive is counted, and this number is compared to a threshold. For example, if $\overline{\gamma} = 1$ at least one sensor must declare the event as positive to make the FC declare the event as positive as well. It is clear that when using this FR, $\overline{\gamma}$ needs to be an integer number going from 0 to $N_K + 1$ (which are the limit cases). Q_D and Q_F for a given threshold can be obtained through numerical methods (Salvo Rossi et al., 2015).

A more straightforward case is the homogeneous scenario, where all the sensors have the same performances $P_d = P_{d,k}$ and $P_f = P_{f,k}$ for any $k = 1,2,\dots,N_K$. In this case, performances can be written in a closed-form according to the ccdf of the binomial distribution (Salvo Rossi et al., 2015):

$$\begin{cases} Q_d = \sum_{i=\overline{\gamma}}^{N_K} \binom{N_K}{\overline{\gamma}} {P_d}^i (1 - P_d)^{N_K - i - 1} \\ Q_f = \sum_{i=\overline{\gamma}}^{N_K} \binom{N_K}{\overline{\gamma}} {P_f}^i \left(1 - P_f\right)^{N_K - i - 1} \end{cases}$$

To be able to compute all the possible values of Q_D and Q_F, it is possible to use the form involving the regularized incomplete beta function (Aludaat, 2018):

$$\begin{cases} Q_d = \dfrac{\int_0^{P_d} x^{\overline{\gamma}} (1 - x)^{N_K - \overline{\gamma}} dx}{\int_0^1 x^{\overline{\gamma}} (1 - x)^{N_K - \overline{\gamma}} dx} \\ Q_f = \dfrac{\int_0^{P_f} x^{\overline{\gamma}} (1 - x)^{N_K - \overline{\gamma}} dx}{\int_0^1 x^{\overline{\gamma}} (1 - x)^{N_K - \overline{\gamma}} dx} \end{cases}$$

Also in this case, some numerical methods or approximation to calculate global performances exist. The CR is useful in case there is not enough information to be able to know the different performances of the sensors, and when one wants to apply a simple FR. Nevertheless, as the assumption of homogeneity gets closer to the real case, CR becomes more and more reliable. This because the CR is the UMP test in case of homogeneous scenario. In such a scenario the WFR would not benefit of the presence of weights since they would all be equal, becoming as powerful as the CR, which would provide the further advantage of being more efficient to perform (multiplications and logarithm would no longer be present).

In this work, the situation cannot be considered homogeneous; the only way would have been to continually have the target at the centroid of the sensor positions. This is not the case since the leak may come from different positions in the template.

In this work, in order to know the global probabilities, a brute force method was proposed (Candy and Breitfeller, 2013):

- Generate two series of signals (one corresponding to a series of positive events, the other one to a series of negative events according to the signal model introduced previously). These series must be made of a sufficiently high number of instants to approximate the exact solution better;
- For each instant, apply the FR and compare it with a threshold;

- Estimate the probability of detection and false alarm.

Two FRs were introduced in this section. These are two among the most essential and common FRs.

4.3.3 Evaluation of Detection Performance

There are many ways to evaluate the performances of a detection system; each of them sees the problem of binary classification from a different perspective. Parameters used are mainly those that can be retrieved from the confusion matrix. Performance evaluation may regard the FR/test from a general point of view, without focusing on a specific threshold (this can be good when the FR must be selected), or may regard both the FR/test and a specific threshold (this can be good for picking up a suitable threshold after the FR/test is chosen).

4.3.3.1 Receiver Operating Characteristic curve

Among the most popular tools that are used when evaluating FR/test performances is the Receiver Operating Characteristic (ROC) curve. This tool consists of a two-dimensional graph in which the FPR is plotted on the x-axis ($FPR = P_f$ or $FPR = Q_f$ according to whether the evaluation is made at sensor level or FC level) whereas the TPR is plotted on the y-axis ($TPR = P_d$ or $TPR = P_d$ for the same reason) (Fawcett, 2006).

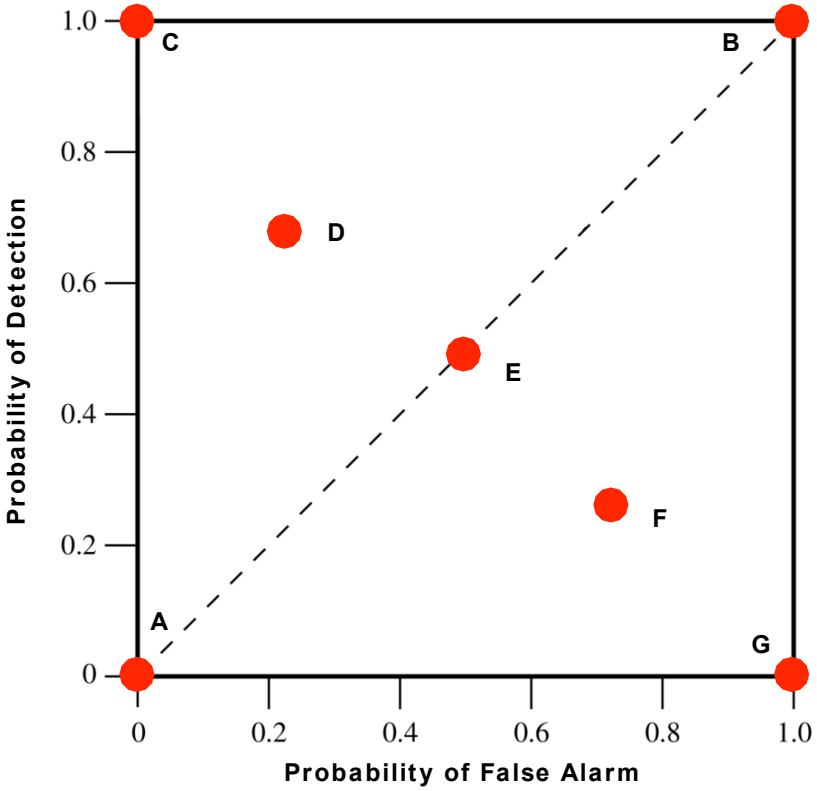


Figure 4.8 Structure of the ROC space

Each point of the ROC space has certain specific features. In Figure 4.8, some peculiar points have been highlighted:

- $A = (0,0)$ This point represents the situation in which either $TPR = 0$ or $FPR = 0$. The explanation to this may be a too high value of threshold that leads the FC/sensor

to never declare the event as positive. It is located on the bottom-left corner of the graph.

- $B = (1,1)$ This point conceptually represents the opposite of A. In this scenario, the threshold is set so small that the FC/sensor always declares the event as positive leading to $TPR = 1$ and $FPR = 1$. It is located on the top-right corner of the graph.

Both points A and B can be obtained by having the highest and the lowest possible value of the threshold, regardless of the applied test/FR. This explains why these two points are located on the so-called chance line. The chance line (or line of-no-discrimination), here represented as a dotted line, is a line going from A to B which is the locus of points obtained by a random classifier, in fact the probability of having a correct detection is equal to the probability of having a false alarm ($TPR = FPR$). An example of such a scenario is the one represented by point E. Above this line, in the north-west section of the ROC space, it is possible to find points representing a more desirable situation, in that section, in fact, $TPR > FPR$. On the other hand, the south-east section of the plot represents the area in which $TPR < FPR$, which is the worst area since even a random classifier would be preferable. F is a point in that area.

- $C = (0,1)$ This point represents a perfect binary classification, in fact, $TPR = 1$ and $FPR = 0$, meaning that no detection errors occur. It is located on the top-left corner of the graph.
- $G = (1,0)$ This point represents the worst-case scenario. In this situation, the classifier completely reverses what the correct output should be. Positive events are classified as negative, while negative events are classified as positive.

Now that ROC space is defined, it is possible to introduce the ROC curve. Once a FR/test and a threshold are fixed, it is possible to apply those to a set of data. The produced confusion matrix corresponds to a point in the ROC space. Varying the threshold, it is possible to see how this point progressively moves towards point A or B (increasing the value of the threshold the point will move towards A, otherwise towards B). This change of threshold value produces a curve called the ROC curve (Figure 4.9):

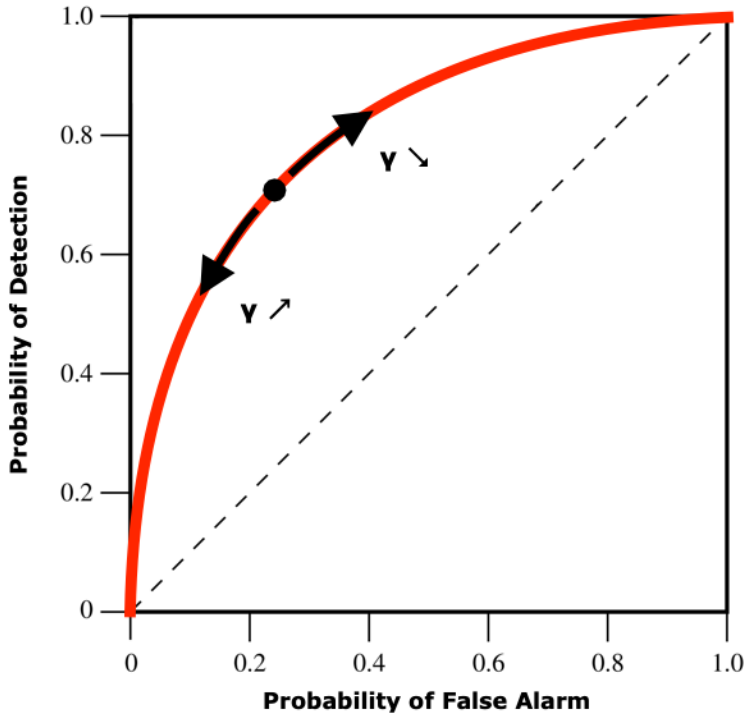


Figure 4.9 Example of ROC curve

The ROC curve is more useful when classes are balanced (Saito and Rehmsmeier, 2015). Through the knowledge of the ROC curve of a FR/test applied to a dataset, it is possible to retrieve some useful parameters like the Area Under the Curve (AUC).

4.3.3.2 Area Under the Curve

The Area Under the Curve (AUC) is a parameter that has the aim of synthesizing the performance of the ROC curve by using one scalar value. This scalar value is the area below the ROC curve, and has the following mathematical expression (Ciuonzo et al., 2013):

$$AUC = \int_0^1 P_d(\gamma) dP_f(\gamma)$$

Of course, the equation is correct also for $Q_d(\overline{\gamma})$ and $Q_f(\overline{\gamma})$ when evaluating FC performances. The integral form may not be applicable since an explicit form for $P_d(\gamma)$ and $P_f(\gamma)$ may not exist or may not be available. Another way it is to use the trapezoidal rule (Bradley, 1997) as an approximation of the previously introduced definite integral. Considering how a ROC space is structured: $AUC \in [0,1]$. $AUC = 0$ when the FR/test is such that, by increasing the threshold, the point moves from B to A passing through point G. This FR/test is the worst possible, and, when coupled with the threshold resulting in point G, it is the least accurate binary classification method. When $AUC = 1$ it means that FR/test is such that, by increasing the threshold, the point moves from B to A passing through point C. This FR/test is the best possible, and, when coupled with the threshold resulting in point C, it is the most accurate binary classification method that may be used. A $AUC = 0.5$ is the same as a ROC curve overlapping the chance line; this makes a binary classification with such value of AUC be classified as a random classifier. Assuming that the classifier performs at least as accurate as a random classifier, it is reasonable to assume that $AUC \in [0.5,1]$. In this case, AUC can be adjusted so that its range of value

moves from $[0.5,1]$ to $[0,1]$ by using the Gini coefficient defined the following equation (Hand, 2001):

$$Gini = 2 \cdot AUC - 1$$

A rule of thumb to make a preliminary judgment after obtaining an AUC value can be the following (Hosmer and Lemeshow, 2004):

- $AUC = 0.5$ it suggests no classification;
- $0.7 \leq AUC < 0.8$ acceptable classification;
- $0.8 \leq AUC < 0.9$ excellent classification;
- $AUC \geq 0.9$ outstanding classification.

AUC is a method that has some drawbacks: it considers the entire domain of the ROC curve even in those areas which are less likely to be of interest. Techniques like partial-AUC can be used to overcome this problem (but it introduces other issues like how to evaluate the arbitrary choice of the range of values considered for its calculation) (Ma et al., 2013). Also, the value of AUC weights the probability of detection and false alarm equally (Lobo et al., 2008).

4.3.3.3 Threshold Selection based on the ROC curve

The choice of the threshold that must be applied to a particular FR/test depends mainly on the specific application/field in which the classifier is supposed to work. The reason is that a different threshold will be displayed as a different point in the ROC and PR space with different values of TPR and FPR and different applications may have different requirements that could lead to the choice of a specific threshold rather than another.

Some parameters are here introduced to have a brief overview of the possible indexes that can be used to evaluate the performance of a FR/test coupled with a certain numeric threshold. Of course, many others exist. However, this list contains only those parameters that are used in this work. This because they are among the most popular and are not specific for certain applications.

- Youden's Index (J) (Youden, 1950)

This value can be calculated with the following equation:

$$J(\gamma) = P_d(\gamma) - P_f(\gamma)$$

Looking at the ROC curve, $J(\gamma)$ represents the length of the vertical segment starting from $\left(P_d(\gamma), P_f(\gamma)\right)$ up to the chance line. This means $J \in [-1,1]$. It is one of the most used indexes, and it is known for giving good performances (Rota and Antolini, 2014). The optimal threshold γ^* can be obtained when the generated performances $P_d(\gamma^*)$ and $P_f(\gamma^*)$ lead to the maximum possible value of Youden's index:

$$\gamma^* : J(\gamma^*) = \max_{\gamma}\left[P_d(\gamma) - P_f(\gamma)\right]$$

- The closest-to-(0,1) criterion (d) (Perkins and Schisterman, 2006)

This criterion is based on the feature of the ROC curve in which the perfect classifier is identified with the point $(0,1)$. This method measures the distance from the top-left corner point to the point on the ROC curve that the considered threshold has identified. This is done using the Pythagorean Theorem:

$$d(\gamma) = \sqrt{\left(1 - P_d(\gamma)\right)^2 + P_f(\gamma)^2}$$

It means $d \in [0, \sqrt{2}]$. Alternatively (like in this work) d^2 can be used:

$$d^2(\gamma) = \left(1 - P_d(\gamma)\right)^2 + P_f(\gamma)^2$$

$d^2 \in [0,2]$. In this case, the optimal threshold γ^* can be obtained when the generated performances $P_d(\gamma^*)$ and $P_f(\gamma^*)$ lead to the minimum possible value of the distance from $(0,1)$:

$$\gamma^* : d^2(\gamma^*) = \min_{\gamma} \left[\left(1 - P_d(\gamma)\right)^2 + P_f(\gamma)^2\right]$$

- Concordance Probability Objective Function (CZ) (Liu, 2012)

This parameter is conceptually similar to the idea of AUC. In fact, it measures the area of the rectangle whose angles are $\left(0, P_f(\gamma)\right), \left(P_f(\gamma), P_d(\gamma)\right), \left(1, P_d(\gamma)\right), (1,0)$:

$$CZ(\gamma) = P_d(\gamma) \cdot \left(1 - P_f(\gamma)\right)$$

$CZ \in [0,1]$. In this case, the optimal threshold γ^* can be obtained when the generated performances $P_d(\gamma^*)$ and $P_f(\gamma^*)$ lead to the maximum possible value of CZ:

$$\gamma^* : CZ(\gamma^*) = \max_{\gamma} \left[P_d(\gamma) \cdot \left(1 - P_f(\gamma)\right)\right]$$

The following ROC curve in Figure 4.10 shows the graphical representation of the three parameters just shown applied to a generic point of the curve:

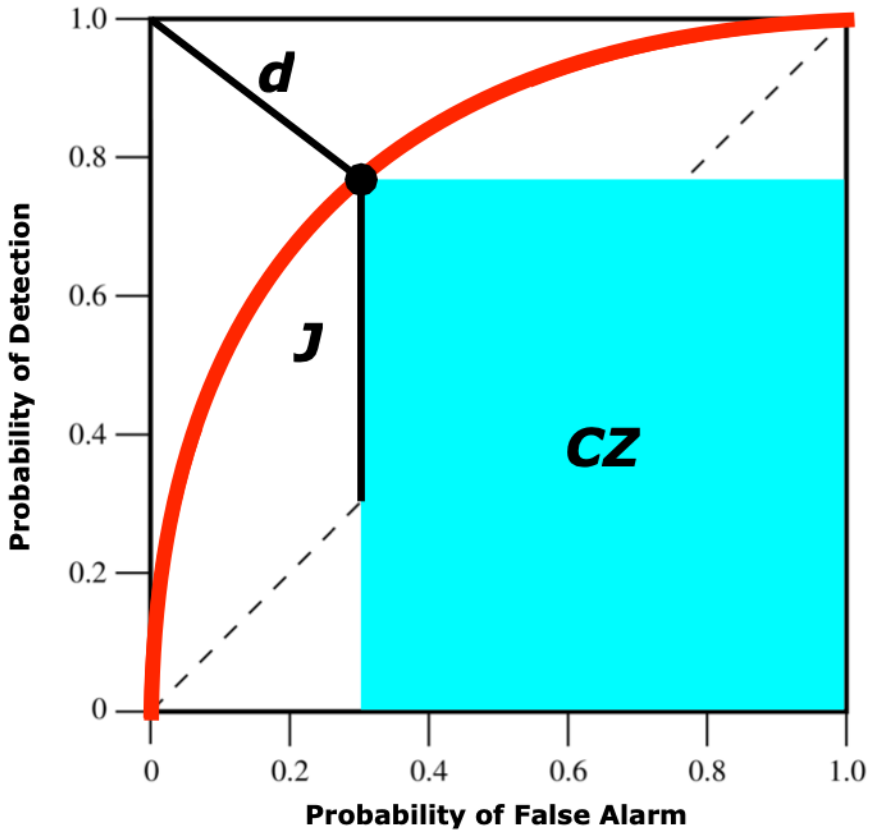


Figure 4.10 Indexes for optimal threshold choice

A significant limitation of CZ and d^2 is that two points on the ROC space having the same values of CZ or d^2 could be on the opposite side of the ROC space with respect to the chance line; this does not happen using J. In this case, two points having the same value of J will be on the same half of the ROC space. The only way for these points to be on opposite sides of the ROC space is by having J with different signs.

4.3.3.4 Precision-Recall Curve

In order to introduce the Precision-Recall (PR) curve, some parameters called Precision and Recall needs an introduction (also known as PPV - Positive Predicted Value):

$$\begin{cases} \text{Precision} = PPV = \dfrac{TP}{TP + FP} \\ \text{Recall} = TPR = \dfrac{TP}{TP + FN} \end{cases}$$

Precision describes how good is the model at predicting the positive class. The recall is equal to the TPR. It is possible to see the advantages over the PR-curve: precision does not depend on TN focusing its attention on how the model predicts the positive class. This can be useful when the classes are unbalanced, meaning that there is not a uniform distribution between positive and negative events. An example can be a situation when the negative class is the dominant class of events. In this case, a large number of FP may lead to a small change in the ROC curve since $FPR = \frac{FP}{FP+TN}$. This scenario would lead to a more visible change in the value of precision (Davis and Goadrich, 2006). In Figure 4.11, it is shown how a PR space looks like:

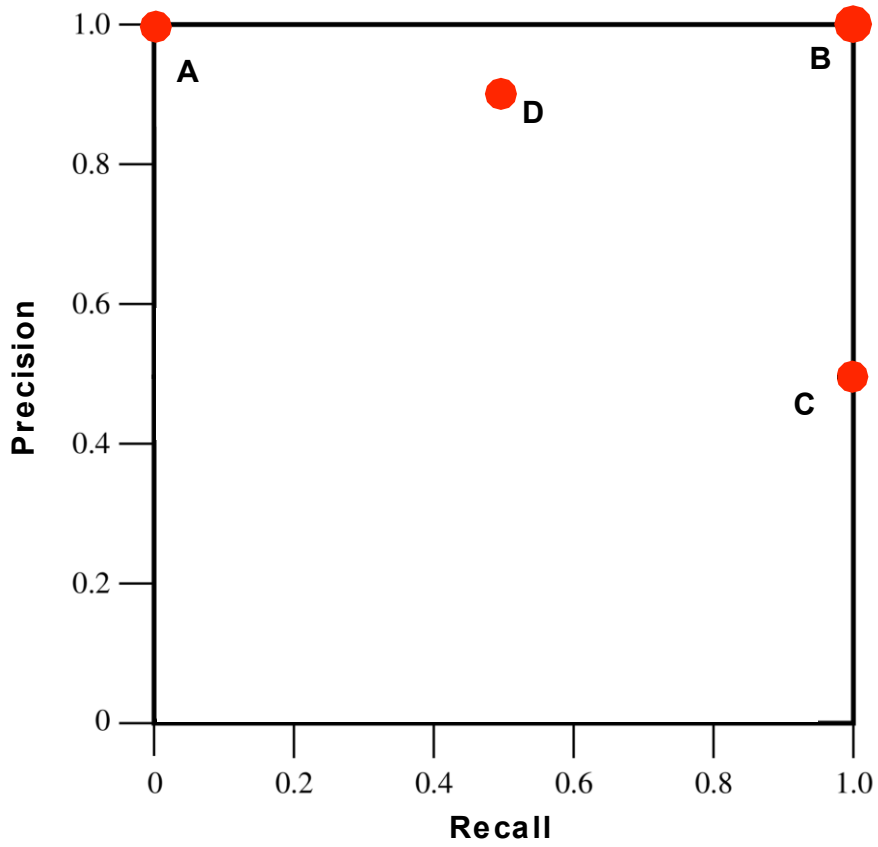


Figure 4.11 Structure of the PR space

- $A = (0,1)$ is a point which represents a threshold so high that the Recall is equal to 0. Precision, in this case, is meaningless because no estimated positive events are present. As the threshold slightly decreases, precision gets closer to 1.
- $B = (1,1)$ is a point in which the threshold is lower than the previous one, and the recall is maximum, but at the same time, precision is maintained at its maximum value too: this is a situation of a perfect classifier.
- $C = (1,0.5)$ is a point representing a very low threshold. In this case, Recall is equal to 1 since every event is classified as positive, but precision is equal to 0.5.
- D is a point representing a lower threshold compared to the one generating point A. In D recall increases with a decrease in precision (unlike point B where precision is kept maximum).

In analogy with the ROC curve, also PR curve is generated by varying the threshold of the FR/test (as shown in Figure 4.12):

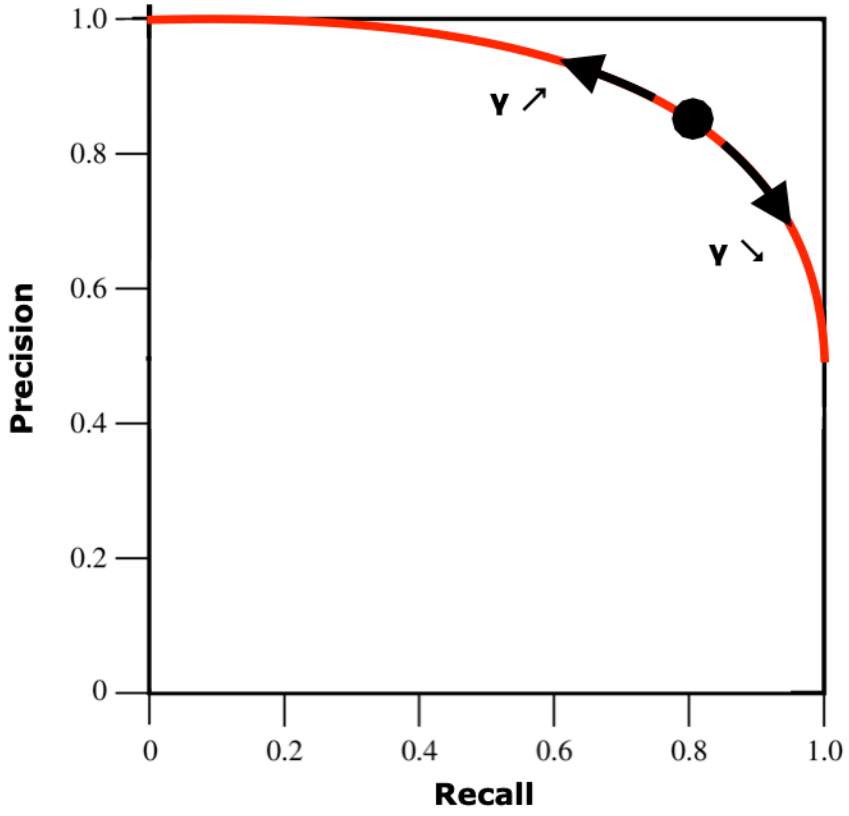


Figure 4.12 Example of PR curve

The area under the curve can also be calculated for the case of the PR curve (Boyd et al., 2013).

4.3.3.5 Threshold Selection Based on the PR-curve

F-measure (or F-score) (Sasaki, 2007) is a method to give a score based on the PR curve. When $\beta = 1$, the F-score becomes the harmonic mean between precision and recall:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{recision + recall}$$

$F_1 \in [0,1]$. The reason for the harmonic mean, rather than an arithmetic mean, is because the arithmetic mean would tend to compensate a low value of one of the two variable by having the other parameter high. The harmonic mean, instead, would give as a result a low value even if only one between precision and recall is low. The F-score can be written in its general form:

$$F_\beta = (\beta^2 + 1) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$

β is used to weight the variables: when $\beta > 1$, F-score becomes more recall-oriented (e.g. F_2); when $\beta < 1$, it becomes more precision-oriented (e.g. $F_{0.5}$).

5 Methods for Sensor Tuning, Fusion Center Tuning and Real-time Algorithm

5.1 Sensor Tuning

It is possible to perform the tuning of the SALDs using the mathematical and physical models introduced so far. These sensors will perform an ET, but an adequate threshold must be chosen for each of them. The tuning algorithm as the aim of retrieving these values. Moreover, the algorithm can be used to estimate performances that will be used as local performances by the FC in case of application of the WFR (as FR for detection) and BBM (for localization).

An algorithm is proposed for the tuning of sensors. It consists of faking the presence of a possible oil spill at any HS. The SNR associated with any of these events will be the same (but it could be improved by simulating different scenarios for any HS according to its technical specifications). This means that the only difference among the different positive events (generated from the different HSs) is dictated by the AAF. A loop will then calculate local performances by continuously changing the local threshold. For each iteration, performances will be calculated for each sensor, assuming different HSs as sources of leakage. In the same iteration, performances will be graded (using an index). Therefore, for each threshold, each sensor has collected multiple grades (one for any HS). However, the aim is to obtain one single result for any sensor at different local thresholds. The suggestion is to average, for any sensor, all the grades associated with the same threshold. A weighted mean accounting for different probabilities of failure of the different HSs is suggested. In this case, since these probabilities are not available, an arithmetic mean is performed. This arithmetic mean will generate an average grade for any sensor at a specific value of the local threshold. The optimal local threshold for a sensor will be the one which generated the best grade.

Figure 5.1 shows the flowchart representing the proposed algorithm for the tuning of sensors:

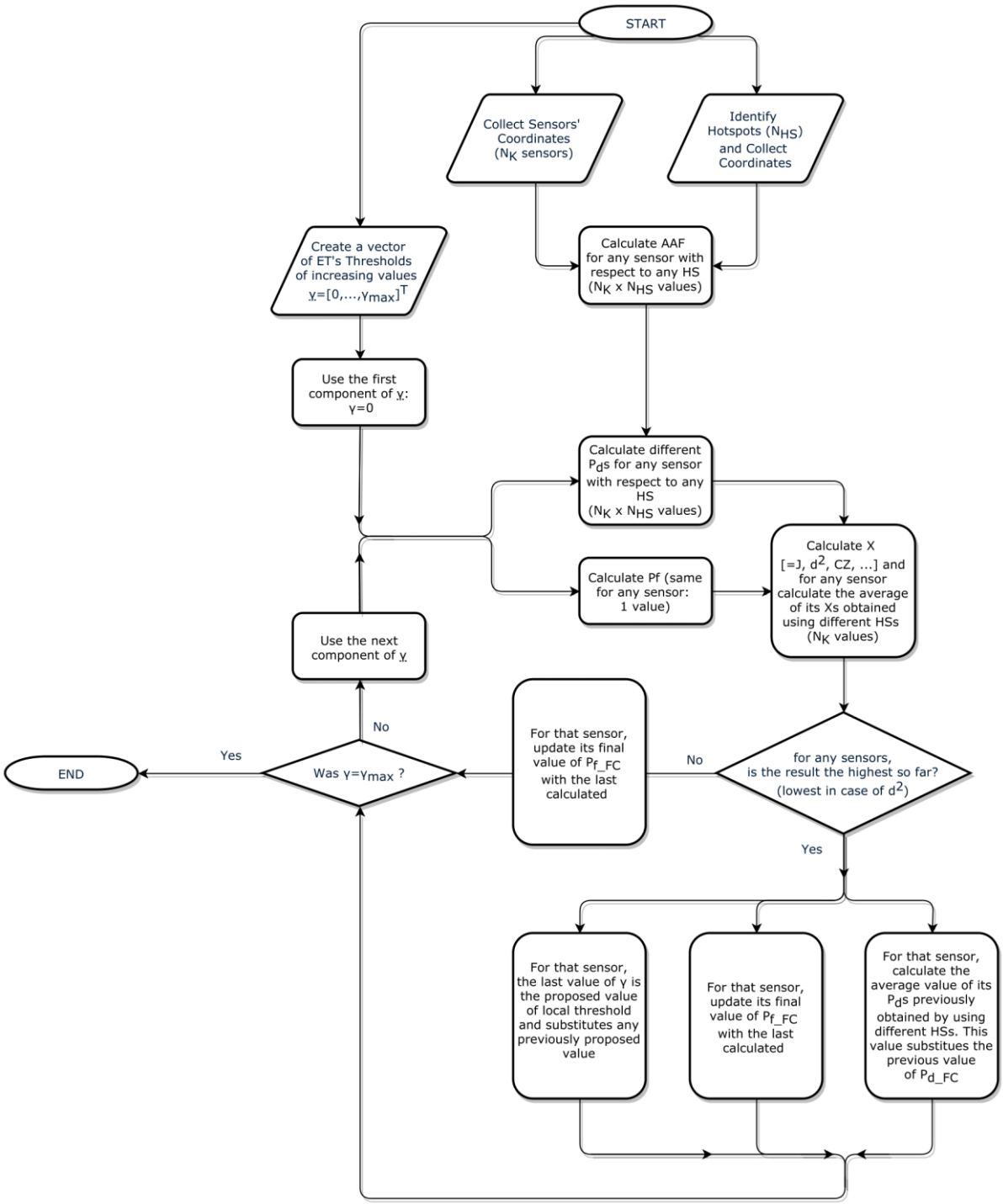


Figure 5.1 Flowchart of the algorithm for tuning of sensors

A detailed step-by-step procedure reporting results is shown in the next section. All the results have been obtained using a MATLAB R2019 script created specifically for this simulation (see Appendix).

5.1.1 Data and Assumptions

The 2D scenario can be seen in Figure 5.2:

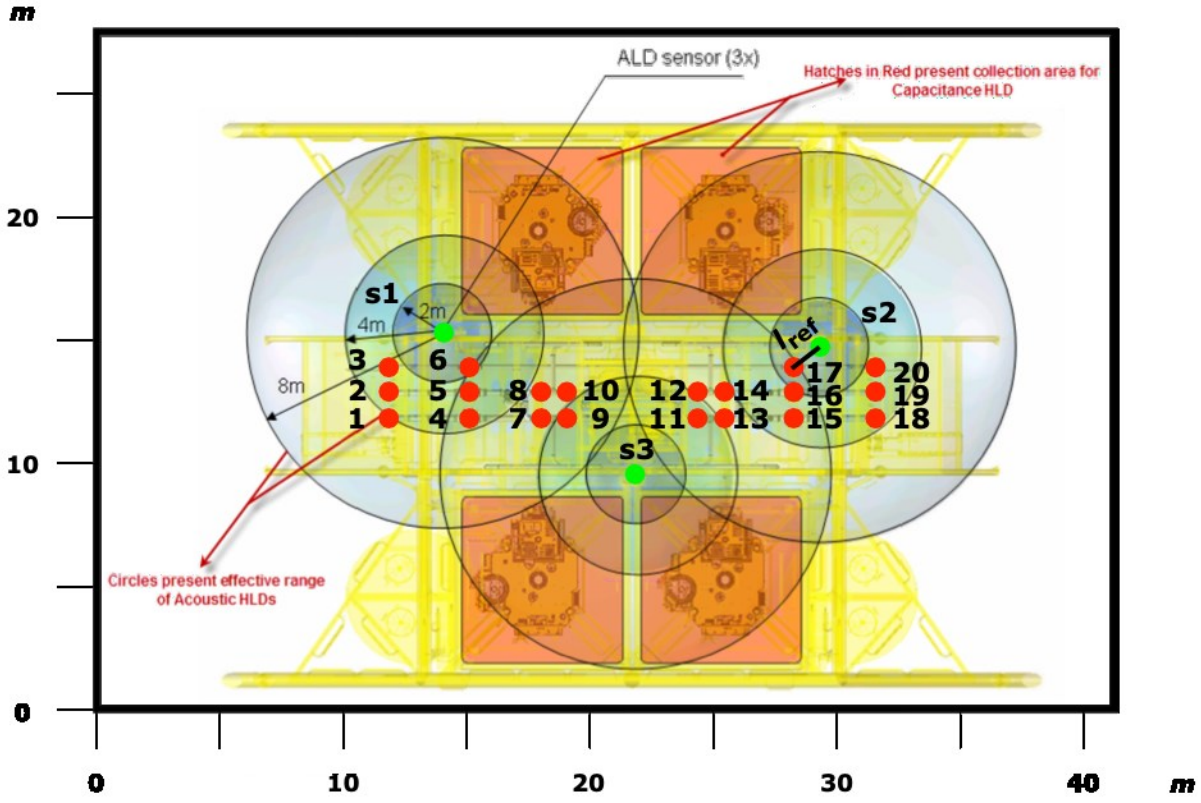


Figure 5.2 Graphical representation of the template top-view from which it is possible to obtain cartesian coordinates. Green dots represent the three SALDs, and red dots represent the hotspots. The reference distance is also indicated. Numbers are used to recognize different HSs and sensors.

Cartesian coordinates are used to simplify calculations. All the calculations made (here and in the next chapters) refer to coordinates obtained from Figure 5.2.

In order to perform the required calculations, it is necessary to know (or suppose) specific data to use as a reference. These are the assumptions and data used in this work:

- Reference distance (l_{ref})

The smallest sensor-hotspot distance is used to define the reference distance: in this case, it is the distance between the north-east sensor (sensor 2) and the hotspot corresponding to a 5 1/8'' hydraulic branch valve (HS number 17):

$$l_{\text{ref}} = 1.47\ m$$

- SNR

It is necessary to assume the average value of SNR in case of leakage at l_{ref} to simulate the behavior of a sensor; this would require tests and the knowledge of the thermal noise at the receiver. The hypothesized value is:

$$SNR_{\text{ref}} = \sigma^2_{\xi,\text{ref}} / \sigma^2_{w,k,\text{ref}} = 20$$

The value of acoustic power produced by an oil spill varies according to different parameters (internal and external pressure, the diameter of the hole, nature of the fluid,…); a separated study should be made to be able to predict this value.

- Reference frequency

It is necessary to know the frequency detected by the sensor to be able to obtain the seawater absorption coefficient from Francois & Garrison method (Francois and Garrison, 1982a, 1982b) which is needed to model the AAF. The AAF is an increasing function of f, for this reason, a low value of sensed frequency is suggested. This will increase the received SNR since thermal noise is also an increasing function of f (Joshy, 2010). It is suggested to have hydrophones able to sense frequencies below $20\,Hz$ (Hamilton and Charalambous, 2013). The latter value is taken as reference for this study:

$$f_{\text{ref}} = 20\,Hz$$

- Reference depth

$$D_{\text{ref}} = 350\,m$$

- Reference temperature

The value of bottom seawater temperature has been taken from Mareano ("Mareano," 2019):

$$T_{\text{ref}} = 3.8°C$$

- Reference salinity

The value of bottom seawater salinity has been taken from Mareano ("Mareano," 2019):

$$S_{\text{ref}} = 35\,ppt$$

- Reference pH

Note that the value of pH of bottom seawater in the Barents Sea is decreasing over time (Wallhead et al., 2017). However, an estimation of its typical value in that area is used (Vetrov and Romankevich, 2004):

$$pH_{\text{ref}} = 8$$

- Spreading coefficient for the geometric spreading loss

The "practical value" is used as suggested in the literature:

$$k_{\text{spr}_{\text{ref}}} = 1.5$$

5.1.2 Coordinates of Sensors and Hotspots

According to Figure 5.2, it is possible to obtain positions of both sensors and HSs.

The next steps will use the following notation:

$\underline{x_S}_k$ where $k = 1, \dots, N_K$ indicates the kth sensor position;

$\underline{x_{HS}}_h$ where $h = 1, \dots, N_{HS}$ indicates the hth HS position;

Table 5.1 reports all the coordinates of the case study:

		Horizontal axis (m)	Vertical axis (m)
Sensor	1	13.97	15.30
	2	29.35	14.66
	3	21.86	9.42
Hotspot	1	11.50	11.75
	2	11.50	12.75
	3	11.50	13.75
	4	15.00	11.75
	5	15.00	12.75
	6	15.00	13.75
	7	18.00	11.75
	8	18.00	12.75
	9	19.00	11.75
	10	19.00	12.75
	11	24.50	11.75
	12	24.50	12.75
	13	25.50	11.75
	14	25.50	12.75
	15	28.20	11.75
	16	28.20	12.75
	17	28.20	13.75
	18	31.70	11.75
	19	31.70	12.75
	20	31.70	13.75

Table 5.1 Coordinates of sensors and hotspots

5.1.3 Calculation of Reference AAFs

For any sensor, a reference value of AAF must be calculated with respect to all the identified hotspots.

$$AAF_{\mathrm{ref}_{h,k}} = AAF\left(\underline{x_{HS}}_h, \underline{x_S}_k, l_{\mathrm{ref}}, k_{\mathrm{spr}_{\mathrm{ref}}}, f_{\mathrm{ref}}, \alpha\left(D_{\mathrm{ref}}, T_{\mathrm{ref}}, S_{\mathrm{ref}}, pH_{\mathrm{ref}}, c(D_{\mathrm{ref}}, T_{\mathrm{ref}}, S_{\mathrm{ref}})\right)\right)$$

Results below were obtained using the equation of Francois & Garrison for calculation of seawater absorption coefficient (Francois and Garrison, 1982a, 1982b) and UNESCO equation for sound speed calculation (Wong and Zhu, 1995).

As it can be seen from Table 5.2, $AAF_{\mathrm{ref}_{h=17,k=2}} = 1$ since the distance between sensor 2 and HS 17 corresponds to $l_{\mathrm{ref}} = 1.47\,m$.

		Sensor		
		1	2	3
Hotspot	1	0.4444	0.1520	0.2265
	2	0.5153	0.1528	0.2224
	3	0.5972	0.1533	0.2173
	4	0.4999	0.1780	0.3018
	5	0.6240	0.1796	0.2904
	6	0.8364	0.1805	0.2772
	7	0.3777	0.2104	0.4307
	8	0.4129	0.2133	0.3928
	9	0.3410	0.2244	0.5006
	10	0.3642	0.2281	0.4394
	11	0.2190	0.3634	0.5184
	12	0.2232	0.3863	0.4503
	13	0.2059	0.4093	0.4446
	14	0.2092	0.4465	0.4026
	15	0.1778	0.5664	0.3181
	16	0.1797	0.7304	0.3044
	17	0.1811	1.0000	0.2889
	18	0.1520	0.4955	0.2350
	19	0.1531	0.5805	0.2303
	20	0.1538	0.6663	0.2245

Table 5.2 Reference AAF for any sensor with respect to any hotspot

5.1.4 Vector of Possible Local Thresholds

A vector of the possible values of the local threshold (γ) must be created. Since the ET is the test the sensors will perform, it is reasonable that this vector has the following form:

$$[0, \dots, \gamma_{\max}]^T$$

Of course, the higher $\gamma_{\max}$ the easier it will be not to miss any potential optimal value of the local thresholds. Once $\gamma_{\max}$ is chosen, it will be necessary to make sure that intermediate values are in a sufficient number to be able to get an accurate final result.

5.1.5 Computation of Reference Performances and Indexes

This step is a loop that must be computed for any value of threshold present in the previously created vector.

It is required to calculate the local probabilities of detection and false alarm of any sensor given that the target may be located on any HS. This means that for any kth sensor, N_{HS} local probabilities of detection must be calculated and only one probability of false alarm (since the latter does not depend on the AAF).

$$
\begin{cases}
\text{for any } k = 1, \dots, N_K \\
\begin{cases}
P_{d,\mathrm{ref}_{h=1,k}}\big|_{\gamma=0} = 2Q\left(\sqrt{\dfrac{\gamma[=0]}{\sigma^2_{w,\mathrm{ref}} + \sigma^2_{\xi,\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=1,k}}}}\right) = 2Q\left(\sqrt{\dfrac{\gamma[=0]}{1 + SNR_{\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=1,k}}}}\right) \\
\vdots \\
P_{d,\mathrm{ref}_{h=\cdots,k}}\big|_{\gamma=0} = 2Q\left(\sqrt{\dfrac{\gamma[=0]}{\sigma^2_{w,\mathrm{ref}} + \sigma^2_{\xi,\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=\cdots,k}}}}\right) = 2Q\left(\sqrt{\dfrac{\gamma[=0]}{1 + SNR_{\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=\cdots,k}}}}\right) \\
\vdots \\
P_{d,\mathrm{ref}_{h=N_{HS},k}}\big|_{\gamma=0} = 2Q\left(\sqrt{\dfrac{\gamma[=0]}{\sigma^2_{w,\mathrm{ref}} + \sigma^2_{\xi,\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=N_{HS},k}}}}\right) = 2Q\left(\sqrt{\dfrac{\gamma[=0]}{1 + SNR_{\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=N_{HS},k}}}}\right) \\
P_{f,\mathrm{ref}_k}\big|_{\gamma=0} = 2Q\left(\sqrt{\dfrac{\gamma[=0]}{\sigma^2_{w,\mathrm{ref}}}}\right) = 2Q\left(\sqrt{\gamma[=0]}\right)
\end{cases} \\
\vdots \\
\begin{cases}
P_{d,\mathrm{ref}_{h=1,k}}\big|_{\gamma=\cdots} = 2Q\left(\sqrt{\dfrac{\gamma[=\cdots]}{\sigma^2_{w,\mathrm{ref}} + \sigma^2_{\xi,\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=1,k}}}}\right) = 2Q\left(\sqrt{\dfrac{\gamma[=\cdots]}{1 + SNR_{\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=1,k}}}}\right) \\
\vdots \\
P_{d,\mathrm{ref}_{h=\cdots,k}}\big|_{\gamma=\cdots} = 2Q\left(\sqrt{\dfrac{\gamma[=\cdots]}{\sigma^2_{w,\mathrm{ref}} + \sigma^2_{\xi,\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=\cdots,k}}}}\right) = 2Q\left(\sqrt{\dfrac{\gamma[=\cdots]}{1 + SNR_{\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=\cdots,k}}}}\right) \\
\vdots \\
P_{d,\mathrm{ref}_{h=N_{HS},k}}\big|_{\gamma=\cdots} = 2Q\left(\sqrt{\dfrac{\gamma[=\cdots]}{\sigma^2_{w,\mathrm{ref}} + \sigma^2_{\xi,\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=N_{HS},k}}}}\right) = 2Q\left(\sqrt{\dfrac{\gamma[=\cdots]}{1 + SNR_{\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=N_{HS},k}}}}\right) \\
P_{f,\mathrm{ref}_k}\big|_{\gamma=\cdots} = 2Q\left(\sqrt{\dfrac{\gamma[=\cdots]}{\sigma^2_{w,\mathrm{ref}}}}\right) = 2Q\left(\sqrt{\gamma[=\cdots]}\right)
\end{cases} \\
\vdots \\
\begin{cases}
P_{d,\mathrm{ref}_{h=1,k}}\big|_{\gamma=\gamma_{\max}} = 2Q\left(\sqrt{\dfrac{\gamma[=\gamma_{\max}]}{\sigma^2_{w,ref} + \sigma^2_{\xi,ref} \cdot AAF^2_{ref_{h=1,k}}}}\right) = 2Q\left(\sqrt{\dfrac{\gamma[=\gamma_{\max}]}{1 + SNR_{ref} \cdot AAF^2_{ref_{h=1,k}}}}\right) \\
\vdots \\
P_{d,\mathrm{ref}_{h=\cdots,k}}\big|_{\gamma=\gamma_{\max}} = 2Q\left(\sqrt{\dfrac{\gamma[=\gamma_{\max}]}{\sigma^2_{w,\mathrm{ref}} + \sigma^2_{\xi,\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=\cdots,k}}}}\right) = 2Q\left(\sqrt{\dfrac{\gamma[=\gamma_{\max}]}{1 + SNR_{\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=\cdots,k}}}}\right) \\
\vdots \\
P_{d,\mathrm{ref}_{h=N_{HS},k}}\big|_{\gamma=\gamma_{\max}} = 2Q\left(\sqrt{\dfrac{\gamma[=\gamma_{\max}]}{\sigma^2_{w,\mathrm{ref}} + \sigma^2_{\xi,\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=N_{HS},k}}}}\right) = 2Q\left(\sqrt{\dfrac{\gamma[=\gamma_{\max}]}{1 + SNR_{\mathrm{ref}} \cdot AAF^2_{\mathrm{ref}_{h=N_{HS},k}}}}\right) \\
P_{f,\mathrm{ref}_k}\big|_{\gamma=\gamma_{\max}} = 2Q\left(\sqrt{\dfrac{\gamma[=\gamma_{\max}]}{\sigma^2_{w,\mathrm{ref}}}}\right) = 2Q\left(\sqrt{\gamma[=\gamma_{\max}]}\right)
\end{cases}
\end{cases}
$$

Once all these values are gathered, it is possible to apply one of the indexes X for the selection of a threshold (where X can be J, d^2, CZ which were introduced in the section about threshold selection). Though, for any sensor, N_{HS} local probabilities of detection were calculated. As a consequence, N_{HS} different values of X will be obtained for any sensor: each of them considering a different HS. In this work, all the HS present in the manifold have been assumed to have an equal probability of failure. For this reason, all the values of X obtained for a single sensor (N_{HS} values) can be reduced to a single $\overline{X}$ value using an arithmetic mean:

$$\begin{cases} \overline{J_{\mathrm{ref}_k}}\big|_{\gamma=0} = \dfrac{\sum_{h=1}^{N_{HS}}\left[P_{d,\mathrm{ref}_{h,k}}\big|_{\gamma=0} - P_{f,\mathrm{ref}_k}\big|_{\gamma=0}\right]}{N_{HS}} \\ \vdots \\ \overline{J_{\mathrm{ref}_k}}\big|_{\gamma=\dots} = \dfrac{\sum_{h=1}^{N_{HS}}\left[P_{d,\mathrm{ref}_{h,k}}\big|_{\gamma=\dots} - P_{f,\mathrm{ref}_k}\big|_{\gamma=\dots}\right]}{N_{HS}} \\ \vdots \\ \overline{J_{\mathrm{ref}_k}}\big|_{\gamma=\gamma_{\max}} = \dfrac{\sum_{h=1}^{N_{HS}}\left[P_{d,\mathrm{ref}_{h,k}}\big|_{\gamma=\gamma_{\max}} - P_{f,\mathrm{ref}_k}\big|_{\gamma=\gamma_{\max}}\right]}{N_{HS}} \end{cases} \quad \text{or weighted means (if possible)}$$

$$\begin{cases} \overline{d^2{}_{\mathrm{ref}_k}}\big|_{\gamma=0} = \dfrac{\sum_{h=1}^{N_{HS}}\left[\left(1-P_{d,\mathrm{ref}_{h,k}}\big|_{\gamma=0}\right)^2 + \left(P_{f,\mathrm{ref}_k}\big|_{\gamma=0}\right)^2\right]}{N_{HS}} \\ \vdots \\ \overline{d^2{}_{\mathrm{ref}_k}}\big|_{\gamma=\dots} = \dfrac{\sum_{h=1}^{N_{HS}}\left[\left(1-P_{d,\mathrm{ref}_{h,k}}\big|_{\gamma=\dots}\right)^2 + \left(P_{f,\mathrm{ref}_k}\big|_{\gamma=\dots}\right)^2\right]}{N_{HS}} \\ \vdots \\ \overline{d^2{}_{\mathrm{ref}_k}}\big|_{\gamma=\gamma_{\max}} = \dfrac{\sum_{h=1}^{N_{HS}}\left[\left(1-P_{d,\mathrm{ref}_{h,k}}\big|_{\gamma=\gamma_{\max}}\right)^2 + \left(P_{f,\mathrm{ref}_k}\big|_{\gamma=\gamma_{\max}}\right)^2\right]}{N_{HS}} \end{cases} \quad \text{or weighted means (if possible)}$$

$$\begin{cases} \overline{CZ_{\mathrm{ref}_k}}\big|_{\gamma=0} = \dfrac{\sum_{h=1}^{N_{HS}}\left[P_{d,\mathrm{ref}_{h,k}}\big|_{\gamma=0} \cdot \left(1-P_{f,\mathrm{ref}_k}\big|_{\gamma=0}\right)\right]}{N_{HS}} \\ \vdots \\ \overline{CZ_{\mathrm{ref}_k}}\big|_{\gamma=\dots} = \dfrac{\sum_{h=1}^{N_{HS}}\left[P_{d,\mathrm{ref}_{h,k}}\big|_{\gamma=\dots} \cdot \left(1-P_{f,\mathrm{ref}_k}\big|_{\gamma=\dots}\right)\right]}{N_{HS}} \\ \vdots \\ \overline{CZ_{\mathrm{ref}_k}}\big|_{\gamma=\gamma_{\max}} = \dfrac{\sum_{h=1}^{N_{HS}}\left[P_{d,\mathrm{ref}_{h,k}}\big|_{\gamma=\gamma_{\max}} \cdot \left(1-P_{f,\mathrm{ref}_k}\big|_{\gamma=\gamma_{\max}}\right)\right]}{N_{HS}} \end{cases} \quad \text{or weighted means (if possible)}$$

5.1.6 Selection of Optimal Local Threshold

Now, for any sensor, there exists a set of value of $\overline{X}$ (X may be any of the three indices indicated before), this set has its cardinality equal to the number of components of $[0, \dots, \gamma_{\max}]^T$ since values of $\overline{X}$ has been generated from the different values of γ.

The selected local threshold will be the one yielding the highest value $\overline{X}$ among all those generated (highest value in case of J and CZ, lowest in case of d^2).

For any sensor $k = 1, \dots, N_K$, the optimal local threshold will be:

$$\gamma_k = \gamma\left(\max_{\gamma=0,\dots,\gamma_{\max}}\left\{\overline{J_{\mathrm{ref}_k}}\big|_{\gamma}\right\}\right)$$

$$\gamma_k = \gamma\left(\min_{\gamma=0,\dots,\gamma_{\max}}\left\{\overline{d^2{}_{\mathrm{ref}_k}}\big|_{\gamma}\right\}\right)$$

$$\gamma_k = \gamma\left(\max_{\gamma=0,\dots,\gamma_{\max}}\left\{\overline{CZ_{\mathrm{ref}_k}}\big|_{\gamma}\right\}\right)$$

Table 5.3 summarizes the obtained thresholds with the three different indexes:

		Sensor		
		1	2	3
J_{ref}	Index value	0.2566	0.2735	0.2687
	Threshold	1.8336	1.9239	1.7378
CZ_{ref}	Index value	0.3799	0.3895	0.3894
	Threshold	1.0346	1.0921	1.0592
$d^2{}_{\text{ref}}$	Index value	0.3116	0.3038	0.2906
	Threshold	0.8942	0.9217	0.9432

Table 5.3 Optimal local thresholds for ET according to different indexes

These values were obtained with a threshold vector $[0, \dots, \gamma_{\max}]^T$ where the distance between two consecutive elements was 0.00001.

It is worth to remind that reported values are normalized values since $\sigma_{w,k}^2 = 1$ for each sensor (as the sensors are the same model); this means those values reported in Table 5.3 are not the real values but, once known the value of thermal noise associated to the receivers, it is possible to know the real values of thresholds that are going to be used in the LDS.

5.2 Fusion Center Tuning

Unlike the proposed method for sensor tuning, in this phase, in order to compute the global probabilities at different threshold values, a brute force method is proposed (Candy and Breitfeller, 2013). The reason is that the possibility of using equations in a closed-form is not always possible, and using a brute force method makes it easier to change input parameters without having to elaborate a new mathematical expression.

The methods change according to the kind of FR the FC is expected to perform in operating conditions: the CR or the WFR. Both methods start with the simulation of a set of negative events and a set of positive events by generating samples of reference signals. The choice of FC global threshold will be based on the analysis of different performances yielded by continuously trying new values of the global threshold. FC tuning encompasses the sensor tuning phase, from which specific results must be retrieved.

The results shown in the next sections are based on the results and method explained in the previous chapter.

5.2.1 Tuning of a Fusion Center Performing the Counting Rule

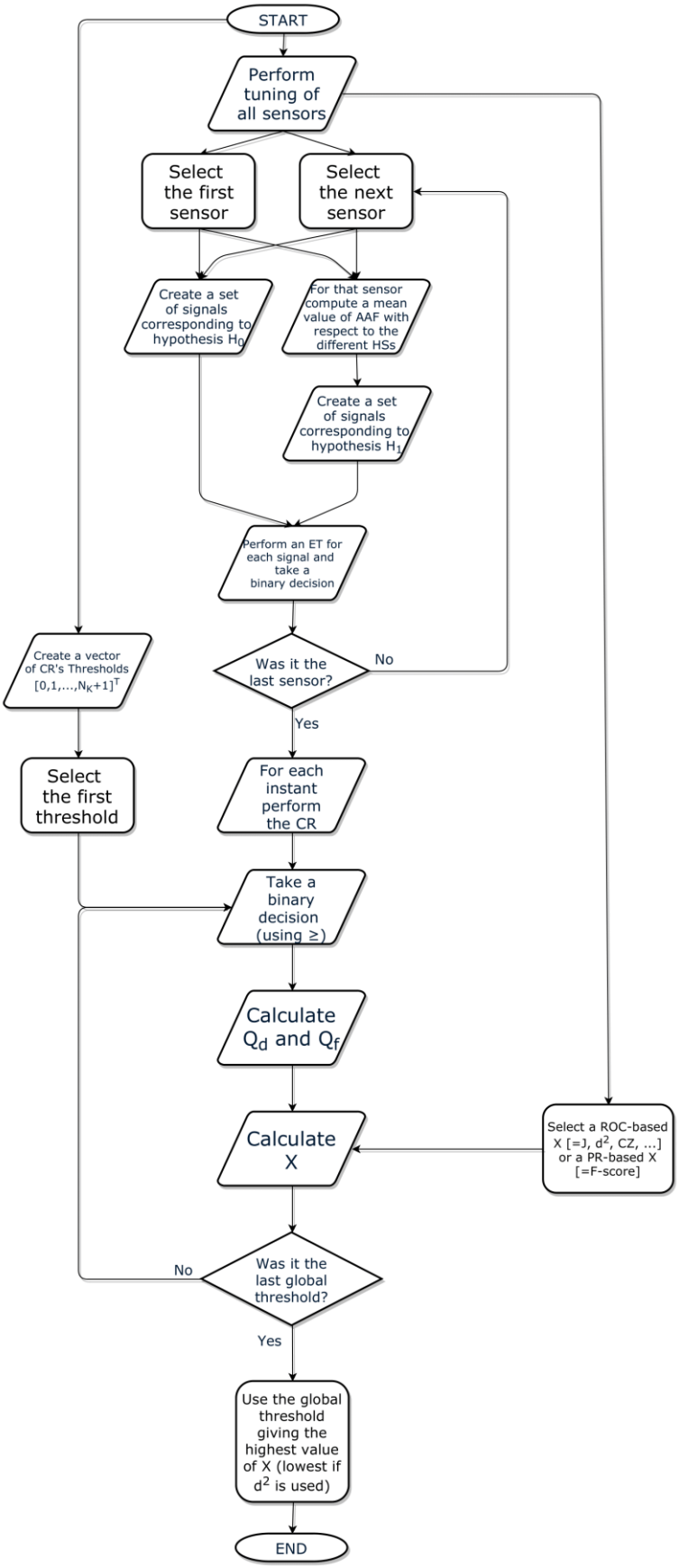


Figure 5.3 Flowchart of the algorithm for tuning of a FC using CR

5.2.1.1 Calculation of Reference AAFs

Before the procedure starts, it is required that the tuning of the sensors has already been carried out, and its results are available.

