Joona Koskiahde

Decentralized Detection in Realistic Sensor Networks

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 26.12.2011

Thesis supervisor:

Prof. Andreas Richter

Thesis instructor:

D.Sc. (Tech.) Jan Eriksson

Aalto University
School of Electrical
Engineering

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN TIIVISTELMÄ

Tekijä: Joona Koskiahde		
Työn nimi: Hajautettu ilmaisu toteutettavissa langattomissa sensoriverkoissa		
Päivämäärä: 26.12.2011	Kieli: Englanti	Sivumäärä:8+54
Signaalinkäsittelyn ja akustiikan laitos		
Professuuri: Signaalinkäsittely		Koodi: S-88
Valvoja: Prof. Andreas Richter		
Ohjaaja: TkT Jan Eriksson		

Tämä työ käsittelee kohteen ilmaisua sensoriverkolla, joka koostuu äänisensoreista. Työn pääpaino on epäideaalisen tilanteen käsittelyllä, jossa monet hajautettua ilmaisua käsittelevät oletukset, joita alan kirjallisuudessa tehdään, eivät enää päde. Sensoriverkko koostuu mielivaltaiseen verkkotopologiaan asetetuista sensoreista ja fuusiokeskuksesta, ja tavoite on ilmaista verkkoa lähestyvä kohde, joka tuottaa äänisignaalia. Tiedon käsittelyyn sensoreilla ja fuusiokeskuksella esitetään kaksi erilaista algoritmia. Toinen algoritmeista perustuu suurimman uskottavuuden menetelmään ja toinen on heuristinen, klassiseen ilmaisuteoriaan perustuva, lähestymistapa ongelmaan.

Algoritmien suorituskykyä tutkitaan simulaatioiden avulla. Heuristisen algoritmin suorituskyky on huomattavasti parempi kaikissa simuloiduissa tilanteissa. Algoritmien johdossa taustakohina oletettiin normaalijakautuneeksi, mutta simulaatioiden perusteella algoritmit toimivat kohtuullisen hyvin myös pidempihäntäisen taustakohinajakauman tapauksessa. Heuristinen algoritmi tarjoaa paremman suorituskyvyn lisäksi myös helpomman tavan asettaa kynnysarvoparametrit niin, että sensoreilla ja fuusiokeskuksella on haluttu väärän hälytyksen todennäköisyys.

Avainsanat: Hajautettu ilmaisu, Ilmaisuteoria, Sensoriverkot, Akustiset sensorit, Suurimman uskottavuuden menetelmä

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Joona Koskiahde

Title: Decentralized Detection in Realistic Sensor Networks

Date: 26.12.2011 Language: English Number of pages:8+54

Department of Signal Processing and Acoustics

Professorship: Signal Processing Code: S-88

Supervisor: Prof. Andreas Richter

Instructor: D.Sc. (Tech.) Jan Eriksson

This thesis discusses the detection of a target using a network of acoustic sensors. The focus of the work is on considering what to do in a non-ideal situation, where many of the assumptions often made in decentralized detection literature are no longer valid. The sensors and a fusion center are grouped in an arbitrary formation, and the object is to detect an approaching target which emits a sound signal. Two different schemes are considered for processing the data at sensors and the fusion center. One of the schemes is based on maximum likelihood estimation and the other one is a heuristic approach based on classical detection theory.

The performances of the two schemes are studied in simulations. The heuristic scheme has a better detection performance for a given false alarm rate with all different sets of settings for the simulation. In derivation of the schemes, the background acoustic noise is assumed to be normal distributed, but, according to the simulations, the schemes still work relatively well under a long tailed noise distribution. In addition to better performance, the heuristic scheme offers easier setup of threshold values and approximation of false alarm rates for given thresholds using simple equations.

Keywords: Decentralized detection, Distributed detection, Detection Theory, Sensor networks, Acoustic Sensors, Maximum likelihood

Preface

I would like to thank my instructor D.Sc. (Tech.) Jan Eriksson and supervisor Prof. Andreas Richter for help with the subject matter and general tips for writing a scientific text. Also, special thanks go to Mr. Jari Nieminen for help with modeling wireless communications in simulations and my brother, Mr. Timo Koskiahde, for proofreading and comments on readability.

Otaniemi, 26.12.2011

Joona Koskiahde

Contents

List of symbols

A	attenuation of a sound signal in decibels
A_{atm}	attenuation of a sound signal due to atmospheric absorption
A_{bar}	attenuation of a sound signal due to a barrier
A_{gr}	attenuation of a sound signal due to the ground effect
A_{div}	attenuation of a sound signal due to geometrical divergence
A_{misc}	attenuation of a sound signal due to miscellaneous effects
A_i	signal of the target at sensor i
α	a multiplier in the decision rule of the fusion center in the maximum likelihood scheme
α_s	attenuation parameter of the atmospheric absroption
B	number of sensors that have something to send at a given time
β	multiplier for the decision rule of the fusion center in the combining standard deviations scheme
d	distance a sound propagates in the air
d_0	reference distance of 1 m
D_c	directivity correction in decibels
$f(\mathbf{x}_i)$	a function of the measurements of sensor i
γ	threshold of a likelihood ratio test
γ_0	threshold of the decision rule of the fusion center
γ_i	threshold of a likelihood ratio test of sensor i
γ'_i	threshold to which test statistic $T(\mathbf{x}_i)$ is compared
$\Gamma(x)$	gamma function
H_0	null hypothesis, target absent hypothesis
H_1	alternative hypothesis, target present hypothesis
i	sensor index
k	sensor measurement index
k_G	shape parameter of a gamma distribution
K	number of measurements the sensors take into account in their decision rule
$l(\mathbf{x}; H_0)$	likelihood function of the measurements under H_0
$l(\mathbf{x}; H_1)$	likelihood function of the measurements under H_1
$L_{fT}(DW)$	continuous downwind octave-band sound pressure level
L_P	sound pressure level at a sensor due to the target
L_{P1m}	sound pressure level at 1 m from the source
L_W	sound power level of the source
$\Lambda(\mathbf{x})$	likelihood ratio
$\Lambda_i(\mathbf{x}_i)$	likelihood ratio at sensor i
$\tilde{\Lambda}_i(\mathbf{x}_i)$	likelihood ratio at sensor i with estimated likelihoods
M	binomially distributed random variable
μ	expected value of a random variable
μ_i	expected value of the measurements of sensor i
$\mu_{0,i}$	expected value of the measurements of sensor i under H_0