For any sensor, N_{HS} values of AAF were calculated. In order to have a performing algorithm (from a computational perspective), it was decided to calculate a global value of AAF for any sensor instead of iterating all the steps for any different HS. For any sensor, this global value is calculated by averaging all the AAFs obtained accounting for the different HSs. The average should take into consideration the probability of failure of the equipment associated with the HSs but, as this information is not available now, the arithmetic mean will be performed:

$$\overline{AAF_{\text{ref}_k}} = \frac{\sum_{h=1}^{N_{HS}} AAF_{\text{ref}_{h,k}}}{N_{HS}} \text{ or weighted mean (if possible)}$$

Table 5.4 shows the result of the previous equation based on values shown in Table 5.2:

	Sensor		
	1	2	3
$\overline{\boldsymbol{AAF}_{\text{ref}}}$	0.3434	0.3758	0.3358

Table 5.4 Reference AAFs for any sensor

5.2.1.2 Generation of Samples of Signals

It is required to create two sets of signals for any sensor. Every set corresponds to a class of events (positive and negative). Both sets will be made of t_{test} consecutive instants represented by t_{test} signal amplitudes. The value of t_{test} must be sufficiently high to guarantee a final solution as accurate as possible with respect to the theoretical one.

$$\begin{cases} H_1: & y_{k,t} = \xi_{k,t} \cdot \overline{AAF_k} + w_{k,t} & t = 1, \dots, t_{\text{test}} \\ H_0: & y_{k,t} = w_{k,t} & t = t_{\text{test}} + 1, \dots, 2t_{\text{test}} \end{cases}$$

$$\text{Where} \begin{cases} \xi_k \sim \mathcal{N}(0, SNR) \\ w_k \sim \mathcal{N}(0,1) \end{cases}$$

Of course, other pdfs can be used, but this also means that the equations used during the sensor tuning should be changed accordingly, since they were based on the same pdfs written above.

5.2.1.3 Sensor Test and Decision

Each sensor has to perform the ET on each signal of its sets (one made of signals belonging to positive events, the other of signals belonging to negative events) and make a local binary decision based on the threshold calculated after tuning the sensors (Table 5.3).

For each sensor $k = 1, \dots, N_K$:

$$d_{k,t} = \begin{cases} 1, & y_{k,t}^2 \geq \gamma_k \\ 0, & y_{k,t}^2 < \gamma_k \end{cases} \quad t = 1, \dots, 2t_{\text{test}}$$

5.2.1.4 Fusion Rule

A FR is applied to summarize all the local decisions simulating the behavior of the FC. In this case, the CR is used.

For each instant $t = 1, \dots, 2t_{\text{test}}$:

$$\Lambda_{\mathrm{CR}_t} = \sum_{k=1}^{N_K} d_{k,t}$$

This step creates a vector of $2t_{\mathrm{test}}$ elements.

5.2.1.5 Vector of Possible Global Thresholds

Due to the nature of the CR, the vector containing all the possible values of the global threshold will be:

$$[0,1,\dots,N_K+1]^T$$

These elements are those values of global threshold $\overline{\gamma}$ sufficient to create a complete ROC curve.

5.2.1.6 Global Decision

For any value of the global threshold $\overline{\gamma}$, a global decision d_{FC} for each instant will be taken based on the results of the CR previously obtained.

For each instant $t = 1,\dots,2t_{\mathrm{test}}$:

$$\begin{cases} d_{\mathrm{FC}_t}\big|_{\overline{\gamma}=0} = \begin{cases} 1, & \Lambda_{\mathrm{FC}_t} \geq \overline{\gamma}[=0] \\ 0, & \Lambda_{\mathrm{FC}_t} < \overline{\gamma}[=0] \end{cases} \\ d_{\mathrm{FC}_t}\big|_{\overline{\gamma}=1} = \begin{cases} 1, & \Lambda_{\mathrm{FC}_t} \geq \overline{\gamma}[=1] \\ 0, & \Lambda_{\mathrm{FC}_t} < \overline{\gamma}[=1] \end{cases} \\ \vdots \\ d_{\mathrm{FC}_t}\big|_{\overline{\gamma}=N_K+1} = \begin{cases} 1, & \Lambda_{\mathrm{FC}_t} \geq \overline{\gamma}[=N_K+1] \\ 0, & \Lambda_{\mathrm{FC}_t} < \overline{\gamma}[=N_K+1] \end{cases} \end{cases}$$

Note that it is possible to use this alternative version: $d_{\mathrm{FC}_t}\big|_{\overline{\gamma}} = \begin{cases} 1, & \Lambda_{\mathrm{FC}_t} > \overline{\gamma} \\ 0, & \Lambda_{\mathrm{FC}_t} \leq \overline{\gamma} \end{cases}$

In that case, the threshold vector should be $[-1,0,\dots,N_K]^T$.

5.2.1.7 Performance Evaluation and Threshold Selection

After computing the previous loop, results generated using the different threshold must be evaluated: global probabilities of detection and global probabilities of false alarm have to be calculated for any threshold. These couples of values can then be graded using either one of the indexes shown during the tuning of the sensors or a PR-based index like the F-score (F_β). However, PR-based indexes should be used only when the disproportion between instants in the positive and negative classes is close to the real scenario, in this case t_{test} instants were used for both scenarios, meaning that a ROC-based index (J, d^2, or CZ) is more suitable.

$$\begin{cases} \begin{cases} Q_{d,\mathrm{ref}}\big|_{\overline{\gamma}=0} = \dfrac{\sum_{t=1}^{t_{\mathrm{test}}}\left[d_{\mathrm{FC}_t}\big|_{\overline{\gamma}=0}\right]}{t_{\mathrm{test}}} \\ Q_{d,\mathrm{ref}}\big|_{\overline{\gamma}=1} = \dfrac{\sum_{t=1}^{t_{\mathrm{test}}}\left[d_{\mathrm{FC}_t}\big|_{\overline{\gamma}=1}\right]}{t_{\mathrm{test}}} \\ \vdots \\ Q_{d,\mathrm{ref}}\big|_{\overline{\gamma}=N_K+1} = \dfrac{\sum_{t=1}^{t_{\mathrm{test}}}\left[d_{\mathrm{FC}_t}\big|_{\overline{\gamma}=N_K+1}\right]}{t_{\mathrm{test}}} \end{cases} \\ \begin{cases} Q_{f,\mathrm{ref}}\big|_{\overline{\gamma}=0} = \dfrac{\sum_{t_{\mathrm{test}}+1}^{2t_{\mathrm{test}}}\left[d_{\mathrm{FC}_t}\big|_{\overline{\gamma}=0}\right]}{t_{\mathrm{test}}} \\ Q_{f,\mathrm{ref}}\big|_{\overline{\gamma}=1} = \dfrac{\sum_{t_{\mathrm{test}}+1}^{2t_{\mathrm{test}}}\left[d_{\mathrm{FC}_t}\big|_{\overline{\gamma}=1}\right]}{t_{\mathrm{test}}} \\ \vdots \\ Q_{f,\mathrm{ref}}\big|_{\overline{\gamma}=N_K+1} = \dfrac{\sum_{t_{\mathrm{test}}+1}^{t_{\mathrm{test}}}\left[d_{\mathrm{FC}_t}\big|_{\overline{\gamma}=N_K+1}\right]}{t_{\mathrm{test}}} \end{cases} \end{cases}$$

From Figure 5.4, the ROC curves generated by applying the CR can be seen. The difference among these curves is derived from the different indexes used to select the optimal local threshold for any sensor. The local thresholds used for the sensors, in fact, are those reported in Table 5.3. Values were obtained assuming $t_{\mathrm{test}} = 5000$.

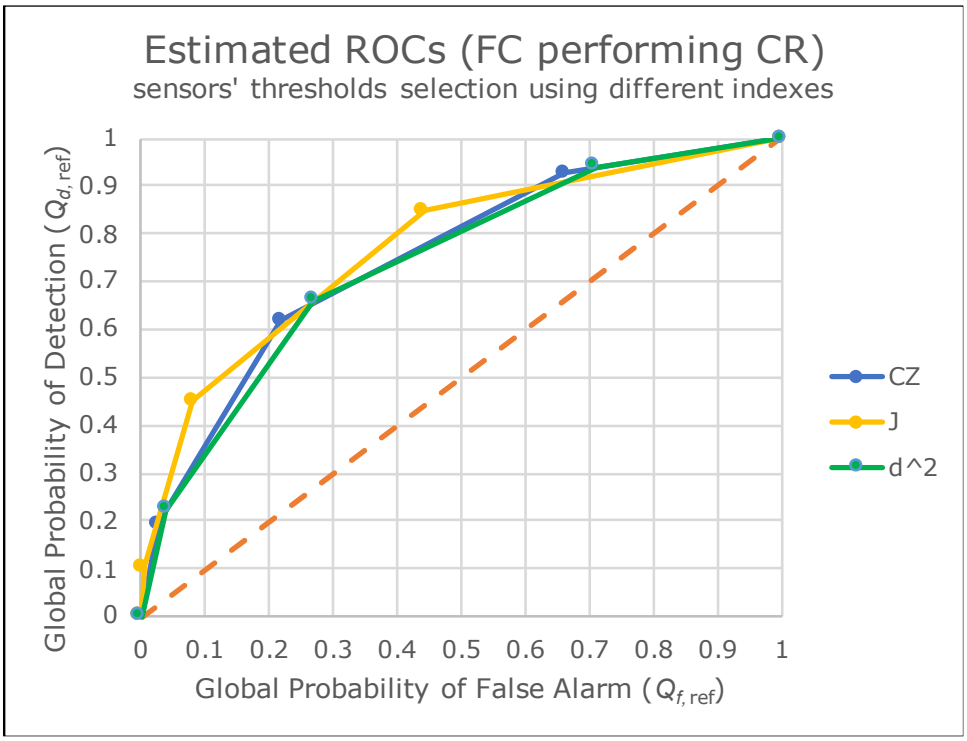


Figure 5.4 ROC curves of the performance test carried out during the FC tuning (in case it will perform the CR). Different colors represent different indexes used for sensor tuning.

The indexes must be calculated for any threshold, which means that nine results of possible optimal global thresholds will be calculated (because the three indexes will be applied for each point of the three ROC curves). However, as the CR applied was applied, the number of possible optimal global thresholds will be undoubtedly lower than nine.

$$\begin{cases} J_{\text{ref}}|_{\overline{\gamma}=0} = Q_{d,\text{ref}}|_{\overline{\gamma}=0} - Q_{f,\text{ref}}|_{\overline{\gamma}=0} \\ J_{\text{ref}}|_{\overline{\gamma}=1} = Q_{d,\text{ref}}|_{\overline{\gamma}=1} - Q_{f,\text{ref}}|_{\overline{\gamma}=1} \\ \vdots \\ J_{\text{ref}}|_{\overline{\gamma}=N_K+1} = Q_{d,\text{ref}}|_{\overline{\gamma}=N_K+1} - Q_{f,\text{ref}}|_{\overline{\gamma}=N_K+1} \end{cases}$$

$$\begin{cases} {d^2}_{\text{ref}}|_{\overline{\gamma}=0} = \left(1 - Q_{d,\text{ref}}|_{\overline{\gamma}=0}\right)^2 + \left(Q_{f,\text{ref}}|_{\overline{\gamma}=0}\right)^2 \\ {d^2}_{\text{ref}}|_{\overline{\gamma}=1} = \left(1 - Q_{d,\text{ref}}|_{\overline{\gamma}=1}\right)^2 + \left(Q_{f,\text{ref}}|_{\overline{\gamma}=1}\right)^2 \\ \vdots \\ {d^2}_{\text{ref}}|_{\overline{\gamma}=N_K+1} = \left(1 - Q_{d,\text{ref}}|_{\overline{\gamma}=N_K+1}\right)^2 + \left(Q_{f,\text{ref}}|_{\overline{\gamma}=N_K+1}\right)^2 \end{cases}$$

$$\begin{cases} CZ_{\text{ref}}|_{\overline{\gamma}=0} = Q_{d,\text{ref}}|_{\overline{\gamma}=0} \cdot \left(1 - Q_{f,\text{ref}}|_{\overline{\gamma}=0}\right) \\ CZ_{\text{ref}}|_{\overline{\gamma}=1} = Q_{d,\text{ref}}|_{\overline{\gamma}=1} \cdot \left(1 - Q_{f,\text{ref}}|_{\overline{\gamma}=1}\right) \\ \vdots \\ CZ_{\text{ref}}|_{\overline{\gamma}=N_K+1} = Q_{d,\text{ref}}|_{\overline{\gamma}=N_K+1} \cdot \left(1 - Q_{f,\text{ref}}|_{\overline{\gamma}=N_K+1}\right) \end{cases}$$

The chosen global threshold ($\overline{\gamma}_{\text{FC}}$) will be the one which maximizes one of the indexes (minimizes in case d^2 is chosen):

$$\overline{\gamma}_{\text{FC}} = \overline{\gamma}\left(\max_{\gamma=0,1,\ldots,N_K+1}\{J_{\text{ref}}|_{\overline{\gamma}}\}\right)$$

$$\overline{\gamma}_{\text{FC}} = \overline{\gamma}\left(\min_{\gamma=0,1,\ldots,N_K+1}\{{d^2}_{\text{ref}}|_{\overline{\gamma}}\}\right)$$

$$\overline{\gamma}_{\text{FC}} = \overline{\gamma}\left(\max_{\gamma=0,1,\ldots,N_K+1}\{CZ_{\text{ref}}|_{\overline{\gamma}}\}\right)$$

5.2.1.8 Results

- Results in Case J is used for Local Threshold Selection

In Table 5.5, performances of the FC are reported in case the tuning of the sensors was carried out applying J index.

In this scenario, regardless of the index used for the FC tuning, $\overline{\gamma} = 1$ is the optimal local threshold (as in the previous case). The point on the ROC curve can be graphically seen in Figure 5.5.

		Performances		**Index**			**AUC**
		$Q_{d,\text{ref}}$	$Q_{f,\text{ref}}$	J_{ref}	d^2	CZ	
Threshold	0	1	1	0	1	0	0.7704
	1	**0.8480**	**0.4431**	**0.4049**	**0.2194**	**0.4723**	
	2	0.4496	0.0816	0.3680	0.3096	0.4129	
	3	0.1018	0.0055	0.0963	0.8067	0.1013	
	4	0	0	0	1	0	

Table 5.5 FC performances (in case it will perform the CR) at different thresholds evaluated by *J*, d^2, and *CZ*. Optimal global threshold is reported in bold. These results are generated when sensor local thresholds have been selected using *J*.

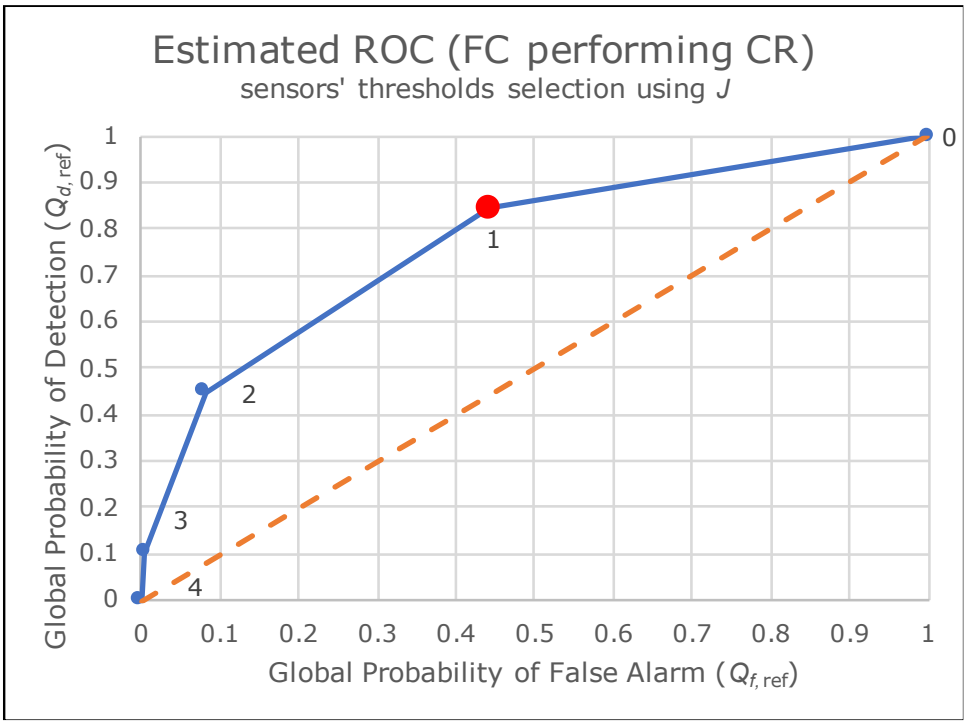


Figure 5.5 ROC curve of the performance test carried out during the tuning of the FC (in case it will perform the CR) when local thresholds have been selected using *J*. Red dot on the curve is the optimal global threshold.

- Results in Case d^2 is used for Local Threshold Selection

in Table 5.6, performances of the FC are reported in case the tuning of the sensors was carried out applying d^2 index, are reported.

In this scenario, regardless of the index used for the FC tuning, $\overline{\gamma} = 2$ is the optimal threshold. The point on the ROC curve can be graphically seen in Figure 5.6.

		Performances		**Index**			**AUC**
		$Q_{d,\mathrm{ref}}$	$Q_{f,\mathrm{ref}}$	J_{ref}	${d^2}_{\mathrm{ref}}$	CZ	
Threshold	0	1	1	0	1	0	0.7393
	1	0.9397	0.7102	0.2296	0.5080	0.2724	
	2	**0.6591**	**0.2689**	**0.3902**	**0.1885**	**0.4819**	
	3	0.2236	0.0401	0.1835	0.6044	0.2146	
	4	0	0	0	1	0	

Table 5.6 FC performances (in case it will perform the CR) at different thresholds evaluated by *J*, d^2, and *CZ*. The optimal global threshold is reported in bold. These results are generated when sensor local thresholds have been selected using d^2.

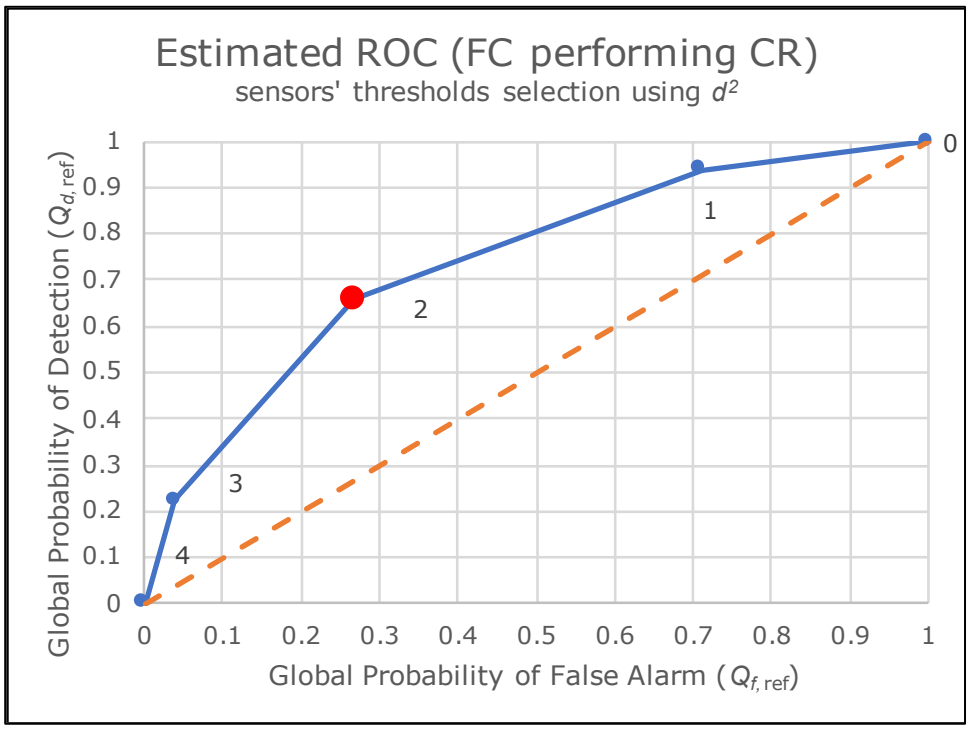


Figure 5.6 ROC curve of the performance test carried out during the tuning of the FC (in case it will perform the CR) when local thresholds have been selected using d^2. Red dot on the curve is the optimal global threshold.

- Results in Case CZ is used for Local Threshold Selection

In Table 5.7, performances of the FC, in case the tuning of the sensors was carried out applying CZ index, are reported.

In this scenario, regardless of the index used for the FC tuning, $\overline{\gamma} = 2$ is the optimal one (as in the previous case). The point on the ROC curve can be graphically seen in Figure 5.7.

		Performances		**Index**			**AUC**
		$Q_{d,\mathrm{ref}}$	$Q_{f,\mathrm{ref}}$	J_{ref}	${d^2}_{\mathrm{ref}}$	CZ_{ref}	
Threshold	0	1	1	0	1	0	0.7473
	1	0.9247	0.6628	0.2619	0.4450	0.3118	
	2	**0.6192**	**0.2192**	**0.4000**	**0.1931**	**0.4834**	
	3	0.1912	0.0272	0.1640	0.6549	0.1860	
	4	0	0	0	1	0	

Table 5.7 FC performances (in case it will perform the CR) at different thresholds evaluated by *J*, d^2, and *CZ*. The optimal global threshold is reported in bold. These results are generated when sensor local thresholds have been selected using *CZ*.

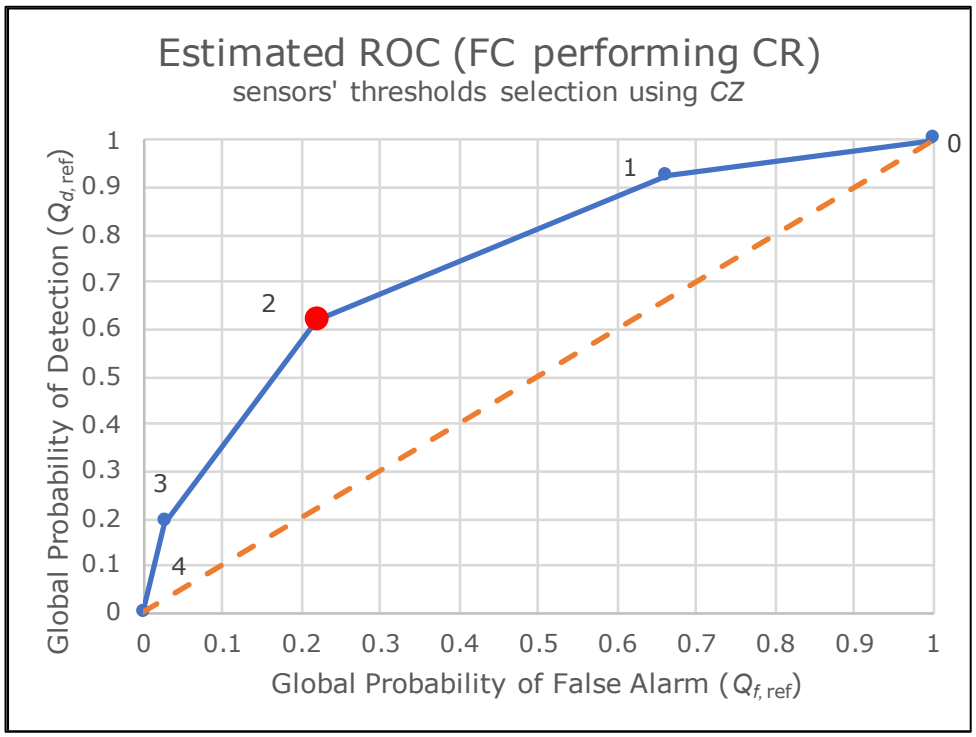


Figure 5.7 ROC curve of the performance test carried out during the tuning of the FC (index in case it will perform the CR) when local thresholds have been selected using *CZ*. Red dot on the curve is the optimal global threshold.

5.2.2 Tuning of a Fusion Center Performing the Weighted Fusion Rule

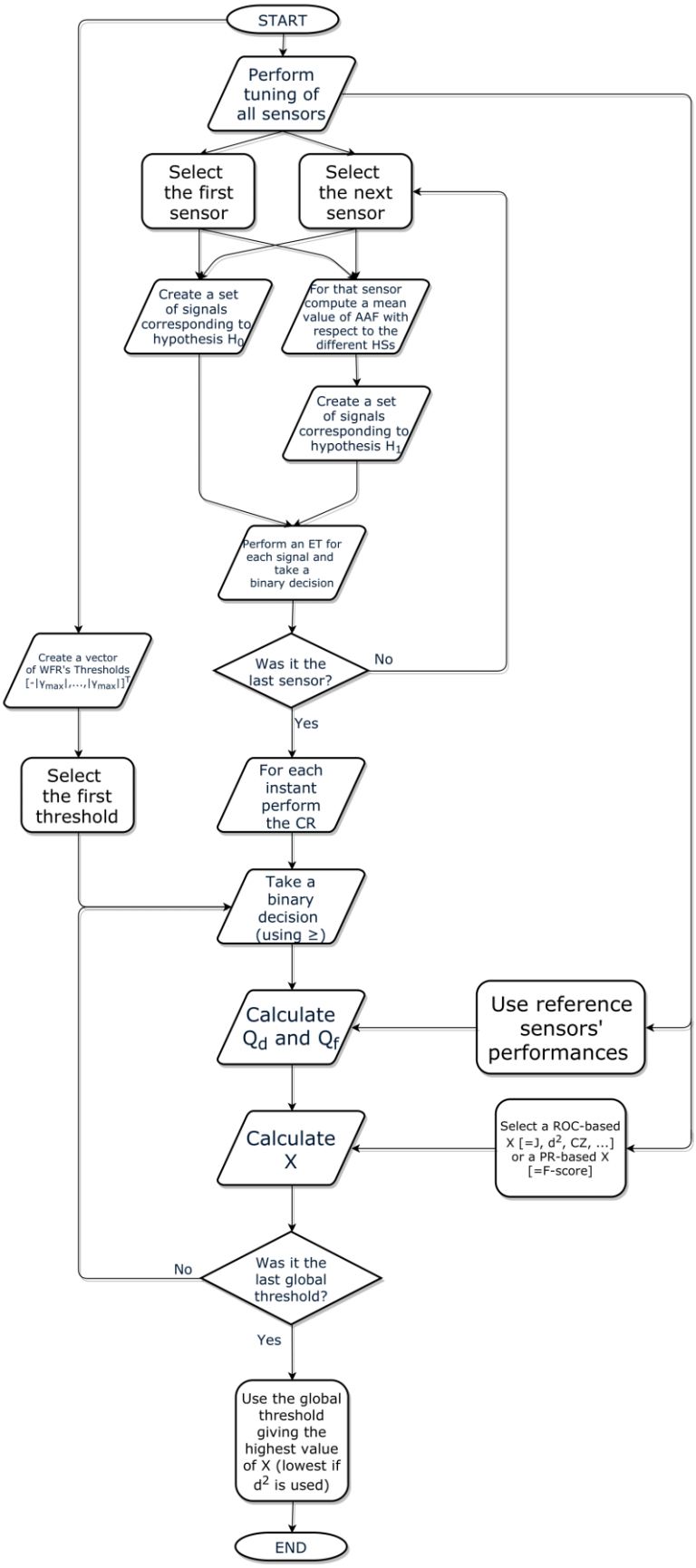


Figure 5.8 Flowchart of the algorithm for tuning of a FC using WFR

The algorithm for the tuning of a FC that is expected to perform a WFR is the same as the one used in case of CR, except for some steps:

- A different FR is applied when testing any possible value of the global threshold (the WFR instead of the CR).

- The possible values of the global threshold are different.
  Unlike the CR which produces a non-negative integer number as an outcome, the WFR produces a real number as a result (either positive or negative). For this reason, like in the case of the sensor local threshold, a tentative vector must be provided.

  Given a value $\overline{\gamma}_{\max}$, the vector will be:

  $$\left[-\left|\overline{\gamma}_{\max}\right|, \dots, \left|\overline{\gamma}_{\max}\right|\right]^T$$

  Also in this case, the higher $\left|\overline{\gamma}_{\max}\right|$ the easier it will be not to miss any potential optimal value of the global threshold. Also, the higher the number of elements present in this vector, the easier it will be to have an accurate final result.

- When the WFR is applied, it is necessary to know the sensor performances; this, as already explained, cannot be theoretically done because the leakage position is unknown (leakage point is a NCT). To overcome this issue, average reference performances for every sensor are considered. These performances can be obtained from the tuning of the sensors.

  When performances of each threshold for any sensor were calculated (during the tuning of the sensors), for each threshold there were N_{HS} values of reference local probability of detection and one value of reference local probability of false alarm.

  The proposed solution is to gather all the performances of any sensor that are associated with its selected optimal local threshold and calculate a global value of performances:

  For any $k = 1, \dots, N_K$

  $$\begin{cases} P_{d,FC_k} = \dfrac{\Sigma_{h=1}^{N_{HS}}\left(P_{d,\mathrm{ref}_{h,k}}\Big|_{\gamma_k}\right)}{N_{HS}} \\ P_{f,FC_k} = P_{f,\mathrm{ref}_k}\Big|_{\gamma_k} \end{cases} \text{ or weighted mean (if possible)}$$

  These values of probability of detection are arithmetic means of all the probabilities calculated with respect to different HSs. The arithmetic mean assumes that all the HSs are equally likely to be leaking sources. Again, a weighted average based on equipment failure probabilities as weights would be more suitable. The performances used are shown in Table 5.8 where different performances were calculated according to the index used for the selection of the local threshold.

		Sensor 1			
		$P_{d,\mathrm{FC}}$	$P_{f,\mathrm{FC}}$	$\ln\left(\frac{P_{d,\mathrm{FC}}}{P_{f,\mathrm{FC}}}\right)$	Index value
Index used	$\boldsymbol{J}$	0.4323	0.1757	0.9002	$J = 0.2566$
	$\boldsymbol{CZ}$	0.5499	0.3091	0.5761	$CZ = 0.3563$
	$\boldsymbol{d^2}$	0.5774	0.3443	0.5169	$d^2 = 0.3532$
		Sensor 2			
		$P_{d,\mathrm{FC}}$	$P_{f,\mathrm{FC}}$	$\ln\left(\frac{P_{d,FC}}{P_{f,\mathrm{FC}}}\right)$	Index value
Index used	$\boldsymbol{J}$	0.4389	0.1654	0.9758	$J = 0.2735$
	$\boldsymbol{CZ}$	0.5533	0.2960	0.6255	$CZ = 0.3663$
	$\boldsymbol{d^2}$	0.5847	0.3370	0.5510	$d^2 = 0.3422$
		Sensor 3			
		$P_{d,\mathrm{FC}}$	$P_{f,\mathrm{FC}}$	$\ln\left(\frac{P_{d,\mathrm{FC}}}{P_{f,\mathrm{FC}}}\right)$	Index value
Index used	$\boldsymbol{J}$	0.4561	0.1874	0.8893	$J = 0.2687$
	$\boldsymbol{CZ}$	0.5591	0.3034	0.6112	$CZ = 0.3706$
	$\boldsymbol{d^2}$	0.5812	0.3315	0.5615	$d^2 = 0.3310$

Table 5.8 Reference performances that are used for FC tuning in case it will perform the WFR and are also used by the FC for either detection or localization.

The WFR gives priority to that information coming from sensors whose performance is higher but, it is visible (from Table 5.8) that, once selected a specific index for the selection of local thresholds, performances do not change too much among different sensors. This can be explained as the centroid of sensor positions is near the centroid of HS positions. Also, it must be accounted that these performances are yielded from average local indexes, making those differences among sensor performances even more insignificant; this can compromise the benefits of using the WFR as these performances will also be used by the FC during either real-time detection or localization in operating condition.

The equations related to this method are the following:

$$
\begin{cases}
\text{From sensor tuning} \begin{cases}
\overline{AAF_k} = \dfrac{\sum_{h=1}^{N_{HS}} AAF_{h,k}}{N_{HS}} \\
P_{d,\mathrm{FC}_k} = \dfrac{\sum_{h=1}^{N_{HS}} \left(P_{d,\mathrm{ref}_{h,k}} \Big|_{\gamma_k} \right)}{N_{HS}} \\
P_{f,\mathrm{FC}_k} = P_{f,\mathrm{ref}_k} \Big|_{\gamma_k}
\end{cases} \\
\text{Events simulation} \begin{cases}
\begin{cases}
H_1: & y_{k,t} = \xi_{k,t} \cdot \overline{AAF_k} + w_{k,t} & t = 1, \dots, t_{\mathrm{test}} \\
H_0: & y_{k,t} = w_{k,t} & t = t_{\mathrm{test}} + 1, \dots, 2t_{\mathrm{test}}
\end{cases} \\
\xi_k \sim \mathcal{N}(0, SNR) \\
w_k \sim \mathcal{N}(0,1)
\end{cases} \\
\text{Sensor detection and FR} \begin{cases}
d_{k,t} = \begin{cases} 1, & y_{k,t}^2 \geq \gamma_k \\ 0, & y_{k,t}^2 < \gamma_k \end{cases} \quad k = 1, \dots, N_K \\
\Lambda_{\mathrm{FC}_t} = \sum_{k=1}^{N_K} \left[d_{k,t} \cdot \ln\left(\dfrac{P_{d,\mathrm{FC}_k}}{P_{f,\mathrm{FC}_k}} \right) + (1 - d_{k,t}) \cdot \ln\left(\dfrac{1 - P_{d,\mathrm{FC}_k}}{1 - P_{f,\mathrm{FC}_k}} \right) \right]
\end{cases} t = 1, \dots, 2t_{\mathrm{test}} \\
\text{FC detection} \begin{cases}
d_{\mathrm{FC}_t} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} = \begin{cases} 1, & \Lambda_{\mathrm{FC}_t} \geq \overline{\gamma} [= -|\overline{\gamma}_{\max}|] \\ 0, & \Lambda_{\mathrm{FC}_t} < \overline{\gamma} [= -|\overline{\gamma}_{\max}|] \end{cases} \\
\vdots \qquad\qquad t = 1, \dots, 2t_{test} \\
d_{\mathrm{FC}_t} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} = \begin{cases} 1, & \Lambda_{\mathrm{FC}_t} \geq \overline{\gamma} [= |\overline{\gamma}_{\max}|] \\ 0, & \Lambda_{\mathrm{FC}_t} < \overline{\gamma} [= |\overline{\gamma}_{\max}|] \end{cases}
\end{cases} \\
\text{FC performances} \begin{cases}
\begin{cases}
Q_{d,\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} = \dfrac{\sum_{t=1}^{t_{\mathrm{test}}} \left[d_{\mathrm{FC}_t} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} \right]}{t_{\mathrm{test}}} \\
\vdots \\
Q_{d,\mathrm{ref}} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} = \dfrac{\sum_{t=1}^{t_{\mathrm{test}}} \left[d_{\mathrm{FC}_t} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} \right]}{t_{\mathrm{test}}}
\end{cases} \\
\begin{cases}
Q_{f,\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} = \dfrac{\sum_{t_{\mathrm{test}}+1}^{2t_{\mathrm{test}}} \left[d_{\mathrm{FC}_t} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} \right]}{t_{\mathrm{test}}} \\
\vdots \\
Q_{f,\mathrm{ref}} \big|_{\overline{\gamma} = |\overline{\gamma}_{test}|} = \dfrac{\sum_{t_{\mathrm{test}}+1}^{2t_{\mathrm{test}}} \left[d_{\mathrm{FC}_t} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} \right]}{t_{\mathrm{test}}}
\end{cases}
\end{cases} \\
\text{Threshold selection} \begin{cases}
\begin{cases}
J_{\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} = Q_{d,\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} - Q_{f,\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} \\
\vdots \\
J_{\mathrm{ref}} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} = Q_{d,\mathrm{ref}} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} - Q_{f,\mathrm{ref}} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} \\
\overline{\gamma}_{\mathrm{FC}} = \overline{\gamma} \left(\max_{\gamma = -|\overline{\gamma}_{\max}|, \dots, |\overline{\gamma}_{\max}|} \{ J_{\mathrm{ref}} \big|_{\overline{\gamma}} \} \right)
\end{cases} \\
\begin{cases}
{d^2}_{\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} = \left(1 - Q_{d,\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} \right)^2 + \left(Q_{f,\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} \right)^2 \\
\vdots \\
{d^2}_{\mathrm{ref}} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} = \left(1 - Q_{d,\mathrm{ref}} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} \right)^2 + \left(Q_{f,\mathrm{ref}} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} \right)^2 \\
\overline{\gamma}_{\mathrm{FC}} = \overline{\gamma} \left(\min_{\gamma = -|\overline{\gamma}_{\max}|, \dots, |\overline{\gamma}_{\max}|} \{ {d^2}_{\mathrm{ref}} \big|_{\overline{\gamma}} \} \right)
\end{cases} \\
\begin{cases}
CZ_{\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} = Q_{d,\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} \cdot \left(1 - Q_{f,\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} \right) \\
\vdots \\
CZ_{\mathrm{ref}} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} = Q_{d,\mathrm{ref}} \big|_{\overline{\gamma} = -|\overline{\gamma}_{\max}|} \cdot \left(1 - Q_{f,\mathrm{ref}} \big|_{\overline{\gamma} = |\overline{\gamma}_{\max}|} \right) \\
\overline{\gamma}_{\mathrm{FC}} = \overline{\gamma} \left(\max_{\gamma = -|\overline{\gamma}_{\max}|, \dots, |\overline{\gamma}_{\max}|} \{ CZ_{\mathrm{ref}} \big|_{\overline{\gamma}} \} \right)
\end{cases}
\end{cases}
\end{cases}
$$

5.2.2.1 Results

Figure 5.9 shows the ROC curves obtained during the performance evaluation carried out during the tuning of the FC. Also in this case, results are different according to what kind of index was used when selecting the optimal local threshold. Reported results were obtained using a threshold vector $[-|\bar{\gamma}_{\max}|, \dots, |\bar{\gamma}_{\max}|]^T$ where two consecutive values have distance equal to 0.001 and $t_{\text{test}} = 5000$.

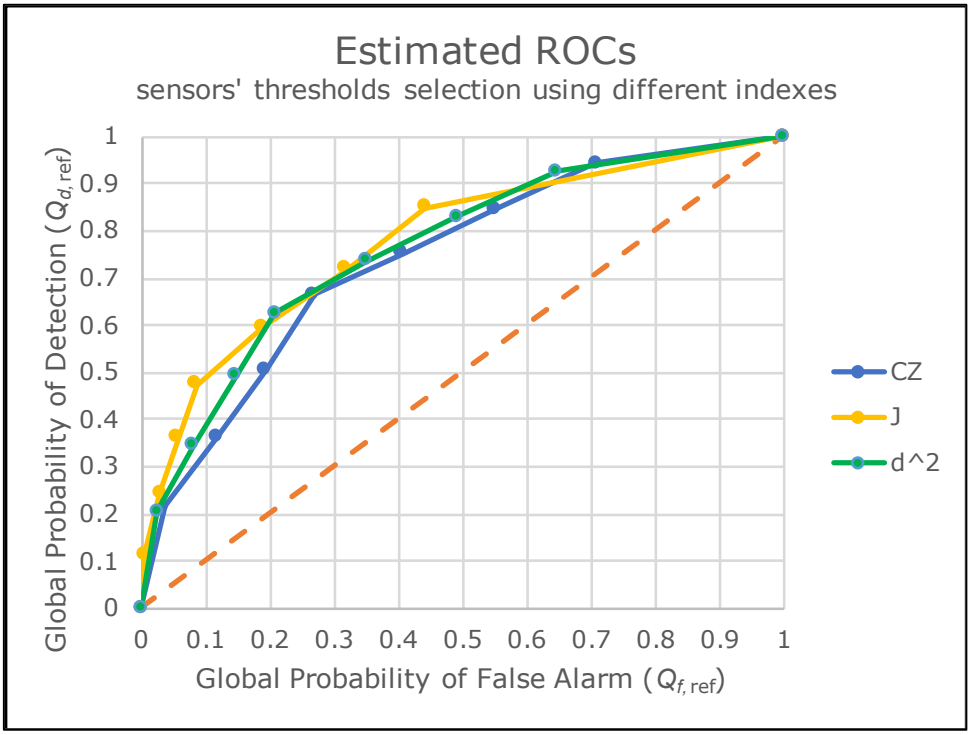


Figure 5.9 ROC curves of the performance test carried out during the FC tuning (in case it will perform the WFR). Different colors represent different indexes used for sensor tuning.

The analysis of the single curves will follow the same procedure seen for the tuning of a FC expected to perform the CR.

- Results in Case J is used for Local Threshold Selection

In Table 5.9, performances of the FC, in case the tuning of the sensors was carried out applying J index, are reported.

It can be seen that the optimal global threshold changes according to what index was used for the FC tuning: $\bar{\gamma} = 0.1200$ in case J is used and : $\bar{\gamma} = 0.1020$ in case d^2 or CZ is used. The results can be graphically seen in Figure 5.10, where the ROC curve is reported.

		Performances		Index			AUC
		$Q_{d,\mathrm{ref}}$	$Q_{f,\mathrm{ref}}$	J	d^2	CZ	
Threshold	-∞	1	1	0	1	0	0.7783
	-1.1710	0.8494	0.4417	0.4077	0.2178	0.4742	
	0.1020	**0.7196**	**0.3183**	**0.4013**	**0.1799**	**0.4906**	
	0.1200	**0.5980**	**0.1905**	**0.4075**	0.1979	0.4841	
	0.2020	0.4750	0.0857	0.3893	0.2830	0.4343	
	1.3930	0.3603	0.0573	0.3029	0.4125	0.3396	
	1.4750	0.2442	0.0306	0.2136	0.5722	0.2367	
	1.4930	0.1105	0.0048	0.1057	0.7913	0.1100	
	+∞	0	0	0	1	0	

Table 5.9 FC performances (in case it will perform the WFR) at different thresholds evaluated by *J*, d^2, and *CZ*. The optimal global thresholds are reported in bold. These results are generated when sensor local thresholds have been selected using *J*.

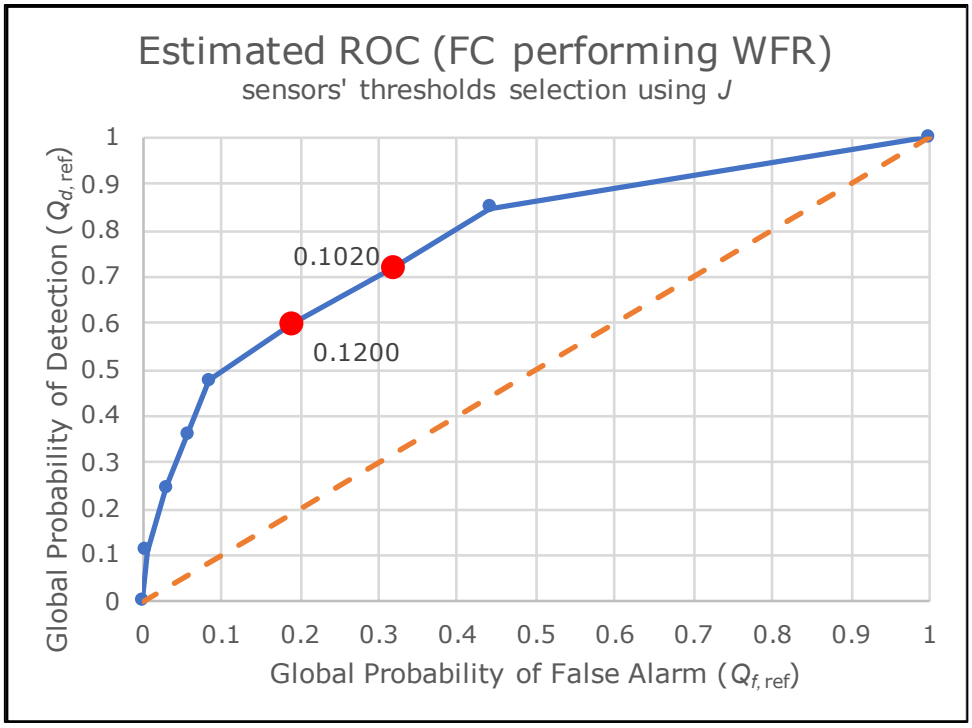


Figure 5.10 ROC curve of the performance test carried out during the tuning of the FC (in case it will perform the WFR) when local thresholds have been selected using *J*. Red dots on the curve are the optimal global thresholds.

- Results in Case d^2 is used for Local Threshold Selection

In Table 5.10, performances of FC, in case the tuning of the sensors was carried out applying d^2 index, are reported.

It can be seen that the optimal global threshold is the same for any index used for the FC tuning: $\overline{\gamma} = -0.3450$. Results can be graphically seen in Figure 5.11, where the ROC curve is reported.

		Performances		Index			AUC
		$Q_{d,\mathrm{ref}}$	$Q_{f,\mathrm{ref}}$	J_{ref}	d^2	CZ	
Threshold	-∞	1	1	0	1	0	0.7418
	-1.3740	0.9423	0.7084	0.2339	0.5052	0.2748	
	-0.4180	0.8478	0.5520	0.2958	0.3278	0.3798	
	-0.3550	0.7533	0.4060	0.3473	0.2257	0.4475	
	-0.3450	**0.6642**	**0.2693**	**0.3950**	**0.1853**	**0.4854**	
	0.6010	0.5059	0.1936	0.3122	0.2817	0.4079	
	0.6110	0.3626	0.1166	0.2460	0.4198	0.3203	
	0.6740	0.2162	0.0366	0.1796	0.6156	0.2083	
	+∞	0	0	0	1	0	

Table 5.10 FC performances (in case it will perform the WFR) at different thresholds evaluated by *J*, d^2, and *CZ*. The optimal global threshold is reported in bold. These results are generated when sensor local thresholds have been selected using d^2.

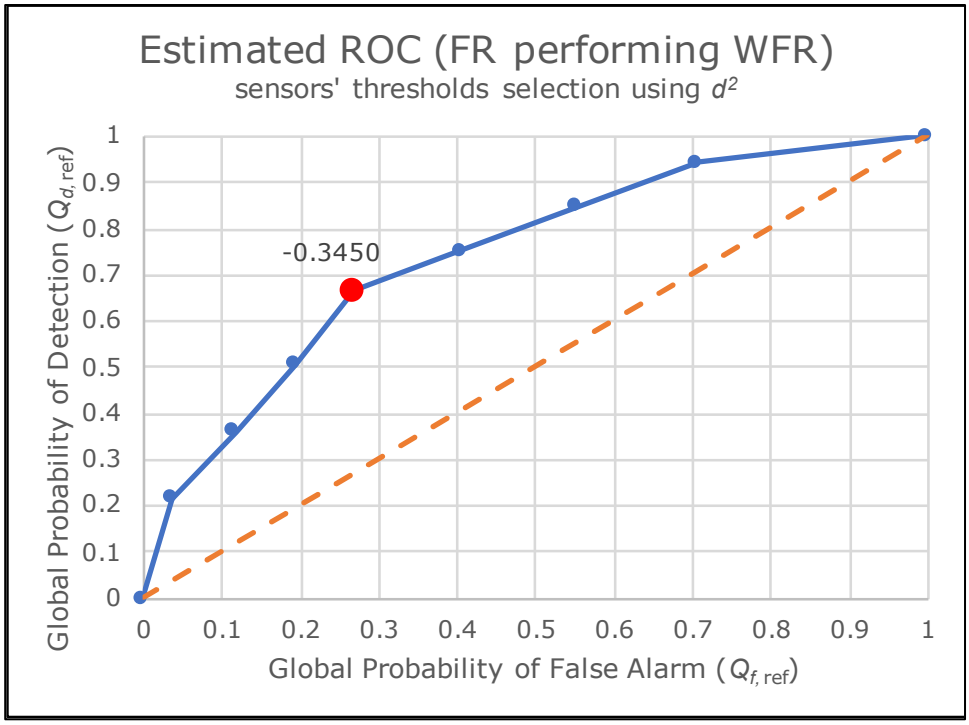


Figure 5.11 ROC curve of the performance test carried out during the tuning of the FC (in case it will perform the WFR) when local thresholds have been selected using d^2. Red dot on the curve is the optimal global threshold.

- Results in Case CZ is used for Local Threshold Selection

In Table 5.11, performances of FC, in case the tuning of the sensors was carried out applying CZ index, are reported.

It can be seen that the optimal global threshold is the same for any index used for the FC tuning: $\overline{\gamma} = -0.2600$. Results can be graphically seen in Figure 5.12, where the ROC curve is reported.

		Performances		Index			AUC
		$Q_{d,\mathrm{ref}}$	$Q_{f,\mathrm{ref}}$	J_{ref}	d^2	CZ	
Threshold	-∞	1	1	0	1	0	0.7637
	-1.3400	0.9273	0.6470	0.2803	0.4239	0.3273	
	-0.3360	0.8306	0.4923	0.3382	0.2711	0.4216	
	-0.2720	0.7374	0.3514	0.3860	0.1924	0.4783	
	-0.2600	**0.6269**	**0.2088**	**0.4181**	**0.1828**	**0.4960**	
	0.7330	0.4944	0.1490	0.3454	0.2778	0.4208	
	0.7450	0.3462	0.0826	0.2637	0.4342	0.3177	
	0.8090	0.2054	0.0256	0.1798	0.6320	0.2002	
	+∞	0	0	0	1	0	

Table 5.11 FC performances (in case it will perform the WFR) at different thresholds evaluated by *J*, d^2, and *CZ*. The optimal global threshold is reported in bold. These results are generated when sensor local thresholds have been selected using *CZ*.

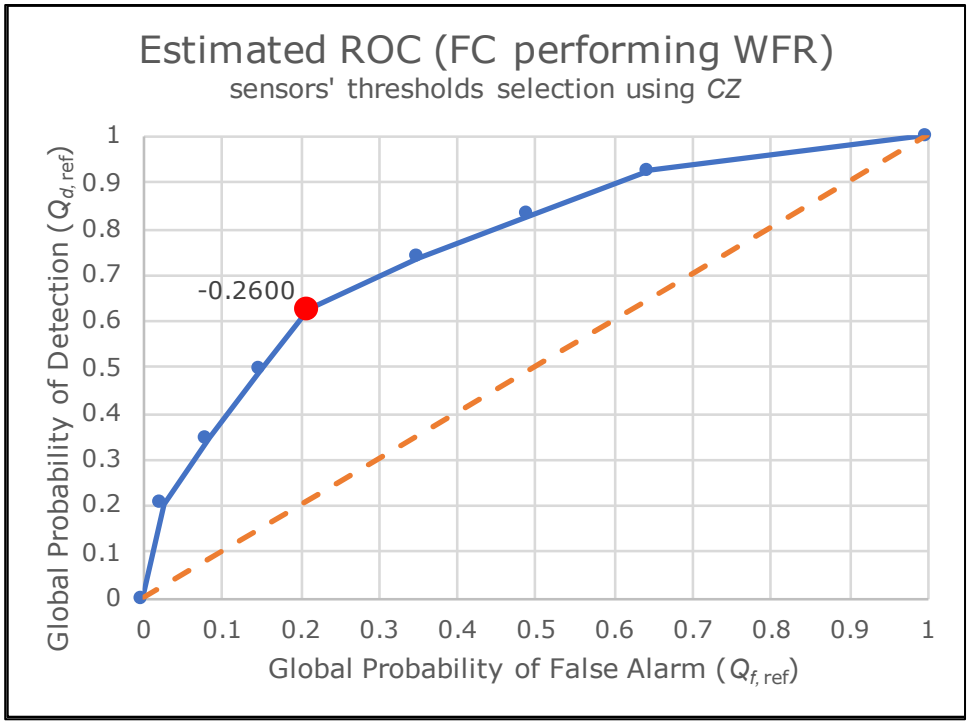


Figure 5.12 ROC curve of the performance test carried out during the tuning of the FC (in case it will perform the WFR) when local thresholds have been selected using *CZ*. Red dot on the curve is the optimal global threshold.

5.2.3 Observations

A general observation is on the choice of the threshold, more specifically on which index to use when select it. The choice of the index should be made knowing the system as a whole, not only focusing on the LDS. The choice should be based on the requirements dictated by a specific reliability study. Some examples could be to decide the threshold that ensures a specific minimum value of the probability of detection or a specific maximum value of the probability of false alarm.

In this study, this information is not available. However, a final threshold must be chosen. As explained in section 4.3.3.2, CZ and d^2 have the limitation of having little theoretical meaning (they are only based on the geometrical features of the ROC space). As already said, the same value of index may be referring to points on the opposite side of the chance line. Whereas, J is the only index among these from which, only by looking

at the sign of the index, it is possible to tell whether the threshold will have $Q_d > Q_f$ or not. For this reason, J will be used for the selection of the optimal global threshold.

Looking at Table 5.12 and Table 5.13, it can be seen how a FC tuned to perform the WFR should be able to provide slightly better results than the CR case (as global indexes for optimal thresholds shows more performing values). This may not be true in an absolute way since the real target position will alter the results. In fact, it is vital to remember that these results are not representative of a real scenario; they are only average values based on the assumption that all the HSs have the same failure probability. Also, tuning is based on physical models for signal simulation to which reference parameters were applied. These assumptions make these results different from a real scenario.

5.2.3.1 Observations on the Tuning of a FC performing the CR

From what observed, in order to maximize an index at FC level when the CR is expected to be performed (or minimize it, as in the case of d^2), the required action is to use the same index to select the local threshold at sensors level; for example, if the tuning of the FC aims to find the global threshold able to maximize J, the best idea is to tune the sensors selecting those local thresholds able to maximize J at local level also. In Table 5.12, it is possible to see how it can be obtained the optimal global threshold according to the different indexes starting from an appropriate tuning of the sensors:

	Case 1			**Case 2**			**Case 3**		
	Sen 1	**Sen 2**	**Sen 3**	Sen 1	Sen 2	Sen 3	Sen 1	Sen 2	Sen 3
Local Thr.	$\mathbf{1.8336}$	$\mathbf{1.9239}$	$\mathbf{1.7378}$	0.8942	0.9217	0.9432	1.0346	1.0921	1.0592
Sensor Index	$\overline{J} = \mathbf{0.2566}$	$\overline{J} = \mathbf{0.2735}$	$\overline{J} = \mathbf{0.2687}$	$\overline{d^2} = 0.3116$	$\overline{d^2} = 0.3038$	$\overline{d^2} = 0.2906$	$\overline{CZ} = 0.3799$	$\overline{CZ} = 0.3895$	$\overline{CZ} = 0.3894$
Global Thr.	$\mathbf{1}$			2			2		
FC Index	$\boldsymbol{J = 0.4049}$			$d^2 = 0.1885$			$CZ = 0.4688$		

Table 5.12 Table showing three different cases. Each case aims to obtain the optimal global threshold according to a specific index in case the FC performs the CR.

The suggested procedure is to select the sensor thresholds using J and use the same index to select the global threshold at FC level.

5.2.3.2 Observations on the Tuning of a FC Performing the WFR

In Table 5.13, it is possible to see how the optimal global threshold can be obtained according to the different indexes starting from an appropriate tuning of the sensors if the FC is expected to perform the WFR. It is visible that in order to be able to get the best value of a specified index at FC level, it may be necessary to tune sensors through the optimization of a different index. This introduces the fact that it would have been more accurate to perform an iterative loop where the sensor local thresholds were not selected to maximize (or minimize) a specific index at sensor level but at FC level. However, this would have required great computational effort. This because not only any non-negative value is a possible sensor local threshold, but also because when applying the WFR, any real value is a possible global threshold. Performing an iterative loop trying out any possible combination may result cumbersome. This is the main reason why local indexes were used for sensor tuning.

The suggested procedure is to select the sensor thresholds using CZ and use J to select the global threshold at FC level.

	Case 1			Case 2			Case 3		
	Sen 1	**Sen 2**	**Sen 3**	Sen 1	Sen 2	Sen 3	Sen 1	Sen 2	Sen 3
Local Thr.	$\mathbf{1.0346}$	$\mathbf{1.0921}$	$\mathbf{1.0592}$	1.7555	1.8183	1.7347	1.0346	1.0921	1.0592
Sensor Index	$\overline{\boldsymbol{CZ}} = \mathbf{0.3799}$	$\overline{\boldsymbol{CZ}} = \mathbf{0.3895}$	$\overline{\boldsymbol{CZ}} = \mathbf{0.3894}$	$\bar{J} = 0.2398$	$\bar{J} = 0.2520$	$\bar{J} = 0.2684$	$\overline{CZ} = 0.3799$	$\overline{CZ} = 0.3895$	$\overline{CZ} = 0.3894$
Global Thr.	$\mathbf{0.0640}$			0.1020			-0.2600		
FC Index	$\boldsymbol{J} = \mathbf{0.4181}$			$d^2 = 0.1799$			$CZ = 0.4815$		

Table 5.13 Table showing three different cases. Each case aims to obtain the optimal global threshold according to a specific index in case the FC performs the WFR.

5.3 Real-time Algorithm

Once the tuning of either the sensors or the FC is completed, the LDS can be tested in a real/simulated scenario. An algorithm for real-time detection and localization must be developed and will consist of a set of operations the sensors and the FC must perform during the operating condition.

The algorithm will be instant-based, it assumes perfect instant coordination among the sensors and the FC where every action, operation, or computation is instantaneous (no transmission latency or processing time is taken into consideration).

Also in this case, it is necessary to remember that the reported equations refer to a notation where values are normalized considering $\sigma_{w,k}^2 = 1$.

5.3.1 Sensor Level

As seen, each sensor performs a test on the received signal. Many tests could be carried out. In this work, the pdfs of the signals in case of either leak or no leak, were modeled in such a way that the ET is the UMP test.

The algorithm for the kth sensor is shown in Figure 5.13:

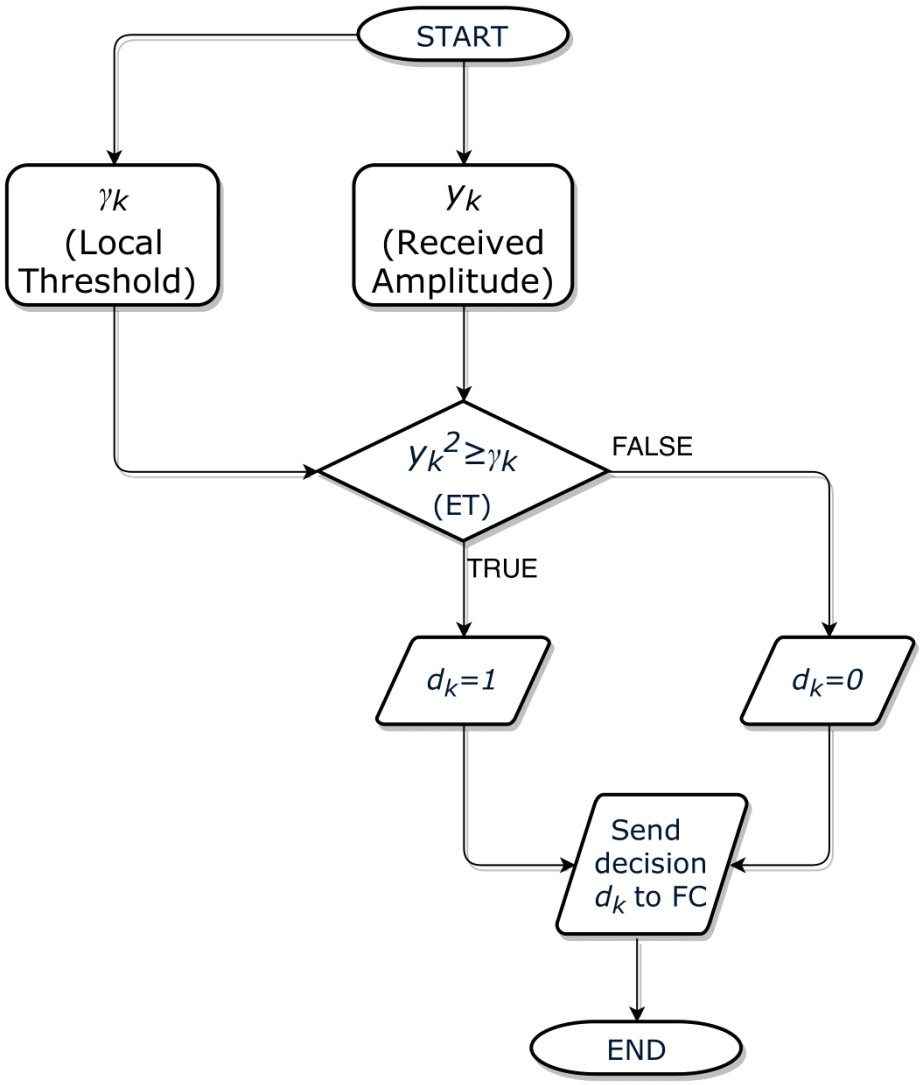


Figure 5.13 Flowchart of the algorithm that a sensor must perform each instant

Each instant t, the sensor must perform a series of actions that can be written in the following mathematical form:

$$d_{k,t} = \begin{cases} 1, & y_{k,t}^2 \geq \gamma_k \\ 0, & y_{k,t}^2 < \gamma_k \end{cases} \quad k = 1, \dots, N_K$$

This means that at any instant t, any sensor transmits its local decision $d_{k,t}$ to the FC.

This algorithm repeats itself each instant.

5.4 Fusion Center Level (Detection and Localization)

5.4.1 Localization Techniques

Localization is a fundamental task when it comes to WSN dealing with NCT. ROVs inspections can constitute a large part of maintenance costs (Mai et al., 2016). An efficient localization system may increase the ability to use ROVs in a more efficient way leading to a reduction of its associated operating cost. Here is explained the principles on which some proposed methods to localize the oil spill have been developed in this work. How these methods can be applied will be discussed in the section dedicated to the real-

time algorithm, which will focus on how to use these methods in the detection of an oil spill. Four methods have been developed. They are summarized in the scheme in Figure 5.14:

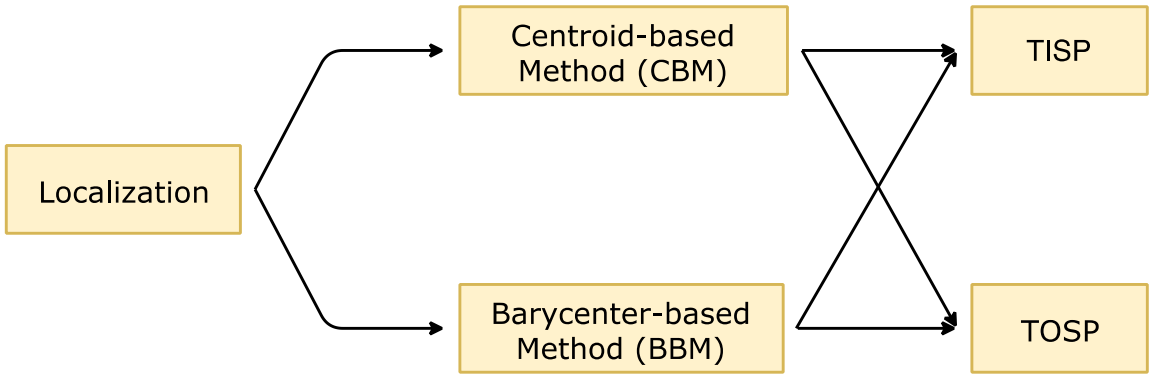


Figure 5.14 Summary of the developed localization techniques

TISP is the acronym for Target Inside Sensors' Perimeter, while TOSP is the acronym for Target Outside Sensors' Perimeter. These localization methods are all applied when the FC has declared the event as positive. If FC declares the event as negative, there will not be any attempt to estimate the target position.

- Centroid-based method (CBM-TISP)

When FC declares the event as positive, it also calculates an instantaneous position averaging the cartesian coordinates of the sensors that declared the event as positive, neglecting those that declared it as negative using the following equation:

$$\underline{x_T}_{\text{estimated}} = \frac{\sum_{k=1}^{N_K} \left(d_k \cdot \underline{x_S}_k \right)}{\sum_{k=1}^{N_K} d_k}$$

In Figure 5.15, the graphical representation of an example of an application of the previous equation is reported.

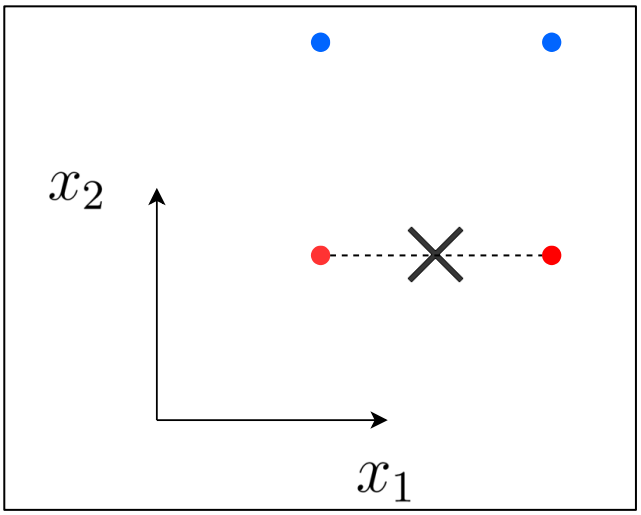


Figure 5.15 Centroid-based Method/Target Inside Sensors' Perimeter (CBM/TISP). Red dots are sensors declaring the event as positive, blue dots are sensors declaring the event as negative, and the cross is the estimated target position.

- Barycentric-based method (BBM-TISP)

In this case, the mean is not arithmetic, but it is weighted on the ratio of the sensor performances. The reason why this method is an improvement is that in the CBM the position is only dependent on the sensor declaration d_k and on its position. Let us suppose one sensor decided $d_k = 1$. If that sensor had been closer to the target, it would still have decided $d_k = 1$. Using BBM, this change of position would result in a different result as this change of sensor position would have been compensated with a change of performances, which are used as weights in barycenter calculation. Also in this case, only the sensors declaring the event as positive are taken into consideration:

$$\underline{x_T}_{\text{estimated}} = \frac{\sum_{k=1}^{N_K} \left(d_k \cdot \underline{x_S}_k \cdot \frac{P_{d,k}}{P_{f,k}} \right)}{\sum_{k=1}^{N_K} \left(d_k \cdot \frac{P_{d,k}}{P_{f,k}} \right)}$$

Figure 5.16 is the graphical representation showing how the estimation may be affected in case the bottom-left dot represents a sensor whose performances ($P_{d,k}/P_{f,k}$) are higher than the bottom-right one.

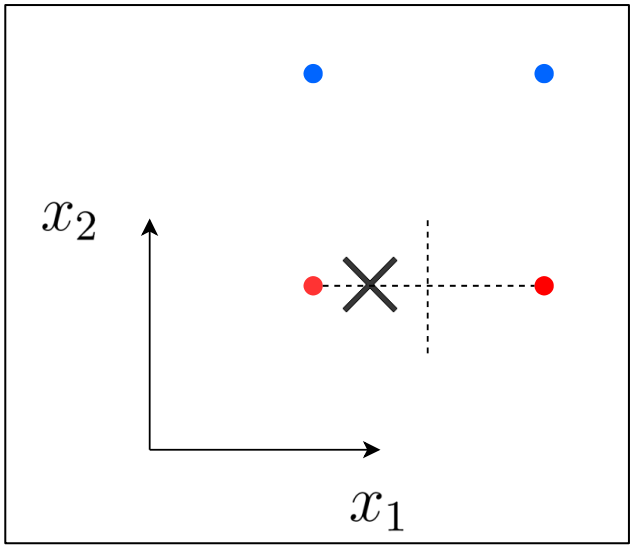


Figure 5.16 Barycenter-based Method/Target Inside Sensors' Perimeter (BBM/TISP). The bottom-left dot is a sensor whose performances ($P_{d,k}/P_{f,k}$) are higher than the bottom-right one. Red dots are sensors declaring the event as positive, blue dots are sensors declaring the event as negative, and the cross is the estimated target position. The estimated target location has moved towards the bottom-left sensor.

To obtain the graphical representation above these performances were considered:

$$\left(\frac{P_d}{P_f}\right)_{\text{Bottom-right sensor}} < \left(\frac{P_d}{P_f}\right)_{\text{Bottom-left sensor}}$$

The two methods showed above have the mathematical limit that the estimated target position can never be outside the perimeter of the polygon formed by using the sensors as vertices. This methods (TISP), in fact, should only be used in case the desired outcome is to find a position inside the sensors' perimeter. Variations of the TISP methods are those in which it is possible to find a target which is outside the sensors' perimeter (TOSP) by giving importance also to those sensors that declared the event as negative.

- Centroid-based method (CBM-TOSP)

As before, this CBM is based on averaging the positions of the sensors. The step-by-step description of the final equation is the following:

- The centroid of the sensors that declared the event as positive is calculated with the following equation (same as CBM-TISP):

$$\frac{\sum_{i=1}^{N_K}\left(d_i \cdot \underline{x_{S_i}}\right)}{\sum_{i=1}^{N_K} d_i}$$

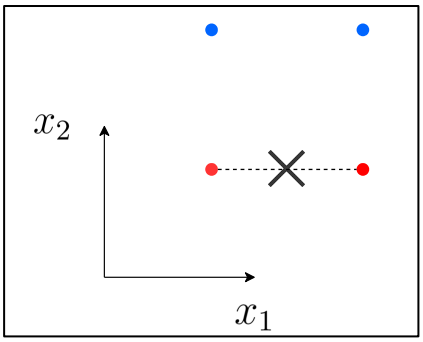


Figure 5.17 Step 1 of CBM-TOSP. Red dots are sensors declaring the event as positive, blue dots are sensors declaring the event as negative.

- For each sensor declaring the event as negative, a position symmetrically located on the opposite side of the line passing through the sensor and the previously obtained position is calculated. The following equation represents the calculation of the abovementioned position for the kth sensor. $(1-d_k)$ is used to keep the result only when the calculation is applied to the kth sensor if it declared the event as negative:

$$\left[2 \cdot \frac{\sum_{i=1}^{N_K}\left(d_i \cdot \underline{x_{S_i}}\right)}{\sum_{i=1}^{N_K} d_i} - \underline{x_{S_k}}\right] \cdot (1-d_k)$$

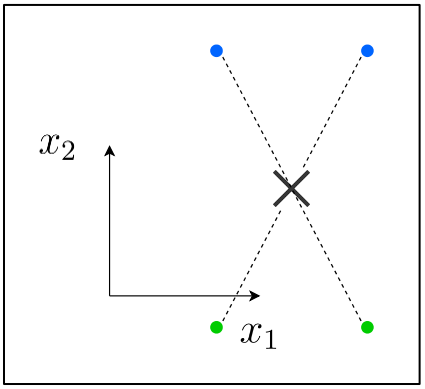


Figure 5.18 Step 2 of CBM-TOPS. Red dots are the newly calculated positions, blue dots are sensors declaring the event as negative.

As it is possible to see from Figure 5.18, the green dots represent the newly calculated positions, each of them is located on the line passing through the corresponding sensor and the position calculated in step 1 maintaining the same distance from it.

- Now the centroid of the positions of the sensors that declared the event as positive plus the newly calculated positions must be obtained. This is done by averaging their coordinates, and the result will represent the estimated target position :

$$\underline{x_T}_{\text{estimated}} = \frac{\sum_{k=1}^{N_K} \left\{ d_k \cdot \underline{x_S}_k + \left[2 \cdot \frac{\sum_{i=1}^{N_K} \left(d_i \cdot \underline{x_S}_i \right)}{\sum_{i=1}^{N_K} d_i} - \underline{x_S}_k \right] \cdot (1 - d_k) \right\}}{N_K}$$

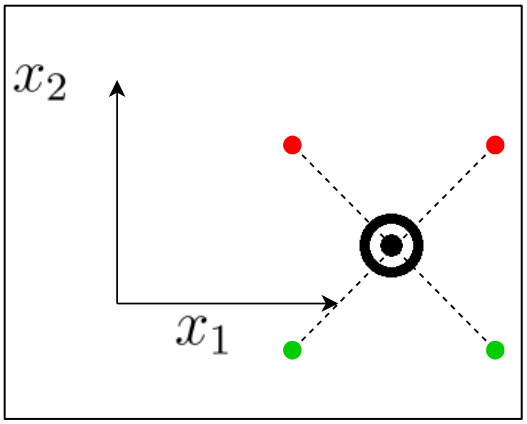


Figure 5.19 Step 3 of CBM-TOSP. Red dots are sensors declaring the event as positive, blue dots are sensors declaring the event as negative, and the double circle is the estimated target position.

As it is possible to observe from the graphical representation in Figure 5.20, this method can give a result outside the sensors' perimeter. This is done by faking the existence of some sensors (declaring the event as positive) outside the perimeter. According to the mathematics used to build the method, it is clear that TOPS methods do not extend the localization up to unlimited distances outside the sensors' perimeter. The comparison can be seen below:

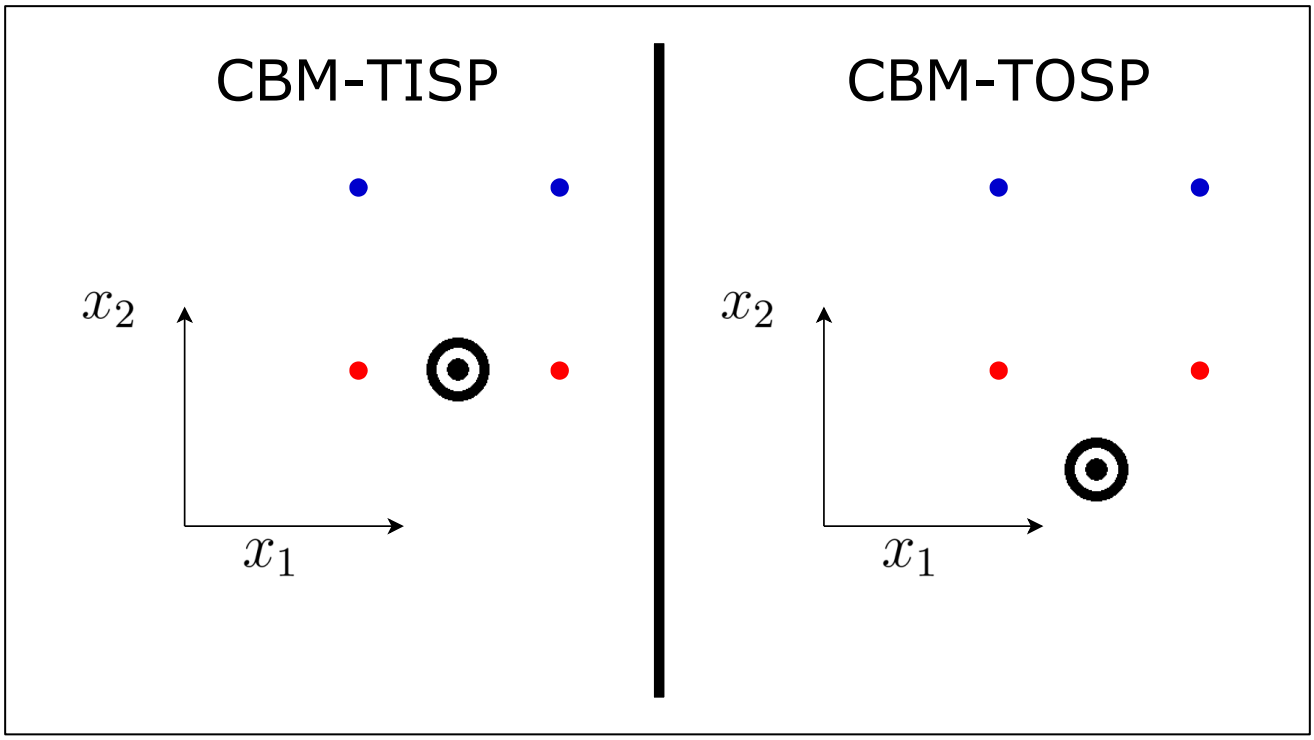


Figure 5.20 Comparison between CBM-TISP (on the left) and CBM-TOSP (on the right). Red dots are sensors declaring the event as positive, blue dots are sensors declaring the event as negative, and the double circles are the estimated positions of the target.

- Barycenter-based method (BBM-TOSP)

In analogy to what explained so far, this method can give the estimated target position outside of the sensors' perimeter with the additional feature of weighting any position based on the sensor performances. Also in this case, a step-by-step procedure on how the equation is built is presented:

- The barycenter of the sensors that declared the event as positive is calculated as the following (same as BBM-TISP):

$$\frac{\sum_{i=1}^{N_K}\left(d_i \cdot \underline{x_{S_i}} \cdot \frac{P_{d,i}}{P_{f,i}}\right)}{\sum_{i=1}^{N_K}\left(d_i \cdot \frac{P_{d,i}}{P_{f,i}}\right)}$$

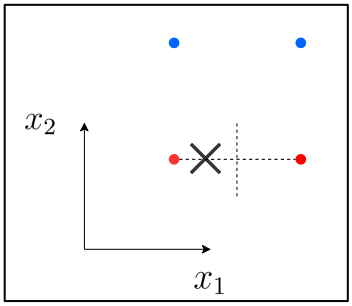


Figure 5.21 Step 1 of BBM-TOPS. Red dots are sensors declaring the event as positive, blue dots are sensors declaring the event as negative.

- For each sensor that declared the event as negative, a position symmetrically located on the opposite side of the line passing through the sensor and the

previously obtained position is calculated, this fakes the existence of sensors declaring the event as positive. The following equation represents the calculation of the abovementioned position for the kth sensor. $(1 - d_k)$ is used to keep the result only when the calculation is applied to the kth sensor if it declared the event as negative:

$$\left[2 \cdot \frac{\sum_{i=1}^{N_K}\left(d_i \cdot \underline{x_S}_i \cdot \frac{P_{d,i}}{P_{f,i}}\right)}{\sum_{i=1}^{N_K}\left(d_i \cdot \frac{P_{d,i}}{P_{f,i}}\right)} - \underline{x_S}_k\right] \cdot (1 - d_k)$$

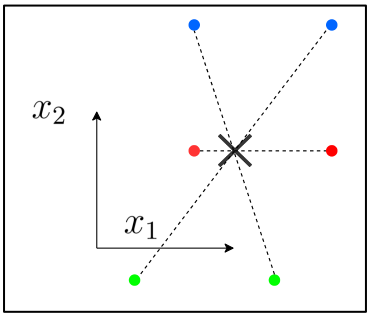


Figure 5.22 Step 2 of BBM-TOPS. Red dots are sensors declaring the event as positive, blue dots are sensors declaring the event as negative, and green dots are the newly calculated positions

- Now the barycenter of the positions of the sensors that declared the event as positive plus the newly calculated positions must be obtained. This is done by carrying out a weighted average of their coordinates using $P_{d,k}/P_{f,k}$ as weights. Positions calculated in the previous step are weighted using performances of the correspondent sensors from which they have been obtained. The result will represent the estimated target position:

$$\underline{x_T}_{\text{estimated}} = \frac{\sum_{k=1}^{N_K}\left\{d_k \cdot \underline{x_S}_k \cdot \frac{P_{d,k}}{P_{f,k}} + \left[2 \cdot \frac{\sum_{i=1}^{N_K}\left(d_i \cdot \underline{x_S}_i \cdot \frac{P_{d,i}}{P_{f,i}}\right)}{\sum_{i=1}^{N_K}\left(d_i \cdot \frac{P_{d,i}}{P_{f,i}}\right)} - \underline{x_S}_k\right] \cdot (1 - d_k) \cdot \frac{P_{d,k}}{P_{f,k}}\right\}}{\sum_{k=1}^{N_K} \frac{P_{d,k}}{P_{f,k}}}$$

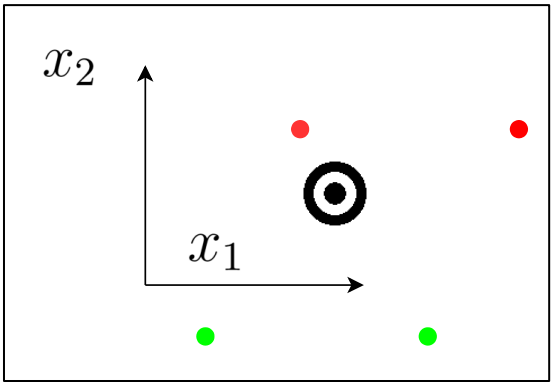


Figure 5.23 Step 3 of BBM-TOSP. Red dots are sensors declaring the event as positive, blue dots are sensors declaring the event as negative, and the double circle is the estimated target position.

To obtain the graphical representation above these performances were considered:

$$\left(\frac{P_d}{P_f}\right)_{\text{Top-left sensor}} < \left(\frac{P_d}{P_f}\right)_{\text{Top-right sensor}} < \left(\frac{P_d}{P_f}\right)_{\text{Bottom-right sensor}} < \left(\frac{P_d}{P_f}\right)_{\text{Bottom-left sensor}}$$

Like before, it is possible to make a graphical comparison between the BBM-TISP and BBM-TOSP where it is possible to see how BBM-TOSP can give results outside the sensors' perimeter (Figure 5.24):

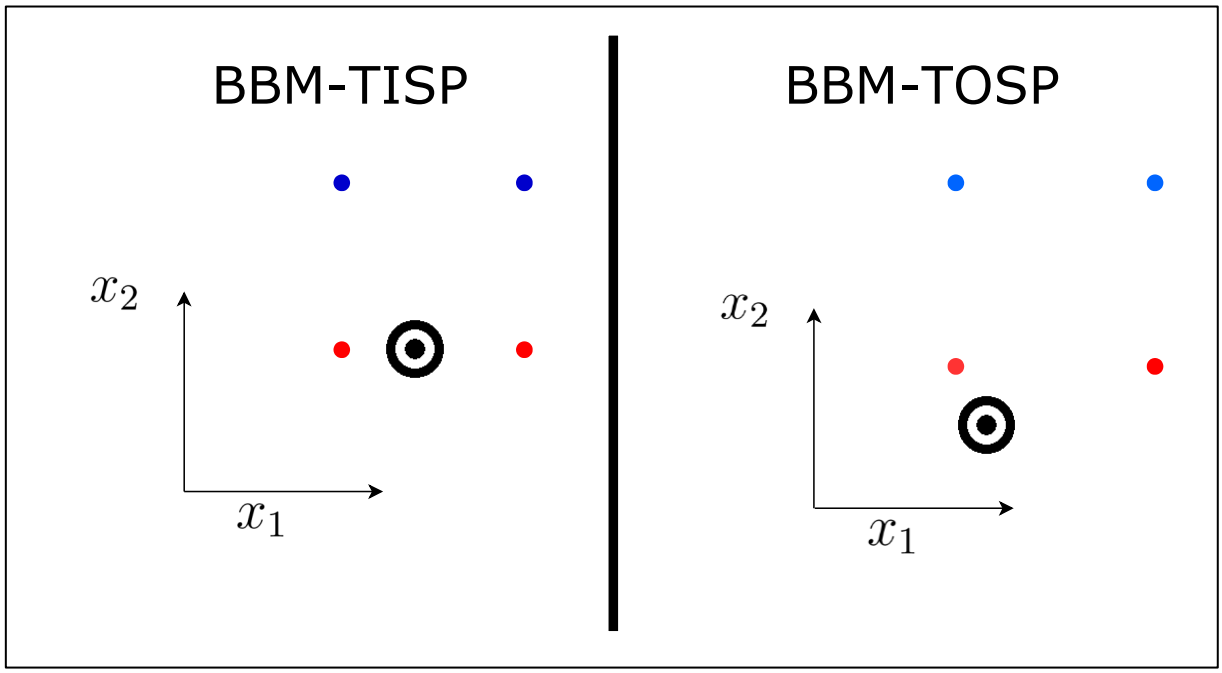


Figure 5.24 Comparison between BBM-TISP and BBM-TOSP. Red dots are sensors declaring the event as positive, blue dots are sensors declaring the event as negative, and the double circles are the estimated positions of the target.

What described so far are four techniques that can estimate the target position in those instants in which FC declares the event as positive. The calculated positions may be inaccurate since the sensors have $P_d < 1$ and $P_f > 0$ making the fusion center have global performances $Q_d < 1 \text{ and } Q_f > 0$. This may lead to unreliable results. To have a more reliable estimated position as a result, it is possible to calculate the centroid (arithmetic mean) of all the obtained position up to the instant of interest. This can be done since the target does not move, which means all instantaneous FC decisions can be considered equally reliable (it is the reason why the arithmetic mean is performed). Let us then have N instants in which FC declared the event as positive. This means that N instantaneous positions have been calculated so far by one of the four methods above reported. For the Nth instant, the calculated position ($\underline{x_T}_{\text{estimated},N}$) will not be the final positions, but it will be part of the arithmetic mean among all the calculated positions up to that instant leading to a more reliable position ($\underline{X_T}_{\text{estimated},N}$):

$$\begin{cases} \underline{X_T}_{\text{estimated},1} = \underline{x_T}_{\text{estimated},1} \\ \underline{X_T}_{\text{estimated},2} = \dfrac{\underline{x_T}_{\text{estimated},1} + \underline{x_T}_{\text{estimated},2}}{2} \\ \vdots \\ \underline{X_T}_{\text{estimated},N} = \dfrac{\underline{x_T}_{\text{estimated},1} + \underline{x_T}_{\text{estimated},2} + \cdots + \underline{x_T}_{\text{estimated},N}}{N} \end{cases}$$

The latter system can be re-written in a way in which it is not necessary to have N positions memorized, but only two positions plus a scalar value (N):

$$\begin{cases} \underline{X_T}_{\text{estimated},1} = \underline{x_T}_{\text{estimated},1} \\ \underline{X_T}_{\text{estimated},2} = \dfrac{1 \cdot \underline{X_T}_{\text{estimated},1} + \underline{x_T}_{\text{estimated},2}}{2} \\ \vdots \\ \underline{X_T}_{\text{estimated},N} = \dfrac{(N-1) \cdot \underline{X_T}_{\text{estimated},N-1} + \underline{x_T}_{\text{estimated},N}}{N} \end{cases}$$

To conclude, four are the developed localization techniques. It is clear how CBM is inspired by CR, which considers sensors equally performing, and CBM is inspired by WFR, where positions/decisions are combined and weighted. As in the case of the FRs for detection, where WFR ends up being equivalent to CR when sensor performances are equal, BBM ends up being equivalent to CBM in such a scenario. For this reason, it is coherent to use CBM when the FC bases its decision on the CR while BBM should be used when FC uses the WFR. Moreover, TISP methods should be used when there is no doubt that the target position will be inside the sensors' perimeter, while TOSP methods should be used when the target may be located outside the sensors' perimeter. Table 5.14 summarizes what said so far:

	CBM-TISP	**CBM-TOSP**	**BBM-TISP**	**BBM-TOSP**
Suggested coupled FR	CR	CR	WFR	WFR
Performance-based	No	No	Yes	Yes
Possible results outside sensors' perimeter	No	Yes	No	Yes

Table 5.14 Summary of localization techniques properties

A comment that could be raised is that the real target position is unknown, but either the WFR or the BBM are based on the knowledge of sensor performances which strictly depends on their distance to the target. This is a clear limitation; in fact, the ideal scenario would be to know the exact performance of any single sensor. The proposed solution is to use either the WFR or the BBM with the same a priori estimation of sensor performances.

In this work, two families of methods are proposed for the algorithm that must be performed by the FC:

- FC performing the CR for detection and the CBM for localization (TISP or TOSP method);
- FC performing the WFR for detection and the BBM for localization (TISP or TOSP method).

In case the CR is used, it is reasonable to assume that reference sensor performances are not available, or they are not considered reliable enough to be used. In this scenario, CBM can be used for localization as it is not based on sensor performances, but only on positions. The second method, instead, consists of coupling the WFR with the BBM and it is the most complex as it introduces the knowledge of reference performances, which can be obtained from the sensor tuning (Table 5.8) by assessing the probabilities of failure of manifold components.

5.4.2 Fusion Center performing the Counting Rule and the Centroid-Based Method (CR+CBM)

Figure 5.25 shows the algorithm the FC must perform if the CR and the CBM are applied. This must be performed each instant. Instants are synchronized with those used in the algorithm dedicated to local detection.

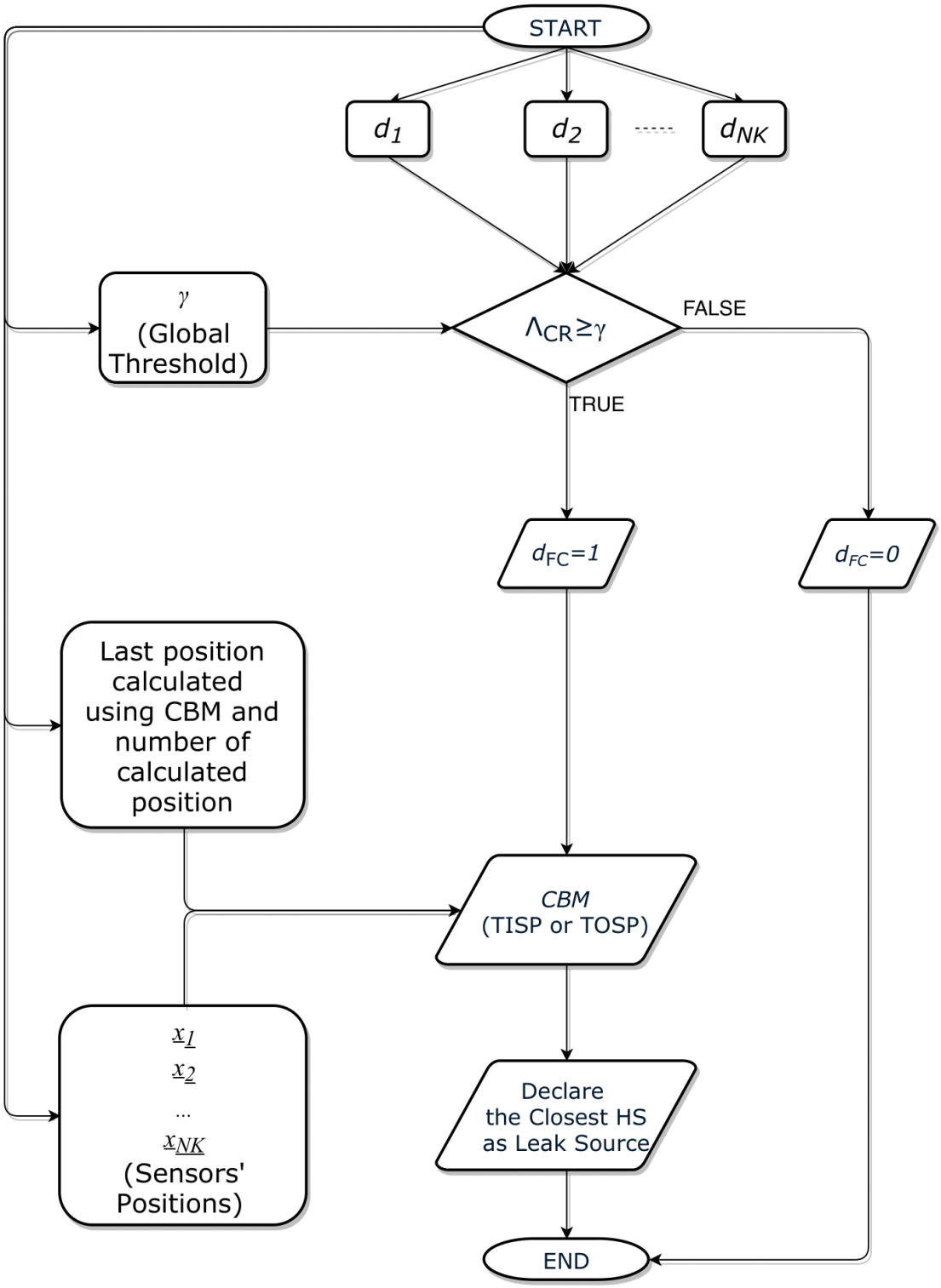


Figure 5.25 Flowchart of the real-time algorithm the FC must perform. CR and CBM are applied in this case

The algorithm in Figure 5.25 can be written in a mathematical form.

For any instant t, the FC must perform the CR:

$$\begin{cases} \Lambda_{\mathrm{CR}_t} = \sum_{k=1}^{N_K} d_{k,t} \\ d_{\mathrm{FC}_t} = \begin{cases} 1, & \Lambda_{\mathrm{CR}_t} \geq \overline{\gamma} \\ 0, & \Lambda_{\mathrm{CR}_t} < \overline{\gamma} \end{cases} \end{cases}$$

In case a spill is detected ($d_{\mathrm{FC}_t} = 1$), the FC updates the number of instants in which it declared a positive event (t^* indicates the counter):

$$t^* = \sum_{i=1}^{t} i \cdot d_{\mathrm{FC}_i}$$

In this case, the FC can proceed with the localization of the leakage point by application of the CBM in one of its two forms (TISP or TOSP):

$$\text{TISP:} \quad \underline{x_T}_{\text{estimated}_{t^*}} = \frac{\sum_{k=1}^{N_K} \left(d_{k,t} \cdot \underline{x_S}_k \right)}{\sum_{k=1}^{N_K} d_{k,t}}$$

$$\text{TOSP:} \quad \underline{x_T}_{\text{estimated}_{t^*}} = \frac{\sum_{k=1}^{N_K} \left\{ d_{k,t} \cdot \underline{x_S}_k + \left[2 \cdot \frac{\sum_{i=1}^{N_K} \left(d_{i,t} \cdot \underline{x_S}_i \right)}{\sum_{i=1}^{N_K} d_{i,t}} - \underline{x_S}_k \right] \cdot \left(1 - d_{k,t} \right) \right\}}{N_K}$$

The FC can now take a more reliable decision regarding the position based on previously estimated positions and positions of the HSs:

$$\begin{cases} \underline{X_T}_{\text{estimated}_{t^*}} = \begin{cases} \underline{x_T}_{\text{estimated}_{t^*}}, & t^* = 1 \\ \dfrac{(t^* - 1) \cdot \underline{x_T}_{\text{estimated}_{t^*-1}} + \underline{x_T}_{\text{estimated}_{t^*}}}{t^*}, & t^* \neq 1 \end{cases} \\ \underline{X_T}_{\text{localized}_{t^*}} = \underline{x_{HS}} \left(\min_{h=1,\ldots,N_{HS}} \left\| \underline{X_T}_{\text{estimated}_{t^*}} - \underline{x_{HS}}_h \right\| \right) \end{cases}$$

- t^* is the number of instants the FC declared the existence of a leakage;
- $\underline{x_T}_{\text{estimated}_{t^*}}$ is the position calculated by the FC considering the information received by the sensors in that specific moment;
- $\underline{X_T}_{\text{estimated}_{t^*}}$ is the overall position calculated by also accounting for the previous estimations (this assumes the target does not change its position over time). This calculation is done by performing an arithmetic mean;
- $\underline{X_T}_{\text{localized}_{t^*}}$ is the closest HS to $\underline{X_T}_{\text{estimated}_{t^*}}$ and it is the final position declared by the FC. The method is based on the principle that only HSs can experience a leakage.

5.4.3 Fusion Center performing the Weighted Fusion Rule and the Barycenter-Based Method (WFR+BBM)

Figure 5.26 shows the algorithm the FC must perform if the WFR and the BBM are applied. This must be performed each instant by the FC. Instants are synchronized with those used in the algorithm dedicated to local detection.

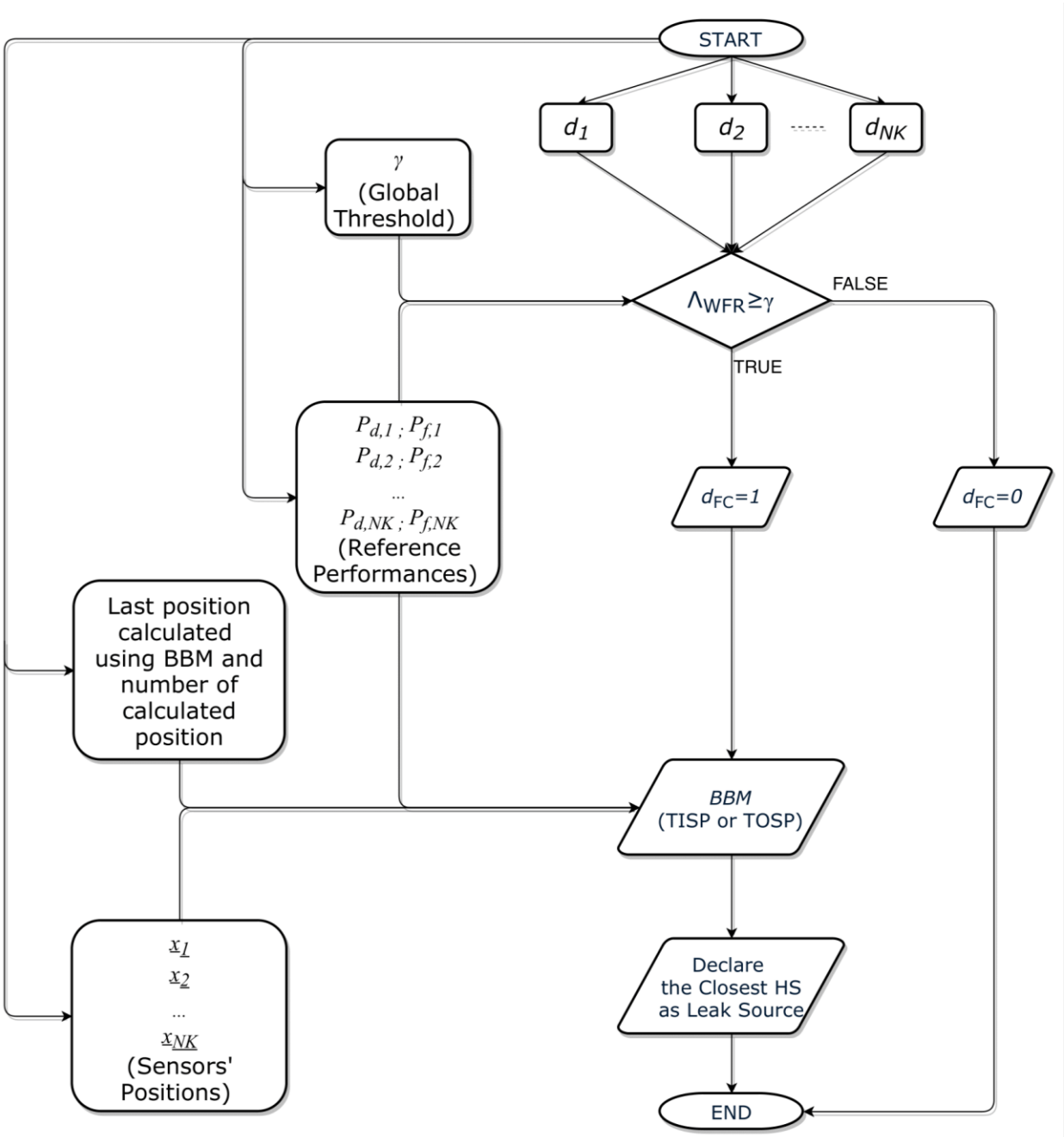


Figure 5.26 Flowchart of the real-time algorithm the FC must perform. WFR and BBM are applied in this case

The algorithm in Figure 5.26 can be written in the following mathematical form.

For any instant t, the FC must perform the WFR:

$$\begin{cases} \Lambda_{\mathrm{WFR}_t} = \sum_{k=1}^{N_K} \left[d_{k,t} \cdot \ln\left(\frac{P_{d,\mathrm{FC}_k}}{P_{f,\mathrm{FC}_k}}\right) + \left(1 - d_{k,t}\right) \cdot \ln\left(\frac{1 - P_{d,\mathrm{FC}_k}}{1 - P_{f,\mathrm{FC}_k}}\right) \right] \\ d_{\mathrm{FC}_t} = \begin{cases} 1, & \Lambda_{\mathrm{WFR}_t} \geq \overline{\gamma} \\ 0, & \Lambda_{\mathrm{WFR}_t} < \overline{\gamma} \end{cases} \end{cases}$$

In case a spill is detected ($d_{\mathrm{FC}_t} = 1$), the FC updates t^*:

$$t^* = \sum_{i=1}^{t} i \cdot d_{\mathrm{FC}_i}$$

In this case, the FC can proceed with the localization of the leakage point by applying the BBM in one of its two forms (TISP or TOSP):

$$\text{TISP:} \quad \underline{x_T}_{\text{estimated}_{t^*}} = \frac{\sum_{k=1}^{N_K} \left(d_{k,t} \cdot \underline{x_S}_k \cdot \frac{P_{d,\mathrm{FC}_k}}{P_{f,\mathrm{FC}_k}} \right)}{\sum_{k=1}^{N_K} \left(d_{k,t} \cdot \frac{P_{d,\mathrm{FC}_k}}{P_{f,\mathrm{FC}_k}} \right)}$$

$$\text{TOSP:} \quad \underline{x_T}_{\text{estimated}_{t^*}} = \frac{\sum_{k=1}^{N_K} \left\{ d_{k,t} \cdot \underline{x_S}_k \cdot \frac{P_{d,\mathrm{FC}_k}}{P_{f,\mathrm{FC}_k}} + \left[2 \cdot \frac{\sum_{i=1}^{N_K} \left(d_{i,t} \cdot \underline{x_S}_i \cdot \frac{P_{d,\mathrm{FC}_i}}{P_{f,\mathrm{FC}_i}} \right)}{\sum_{i=1}^{N_K} \left(d_{i,t} \cdot \frac{P_{d,\mathrm{FC}_i}}{P_{f,\mathrm{FC}_i}} \right)} - \underline{x_S}_k \right] \cdot \left(1 - d_{k,t}\right) \cdot \frac{P_{d,\mathrm{FC}_k}}{P_{f,\mathrm{FC}_k}} \right\}}{\sum_{k=1}^{N_K} \frac{P_{d,\mathrm{FC}_k}}{P_{f,\mathrm{FC}_k}}}$$

The FC can now take a more reliable decision regarding the position based on previously calculated positions and positions of the HSs:

$$\begin{cases} \underline{X_T}_{\text{estimated}_{t^*}} = \begin{cases} \underline{x_T}_{\text{estimated}_{t^*}}, & t^* = 1 \\ \dfrac{(t^* - 1) \cdot \underline{x_T}_{\text{estimated}_{t^*-1}} + \underline{x_T}_{\text{estimated}_{t^*}}}{t^*}, & t^* \neq 1 \end{cases} \\ \underline{X_T}_{\text{localized}_{t^*}} = \underline{x_{HS}} \left(\min_{h=1,\ldots,N_{HS}} \left\| \underline{X_T}_{\text{estimated}_{t^*}} - \underline{x_{HS}}_h \right\| \right) \end{cases}$$

- P_{d,FC_k} and P_{f,FC_k} are the reference performances of the kth sensor;
- t^* is the number of instants the FC declared the existence of a leakage;
- $\underline{x_T}_{\text{estimated}_{t^*}}$ is the position calculated by the FC considering the information received by the sensors in that specific moment;
- $\underline{X_T}_{\text{estimated}_{t^*}}$ is the overall position calculated by also accounting for the previous estimations (this assumes the target does not change its position over time). This calculation is done by performing an arithmetic mean;
- $\underline{X_T}_{\text{localized}_{t^*}}$ is the closest HS to $\underline{X_T}_{\text{estimated}_{t^*}}$ and it is the final position declared by the FC. The method is based on the principle that only HSs can experience a leakage.

6 Test Results

6.1 Oil Spill Modeling

The simulation is carried out modeling the scenario as binary:

$$\begin{cases} H_1: \text{positive event (leak)} \\ H_0: \text{negative event (no leak)} \end{cases}$$

Note that the adjectives "positive" and "negative" only refer to the presence or absence of the spill, respectively. This means, according to the binary assumption, that all the releases are considered equally over time despite differences in flow rate (or other physical specifications). The model used to create a sequence of binary event is the Binary Markov Chain (BMC). The fundamental property of a time-discrete Markov Chain of n successive instants is the Markov Property (Ruebeck et al., 2018):

$$p(X_n = x_n | X_{n-1} = x_{n-1}, \dots, X_1 = x_1) = p(X_n = x_n | X_{n-1} = x_{n-1})$$

which indicates that the conditional probability density of the system of being in a specific state at instant n, once known the value $X_{n-1} = x_{n-1}$, is uniquely determined and not affected by the knowledge of the values at earlier instants (Van Kampen, 2011).

In the specific case of this study (BMC), the Markov Property has the further condition that X_i (where $i \in \{1, \dots, n\}$) can only assume two values $\{H_0, H_1\}$. The scheme in Figure 6.1 clearly shows these properties:

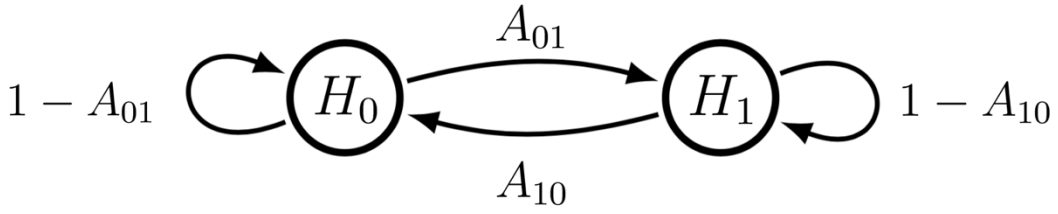


Figure 6.1 Binary Markov Chain

Values next to arrows are the probabilities of the Markov process for a system to change from a state to another. The diagram also shows that only two values (A_{01} and A_{10}) are necessary to define a BMC. In this study, two different values (p and r) were used to define the process:

$$\begin{cases} p = \dfrac{A_{01}}{A_{01} + A_{10}} \\ r = \dfrac{1 - A_{01}}{A_{10}} \end{cases} \Rightarrow \begin{cases} A_{10} = \dfrac{1 - p}{p + r - pr} \\ A_{01} = \dfrac{p}{p + r - pr} \end{cases}$$

p represents the steady-state probability of a system to be in the state H_1 (in analogy with steady-state availability for reparable components used in reliability engineering (O'Connor and Kleyner, 2012)).

r is the ratio between the probability of the system to remain in state H_0 and the probability of switching from state H_1 to H_0. It represents the trend of the system to stay in the same state with respect to a transition between two states.

Using a BMC to simulate the pattern of underwater hydrocarbon leaks over time has some advantages and disadvantages (Kushner and Dupuis, 2001; O'Connor and Kleyner, 2012):

- Markov Analysis is fast to simulate (acceptable amount of computational work);
- It makes the problem depend only on two variables (p and r);
- The binary pattern can be easily modeled, acting on variables (p and r), which directly shape the release pattern.

However, on the other hand:

- The probability of the system to changes from H_1 to H_0 and vice versa is constant in time (degradation of the system is not accounted);
- The use of a stochastic model like a BMC rather than a deterministic physical model makes the simulation independent from physical data (such as template conditions, PVT properties, environmental conditions, etc.).
- BMC approximation can only simulate the presence of an oil spill or its absence without giving any further information (such as leak flow rate, acoustic information associated with the leak, etc.).

Simulations of possible oil spills were carried out using MATLAB.

As the simulation is carried out by using the software, and not employing a real test, some assumptions were made:

- The release generates an acoustic signal which has the same specifications of the one used during the tuning of the sensors and the FC; this is the equivalent of assuming that the models for signal modeling used for the tuning of the sensors and the FC are perfectly accurate;
- The leakage point will be located in one of the HSs identified during the analysis of the SPS;
- Only a single point will experience leakage.

A BMC was used to create a sequential pattern of events. The values used to define it were:

$$\begin{cases} p = 1/3 \\ r = 2 \end{cases} \Rightarrow \begin{cases} A_{10} = 0.4 \\ A_{01} = 0.2 \end{cases}$$

According to Figure 6.1, these values mean that an event, either positive or negative, which is occurring at a precise instant has 80% of probabilities of being in the same state the next instant if the event is negative, while it has 60% of probabilities if the event is positive. This pattern was chosen in order to create an imbalance oriented towards negative events (no leak). The imbalance will not influence the outcome of the ROC curve. Figure 6.2 shows an example of one hundred seconds modeled according to the above-described pattern. Each second correspond to an instant.

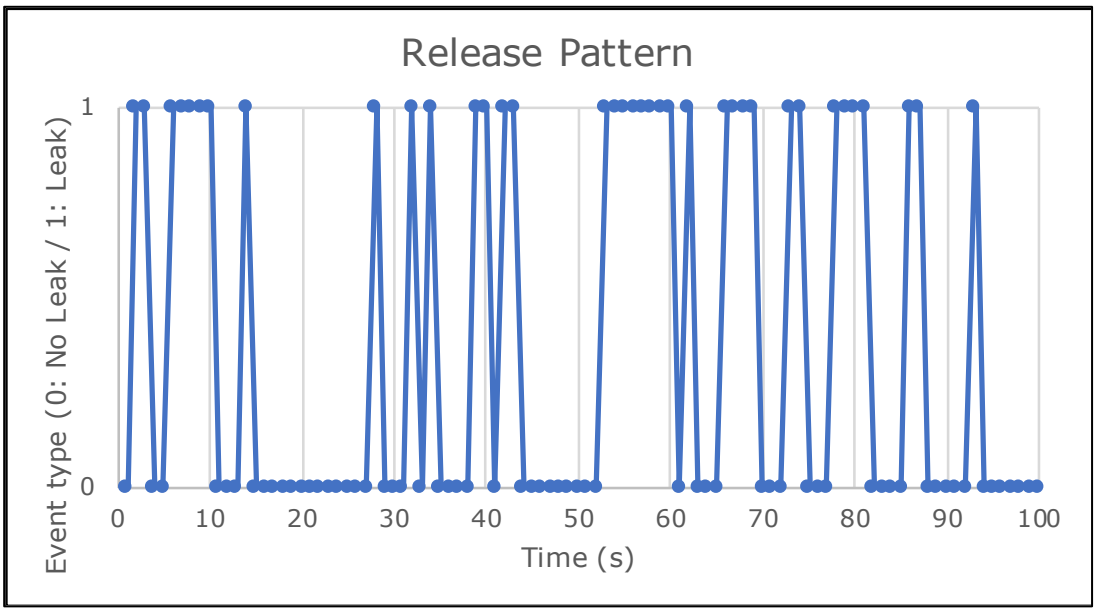


Figure 6.2 Release pattern generated through a BMC (*p*=1/3; *r*=2)

For each instant (lasting one second), a different signal is received by any sensor. The nature of these signals depends on the class of the event corresponding to that instant:

$$\begin{cases} H_0: & y_k = w_k \\ H_1: & y_k = \xi_k \cdot AAF\left(\underline{x_T}, \underline{x_S}_k\right) + w_k \end{cases}$$

Where:

$$\begin{cases} \xi_k \sim \mathcal{N}(0, SNR = 20) \\ w_k \sim \mathcal{N}(0,1) \end{cases}$$

Therefore, for each instant, sensors perform the ET while the FC performs the FR for either detection or localization based on one of the real-time algorithms shown in the previous chapter.

Each simulation consists of 10^4 instants. Since one instant corresponds to $1\ s$, this number of instants is equivalent to $2h\ 46min\ 40s$.

With the above-mentioned release pattern, it is possible to show the estimated PR curves based on the values obtained during the tuning of the FC either in case of use of the CR or the WFR. This was not possible during the tuning procedure of both sensors and FC, as the release pattern was unknown. As explained in Section 4.3.3.5, the F-score can be used to evaluate the capacity of the leak detection system to increase the recall maintaining a good value of precision. Here, F_1 and $F_{0.5}$ were calculated even though further analysis may show which value of β is considered optimal for this application.

Oil spill detection is an application in which the imbalance between classes of events is supposed to be in favor of the negative one (overall time in which spills happen is hopefully lower than the time no leaks are experienced). This makes precision a vital parameter (difficult to estimate a priori as the imbalance between classes may not be easily predictable). The severity of an oil spill makes the precision even more important; this is the reason why the value of $F_{0.5}$ is also shown. $F_{0.5}$ in fact weights precision more than recall.

Table 6.1 and Figure 6.3 show, for any value of the global threshold, the expected values of precision and recall and the resulting F_1 and $F_{0.5}$ values for a FC applying the CR. It is interesting to notice that $\overline{\gamma} = 1$ (the chosen threshold) maximize F_1 too, while $\overline{\gamma} = 2$ should maximize $F_{0.5}$.

		Precision	Recall	F_1	$F_{0.5}$
Threshold	0	0.3285	1	0.4945	0.3795
	1	**0.4844**	**0.8480**	**0.6166**	**0.5298**
	2	0.7396	0.4497	0.5593	0.6551
	3	0.9088	0.1018	0.1832	0.3516
	4	1	0	0	0

Table 6.1 Expected values of precision, recall, *F₁* and *F₀.₅* at different thresholds for the FC using the CR.