$\mu_{1,i}$	expected value of the measurements of sensor i under H_1
$\hat{\mu}_{0,i}$	estimate of $\mu_{0,i}$
N	number of sensors in the network
$\mathcal{N}(\mu, \sigma^2)$	normal distribution with parameters μ and σ^2
p	probability with which each of the B sensors transmit in a time slot
p_{opt}	approximately optimal value for p
$p(\mathbf{x})$	probability density function
$p(\mathbf{x}; H_0)$	probability density function of measurements under H_0
$p(\mathbf{x}; H_1)$	probability density function of measurements under H_1
$\phi_0(\phi_1, \ldots, \phi_N)$	decision rule of the fusion center
$\tilde{\phi}_0(\tilde{\phi}_1, \ldots, \tilde{\phi}_N)$	decision rule of the fusion center in the maximum likelihood scheme
$\phi_i(\mathbf{x}_i)$	local decision rule of sensor i
$\tilde{\phi}_i(\mathbf{x}_i)$	local decision rule of sensor i in the maximum likelihood scheme
P_D	probability of detection
P_{FA}	probability of false alarm
P_M	probability of miss
P_{SNR}	probability that a packet sent from a sensor is lost due to a low signal-to-noise ratio
$\Phi(x)$	cumulative distribution function of the standard normal distribution
$Q(x)$	Q-function, complementary cumulative distribution function of the standard normal distribution
$Q^{-1}(x)$	inverse Q-function
ρ	value associated with the censoring region of the sensors
ρ_i	value associated with the censoring region of sensor i
S	number of time slots in an active period
σ_i^2	variance of the background noise distribution at sensor i
$\hat{\sigma}_i^2$	estimate of σ_i^2
θ_G	scale parameter of a gamma distribution
$T(\mathbf{x}_i)$	test statistic of the combining standard deviations scheme
x	a single measurement
$x_i[k]$	a single measurement of sensor i
$\mathbf{x}$	vector of measurements
$\mathbf{x}_i$	vector of measurements of sensor i
$y_i(\mathbf{x}_i)$	normalized measurements of sensor i

1 Introduction

This thesis focuses on detecting a target with a network of sensors. The general introductions to sensor networks and the situation under consideration are given in this chapter. Additionally, organization of topics and objectives of the thesis are discussed.

1.1 Sensor networks

Sensor networks consist of sensors which measure some physical properties of the environment and are able to communicate with each other. The sensors are usually battery-powered and equipped with low-power computing hardware, and a radio transceiver [1]. Typically, there is also a node with heavier computing capabilities in the network called a fusion center, which is used to combine information from the sensors [2].

Energy consumption is an important aspect in the design of sensor networks since the sensors usually operate on battery-power. Therefore, smaller energy consumption results in longer lifetime of the network. This should be taken into account in all levels of design. Applications should generate little traffic into the network, network layer packets should have minimum overhead, and radios should transmit at moderate power levels.

There are numerous military and civil scenarios where sensor networks may be utilized. Deployed on the battlefield, a sensor network can detect, classify, and track enemy movements. In environmental studies, a sensor network can be used for, e.g., habitat monitoring or measuring temperature, wind speed, and humidity. [3]

1.2 Main scenario

In surveillance applications, the first and often the most critical step is detection of an intruder or a target. Naturally, if a target is to be classified, localized, or tracked, it must be detected first. In this thesis, the focus is on detecting a target using a sensor network. The main scenario considered is detection of a vehicle in a forest using acoustic sensors. This scenario could arise, e.g., in a military setting where a sensor network is deployed along a forest road to give an alarm if unknown vehicles move on the road.

In designing a good way to process the measurements from the sensors to detect the target, attention is turned to detection theory. Some basic principles and definitions are described from classical detection theory. Fundamentals of decentralized detection are also presented, where part of the decision making process is distributed to the sensors.

Although the vehicle could be detected with, e.g., seismic sensors [4], acoustic sensors are discussed here since the propagation of sound and hearing the target are relatively easy to understand intuitively. It should be clear that basic concepts of decision theory apply just as well to seismic and other types of measurements similarly to acoustic measurements.

1.3 Objectives of the thesis

The purpose of this thesis is to devise a suitable detection scheme for the main scenario described above. The reasons why many of the schemes presented in detection literature are not suitable for this scenario are discussed. Two different approaches to the problem of the main scenario are considered, and their performances are compared in simulations. Both approaches have parameters which need to be set. Methods to find practical values for these parameters are discussed, as well.

1.4 Organization of the thesis

Classical centralized detection is described in Chapter 2. In Chapter 3, decentralized detection with a sensor network is discussed. How to apply the theory of classical and decentralized detection to practical scenarios, especially the main scenario described in Chapter 1.2, is considered in Chapter 4. Chapter 5 highlights some characteristics of other scenarios that are related to the main scenario. To find out the performance of the proposed detection algorithms, a number of simulations were run. These simulations and their results are described in Chapter 6. Chapter 7 concludes the thesis and proposes directions for future research.

2 Classical detection

This chapter is an introduction to classical detection theory. In classical detection theory, only a single sensor and its measurements are considered. Some basic concepts and terminology are discussed here. These are needed later when the problem of the main scenario is discussed in detail.

2.1 Introduction

In the main scenario (Chapter 1.2) the objective is to infer if a target is absent or present in the network. The target emits a sound signal, and based on sound measurements a decision has to be made on absence or presence of the target. Measurements are perturbed by random background noise which makes the detection of the target signal nontrivial. Detecting a signal in noise is essentially a statistical hypothesis testing problem [5]. Absence or presence of the target correspond to null hypothesis (H_0) and alternative hypothesis (H_1), respectively. This is a binary hypothesis testing problem since there are only two different hypotheses.

Since measurements are disturbed by random background noise, the measurements are modeled as random variables. Under each hypothesis, H_0 and H_1, the measurements have a different distribution, e.g., the target signal may increase the mean of the distribution of background noise. In detection, the objective is to infer from which distribution the measurements are drawn. Since the background noise is usually modeled as a continuous distribution, a fixed set of measurements may have been generated by both distributions. Therefore, there is a possibility of making an error in the inference. A decision can be made that the target is present, although it is not, or, on the other hand, a decision can be made that there is no target, although there is one. The probability of deciding the target is present, even though it is not, is called the probability of false alarm (P_{FA}). The probability of deciding the target is absent, even though there is one, is called the probability of miss (P_M). The probability of making a correct decision when the target is present is called the probability of detection (P_D). Notice that $P_D = 1 - P_M$.