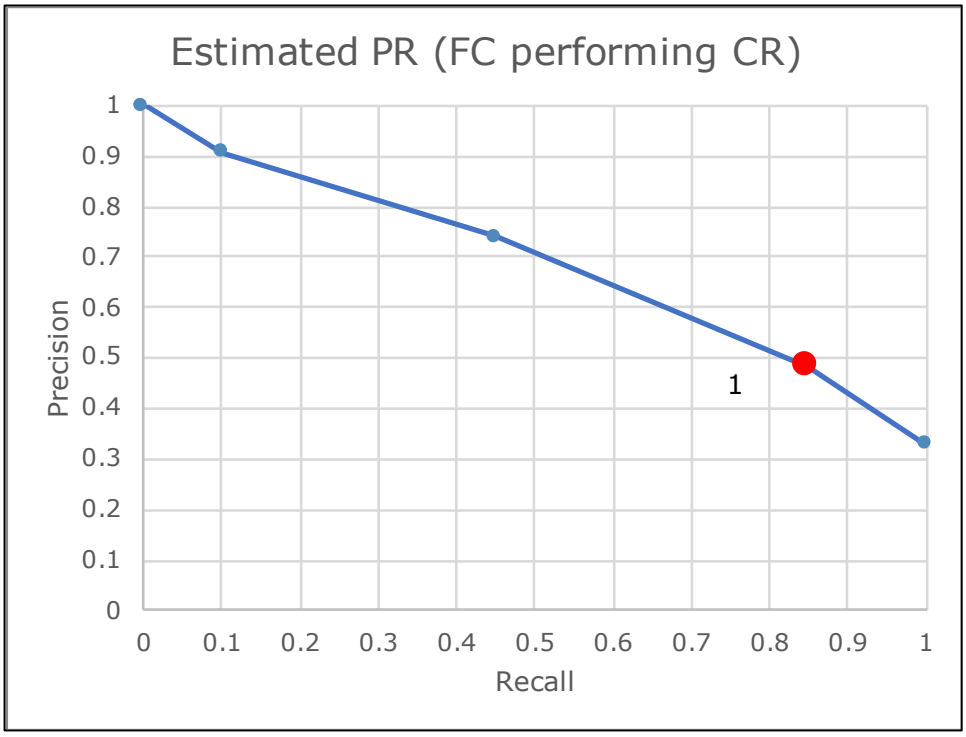


Figure 6.3 Expected PR curve when the FC applies the CR.

Table 6.2 and Figure 6.4 show, for any value of the global threshold, the expected values of precision and recall and the resulting F_1 and $F_{0.5}$ values for a FC applying the WFR. It is interesting to notice that $\overline{\gamma} = -0.2600$ (the chosen threshold), not only it maximizes J, but it also maximizes either F_1 or $F_{0.5}$.

		Precision	Recall	F_1	$F_{0.5}$
Threshold	-∞	0.3317	1	0.4982	0.3829
	-1.3400	0.4098	0.9273	0.5684	0.4612
	-0.3360	0.4479	0.8306	0.5819	0.4933
	-0.2720	0.4998	0.7374	0.5958	0.5342
	-0.2600	**0.5931**	**0.6269**	**0.6095**	**0.5995**
	0.7330	0.6136	0.4944	0.5476	0.5854
	0.7450	0.6661	0.3462	0.4556	0.5622
	0.8090	0.7764	0.2054	0.3250	0.4997
	+∞	1	0	0	0

Table 6.2 Expected values of precision, recall, *F₁* and *F₀.₅* at different thresholds for the FC using the WFR.

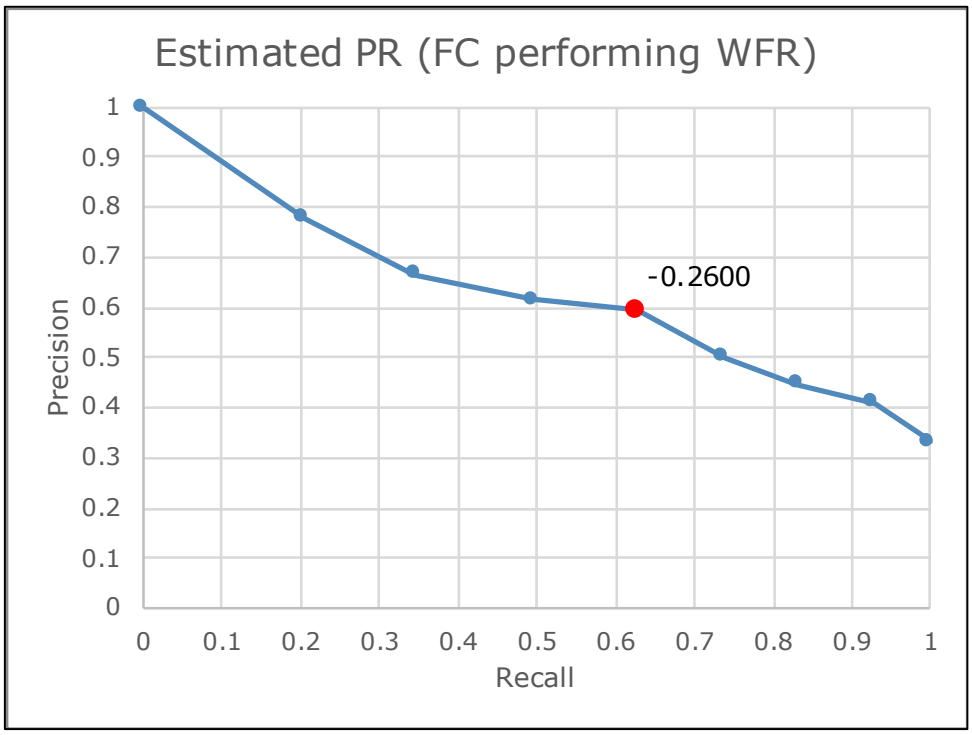


Figure 6.4 Expected PR curve when the FC applies the WFR.

This simulation tested different HSs as leakage source using four different methods:

- Detection: CR + Localization: CBM-TISP
  Index for the selection of the optimal sensor thresholds: J
- Detection: CR + Localization: CBM-TOSP
  Index for the selection of the optimal sensor thresholds: J
- Detection: WFR + Localization: BBM-TISP
  Index for the selection of the optimal sensor thresholds: CZ
- Detection: WFR + Localization: BBM-TOSP
  Index for the selection of the optimal sensor thresholds: CZ

About the Sensors/FC, tuning was carried out to have maximum reference value of J (the word "reference" is used because being the leakage point is a NCT, the index value is only an estimation based on the HSs).

For this reason, values reported in bold in Table 5.12, and Table 5.13 will be used when applying CR and WFR, respectively.

6.2 Leak Sources Tested

Three different HSs have been considering as leakage sources: the connection with the gas lift line (HS 3), a 5 1/8'' Hydraulic Valve functioning as Branch Valve (HS 10), and a 12'' Header Isolation Valve used for ROV connection (HS 17). Numbers are given according to Figure 5.2. These were for some specific reasons:

- The connection with the gas lift line (HS 3) is outside sensors' perimeter. This will evaluate how TOSP methods may improve localization with respect to TISP methods;
- The 5 1/8'' Hydraulic Valve - Branch Valve (HS 10) is in a central position, which means that no sensor is close to that HS. Detection could be affected in this case

because of the acoustic attenuation. Moreover, many HSs are present close to HS 10. This issue will check how the localization techniques perform when many HSs are close to the leaking one;

- The 12'' Header Isolation Valve - ROV Valve (HS 17) is the closest to a sensor (Sensor 2). The distance between HS 17 and Sensor 2, in fact, corresponds to l_{ref}. The fact of being close to a sensor will test if the WFR brings any improvement to the CR in this scenario.

6.3 Leak Detection Results

6.3.1 Leak from Connection with Gas Lift Line (Hotspot 3)

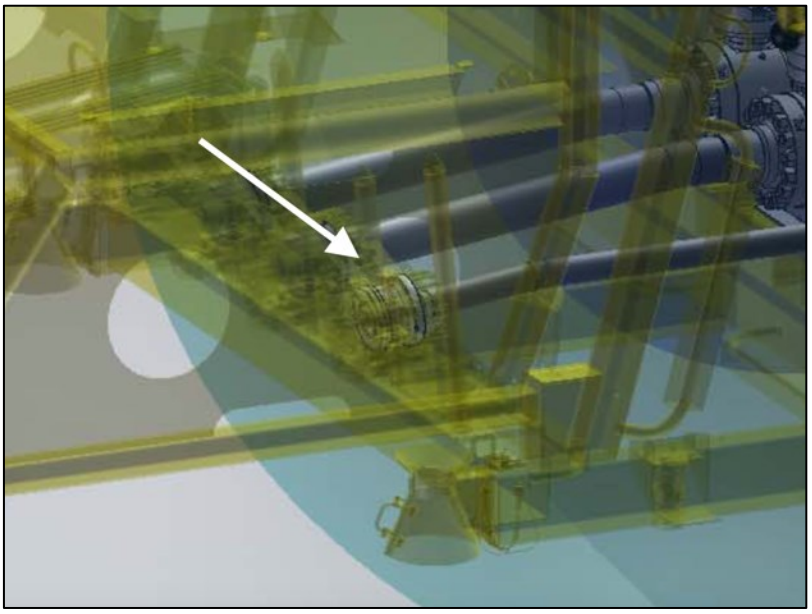


Figure 6.5 Close-up on the connection with the gas lift line (Røsby, 2011)

Signal amplitude features are shown: Figure 6.6 shows AAF over the distance target-sensor, while Figure 6.7 shows the received amplitudes at the different sensors. It is visible that sensors which are more distant to the target (experiencing a higher amplitude attenuation) tend to have a higher noise contribution.

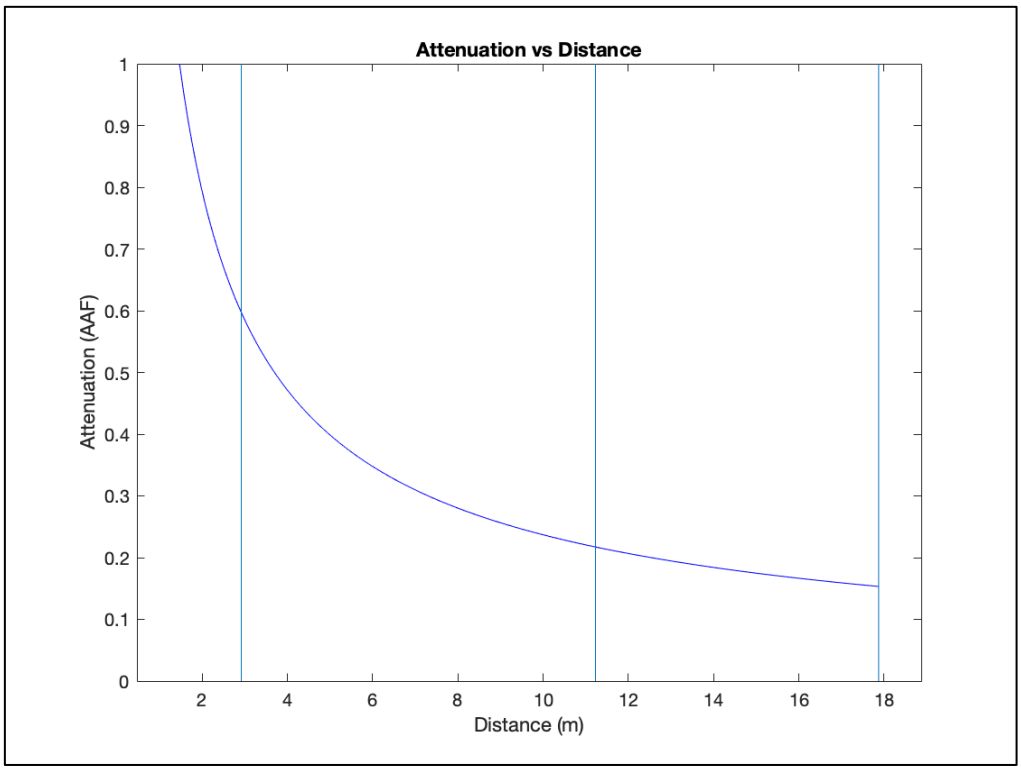


Figure 6.6 AAF vs. Distance. Hotspot 3 is the leakage source. From left to right, first vertical line is distance to Sens 1, second vertical line is distance to Sens 3, third vertical line is distance to Sens 2.

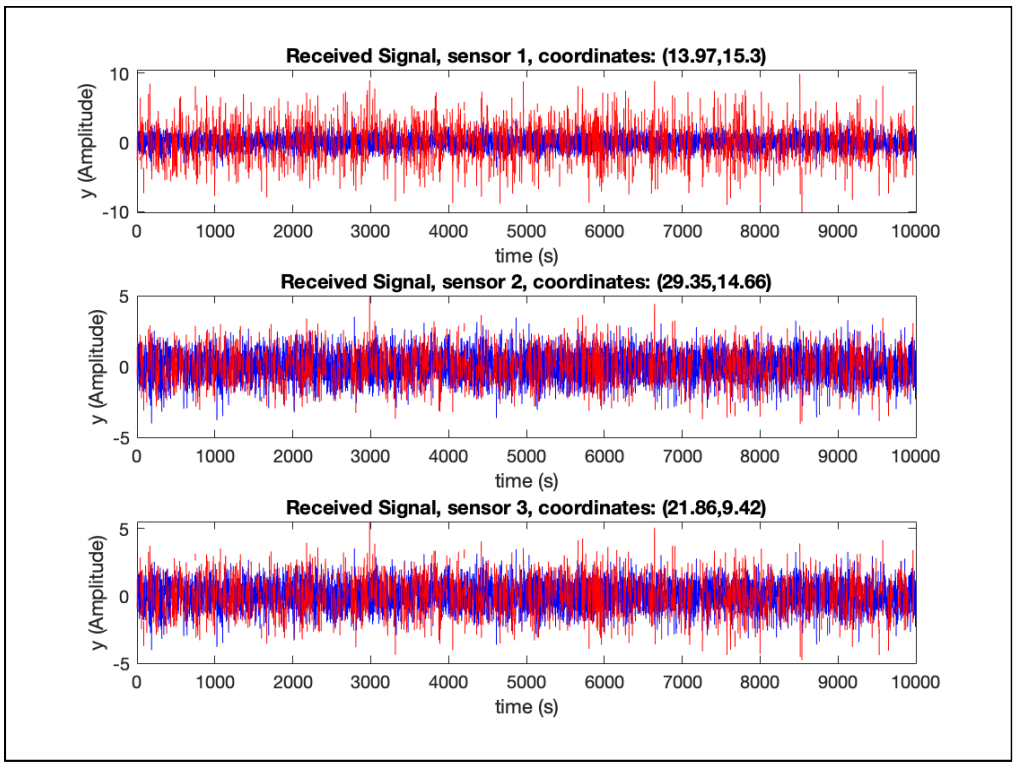


Figure 6.7 Time-domain (in seconds). Hotspot 3 is the leakage source. Red line is the signal associated with a positive event (leakage + noise). Blue line is the signal associated with a negative event (noise only).

6.3.1.1 Hotspot 3: Counting Rule

Table 6.3 shows the estimated performances of the sensors (average values obtained during tuning):

	Estimated		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.8336	1.9239	1.7378
Sensor Index	$\bar{J} = 0.2566$	$\bar{J} = 0.2735$	$\bar{J} = 0.2687$

Table 6.3 Local index *J* estimated during tuning of sensors in case of a FC performing CR.

Table 6.4 shows the real performances of the sensors:

	Real		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.8336	1.9239	1.7378
P_f	0.1757	0.1654	0.1874
P_d	0.6349	0.2526	0.3444
Sensor Index	$J = 0.4952$	$J = 0.0872$	$J = 0.1570$

Table 6.4 Real sensor performances in case Hotspot 3 is the leakage point and FC has to perform CR.

These performances can be plotted in the ROC space. Figure 6.8 shows the different ROC curves of the three sensors. Points represent real performances reported in Table 6.4. The curves show how local performances would have been in case sensors had been tuned using different local thresholds.

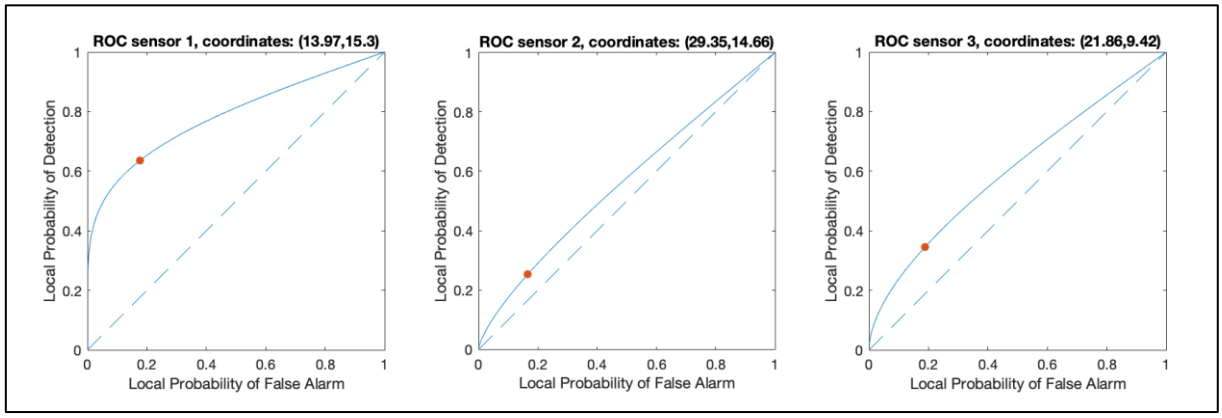


Figure 6.8 ROC curves of the three sensors when Hotspot 3 is the leakage source and FC uses CR. Red dots are the actual performances with the current local thresholds.

Figure 6.9 shows the probability distribution of the received signal amplitude dividing signals in positive (H_1) and negative (H_0) for any sensor. As shown in Figure 4.7, also here local thresholds for the ET are reported as vertical lines.

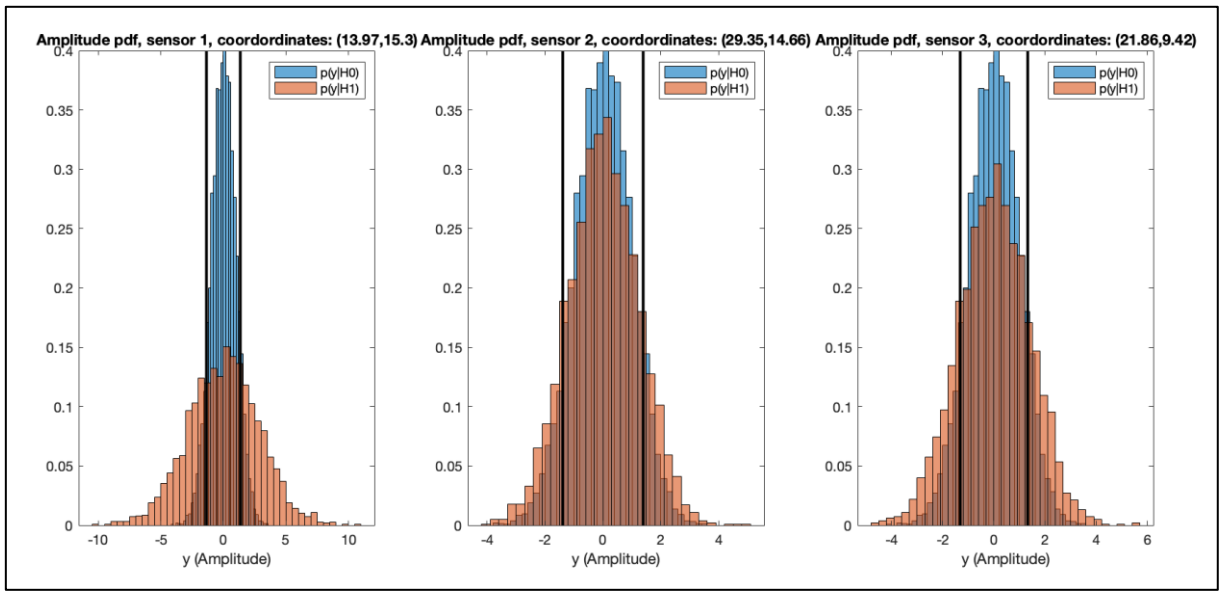


Figure 6.9 Amplitude distribution probability of received signals among the different sensors when Hotspot 3 is the leakage source when FC uses CR. Vertical lines represent the ET thresholds of each sensor.

Table 6.5 reports the estimated performances obtained during the tuning phase of the FC. Real values from the simulation are reported. Figure 6.10 shows, whereas, the estimated ROC curve obtained during the tuning phase where the estimated and the real performances have been highlighted.

	Global Threshold	Performances		Index
		Q_d	Q_f	J
Estimated	1	0.8480	0.4431	0.4049
Real		0.8241	0.4295	0.3946

Table 6.5 Comparison between estimated and real FC performances in case Hotspot 3 is the leakage source and FC performs CR.

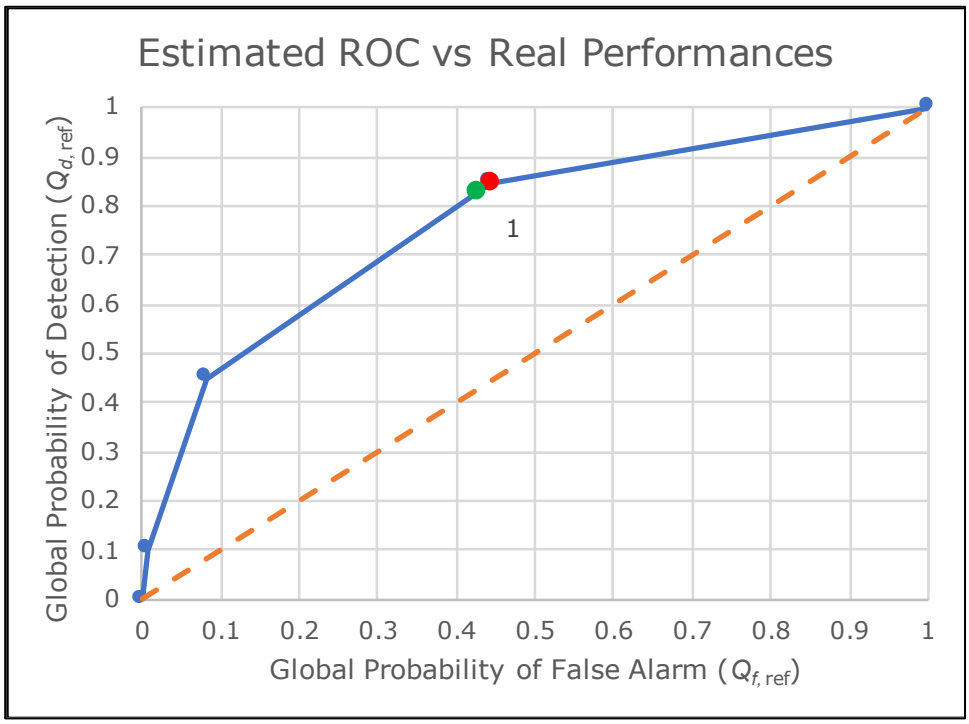


Figure 6.10 ROC curve of performances estimated during tuning of FC in case of CR. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance when Hotspot 3 is the leakage source.

Now that the imbalance between the two types of event is known, also precision can be evaluated. Table 6.6 shows two different values of F_1 and $F_{0.5}$. The estimated value is in case FC performances were as estimated during the tuning phase; the real value is the one obtained from the simulation. Moreover, Figure 6.11 shows the entire PR-curve for all the thresholds in case FC performances were those estimated during the tuning phase.

	Global Threshold	Precision	Recall	F_1	$F_{0.5}$
Estimated	1	0.4844	0.8480	0.6166	0.5298
Real		0.5084	0.8241	0.6289	0.5506

Table 6.6 Comparison between estimated and real FC values of precision and recall in case Hotspot 3 is the leakage source and FC performs CR.

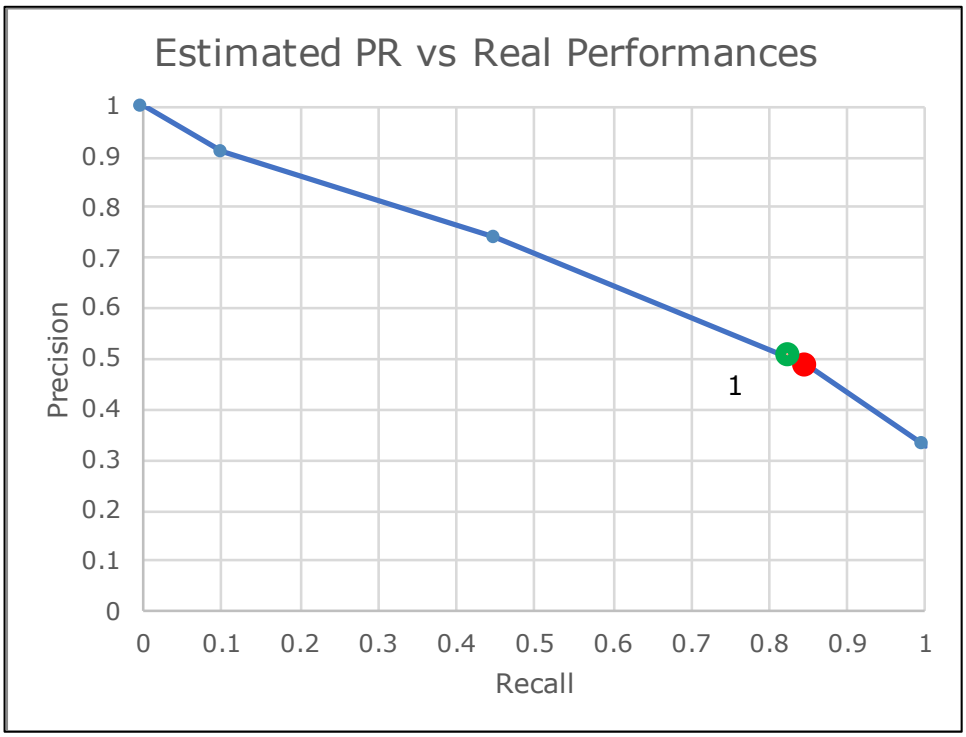


Figure 6.11 PR curve in case Hotspot 3 is the leakage source and FC is performing CR with the performances estimated during FC tuning. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance.

6.3.1.2 Hotspot 3: Weighted Fusion Rule

Table 6.7 shows the estimated performances of the sensors (average values obtained during tuning):

	Estimated		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.0346	1.0921	1.0592
Sensor Index	$\overline{CZ} = 0.3799$	$\overline{CZ} = 0.3895$	$\overline{CZ} = 0.3894$

Table 6.7 Local index CZ estimated during tuning of sensors in case of a FC performing WFR.

Table 6.8 shows the real performances of the sensors:

	Real		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.0346	1.0921	1.0592
P_f	0.3091	0.2960	0.3034
P_d	0.7213	0.3887	0.4604
Sensor Index	$CZ = 0.4983$	$CZ = 0.2736$	$CZ = 0.3207$

Table 6.8 Real sensor performances in case Hotspot 3 is the leakage point and FC has to perform WFR.

These performances can be plotted in the ROC space. Figure 6.12 shows the different ROC curves of the three sensors. Points represent real performances reported in Table 6.8. The curves show how local performances would have been in case sensors had been tuned using different local thresholds.

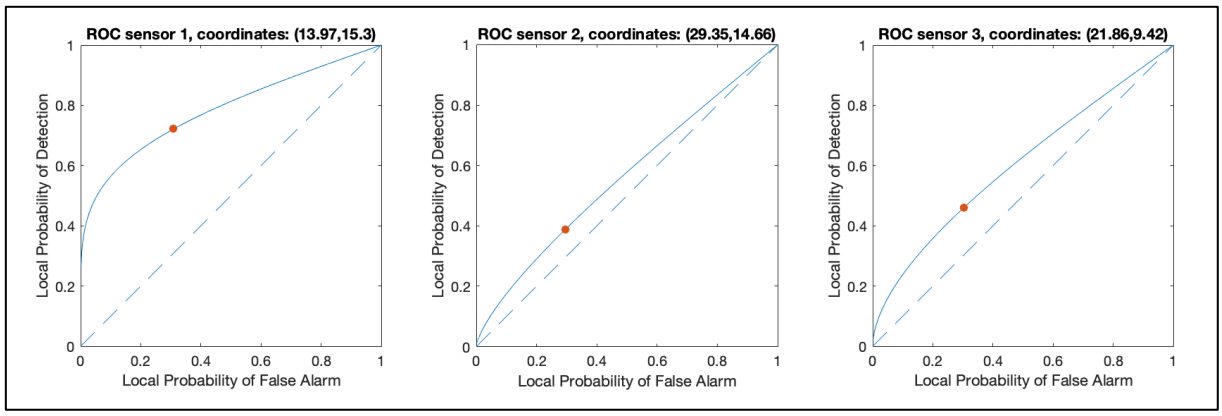


Figure 6.12 ROC curves of the three sensors when Hotspot 3 is the leakage source and FC uses WFR. Red dots are the actual performances with the current local thresholds.

Figure 6.13 shows the probability distribution of the received signal amplitude dividing signals in positive (H_1) and negative (H_0) for any sensor. As shown in Figure 4.7, also here local thresholds for the ET are reported as vertical lines.

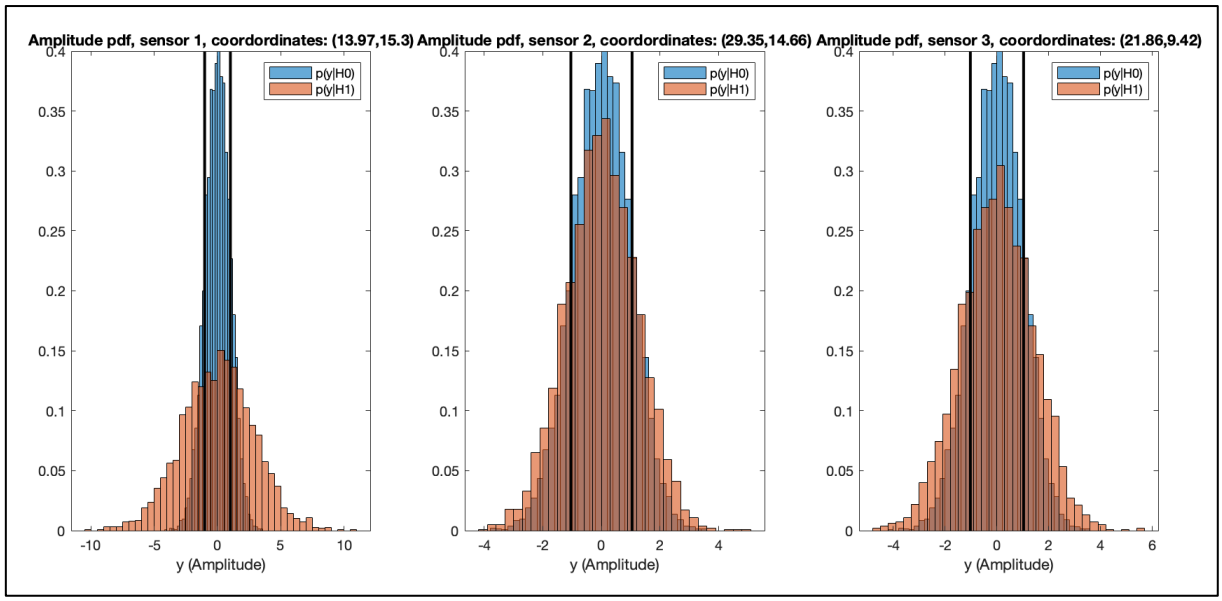


Figure 6.13 Amplitude distribution probability of received signals among the different sensors when Hotspot 3 is the leakage source when FC uses WFR. Vertical lines represent the ET thresholds of each sensor.

Table 6.9 reports the estimated performances obtained during the tuning phase of the FC. Real values from the simulation are reported. Figure 6.14 shows, whereas, the estimated ROC curve obtained during the tuning phase where the estimated and the real performances have been highlighted.

	Global Threshold	Performances		Index
		Q_d	Q_f	J
Estimated	-0.2600	0.6269	0.2088	0.4181
Real		0.5453	0.2251	0.3202

Table 6.9 Comparison between estimated and real FC performances in case Hotspot 3 is the leakage source and FC performs WFR.

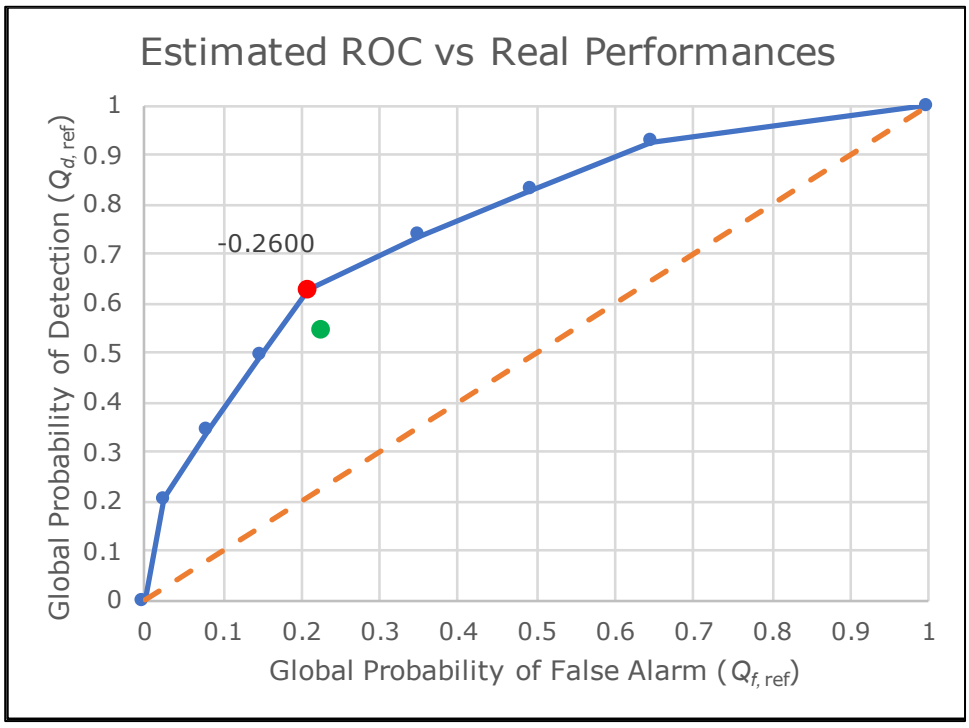


Figure 6.14 ROC curve of performances estimated during tuning of FC in case of WFR. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance when Hotspot 3 is the leakage source.

Now that the imbalance between the two types of events is known, also precision can be evaluated. Table 6.10 shows two different values of F_1. The estimated value is in case FC performances were as estimated during the tuning phase; the real value is the one obtained from the simulation. Moreover, Figure 6.11 shows the entire PR-curve for all the thresholds in case FC performances were those estimated during the tuning phase.

	Global Threshold	Precision	Recall	F_1	$F_{0.5}$
Estimated	-0.2600	0.5931	0.6269	0.6095	0.5995
Real		0.5390	0.5453	0.5421	0.5403

Table 6.10 Comparison between estimated and real FC values of precision and recall in case Hotspot 3 is the leakage source and FC performs WFR.

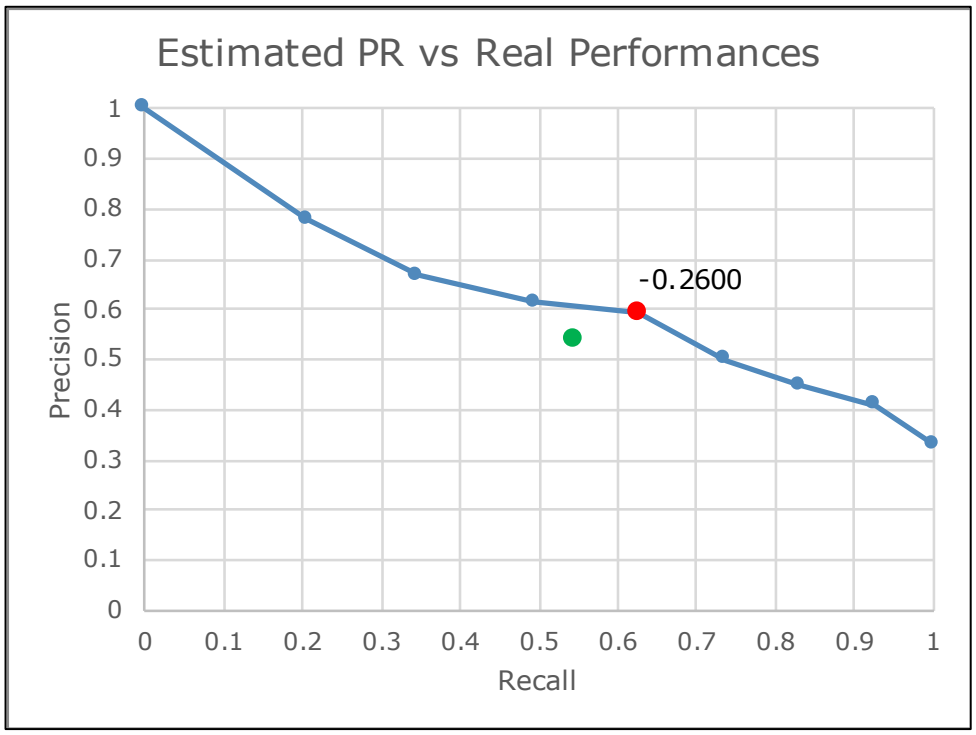


Figure 6.15 PR curve in case Hotspot 3 is the leakage source and FC is performing WFR with the performances estimated during FC tuning. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance.

6.3.2 Leak from Branch Valve (Hotspot 10)

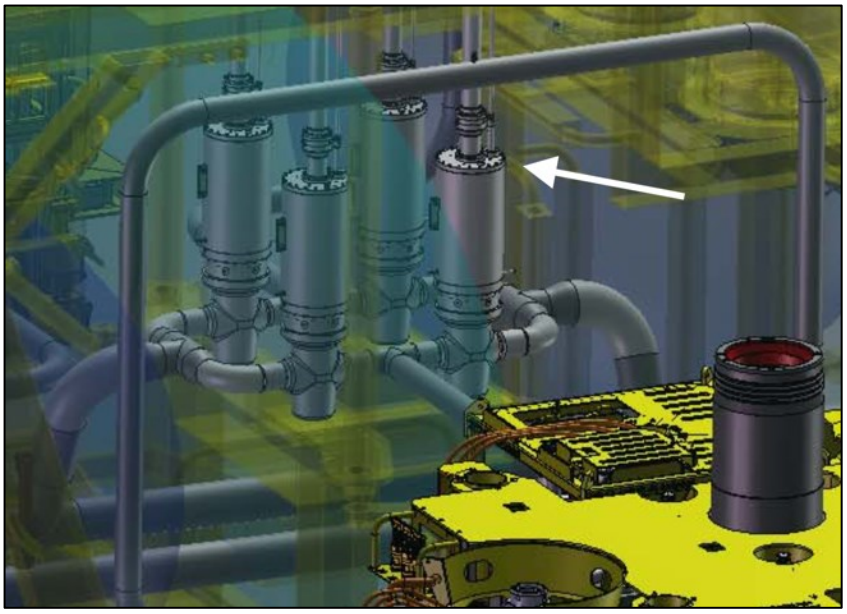


Figure 6.16 Close-up on the 5 1/8'' Hydraulic Valve - Branch Valve (Røsby, 2011)

Signal amplitude features are shown: Figure 6.17 shows AAF over the distance target-sensor, while Figure 6.18 shows the received amplitudes at the different sensors. It is visible that sensors which are more distant to the target (experiencing a higher amplitude attenuation) tend to have a higher noise contribution.

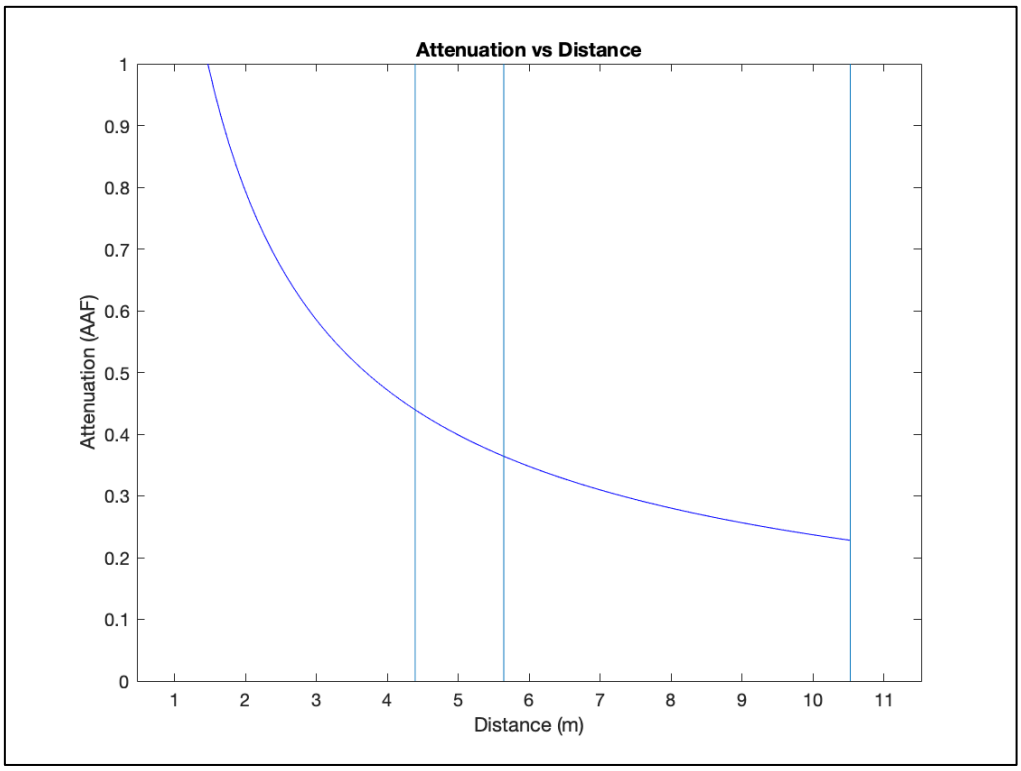


Figure 6.17 AAF vs Distance. Hotspot 10 is the leakage source. From left to right, first vertical line is distance to Sens 3, second vertical line is distance to Sens 1, third vertical line is distance to Sens 2.

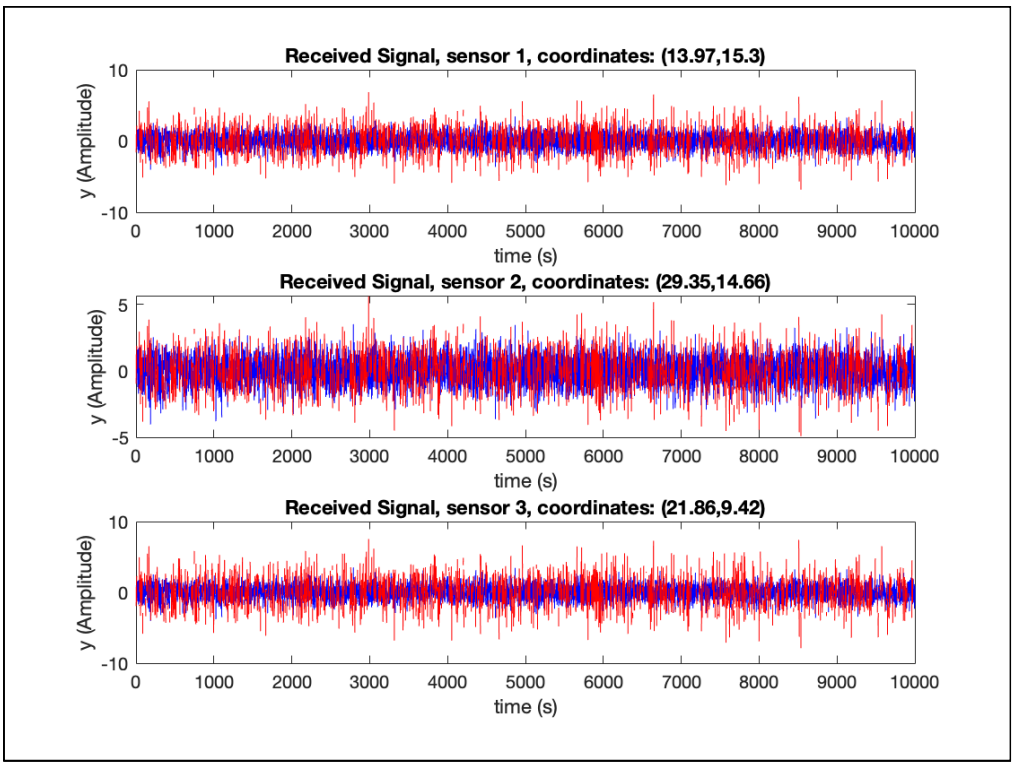


Figure 6.18 Time-domain (in seconds). Hotspot 10 is the leakage source. Red line is the signal associated with a positive event (leakage + noise). Blue line is the signal associated with a negative event (noise only).

6.3.2.1 Hotspot 10: Counting Rule

Table 6.11 shows estimated performances of the sensors (average values obtained during tuning):

	Estimated		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.8336	1.9239	1.7378
Sensor Index	$\bar{J} = 0.2566$	$\bar{J} = 0.2735$	$\bar{J} = 0.2687$

Table 6.11 Local index J estimated during tuning of sensors in case of a FC performing CR.

Table 6.12 shows real performances of the sensors:

	Real		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.8336	1.9239	1.7378
P_f	0.1757	0.1654	0.1874
P_d	0.4786	0.3315	0.5499
Sensor Index	$J = 0.3029$	$J = 0.1661$	$J = 0.3625$

Table 6.12 Real sensor performances in case Hotspot 10 is the leakage point and FC has to perform CR.

These performances can be plotted in the ROC space. Figure 6.19 shows the different ROC curves of the three sensors. Points represent real performances reported in Table 6.12. The curves show how local performances would have been in case sensors had been tuned using different local thresholds.

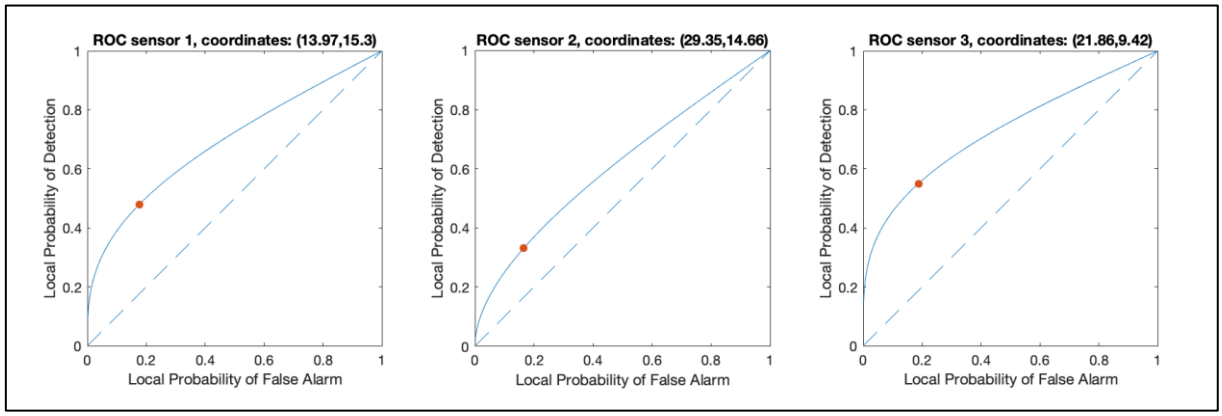


Figure 6.19 ROC curves of the three sensors when Hotspot 10 is the leakage source and FC uses CR. Red dots are the actual performances with the current local thresholds.

Figure 6.20 shows the probability distribution of the received signal amplitude dividing signals in positive (H_1) and negative (H_0) for any sensor. As shown in Figure 4.7, also here local thresholds for the ET are reported as vertical lines.

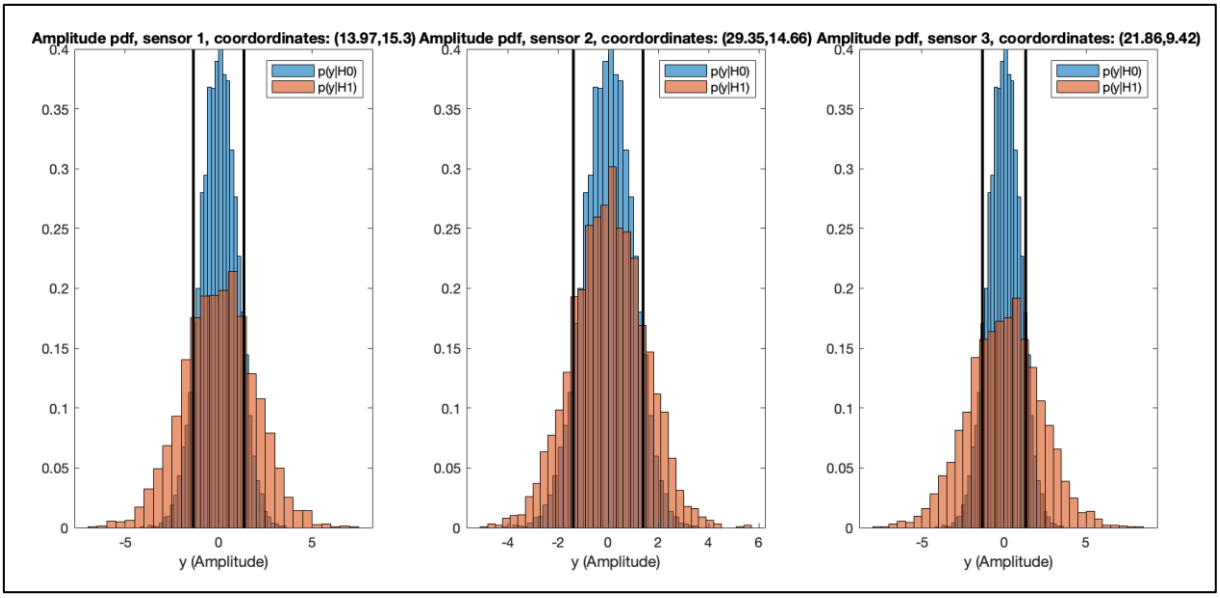


Figure 6.20 Amplitude distribution probability of received signals among the different sensors when Hotspot 10 is the leakage source when FC uses CR. Vertical lines represent the ET thresholds of each sensor.

Table 6.13 reports the estimated performances obtained during the tuning phase of the FC. Real values from the simulation are reported. Figure 6.21 shows, whereas, the estimated ROC curve obtained during the tuning phase where the estimated and the real performances have been highlighted.

	Global Threshold	Performances		Index
		Q_d	Q_f	J
Estimated	1	0.8480	0.4431	0.4049
Real		0.8375	0.4557	0.3818

Table 6.13 Comparison between estimated and real FC performances in case Hotspot 3 is the leakage source and FC performs CR.

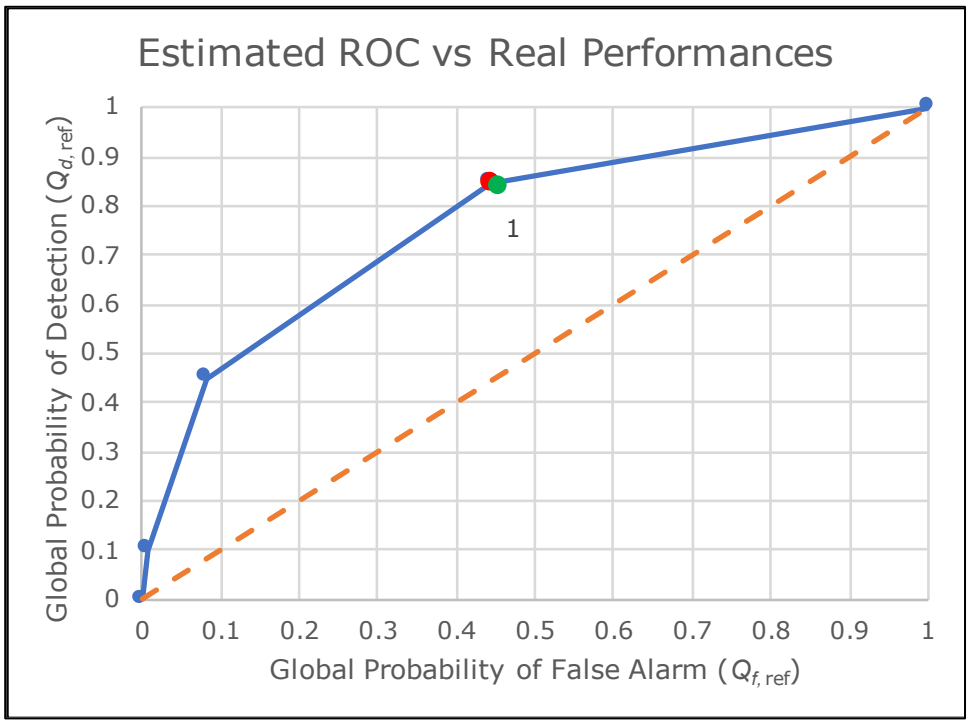


Figure 6.21 ROC curve of performances estimated during tuning of FC in case of CR. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance when Hotspot 10 is the leakage source.

Now that the imbalance between the two types of event is known, also precision can be evaluated. Table 6.14 shows two different values of F_1 and $F_{0.5}$. The estimated value is in case FC performances were as estimated during the tuning phase; the real value is the one obtained from the simulation. Moreover, Figure 6.22 shows the entire PR-curve for all the thresholds in case FC performances were those estimated during the tuning phase.

	Global Threshold	Precision	Recall	F_1	$F_{0.5}$
Estimated	1	0.4844	0.8480	0.6166	0.5298
Real		0.4737	0.8375	0.6051	0.5188

Table 6.14 Comparison between estimated and real FC values of precision and recall in case Hotspot 10 is the leakage source and FC performs CR.

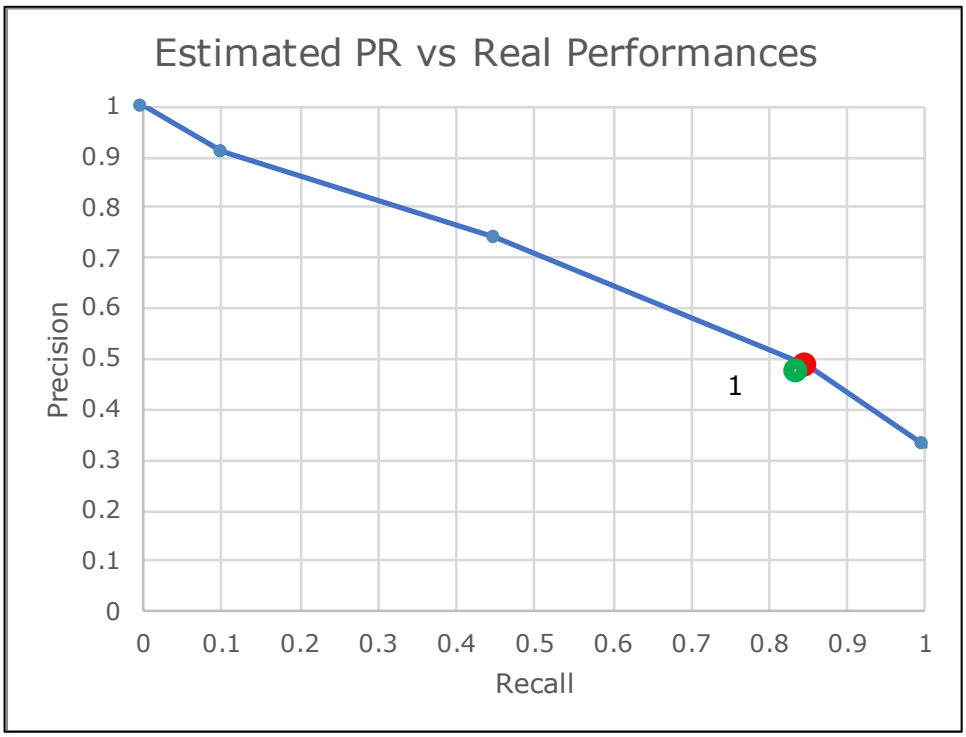


Figure 6.22 PR curve in case Hotspot 10 is the leakage source and FC is performing CR with the performances estimated during FC tuning phase. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance.

6.3.2.2 Hotspot 10: Weighted Fusion Rule

Table 6.15 shows estimated performances of the sensors (average values obtained during tuning):

	Estimated		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.0346	1.0921	1.0592
Sensor Index	$\overline{CZ} = 0.3799$	$\overline{CZ} = 0.3895$	$\overline{CZ} = 0.3894$

Table 6.15 Local index CZ estimated during tuning of sensors in case of a FC performing WFR.

Table 6.16 shows real performances of the sensors:

	Real		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.0346	1.0921	1.0592
$\boldsymbol{P_f}$	0.3091	0.2960	0.3034
$\boldsymbol{P_d}$	0.5946	0.4644	0.6407
Sensor Index	$CZ = 0.4108$	$CZ = 0.3269$	$CZ = 0.4463$

Table 6.16 Real sensor performances in case Hotspot 10 is the leakage point and FC has to perform WFR.

These performances can be plotted in the ROC space. Figure 6.23 shows the different ROC curves of the three sensors. Points represent real performances reported in Table 6.16. The curves show how local performances would have been in case sensors had been tuned using different local thresholds.

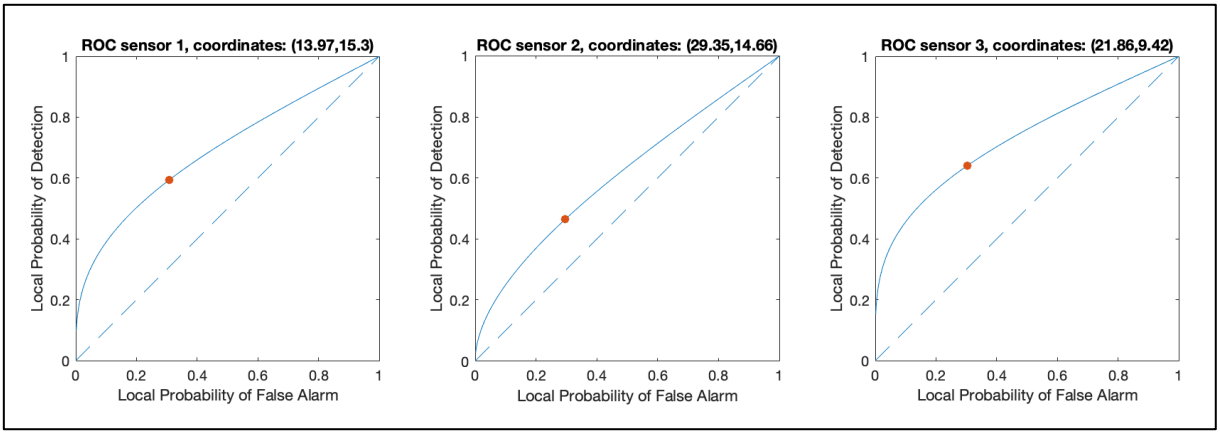


Figure 6.23 ROC curves of the three sensors when Hotspot 10 is the leakage source and FC uses WFR. Red dots are the actual performances with the current local thresholds.

Figure 6.24 shows the probability distribution of the received signal amplitude dividing signals in positive (H_1) and negative (H_0) for any sensor. As shown in Figure 4.7, also here local thresholds for the ET are reported as vertical lines.

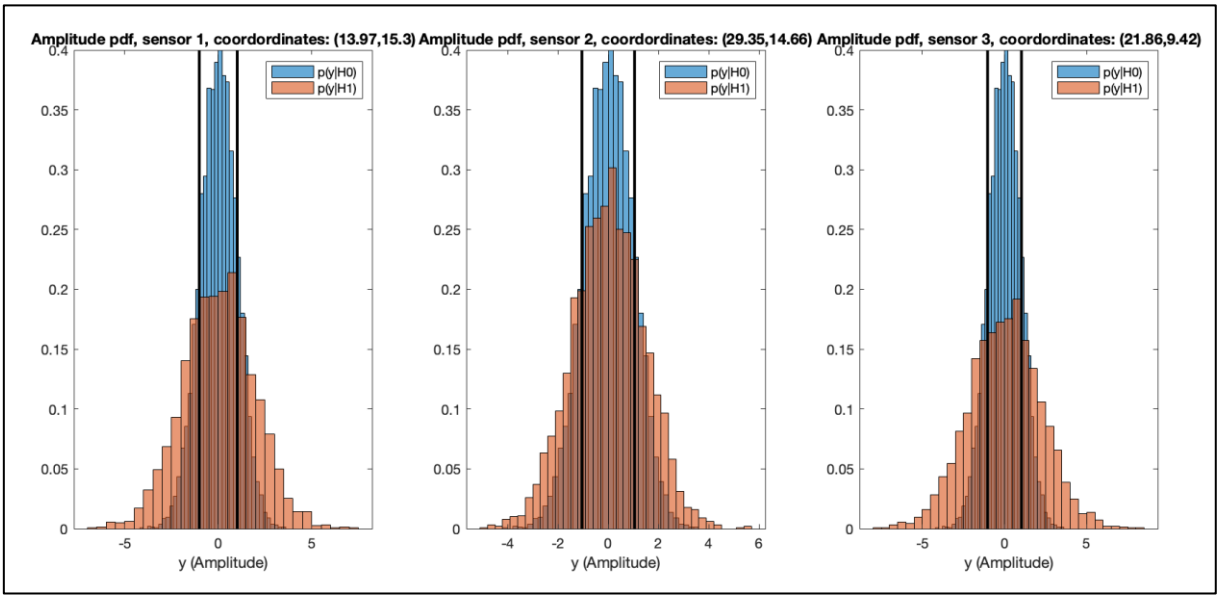


Figure 6.24 Amplitude distribution probability of received signals among the different sensors when Hotspot 10 is the leakage source when FC uses WFR. Vertical lines represent the ET thresholds of each sensor.

Table 6.17 reports the estimated performances obtained during the tuning phase of the FC. Real values from the simulation are reported. Figure 6.25 shows, whereas, the estimated ROC curve obtained during the tuning phase where the estimated and the real performances have been highlighted.

	Global Threshold	Performances		Index
		Q_d	Q_f	J
Estimated	-0.2600	0.6269	0.2088	0.4181
Real		0.6005	0.2234	0.3771

Table 6.17 Comparison between estimated and real FC performances in case Hotspot 10 is the leakage source and FC performs WFR.

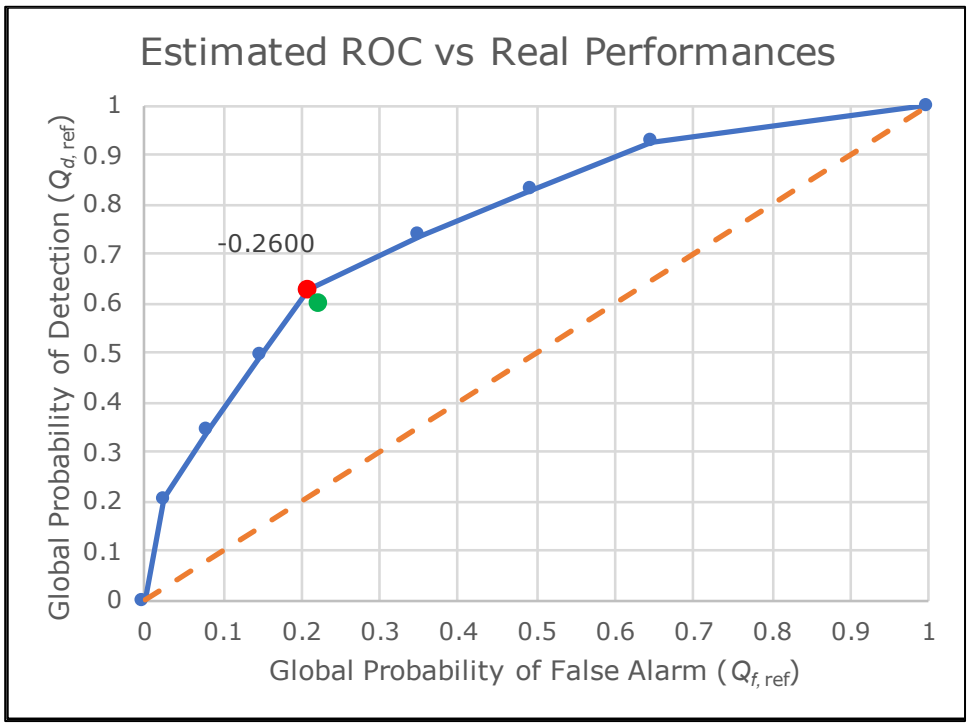


Figure 6.25 ROC curve of performances estimated during tuning of FC in case of WFR. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance when Hotspot 10 is the leakage source.

Now that the imbalance between the two types of events is known, also precision can be evaluated. Table 6.10 shows two different values of F_1 and $F_{0.5}$. The estimated value is in case FC performances were as estimated during the tuning phase; the real value is the one obtained from the simulation. Moreover, Figure 6.11 shows the entire PR-curve for all the thresholds in case FC performances were those estimated during the tuning phase.

	Global Threshold	Precision	Recall	F_1	$F_{0.5}$
Estimated	-0.2600	0.5931	0.6269	0.6095	0.5995
Real		0.5682	0.6005	0.5839	0.5744

Table 6.18 Comparison between estimated and real FC values of precision and recall in case Hotspot 10 is the leakage source and FC performs WFR.

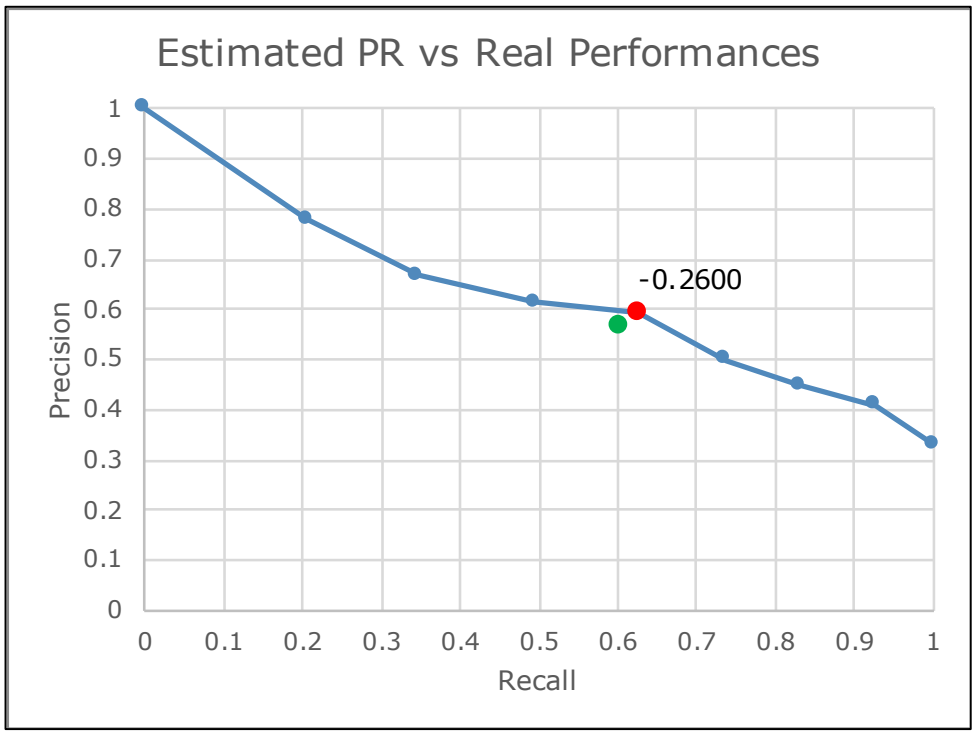


Figure 6.26 PR curve in case Hotspot 10 is the leakage source and FC is performing WFR with the performances estimated during FC tuning. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance.

6.3.3 Leak from Header Isolation Valve – ROV Valve (Hotspot 17)

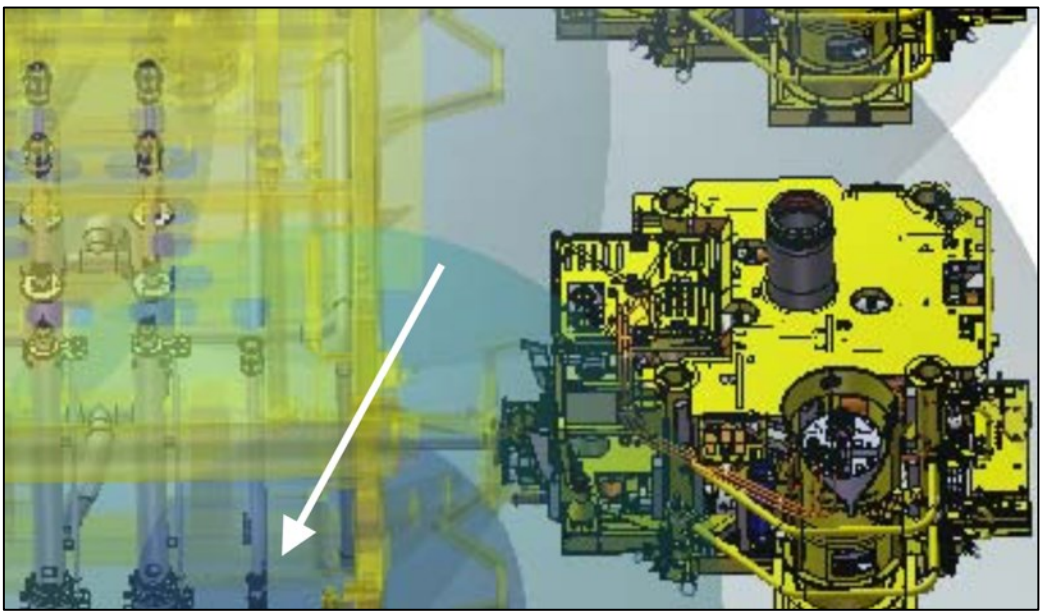


Figure 6.27 Close-up on the 12'' Header Isolation Valve - ROV Valve (Røsby, 2011)

Signal amplitude features are shown: Figure 6.28 shows AAF over the distance target-sensor, while Figure 6.29 shows the received amplitudes at the different sensors. It is visible that sensors which are more distant to the target (experiencing a higher amplitude attenuation) tend to have a higher noise contribution.

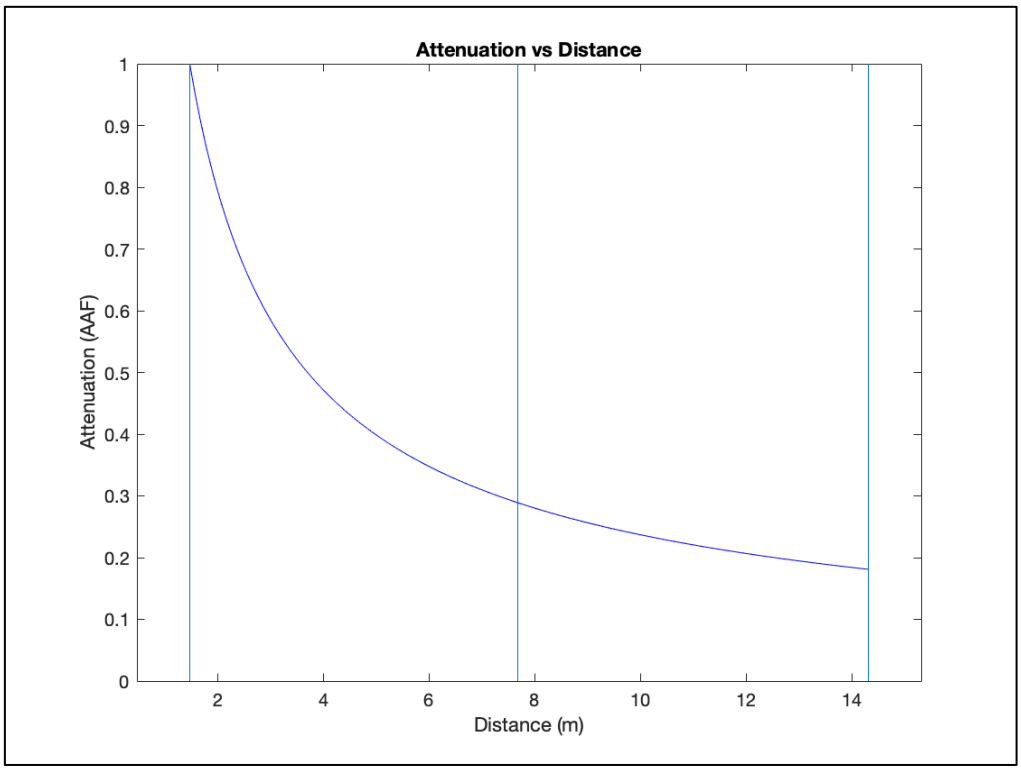


Figure 6.28 AAF vs Distance. Hotspot 17 is the leakage source. From left to right, first vertical line is distance to Sens 2, second vertical line is distance to Sens 3, third vertical line is distance to Sens 1.

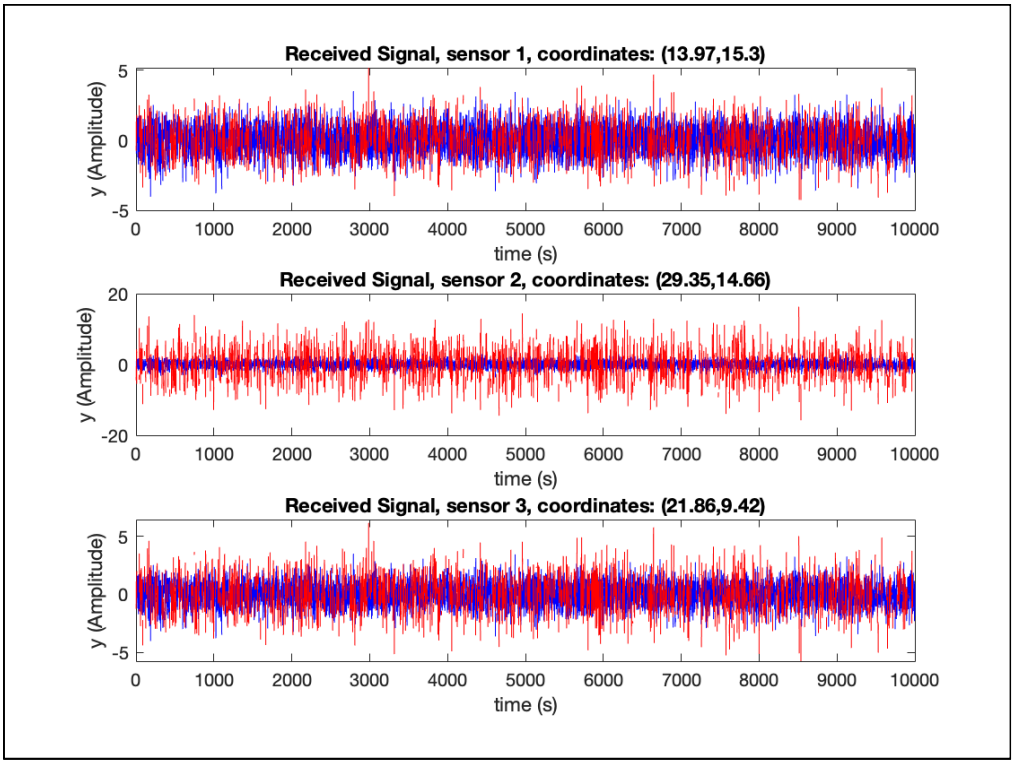


Figure 6.29 Time-domain (in seconds). Hotspot 17 is the leakage source. Red line is the signal associated with a positive event (leakage + noise). Blue line is the signal associated with a negative event (noise only).

6.3.3.1 Hotspot 17: Counting Rule

Table 6.19 shows estimated performances of the sensors (average values obtained during tuning):

	Estimated		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.8336	1.9239	1.7378
Sensor Index	$\bar{J} = 0.2566$	$\bar{J} = 0.2735$	$\bar{J} = 0.2687$

Table 6.19 Local index J estimated during tuning of sensors in case of a FC performing CR.

Table 6.20 shows real performances of the sensors:

	Real		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.8336	1.9239	1.7378
P_f	0.1757	0.1654	0.1874
P_d	0.2927	0.7621	0.4198
Sensor Index	$J = 0.1170$	$J = 0.5967$	$J = 0.2324$

Table 6.20 Real sensor performances in case Hotspot 17 is the leakage point and FC has to perform CR.

These performances can be plotted in the ROC space. Figure 6.30 shows the different ROC curves of the three sensors. Points represent real performances reported in Table 6.20. The curves show how local performances would have been in case sensors had been tuned using different local thresholds.

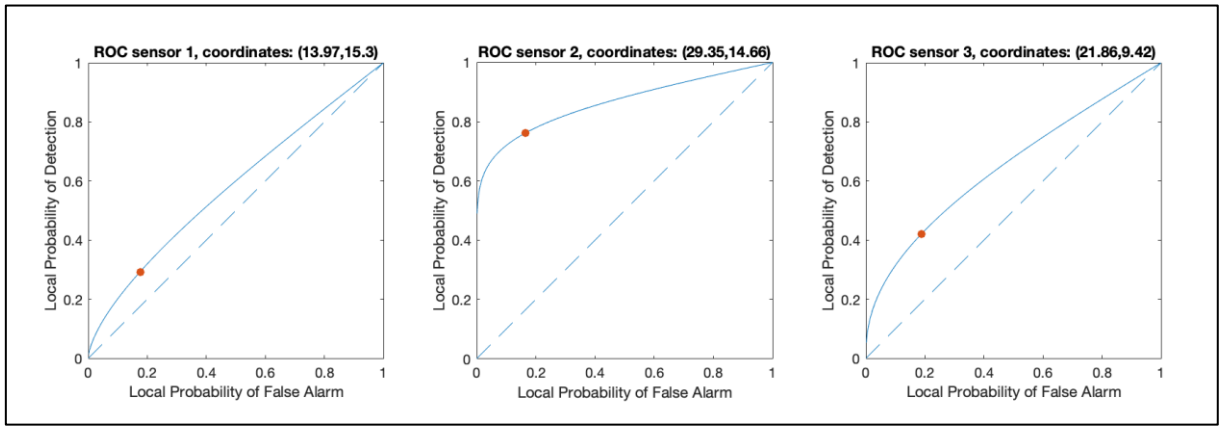


Figure 6.30 ROC curves of the three sensors when Hotspot 17 is the leakage source and FC uses CR. Red dots are the actual performances with the current local thresholds.

Figure 6.31 shows the probability distribution of the received signal amplitude dividing signals in positive (H_1) and negative (H_0) for any sensor. As shown in Figure 4.7, also here local thresholds for the ET are reported as vertical lines.

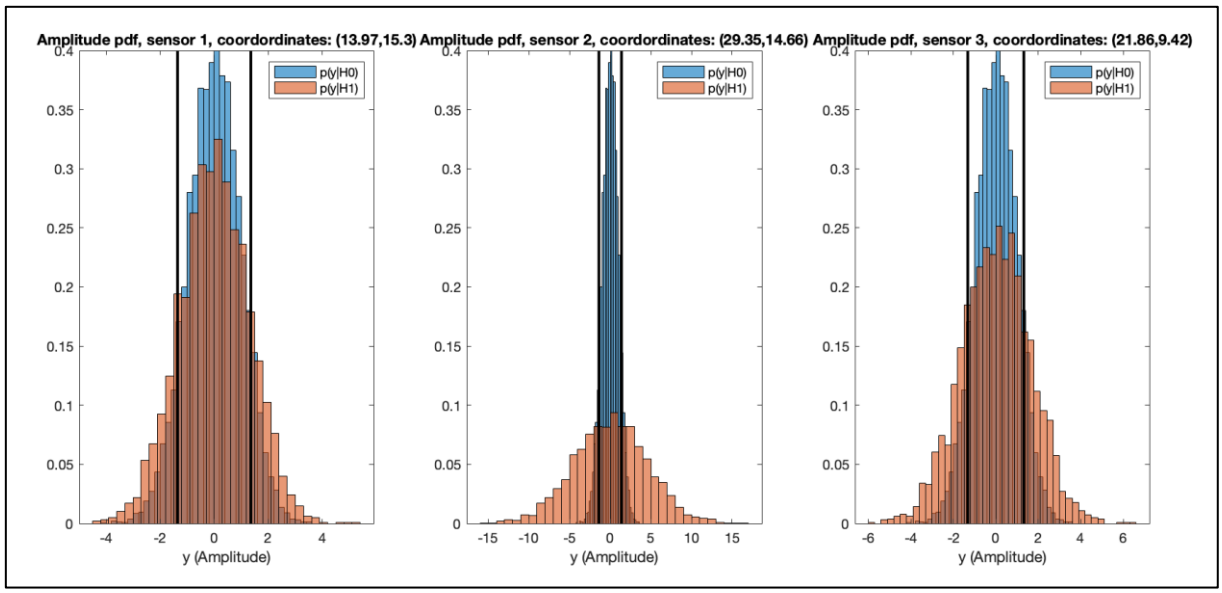


Figure 6.31 Amplitude distribution probability of received signals among the different sensors when Hotspot 17 is the leakage source when FC uses CR. Vertical lines represent the ET thresholds of each sensor.

Table 6.21 reports the estimated performances obtained during the tuning phase of the FC. Real values from the simulation are reported. Figure 6.32 shows, whereas, the estimated ROC curve obtained during the tuning phase where the estimated and the real performances have been highlighted.

	Global Threshold	Performances		Index
		Q_d	Q_f	J
Estimated	1	0.8480	0.4431	0.4049
Real		0.9083	0.4332	0.4751

Table 6.21 Comparison between estimated and real FC performances in case Hotspot 17 is the leakage source and FC performs CR.

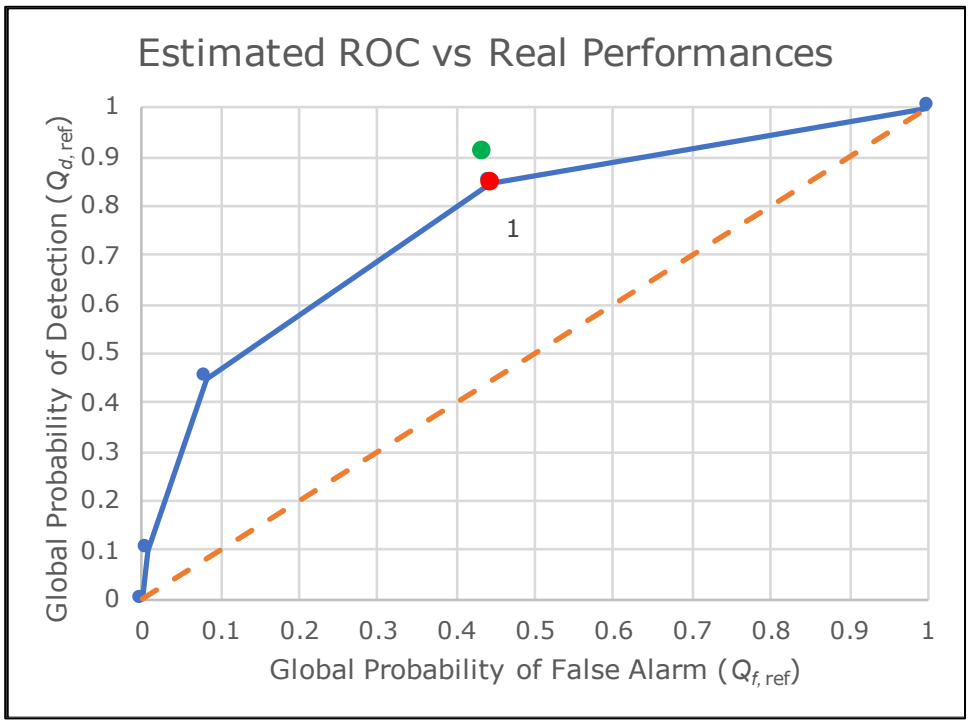


Figure 6.32 ROC curve of performances estimated during tuning of FC in case of CR. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance when Hotspot 17 is the leakage source.

Now that the imbalance between the two types of event is known, also precision can be evaluated. Table 6.22 shows two different values of F_1 and $F_{0.5}$. The estimated value is in case FC performances were as estimated during the tuning phase; the real value is the one obtained from the simulation. Moreover, Figure 6.33 shows the entire PR-curve for all the thresholds in case FC performances were those estimated during the tuning phase.

	Global Threshold	Precision	Recall	F_1	$F_{0.5}$
Estimated	1	0.4844	0.8480	0.6166	0.5298
Real		0.5097	0.9083	0.6530	0.5587

Table 6.22 Comparison between estimated and real FC values of precision and recall in case Hotspot 17 is the leakage source and FC performs CR.

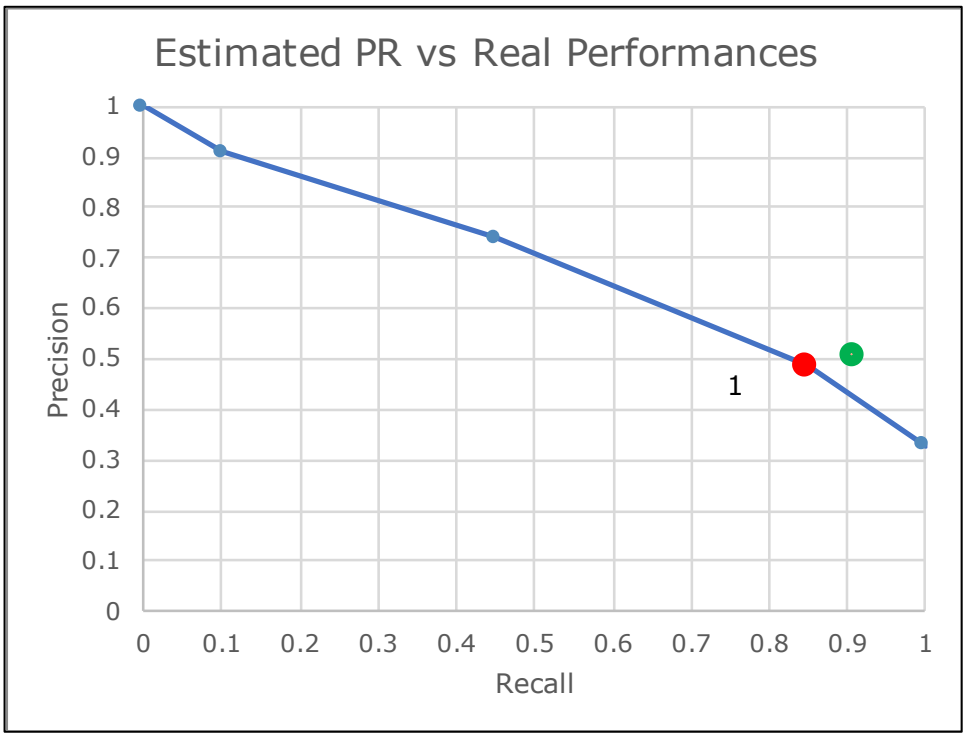


Figure 6.33 PR curve in case Hotspot 17 is the leakage source and FC is performing CR with the performances estimated during FC tuning. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance.

6.3.3.2 Hotspot 17: Weighted Fusion Rule

Table 6.23 shows estimated performances of the sensors (average values obtained during tuning):

	Estimated		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.0346	1.0921	1.0592
Sensor Index	$\overline{CZ} = 0.3799$	$\overline{CZ} = 0.3895$	$\overline{CZ} = 0.3894$

Table 6.23 Local index CZ estimated during tuning of sensors in case of a FC performing WFR.

Table 6.24 shows real performances of the sensors:

	Real		
	Sensor 1	Sensor 2	Sensor 3
Local Threshold	1.0346	1.0921	1.0592
P_f	0.3091	0.2960	0.3034
P_d	0.4293	0.8196	0.5288
Sensor Index	$CZ = 0.2966$	$CZ = 0.5770$	$CZ = 0.3684$

Table 6.24 Real sensor performances in case Hotspot 17 is the leakage point and FC has to perform WFR.

These performances can be plotted in the ROC space. Figure 6.34 shows the different ROC curves of the three sensors. Points represent real performances reported in Table 6.24. The curves show how local performances would have been in case sensors had been tuned using different local thresholds.

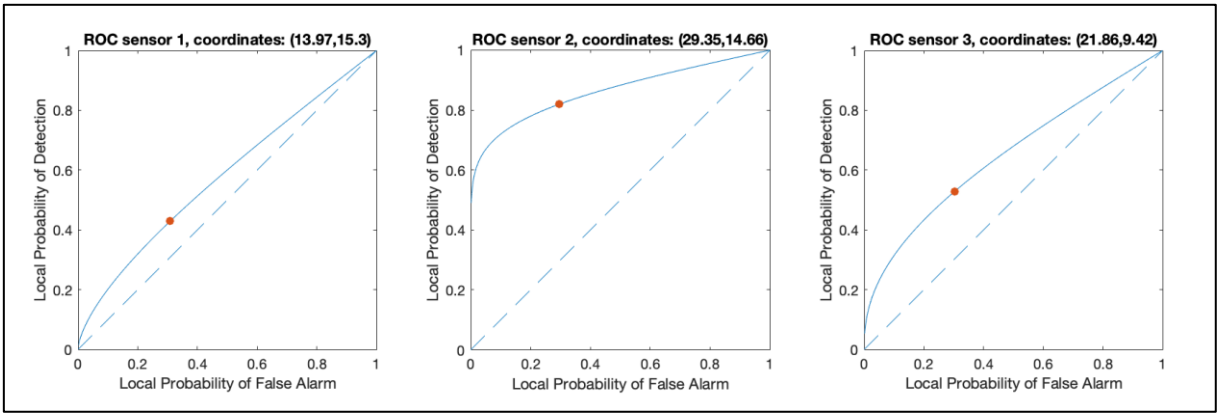


Figure 6.34 ROC curves of the three sensors when Hotspot 17 is the leakage source and FC uses WFR. Red dots are the actual performances with the current local thresholds.

Figure 6.35 shows the probability distribution of received signal amplitude dividing signals in positive (H_1) and negative (H_0) for any sensor. As shown in Figure 4.7, also here local thresholds for the ET are reported as vertical lines.

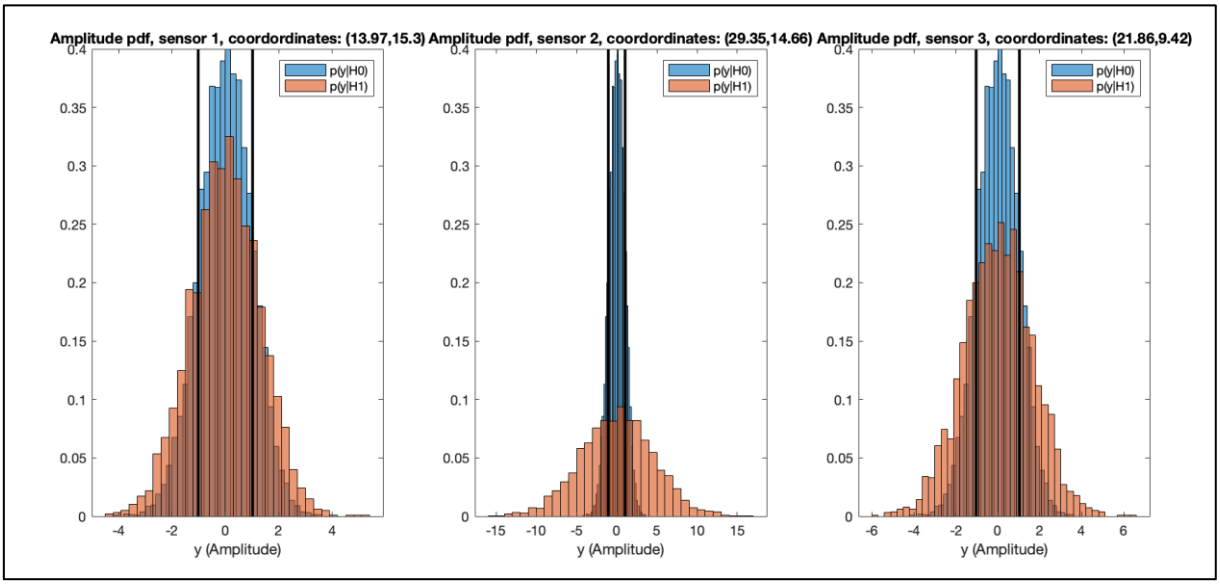


Figure 6.35 Amplitude distribution probability of received signals among the different sensors when Hotspot 17 is the leakage source when FC uses WFR. Vertical lines represent the ET thresholds of each sensor.

Table 6.25 reports the estimated performances obtained during the tuning phase of the FC. Real values from the simulation are reported. Figure 6.36 shows, whereas, the estimated ROC curve obtained during the tuning phase where the estimated and the real performances have been highlighted.

	Global Threshold	Performances		Index
		Q_d	Q_f	J
Estimated	-0.2600	0.6269	0.2088	0.4181
Real		0.6378	0.2343	0.3035

Table 6.25 Comparison between estimated and real FC performances in case Hotspot 17 is the leakage source and FC performs WFR.

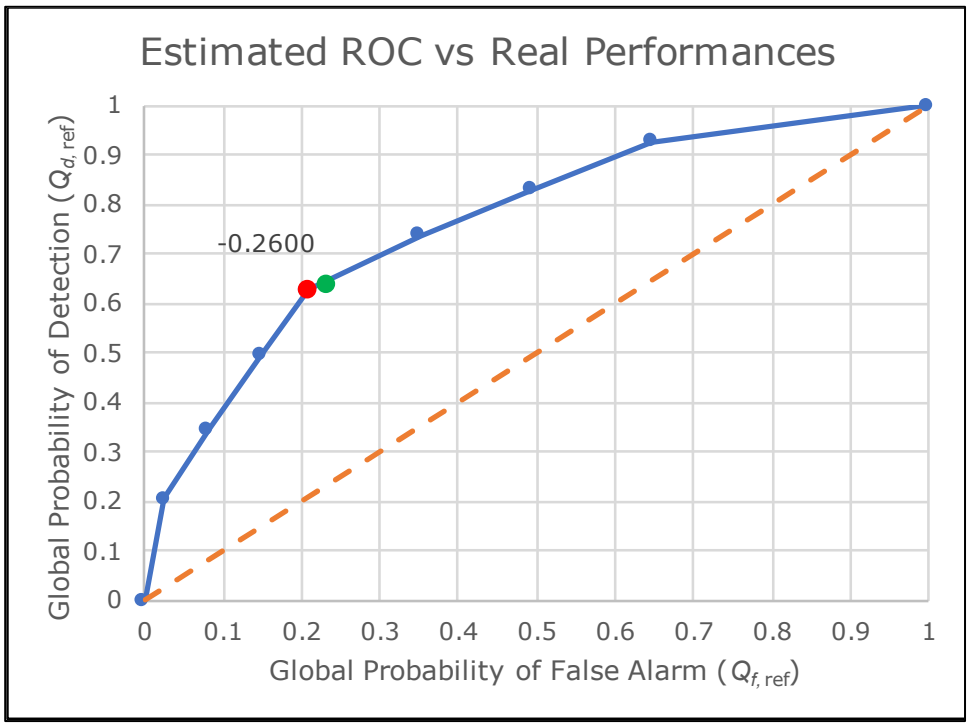


Figure 6.36 ROC curve of performances estimated during tuning of FC in case of WFR. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance when Hotspot 17 is the leakage source.

Now that the imbalance between the two types of event is known, also precision can be evaluated. Table 6.26 shows two different values of F_1 and $F_{0.5}$. The estimated value is in case FC performances were as estimated during the tuning phase; the real value is the one obtained from the simulation. Moreover, Figure 6.37 shows the entire PR-curve for all the thresholds in case FC performances were those estimated during the tuning phase.

	Global Threshold	Precision	Recall	F_1	$F_{0.5}$
Estimated	-0.2600	0.5931	0.6269	0.6095	0.5995
Real		0.5722	0.6478	0.6032	0.5842

Table 6.26 Comparison between estimated and real FC values of precision and recall in case Hotspot 17 is the leakage source and FC performs WFR.

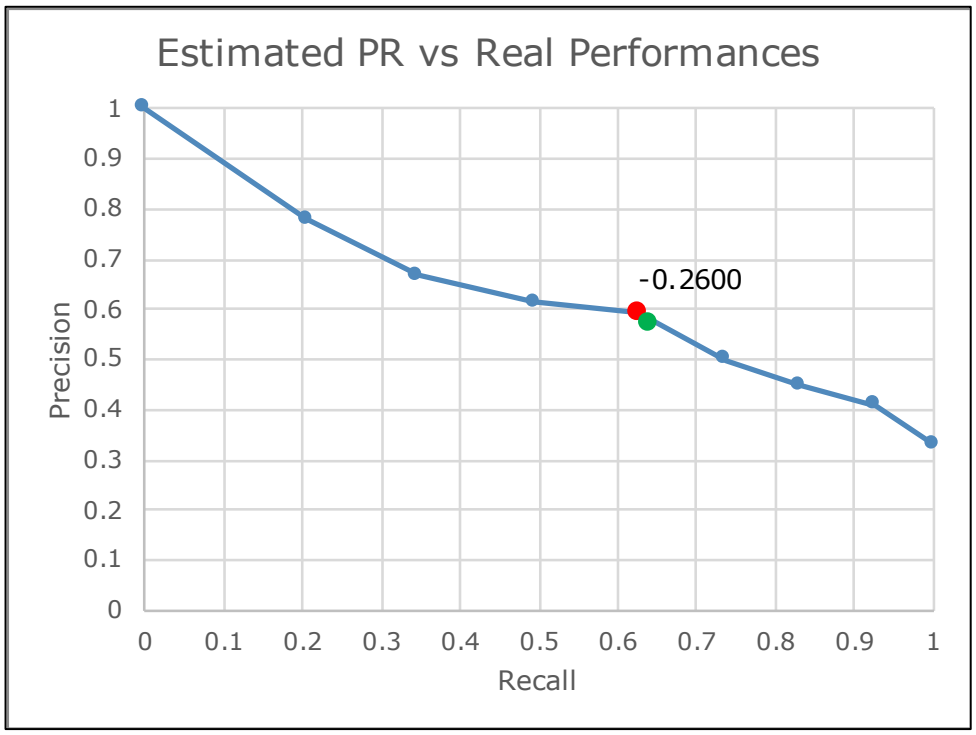


Figure 6.37 PR curve in case Hotspot 17 is the leakage source and FC is performing WFR with the performances estimated during FC tuning. Red dot is the expected performance at the selected optimal threshold. Green dot is the real performance.

6.4 Leak Localization Results

Leak localization performances will be evaluated on the basis of different parameters; one of these is time. The reason why time is important is that most of the localization techniques were able to identify the same hotspots as possible leakage sources during the different tests but in a different amount of time. The time needed to reach a stable result will be then a key performance indicator together with the distance between the identified hotspot and the actual leak source.

A necessary remark must be made: localization performances are influenced by the specific order in which events happen; this does not happen when evaluating detection performances. The reason is that when calculating TPR, FPR and precision, time is not a variable; moreover the FRs/tests adopted here do not change parameters or thresholds dynamically on the basis of new evidence. Two independent tests leading to the same values of true positive rate, false positive rate, and precision will likely lead to two different values of time needed for localization.

This characteristic of the localization techniques had to be taken into account in order not to discriminate a technique with respect to another only because one single test was carried out. For each hotspot (HS 3, HS 10 and HS 17), any technique (CR+CBM and WFR+BBM) is evaluated by applying 20 tests to make sure the abovementioned problem can be overcome. Any test is based on a BMC having $p = 1/3$ and $r = 2$. The BMC will dictate the release pattern, which will be different every time because of the probabilistic nature of the BMC.

The localization procedure takes place those instants in which the FC declare a positive event; this makes the localization procedure "inactive" all the other instants. For this reason, the time necessary for the localization will only take into account those instants

where the FC declared the event as positive, and the localization procedure was ongoing; this aims to better evaluate localization performances without having them unnecessarily too much affected by other external variables.

Hotspot	Technique	Target Localized	Final Distance (*m*)	Positive Detections Needed for Stabilization	
				Mean	Corrected Sample Standard Deviation
(3) Connection With Gas Lift Line	CR+CBM-TISP	No	7.56	21	17
	CR+CBM-TOSP			1648	1885
	WFR+BBM-TISP			40	51
	WFR+BBM-TOSP			20	23
(10) Branch Valve	CR+CBM-TISP	Yes	0	61	60
	CR+CBM-TOSP			622	615
	WFR+BBM-TISP			8	7
	WFR+BBM-TOSP			97	106
(17) Header Isolation Valve	CR+CBM-TISP	No	3.83	22	14
	CR+CBM-TOSP			1358	1998
	WFR+BBM-TISP			4	5
	WFR+BBM-TOSP			14	10

Table 6.27 Evaluation of localization performances of different techniques after 20 tests

7 Discussion

Table 7.1 reports a summary of the results obtained at sensor level during the simulation.

			γ	P_d	P_f	Local Index
FC performing CR	Sen 1	HS 3	1.8336	0.6349	0.1757	J=0.4952
		HS 10		0.4786		J=1.8336
		HS 17		0.2927		J=0.1170
	Sen 2	HS 3	1.9239	0.2526	0.1654	J=0.0872
		HS 10		0.3315		J=0.1661
		HS 17		0.7621		J=0.5967
	Sen 3	HS 3	1.7378	0.3444	0.1874	J=0.1570
		HS 10		0.5499		J=0.3625
		HS 17		0.4198		J=0.2324
FC performing WFR	Sen 1	HS 3	1.0346	0.7213	0.3091	CZ=0.4983
		HS 10		0.5946		CZ=0.4108
		HS 17		0.4293		CZ=0.2966
	Sen 2	HS 3	1.0921	0.3887	0.2960	CZ=0.2736
		HS 10		0.4644		CZ=0.3269
		HS 17		0.8196		CZ=0.5770
	Sen 3	HS 3	1.0592	0.4604	0.3034	CZ=0.3207
		HS 10		0.6407		CZ=0.4463
		HS 17		0.5288		CZ=0.3684

Table 7.1 Summary of the results obtained at sensor level after the tests

As seen in the chapter dedicated to the FC tuning, the procedure starts by selecting an index for FC performances evaluation (it can be ROC-based or PR-based). This would mean sensors have to be tuned looking for the best combination of local thresholds. This best combination should be the one generating a FC level- ROC curve (or PR curve) that, varying the global threshold, can maximize the previously decided index.

In this simulation, the local thresholds are obtained maximizing J if FC performs the CR, and maximizing CZ if FC performs the WFR. The ideal case would have been not to use indexes for the selection of the local thresholds, but to test any possible combination of local threshold among the different sensors as previously said. Nonetheless, this would have required excessive computational power. The values reported in Table 7.1 show the real performances of the sensors with the selected local thresholds which, according to what said before, should be able to maximize J at FC level (assuming that FC works with an appropriate global threshold). Of course, this may not be true as the selection of all the thresholds (local and global) was made during the tuning phase when some assumptions had to be carried out.

J has been selected as the index for global threshold selection as it is the only index (among the most popular that could fit this simulation) able to ensure specific conditions:

- It has a meaning even if examined outside of the ROC space. Unlike d^2 (which is a square distance in the ROC space), and CZ (which is the measure of an area in the ROC space), J represents either the length of a segment in the ROC space (see

Figure 4.10), or the difference, in terms of probability of detection, between the examined classifier and a random classifier having both the same probabilities of false alarm.

- Two points of a ROC curve yielding the same values of J both represent points on the same half of the ROC space (where the chance line is dividing the space). This may not be true with d^2 and CZ in which two equal values could be located on the opposite side of the ROC space with respect to the chance line.

As two different indexes were used at sensor level (according to the FR to be used), it is clear from Table 7.1 how this produced a visible difference among the values of local thresholds. In case J is used (for FC performing the CR) the interval goes from 1.74 to 1.92, while in case CZ is used (for FC performing the WFR) the interval goes from 1.03 to 1.09. Being the centroid of the hotspots close to the centroid of the sensors, the difference of local thresholds among the different sensors is not too vast (both the abovementioned ranges of values are pretty narrow).

About local performances, all the registered values of local probability of false alarm were lower or equal to 0.3, local probabilities of detection varied a lot according to the distance between sensor and target. HS 3 has Sensor 1 as its closest sensor: P_ds of Sensor 1, in case HS 3 was the target, were always greater or equal to 0.63, while they were always lower or equal to 0.46 for the other two sensors. Same for HS 17 as target and Sensors 2: its P_ds were always greater or equal to 0.76, while they were always lower or equal to 0.53 for other sensors. HS 10 is close to the centroid of the sensors. For this reason, all the values of P_d ranged from 0.33 to 0.64 for all sensors (higher values can be found for Sensor 3 being closer to HS 10 than the other two sensors).

Table 7.2, instead, reports a summary of the results obtained carrying out the simulations. Expected values are those obtained during the FC tuning (except for expected values of precision which were available only after the release pattern was known) and can be considered as average values among all HSs as both the tuning procedure and the test are based on the same starting hypothesis (and since failure probability of the different HSs are assumed equal).

		$\overline{\gamma}$	Q_d	Q_f	Precision	J	F_1	$F_{0.5}$
CR	*Estim.*	1	*0.8480*	*0.4431*	*0.4844*	*0.4049*	*0.6166*	*0.5298*
	HS 3		0.8241	0.4295	0.5084	0.3946	0.6289	0.5506
	HS 10		0.8375	0.4557	0.4373	0.3818	0.6051	0.5188
	HS 17		0.9083	0.4332	0.5097	0.4751	0.6530	0.5587
WFR	*Estim.*	-0.2600	*0.6269*	*0.2088*	*0.5931*	*0.4181*	*0.6095*	*0.5995*
	HS 3		0.5453	0.2251	0.5390	0.3202	0.5421	0.5403
	HS 10		0.6005	0.2234	0.5682	0.3771	0.5839	0.5744
	HS 17		0.6378	0.2343	0.5722	0.3035	0.6032	0.5842

Table 7.2 Summary of the results obtained at FC level (detection)

From Table 7.2, it is possible to notice that slightly higher values of J are expected to be obtained when the WFR is used; this seems not to match the indexes obtained with the tested HSs, where indexes are higher in case the CR is used. This difference between the average indexes and the indexes obtained during the tests shows there is not an absolute superiority of the WFR with respect to the CR in this case study. Looking at the performances, it can be seen how those obtained with the CR are very different with respect to those obtained with the WFR: this is due to the different indexes used for sensor tuning (J for the CR, CZ for the WFR). Generally speaking, when WFR was used,

lower values of global probabilities of detection and global probability of false alarm were present.

Precision faced better results when WFR was applied, nevertheless F_1 is lower. This decrease is due to the lower values of Q_d with respect to the CR. It is interesting to notice that the chosen values of global thresholds (either for the CR or the WFR) are those that would have also led to the highest estimated values of F_1 (Table 6.1 and Table 6.2). A way to improve precision could be by giving higher priority to precision with respect to recall by using $\beta < 1$ (for example $F_{0.5}$) and to take into account this index during the selection of threshold (if the release pattern is known). As it can be seen, $F_{0.5}$ gives better results when the WFR is applied as recall becomes less important than precision, even though it is important to remind that sensors and FC were tuned to optimize neither F_1 nor $F_{0.5}$, but J.

From Table 6.27, a summary of the results of localization is provided.

Simulations concerning HS 10 were always successful from a localization point of view using any method, this because of its central position. None of the other methods could correctly localize the other leakage sources.

All methods gave the same results in terms of the final distance between the estimated position and the target, so what can be evaluated is the number of positive FC detection needed for the stabilization. Stabilization occurs since all these methods base their estimations at a precise instant, also taking into account the estimations obtained the instants before.

The worst result is given by CR+CBM-TOSP method, which was not able to give stable results in reasonable mean times in any tested HS. Moreover, using this method, standard deviations for the different HSs were the highest, making this method unreliable.

It is worth to mention how in HS 10 and HS 17, which were both inside the sensors' perimeter, WFR+BBM-TISP method gave the best results from either a stabilization time point of view or a standard deviation point of view (meaning that test has a good reproducibility).

HS 3, instead, is outside the sensors' perimeter; in this case, WFR+BBM-TOSP method gave the best result. CR+CBM-TISP gave good results in all cases.

As explained, every localization method used here averages the position calculated at a certain instant with all the previously calculated positions up to that moment in order to obtain a more reliable final position to declare. This, together with the fact that attempts of localization are only made when FC declares the event as positive, makes it vital to have a good value of precision when it comes to detection performances. In fact, if the FC declares a false alarm, an estimated leakage position will be calculated and, not only this position will be wrong, but it will affect next estimated positions.

From what observed from these results, the WFR should be preferred as on an average it should give better values of J (as seen during FC tuning). Nonetheless, performances obtained using the WFR and using the CR may overlap according to the specific target position (this because the centroid of the sensors is close to the centroid of the HSs), and in some cases (as shown by the tests) they may be lower if using the WFR. The WFR brings an advantage also from a management point of view: new sensors can be

installed in any position by just registering their values of estimated performance so that FC can consider their local decisions with the appropriate weights.

About the leakage source localization, it is suggested to use either WFR+BBM-TISP method or WFR+BBM-TOSP method as they both showed good performances (TISP method when the target was inside sensor's perimeter, TOSP method when it was outside sensors' perimeter).

A different sensor configuration is suggested after the analysis of this case study when it comes to localization performances. The reason is that HSs which are far from the centroid of the sensors, in case of leakage, cannot be localized accurately, especially those outside the sensors' perimeter. Adding one sensor in order to form a rectangular perimeter able to include all the HSs may be a good idea to improve localization when using the TISP method. HSs can also be outside the perimeter; this would improve performances of detection since average distances between sensors and hot spots would reduce, but in such a scenario the TOSP method should be used for localization. In case three sensors are used, it is suggested to modify the geometry according to the desired final performances; this would mean to encompass the positioning of the sensors as part of the tuning procedure.

8 Conclusion and Further Works

The idea of this work was to study, analyze, compare, and finally propose techniques for the detection and localization of oil spills from subsea templates using the Goliat FPSO as a case study. In this work, only one kind of technology was taken into consideration; this means there was not the claim to assess whether or not the actual overall leak detection system can be considered reliable since, in order to affirm this, also other LDSs used should have been considered. The aim was to focus on the performances of passive acoustic sensors using a statistical signal processing approach for the study of the detection performances. Localization performances were also studied, even though such sensors were not installed to locate leakages. Detection using sensor networks is an existing field; for this reason, enough theoretical literature was present to support out the study. Less literature was present when detection is applied to subsea oil spill detection systems. This kind of study can be considered interdisciplinary and needs the know-how from either a statistical signal processing or safety & reliability and risk management point of view. The present study showed how the detection techniques already existing from other fields can be adapted, improved, and then implemented in a subsea leak detection system, and how these detection techniques can be furtherly improved with appropriate sensor tuning and FC tuning. Localization by using distributed sensor networks is a relatively new field, which means that localization techniques are subjected to continuous improvements. At the time of this work, the literature concerning the localization of subsea oil spills using distributed sensor networks is extremely poor and what present was not able to fit this case study since based on different premises and different technologies; for this reason, some new methods for localizations have been developed. These methods were developed in a way that they could be coherent with the proposed detection methods and showed good performances as long as the geometrical disposition of the sensors is adequate for this task. In this work it was shown how the current number of sensors and their geometrical disposition is not optimal for an efficient oil spill localization.

The simulations showed a system which needs to be supported by other leak detection systems (capacitive sensors and internal leak detection systems in this case). This system alone, in fact, can satisfyingly detect oil spills, but the current geometrical configuration does not allow an accurate localization in all areas of the template.

A successful localization is possible only when leakage is close to the centroid of the sensors; in all the other cases, the FC estimated some positions which were close to the leakage point but did not match it.

This work can be furtherly developed. These are some suggestions regarding topics that can be developed and added to complete the present work:

- Integrate the proposed tuning techniques with an algorithm for an optimal sensor positioning;
- Model the AAFs in a more accurate way accounting for the real geometry of the template and accounting for other phenomena causing the transmission loss;
- Model the entire LDS so that the fusion center can take advantage of different technologies;

- Assign weights to the different hotspots:
  All the hotspots were considered to have the same probability of failure. A reliability study is suggested to obtain more performing local and global thresholds during the tuning phases and to have more reliable reference performances when carrying out the WFR, and the BBM;
- Any hotspot corresponds to some equipment having precise technical and process specifications. Leakages should be different from an acoustic point of view when considering different hotspots. A study in order to have different values of SNR for different hotspots should be carried out;
- A study aimed at obtaining more realistic amplitude pdfs associated with the underwater oil spill, and underwater noise (not only thermal noise) should be carried out; this may be carried out together with a study on the optimal sensed frequency able to maximize the SNR;
- A study on the behavior of oil spills over time to be able to shape the release pattern should be carried out in a way that real scenario could be better simulated; this would allow better use of parameters like precision during the tuning phase. If this is done, a study on which value of β to use when calculating the F-score should be performed;
- A study on which index for the selection of the thresholds for this application should be made (a reliability study on the entire system may be useful);
- A study focused on the CBA as a driver for configuration selection;
- Through the fusion center, being able to update reference local performances (used when carrying out the WFR and the BBM), global threshold, and local thresholds based on the current estimated position. The sensor and FC tuning were performed using mean values corresponding to different hotspots. A dynamic approach would change the previously listed parameters as new evidence is obtained. In order to avoid error propagation, this updating procedures may be carried out only after a certain number of events are collected.

This work aims to give a contribution to the Oil&Gas industry by introducing new methodologies for subsea oil spill risk management, which is critical nowadays, especially in a sensitive area like the Barents Sea. These methodologies have been successfully adapted from different fields (often wholly extraneous to the O&G industry) or specifically developed for this work creating a starting point for further studies.