Figure 1 illustrates a case where there is only one measurement. Probability density functions of the measurements under H_0 and H_1 are denoted by $p(\mathbf{x}; H_0)$ and $p(\mathbf{x}; H_1)$, respectively. The figure illustrates a decision rule which decides in favor of H_1 when $x > 2$. The probability of detection equals the green area, and the probability of false alarm equals the red area. It should be obvious from the figure that P_{FA} can decreased by moving the threshold right, but at the same time P_D decreases. Conversely, both P_D and P_{FA} can be increased at the same time by moving the threshold left. This is due to the fact that P_{FA} and P_D cannot be adjusted to opposing directions at the same time by adjusting the threshold [5].

Naturally, a good detector has a high P_D and a low P_{FA}. A detector that maximizes P_D for a given P_{FA}, assuming the distributions under both hypotheses are completely known, is described next.

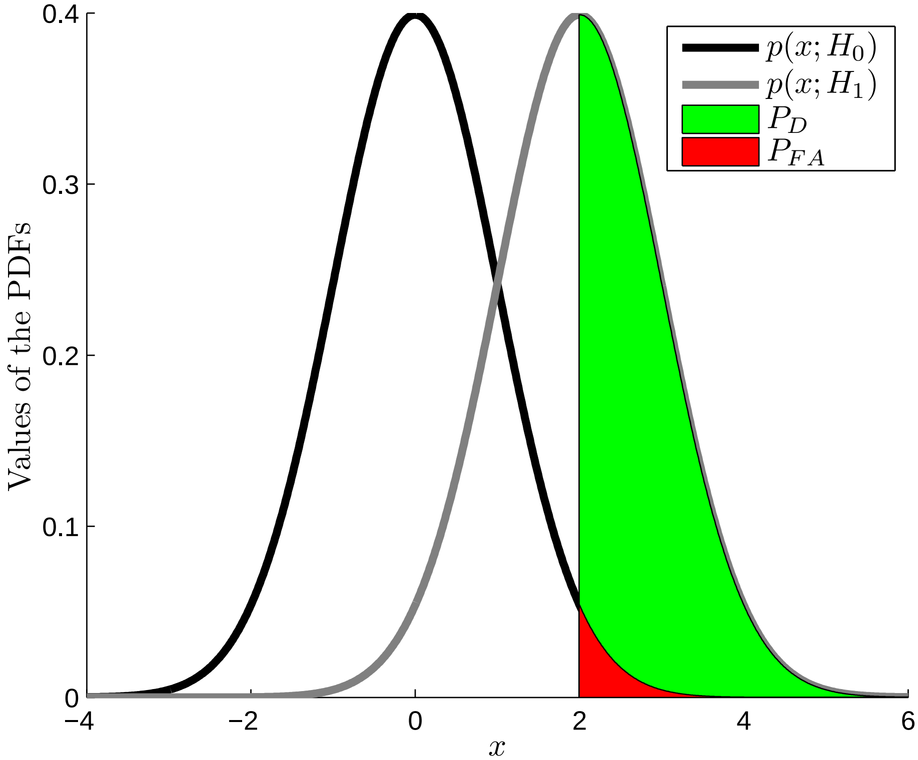

Figure 1: Probabilities of detection and false alarm for two example distributions using only one measurement x.

2.2 Neyman–Pearson theorem

The Neyman–Pearson (NP) theorem [6] states that to maximize the probability of detection (P_D) for a given probability of false alarm (P_{FA}), decide H_1 if

$$\Lambda(\mathbf{x}) = \frac{l(\mathbf{x}; H_1)}{l(\mathbf{x}; H_0)} > \gamma, \tag{1}$$

where $\mathbf{x}$ is the vector of measurements, $l(\mathbf{x}; H_0)$ and $l(\mathbf{x}; H_1)$ are the likelihood functions of the measurements under H_0 and H_1, respectively, and γ is the threshold which satisfies the given P_{FA}. The likelihood function is effectively the same as the probability density function of the measurements, i.e., $l(x) = p(x)$, only the interpretation is different. In the probability density function, the variable is x, and the parameters of the distribution are fixed. In the likelihood function, the variables are the parameters of the distribution, and x is fixed. Since $\Lambda(\mathbf{x})$ is the ratio of the likelihoods of H_1 and H_0, it is called the likelihood ratio. The whole test (1), including both the likelihood ratio and the threshold, is termed the likelihood ratio test (LRT).

The value of the threshold γ can be obtained from the following relations between

γ and P_{FA}

$$P_{FA} = \Pr\{\Lambda(\mathbf{x}) > \gamma; H_0\} = \int_{\{\mathbf{x}:\Lambda(\mathbf{x})>\gamma\}} l(\mathbf{x}; H_0)\,\mathrm{d}\mathbf{x}, \tag{2}$$

where the notation $\mathbf{x} : \Lambda(\mathbf{x}) > \gamma$ indicates the region of $\mathbf{x}$ where $\Lambda(\mathbf{x}) > \gamma$ [5].

2.3 Composite hypothesis testing

In the discussion above, it was assumed that the distributions of the measurements under both hypotheses are completely known. The scenario is termed simple hypothesis testing. If the distributions contain unknown parameters, the problem is called composite hypothesis testing. This is more akin to the main scenario (Chapter 1.2) where the objective is to detect a sound signal of *unknown* amplitude in noise. The first approach to designing a test for distributions containing unknown parameters is to design an NP test (1) assuming the unknown parameters are known. The test should be then modified so that it does not depend on the unknown parameters anymore, if possible. The resulting test is optimal in the Neyman-Pearson sense since it is an NP test. Another approach is to build the so-called generalized likelihood ratio test. In the generalized likelihood ratio test, the unknown parameters in the distributions are first estimated, and then the distributions are used in the likelihood ratio test with the estimated parameters. [5]

3 Decentralized detection

Instead of the situation of a single sensor of the previous chapter, detection with several sensors is considered in this chapter. This gives rise to a new problem of fusing the information from the sensors. Also, a method to save significant amounts of energy in this type of situation is discussed.

3.1 Introduction

In classical detection of Chapter 2, only one sensor, which makes a decision based on its own measurements, is considered. By contrast, in centralized and decentralized detection, a network of sensors is considered [7]. In centralized detection, the sensors simply send all their raw data to the fusion center for decision making. This may be unnecessary at times and requires constant communication between the sensors and the fusion center. Thus, in decentralized detection, the sensors have their own local decision rules that define what and when to send to the fusion center [8], which combines the information it receives from the sensors. The sensors might not always send something, and when they send, they do not necessarily send their measurements as such but some function of the measurements.