References

A. Rasmussen, J., J. Piette, M., 1984. Comparison of the costs and the results in the onshore and offshore search for oil and gas. Energy J. 5, 159–164. https://doi.org/10.5547/ISSN0195-6574-EJ-Vol5-No1-11

Ainslie, M.A., McColm, J.G., 1998. A simplified formula for viscous and chemical absorption in sea water. J. Acoust. Soc. Am. 103, 1671–1672. https://doi.org/10.1121/1.421258

Al-Dharrab, S., Uysal, M., Duman, T.M., 2013. Cooperative underwater acoustic communications [Accepted From Open Call]. IEEE Commun. Mag. 51, 146–153. https://doi.org/10.1109/MCOM.2013.6553691

Aludaat, K.M., 2018. On the beta cumulative distribution function. Appl. Math. Sci. 12, 461–466. https://doi.org/10.12988/ams.2018.8241

Austine, K., 2017. Oil and gas pipeline leak detection industry - A growing business and what this mean to surveyors.

Aven, T., Krohn, B.S., 2014. A new perspective on how to understand, assess and manage risk and the unforeseen. Reliab. Eng. Syst. Saf. 121, 1–10. https://doi.org/10.1016/j.ress.2013.07.005

Bai, Y., Bai, Q., 2010a. Subsea Manifolds, in: Subsea Engineering Handbook. Elsevier, pp. 571–632. https://doi.org/10.1016/B978-1-85617-689-7.10019-6

Bai, Y., Bai, Q., 2010b. Subsea Wellheads and Trees, in: Subsea Engineering Handbook. Elsevier, pp. 703–761. https://doi.org/10.1016/B978-1-85617-689-7.10022-6

Bai, Y., Bai, Q., 2010c. Subsea Control, in: Subsea Engineering Handbook. Elsevier, pp. 193–224. https://doi.org/10.1016/B978-1-85617-689-7.10007-X

Bai, Y., Bai, Q., 2010d. Subsea Umbilical Systems, in: Subsea Engineering Handbook. Elsevier, pp. 797–825. https://doi.org/10.1016/B978-1-85617-689-7.10024-X

Bai, Y., Bai, Q., 2010e. Subsea Production Risers, in: Subsea Engineering Handbook. Elsevier, pp. 853–890. https://doi.org/10.1016/B978-1-85617-689-7.10026-3

Benestad, 2019. Hydrocarbon Leak Detector [WWW Document]. Benestad. URL http://www.phaze.no/hydrocarbon-leak-detector (accessed 6.12.19).

Beyer, J., Trannum, H.C., Bakke, T., Hodson, P.V., Collier, T.K., 2016. Environmental effects of the Deepwater Horizon oil spill: A review. Mar. Pollut. Bull. 110, 28–51. https://doi.org/10.1016/j.marpolbul.2016.06.027

Bjørnbom, E., 2017. Goliat lekkasjedeteksjon - utforming og erfaringer med fokus på havbunnsdeteksjon. Presented at the Ptil og Miljødirektoratet.

Bjørnbom, E., 2011. Goliat – Leak detection and monitoring from template to satellite. Presented at the OLF/PSA Subsea leak Detection Seminar.

Bjørnbom, E., Hansen, O., Melhus, M.L., Klevstad, U., 2016. Environmental Solutions

Implemented for the Goliat Offshore Oil Field Development - The First Oil Field Development in the Norwegian Barents Sea, in: SPE International Conference and Exhibition on Health, Safety, Security, Environment, and Social Responsibility. Presented at the SPE International Conference and Exhibition on Health, Safety, Security, Environment, and Social Responsibility, Society of Petroleum Engineers, Stavanger, Norway. https://doi.org/10.2118/179434-MS

Boyd, K., Eng, K.H., Page, C.D., 2013. Area under the Precision-Recall Curve: Point Estimates and Confidence Intervals, in: Salinesi, C., Norrie, M.C., Pastor, Ó. (Eds.), Advanced Information Systems Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 451–466. https://doi.org/10.1007/978-3-642-40994-3_29

Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit. 30, 1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2

Brennan, C.W., 2017. Basic Acoustic Theory.

Bucelli, M., Utne, I., Salvo Rossi, P., Paltrinieri, N., Cozzani, V., 2018. A preliminary approach to subsea risk management using sensor network information, in: Safety and Reliability – Safe Societies in a Changing World. Presented at the 28th International European Safety and Reliability Conference (ESREL 2018), Trondheim. https://doi.org/10.1201/9781351174664

Candy, J.V., Breitfeller, E.F., 2013. Receiver Operating Characteristic (ROC) Curves: An Analysis Tool for Detection Performance (No. LLNL-TR-642693, 1093414). https://doi.org/10.2172/1093414

Carlos-Mancilla, M., López-Mellado, E., Siller, M., 2016. Wireless Sensor Networks Formation: Approaches and Techniques. J. Sens. 2016, 1–18. https://doi.org/10.1155/2016/2081902

Chair, Z., Varshney, P.K., 1986. Optimal Data Fusion in Multiple Sensor Detection Systems. IEEE Trans. Aerosp. Electron. Syst. AES-22, 98–101. https://doi.org/10.1109/TAES.1986.310699

Chen, C., Millero, F.J., 1977. Speed of sound in seawater at high pressures. J. Acoust. Soc. Am. 62, 1129–1135. https://doi.org/10.1121/1.381646

Chen, W., Holm, S., 2003. Modified Szabo's wave equation models for lossy media obeying frequency power law. J. Acoust. Soc. Am. 114, 2570. https://doi.org/10.1121/1.1621392

Ciuonzo, D., Romano, G., Salvo Rossi, P., 2013. Performance Analysis and Design of Maximum Ratio Combining in Channel-Aware MIMO Decision Fusion. IEEE Trans. Wirel. Commun. 12, 4716–4728. https://doi.org/10.1109/TWC.2013.071913.130269

Ciuonzo, D., Salvo Rossi, P., 2017. Distributed detection of a non-cooperative target via generalized locally-optimum approaches. Inf. Fusion 36, 261–274. https://doi.org/10.1016/j.inffus.2016.12.006

Coppens, A.B., 1981. Simple equations for the speed of sound in Neptunian waters. J. Acoust. Soc. Am. 69, 862–863. https://doi.org/10.1121/1.385486

Davis, J., Goadrich, M., 2006. The relationship between Precision-Recall and ROC curves,

in: Proceedings of the 23rd International Conference on Machine Learning - ICML '06. Presented at the 23rd international conference, ACM Press, Pittsburgh, Pennsylvania, pp. 233–240. https://doi.org/10.1145/1143844.1143874

Del Grosso, V.A., 1974. New equation for the speed of sound in natural waters (with comparisons to other equations). J. Acoust. Soc. Am. 56, 1084–1091. https://doi.org/10.1121/1.1903388

DNV-GL, 2016. RP-F302 Offshore leak detection.

DNV-GL, 2014. Joint Industry Project to enhance the offshore leak detection approach [WWW Document]. DNV GL. URL https://www.dnvgl.com/news/joint-industry-project-to-enhance-the-offshore-leak-detection-approach-8096 (accessed 6.5.19).

DNV-GL, 2010. DNV-RP-F302: Selection and use of Subsea Leak Detection Systems.

Domingo, M.C., 2008. Overview of channel models for underwater wireless communication networks. Phys. Commun. 1, 163–182. https://doi.org/10.1016/j.phycom.2008.09.001

Eni, 2015. Goliat Field Trip. Hammerfest.

Eni Norge, 2015. Goliat crude oil blend.

Etter, P.C., 2018. Underwater Acoustic Modeling and Simulation, 5 edition. ed. CRC Press, Boca Raton.

European Union, 2010. Directive 2010/75/EU of the European Parliament and of the Council of 24 November 2010 on industrial emissions (integrated pollution prevention and control). Off. J. Eur. Union.

Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognit. Lett. 27, 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Fisher, F.H., Simmons, V.P., 1977. Sound absorption in sea water. J. Acoust. Soc. Am. 62, 558–564. https://doi.org/10.1121/1.381574

Fofonoff, N.P., Millard Jr., R.C., 1983. Algorithms for computation of fundamental properties of seawater. UNESCO Tech. Pap. Mar. Sci. 44.

Francois, R.E., Garrison, G.R., 1982a. Sound absorption based on ocean measurements: Part I: Pure water and magnesium sulfate contributions. J. Acoust. Soc. Am. 72, 896–907. https://doi.org/10.1121/1.388170

Francois, R.E., Garrison, G.R., 1982b. Sound absorption based on ocean measurements. Part II: Boric acid contribution and equation for total absorption. J. Acoust. Soc. Am. 72, 1879–1890. https://doi.org/10.1121/1.388673

Frye, H.W., Pugh, J.D., 1971. A New Equation for the Speed of Sound in Seawater. J. Acoust. Soc. Am. 50, 384–386. https://doi.org/10.1121/1.1912645

Geiger, G., 2006. State-of-the-Art in Leak Detection and Localisation. Presented at the Pipeline Technology Conference.

Gil, P., Alacid, B., 2018. Oil Spill Detection in Terma-Side-Looking Airborne Radar Images Using Image Features and Region Segmentation. Sensors 18, 151. https://doi.org/10.3390/s18010151

Girard, F., Fisher, C.R., 2018. Long-term impact of the Deepwater Horizon oil spill on deep-sea corals detected after seven years of monitoring. Biol. Conserv. 225, 117–127. https://doi.org/10.1016/j.biocon.2018.06.028

Gyo Lee, Y., Garza-Gomez, X., Lee, R.M., 2018. Ultimate Costs of the Disaster: Seven Years After the *Deepwater Horizon* Oil Spill. J. Corp. Account. Finance 29, 69–79. https://doi.org/10.1002/jcaf.22306

Hamilton, S., Charalambous, B., 2013. Leak Detection, Utility and Infrastructure Management Bundle. IWA Publishing.

Hand, D.J., 2001. A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. Mach. Learn. 45, 171–186.

Hasselström, L., Cole, S., Håkansson, C., Khaleeva, Y., Noring, M., Soutukorva, Å., 2012. The value of Ecosystem services at risk from oil spills in the Barents Sea. Presented at the 12th Biennial Conference of the International Society for Ecological Economics, Rio de Janeiro, Brazil.

Health and Safety Executive, 2019. ALARP - As low as reasonably practicable [WWW Document]. Health Saf. Exec. URL http://www.hse.gov.uk/comah/alarp.htm (accessed 6.12.19).

Hester, M.W., Willis, J.M., Rouhani, S., Steinhoff, M.A., Baker, M.C., 2016. Impacts of the Deepwater Horizon oil spill on the salt marsh vegetation of Louisiana. Environ. Pollut. 216, 361–370. https://doi.org/10.1016/j.envpol.2016.05.065

Hong, M., Yanjie, Y., 2009. Studies on Marine Oil Spills and Their Ecological Damage 5. https://doi.org/10.1007/s11802-009-00312-5

Hosmer, D.W., Lemeshow, S., 2004. Applied Logistic Regression, Applied Logistic Regression. Wiley.

Ivanov, A.Yu., Zatyagalova, V.V., 2008. A GIS approach to mapping oil spills in a marine environment. Int. J. Remote Sens. 29, 6297–6313. https://doi.org/10.1080/01431160802175587

Jawarkar, U., Panchore, P., Deshmukh, S., 2013. Overview of wireless sensor network and its applications. Int. J. Electron. Commun. Comput. Eng. 4, 29–32.

Jing, Z., Pan, H., Li, Y., Dong, P., 2018. Non-Cooperative Target Tracking, Fusion and Control: Algorithms and Advances, Information Fusion and Data Science. Springer International Publishing.

Joshy, S., 2010. Capacity of Underwater Wireless Communication Channel With Different Acoustic Propagation Loss Models. Int. J. Comput. Netw. Commun. 2, 192–204. https://doi.org/10.5121/ijcnc.2010.2512

Jurafsky, D., Martin, J.H., 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall Series in Artifi. Pearson Prentice Hall.

Kaplan, S., Garrick, B.J., 1981. On The Quantitative Definition of Risk. Risk Anal. 1, 11–27. https://doi.org/10.1111/j.1539-6924.1981.tb01350.x

Kay, S.M., 1998. Fundamentals of Statistical Signal Processing: Detection Theory.

Pearson College Div, Englewood Cliffs, N.J.

Kongsberg, 2016. Increased use of satellite monitoring on the Norwegian Shelf - KSAT - Kongsberg Satellite Services [WWW Document]. Kongsberg. URL https://www.kongsberg.com/ksat/news-and-media/news-archive/2016/increased-use-of-satellite-monitoring-on-the-norwegian-shelf/ (accessed 6.10.19).

Konstantinidou, M., Christou, M., European Commission, Joint Research Centre, Institute for Energy and Transport, 2012. Safety of offshore oil and gas operations lessons from past accident analysis: ensuring EU hydrocarbon supply through better control of major hazards. Publications Office, Luxembourg.

Kosta, U., 2002. Statistical Evaluation of the Underwater Detection.

Kularia, Y., Kohli, S., Bhattacharya, P.P., 2016. Analysis of Acoustic Channel Characteristics for Underwater Wireless Sensors Networks. Int. J. Comput. Sci. Inf. Technol. Control Eng. IJCSITCE 3. https://doi.org/10.5121/ijcsitce.2016.3202

Kushner, H.J., Dupuis, P., 2001. The Markov Chain Approximation Method: Introduction, in: Kushner, H.J., Dupuis, P. (Eds.), Numerical Methods for Stochastic Control Problems in Continuous Time, Stochastic Modelling and Applied Probability. Springer New York, New York, NY, pp. 67–88. https://doi.org/10.1007/978-1-4613-0007-6_5

Leroy, C.C., 1969. Development of Simple Equations for Accurate and More Realistic Calculation of the Speed of Sound in Seawater. J. Acoust. Soc. Am. 46, 216–226. https://doi.org/10.1121/1.1911673

Leroy, C.C., Robinson, S.P., Goldsmith, M.J., 2008. A new equation for the accurate calculation of sound speed in all oceans. J. Acoust. Soc. Am. 124, 2774–2782. https://doi.org/10.1121/1.2988296

Ling, Q., Tian, Z., Yin, Y., Li, Y., 2009. Localized Structural Health Monitoring Using Energy-Efficient Wireless Sensor Networks. IEEE Sens. J. 9, 1596–1604. https://doi.org/10.1109/JSEN.2009.2019318

Liu, X., 2012. Classification accuracy and cut point selection. Stat. Med. 31, 2676–2686. https://doi.org/10.1002/sim.4509

Lobo, J.M., Jiménez-Valverde, A., Real, R., 2008. AUC: a misleading measure of the performance of predictive distribution models. Glob. Ecol. Biogeogr. 17, 145–151. https://doi.org/10.1111/j.1466-8238.2007.00358.x

Loeng, H., Drinkwater, K., 2007. An overview of the ecosystems of the Barents and Norwegian Seas and their response to climate variability. Deep Sea Res. Part II Top. Stud. Oceanogr. 54, 2478–2500. https://doi.org/10.1016/j.dsr2.2007.08.013

Lovett, J.R., 1978. Merged seawater sound-speed equations. J. Acoust. Soc. Am. 63, 1713–1718. https://doi.org/10.1121/1.381909

Lurton, X., 2010. Underwater acoustic wave propagation, in: An Introduction to Underwater Acoustics: Principles and Applications. Springer Berlin Heidelberg.

Ma, H., Bandos, A.I., Rockette, H.E., Gur, D., 2013. On use of partial area under the ROC curve for evaluation of diagnostic performance. Stat. Med. 32, 3449–3458. https://doi.org/10.1002/sim.5777

Mackenzie, K.V., 1981. Nine-term equation for sound speed in the oceans. J. Acoust. Soc. Am. 70, 807–812. https://doi.org/10.1121/1.386920

Mai, C., Pedersen, S., Hansen, L., Jepsen, K.L., Zhenyu Yang, 2016. Subsea infrastructure inspection: A review study, in: 2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS). Presented at the 2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS), pp. 71–76. https://doi.org/10.1109/USYS.2016.7893928

Mareano [WWW Document], 2019. . Mareano. URL http://www.mareano.no/kart/mareano.html#maps/4050 (accessed 6.6.19).

Medwin, H., 1975. Speed of sound in water: A simple equation for realistic parameters. J. Acoust. Soc. Am. 58, 1318–1319. https://doi.org/10.1121/1.380790

Millero, F.J., Li, X., 1994. Comments on '"On equations for the speed of sound in seawater"' [J. Acoust. Soc. Am. **93** , 255–275 (1993)]. J. Acoust. Soc. Am. 95, 2757–2759. https://doi.org/10.1121/1.409844

Milne, R., 2017. Oil exploration in Norwegian Arctic faces sea of opposition. Financ. Times.

Ministry of Climate and Environment, 1981. Pollution Control Act - Act of 13 March 1981 No.6 Concerning Protection Against Pollution and Concerning Waste (Lov). Ministry of Climate and Environment.

Naxys, 2019. Single Acoustic Leak Detector (SALD) For hotspot & limited area.

Naxys, 2011. Naxys Acoustic Leak Detector - Qualification Status and Field Experience.

Neyman, J., Pearson, E.S., 1933. On the Problem of the Most Efficient Tests of Statistical Hypotheses. Philos. Trans. R. Soc. Lond. Ser. Contain. Pap. Math. Phys. Character 231, 289–337. https://doi.org/10.1098/rsta.1933.0009

Nordbø, H.T., 2010. Goliat Produced Water Management. Stavanger.

Norwegian Coastal Administration, 2011. Automatic Identification System (AIS) [WWW Document]. Nor. Coast. Adm. URL https://www.kystverket.no/en/EN_Maritime-Services/Reporting-and-Information-Services/Automatic-Identification-System-AIS/ (accessed 6.10.19).

Norwegian Environment Agency, 2016. Environmental monitoring of petroleum activities on the Norwegian continental shelf. Oslo.

Norwegian Petroleum Directorate, 2019a. Activity per sea area [WWW Document]. norwegianpetroleum.no. URL https://www.norskpetroleum.no/en/developments-and-operations/activity-per-sea-area/ (accessed 6.4.19).

Norwegian Petroleum Directorate, 2019b. Field - FactPages - Norwegian Petroleum Directorate [WWW Document]. Factpages. URL http://factpages.npd.no/factpages/default.aspx?culture=en&nav1=field&nav2=PageView%7CProducing (accessed 6.6.19).

Norwegian Petroleum Directorate, 2018. Resource Report Exploration 2018. Stavanger.

O'Connor, P., Kleyner, A., 2012. Practical Reliability Engineering, Quality and Reliability Engineering Series. Wiley.

Paltrinieri, N., Khan, F., 2016. Dynamic Risk Analysis in the Chemical and Petroleum Industry: Evolution and Interaction with Parallel Disciplines in the Perspective of Industrial Application. Elsevier Science.

Paltrinieri, N., Khan, F., Amyotte, P., Cozzani, V., 2014. Dynamic approach to risk management: Application to the Hoeganaes metal dust accidents. Process Saf. Environ. Prot. 92, 669–679. https://doi.org/10.1016/j.psep.2013.11.008

Perkins, N.J., Schisterman, E.F., 2006. The Inconsistency of "Optimal" Cutpoints Obtained using Two Criteria based on the Receiver Operating Characteristic Curve. Am. J. Epidemiol. 163, 670–675. https://doi.org/10.1093/aje/kwj063

Petroleum Safety Authority Norway, Norwegian Environment Agency, Norwegian Directorate of Health, Norwegian Food Safety Authority, 2016. Regulations relating to conducting petroleum activities (the activities regulations).

Petroleum Safety Authority Norway, Norwegian Environment Agency, Norwegian Directorate of Health, Norwegian Food Safety Authority, Norwegian Radiation Protection Authority, 2017. Regulations relating to management and the duty to provide information in the petroleum activities and at certain onshore facilities (the management regulations).

Phaze, 2019. Hydrocarbon Leak Detector - Environmental Monitoring.

Richards, D.A., 2012. Oil Spill Detection with Infrared Imaging.

Røsby, E., 2011. Goliat development project Subsea leak detection design. Presented at the OLF/PSA Subsea leak Detection Seminar.

Rota, M., Antolini, L., 2014. Finding the optimal cut-point for Gaussian and Gamma distributed biomarkers. Comput. Stat. Data Anal. 69, 1–14. https://doi.org/10.1016/j.csda.2013.07.015

Ruebeck, J., James, R.G., Mahoney, J.R., Crutchfield, J.P., 2018. Prediction and Generation of Binary Markov Processes: Can a Finite-State Fox Catch a Markov Mouse? Chaos Interdiscip. J. Nonlinear Sci. 28, 013109. https://doi.org/10.1063/1.5003041

Ruixin Niu, Varshney, P.K., 2005. Decision fusion in a wireless sensor network with a random number of sensors, in: Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Presented at the Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., p. iv/861-iv/864 Vol. 4. https://doi.org/10.1109/ICASSP.2005.1416145

Rushton, A.G., Reston, S.D., 2006. CBA, ALARP and industrial safety in the United Kingdom. Saf. Reliab. 26, 24–33. https://doi.org/10.1080/09617353.2006.11690825

Saito, T., Rehmsmeier, M., 2015. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PLoS ONE 10. https://doi.org/10.1371/journal.pone.0118432

Salvo Rossi, P., 2019. Distributed Detection and Localization - A Statistical Signal Processing Approach.

Salvo Rossi, P., Ciuonzo, D., Ekman, T., Hefeng Dong, 2015. Energy Detection for MIMO Decision Fusion in Underwater Sensor Networks. IEEE Sens. J. 15, 1630–1640.

https://doi.org/10.1109/JSEN.2014.2364856

Sasaki, Y., 2007. The truth of the F-measure 5.

Schulkin, M., Marsh, H.W., 1962. Sound Absorption in Sea Water. J. Acoust. Soc. Am. 34, 864–865. https://doi.org/10.1121/1.1918213

Sehgal, A., Tumar, I., Schönwälder, J., 2010. AquaTools - An Underwater Acoustic Networking Simulation Toolkit. Presented at the IEEE Oceans 2010 Asia-Pacific.

Sevan SSP, 2019. Goliat FPSO [WWW Document]. Sevan SSP. URL https://sevanssp.com/goliat/ (accessed 6.6.19).

Shafer, S., Gross, M., Hillegeist, P., 2013. The Economic Benefits of Increasing U.S. Access to Offshore Oil and Natrural Gas Resources in the Atlantic. Quest Offshore Resources Inc.

Speight, J.G., 2014. Handbook of Offshore Oil and Gas Operations. Elsevier Science.

Szabo, T.L., 1994. Time domain wave equations for lossy media obeying a frequency power law. J. Acoust. Soc. Am. 96, 491–500. https://doi.org/10.1121/1.410434

Talib, K.H., Othman, M.Y., Saiful Aman Hj Sulaiman, Wazir, M.A.M., Azizan, A., 2011. Determination of speed of sound using empirical equations and SVP, in: 2011 IEEE 7th International Colloquium on Signal Processing and Its Applications. Presented at the 2011 IEEE 7th International Colloquium on Signal Processing and its Applications, pp. 252–256. https://doi.org/10.1109/CSPA.2011.5759882

Thorp, W.H., 1967. Analytic Description of the Low-Frequency Attenuation Coefficient. J. Acoust. Soc. Am. 42, 270–270. https://doi.org/10.1121/1.1910566

Topouzelis, K., 2008. Oil Spill Detection by SAR Images: Dark Formation Detection, Feature Extraction and Classification Algorithms. Sensors 8, 6642–6659. https://doi.org/10.3390/s8106642

U.S. Energy Information Administration, 2016. Offshore production nearly 30% of global crude oil output in 2015. Today Energy. URL https://www.eia.gov/todayinenergy/detail.php?id=28492 (accessed 6.3.19).

Van Kampen, N.G., 2011. Stochastic Processes in Physics and Chemistry, 3rd ed, North-Holland Personal Library. Elsevier Science.

Vårdal, O., 2010. Goliat - The Concept Solution. Oslo.

Vetrov, A., Romankevich, E., 2004. Carbon Cycle in the Russian Arctic Seas, 2004 edition. ed. Springer, Berlin ; New York.

Wallhead, P.J., Bellerby, R.G.J., Silyakova, A., Slagstad, D., Polukhin, A.A., 2017. Bottom Water Acidification and Warming on the Western Eurasian Arctic Shelves: Dynamical Downscaling Projections: SHELF BOTTOM ACIDIFICATION IN THE ARCTIC. J. Geophys. Res. Oceans 122, 8126–8144. https://doi.org/10.1002/2017JC013231

Wijnen, P., 2017. Survey: Majority choose climate before oil. Nor. Today.

Wilson, W.D., 1960. Speed of Sound in Sea Water as a Function of Temperature, Pressure, and Salinity. J. Acoust. Soc. Am. 32, 641–644. https://doi.org/10.1121/1.1908167

Wong, G.S.K., Zhu, S., 1995. Speed of sound in seawater as a function of salinity, temperature, and pressure. J. Acoust. Soc. Am. 97, 1732–1736. https://doi.org/10.1121/1.413048

Wulff, A., 2016. Goliat – kjempen i Barentshavet.

WWF, 2004. The Barents Sea Ecoregion. A Biodiversity Assessment.

Xiao, Y. (Ed.), 2010. Underwater Acoustic Sensor Networks, 1 edition. ed. Auerbach Publications.

Yan, Y., Wang, H., Shen, X., Zhong, X., 2015. Decision Fusion with Channel Errors in Distributed Decode-Then-Fuse Sensor Networks. Sensors 15, 19157–19180. https://doi.org/10.3390/s150819157

Youden, W.J., 1950. Index for rating diagnostic tests. Cancer 3, 32–35. https://doi.org/10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3

Appendices

Appendix 1: MATLAB Script (Hotspot 10: CR + CBM-TISP)

Appendix 2: MATLAB Script (Hotspot 10: CR + CBM-TOSP)

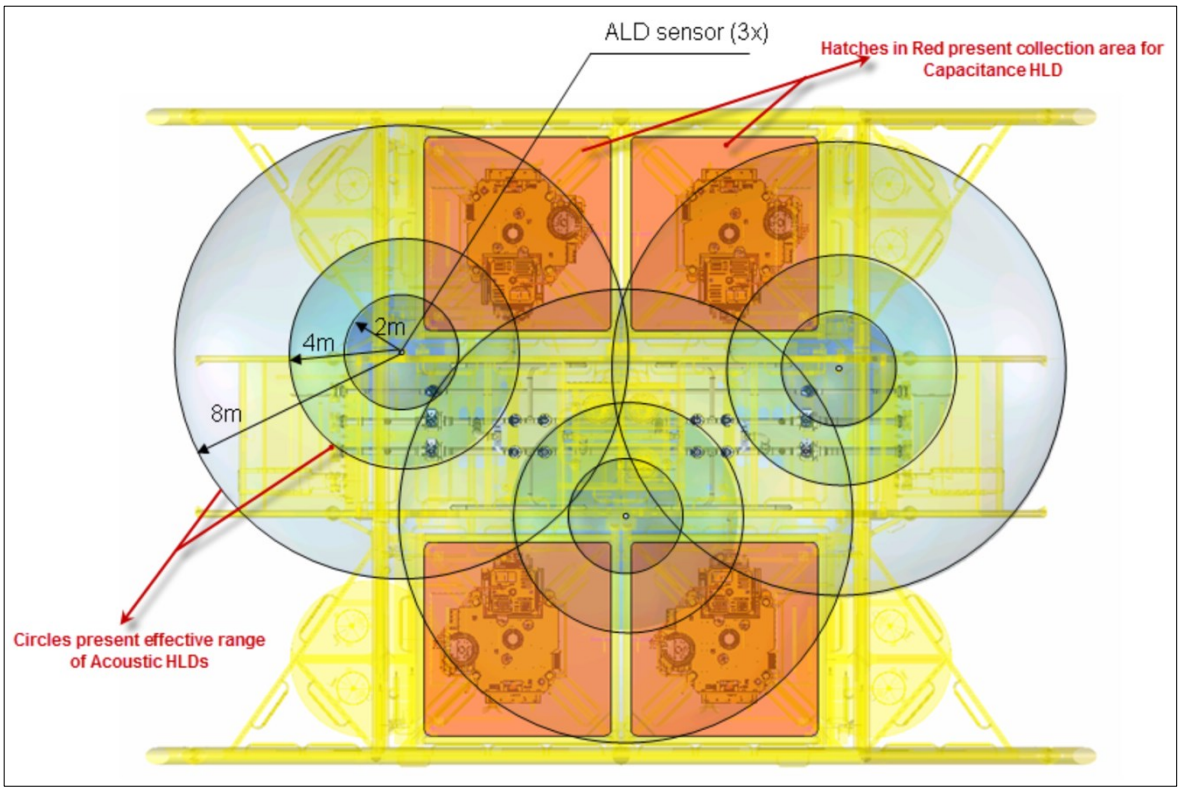
Appendix 3: MATLAB Script (Hotspot 10: WFR + BBM-TISP)

Appendix 4: MATLAB Script (Hotspot 10: WFR + BBM-TOSP)

Appendix 1: MATLAB Script (Hotspot 10: CR + CBM-TISP)

This MATLAB script was written using MATLAB R2019; it simulates the release in HS 10, the FC performs the CR for detection and CBM-TISP for localization. All the settings are those described in the previous chapters.

MATLAB, in order to run the animation, must have the following picture in a folder accessible by MATLAB. The file must be named "template.png".




```
%%%%%  Oil Spill Detection and Localization    %%%%
%%%%%                CR + CBM-TISP             %%%%
%%%%%                 Hotspot 10               %%%%
%%%%%          Author: Gianluca Tabella        %%%%

clear
close all
clc

%% Release Modelling

disp('Release Modelling') % display the progress

p=1/3;  % parameter for release shape
r=2;    % parameter for release shape
N=10000; % number of instants
X=binary_source(p,r,N); % function giving an array that simulates the
events before the continuous release
tot=length(X); % total number of records
rel=nnz(X); % number of releases (positive events)
non_rel=tot-rel; % number of non-releases (negative events)

%% Signal Modelling

disp('Signal Modelling') % display the progress
```

```
f=20e-3; % kHz - sound frequency
T=3.8; % ∞C - temperature (T=3.8∞C at Goliat)
S=35; % ppt - salinity (S=35 ppt at Goliat)
D=350; % m - depth (D=350-400 m at Goliat)
pH=8; % pH of water (pH=8 at Goliat)
spread_cf=1.5; % spreading coeff. (spherical=2;cylindrical=1;practical=1.5)

c=speed_sound(D,T,S); % m/s - speed of sound underwater
alpha_water=absorption_FG(f,T,S,D,pH,c); % dB/km - absorption due to water
hotspot_crd=[11.5,11.75 ; 11.5,12.75 ; 11.5,13.75 ; 15,11.75 ; 15,12.75 ;
15,13.75 ; 18,11.75 ; 18,12.75 ; 19,11.75 ; 19,12.75 ; 24.5,11.75 ;
24.5,12.75 ; 25.5,11.75 ; 25.5,12.75 ; 28.2,11.75 ; 28.2,12.75 ; 28.2,13.75
; 31.7,11.75 ; 31.7,12.75; 31.7,13.75]; % m - hotspots' coordinates
HS=length(hotspot_crd(:,1)); % number of hot spots
target=10;
target_crd=hotspot_crd(target,:); % m - target coordinates
sensors_crd=[13.97,15.3 ; 29.35,14.66 ; 21.86,9.42]; % m - sensor
coordinates
l=transpose(sqrt((target_crd(1)-sensors_crd(:,1)).^2+(target_crd(2)-
sensors_crd(:,2)).^2)); % m - distance of sensors from target
K=length(l); % number of sensors
l_FC=zeros(length(hotspot_crd(:,1)),K);
for i=1:length(hotspot_crd(:,1))  % m - (HSxK matrix) distances between HS
hot spots and K sensors
    for j=1:K
        l_FC(i,j)=sqrt((hotspot_crd(i,1)-
sensors_crd(j,1))^2+(hotspot_crd(i,2)-sensors_crd(j,2))^2);
    end
end
l0=min(min(l_FC)); % m - reference distance
TL=alpha_water*(l-l0)*10^-3+spread_cf*10*log10(l/l0); % dB - Transmission
Loss
AAF=exp(-TL./(20*log10(exp(1)))); % AAF (TL converted in Nepers)
var_noise=1; % variance of noise (gaussian)
SNR=20; % signal-to-noise ratio at closest (sensor/hot spot)
var_fadsig=var_noise*SNR; % variance of fading (gaussian)

noise_ampl=zeros(K,non_rel);
for i=1:K
    noise_ampl(i,:)=sqrt(var_noise)*randn(1,non_rel);
end
rel_ampl=zeros(K,rel);
for i=1:K

rel_ampl(i,:)=AAF(i)*sqrt(var_fadsig)*randn(1,rel)+sqrt(var_noise)*randn(1,
rel);
end
u=1;
v=1;
ampl=zeros(K,tot); % pre-allocation for final scenario
for i=1:tot
    switch X(i)
        case 1
            for j=1:K
                ampl(j,i)=rel_ampl(j,u);
            end
            u=u+1;
        case 0
            for j=1:K
                ampl(j,i)=noise_ampl(j,v);
```

```
            end
            v=v+1;
    end
end

%% Sensors' Settings

disp('Sensors setting calculation') % display the progress

th_loc_poss=0:.0001:10; % threshold for local decision (non-negative)

f_FC=f; % kHz - sound frequency
T_FC=T; % ∞C - temperature (T=3.8∞C at Goliat)
S_FC=S; % ppt - salinity (S=35 ppt at Goliat)
D_FC=D; % m - depth (D=350-400 m at Goliat)
pH_FC=pH; % pH of water (pH=8 at Goliat)
spread_cf_FC=spread_cf; % spreading coeff.
(spherical=2;cylindrical=1;practical=1.5)
c_FC=speed_sound(D_FC,T_FC,S_FC); % m/s - speed of sound underwater
alpha_water_FC=absorption_FG(f_FC,T_FC,S_FC,D_FC,pH_FC,c_FC); % dB/km -
absorption due to water
TL_FC=alpha_water_FC*(l_FC-l0)*10^-3+spread_cf_FC*10*log10(l_FC/l0); % dB -
Transmission Loss
AAF_FC=exp(-TL_FC/(20*log10(exp(1)))); % AAF (TL converted in Nepers)
var_noise_FC=1; % variance of noise (gaussian)
SNR_FC=20; % signal-to-noise ratio at closest (sensor/hot spot)
var_fadsig_FC=var_noise_FC*SNR_FC; % variance of fading (gaussian)
j_loc=zeros(HS,K);
j_loc_db=zeros(length(th_loc_poss),K);
th_loc=zeros(1,K);
for i=1:length(th_loc_poss)
    pfa_FC_poss=2*qfunc(sqrt(th_loc_poss(i)/var_noise_FC)); % array of
probabilities of false alarm for the single sensors

pd_FC_poss=2*qfunc(sqrt(th_loc_poss(i)./(var_fadsig_FC*AAF_FC.^2+var_noise_
FC))); % array of the corresponing probabilities of detection
    j_loc=pd_FC_poss-pfa_FC_poss;
    j_loc_db(i,:)=mean(j_loc);
end

for i=1:K
    th_loc(i)=th_loc_poss(j_loc_db(:,i)==max(j_loc_db(:,i)));
end

pfa_FC=2*qfunc(sqrt(th_loc/var_noise_FC)); % array of probabilities of
false alarm for the single sensors
pd_FC=zeros(HS,K);
for i=1:K
pd_FC(:,i)=2*qfunc(sqrt(th_loc(i)./(var_fadsig_FC*AAF_FC(:,i).^2+var_noise_
FC))); % array of the corresponing probabilities of detection
end
pd_FC=mean(pd_FC);

%% FC's Settings

disp('FC Settings Calculation') % display the progress

p_FC=p;  % parameter for release shape
r_FC=r;    % parameter for release shape
N_FC=N; % number of instants
```

```
X_FC=binary_source(p_FC,r_FC,N_FC); % function giving an array that
simulates the events before the continuous release
tot_FC=length(X_FC); % total number of records
rel_FC=nnz(X_FC); % number of releases (positive events)
non_rel_FC=tot_FC-rel_FC; % number of non-releases (negative events)
noise_ampl_FC=zeros(K,non_rel_FC);
for i=1:K
    noise_ampl_FC(i,:)=sqrt(var_noise_FC)*randn(1,non_rel_FC);
end
rel_ampl_FC=zeros(K,rel_FC);
for i=1:K

rel_ampl_FC(i,:)=mean(AAF_FC(:,i))*sqrt(var_fadsig_FC)*randn(1,rel_FC)+sqrt
(var_noise_FC)*randn(1,rel_FC);
end
u=1;
v=1;
ampl_FC=zeros(K,tot_FC); % pre-allocation for final scenario
for i=1:tot_FC
    switch X_FC(i)
        case 1
            for j=1:K
                ampl_FC(j,i)=rel_ampl_FC(j,u);
            end
            u=u+1;
        case 0
            for j=1:K
                ampl_FC(j,i)=noise_ampl_FC(j,v);
            end
            v=v+1;
    end
end

detect_FC=zeros(K,tot_FC); % matrix containing all the decisions of any
sensor during the simulation (every row is a sensor)
for i=1:K
    for j=1:tot_FC
        if ampl_FC(i,j)^2>=th_loc(i)
            detect_FC(i,j)=1;
        else
            detect_FC(i,j)=0;
        end
    end
end

th=-1:K; % array of possibile thresholds
no_thr=length(th); % number of possible thresholds

result_FC=zeros(no_thr,tot_FC); % matrix containing all the decisions of
the FC for different thresholds
lambda_FC=zeros(1,tot_FC); % vector containing all the FR's results

for j=1:no_thr % for each threshold
    sentence=['FC setting calculation: elaboration threshold number '
,num2str(j), ' of ' ,num2str(no_thr)];
    disp(sentence)
    for i=1:tot_FC % for each instant
        lambda_FC(i)=sum(detect_FC(:,i)); % value of FR
        if lambda_FC(i)>th(j) % for-loop determining the decision of the FC
based on CR
            result_FC(j,i)=1;
```

```
        else
            result_FC(j,i)=0;
        end
    end
end

%% FC's Performances

FP=zeros(1,no_thr); % array of number of false alarms at different
thresholds
TP=zeros(1,no_thr); % array of number of detections at different thresholds
FN=zeros(1,no_thr); % array of number of false negatives at different
thresholds
TN=zeros(1,no_thr); % array of number of true negatives at different
thresholds
Q_F=zeros(1,no_thr); % array of probabilities of false alarm at different
thresholds
Q_D=zeros(1,no_thr); % array of probabilities of detecion at different
thresholds
precision=zeros(1,no_thr); % array of precisions at different thresholds

for i=1:no_thr % for each threshold
        sentence=['FC setting calculation: evaluation threshold number '
,num2str(i), ' of ' ,num2str(no_thr)];
        disp(sentence) % display the progress

        fa=find((X_FC==0)&(result_FC(i,:)==1));
        FP(i)=length(fa);    % number of false alarms FP
        Q_F(i)=FP(i)/non_rel_FC; % FPR

        d=find((X_FC==1)&(result_FC(i,:)==1));
        TP(i)=length(d); % number of detection (true positives)
        Q_D(i)=TP(i)/rel_FC; % TPR - True Positive Rate - Probability of
Detection

        FN(i)=rel_FC-TP(i); % False negatives
        TN(i)=non_rel_FC-FP(i); % True Negatives

        precision(i)=TP(i)/(TP(i)+FP(i)); % precision
end
precision(isnan(precision))=1; % built to overcome the presence of NaN
results when precision converges to 1

%% Removal of Useless Thresholds

disp('Removal of Useless Thresholds from FC setting calculation history')
j=2; % loop used to remove results when no change was determined varying
the threshold
result_FC_new(1,:)=result_FC(1,:);
Q_D_new(1)=Q_D(1);
Q_F_new(1)=Q_F(1);
precision_new(1)=precision(1);
th_new(1)=th(1);
TP_new(1)=TP(1);
FP_new(1)=FP(1);
FN_new(1)=FN(1);
TN_new(1)=TN(1);
for i=2:no_thr
    if Q_D(i)==Q_D(i-1) && Q_F(i)==Q_F(i-1) && precision(i)==precision(i-1)
    else
        Q_D_new(j)=Q_D(i);
```

```
            Q_F_new(j)=Q_F(i);
            precision_new(j)=precision(i);
            th_new(j)=th(i);
            TP_new(j)=TP(i);
            FP_new(j)=FP(i);
            FN_new(j)=FN(i);
            TN_new(j)=TN(i);
            result_FC_new(j,:)=result_FC(i,:);
            j=j+1;
        end
    end
Q_D=Q_D_new;
Q_F=Q_F_new;
precision=precision_new;
th=th_new;
TP=TP_new;
FP=FP_new;
FN=FN_new;
TN=TN_new;
result_FC=result_FC_new;
no_thr=length(th);
clearvars Q_D_new Q_F_new precision_new th_new TP_new FP_new FN_new TN_new
result_FC_new

%% Analysis of Parameters

disp('Analysis of Parameters for FC setting')

AUC=-trapz(Q_F,Q_D); % calculation of Area Under the Curve

% the matrix called "perf" has rows representing each indicator
% for each raw of the matrix the 1st value is the cut-off threshold at
indicator¥s maximum value
% the 2nd value is the Prob. of Detection at cut off threshold
% the 3rd value is the Prob. of False Alarm at cut off threshold
% the 4th value is the indicator¥s maximum value

J=Q_D-Q_F; % calculation of Youden's indexes
perf(1,4)=max(J); % maximum value of Youden's indexes
ind=find(J==perf(1,4)); % array's index for max Youden's index
perf(1,1)=th(ind); % cut off threshold at max J value
perf(1,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(1,2)=Q_D(ind); % Prob. of Detection at cut off threshold

beta_value=1; % 1: pr and rec equally weighted, 2: recall weighted higher
than precision, 0.5: precision weighted higher than recall
F_beta=(1+beta_value^2)*(precision.*Q_D)./(beta_value^2*precision+Q_D); %
F_value (armonic mean between precision and recall, with weight)
perf(2,4)=max(F_beta); % maximum value of F_beta
ind=find(F_beta==perf(2,4)); % array's index for max F_beta
perf(2,1)=th(ind); % cut off threshold at max F_beta
perf(2,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(2,2)=Q_D(ind); % Prob. of Detection at cut off threshold

dist_roc=(1-Q_D).^2+Q_F.^2; % sq. distance to (0,1)
perf(3,4)=min(dist_roc); % minimum sq. distance to (0,1)
ind=find(dist_roc==perf(3,4)); % array's index for min sq. distance to
(0,1)
perf(3,1)=th(ind); % cut off threshold at min sq. distance to (0,1)
perf(3,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(3,2)=Q_D(ind); % Prob. of Detection at cut off threshold
```

```
CZ=Q_D.*(1-Q_F); % CZ index
perf(4,4)=max(CZ); % maximum value of CZ index
ind=find(CZ==perf(4,4)); % array's index for max CZ index
perf(4,1)=th(ind); % cut off threshold at max CZ index
perf(4,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(4,2)=Q_D(ind); % Prob. of Detection at cut off threshold

%% Sensors' Detection

detect=zeros(K,tot); % matrix containing all the decisions of any sensor
during the simulation (every row is a sensor)
for i=1:K
    sentence=['simulating detection: sensor ' ,num2str(i), ' of '
,num2str(K)]; % display the progress
    disp(sentence)
    for j=1:tot
        if ampl(i,j)^2>=th_loc(i)
            detect(i,j)=1;
        else
            detect(i,j)=0;
        end
    end
end

Q_F_test_loc=zeros(1,K);
Q_D_test_loc=zeros(1,K);

for i=1:K
    fa_test_loc=find((X==0)&(detect(i,:)==1));
    FP_test_loc=length(fa_test_loc);    % number of false alarms FP
    Q_F_test_loc(i)=FP_test_loc/non_rel; % FPR

    d_test_loc=find((X==1)&(detect(i,:)==1));
    TP_test_loc=length(d_test_loc); % number of detection (true positives)
    Q_D_test_loc(i)=TP_test_loc/rel; % TPR - True Positive Rate -
Probability of Detection
end

%% Fusion Center's Detection and Localization

inst_FC=1; % number of instants to take into account in the TCR
time_th=0; % threshold used in TCR
SP_calc_inst=zeros(tot,2);
index_rel=0;
SP_calc_db=zeros(tot,2);
SP_num_db(1:2,1:tot)=NaN;
SP_calc=[0 0];

dist_SP_HS=zeros(1,HS);
result=zeros(1,tot); % matrix containing all the decisions of the FC for
different thresholds
lambda=zeros(1,tot); % vector containing all the FR's results
lambda_time=zeros(1,tot); % vector containing all the TCR's results
final_result=zeros(1,tot); % matrix containing final decision (WFR+TCR)

for i=1:tot % for each instant
        if rem(i,100)==0 || i==1 % display the progress
        sentence=['Instant simulated by FC: ' ,num2str(i), ' of '
,num2str(tot)];
        disp(sentence)
```

```
        end
        lambda(i)=sum(detect(:,i)); % value of FR
        if lambda(i)>perf(1,1) % for-loop determining the decision of the
FC based on CR
            result(i)=1;
        else
            result(i)=0;
        end

        if result(i)==0
            inst=inst_FC;
            if inst>i % necessary for the first instants (when there aren't
enough previous data)
                inst=i;
            end
            lambda_time(i)=sum(result(i-inst+1:i));
            if lambda_time(i)>time_th % for-loop determining the final
decision of the FC
                final_result(i)=1;
            else
                final_result(i)=0;
            end
            else
             final_result(i)=1;
        end

        if result(i)==1 && sum(detect(:,i))>=1
            index_rel=index_rel+1;

SP_calc_inst(index_rel,:)=sum(sensors_crd.*detect(:,i))/sum(detect(:,i));
            if index_rel==1
                SP_calc=[SP_calc_inst(index_rel,1)
SP_calc_inst(index_rel,2)];
            else
                SP_calc=[((index_rel-
1)*SP_calc(1)+SP_calc_inst(index_rel,1))/index_rel ((index_rel-
1)*SP_calc(2)+SP_calc_inst(index_rel,2))/index_rel];
            end
            for count_HS=1:HS
                dist_SP_HS(count_HS)=sqrt((SP_calc(1)-
hotspot_crd(count_HS,1))^2+(SP_calc(2)-hotspot_crd(count_HS,2))^2);
            end
            SP_num=find(dist_SP_HS==min(dist_SP_HS));
            SP_calc_db(i,:)=SP_calc;
            SP_num_db(1:length(SP_num),i)=transpose(SP_num);
        else
            if sum(sum(SP_calc_inst))==0
                SP_calc_db(i,:)=NaN;
            else
                SP_calc_db(i,:)=SP_calc_db(i-1,:);
            end
        end
        if final_result(i)==1 && result(i)==0
            SP_calc_db(i,:)=SP_calc;
            SP_num_db(1:length(SP_num),i)=transpose(SP_num);
        end
end

fa_test=find((X==0)&(final_result==1));
FP_test=length(fa_test);    % number of false alarms FP
Q_F_test=FP_test/non_rel; % FPR
```

```
d_test=find((X==1)&(final_result==1));
TP_test=length(d_test); % number of detection (true positives)
Q_D_test=TP_test/rel; % TPR - True Positive Rate - Probability of Detection

precision_test=TP_test/(TP_test+FP_test); % precision

%% Animation

answer = questdlg('Do you want to see the animation?','Yes','No');
switch answer
    case 'Yes'
        answer = 1;
    case 'No'
        answer = 0;
end
if answer==1
sentence='Show Animation';
disp(sentence)
img=imread('template.png');
figure(1)
for i=1:tot
hold off
imagesc([0 41.26],[0 27.45],flipud(img));
set(gca,'ydir','normal');
set(gcf, 'Position', get(0, 'Screensize'));
xlim([0 41.26])
ylim([0 27.45])
xlabel('x (meters)')
ylabel('y (meters)')
grid on
hold on
scatter(hotspot_crd(:,1),hotspot_crd(:,2),400,'filled','r')
if X(i)==1
    scatter(target_crd(1),target_crd(2),400,'filled','b')
end
scatter(sensors_crd(:,1),sensors_crd(:,2),400,'filled','g')
    if isnan(SP_calc_db(i,1))==0
        scatter(SP_calc_db(i,1),SP_calc_db(i,2),100,'filled','m')
    end
    if isnan(SP_num_db(1,i))==0 && isnan(SP_num_db(2,i))==1

scatter(hotspot_crd(SP_num_db(1,i),1),hotspot_crd(SP_num_db(1,i),2),100,'fi
lled','y')
    else
        if isnan(SP_num_db(1,i))==0 && isnan(SP_num_db(2,i))==0

scatter(hotspot_crd(SP_num_db(1:size(SP_num_db,1),i),1),hotspot_crd(SP_num_
db(1:size(SP_num_db,1),i),2),100,'filled','y')
        end
    end
title(['2D Scenario. Instant number: ' num2str(i) ' of ' num2str(tot)])
drawnow
end
end

%% Plot

disp('Plot and curves generation')

figure(1) % release plot
```

```
plot(X)
xlabel('time (s)')
ylabel('0 = Non-release / 1 = Release')
xlim([0 tot])
ylim([0 1])

figure(2) % Square attenuation vs. Distance
x_plot=l0:.1:max(l);
x_plot=[x_plot max(l)];
TL_plot=alpha_water*(x_plot-l0)*10^-3+spread_cf*10*log10(x_plot/l0);
AAF_plot=exp(-TL_plot./(20*log10(exp(1))));
plot(x_plot,AAF_plot,'b')
xlabel('Distance (m)')
ylabel('Attenuation (AAF)')
for i=1:K
hold on
line([l(i) l(i)],[0 1]) % vertical lines representing sensors position
end
hold off
xlim([l0-1 max(l)+1])
ylim([0 max(AAF_plot)])
title('Attenuation vs Distance')

figure(3) % wave amplitude vs time plot
for i=1:K
    z0=ampl(i,:);
    z1=ampl(i,:);
    for j=1:tot
        switch X(j)
            case 0
                z1(j)=NaN;
            case 1
                z0(j)=NaN;
        end
    end
    subplot(K,1,i)
    plot(z0,'b')
    hold on
    plot(z1,'r')
    xlabel('time (s)')
    ylabel('y (Amplitude)')
    title(['Received Signal, sensor ' num2str(i) ', coordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
    hold off
end

figure(4) % Amplitude Distribution Plot
for i=1:K
subplot(1,K,i)
histogram(noise_ampl(i,:),'Normalization','pdf')
hold on
histogram(rel_ampl(i,:),'Normalization','pdf')
hold on
line([sqrt(th_loc(i))
sqrt(th_loc(i))],get(gca,'ylim'),'LineWidth',2,'Color','k')
line([-sqrt(th_loc(i)) -
sqrt(th_loc(i))],get(gca,'ylim'),'LineWidth',2,'Color','k')
legend('p(y|H0)','p(y|H1)')
title(['Amplitude pdf, sensor ' num2str(i) ', coordordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
xlabel('y (Amplitude)')
```

```
hold off
end

figure(5)
for i=1:K
    subplot(1,K,i)
    th_loc_poss=0:.01:10;

plot(2*qfunc(sqrt(th_loc_poss/var_noise)),2*qfunc(sqrt(th_loc_poss/(var_fad
sig*AAF_FC(target,i)^2+var_noise))))
    hold on

scatter(2*qfunc(sqrt(th_loc(i)/var_noise)),2*qfunc(sqrt(th_loc(i)/(var_fads
ig*AAF_FC(target,i)^2+var_noise))),'filled')
    line([0 1],[0 1],'LineStyle','--')
    hold off
    title(['ROC sensor ' num2str(i) ', coordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
    xlabel('Local Probability of False Alarm')
    ylabel('Local Probability of Detection')
    xlim([0 1])
    ylim([0 1])
    pbaspect([1 1 1])
end

figure(6) % ROC curve - linear
plot(Q_F,Q_D,'-or')
hold on
line([0 1],[0 1],'LineStyle','--') % chance line
scatter(Q_F_test,Q_D_test,'filled')
hold off
title('ROC CURVE - linear')
xlabel('Global Probability of False Alarm')
ylabel('Global Probability of Detection')
xlim([0 1])
ylim([0 1])

figure(7) % ROC curve - semilog
semilogx(Q_F,Q_D,'-or')
hold on
scatter(Q_F_test,Q_D_test,'filled')
hold off
title('ROC CURVE - semilog')
xlabel('Global Probability of False Alarm')
ylabel('Global Probability of Detection')
xlim([Q_F(end-1) 1])
ylim([Q_D(end-1)-0.05 1])

figure(8) % PR curve - linear
plot(Q_D,precision,'-or')
hold on
scatter(Q_D_test,precision_test,'filled')
hold off
title('PR CURVE')
xlabel('Recall')
ylabel('Precision')
xlim([0 1])
ylim([0 1])

SP_num_db_plot=zeros(1,index_rel);
dist_plot=zeros(1,index_rel);
```

```
j=1;
for i=1:N
    if result(i)==1
        SP_num_db_plot(j)=SP_num_db(1,i);
        dist_plot(j)=sqrt((target_crd(1)-
hotspot_crd(SP_num_db_plot(j),1))^2+(target_crd(2)-
hotspot_crd(SP_num_db_plot(j),2))^2);
        j=j+1;
    end
end
AUC_localization=trapz(1:index_rel,dist_plot);
figure(9)
plot(1:index_rel,dist_plot)
title(['Distance from Target. Area under curve (no. of instants: '
num2str(index_rel) ') = ' num2str(AUC_localization) ''])
xlabel('Time (s) / Fusion Center Positive Detection')
ylabel('Distance from Target (m)')
xlim([1 index_rel])
ylim([0 max(dist_plot)])

%% Memory Cleaning

disp('Memory Cleaning')
clearvars x_plot TL_plot AAF_plot x d fa j i q s sentence img u v
noise_ampl_FC rel_ampl_FC ind th_loc_poss pfa_FC_poss pd_FC_poss
fa_test_loc FP_test_loc d_test_loc TP_test_loc % deleting variables

%% Functions

function x=binary_source(p,r,N)

A10=(1-p)/(p+r-p*r);
A01=p/(p+r-p*r);

x=zeros(1,N); % pre-allocation of memory to improve code performances

for n=2:N % it starts from 2 because we want the initial state n=1 to be
without release
    switch x(n-1)
        case 0
            x(n)=(rand<A01);
        case 1
            x(n)=(rand<1-A10);
    end
end
end

function y=speed_sound(D,T,S)

% it calculates sound speed using corrected UNESCO algorithm

c0=1402.388+5.03711*T-5.80852e-2*T^2+3.3420e-4*T^3-1.47800e-6*T^4+3.1464e-
9*T^5;
c1=0.153563+6.8982e-4*T-8.1788e-6*T^2+1.3621e-7*T^3-6.1185e-10*T^4;
c2=3.1260e-5-1.7107e-6*T+2.5974e-8*T^2-2.5335e-10*T^3+1.0405e-12*T^4;
c3=-9.7729e-9-3.8504e-10*T-2.3643e-12*T^2;
A0=1.389-1.262e-2*T+7.164e-5*T^2+2.006e-6*T^3-3.21e-8*T^4;
A1=9.4742e-5-1.2580e-5*T-6.4885e-8*T^2+1.0507e-8*T^3-2.0122e-10*T^4;
A2=-3.9064e-7+9.1041e-9*T-1.6002e-10*T^2+7.988e-12*T^3;
A3=1.100e-10+6.649e-12*T-3.389e-13*T^2;
P=D/10;
```

```
A=A0+A1*P+A2*P^2+A3*P^3;
B=-1.922e-2-4.42e-5*T+(7.3637e-3+1.7945e-7*T)*P;
C=-7.9836e-6*P+1.727e-3;
y=c0+c1*P+c2*P^2+c3*P^3+A*S+B*S^(3/2)+C*S^2;
end

function y=absorption_FG(f,T,S,D,pH,c)

% this function calculates the attenuation coefficient in bB/km
% using Francois-Garrison method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m
% "pH" is the indicator of water acidity

theta=273.15+T; % K, temperature

% excess absorption due to Boric Acid (H3BO3)
A1=(8.86/c)*10^(0.78*pH-5); % dB/(km*kHz)
P1=1; % nondimensional pressure correction factor
f1=2.8*sqrt(S/35)*10^(4-1245/theta); % kHz, relaxation frequencies of H3BO3
alpha1=(A1*P1*f1*f.^2)/(f1^2+f.^2); % dB/km

% excess absorption due to Magnesium Sulfate (MgSO4)
A2=21.44*(S/c)*(1+0.025*T); % dB/(km*kHz)
P2=1-1.37e-4*D+6.2e-9*D^2; % nondimensional pressure correction factor
f2=(8.17*10^(8-1990/theta))/(1+0.0018*(S-35)); % kHz, relaxation
frequencies of MgSO4
alpha2=(A2*P2*f2*f.^2)/(f2^2+f.^2); % dB/km

% absorption due to water
if T<=20
    A3=4.937e-4-2.59e-5*T+9.11e-7*T^2-1.5e-8*T^3; % dB/(km*kHz^2)
else
    A3=3.964e-4-1.146e-5*T+1.45e-7*T^2-6.5e-10*T^3; % dB/(km*kHz^2)
end
P3=1-3.83e-5*D+4.9e-10*D^2; % nondimensional pressure correction factor
alpha3=A3*P3*f.^2; % dB/km

y=alpha1+alpha2+alpha3;
end
function y=absorption_FS(f,T,D)

% this function calculates the attenuation coefficient in bB/km
% using Fisher-Simmons method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "D" is the depth in m

f=f*1000; % method is written using Hz
theta=273.15+T; % K, temperature
P=D/10+1; % atm, absolute pressure estimation (supposing 10 m_seawater = 1
atm)

% excess absorption due to Boric Acid (H3BO3)
A1=1.03e-8+2.36e-10*T-5.22e-12*T^2; % Np/(m*Hz)
P1=1; % nondimensional pressure correction factor
f1=1.32e3*theta*exp(-1700/theta); % Hz, relaxation frequencies of H3BO3
alpha1=(A1*P1*f1*f.^2)/(f1^2+f.^2); % Np/m
```

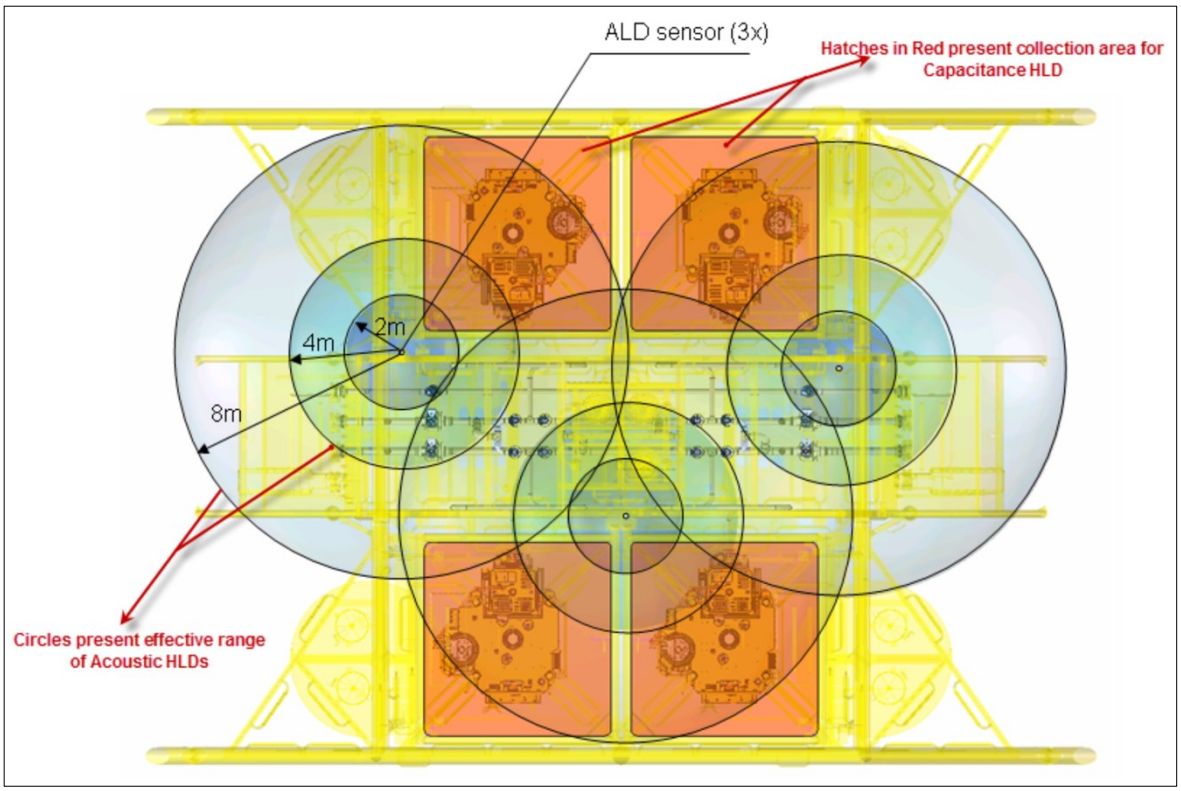
```
% excess absorption due to Magnesium Sulfate (MgSO4)
A2=5.62e-8+7.52e-10*T; % Np/(m*Hz)
P2=1-10.3e-4*P+3.7e-7*P^2; % nondimensional pressure correction factor
f2=1.55e7*theta*exp(-3052/theta); % Hz, relaxation frequencies of MgSO4
alpha2=(A2*P2*f2*f.^2)/(f2^2+f.^2); % Np/m

% absorption due to water
A3=(55.9-2.37*T+4.77e-2*T^2-3.48e-4*T^3)*10^-15; % Np/(m*Hz^2)
P3=1-384e-4*P+7.57e-8*P^2; % nondimensional pressure correction factor
alpha3=A3*P3*f.^2; % Np/m

y=alpha1+alpha2+alpha3; % Np/m
y=y*1000*20*log10(exp(1)); % dB/km
end
function y=absorption_SM(f,T,S,D)

% this function calculates the attenuation coefficient in bB/km
% using Shulkin-Marsh method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m

theta=273.15+T; % K, temperature
P=D/10+1; % atm, absolute pressure estimation (supposing 10 m_seawater = 1
atm)
A=2.34e-6;
B=3.38e-6;
fT=21.9*10^(6-1520/theta); % kHz
y=((S*A*fT*f.^2)/(fT^2+f.^2)+(B*f.^2)/fT)*(1-6.54e-4*P); % Np/m
y=y*1000*20*log10(exp(1)); % dB/km
end
function y=absorption_AM(f,T,S,D,pH)

% this function calculates the attenuation coefficient in bB/km
% using Ainslie-McColm method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m
% "pH" is the indicator of water acidity

D=D/1000; % method is written using km

% excess absorption due to Boric Acid (H3BO3)
f1=0.78*exp(T/26)*sqrt(S/35); % kHz, relaxation frequencies of H3BO3
alpha1=0.106*(f1*f.^2)/(f1^2+f.^2)*exp((pH-8)/0.56); % dB/km

% excess absorption due to Magnesium Sulfate (MgSO4)
f2=42*exp(T/17); % kHz, relaxation frequencies of MgSO4
alpha2=0.52*(1+T/43)*(S/35)*(f2*f.^2)/(f2^2+f.^2)*exp(-D/6); % dB/km

% absorption due to water
alpha3=4.9e-4*f.^2*exp(-T/27+D/17); % dB/km

y=alpha1+alpha2+alpha3;
end
function y=absorption_T(f)

% this function calculates the attenuation coefficient in bB/km
% using Thorp method
```

```
% "f" is the frequency of the sound in kHz

y=1.0936132983*(0.1*f.^2/(1+f.^2)+40*f.^2/(4100+f.^2));
end
```

Appendix 2: MATLAB Script (Hotspot 10: CR + CBM-TOSP)

This MATLAB script was written using MATLAB R2019; it simulates the release in HS 10, the FC performs the CR for detection and CBM-TOSP for localization. All the settings are those described in the previous chapters.

MATLAB, in order to run the animation, must have the following picture in a folder accessible by MATLAB. The file must be named "template.png".




```
%%%%%  Oil Spill Detection and Localization    %%%%
%%%%%                CR + CBM-TOSP             %%%%
%%%%%                 Hotspot 10               %%%%
%%%%%          Author: Gianluca Tabella        %%%%

clear
close all
clc

%% Release Modelling

disp('Release Modelling') % display the progress

p=1/3;  % parameter for release shape
r=2;    % parameter for release shape
N=10000; % number of instants
X=binary_source(p,r,N); % function giving an array that simulates the
events before the continuous release
tot=length(X); % total number of records
rel=nnz(X); % number of releases (positive events)
non_rel=tot-rel; % number of non-releases (negative events)

%% Signal Modelling

disp('Signal Modelling') % display the progress
```

```
f=20e-3; % kHz - sound frequency
T=3.8; % ∞C - temperature (T=3.8∞C at Goliat)
S=35; % ppt - salinity (S=35 ppt at Goliat)
D=350; % m - depth (D=350-400 m at Goliat)
pH=8; % pH of water (pH=8 at Goliat)
spread_cf=1.5; % spreading coeff. (spherical=2;cylindrical=1;practical=1.5)

c=speed_sound(D,T,S); % m/s - speed of sound underwater
alpha_water=absorption_FG(f,T,S,D,pH,c); % dB/km - absorption due to water
hotspot_crd=[11.5,11.75 ; 11.5,12.75 ; 11.5,13.75 ; 15,11.75 ; 15,12.75 ;
15,13.75 ; 18,11.75 ; 18,12.75 ; 19,11.75 ; 19,12.75 ; 24.5,11.75 ;
24.5,12.75 ; 25.5,11.75 ; 25.5,12.75 ; 28.2,11.75 ; 28.2,12.75 ; 28.2,13.75
; 31.7,11.75 ; 31.7,12.75; 31.7,13.75]; % m - hotspots' coordinates
HS=length(hotspot_crd(:,1)); % number of hot spots
target=10;
target_crd=hotspot_crd(target,:); % m - target coordinates
sensors_crd=[13.97,15.3 ; 29.35,14.66 ; 21.86,9.42]; % m - sensor
coordinates
l=transpose(sqrt((target_crd(1)-sensors_crd(:,1)).^2+(target_crd(2)-
sensors_crd(:,2)).^2)); % m - distance of sensors from target
K=length(l); % number of sensors
l_FC=zeros(length(hotspot_crd(:,1)),K);
for i=1:length(hotspot_crd(:,1))  % m - (HSxK matrix) distances between HS
hot spots and K sensors
    for j=1:K
        l_FC(i,j)=sqrt((hotspot_crd(i,1)-
sensors_crd(j,1))^2+(hotspot_crd(i,2)-sensors_crd(j,2))^2);
    end
end
l0=min(min(l_FC)); % m - reference distance
TL=alpha_water*(l-l0)*10^-3+spread_cf*10*log10(l/l0); % dB - Transmission
Loss
AAF=exp(-TL./(20*log10(exp(1)))); % AAF (TL converted in Nepers)
var_noise=1; % variance of noise (gaussian)
SNR=20; % signal-to-noise ratio at closest (sensor/hot spot)
var_fadsig=var_noise*SNR; % variance of fading (gaussian)

noise_ampl=zeros(K,non_rel);
for i=1:K
    noise_ampl(i,:)=sqrt(var_noise)*randn(1,non_rel);
end
rel_ampl=zeros(K,rel);
for i=1:K

rel_ampl(i,:)=AAF(i)*sqrt(var_fadsig)*randn(1,rel)+sqrt(var_noise)*randn(1,
rel);
end
u=1;
v=1;
ampl=zeros(K,tot); % pre-allocation for final scenario
for i=1:tot
    switch X(i)
        case 1
            for j=1:K
                ampl(j,i)=rel_ampl(j,u);
            end
            u=u+1;
        case 0
            for j=1:K
                ampl(j,i)=noise_ampl(j,v);
```

```
            end
            v=v+1;
    end
end

%% Sensors' Settings

disp('Sensors setting calculation') % display the progress

th_loc_poss=0:.0001:10; % threshold for local decision (non-negative)

f_FC=f; % kHz - sound frequency
T_FC=T; % ∞C - temperature (T=3.8∞C at Goliat)
S_FC=S; % ppt - salinity (S=35 ppt at Goliat)
D_FC=D; % m - depth (D=350-400 m at Goliat)
pH_FC=pH; % pH of water (pH=8 at Goliat)
spread_cf_FC=spread_cf; % spreading coeff.
(spherical=2;cylindrical=1;practical=1.5)
c_FC=speed_sound(D_FC,T_FC,S_FC); % m/s - speed of sound underwater
alpha_water_FC=absorption_FG(f_FC,T_FC,S_FC,D_FC,pH_FC,c_FC); % dB/km -
absorption due to water
TL_FC=alpha_water_FC*(l_FC-l0)*10^-3+spread_cf_FC*10*log10(l_FC/l0); % dB -
Transmission Loss
AAF_FC=exp(-TL_FC/(20*log10(exp(1)))); % AAF (TL converted in Nepers)
var_noise_FC=1; % variance of noise (gaussian)
SNR_FC=20; % signal-to-noise ratio at closest (sensor/hot spot)
var_fadsig_FC=var_noise_FC*SNR_FC; % variance of fading (gaussian)
j_loc=zeros(HS,K);
j_loc_db=zeros(length(th_loc_poss),K);
th_loc=zeros(1,K);
for i=1:length(th_loc_poss)
    pfa_FC_poss=2*qfunc(sqrt(th_loc_poss(i)/var_noise_FC)); % array of
probabilities of false alarm for the single sensors

pd_FC_poss=2*qfunc(sqrt(th_loc_poss(i)./(var_fadsig_FC*AAF_FC.^2+var_noise_
FC))); % array of the corresponing probabilities of detection
    j_loc=pd_FC_poss-pfa_FC_poss;
    j_loc_db(i,:)=mean(j_loc);
end

for i=1:K
    th_loc(i)=th_loc_poss(j_loc_db(:,i)==max(j_loc_db(:,i)));
end

pfa_FC=2*qfunc(sqrt(th_loc/var_noise_FC)); % array of probabilities of
false alarm for the single sensors
pd_FC=zeros(HS,K);
for i=1:K
pd_FC(:,i)=2*qfunc(sqrt(th_loc(i)./(var_fadsig_FC*AAF_FC(:,i).^2+var_noise_
FC))); % array of the corresponing probabilities of detection
end
pd_FC=mean(pd_FC);

%% FC's Settings

disp('FC Settings Calculation') % display the progress

p_FC=p;  % parameter for release shape
r_FC=r;    % parameter for release shape
N_FC=N; % number of instants
```

```
X_FC=binary_source(p_FC,r_FC,N_FC); % function giving an array that
simulates the events before the continuous release
tot_FC=length(X_FC); % total number of records
rel_FC=nnz(X_FC); % number of releases (positive events)
non_rel_FC=tot_FC-rel_FC; % number of non-releases (negative events)
noise_ampl_FC=zeros(K,non_rel_FC);
for i=1:K
    noise_ampl_FC(i,:)=sqrt(var_noise_FC)*randn(1,non_rel_FC);
end
rel_ampl_FC=zeros(K,rel_FC);
for i=1:K

rel_ampl_FC(i,:)=mean(AAF_FC(:,i))*sqrt(var_fadsig_FC)*randn(1,rel_FC)+sqrt
(var_noise_FC)*randn(1,rel_FC);
end
u=1;
v=1;
ampl_FC=zeros(K,tot_FC); % pre-allocation for final scenario
for i=1:tot_FC
    switch X_FC(i)
        case 1
            for j=1:K
                ampl_FC(j,i)=rel_ampl_FC(j,u);
            end
            u=u+1;
        case 0
            for j=1:K
                ampl_FC(j,i)=noise_ampl_FC(j,v);
            end
            v=v+1;
    end
end

detect_FC=zeros(K,tot_FC); % matrix containing all the decisions of any
sensor during the simulation (every row is a sensor)
for i=1:K
    for j=1:tot_FC
        if ampl_FC(i,j)^2>=th_loc(i)
            detect_FC(i,j)=1;
        else
            detect_FC(i,j)=0;
        end
    end
end

th=-1:K; % array of possibile thresholds
no_thr=length(th); % number of possible thresholds

result_FC=zeros(no_thr,tot_FC); % matrix containing all the decisions of
the FC for different thresholds
lambda_FC=zeros(1,tot_FC); % vector containing all the FR's results

for j=1:no_thr % for each threshold
    sentence=['FC setting calculation: elaboration threshold number '
,num2str(j), ' of ' ,num2str(no_thr)];
    disp(sentence)
    for i=1:tot_FC % for each instant
        lambda_FC(i)=sum(detect_FC(:,i)); % value of FR
        if lambda_FC(i)>th(j) % for-loop determining the decision of the FC
based on CR
            result_FC(j,i)=1;
```

```
        else
            result_FC(j,i)=0;
        end
    end
end

%% FC's Performances

FP=zeros(1,no_thr); % array of number of false alarms at different
thresholds
TP=zeros(1,no_thr); % array of number of detections at different thresholds
FN=zeros(1,no_thr); % array of number of false negatives at different
thresholds
TN=zeros(1,no_thr); % array of number of true negatives at different
thresholds
Q_F=zeros(1,no_thr); % array of probabilities of false alarm at different
thresholds
Q_D=zeros(1,no_thr); % array of probabilities of detecion at different
thresholds
precision=zeros(1,no_thr); % array of precisions at different thresholds

for i=1:no_thr % for each threshold
        sentence=['FC setting calculation: evaluation threshold number '
,num2str(i), ' of ' ,num2str(no_thr)];
        disp(sentence) % display the progress

        fa=find((X_FC==0)&(result_FC(i,:)==1));
        FP(i)=length(fa);    % number of false alarms FP
        Q_F(i)=FP(i)/non_rel_FC; % FPR

        d=find((X_FC==1)&(result_FC(i,:)==1));
        TP(i)=length(d); % number of detection (true positives)
        Q_D(i)=TP(i)/rel_FC; % TPR - True Positive Rate - Probability of
Detection

        FN(i)=rel_FC-TP(i); % False negatives
        TN(i)=non_rel_FC-FP(i); % True Negatives

        precision(i)=TP(i)/(TP(i)+FP(i)); % precision
end
precision(isnan(precision))=1; % built to overcome the presence of NaN
results when precision converges to 1

%% Removal of Useless Thresholds

disp('Removal of Useless Thresholds from FC setting calculation history')
j=2; % loop used to remove results when no change was determined varying
the threshold
result_FC_new(1,:)=result_FC(1,:);
Q_D_new(1)=Q_D(1);
Q_F_new(1)=Q_F(1);
precision_new(1)=precision(1);
th_new(1)=th(1);
TP_new(1)=TP(1);
FP_new(1)=FP(1);
FN_new(1)=FN(1);
TN_new(1)=TN(1);
for i=2:no_thr
    if Q_D(i)==Q_D(i-1) && Q_F(i)==Q_F(i-1) && precision(i)==precision(i-1)
    else
        Q_D_new(j)=Q_D(i);
```

```
            Q_F_new(j)=Q_F(i);
            precision_new(j)=precision(i);
            th_new(j)=th(i);
            TP_new(j)=TP(i);
            FP_new(j)=FP(i);
            FN_new(j)=FN(i);
            TN_new(j)=TN(i);
            result_FC_new(j,:)=result_FC(i,:);
            j=j+1;
        end
    end
Q_D=Q_D_new;
Q_F=Q_F_new;
precision=precision_new;
th=th_new;
TP=TP_new;
FP=FP_new;
FN=FN_new;
TN=TN_new;
result_FC=result_FC_new;
no_thr=length(th);
clearvars Q_D_new Q_F_new precision_new th_new TP_new FP_new FN_new TN_new
result_FC_new

%% Analysis of Parameters

disp('Analysis of Parameters for FC setting')

AUC=-trapz(Q_F,Q_D); % calculation of Area Under the Curve

% the matrix called "perf" has rows representing each indicator
% for each raw of the matrix the 1st value is the cut-off threshold at
indicator¥s maximum value
% the 2nd value is the Prob. of Detection at cut off threshold
% the 3rd value is the Prob. of False Alarm at cut off threshold
% the 4th value is the indicator¥s maximum value

J=Q_D-Q_F; % calculation of Youden's indexes
perf(1,4)=max(J); % maximum value of Youden's indexes
ind=find(J==perf(1,4)); % array's index for max Youden's index
perf(1,1)=th(ind); % cut off threshold at max J value
perf(1,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(1,2)=Q_D(ind); % Prob. of Detection at cut off threshold

beta_value=1; % 1: pr and rec equally weighted, 2: recall weighted higher
than precision, 0.5: precision weighted higher than recall
F_beta=(1+beta_value^2)*(precision.*Q_D)./(beta_value^2*precision+Q_D); %
F_value (armonic mean between precision and recall, with weight)
perf(2,4)=max(F_beta); % maximum value of F_beta
ind=find(F_beta==perf(2,4)); % array's index for max F_beta
perf(2,1)=th(ind); % cut off threshold at max F_beta
perf(2,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(2,2)=Q_D(ind); % Prob. of Detection at cut off threshold

dist_roc=(1-Q_D).^2+Q_F.^2; % sq. distance to (0,1)
perf(3,4)=min(dist_roc); % minimum sq. distance to (0,1)
ind=find(dist_roc==perf(3,4)); % array's index for min sq. distance to
(0,1)
perf(3,1)=th(ind); % cut off threshold at min sq. distance to (0,1)
perf(3,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(3,2)=Q_D(ind); % Prob. of Detection at cut off threshold
```

```
CZ=Q_D.*(1-Q_F); % CZ index
perf(4,4)=max(CZ); % maximum value of CZ index
ind=find(CZ==perf(4,4)); % array's index for max CZ index
perf(4,1)=th(ind); % cut off threshold at max CZ index
perf(4,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(4,2)=Q_D(ind); % Prob. of Detection at cut off threshold

%% Sensors' Detection

detect=zeros(K,tot); % matrix containing all the decisions of any sensor
during the simulation (every row is a sensor)
for i=1:K
    sentence=['simulating detection: sensor ' ,num2str(i), ' of '
,num2str(K)]; % display the progress
    disp(sentence)
    for j=1:tot
        if ampl(i,j)^2>=th_loc(i)
            detect(i,j)=1;
        else
            detect(i,j)=0;
        end
    end
end

Q_F_test_loc=zeros(1,K);
Q_D_test_loc=zeros(1,K);

for i=1:K
    fa_test_loc=find((X==0)&(detect(i,:)==1));
    FP_test_loc=length(fa_test_loc);    % number of false alarms FP
    Q_F_test_loc(i)=FP_test_loc/non_rel; % FPR

    d_test_loc=find((X==1)&(detect(i,:)==1));
    TP_test_loc=length(d_test_loc); % number of detection (true positives)
    Q_D_test_loc(i)=TP_test_loc/rel; % TPR - True Positive Rate -
Probability of Detection
end

%% Fusion Center's Detection and Localization

inst_FC=1; % number of instants to take into account in the TCR
time_th=0; % threshold used in TCR
SP_calc_inst=zeros(tot,2);
index_rel=0;
SP_calc_db=zeros(tot,2);
SP_num_db(1:2,1:tot)=NaN;
SP_calc=[0 0];

dist_SP_HS=zeros(1,HS);
result=zeros(1,tot); % matrix containing all the decisions of the FC for
different thresholds
lambda=zeros(1,tot); % vector containing all the FR's results
lambda_time=zeros(1,tot); % vector containing all the TCR's results
final_result=zeros(1,tot); % matrix containing final decision (WFR+TCR)

for i=1:tot % for each instant
        if rem(i,100)==0 || i==1 % display the progress
        sentence=['Instant simulated by FC: ' ,num2str(i), ' of '
,num2str(tot)];
        disp(sentence)
```

```
        end
        lambda(i)=sum(detect(:,i)); % value of FR
        if lambda(i)>perf(1,1) % for-loop determining the decision of the
FC based on CR
            result(i)=1;
        else
            result(i)=0;
        end

        if result(i)==0
            inst=inst_FC;
            if inst>i % necessary for the first instants (when there aren't
enough previous data)
                inst=i;
            end
            lambda_time(i)=sum(result(i-inst+1:i));
            if lambda_time(i)>time_th % for-loop determining the final
decision of the FC
                final_result(i)=1;
            else
                final_result(i)=0;
            end
            else
             final_result(i)=1;
        end

        if result(i)==1 && sum(detect(:,i))>=1
            index_rel=index_rel+1;

SP_calc_inst(index_rel,:)=sum((sensors_crd.*detect(:,i))+((2*sum(sensors_cr
d.*detect(:,i))/sum(detect(:,i))-(1-detect(:,i)).*sensors_crd).*(1-
detect(:,i))))/K;
            if index_rel==1
                SP_calc=[SP_calc_inst(index_rel,1)
SP_calc_inst(index_rel,2)];
            else
                SP_calc=[((index_rel-
1)*SP_calc(1)+SP_calc_inst(index_rel,1))/index_rel ((index_rel-
1)*SP_calc(2)+SP_calc_inst(index_rel,2))/index_rel];
            end
            for count_HS=1:HS
                dist_SP_HS(count_HS)=sqrt((SP_calc(1)-
hotspot_crd(count_HS,1))^2+(SP_calc(2)-hotspot_crd(count_HS,2))^2);
            end
            SP_num=find(dist_SP_HS==min(dist_SP_HS));
            SP_calc_db(i,:)=SP_calc;
            SP_num_db(1:length(SP_num),i)=transpose(SP_num);
        else
            if sum(sum(SP_calc_inst))==0
                SP_calc_db(i,:)=NaN;
            else
                SP_calc_db(i,:)=SP_calc_db(i-1,:);
            end
        end
        if final_result(i)==1 && result(i)==0
            SP_calc_db(i,:)=SP_calc;
            SP_num_db(1:length(SP_num),i)=transpose(SP_num);
        end
end

fa_test=find((X==0)&(final_result==1));
```

```
FP_test=length(fa_test);    % number of false alarms FP
Q_F_test=FP_test/non_rel; % FPR

d_test=find((X==1)&(final_result==1));
TP_test=length(d_test); % number of detection (true positives)
Q_D_test=TP_test/rel; % TPR - True Positive Rate - Probability of Detection

precision_test=TP_test/(TP_test+FP_test); % precision

%% Animation

answer = questdlg('Do you want to see the animation?','Yes','No');
switch answer
    case 'Yes'
        answer = 1;
    case 'No'
        answer = 0;
end
if answer==1
sentence='Show Animation';
disp(sentence)
img=imread('template.png');
figure(1)
for i=1:tot
hold off
imagesc([0 41.26],[0 27.45],flipud(img));
set(gca,'ydir','normal');
set(gcf, 'Position', get(0, 'Screensize'));
xlim([0 41.26])
ylim([0 27.45])
xlabel('x (meters)')
ylabel('y (meters)')
grid on
hold on
scatter(hotspot_crd(:,1),hotspot_crd(:,2),400,'filled','r')
if X(i)==1
    scatter(target_crd(1),target_crd(2),400,'filled','b')
end
scatter(sensors_crd(:,1),sensors_crd(:,2),400,'filled','g')
    if isnan(SP_calc_db(i,1))==0
        scatter(SP_calc_db(i,1),SP_calc_db(i,2),100,'filled','m')
    end
    if isnan(SP_num_db(1,i))==0 && isnan(SP_num_db(2,i))==1

scatter(hotspot_crd(SP_num_db(1,i),1),hotspot_crd(SP_num_db(1,i),2),100,'fi
lled','y')
    else
        if isnan(SP_num_db(1,i))==0 && isnan(SP_num_db(2,i))==0

scatter(hotspot_crd(SP_num_db(1:size(SP_num_db,1),i),1),hotspot_crd(SP_num_
db(1:size(SP_num_db,1),i),2),100,'filled','y')
        end
    end
title(['2D Scenario. Instant number: ' num2str(i) ' of ' num2str(tot)])
drawnow
end
end

%% Plot

disp('Plot and curves generation')
```

```
figure(1) % release plot
plot(X)
xlabel('time (s)')
ylabel('0 = Non-release / 1 = Release')
xlim([0 tot])
ylim([0 1])

figure(2) % Square attenuation vs. Distance
x_plot=l0:.1:max(l);
x_plot=[x_plot max(l)];
TL_plot=alpha_water*(x_plot-l0)*10^-3+spread_cf*10*log10(x_plot/l0);
AAF_plot=exp(-TL_plot./(20*log10(exp(1))));
plot(x_plot,AAF_plot,'b')
xlabel('Distance (m)')
ylabel('Attenuation (AAF)')
for i=1:K
hold on
line([l(i) l(i)],[0 1]) % vertical lines representing sensors position
end
hold off
xlim([l0-1 max(l)+1])
ylim([0 max(AAF_plot)])
title('Attenuation vs Distance')

figure(3) % wave amplitude vs time plot
for i=1:K
    z0=ampl(i,:);
    z1=ampl(i,:);
    for j=1:tot
        switch X(j)
            case 0
                z1(j)=NaN;
            case 1
                z0(j)=NaN;
        end
    end
    subplot(K,1,i)
    plot(z0,'b')
    hold on
    plot(z1,'r')
    xlabel('time (s)')
    ylabel('y (Amplitude)')
    title(['Received Signal, sensor ' num2str(i) ', coordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
    hold off
end

figure(4) % Amplitude Distribution Plot
for i=1:K
subplot(1,K,i)
histogram(noise_ampl(i,:),'Normalization','pdf')
hold on
histogram(rel_ampl(i,:),'Normalization','pdf')
hold on
line([sqrt(th_loc(i))
sqrt(th_loc(i))],get(gca,'ylim'),'LineWidth',2,'Color','k')
line([-sqrt(th_loc(i)) -
sqrt(th_loc(i))],get(gca,'ylim'),'LineWidth',2,'Color','k')
legend('p(y|H0)','p(y|H1)')
```

```
title(['Amplitude pdf, sensor ' num2str(i) ', coordordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
xlabel('y (Amplitude)')
hold off
end

figure(5)
for i=1:K
    subplot(1,K,i)
    th_loc_poss=0:.01:10;

plot(2*qfunc(sqrt(th_loc_poss/var_noise)),2*qfunc(sqrt(th_loc_poss/(var_fad
sig*AAF_FC(target,i)^2+var_noise))))
    hold on

scatter(2*qfunc(sqrt(th_loc(i)/var_noise)),2*qfunc(sqrt(th_loc(i)/(var_fads
ig*AAF_FC(target,i)^2+var_noise))),'filled')
    line([0 1],[0 1],'LineStyle','--')
    hold off
    title(['ROC sensor ' num2str(i) ', coordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
    xlabel('Local Probability of False Alarm')
    ylabel('Local Probability of Detection')
    xlim([0 1])
    ylim([0 1])
    pbaspect([1 1 1])
end

figure(6) % ROC curve - linear
plot(Q_F,Q_D,'-or')
hold on
line([0 1],[0 1],'LineStyle','--') % chance line
scatter(Q_F_test,Q_D_test,'filled')
hold off
title('ROC CURVE - linear')
xlabel('Global Probability of False Alarm')
ylabel('Global Probability of Detection')
xlim([0 1])
ylim([0 1])

figure(7) % ROC curve - semilog
semilogx(Q_F,Q_D,'-or')
hold on
scatter(Q_F_test,Q_D_test,'filled')
hold off
title('ROC CURVE - semilog')
xlabel('Global Probability of False Alarm')
ylabel('Global Probability of Detection')
xlim([Q_F(end-1) 1])
ylim([Q_D(end-1)-0.05 1])

figure(8) % PR curve - linear
plot(Q_D,precision,'-or')
hold on
scatter(Q_D_test,precision_test,'filled')
hold off
title('PR CURVE')
xlabel('Recall')
ylabel('Precision')
xlim([0 1])
ylim([0 1])
```

```
SP_num_db_plot=zeros(1,index_rel);
dist_plot=zeros(1,index_rel);
j=1;
for i=1:N
    if result(i)==1
        SP_num_db_plot(j)=SP_num_db(1,i);
        dist_plot(j)=sqrt((target_crd(1)-
hotspot_crd(SP_num_db_plot(j),1))^2+(target_crd(2)-
hotspot_crd(SP_num_db_plot(j),2))^2);
        j=j+1;
    end
end
AUC_localization=trapz(1:index_rel,dist_plot);
figure(9)
plot(1:index_rel,dist_plot)
title(['Distance from Target. Area under curve (no. of instants: '
num2str(index_rel) ') = ' num2str(AUC_localization) ''])
xlabel('Time (s) / Fusion Center Positive Detection')
ylabel('Distance from Target (m)')
xlim([1 index_rel])
ylim([0 max(dist_plot)])

%% Memory Cleaning

disp('Memory Cleaning')
clearvars x_plot TL_plot AAF_plot x d fa j i q s sentence img u v
noise_ampl_FC rel_ampl_FC ind th_loc_poss pfa_FC_poss pd_FC_poss
fa_test_loc FP_test_loc d_test_loc TP_test_loc % deleting variables

%% Functions

function x=binary_source(p,r,N)

A10=(1-p)/(p+r-p*r);
A01=p/(p+r-p*r);

x=zeros(1,N); % pre-allocation of memory to improve code performances

for n=2:N % it starts from 2 because we want the initial state n=1 to be
without release
    switch x(n-1)
        case 0
            x(n)=(rand<A01);
        case 1
            x(n)=(rand<1-A10);
    end
end
end

function y=speed_sound(D,T,S)

% it calculates sound speed using corrected UNESCO algorithm

c0=1402.388+5.03711*T-5.80852e-2*T^2+3.3420e-4*T^3-1.47800e-6*T^4+3.1464e-
9*T^5;
c1=0.153563+6.8982e-4*T-8.1788e-6*T^2+1.3621e-7*T^3-6.1185e-10*T^4;
c2=3.1260e-5-1.7107e-6*T+2.5974e-8*T^2-2.5335e-10*T^3+1.0405e-12*T^4;
c3=-9.7729e-9-3.8504e-10*T-2.3643e-12*T^2;
A0=1.389-1.262e-2*T+7.164e-5*T^2+2.006e-6*T^3-3.21e-8*T^4;
A1=9.4742e-5-1.2580e-5*T-6.4885e-8*T^2+1.0507e-8*T^3-2.0122e-10*T^4;
```

```
A2=-3.9064e-7+9.1041e-9*T-1.6002e-10*T^2+7.988e-12*T^3;
A3=1.100e-10+6.649e-12*T-3.389e-13*T^2;
P=D/10;
A=A0+A1*P+A2*P^2+A3*P^3;
B=-1.922e-2-4.42e-5*T+(7.3637e-3+1.7945e-7*T)*P;
C=-7.9836e-6*P+1.727e-3;
y=c0+c1*P+c2*P^2+c3*P^3+A*S+B*S^(3/2)+C*S^2;
end

function y=absorption_FG(f,T,S,D,pH,c)

% this function calculates the attenuation coefficient in bB/km
% using Francois-Garrison method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m
% "pH" is the indicator of water acidity

theta=273.15+T; % K, temperature

% excess absorption due to Boric Acid (H3BO3)
A1=(8.86/c)*10^(0.78*pH-5); % dB/(km*kHz)
P1=1; % nondimensional pressure correction factor
f1=2.8*sqrt(S/35)*10^(4-1245/theta); % kHz, relaxation frequencies of H3BO3
alpha1=(A1*P1*f1*f.^2)/(f1^2+f.^2); % dB/km

% excess absorption due to Magnesium Sulfate (MgSO4)
A2=21.44*(S/c)*(1+0.025*T); % dB/(km*kHz)
P2=1-1.37e-4*D+6.2e-9*D^2; % nondimensional pressure correction factor
f2=(8.17*10^(8-1990/theta))/(1+0.0018*(S-35)); % kHz, relaxation
frequencies of MgSO4
alpha2=(A2*P2*f2*f.^2)/(f2^2+f.^2); % dB/km

% absorption due to water
if T<=20
    A3=4.937e-4-2.59e-5*T+9.11e-7*T^2-1.5e-8*T^3; % dB/(km*kHz^2)
else
    A3=3.964e-4-1.146e-5*T+1.45e-7*T^2-6.5e-10*T^3; % dB/(km*kHz^2)
end
P3=1-3.83e-5*D+4.9e-10*D^2; % nondimensional pressure correction factor
alpha3=A3*P3*f.^2; % dB/km

y=alpha1+alpha2+alpha3;
end
function y=absorption_FS(f,T,D)

% this function calculates the attenuation coefficient in bB/km
% using Fisher-Simmons method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "D" is the depth in m

f=f*1000; % method is written using Hz
theta=273.15+T; % K, temperature
P=D/10+1; % atm, absolute pressure estimation (supposing 10 m_seawater = 1
atm)

% excess absorption due to Boric Acid (H3BO3)
A1=1.03e-8+2.36e-10*T-5.22e-12*T^2; % Np/(m*Hz)
P1=1; % nondimensional pressure correction factor
```

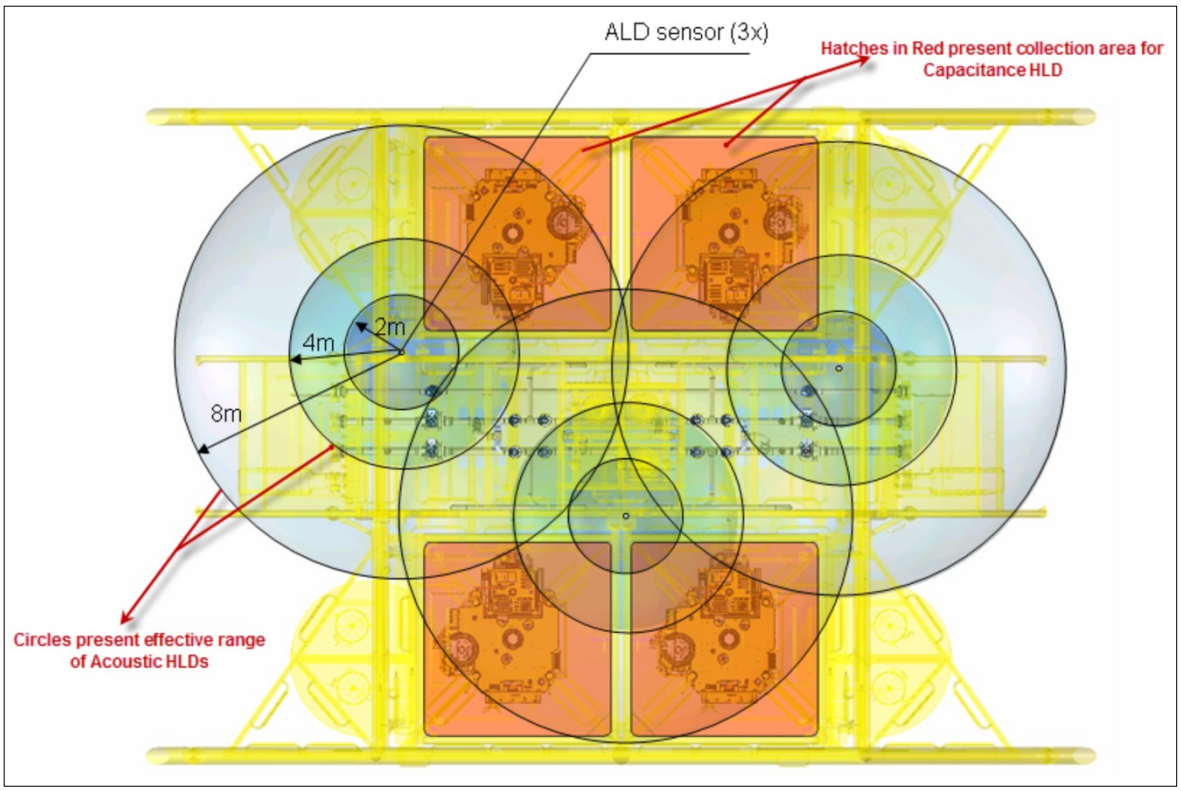
```
f1=1.32e3*theta*exp(-1700/theta); % Hz, relaxation frequencies of H3BO3
alpha1=(A1*P1*f1*f.^2)/(f1^2+f.^2); % Np/m

% excess absorption due to Magnesium Sulfate (MgSO4)
A2=5.62e-8+7.52e-10*T; % Np/(m*Hz)
P2=1-10.3e-4*P+3.7e-7*P^2; % nondimensional pressure correction factor
f2=1.55e7*theta*exp(-3052/theta); % Hz, relaxation frequencies of MgSO4
alpha2=(A2*P2*f2*f.^2)/(f2^2+f.^2); % Np/m

% absorption due to water
A3=(55.9-2.37*T+4.77e-2*T^2-3.48e-4*T^3)*10^-15; % Np/(m*Hz^2)
P3=1-384e-4*P+7.57e-8*P^2; % nondimensional pressure correction factor
alpha3=A3*P3*f.^2; % Np/m

y=alpha1+alpha2+alpha3; % Np/m
y=y*1000*20*log10(exp(1)); % dB/km
end
function y=absorption_SM(f,T,S,D)

% this function calculates the attenuation coefficient in bB/km
% using Shulkin-Marsh method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m

theta=273.15+T; % K, temperature
P=D/10+1; % atm, absolute pressure estimation (supposing 10 m_seawater = 1
atm)
A=2.34e-6;
B=3.38e-6;
fT=21.9*10^(6-1520/theta); % kHz
y=((S*A*fT*f.^2)/(fT^2+f.^2)+(B*f.^2)/fT)*(1-6.54e-4*P); % Np/m
y=y*1000*20*log10(exp(1)); % dB/km
end
function y=absorption_AM(f,T,S,D,pH)

% this function calculates the attenuation coefficient in bB/km
% using Ainslie-McColm method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m
% "pH" is the indicator of water acidity

D=D/1000; % method is written using km

% excess absorption due to Boric Acid (H3BO3)
f1=0.78*exp(T/26)*sqrt(S/35); % kHz, relaxation frequencies of H3BO3
alpha1=0.106*(f1*f.^2)/(f1^2+f.^2)*exp((pH-8)/0.56); % dB/km

% excess absorption due to Magnesium Sulfate (MgSO4)
f2=42*exp(T/17); % kHz, relaxation frequencies of MgSO4
alpha2=0.52*(1+T/43)*(S/35)*(f2*f.^2)/(f2^2+f.^2)*exp(-D/6); % dB/km

% absorption due to water
alpha3=4.9e-4*f.^2*exp(-T/27+D/17); % dB/km

y=alpha1+alpha2+alpha3;
end
function y=absorption_T(f)
```

```
% this function calculates the attenuation coefficient in bB/km
% using Thorp method
% "f" is the frequency of the sound in kHz

y=1.0936132983*(0.1*f.^2/(1+f.^2)+40*f.^2/(4100+f.^2));
end
```

Appendix 3: MATLAB Script (Hotspot 10: WFR + BBM-TISP)

This MATLAB script was written using MATLAB R2019; it simulates the release in HS 10, the FC performs the WFR for detection and BBM-TISP for localization. All the settings are those described in the previous chapters.

MATLAB, in order to run the animation, must have the following picture in a folder accessible by MATLAB. The file must be named "template.png".




```
%%%%%  Oil Spill Detection and Localization    %%%%
%%%%%               WFR + BBM-TISP             %%%%
%%%%%                 Hotspot 10               %%%%
%%%%%         Author: Gianluca Tabella         %%%%

clear
close all
clc

%% Release Modelling

disp('Release Modelling') % display the progress

p=1/3;  % parameter for release shape
r=2;    % parameter for release shape
N=10000; % number of instants
X=binary_source(p,r,N); % function giving an array that simulates the
events before the continuous release
tot=length(X); % total number of records
rel=nnz(X); % number of releases (positive events)
non_rel=tot-rel; % number of non-releases (negative events)

%% Signal Modelling

disp('Signal Modelling') % display the progress
```

```
f=20e-3; % kHz - sound frequency
T=3.8; % ∞C - temperature (T=3.8∞C at Goliat)
S=35; % ppt - salinity (S=35 ppt at Goliat)
D=350; % m - depth (D=350-400 m at Goliat)
pH=8; % pH of water (pH=8 at Goliat)
spread_cf=1.5; % spreading coeff. (spherical=2;cylindrical=1;practical=1.5)

c=speed_sound(D,T,S); % m/s - speed of sound underwater
alpha_water=absorption_FG(f,T,S,D,pH,c); % dB/km - absorption due to water
hotspot_crd=[11.5,11.75 ; 11.5,12.75 ; 11.5,13.75 ; 15,11.75 ; 15,12.75 ;
15,13.75 ; 18,11.75 ; 18,12.75 ; 19,11.75 ; 19,12.75 ; 24.5,11.75 ;
24.5,12.75 ; 25.5,11.75 ; 25.5,12.75 ; 28.2,11.75 ; 28.2,12.75 ; 28.2,13.75
; 31.7,11.75 ; 31.7,12.75; 31.7,13.75]; % m - hotspots' coordinates
HS=length(hotspot_crd(:,1)); % number of hot spots
target=10;
target_crd=hotspot_crd(target,:); % m - target coordinates
sensors_crd=[13.97,15.3 ; 29.35,14.66 ; 21.86,9.42]; % m - sensor
coordinates
l=transpose(sqrt((target_crd(1)-sensors_crd(:,1)).^2+(target_crd(2)-
sensors_crd(:,2)).^2)); % m - distance of sensors from target
K=length(l); % number of sensors
l_FC=zeros(length(hotspot_crd(:,1)),K);
for i=1:length(hotspot_crd(:,1))  % m - (HSxK matrix) distances between HS
hot spots and K sensors
    for j=1:K
        l_FC(i,j)=sqrt((hotspot_crd(i,1)-
sensors_crd(j,1))^2+(hotspot_crd(i,2)-sensors_crd(j,2))^2);
    end
end
l0=min(min(l_FC)); % m - reference distance
TL=alpha_water*(l-l0)*10^-3+spread_cf*10*log10(l/l0); % dB - Transmission
Loss
AAF=exp(-TL./(20*log10(exp(1)))); % AAF (TL converted in Nepers)
var_noise=1; % variance of noise (gaussian)
SNR=20; % signal-to-noise ratio at closest (sensor/hot spot)
var_fadsig=var_noise*SNR; % variance of fading (gaussian)

noise_ampl=zeros(K,non_rel);
for i=1:K
    noise_ampl(i,:)=sqrt(var_noise)*randn(1,non_rel);
end
rel_ampl=zeros(K,rel);
for i=1:K

rel_ampl(i,:)=AAF(i)*sqrt(var_fadsig)*randn(1,rel)+sqrt(var_noise)*randn(1,
rel);
end
u=1;
v=1;
ampl=zeros(K,tot); % pre-allocation for final scenario
for i=1:tot
    switch X(i)
        case 1
            for j=1:K
                ampl(j,i)=rel_ampl(j,u);
            end
            u=u+1;
        case 0
            for j=1:K
                ampl(j,i)=noise_ampl(j,v);
```

```
            end
            v=v+1;
    end
end

%% Sensors' Settings

disp('Sensors setting calculation') % display the progress

th_loc_poss=0:.00001:10; % threshold for local decision (non-negative)

f_FC=f; % kHz - sound frequency
T_FC=T; % ∞C - temperature (T=3.8∞C at Goliat)
S_FC=S; % ppt - salinity (S=35 ppt at Goliat)
D_FC=D; % m - depth (D=350-400 m at Goliat)
pH_FC=pH; % pH of water (pH=8 at Goliat)
spread_cf_FC=spread_cf; % spreading coeff.
(spherical=2;cylindrical=1;practical=1.5)
c_FC=speed_sound(D_FC,T_FC,S_FC); % m/s - speed of sound underwater
alpha_water_FC=absorption_FG(f_FC,T_FC,S_FC,D_FC,pH_FC,c_FC); % dB/km -
absorption due to water
TL_FC=alpha_water_FC*(l_FC-l0)*10^-3+spread_cf_FC*10*log10(l_FC/l0); % dB -
Transmission Loss
AAF_FC=exp(-TL_FC/(20*log10(exp(1)))); % AAF (TL converted in Nepers)
var_noise_FC=1; % variance of noise (gaussian)
SNR_FC=20; % signal-to-noise ratio at closest (sensor/hot spot)
var_fadsig_FC=var_noise_FC*SNR_FC; % variance of fading (gaussian)
CZ_loc=zeros(HS,K);
CZ_loc_db=zeros(length(th_loc_poss),K);
th_loc=zeros(1,K);
for i=1:length(th_loc_poss)
    pfa_FC_poss=2*qfunc(sqrt(th_loc_poss(i)/var_noise_FC)); % array of
probabilities of false alarm for the single sensors

pd_FC_poss=2*qfunc(sqrt(th_loc_poss(i)./(var_fadsig_FC*AAF_FC.^2+var_noise_
FC))); % array of the corresponing probabilities of detection
    CZ_loc=pd_FC_poss.*(1-pfa_FC_poss);
    CZ_loc_db(i,:)=mean(CZ_loc);
end

for i=1:K
    th_loc(i)=th_loc_poss(CZ_loc_db(:,i)==max(CZ_loc_db(:,i)));
end

pfa_FC=2*qfunc(sqrt(th_loc/var_noise_FC)); % array of probabilities of
false alarm for the single sensors
pd_FC=zeros(HS,K);
for i=1:K
pd_FC(:,i)=2*qfunc(sqrt(th_loc(i)./(var_fadsig_FC*AAF_FC(:,i).^2+var_noise_
FC))); % array of the corresponing probabilities of detection
end
pd_FC=mean(pd_FC);

%% FC's Settings

disp('FC Settings Calculation') % display the progress

p_FC=p;  % parameter for release shape
r_FC=r;    % parameter for release shape
N_FC=N; % number of instants
```

```
X_FC=binary_source(p_FC,r_FC,N_FC); % function giving an array that
simulates the events before the continuous release
tot_FC=length(X_FC); % total number of records
rel_FC=nnz(X_FC); % number of releases (positive events)
non_rel_FC=tot_FC-rel_FC; % number of non-releases (negative events)
noise_ampl_FC=zeros(K,non_rel_FC);

for i=1:K
    noise_ampl_FC(i,:)=sqrt(var_noise_FC)*randn(1,non_rel_FC);
end
rel_ampl_FC=zeros(K,rel_FC);
for i=1:K

rel_ampl_FC(i,:)=mean(AAF_FC(:,i))*sqrt(var_fadsig_FC)*randn(1,rel_FC)+sqrt
(var_noise_FC)*randn(1,rel_FC);
end
u=1;
v=1;
ampl_FC=zeros(K,tot_FC); % pre-allocation for final scenario
for i=1:tot_FC
    switch X_FC(i)
        case 1
            for j=1:K
                ampl_FC(j,i)=rel_ampl_FC(j,u);
            end
            u=u+1;
        case 0
            for j=1:K
                ampl_FC(j,i)=noise_ampl_FC(j,v);
            end
            v=v+1;
    end
end

detect_FC=zeros(K,tot_FC); % matrix containing all the decisions of any
sensor during the simulation (every row is a sensor)
for i=1:K
    for j=1:tot_FC
        if ampl_FC(i,j)^2>=th_loc(i)
            detect_FC(i,j)=1;
        else
            detect_FC(i,j)=0;
        end
    end
end

th=-4:0.001:4; % array of possibile thresholds
no_thr=length(th); % number of possible thresholds

result_FC=zeros(no_thr,tot_FC); % matrix containing all the decisions of
the FC for different thresholds
lambda_FC=zeros(1,tot_FC); % vector containing all the FR's results

for j=1:no_thr % for each threshold
    sentence=['FC setting calculation: elaboration threshold number '
,num2str(j), ' of ' ,num2str(no_thr)];
    disp(sentence)
    for i=1:tot_FC % for each instant

lambda_FC(i)=sum(transpose(detect_FC(:,i)).*log(pd_FC./pfa_FC)+transpose((1
-detect_FC(:,i))).*log((1-pd_FC)./(1-pfa_FC))); % value of FR
```

```
        if lambda_FC(i)>th(j) % for-loop determining the decision of the FC
based on WFR
            result_FC(j,i)=1;
        else
            result_FC(j,i)=0;
        end
    end
end

%% FC's Performances

FP=zeros(1,no_thr); % array of number of false alarms at different
thresholds
TP=zeros(1,no_thr); % array of number of detections at different thresholds
FN=zeros(1,no_thr); % array of number of false negatives at different
thresholds
TN=zeros(1,no_thr); % array of number of true negatives at different
thresholds
Q_F=zeros(1,no_thr); % array of probabilities of false alarm at different
thresholds
Q_D=zeros(1,no_thr); % array of probabilities of detecion at different
thresholds
precision=zeros(1,no_thr); % array of precisions at different thresholds

for i=1:no_thr % for each threshold
        sentence=['FC setting calculation: evaluation threshold number '
,num2str(i), ' of ' ,num2str(no_thr)];
        disp(sentence) % display the progress

        fa=find((X_FC==0)&(result_FC(i,:)==1));
        FP(i)=length(fa);    % number of false alarms FP
        Q_F(i)=FP(i)/non_rel_FC; % FPR

        d=find((X_FC==1)&(result_FC(i,:)==1));
        TP(i)=length(d); % number of detection (true positives)
        Q_D(i)=TP(i)/rel_FC; % TPR - True Positive Rate - Probability of
Detection

        FN(i)=rel_FC-TP(i); % False negatives
        TN(i)=non_rel_FC-FP(i); % True Negatives

        precision(i)=TP(i)/(TP(i)+FP(i)); % precision
end
precision(isnan(precision))=1; % built to overcome the presence of NaN
results when precision converges to 1

%% Removal of Useless Thresholds

disp('Removal of Useless Thresholds from FC setting calculation history')
j=2; % loop used to remove results when no change was determined varying
the threshold
result_FC_new(1,:)=result_FC(1,:);
Q_D_new(1)=Q_D(1);
Q_F_new(1)=Q_F(1);
precision_new(1)=precision(1);
th_new(1)=th(1);
TP_new(1)=TP(1);
FP_new(1)=FP(1);
FN_new(1)=FN(1);
TN_new(1)=TN(1);
for i=2:no_thr
```

```
    if Q_D(i)==Q_D(i-1) && Q_F(i)==Q_F(i-1) && precision(i)==precision(i-1)
    else
        Q_D_new(j)=Q_D(i);
        Q_F_new(j)=Q_F(i);
        precision_new(j)=precision(i);
        th_new(j)=th(i);
        TP_new(j)=TP(i);
        FP_new(j)=FP(i);
        FN_new(j)=FN(i);
        TN_new(j)=TN(i);
        result_FC_new(j,:)=result_FC(i,:);
        j=j+1;
    end
end
Q_D=Q_D_new;
Q_F=Q_F_new;
precision=precision_new;
th=th_new;
TP=TP_new;
FP=FP_new;
FN=FN_new;
TN=TN_new;
result_FC=result_FC_new;
no_thr=length(th);
clearvars Q_D_new Q_F_new precision_new th_new TP_new FP_new FN_new TN_new
result_FC_new

%% Analysis of Parameters

disp('Analysis of Parameters for FC setting')

AUC=-trapz(Q_F,Q_D); % calculation of Area Under the Curve

% the matrix called "perf" has rows representing each indicator
% for each raw of the matrix the 1st value is the cut-off threshold at
indicator¥s maximum value
% the 2nd value is the Prob. of Detection at cut off threshold
% the 3rd value is the Prob. of False Alarm at cut off threshold
% the 4th value is the indicator¥s maximum value

J=Q_D-Q_F; % calculation of Youden's indexes
perf(1,4)=max(J); % maximum value of Youden's indexes
ind=find(J==perf(1,4)); % array's index for max Youden's index
perf(1,1)=th(ind); % cut off threshold at max J value
perf(1,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(1,2)=Q_D(ind); % Prob. of Detection at cut off threshold

beta_value=1; % 1: pr and rec equally weighted, 2: recall weighted higher
than precision, 0.5: precision weighted higher than recall
F_beta=(1+beta_value^2)*(precision.*Q_D)./(beta_value^2*precision+Q_D); %
F_value (armonic mean between precision and recall, with weight)
perf(2,4)=max(F_beta); % maximum value of F_beta
ind=find(F_beta==perf(2,4)); % array's index for max F_beta
perf(2,1)=th(ind); % cut off threshold at max F_beta
perf(2,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(2,2)=Q_D(ind); % Prob. of Detection at cut off threshold

dist_roc=(1-Q_D).^2+Q_F.^2; % sq. distance to (0,1)
perf(3,4)=min(dist_roc); % minimum sq. distance to (0,1)
ind=find(dist_roc==perf(3,4)); % array's index for min sq. distance to
(0,1)
```

```
perf(3,1)=th(ind); % cut off threshold at min sq. distance to (0,1)
perf(3,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(3,2)=Q_D(ind); % Prob. of Detection at cut off threshold

CZ=Q_D.*(1-Q_F); % CZ index
perf(4,4)=max(CZ); % maximum value of CZ index
ind=find(CZ==perf(4,4)); % array's index for max CZ index
perf(4,1)=th(ind); % cut off threshold at max CZ index
perf(4,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(4,2)=Q_D(ind); % Prob. of Detection at cut off threshold

%% Sensors' Detection

detect=zeros(K,tot); % matrix containing all the decisions of any sensor
during the simulation (every row is a sensor)
for i=1:K
    sentence=['simulating detection: sensor ' ,num2str(i), ' of '
,num2str(K)]; % display the progress
    disp(sentence)
    for j=1:tot
        if ampl(i,j)^2>=th_loc(i)
            detect(i,j)=1;
        else
            detect(i,j)=0;
        end
    end
end

Q_F_test_loc=zeros(1,K);
Q_D_test_loc=zeros(1,K);

for i=1:K
    fa_test_loc=find((X==0)&(detect(i,:)==1));
    FP_test_loc=length(fa_test_loc);    % number of false alarms FP
    Q_F_test_loc(i)=FP_test_loc/non_rel; % FPR

    d_test_loc=find((X==1)&(detect(i,:)==1));
    TP_test_loc=length(d_test_loc); % number of detection (true positives)
    Q_D_test_loc(i)=TP_test_loc/rel; % TPR - True Positive Rate -
Probability of Detection
end

%% Fusion Center's Detection and Localization

inst_FC=1; % number of instants to take into account in the TCR
time_th=0; % threshold used in TCR
SP_calc_inst=zeros(tot,2);
index_rel=0;
SP_calc_db=zeros(tot,2);
SP_num_db(1:2,1:tot)=NaN;
SP_calc=[0 0];

dist_SP_HS=zeros(1,HS);
result=zeros(1,tot); % matrix containing all the decisions of the FC for
different thresholds
lambda=zeros(1,tot); % vector containing all the FR's results
lambda_time=zeros(1,tot); % vector containing all the TCR's results
final_result=zeros(1,tot); % matrix containing final decision (WFR+TCR)

for i=1:tot % for each instant
        if rem(i,100)==0 || i==1 % display the progress
```

```
        sentence=['Instant simulated by FC: ' ,num2str(i), ' of '
,num2str(tot)];
        disp(sentence)
        end

lambda(i)=sum(transpose(detect(:,i)).*log(pd_FC./pfa_FC)+transpose((1-
detect(:,i))).*log((1-pd_FC)./(1-pfa_FC))); % value of FR
        if lambda(i)>perf(1,1) % for-loop determining the decision of the
FC based on WFR
            result(i)=1;
        else
            result(i)=0;
        end

        if result(i)==0
            inst=inst_FC;
            if inst>i % necessary for the first instants (when there aren't
enough previous data)
                inst=i;
            end
            lambda_time(i)=sum(result(i-inst+1:i));
            if lambda_time(i)>time_th % for-loop determining the final
decision of the FC
                final_result(i)=1;
            else
                final_result(i)=0;
            end
            else
             final_result(i)=1;
        end

        if result(i)==1 && sum(detect(:,i))>=1
            index_rel=index_rel+1;

SP_calc_inst(index_rel,:)=sum((repmat(transpose(pd_FC./pfa_FC),1,size(2,2))
).*sensors_crd.*detect(:,i))/sum((repmat(transpose(pd_FC./pfa_FC),1,size(2,
2))).*detect(:,i));
            if index_rel==1
                SP_calc=[SP_calc_inst(index_rel,1)
SP_calc_inst(index_rel,2)];
            else
                SP_calc=[((index_rel-
1)*SP_calc(1)+SP_calc_inst(index_rel,1))/index_rel ((index_rel-
1)*SP_calc(2)+SP_calc_inst(index_rel,2))/index_rel];
            end
            for count_HS=1:HS
                dist_SP_HS(count_HS)=sqrt((SP_calc(1)-
hotspot_crd(count_HS,1))^2+(SP_calc(2)-hotspot_crd(count_HS,2))^2);
            end
            SP_num=find(dist_SP_HS==min(dist_SP_HS));
            SP_calc_db(i,:)=SP_calc;
            SP_num_db(1:length(SP_num),i)=transpose(SP_num);
        else
            if sum(sum(SP_calc_inst))==0
                SP_calc_db(i,:)=NaN;
            else
                SP_calc_db(i,:)=SP_calc_db(i-1,:);
            end
        end
        if final_result(i)==1 && result(i)==0
            SP_calc_db(i,:)=SP_calc;
```

```
            SP_num_db(1:length(SP_num),i)=transpose(SP_num);
        end
end

fa_test=find((X==0)&(final_result==1));
FP_test=length(fa_test);    % number of false alarms FP
Q_F_test=FP_test/non_rel; % FPR

d_test=find((X==1)&(final_result==1));
TP_test=length(d_test); % number of detection (true positives)
Q_D_test=TP_test/rel; % TPR - True Positive Rate - Probability of Detection

precision_test=TP_test/(TP_test+FP_test); % precision

%% Animation

answer = questdlg('Do you want to see the animation?','Yes','No');
switch answer
    case 'Yes'
        answer = 1;
    case 'No'
        answer = 0;
end
if answer==1
sentence='Show Animation';
disp(sentence)
img=imread('template.png');
figure(1)
for i=1:tot
hold off
imagesc([0 41.26],[0 27.45],flipud(img));
set(gca,'ydir','normal');
set(gcf, 'Position', get(0, 'Screensize'));
xlim([0 41.26])
ylim([0 27.45])
xlabel('x (meters)')
ylabel('y (meters)')
grid on
hold on
scatter(hotspot_crd(:,1),hotspot_crd(:,2),400,'filled','r')
if X(i)==1
    scatter(target_crd(1),target_crd(2),400,'filled','b')
end
scatter(sensors_crd(:,1),sensors_crd(:,2),400,'filled','g')
    if isnan(SP_calc_db(i,1))==0
        scatter(SP_calc_db(i,1),SP_calc_db(i,2),100,'filled','m')
    end
    if isnan(SP_num_db(1,i))==0 && isnan(SP_num_db(2,i))==1

scatter(hotspot_crd(SP_num_db(1,i),1),hotspot_crd(SP_num_db(1,i),2),10,'fil
led','y')
    else
        if isnan(SP_num_db(1,i))==0 && isnan(SP_num_db(2,i))==0

scatter(hotspot_crd(SP_num_db(1:size(SP_num_db,1),i),1),hotspot_crd(SP_num_
db(1:size(SP_num_db,1),i),2),10,'filled','y')
        end
    end
title(['2D Scenario. Instant number: ' num2str(i) ' of ' num2str(tot)])
drawnow
end
```

```
end

%% Plot

disp('Plot and curves generation')

figure(1) % release plot
plot(X)
xlabel('time (s)')
ylabel('0 = Non-release / 1 = Release')
xlim([0 tot])
ylim([0 1])

figure(2) % Square attenuation vs. Distance
x_plot=l0:.1:max(l);
x_plot=[x_plot max(l)];
TL_plot=alpha_water*(x_plot-l0)*10^-3+spread_cf*10*log10(x_plot/l0);
AAF_plot=exp(-TL_plot./(20*log10(exp(1))));
plot(x_plot,AAF_plot,'b')
xlabel('Distance (m)')
ylabel('Attenuation (AAF)')
for i=1:K
hold on
line([l(i) l(i)],[0 1]) % vertical lines representing sensors position
end
hold off
xlim([l0-1 max(l)+1])
ylim([0 max(AAF_plot)])
title('Attenuation vs Distance')

figure(3) % wave amplitude vs time plot
for i=1:K
    z0=ampl(i,:);
    z1=ampl(i,:);
    for j=1:tot
        switch X(j)
            case 0
                z1(j)=NaN;
            case 1
                z0(j)=NaN;
        end
    end
    subplot(K,1,i)
    plot(z0,'b')
    hold on
    plot(z1,'r')
    xlabel('time (s)')
    ylabel('y (Amplitude)')
    title(['Received Signal, sensor ' num2str(i) ', coordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
    hold off
end

figure(4) % Amplitude Distribution Plot
for i=1:K
subplot(1,K,i)
histogram(noise_ampl(i,:),'Normalization','pdf')
hold on
histogram(rel_ampl(i,:),'Normalization','pdf')
hold on
```

```
line([sqrt(th_loc(i))
sqrt(th_loc(i))],get(gca,'ylim'),'LineWidth',2,'Color','k')
line([-sqrt(th_loc(i)) -
sqrt(th_loc(i))],get(gca,'ylim'),'LineWidth',2,'Color','k')
legend('p(y|H0)','p(y|H1)')
title(['Amplitude pdf, sensor ' num2str(i) ', coordordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
xlabel('y (Amplitude)')
hold off
end

figure(5)
for i=1:K
    subplot(1,K,i)
    th_loc_poss=0:.01:10;

plot(2*qfunc(sqrt(th_loc_poss/var_noise)),2*qfunc(sqrt(th_loc_poss/(var_fad
sig*AAF_FC(target,i)^2+var_noise))))
    hold on

scatter(2*qfunc(sqrt(th_loc(i)/var_noise)),2*qfunc(sqrt(th_loc(i)/(var_fads
ig*AAF_FC(target,i)^2+var_noise))),'filled')
    line([0 1],[0 1],'LineStyle','--')
    hold off
    title(['ROC sensor ' num2str(i) ', coordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
    xlabel('Local Probability of False Alarm')
    ylabel('Local Probability of Detection')
    xlim([0 1])
    ylim([0 1])
    pbaspect([1 1 1])
end

figure(6) % ROC curve - linear
plot(Q_F,Q_D,'-or')
hold on
line([0 1],[0 1],'LineStyle','--') % chance line
scatter(Q_F_test,Q_D_test,'filled')
hold off
title('ROC CURVE - linear')
xlabel('Global Probability of False Alarm')
ylabel('Global Probability of Detection')
xlim([0 1])
ylim([0 1])

figure(7) % ROC curve - semilog
semilogx(Q_F,Q_D,'-or')
hold on
scatter(Q_F_test,Q_D_test,'filled')
hold off
title('ROC CURVE - semilog')
xlabel('Global Probability of False Alarm')
ylabel('Global Probability of Detection')
xlim([Q_F(end-1) 1])
ylim([Q_D(end-1)-0.05 1])

figure(8) % PR curve - linear
plot(Q_D,precision,'-or')
hold on
scatter(Q_D_test,precision_test,'filled')
hold off
```

```
title('PR CURVE')
xlabel('Recall')
ylabel('Precision')
xlim([0 1])
ylim([0 1])

SP_num_db_plot=zeros(1,index_rel);
dist_plot=zeros(1,index_rel);
j=1;
for i=1:N
    if result(i)==1
        SP_num_db_plot(j)=SP_num_db(1,i);
        dist_plot(j)=sqrt((target_crd(1)-
hotspot_crd(SP_num_db_plot(j),1))^2+(target_crd(2)-
hotspot_crd(SP_num_db_plot(j),2))^2);
        j=j+1;
    end
end
AUC_localization=trapz(1:index_rel,dist_plot);
figure(9)
plot(1:index_rel,dist_plot)
title(['Distance from Target. Area under curve (no. of instants: '
num2str(index_rel) ') = ' num2str(AUC_localization) ''])
xlabel('Time (s) / Fusion Center Positive Detection')
ylabel('Distance from Target (m)')
xlim([1 index_rel])
ylim([0 max(dist_plot)])

%% Memory Cleaning

disp('Memory Cleaning')
clearvars x_plot TL_plot AAF_plot x d fa j i q s sentence img u v
noise_ampl_FC rel_ampl_FC ind th_loc_poss pfa_FC_poss pd_FC_poss
fa_test_loc FP_test_loc d_test_loc TP_test_loc % deleting variables

%% Functions

function x=binary_source(p,r,N)

A10=(1-p)/(p+r-p*r);
A01=p/(p+r-p*r);

x=zeros(1,N); % pre-allocation of memory to improve code performances

for n=2:N % it starts from 2 because we want the initial state n=1 to be
without release
    switch x(n-1)
        case 0
            x(n)=(rand<A01);
        case 1
            x(n)=(rand<1-A10);
    end
end
end

function y=speed_sound(D,T,S)

% it calculates sound speed using corrected UNESCO algorithm

c0=1402.388+5.03711*T-5.80852e-2*T^2+3.3420e-4*T^3-1.47800e-6*T^4+3.1464e-
9*T^5;
```

```
c1=0.153563+6.8982e-4*T-8.1788e-6*T^2+1.3621e-7*T^3-6.1185e-10*T^4;
c2=3.1260e-5-1.7107e-6*T+2.5974e-8*T^2-2.5335e-10*T^3+1.0405e-12*T^4;
c3=-9.7729e-9-3.8504e-10*T-2.3643e-12*T^2;
A0=1.389-1.262e-2*T+7.164e-5*T^2+2.006e-6*T^3-3.21e-8*T^4;
A1=9.4742e-5-1.2580e-5*T-6.4885e-8*T^2+1.0507e-8*T^3-2.0122e-10*T^4;
A2=-3.9064e-7+9.1041e-9*T-1.6002e-10*T^2+7.988e-12*T^3;
A3=1.100e-10+6.649e-12*T-3.389e-13*T^2;
P=D/10;
A=A0+A1*P+A2*P^2+A3*P^3;
B=-1.922e-2-4.42e-5*T+(7.3637e-3+1.7945e-7*T)*P;
C=-7.9836e-6*P+1.727e-3;
y=c0+c1*P+c2*P^2+c3*P^3+A*S+B*S^(3/2)+C*S^2;
end

function y=absorption_FG(f,T,S,D,pH,c)

% this function calculates the attenuation coefficient in bB/km
% using Francois-Garrison method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m
% "pH" is the indicator of water acidity

theta=273.15+T; % K, temperature

% excess absorption due to Boric Acid (H3BO3)
A1=(8.86/c)*10^(0.78*pH-5); % dB/(km*kHz)
P1=1; % nondimensional pressure correction factor
f1=2.8*sqrt(S/35)*10^(4-1245/theta); % kHz, relaxation frequencies of H3BO3
alpha1=(A1*P1*f1*f.^2)/(f1^2+f.^2); % dB/km

% excess absorption due to Magnesium Sulfate (MgSO4)
A2=21.44*(S/c)*(1+0.025*T); % dB/(km*kHz)
P2=1-1.37e-4*D+6.2e-9*D^2; % nondimensional pressure correction factor
f2=(8.17*10^(8-1990/theta))/(1+0.0018*(S-35)); % kHz, relaxation
frequencies of MgSO4
alpha2=(A2*P2*f2*f.^2)/(f2^2+f.^2); % dB/km

% absorption due to water
if T<=20
    A3=4.937e-4-2.59e-5*T+9.11e-7*T^2-1.5e-8*T^3; % dB/(km*kHz^2)
else
    A3=3.964e-4-1.146e-5*T+1.45e-7*T^2-6.5e-10*T^3; % dB/(km*kHz^2)
end
P3=1-3.83e-5*D+4.9e-10*D^2; % nondimensional pressure correction factor
alpha3=A3*P3*f.^2; % dB/km

y=alpha1+alpha2+alpha3;
end
function y=absorption_FS(f,T,D)

% this function calculates the attenuation coefficient in bB/km
% using Fisher-Simmons method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "D" is the depth in m

f=f*1000; % method is written using Hz
theta=273.15+T; % K, temperature
```