In the rest of this thesis, decentralized detection is disscussed instead of centralized detection, since a centralized setting can be thought of as a special case of a decentralized setting. The fusion center has its own decision rule which determines how to make a decision about the absence or presence of the target based on the information received from the sensors. This is illustrated in Figure 2. Signal from the target at sensor i is denoted by A_i, $\mathbf{x}_i$ are the measurements of sensor i, and $f(\mathbf{x}_i)$ is its function of the measurements that is sent to the fusion center if the sensor decides the target is present. In this representation, the left switch is closed when the target is present in the network, and the right switch is closed when the corresponding sensor decides the target is present. Thus, a sensor's objective is to close the right switch when the left switch is closed, and vice versa. A sensor makes this decision based on its noise-corrupted measurements.

Classical detection could, in principle, be employed in a network of sensors. Then, there would be no fusion center, and any single sensor could set off the alarm. The performance of the system would be worse than in a decentralized system since all information would not be used in the decision making. This can be intuitively understood if sensors are thought to gather evidence of the target. If one sensor has conclusive evidence of the target or several sensors have a weak piece of evidence in a decentralized setting, the fusion center sets off the alarm. In a classical type of system the situation where several sensors have weak, but not conclusive, evidence would not set off the alarm. One might want to compensate for this by setting the sensors to give an alarm already if it has some weak evidence, but this would also increase the rate of false alarms.

In early papers that discussed decentralized detection, mainly quantization of measurements or likelihood ratios was considered, and sensors sent some information to the fusion center all the time [2, 9]. From the viewpoint of total energy

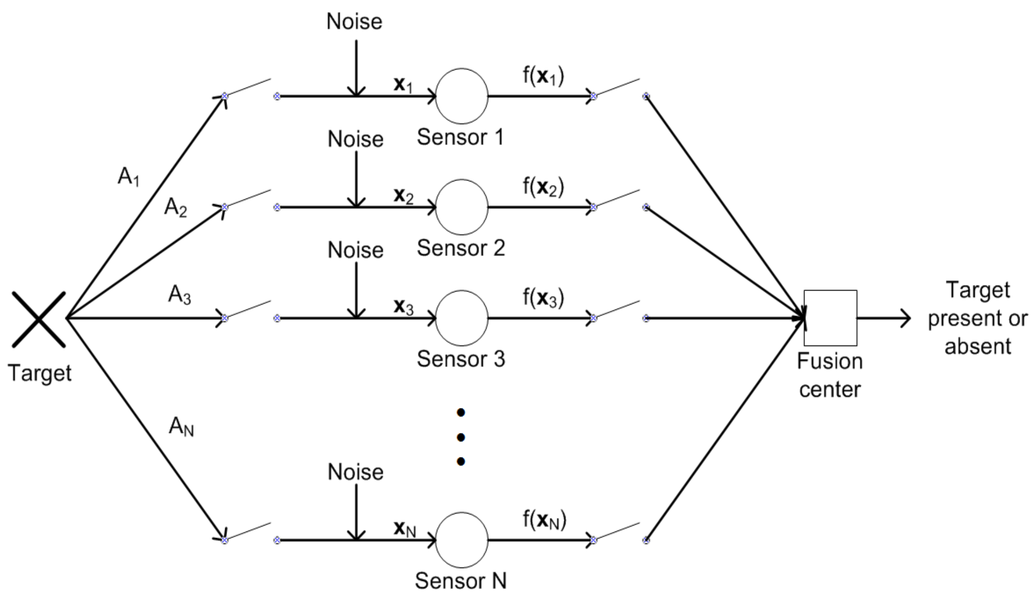

Figure 2: Setup of a decentralized detection scenario.

consumption of the sensors, quantization of measurements which are transmitted at a low data rate is not significant. Packet lengths used by the sensors' radios are typically longer than the amount of bits required to accurately represent the function of measurements in question, and communication protocols often require transmission of side information [10, 11]. Additionally, starting a transmission with radio hardware may consume significant amounts of energy [12]. Thus, in this thesis, energy efficiency is considered by reducing the number of transmissions, not the size of transmissions. The loss in performance due to quantization also becomes insignificant quickly as the number of quantization levels increases from the minimum of two [7].

3.2 Censoring

Saving the limited battery power of the sensors is important, since energy consumption is related to the lifetime of the network. If the null hypothesis is significantly more likely, as in the main scenario (Chapter 1.2), sending messages saying the target is absent all the time is not informative. Large amounts of energy can be saved by not sending anything from the sensors to the fusion center when the target is not observed. This is called censoring.

In considering what the sensors should send to the fusion center, when they send, an assumption is made. Measurements are assumed to be conditionally independent from all other sensors' measurements, conditioned on both hypotheses. This assumption is satisfied under the null hypothesis for local noise [13], especially if the sensors are not too close to each other. This assumption might not hold very well when the target is present in some cases [14]. However, this assumption is made to keep the expressions tractable. The complexity of designing optimal detection algorithms becomes an extensive problem without this assumption [15].

Conditionally independent measurements lead to computing [16] and censoring [17] likelihood ratio tests at the sensors. Each sensor should compute its local likelihood ratio according to equation (1), i.e., assuming N sensors in the network, for each sensor i, $i = 1 \dots N$, the likelihood ratio is

$$\Lambda_i(\mathbf{x}_i) = \frac{l(\mathbf{x}_i; H_1)}{l(\mathbf{x}_i; H_0)}, \tag{3}$$

where $\mathbf{x}_i$ are the conditionally independent and identically distributed measurements of sensor i.

This likelihood ratio (3) should be sent to the fusion center only if it is large enough. It is not obvious that censoring the likelihood ratio when it is smaller than some threshold value is optimal. It is not even obvious that the optimal censoring region should be a single interval. Fortunately, it has been shown that under some mild conditions, the optimal censoring interval is a single interval which has a lower threshold of zero [17]. Thus, for each sensor i, the local decision rule is

$$\phi_i(\mathbf{x}_i) = \begin{cases} \Lambda_i(\mathbf{x}_i), & \text{if } \Lambda_i(\mathbf{x}_i) \geq \gamma_i, \\ \rho_i, & \text{if } \Lambda_i(\mathbf{x}_i) < \gamma_i, \end{cases} \tag{4}$$

where γ_i is the threshold to which the likelihood ratio is compared, and ρ_i is the value associated with the censoring region. This value associated with the censoring region, ρ_i, is needed later when a decision rule for the fusion center is designed. Notice that, according to the censoring principle, nothing is sent to the fusion center if $\Lambda_i(\mathbf{x}_i) < \gamma_i$, although ρ_i is a value associated with this region of $\Lambda_i(\mathbf{x}_i)$. The threshold, γ_i, can be used to adjust the sensitivity of the sensor i, determining how large the measurements must be before making a local decision of presence of the target and sending the likelihood ratio to the fusion center.

The fusion center has a decision rule which it uses to decide if the target is present or absent in the network. Given the local sensor decision rules (4), the optimal decision rule for the fusion center is [17]

$$\phi_0(\phi_1, \dots, \phi_N) = \begin{cases} 1, & \text{if } \prod_{i=1}^{N} \phi_i(\mathbf{x}_i) \geq \gamma_0, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where 1 and 0 correspond to deciding the target is present or absent for the network, respectively, and γ_0 is the decision threshold for the fusion center that can be used to adjust the sensitivity of alarms. Factors of $\prod_{i=1}^{N} \phi_i(\mathbf{x}_i)$ which are not received from sensors correspond to the censored likelihood ratios. As defined in equation (4), they correspond to ρ_i. They are evaluated at the fusion center as

$$\rho_i = \frac{\Pr\{\Lambda_i < \gamma_i; H_1\}}{\Pr\{\Lambda_i < \gamma_i; H_0\}}. \tag{6}$$

However, there is a problem if this sceme is applied to the main scenario (Chapter 1.2). Distribution of the measurements under the target hypothesis is unknown, since the distance of the target from each sensor is unknown. I.e., $l(\mathbf{x}_i; H_1)$ is not completely known since it depends on the loudness of the target and the distance of the target from sensor i. The form of $l(\mathbf{x}_i; H_1)$ is assumed to be known, but at least one parameter is not known. Therefore, the likelihood ratios at the sensors, and the unreceived, censored likelihood ratios at the fusion center cannot be evaluated. However, the distribution of background noise, $p(\mathbf{x}_i; H_0)$, is assumed to be completely known. This means the estimates of mean (11) and variance (12) in Chapter 4.2 are assumed to be so good that their errors can be ignored.

3.3 Maximum likelihood detection

To overcome the problem of not knowing $p(\mathbf{x}_i; H_1)$ completely, the unknown parameter can be estimated using the maximum likelihood principle, and this estimate can be used in the likelihood ratio (3). This approach to hypothesis testing is also called the generalized likelihood ratio test [5]. In maximum likelihood detection, the maximum likelihood estimate of the unknown parameter in $p(\mathbf{x}_i; H_1)$ is computed at each sensor [13].

Maximum likelihood estimation is widely used in practical estimation problems, since it is relatively simple and can often be found quite easily. In general, the maximum likelihood estimate of a parameter μ of a probability density function $p(\mathbf{x})$, where $\mathbf{x}$ are samples from the distribution depending on μ, is the value of μ that maximizes $p(\mathbf{x})$ for a fixed $\mathbf{x}$. This can be understood intuitively as the value of μ that is most likely to have produced the observations $\mathbf{x}$. However, for a finite number of observations, the maximum likelihood estimator has no optimality properties. [20]

Maximum likelihood estimation procedure can be applied here, since $l(\mathbf{x}_i; H_1)$ and $l(\mathbf{x}_i; H_0)$ are essentially probability density functions, as discussed in Chapter 2.2. Assume that $l(\mathbf{x}_i; H_1)$ and $l(\mathbf{x}_i; H_0)$ differ only in one parameter, their expected value, μ_i. Under H_0, the parameter is assumed to be known, $\mu_i = \mu_{0,i}$. This means the sensors are assumed to know the expected value of their measurements, when there is no target but only background noise. Notice that in reality $\mu_{0,i}$ has to be estimated, too, with the estimator given in equation (11). This estimate is assumed to be obtained from such a large amount of data that the estimation error can be ignored, and $\mu_{0,i}$ is assumed to be known. Under H_1, the parameter μ_i is unknown, but it is assumed to be larger than $\mu_{0,i}$. This is a natural assumption since the additional signal from the target increases the expected value of the measurements. The maximum-likelihood estimate of $l(\mathbf{x}_i; H_1)$ can be found by maximizing $l(\mathbf{x}_i; H_1)$ over the parameter μ_i. Therefore, instead of using the classical likelihood ratio (3) at each sensor, each sensor computes

$$\tilde{\Lambda}_i(\mathbf{x}_i) = \frac{\max\limits_{\mu_i} l(\mathbf{x}_i; H_1)}{l(\mathbf{x}_i; H_0)}, \tag{7}$$

and uses $\tilde{\Lambda}_i(\mathbf{x}_i)$ in the decision rule (4), yielding

$$\tilde{\phi}_i(\mathbf{x}_i) = \begin{cases} \tilde{\Lambda}_i(\mathbf{x}_i), & \text{if } \tilde{\Lambda}_i(\mathbf{x}_i) \geq \gamma_i, \\ \rho_i, & \text{if } \tilde{\Lambda}_i(\mathbf{x}_i) < \gamma_i. \end{cases} \tag{8}$$

The fusion center may use a decision rule similar to the one in equation (5) which multiplies the received and censored likelihood ratios together, i.e.,

$$\tilde{\phi}_0(\tilde{\phi}_1, \ldots, \tilde{\phi}_N) = \begin{cases} 1, & \text{if } \prod_{i=1}^{N} \tilde{\phi}(\mathbf{x}_i) \geq \gamma_0, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

However, there is still the same problem as in Chapter 3.2: the fusion center would need ρ_i, corresponding to censored likelihood ratios, to evaluate the right side of equation (9). The fusion center cannot evaluate it from equation (6) since it does not know the distribution under H_1. Unlike the sensors, it cannot estimate the unknown parameter in the distribution under H_1, since the fusion center does not have the measurements of sensor i. It is proposed in [13] that the fusion center should set $\rho_i = \gamma_i$ to provide robustness to uncertainty in the unknown parameter of the distribution under H_1. This requires that the fusion center knows γ_i so the sensors must inform the fusion center of the threshold they use in their decision rule (8).

4 Decentralized detection of vehicle noise in a forest

Whereas the two previous chapters dealt mainly with the theory of detection, this chapter focuses on applying the theory to the main scenario. This chapter considers in detail how the sensors and the fusion center should process their information. Two different approaches are proposed for both the sensors and the fusion center.