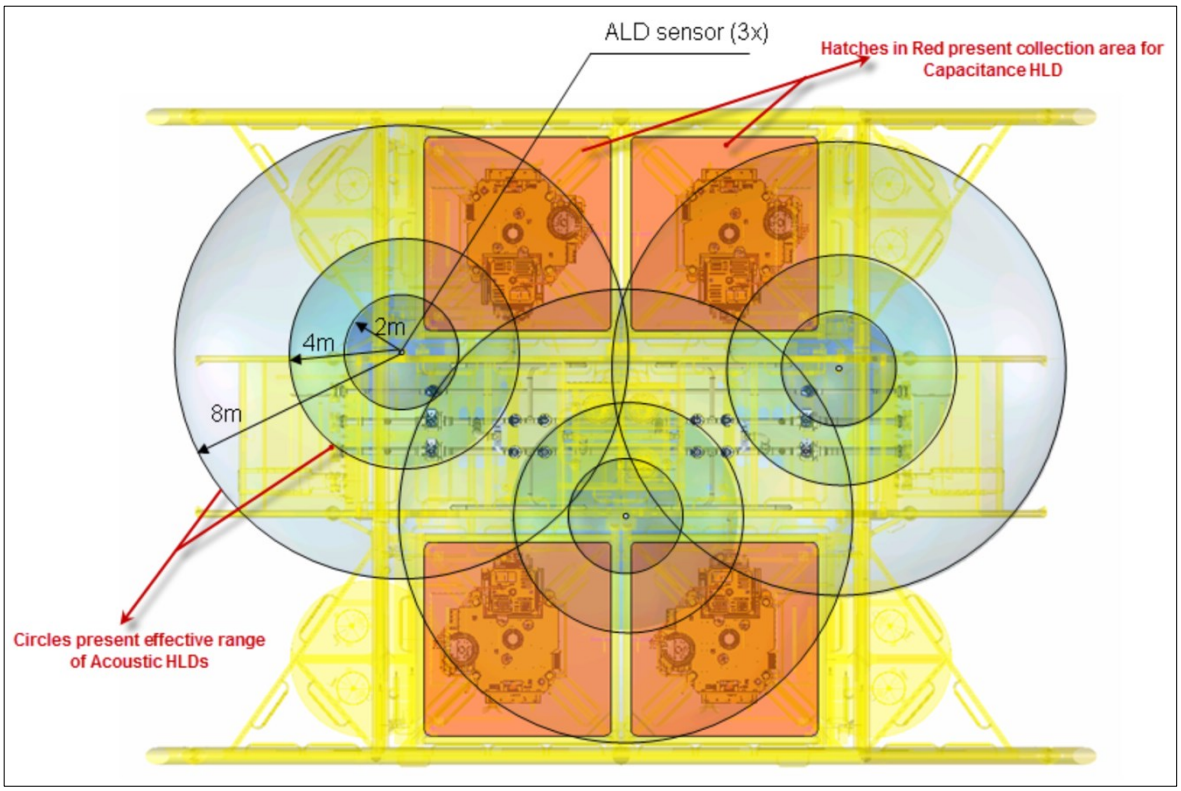
```
P=D/10+1; % atm, absolute pressure estimation (supposing 10 m_seawater = 1
atm)

% excess absorption due to Boric Acid (H3BO3)
A1=1.03e-8+2.36e-10*T-5.22e-12*T^2; % Np/(m*Hz)
P1=1; % nondimensional pressure correction factor
f1=1.32e3*theta*exp(-1700/theta); % Hz, relaxation frequencies of H3BO3
alpha1=(A1*P1*f1*f.^2)/(f1^2+f.^2); % Np/m

% excess absorption due to Magnesium Sulfate (MgSO4)
A2=5.62e-8+7.52e-10*T; % Np/(m*Hz)
P2=1-10.3e-4*P+3.7e-7*P^2; % nondimensional pressure correction factor
f2=1.55e7*theta*exp(-3052/theta); % Hz, relaxation frequencies of MgSO4
alpha2=(A2*P2*f2*f.^2)/(f2^2+f.^2); % Np/m

% absorption due to water
A3=(55.9-2.37*T+4.77e-2*T^2-3.48e-4*T^3)*10^-15; % Np/(m*Hz^2)
P3=1-384e-4*P+7.57e-8*P^2; % nondimensional pressure correction factor
alpha3=A3*P3*f.^2; % Np/m

y=alpha1+alpha2+alpha3; % Np/m
y=y*1000*20*log10(exp(1)); % dB/km
end
function y=absorption_SM(f,T,S,D)

% this function calculates the attenuation coefficient in bB/km
% using Shulkin-Marsh method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m

theta=273.15+T; % K, temperature
P=D/10+1; % atm, absolute pressure estimation (supposing 10 m_seawater = 1
atm)
A=2.34e-6;
B=3.38e-6;
fT=21.9*10^(6-1520/theta); % kHz
y=((S*A*fT*f.^2)/(fT^2+f.^2)+(B*f.^2)/fT)*(1-6.54e-4*P); % Np/m
y=y*1000*20*log10(exp(1)); % dB/km
end
function y=absorption_AM(f,T,S,D,pH)

% this function calculates the attenuation coefficient in bB/km
% using Ainslie-McColm method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m
% "pH" is the indicator of water acidity

D=D/1000; % method is written using km

% excess absorption due to Boric Acid (H3BO3)
f1=0.78*exp(T/26)*sqrt(S/35); % kHz, relaxation frequencies of H3BO3
alpha1=0.106*(f1*f.^2)/(f1^2+f.^2)*exp((pH-8)/0.56); % dB/km

% excess absorption due to Magnesium Sulfate (MgSO4)
f2=42*exp(T/17); % kHz, relaxation frequencies of MgSO4
alpha2=0.52*(1+T/43)*(S/35)*(f2*f.^2)/(f2^2+f.^2)*exp(-D/6); % dB/km
```

```
% absorption due to water
alpha3=4.9e-4*f.^2*exp(-T/27+D/17); % dB/km

y=alpha1+alpha2+alpha3;
end
function y=absorption_T(f)

% this function calculates the attenuation coefficient in bB/km
% using Thorp method
% "f" is the frequency of the sound in kHz

y=1.0936132983*(0.1*f.^2/(1+f.^2)+40*f.^2/(4100+f.^2));
end
```

Appendix 3: MATLAB Script (Hotspot 10: WFR + BBM-TOSP)

This MATLAB script was written using MATLAB R2019; it simulates the release in HS 10, the FC performs the WFR for detection and BBM-TOSP for localization. All the settings are those described in the previous chapters.

MATLAB, in order to run the animation, must have the following picture in a folder accessible by MATLAB. The file must be named "template.png".




```
%%%%%  Oil Spill Detection and Localization    %%%%
%%%%%               WFR + BBM-TOSP             %%%%
%%%%%                 Hotspot 10               %%%%
%%%%%         Author: Gianluca Tabella         %%%%

clear
close all
clc

%% Release Modelling

disp('Release Modelling') % display the progress

p=1/3;  % parameter for release shape
r=2;    % parameter for release shape
N=10000; % number of instants
X=binary_source(p,r,N); % function giving an array that simulates the
events before the continuous release
tot=length(X); % total number of records
rel=nnz(X); % number of releases (positive events)
non_rel=tot-rel; % number of non-releases (negative events)

%% Signal Modelling

disp('Signal Modelling') % display the progress
```

```
f=20e-3; % kHz - sound frequency
T=3.8; % ∞C - temperature (T=3.8∞C at Goliat)
S=35; % ppt - salinity (S=35 ppt at Goliat)
D=350; % m - depth (D=350-400 m at Goliat)
pH=8; % pH of water (pH=8 at Goliat)
spread_cf=1.5; % spreading coeff. (spherical=2;cylindrical=1;practical=1.5)

c=speed_sound(D,T,S); % m/s - speed of sound underwater
alpha_water=absorption_FG(f,T,S,D,pH,c); % dB/km - absorption due to water
hotspot_crd=[11.5,11.75 ; 11.5,12.75 ; 11.5,13.75 ; 15,11.75 ; 15,12.75 ;
15,13.75 ; 18,11.75 ; 18,12.75 ; 19,11.75 ; 19,12.75 ; 24.5,11.75 ;
24.5,12.75 ; 25.5,11.75 ; 25.5,12.75 ; 28.2,11.75 ; 28.2,12.75 ; 28.2,13.75
; 31.7,11.75 ; 31.7,12.75; 31.7,13.75]; % m - hotspots' coordinates
HS=length(hotspot_crd(:,1)); % number of hot spots
target=10;
target_crd=hotspot_crd(target,:); % m - target coordinates
sensors_crd=[13.97,15.3 ; 29.35,14.66 ; 21.86,9.42]; % m - sensor
coordinates
l=transpose(sqrt((target_crd(1)-sensors_crd(:,1)).^2+(target_crd(2)-
sensors_crd(:,2)).^2)); % m - distance of sensors from target
K=length(l); % number of sensors
l_FC=zeros(length(hotspot_crd(:,1)),K);
for i=1:length(hotspot_crd(:,1))  % m - (HSxK matrix) distances between HS
hot spots and K sensors
    for j=1:K
        l_FC(i,j)=sqrt((hotspot_crd(i,1)-
sensors_crd(j,1))^2+(hotspot_crd(i,2)-sensors_crd(j,2))^2);
    end
end
l0=min(min(l_FC)); % m - reference distance
TL=alpha_water*(l-l0)*10^-3+spread_cf*10*log10(l/l0); % dB - Transmission
Loss
AAF=exp(-TL./(20*log10(exp(1)))); % AAF (TL converted in Nepers)
var_noise=1; % variance of noise (gaussian)
SNR=20; % signal-to-noise ratio at closest (sensor/hot spot)
var_fadsig=var_noise*SNR; % variance of fading (gaussian)

noise_ampl=zeros(K,non_rel);
for i=1:K
    noise_ampl(i,:)=sqrt(var_noise)*randn(1,non_rel);
end
rel_ampl=zeros(K,rel);
for i=1:K

rel_ampl(i,:)=AAF(i)*sqrt(var_fadsig)*randn(1,rel)+sqrt(var_noise)*randn(1,
rel);
end
u=1;
v=1;
ampl=zeros(K,tot); % pre-allocation for final scenario
for i=1:tot
    switch X(i)
        case 1
            for j=1:K
                ampl(j,i)=rel_ampl(j,u);
            end
            u=u+1;
        case 0
            for j=1:K
                ampl(j,i)=noise_ampl(j,v);
```

```
            end
            v=v+1;
    end
end

%% Sensors' Settings

disp('Sensors setting calculation') % display the progress

th_loc_poss=0:.00001:10; % threshold for local decision (non-negative)

f_FC=f; % kHz - sound frequency
T_FC=T; % ∞C - temperature (T=3.8∞C at Goliat)
S_FC=S; % ppt - salinity (S=35 ppt at Goliat)
D_FC=D; % m - depth (D=350-400 m at Goliat)
pH_FC=pH; % pH of water (pH=8 at Goliat)
spread_cf_FC=spread_cf; % spreading coeff.
(spherical=2;cylindrical=1;practical=1.5)
c_FC=speed_sound(D_FC,T_FC,S_FC); % m/s - speed of sound underwater
alpha_water_FC=absorption_FG(f_FC,T_FC,S_FC,D_FC,pH_FC,c_FC); % dB/km -
absorption due to water
TL_FC=alpha_water_FC*(l_FC-l0)*10^-3+spread_cf_FC*10*log10(l_FC/l0); % dB -
Transmission Loss
AAF_FC=exp(-TL_FC/(20*log10(exp(1)))); % AAF (TL converted in Nepers)
var_noise_FC=1; % variance of noise (gaussian)
SNR_FC=20; % signal-to-noise ratio at closest (sensor/hot spot)
var_fadsig_FC=var_noise_FC*SNR_FC; % variance of fading (gaussian)
CZ_loc=zeros(HS,K);
CZ_loc_db=zeros(length(th_loc_poss),K);
th_loc=zeros(1,K);
for i=1:length(th_loc_poss)
    pfa_FC_poss=2*qfunc(sqrt(th_loc_poss(i)/var_noise_FC)); % array of
probabilities of false alarm for the single sensors

pd_FC_poss=2*qfunc(sqrt(th_loc_poss(i)./(var_fadsig_FC*AAF_FC.^2+var_noise_
FC))); % array of the corresponing probabilities of detection
    CZ_loc=pd_FC_poss.*(1-pfa_FC_poss);
    CZ_loc_db(i,:)=mean(CZ_loc);
end

for i=1:K
    th_loc(i)=th_loc_poss(CZ_loc_db(:,i)==max(CZ_loc_db(:,i)));
end

pfa_FC=2*qfunc(sqrt(th_loc/var_noise_FC)); % array of probabilities of
false alarm for the single sensors
pd_FC=zeros(HS,K);
for i=1:K
pd_FC(:,i)=2*qfunc(sqrt(th_loc(i)./(var_fadsig_FC*AAF_FC(:,i).^2+var_noise_
FC))); % array of the corresponing probabilities of detection
end
pd_FC=mean(pd_FC);

%% FC's Settings

disp('FC Settings Calculation') % display the progress

p_FC=p;  % parameter for release shape
r_FC=r;    % parameter for release shape
N_FC=N; % number of instants
```

```
X_FC=binary_source(p_FC,r_FC,N_FC); % function giving an array that
simulates the events before the continuous release
tot_FC=length(X_FC); % total number of records
rel_FC=nnz(X_FC); % number of releases (positive events)
non_rel_FC=tot_FC-rel_FC; % number of non-releases (negative events)
noise_ampl_FC=zeros(K,non_rel_FC);
for i=1:K
    noise_ampl_FC(i,:)=sqrt(var_noise_FC)*randn(1,non_rel_FC);
end
rel_ampl_FC=zeros(K,rel_FC);
for i=1:K

rel_ampl_FC(i,:)=mean(AAF_FC(:,i))*sqrt(var_fadsig_FC)*randn(1,rel_FC)+sqrt
(var_noise_FC)*randn(1,rel_FC);
end
u=1;
v=1;
ampl_FC=zeros(K,tot_FC); % pre-allocation for final scenario
for i=1:tot_FC
    switch X_FC(i)
        case 1
            for j=1:K
                ampl_FC(j,i)=rel_ampl_FC(j,u);
            end
            u=u+1;
        case 0
            for j=1:K
                ampl_FC(j,i)=noise_ampl_FC(j,v);
            end
            v=v+1;
    end
end

detect_FC=zeros(K,tot_FC); % matrix containing all the decisions of any
sensor during the simulation (every row is a sensor)
for i=1:K
    for j=1:tot_FC
        if ampl_FC(i,j)^2>=th_loc(i)
            detect_FC(i,j)=1;
        else
            detect_FC(i,j)=0;
        end
    end
end

th=-4:0.001:4; % array of possibile thresholds
no_thr=length(th); % number of possible thresholds

result_FC=zeros(no_thr,tot_FC); % matrix containing all the decisions of
the FC for different thresholds
lambda_FC=zeros(1,tot_FC); % vector containing all the FR's results

for j=1:no_thr % for each threshold
    sentence=['FC setting calculation: elaboration threshold number '
,num2str(j), ' of ' ,num2str(no_thr)];
    disp(sentence)
    for i=1:tot_FC % for each instant

lambda_FC(i)=sum(transpose(detect_FC(:,i)).*log(pd_FC./pfa_FC)+transpose((1
-detect_FC(:,i))).*log((1-pd_FC)./(1-pfa_FC))); % value of FR
```

```
        if lambda_FC(i)>th(j) % for-loop determining the decision of the FC
based on WFR
            result_FC(j,i)=1;
        else
            result_FC(j,i)=0;
        end
    end
end

%% FC's Performances

FP=zeros(1,no_thr); % array of number of false alarms at different
thresholds
TP=zeros(1,no_thr); % array of number of detections at different thresholds
FN=zeros(1,no_thr); % array of number of false negatives at different
thresholds
TN=zeros(1,no_thr); % array of number of true negatives at different
thresholds
Q_F=zeros(1,no_thr); % array of probabilities of false alarm at different
thresholds
Q_D=zeros(1,no_thr); % array of probabilities of detecion at different
thresholds
precision=zeros(1,no_thr); % array of precisions at different thresholds

for i=1:no_thr % for each threshold
        sentence=['FC setting calculation: evaluation threshold number '
,num2str(i), ' of ' ,num2str(no_thr)];
        disp(sentence) % display the progress

        fa=find((X_FC==0)&(result_FC(i,:)==1));
        FP(i)=length(fa);    % number of false alarms FP
        Q_F(i)=FP(i)/non_rel_FC; % FPR

        d=find((X_FC==1)&(result_FC(i,:)==1));
        TP(i)=length(d); % number of detection (true positives)
        Q_D(i)=TP(i)/rel_FC; % TPR - True Positive Rate - Probability of
Detection

        FN(i)=rel_FC-TP(i); % False negatives
        TN(i)=non_rel_FC-FP(i); % True Negatives

        precision(i)=TP(i)/(TP(i)+FP(i)); % precision
end
precision(isnan(precision))=1; % built to overcome the presence of NaN
results when precision converges to 1

%% Removal of Useless Thresholds

disp('Removal of Useless Thresholds from FC setting calculation history')
j=2; % loop used to remove results when no change was determined varying
the threshold
result_FC_new(1,:)=result_FC(1,:);
Q_D_new(1)=Q_D(1);
Q_F_new(1)=Q_F(1);
precision_new(1)=precision(1);
th_new(1)=th(1);
TP_new(1)=TP(1);
FP_new(1)=FP(1);
FN_new(1)=FN(1);
TN_new(1)=TN(1);
for i=2:no_thr
```

```
    if Q_D(i)==Q_D(i-1) && Q_F(i)==Q_F(i-1) && precision(i)==precision(i-1)
    else
        Q_D_new(j)=Q_D(i);
        Q_F_new(j)=Q_F(i);
        precision_new(j)=precision(i);
        th_new(j)=th(i);
        TP_new(j)=TP(i);
        FP_new(j)=FP(i);
        FN_new(j)=FN(i);
        TN_new(j)=TN(i);
        result_FC_new(j,:)=result_FC(i,:);
        j=j+1;
    end
end
Q_D=Q_D_new;
Q_F=Q_F_new;
precision=precision_new;
th=th_new;
TP=TP_new;
FP=FP_new;
FN=FN_new;
TN=TN_new;
result_FC=result_FC_new;
no_thr=length(th);
clearvars Q_D_new Q_F_new precision_new th_new TP_new FP_new FN_new TN_new
result_FC_new

%% Analysis of Parameters

disp('Analysis of Parameters for FC setting')

AUC=-trapz(Q_F,Q_D); % calculation of Area Under the Curve

% the matrix called "perf" has rows representing each indicator
% for each raw of the matrix the 1st value is the cut-off threshold at
indicator¥s maximum value
% the 2nd value is the Prob. of Detection at cut off threshold
% the 3rd value is the Prob. of False Alarm at cut off threshold
% the 4th value is the indicator¥s maximum value

J=Q_D-Q_F; % calculation of Youden's indexes
perf(1,4)=max(J); % maximum value of Youden's indexes
ind=find(J==perf(1,4)); % array's index for max Youden's index
perf(1,1)=th(ind); % cut off threshold at max J value
perf(1,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(1,2)=Q_D(ind); % Prob. of Detection at cut off threshold

beta_value=1; % 1: pr and rec equally weighted, 2: recall weighted higher
than precision, 0.5: precision weighted higher than recall
F_beta=(1+beta_value^2)*(precision.*Q_D)./(beta_value^2*precision+Q_D); %
F_value (armonic mean between precision and recall, with weight)
perf(2,4)=max(F_beta); % maximum value of F_beta
ind=find(F_beta==perf(2,4)); % array's index for max F_beta
perf(2,1)=th(ind); % cut off threshold at max F_beta
perf(2,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(2,2)=Q_D(ind); % Prob. of Detection at cut off threshold

dist_roc=(1-Q_D).^2+Q_F.^2; % sq. distance to (0,1)
perf(3,4)=min(dist_roc); % minimum sq. distance to (0,1)
ind=find(dist_roc==perf(3,4)); % array's index for min sq. distance to
(0,1)
```

```
perf(3,1)=th(ind); % cut off threshold at min sq. distance to (0,1)
perf(3,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(3,2)=Q_D(ind); % Prob. of Detection at cut off threshold

CZ=Q_D.*(1-Q_F); % CZ index
perf(4,4)=max(CZ); % maximum value of CZ index
ind=find(CZ==perf(4,4)); % array's index for max CZ index
perf(4,1)=th(ind); % cut off threshold at max CZ index
perf(4,3)=Q_F(ind); % Prob. of False Alarm at cut off threshold
perf(4,2)=Q_D(ind); % Prob. of Detection at cut off threshold

%% Sensors' Detection

detect=zeros(K,tot); % matrix containing all the decisions of any sensor
during the simulation (every row is a sensor)
for i=1:K
    sentence=['simulating detection: sensor ' ,num2str(i), ' of '
,num2str(K)]; % display the progress
    disp(sentence)
    for j=1:tot
        if ampl(i,j)^2>=th_loc(i)
            detect(i,j)=1;
        else
            detect(i,j)=0;
        end
    end
end

Q_F_test_loc=zeros(1,K);
Q_D_test_loc=zeros(1,K);

for i=1:K
    fa_test_loc=find((X==0)&(detect(i,:)==1));
    FP_test_loc=length(fa_test_loc);    % number of false alarms FP
    Q_F_test_loc(i)=FP_test_loc/non_rel; % FPR

    d_test_loc=find((X==1)&(detect(i,:)==1));
    TP_test_loc=length(d_test_loc); % number of detection (true positives)
    Q_D_test_loc(i)=TP_test_loc/rel; % TPR - True Positive Rate -
Probability of Detection
end

%% Fusion Center's Detection and Localization

inst_FC=1; % number of instants to take into account in the TCR
time_th=0; % threshold used in TCR
SP_calc_inst=zeros(tot,2);
index_rel=0;
SP_calc_db=zeros(tot,2);
SP_num_db(1:2,1:tot)=NaN;
SP_calc=[0 0];

dist_SP_HS=zeros(1,HS);
result=zeros(1,tot); % matrix containing all the decisions of the FC for
different thresholds
lambda=zeros(1,tot); % vector containing all the FR's results
lambda_time=zeros(1,tot); % vector containing all the TCR's results
final_result=zeros(1,tot); % matrix containing final decision (WFR+TCR)

for i=1:tot % for each instant
        if rem(i,100)==0 || i==1 % display the progress
```

```
        sentence=['Instant simulated by FC: ' ,num2str(i), ' of '
,num2str(tot)];
        disp(sentence)
        end

lambda(i)=sum(transpose(detect(:,i)).*log(pd_FC./pfa_FC)+transpose((1-
detect(:,i))).*log((1-pd_FC)./(1-pfa_FC))); % value of FR
        if lambda(i)>perf(1,1) % for-loop determining the decision of the
FC based on WFR
            result(i)=1;
        else
            result(i)=0;
        end

        if result(i)==0
            inst=inst_FC;
            if inst>i % necessary for the first instants (when there aren't
enough previous data)
                inst=i;
            end
            lambda_time(i)=sum(result(i-inst+1:i));
            if lambda_time(i)>time_th % for-loop determining the final
decision of the FC
                final_result(i)=1;
            else
                final_result(i)=0;
            end
            else
             final_result(i)=1;
        end

        if result(i)==1 && sum(detect(:,i))>=1
            index_rel=index_rel+1;

SP_calc_inst(index_rel,:)=(sum(((repmat(transpose(pd_FC./pfa_FC),1,size(2,2
))).*sensors_crd.*detect(:,i))+((repmat(transpose(pd_FC./pfa_FC),1,size(2,2
))).*(repmat(1-
(detect(:,i)),1,size(2,2))).*(2*(sum((repmat(transpose(pd_FC./pfa_FC),1,siz
e(2,2))).*sensors_crd.*detect(:,i))/sum((repmat(transpose(pd_FC./pfa_FC),1,
size(2,2))).*detect(:,i)))-sensors_crd.*(1-
detect(:,i))))))/(sum(pd_FC./pfa_FC));
            if index_rel==1
                SP_calc=[SP_calc_inst(index_rel,1)
SP_calc_inst(index_rel,2)];
            else
                SP_calc=[((index_rel-
1)*SP_calc(1)+SP_calc_inst(index_rel,1))/index_rel ((index_rel-
1)*SP_calc(2)+SP_calc_inst(index_rel,2))/index_rel];
            end
            for count_HS=1:HS
                dist_SP_HS(count_HS)=sqrt((SP_calc(1)-
hotspot_crd(count_HS,1))^2+(SP_calc(2)-hotspot_crd(count_HS,2))^2);
            end
            SP_num=find(dist_SP_HS==min(dist_SP_HS));
            SP_calc_db(i,:)=SP_calc;
            SP_num_db(1:length(SP_num),i)=transpose(SP_num);
        else
            if sum(sum(SP_calc_inst))==0
                SP_calc_db(i,:)=NaN;
            else
                SP_calc_db(i,:)=SP_calc_db(i-1,:);
```

```
            end
        end
        if final_result(i)==1 && result(i)==0
            SP_calc_db(i,:)=SP_calc;
            SP_num_db(1:length(SP_num),i)=transpose(SP_num);
        end
end

fa_test=find((X==0)&(final_result==1));
FP_test=length(fa_test);    % number of false alarms FP
Q_F_test=FP_test/non_rel; % FPR

d_test=find((X==1)&(final_result==1));
TP_test=length(d_test); % number of detection (true positives)
Q_D_test=TP_test/rel; % TPR - True Positive Rate - Probability of Detection

precision_test=TP_test/(TP_test+FP_test); % precision

%% Animation

answer = questdlg('Do you want to see the animation?','Yes','No');
switch answer
    case 'Yes'
        answer = 1;
    case 'No'
        answer = 0;
end
if answer==1
sentence='Show Animation';
disp(sentence)
img=imread('template.png');
figure(1)
for i=1:tot
hold off
imagesc([0 41.26],[0 27.45],flipud(img));
set(gca,'ydir','normal');
set(gcf, 'Position', get(0, 'Screensize'));
xlim([0 41.26])
ylim([0 27.45])
xlabel('x (meters)')
ylabel('y (meters)')
grid on
hold on
scatter(hotspot_crd(:,1),hotspot_crd(:,2),400,'filled','r')
if X(i)==1
    scatter(target_crd(1),target_crd(2),400,'filled','b')
end
scatter(sensors_crd(:,1),sensors_crd(:,2),400,'filled','g')
    if isnan(SP_calc_db(i,1))==0
        scatter(SP_calc_db(i,1),SP_calc_db(i,2),100,'filled','m')
    end
    if isnan(SP_num_db(1,i))==0 && isnan(SP_num_db(2,i))==1

scatter(hotspot_crd(SP_num_db(1,i),1),hotspot_crd(SP_num_db(1,i),2),100,'fi
lled','y')
    else
        if isnan(SP_num_db(1,i))==0 && isnan(SP_num_db(2,i))==0

scatter(hotspot_crd(SP_num_db(1:size(SP_num_db,1),i),1),hotspot_crd(SP_num_
db(1:size(SP_num_db,1),i),2),100,'filled','y')
        end
```

```
    end
title(['2D Scenario. Instant number: ' num2str(i) ' of ' num2str(tot)])
drawnow
end
end

%% Plot

disp('Plot and curves generation')

figure(1) % release plot
plot(X)
xlabel('time (s)')
ylabel('0 = Non-release / 1 = Release')
xlim([0 tot])
ylim([0 1])

figure(2) % Square attenuation vs. Distance
x_plot=l0:.1:max(l);
x_plot=[x_plot max(l)];
TL_plot=alpha_water*(x_plot-l0)*10^-3+spread_cf*10*log10(x_plot/l0);
AAF_plot=exp(-TL_plot./(20*log10(exp(1))));
plot(x_plot,AAF_plot,'b')
xlabel('Distance (m)')
ylabel('Attenuation (AAF)')
for i=1:K
hold on
line([l(i) l(i)],[0 1]) % vertical lines representing sensors position
end
hold off
xlim([l0-1 max(l)+1])
ylim([0 max(AAF_plot)])
title('Attenuation vs Distance')

figure(3) % wave amplitude vs time plot
for i=1:K
    z0=ampl(i,:);
    z1=ampl(i,:);
    for j=1:tot
        switch X(j)
            case 0
                z1(j)=NaN;
            case 1
                z0(j)=NaN;
        end
    end
    subplot(K,1,i)
    plot(z0,'b')
    hold on
    plot(z1,'r')
    xlabel('time (s)')
    ylabel('y (Amplitude)')
    title(['Received Signal, sensor ' num2str(i) ', coordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
    hold off
end

figure(4) % Amplitude Distribution Plot
for i=1:K
subplot(1,K,i)
histogram(noise_ampl(i,:),'Normalization','pdf')
```

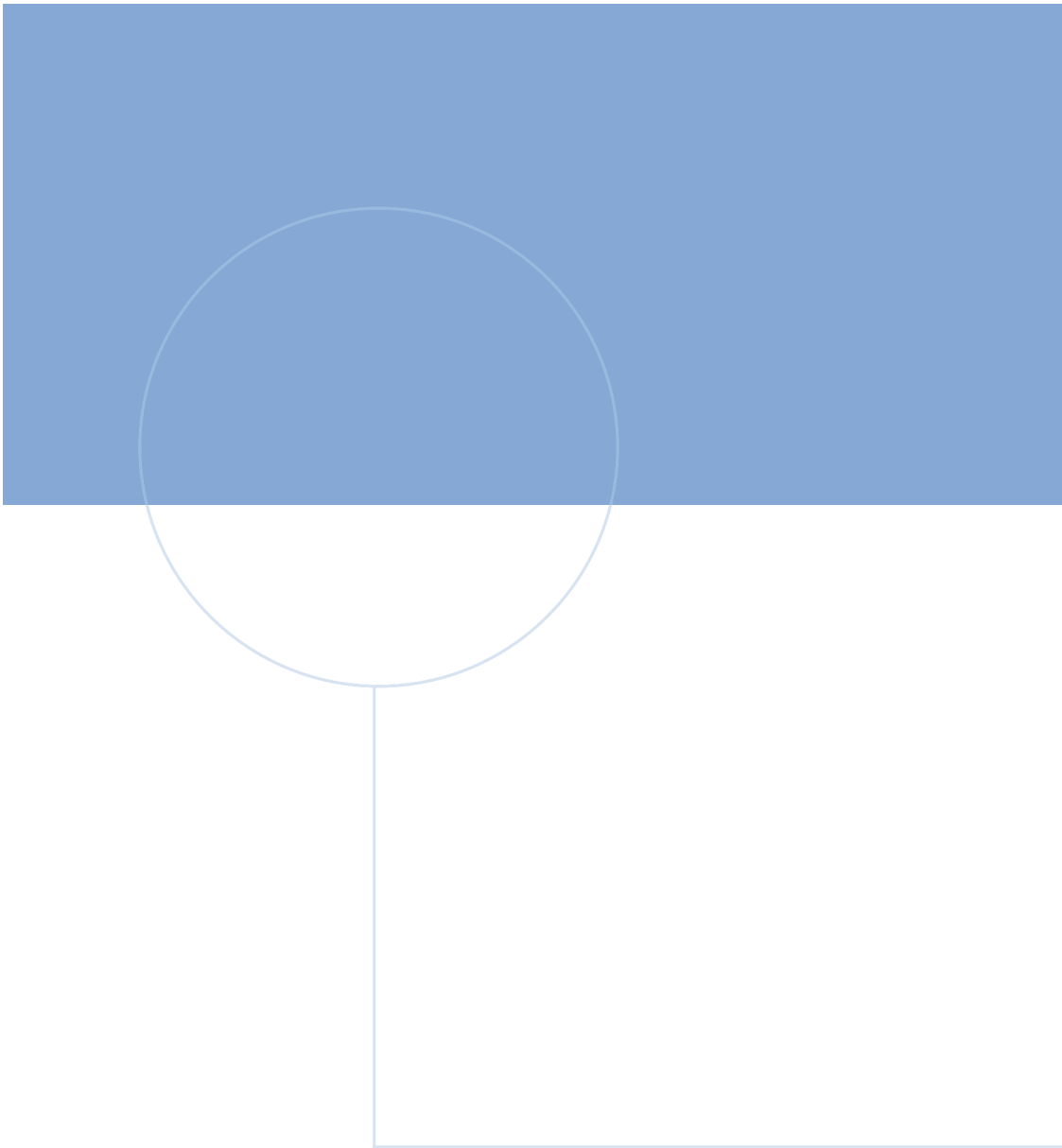
```
hold on
histogram(rel_ampl(i,:),'Normalization','pdf')
hold on
line([sqrt(th_loc(i))
sqrt(th_loc(i))],get(gca,'ylim'),'LineWidth',2,'Color','k')
line([-sqrt(th_loc(i)) -
sqrt(th_loc(i))],get(gca,'ylim'),'LineWidth',2,'Color','k')
legend('p(y|H0)','p(y|H1)')
title(['Amplitude pdf, sensor ' num2str(i) ', coordordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
xlabel('y (Amplitude)')
hold off
end

figure(5)
for i=1:K
    subplot(1,K,i)
    th_loc_poss=0:.01:10;

plot(2*qfunc(sqrt(th_loc_poss/var_noise)),2*qfunc(sqrt(th_loc_poss/(var_fad
sig*AAF_FC(target,i)^2+var_noise))))
    hold on

scatter(2*qfunc(sqrt(th_loc(i)/var_noise)),2*qfunc(sqrt(th_loc(i)/(var_fads
ig*AAF_FC(target,i)^2+var_noise))),'filled')
    line([0 1],[0 1],'LineStyle','--')
    hold off
    title(['ROC sensor ' num2str(i) ', coordinates: ('
num2str(sensors_crd(i,1)) ',' num2str(sensors_crd(i,2)) ')'])
    xlabel('Local Probability of False Alarm')
    ylabel('Local Probability of Detection')
    xlim([0 1])
    ylim([0 1])
    pbaspect([1 1 1])
end

figure(6) % ROC curve - linear
plot(Q_F,Q_D,'-or')
hold on
line([0 1],[0 1],'LineStyle','--') % chance line
scatter(Q_F_test,Q_D_test,'filled')
hold off
title('ROC CURVE - linear')
xlabel('Global Probability of False Alarm')
ylabel('Global Probability of Detection')
xlim([0 1])
ylim([0 1])

figure(7) % ROC curve - semilog
semilogx(Q_F,Q_D,'-or')
hold on
scatter(Q_F_test,Q_D_test,'filled')
hold off
title('ROC CURVE - semilog')
xlabel('Global Probability of False Alarm')
ylabel('Global Probability of Detection')
xlim([Q_F(end-1) 1])
ylim([Q_D(end-1)-0.05 1])

figure(8) % PR curve - linear
plot(Q_D,precision,'-or')
```

```
hold on
scatter(Q_D_test,precision_test,'filled')
hold off
title('PR CURVE')
xlabel('Recall')
ylabel('Precision')
xlim([0 1])
ylim([0 1])

SP_num_db_plot=zeros(1,index_rel);
dist_plot=zeros(1,index_rel);
j=1;
for i=1:N
    if result(i)==1
        SP_num_db_plot(j)=SP_num_db(1,i);
        dist_plot(j)=sqrt((target_crd(1)-
hotspot_crd(SP_num_db_plot(j),1))^2+(target_crd(2)-
hotspot_crd(SP_num_db_plot(j),2))^2);
        j=j+1;
    end
end
AUC_localization=trapz(1:index_rel,dist_plot);
figure(9)
plot(1:index_rel,dist_plot)
title(['Distance from Target. Area under curve (no. of instants: '
num2str(index_rel) ') = ' num2str(AUC_localization) ''])
xlabel('Time (s) / Fusion Center Positive Detection')
ylabel('Distance from Target (m)')
xlim([1 index_rel])
ylim([0 max(dist_plot)])

%% Memory Cleaning

disp('Memory Cleaning')
clearvars x_plot TL_plot AAF_plot x d fa j i q s sentence img u v
noise_ampl_FC rel_ampl_FC ind th_loc_poss pfa_FC_poss pd_FC_poss
fa_test_loc FP_test_loc d_test_loc TP_test_loc % deleting variables

%% Functions

function x=binary_source(p,r,N)

A10=(1-p)/(p+r-p*r);
A01=p/(p+r-p*r);

x=zeros(1,N); % pre-allocation of memory to improve code performances

for n=2:N % it starts from 2 because we want the initial state n=1 to be
without release
    switch x(n-1)
        case 0
            x(n)=(rand<A01);
        case 1
            x(n)=(rand<1-A10);
    end
end
end

function y=speed_sound(D,T,S)

% it calculates sound speed using corrected UNESCO algorithm
```

```
c0=1402.388+5.03711*T-5.80852e-2*T^2+3.3420e-4*T^3-1.47800e-6*T^4+3.1464e-
9*T^5;
c1=0.153563+6.8982e-4*T-8.1788e-6*T^2+1.3621e-7*T^3-6.1185e-10*T^4;
c2=3.1260e-5-1.7107e-6*T+2.5974e-8*T^2-2.5335e-10*T^3+1.0405e-12*T^4;
c3=-9.7729e-9-3.8504e-10*T-2.3643e-12*T^2;
A0=1.389-1.262e-2*T+7.164e-5*T^2+2.006e-6*T^3-3.21e-8*T^4;
A1=9.4742e-5-1.2580e-5*T-6.4885e-8*T^2+1.0507e-8*T^3-2.0122e-10*T^4;
A2=-3.9064e-7+9.1041e-9*T-1.6002e-10*T^2+7.988e-12*T^3;
A3=1.100e-10+6.649e-12*T-3.389e-13*T^2;
P=D/10;
A=A0+A1*P+A2*P^2+A3*P^3;
B=-1.922e-2-4.42e-5*T+(7.3637e-3+1.7945e-7*T)*P;
C=-7.9836e-6*P+1.727e-3;
y=c0+c1*P+c2*P^2+c3*P^3+A*S+B*S^(3/2)+C*S^2;
end

function y=absorption_FG(f,T,S,D,pH,c)

% this function calculates the attenuation coefficient in bB/km
% using Francois-Garrison method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m
% "pH" is the indicator of water acidity

theta=273.15+T; % K, temperature

% excess absorption due to Boric Acid (H3BO3)
A1=(8.86/c)*10^(0.78*pH-5); % dB/(km*kHz)
P1=1; % nondimensional pressure correction factor
f1=2.8*sqrt(S/35)*10^(4-1245/theta); % kHz, relaxation frequencies of H3BO3
alpha1=(A1*P1*f1*f.^2)/(f1^2+f.^2); % dB/km

% excess absorption due to Magnesium Sulfate (MgSO4)
A2=21.44*(S/c)*(1+0.025*T); % dB/(km*kHz)
P2=1-1.37e-4*D+6.2e-9*D^2; % nondimensional pressure correction factor
f2=(8.17*10^(8-1990/theta))/(1+0.0018*(S-35)); % kHz, relaxation
frequencies of MgSO4
alpha2=(A2*P2*f2*f.^2)/(f2^2+f.^2); % dB/km

% absorption due to water
if T<=20
    A3=4.937e-4-2.59e-5*T+9.11e-7*T^2-1.5e-8*T^3; % dB/(km*kHz^2)
else
    A3=3.964e-4-1.146e-5*T+1.45e-7*T^2-6.5e-10*T^3; % dB/(km*kHz^2)
end
P3=1-3.83e-5*D+4.9e-10*D^2; % nondimensional pressure correction factor
alpha3=A3*P3*f.^2; % dB/km

y=alpha1+alpha2+alpha3;
end
function y=absorption_FS(f,T,D)

% this function calculates the attenuation coefficient in bB/km
% using Fisher-Simmons method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "D" is the depth in m
```

```
f=f*1000; % method is written using Hz
theta=273.15+T; % K, temperature
P=D/10+1; % atm, absolute pressure estimation (supposing 10 m_seawater = 1
atm)

% excess absorption due to Boric Acid (H3BO3)
A1=1.03e-8+2.36e-10*T-5.22e-12*T^2; % Np/(m*Hz)
P1=1; % nondimensional pressure correction factor
f1=1.32e3*theta*exp(-1700/theta); % Hz, relaxation frequencies of H3BO3
alpha1=(A1*P1*f1*f.^2)/(f1^2+f.^2); % Np/m

% excess absorption due to Magnesium Sulfate (MgSO4)
A2=5.62e-8+7.52e-10*T; % Np/(m*Hz)
P2=1-10.3e-4*P+3.7e-7*P^2; % nondimensional pressure correction factor
f2=1.55e7*theta*exp(-3052/theta); % Hz, relaxation frequencies of MgSO4
alpha2=(A2*P2*f2*f.^2)/(f2^2+f.^2); % Np/m

% absorption due to water
A3=(55.9-2.37*T+4.77e-2*T^2-3.48e-4*T^3)*10^-15; % Np/(m*Hz^2)
P3=1-384e-4*P+7.57e-8*P^2; % nondimensional pressure correction factor
alpha3=A3*P3*f.^2; % Np/m

y=alpha1+alpha2+alpha3; % Np/m
y=y*1000*20*log10(exp(1)); % dB/km
end
function y=absorption_SM(f,T,S,D)

% this function calculates the attenuation coefficient in bB/km
% using Shulkin-Marsh method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m

theta=273.15+T; % K, temperature
P=D/10+1; % atm, absolute pressure estimation (supposing 10 m_seawater = 1
atm)
A=2.34e-6;
B=3.38e-6;
fT=21.9*10^(6-1520/theta); % kHz
y=((S*A*fT*f.^2)/(fT^2+f.^2)+(B*f.^2)/fT)*(1-6.54e-4*P); % Np/m
y=y*1000*20*log10(exp(1)); % dB/km
end
function y=absorption_AM(f,T,S,D,pH)

% this function calculates the attenuation coefficient in bB/km
% using Ainslie-McColm method
% "f" is the frequency of the sound in kHz
% "T" is the temperature in ∞C
% "S" is the salinity in parts-per-thousand
% "D" is the depth in m
% "pH" is the indicator of water acidity

D=D/1000; % method is written using km

% excess absorption due to Boric Acid (H3BO3)
f1=0.78*exp(T/26)*sqrt(S/35); % kHz, relaxation frequencies of H3BO3
alpha1=0.106*(f1*f.^2)/(f1^2+f.^2)*exp((pH-8)/0.56); % dB/km

% excess absorption due to Magnesium Sulfate (MgSO4)
f2=42*exp(T/17); % kHz, relaxation frequencies of MgSO4
```

```
alpha2=0.52*(1+T/43)*(S/35)*(f2*f.^2)/(f2^2+f.^2)*exp(-D/6); % dB/km

% absorption due to water
alpha3=4.9e-4*f.^2*exp(-T/27+D/17); % dB/km

y=alpha1+alpha2+alpha3;
end
function y=absorption_T(f)

% this function calculates the attenuation coefficient in bB/km
% using Thorp method
% "f" is the frequency of the sound in kHz

y=1.0936132983*(0.1*f.^2/(1+f.^2)+40*f.^2/(4100+f.^2));
end
```



NTNU
Norwegian University of
Science and Technology