4.1 Introduction

From viewpoint of a single sensor, the problem of the main scenario (Chapter 1.2) is essentially a detection of an unknown but positive mean shift in noise. This is exactly the composite hypothesis testing problem described in Chapter 2.3. Therefore, it would make sense to treat the problem at the sensor as a composite hypothesis testing problem and to design some associated fusion rule for the fusion center. As discussed in Chapter 3, according to the censoring paradigm, the sensors consider whether or not to send information to the fusion center. This can be interpreted as a local decision on the absence or presence of the target, but a single sensor deciding the target is present does not start an alarm in the network. When a sensor decides the target is present, it sends some function of its measurements to the fusion center. The decision making of the sensors in the network is discussed in Chapter 4.3.

The fusion center needs some rule according to which it uses the received information to make the final decision on absence or presence of the target. The decision rules for the fusion center that are optimal in some sense, some of which were discussed in Chapter 3, often include an assumption that all sensors observe the same phenomena independently. In the main scenario, the sensors are spread across such a wide area that this assumption is not valid when the target is present. If this were taken into account, the tractability of the expressions for the decision rules might become an issue. Regardless of that, some sort of decision rule for the fusion center has to be designed. Decision rules for the fusion center are discussed in Chapter 4.4.

4.2 Practical considerations

The acoustic sensors used are assumed to be simple omnidirectional microphones which measure sound pressure. The measurements considered in this thesis are measurements of effective sound pressure. Effective sound pressure is the root mean square over a set of instantaneous sound pressure measurements. This set of instantaneous sound pressure measurements is assumed to be taken over a time period, which is short enough that the characteristics of the background noise and target signal stay approximately the same, but long enough that it encompasses several wavelengths in the frequency range of interest. For example, this time period could be 200 ms.

The acoustic background noise, which affects the instantaneous sound pressure measurements, consists of sounds from a large number of sources such as rustling of leaves in the surrounding trees. In other words, the background noise is a sum of many sources of acoustic noise. The central limit theorem states that the sum of several independent random variables converges to a normal distribution as the number random variables in the sum grows to infinity [18]. The individual random variables in the sum need to have a finite expected value and a finite variance, but as long as these are finite, the distributions of the random variables can be arbitrary. Therefore, according to the central limit theorem, the background noise of the instantaneous sound pressure measurements can be approximated by a normal distribution.

As discussed above, the effective sound pressure measurements are root mean squares of sets of normal distributed random variables. Therefore, they are chi distributed. Again invoking the central limit theorem, if there are enough measurements in the set, the chi distribution can be approximated with a normal distribution. In the design of the sensors' decision rules, the acoustic background noise in the forest is modeled by a normal distribution with mean $\mu_{0,i}$ and variance σ_i^2. I.e., each effective sound pressure measurement, henceforth just measurement, in $\mathbf{x}_i$ at sensor i, $x_i[k], k = 1 \ldots K$, is distributed as

$$x_i[k] \sim \begin{cases} \mathcal{N}(\mu_{0,i}, \sigma_i^2) & \text{under } H_0, \\ \mathcal{N}(\mu_{0,i} + A_i, \sigma_i^2) & \text{under } H_1, \end{cases} \tag{10}$$

where A_i is the unknown mean shift depending on the loudness of the target and its distance from sensor i. While the normality assumption might not hold in, e.g., windy conditions, there is some justification for the assumption [19]. Furthermore, measurements $x_i[1] \ldots x_i[K]$ are assumed to be identically and independently distributed for a fixed i, i.e., for each sensor individually. This means that, under each hypothesis, each measurement in the set $x_i[1] \ldots x_i[K]$ at sensor i is drawn from the same normal distribution having the same $\mu_{0,i}$, σ_i^2, and A_i. If the target moves between the measurements, the target signal strength A_i changes and, strictly speaking, the assumption does not hold anymore. However, the K measurements are assumed to be taken during such a short time interval that the change in A_i is negligible.

Naturally, the sensors do not know $\mu_{0,i}$ and σ_i^2, but each sensor can estimate them by taking measurements when the network is initialized. If a sensor takes K measurements $x[k]$, $k = 1 \ldots K$, it can estimate μ as

$$\hat{\mu}_{0,i} = \frac{1}{K} \sum_{k=1}^{K} x[k] \tag{11}$$

and σ_i^2 as

$$\hat{\sigma}_i^2 = \frac{1}{K-1} \sum_{k=1}^{K} (x[k] - \hat{\mu}_{0,i})^2. \tag{12}$$

It can be shown that these estimators for the expected value and variance of a normal distribution are actually mimimum variance unbiased estimators for these parameters [20]. This means they have the smallest variance among all unbiased estimators of these parameters. Thus, the expected value of the squared error of these estimates is the smallest among all unbiased estimators of these parameters.

4.3 Sensors' decision rules

Two different schemes for processing the measurements in sensors are presented in this chapter. The first scheme is based on maximum likelihood estimation, and the second scheme is derived from classical detection theory.

4.3.1 Maximum likelihood scheme

If the maximum likelihood scheme of Chapter 3.3 is to be used in the detection problem, each sensor needs to compute the maximum likelihood estimate of their likelihood ratio (7). Since the measurements are assumed to be distributed according to equation (10), computing the estimated likelihood ratio (7) includes maximizing

$$l(\mathbf{x}_i; H_1) = \frac{1}{(2\pi\sigma_i^2)^{\frac{K}{2}}} e^{-\frac{1}{2\sigma_i^2}\sum_{k=1}^{K}(x_i[k]-\mu_{1,i})^2} \tag{13}$$

with respect to $\mu_{1,i} = \mu_{0,i} + A_i$, i.e., the expected value of the measurements when the target is present. The sensors do not know $\mu_{1,i}$, but they compute the maximum likelihood estimate of it, so that it can be used in computing the estimated likelihood ratio (7). This estimate is derived next. The general outline of the derivation follows [20, pp. 163–164].

Natural logarithm is a strictly increasing function, so the natural logarithm of a function attains is maximum value at the same point where the original function attains its maximum value. Thus, instead of the likelihood funcion (13), its logarithm can be maximized,

$$\begin{aligned} \ln l(\mathbf{x}_i; H_1) &= \ln\left(\frac{1}{(2\pi\sigma_i^2)^{\frac{K}{2}}} e^{-\frac{1}{2\sigma_i^2}\sum_{k=1}^{K}(x_i[k]-\mu_{1,i})^2}\right) && (14) \\ &= -\ln\left((2\pi\sigma_i^2)^{\frac{K}{2}}\right) - \frac{1}{2\sigma_i^2}\sum_{k=1}^{K}(x_i[k]-\mu_{1,i})^2. && (15) \end{aligned}$$

The extrema of a function are found at the points where the derivative equals zero or at the end points of domain of the function. Finding the maxima of $\ln l(\mathbf{x}_i; H_1)$ with respect to $\mu_{1,i}$ is considered next. The expected value of a normal distribution, $\mu_{1,i}$, can take any values in the range $(-\infty, \infty)$. Clearly $\ln l(\mathbf{x}_i; H_1) \longrightarrow -\infty$ as $\mu_{1,i} \longrightarrow \infty$ or $\mu_{1,i} \longrightarrow -\infty$. The function to be maximized, $\ln l(\mathbf{x}_i; H_1)$, is a logarithm of a probability density function, so $\ln l(\mathbf{x}_i; H_1) \geq 0$. Thus, it has to have at least one maximum inside the endpoint of its domain. The extrema inside the domain are found in the points where the derivative equals zero. Taking the derivative with respect to $\mu_{1,i}$ gives

$$\frac{\partial \ln l(\mathbf{x}_i; H_1)}{\partial \mu_{1,i}} = \frac{1}{\sigma_i^2}\sum_{k=1}^{K}(x_i[k]-\mu_{1,i}) = \frac{1}{\sigma_i^2}\left(\sum_{k=1}^{K}x_i[k] - K\mu_{1,i}\right). \tag{16}$$

Setting this to zero, denoting the estimate of $\mu_{1,i}$ as $\hat{\mu}_{1,i}$, and solving for $\hat{\mu}_{1,i}$ yields

$$\hat{\mu}_{1,i} = \frac{1}{K}\sum_{k=1}^{K}x_i[k]. \tag{17}$$

To make sure this is the maximum and not the minimum, the sign of the second derivative is checked. Differentiating function (16) again with respect to $\mu_{1,i}$ gives

$$\frac{\partial^2 \ln l(\mathbf{x}_i; H_1)}{\partial \mu_{1,i}^2} = -\frac{K}{\sigma_i^2}. \tag{18}$$

Since K and σ_i^2 are positive, equation (18) is always negative. Thus, equation(17) gives the maximum, and replacing $\mu_{1,i}$ with its estimate (17) maximizes equation (13). In this scheme, if a sensor decides the target is present, it sends its estimated likelihood ratio (7) to the fusion center.

There is an important distinction between equations (11) and (17), although both are sample averages. The estimate of the average level of acoustic background noise (11) is computed *once* when the network is initialized and no target is present in the network. If necessary, it may be updated, for instance, if it begins to rain, the background noise level grows, but in any case its value changes rarely. On the other hand, the estimate in equation (17) is computed *in every time step* as the sensor makes a decision on presence or absence of the target.

In addition of maximizing function (13), ρ_i and γ_i in function (8) have to be set. It is not simple to determine them so that they would yield a certain probability of false alarm. In this thesis, a computer simulation is used to determine the parameters to yield a suitably low rate of false alarms. If $\rho_i = \gamma_i$ is set, as suggested in Chapter 3.3, and the same value of ρ_i is set to all sensors, then all sensors are set to the same probability of false alarm assuming the same background noise conditions. In this situation, Matlab [21] script of Appendix A can be used to find a suitable value for the parameter ρ.

4.3.2 Classical composite hypothesis testing approach

With the assumption of normal distributed background noise, the problem at each sensor is essentially that of detecting an unknown but positive mean shift in Gaussian noise. This is a special case of the classical composite hypotesis problem of Chapter 2.3. As discussed in Chapter 2.3, first a Neyman–Pearson test (1), or NP test, is designed as if the value of the mean shift were known, and then the test is modified so that it does not depend on the value of the mean shift. Next, an NP test (1) is derived for this setting. The derivation follows the outline of the derivation in [5, pp. 191–194]. In this case, the NP test decides the target is present if

$$\Lambda_i(\mathbf{x}_i) = \frac{l(\mathbf{x}_i; H_1)}{l(\mathbf{x}_i; H_0)} = \frac{\frac{1}{(2\pi\sigma_i^2)^{\frac{K}{2}}} e^{-\frac{1}{2\sigma_i^2}\sum_{k=1}^{K}(x_i[k]-(\mu_{0,i}+A_i))^2}}{\frac{1}{(2\pi\sigma_i^2)^{\frac{K}{2}}} e^{-\frac{1}{2\sigma_i^2}\sum_{k=1}^{K}(x_i[k]-\mu_{0,i})^2}} > \gamma_i, \tag{19}$$

since the joint probability density function of $\mathbf{x}_i$ under each hypothesis is the product of the marginal probability density functions of the measurements $\mathbf{x}_i[k]$. To be exact, $\mu_{0,i}$ should be replaced in the equation (19) with its estimate, $\hat{\mu}_{0,i}$. However, the value of this estimate is assumed to be so close to the true value that its true value can be used in the equation.

Combining the exponentials and canceling out common terms yields

$$e^{-\frac{1}{2\sigma_i^2}\left(-2A_i\sum_{k=1}^{K}x_i[k]+K(A_i^2+2\mu_{0,i}A_i)\right)} > \gamma_i. \tag{20}$$

Taking the natural logarithm of both sides gives

$$-\frac{1}{2\sigma_i^2}\Big(-2A_i\sum_{k=1}^{K}x_i[k]+K(A_i^2+2\mu_{0,i}A_i)\Big) > \ln(\gamma_i). \tag{21}$$

Rearranging the equation yields

$$A_i\sum_{k=1}^{K}x_i[k] > \sigma_i^2\ln(\gamma_i)+\frac{K}{2}(A_i^2+2\mu_{0,i}A_i). \tag{22}$$

Since the target signal A_i is positive at each sensor, both sides can be divided by A_i, and the inequality still holds, resulting in

$$\sum_{k=1}^{K}x_i[k] > \frac{\sigma_i^2}{A_i}\ln(\gamma_i)+\frac{K}{2}(A_i+2\mu_{0,i}). \tag{23}$$

Scaling by $1/K$ yields to the test statistic

$$T(\mathbf{x}_i)=\frac{1}{K}\sum_{k=1}^{K}x_i[k] > \frac{\sigma_i^2}{KA_i}\ln(\gamma_i)+\frac{1}{2}(A_i+2\mu_{0,i})=\gamma_i', \tag{24}$$

where γ_i' denotes the threshold to which the test statistic is compared.

The test statistic does not depend on A_i, but the threshold appears to depend on it. If that would really be the case, the test could not be implemented, since A_i would have to known beforehand, and it was assumed to be unknown. The distribution of a single measurement, $x_i[k]$, when the target is not present, i.e., under H_0, does not depend on A_i (10). The test statistic is just a scaled sum of those measurements. Therefore, the distribution of the test statistic under H_0 does not depend on A_i, either. This should be intuitively clear because the distribution of the noise cannot be dependent on the loudness of some target, which is not even present in the network. Also, because the test statistic is a linear combination of normal distributed variables, it is normal distributed itself. Distribution of $x_i[k]$ under H_0 is known (10), so the parameters for the distribution of $T(\mathbf{x}_i)$ under H_0 can be calculated. The expected value of $T(\mathbf{x}_i)$ under H_0 is

$$\begin{aligned}\mathrm{E}(T(\mathbf{x}_i);H_0) &= \mathrm{E}\left(\frac{1}{K}\sum_{k=1}^{K}x_i[k]\right)=\frac{1}{K}\sum_{k=1}^{K}\mathrm{E}(x_i[k]) \quad (25)\\ &= \frac{1}{K}K\mathrm{E}(x_i[k])=\mathrm{E}(x_i[k])=\mu_{0,i}. \quad (26)\end{aligned}$$

Similarly, the variance of $T(\mathbf{x}_i)$ under H_0 is

$$\mathrm{Var}(T(\mathbf{x}_i);H_0) = \mathrm{Var}\left(\frac{1}{K}\sum_{k=1}^{K}x_i[k]\right) = \frac{1}{K^2}\sum_{k=1}^{K}\mathrm{Var}(x_i[k]) \tag{27}$$

$$= \frac{1}{K^2}K\mathrm{Var}(x_i[k]) = \frac{1}{K}\mathrm{Var}(x_i[k]) = \frac{\sigma_i^2}{K}. \tag{28}$$

Thus, when the target is not present, $T(\mathbf{x}_i) \sim \mathcal{N}(\mu_{0,i}, \frac{\sigma_i^2}{K})$.

Now the probability of false alarm can be related to the threshold of the test, i.e.,

$$P_{FA} = \Pr\{T(\mathbf{x}_i) > \gamma_i'; H_0\} = \Pr\left\{\frac{1}{K}\sum_{k=1}^{K}x_i[k] > \gamma_i'; H_0\right\} \tag{29}$$

$$= \Pr\left\{\frac{\frac{1}{K}\sum_{k=1}^{K}x_i[k]-\mu_{0,i}}{\sqrt{\frac{\sigma_i^2}{K}}} > \frac{\gamma_i'-\mu_{0,i}}{\sqrt{\frac{\sigma_i^2}{K}}}; H_0\right\}. \tag{30}$$

The transformed test statistic is distributed according to the standard normal distribution,

$$\frac{\frac{1}{K}\sum_{k=1}^{K}x_i[k]-\mu_{0,i}}{\sqrt{\frac{\sigma_i^2}{K}}} \sim \mathcal{N}(0,1). \tag{31}$$

Therefore, P_{FA} (30) can be expressed in terms of the complementary cumulative distribution function of the standard normal distribution, the Q-function [22]. The Q-function is defined as

$$Q(x) = 1-\Phi(x), \tag{32}$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution.

Now P_{FA} (30) can be written as

$$P_{FA} = \mathrm{Q}\left(\frac{\gamma_i'-\mu_{0,i}}{\sqrt{\frac{\sigma_i^2}{K}}}\right). \tag{33}$$

Solving for γ_i' gives

$$\gamma_i' = \sqrt{\frac{\sigma_i^2}{K}}\mathrm{Q}^{-1}(P_{FA}) + \mu_{0,i}, \tag{34}$$

where Q^{-1} denotes the inverse Q-function, i.e., $Q(Q^{-1}(x)) = x$. A function has an inverse function, if it is a monotonically increasing or decreasing function [23]. The inverse Q-function exists, since Q-function is a monotonically decreasing function. The threshold (34) is now independent of A_i. A desired false alarm rate can be set, remembering from Chapter 2 that decreasing false alarms also decreases the probability of detection, and the corresponding threshold can be solved from equation

(34). Combining equations (24) and (34) yields to a test which decides the target is present if

$$\frac{1}{K}\sum_{k=1}^{K} x_i[k] > \sqrt{\frac{\sigma_i^2}{K}}\mathrm{Q}^{-1}(P_{FA}) + \mu_{0,i}. \tag{35}$$

Since this is actually an NP test (1), it yields the highest probability of detection for a given probability of false alarm.

Similarly as for H_0, it can be shown that under H_1, $T(\mathbf{x}_i) \sim \mathcal{N}(\mu_{0,i} + A_i, \frac{\sigma_i^2}{K})$. Therefore, as was done for the probability of false alarm, the probability of detection can be expressed with a Q-function as

$$\begin{aligned} P_D &= \Pr\{T(\mathbf{x}_i) > \gamma_i'; H_1\} = \mathrm{Q}\left(\frac{\gamma_i' - \mu_{0,i} - A_i}{\sqrt{\frac{\sigma_i^2}{K}}}\right) && (36) \\ &= \mathrm{Q}\left(\frac{\sqrt{\frac{\sigma_i^2}{K}}\mathrm{Q}^{-1}(P_{FA}) + \mu_{0,i} - \mu_{0,i} - A_i}{\sqrt{\frac{\sigma_i^2}{K}}}\right) && (37) \\ &= \mathrm{Q}\left(\mathrm{Q}^{-1}(P_{FA}) - \sqrt{\frac{KA_i^2}{\sigma_i^2}}\right). && (38) \end{aligned}$$

Since the Q-function is a monotonically decreasing function, equation (38) shows that the probability of detection increases with A_i, the strength of the target signal at the sensor in question. This is natural since it is easier to detect a strong signal than a weak signal. Unlike the probability of false alarm, the probability of detection cannot be evaluated beforehand at each sensor since the sensors do not know A_i.

4.4 Fusion center decision rule

This chapter deals with processing of information the fusion center receives from the sensors. The fusion center needs a decision rule which tells how to make a decision on absence or presence of a target based on the information sent by the sensors. Two schemes are proposed, each corresponding to one of the schemes for sensors, discussed in Chapter 4.3.

4.4.1 Maximum likelihood scheme

One viable option for a fusion center processing method is to employ the maximum likelihood distributed detection scheme (Chapters 3.3 and 4.3.1), even if the assumptions discussed in Chapter 4.1 do not hold. In this case, the fusion center uses the fusion rule given in equation (9) as its decision rule. However, there is still the problem of setting a proper threshold γ_0 for the decision rule. It is desirable to set it to yield a certain probability of false alarm, assuming normal distributed acoustic background noise as in Chapter 4.3. To do that, the distribution of $\prod_{i=1}^{N}\tilde{\phi}(\mathbf{x}_i)$ would have to be known. Even if the fact that each $\tilde{\phi}(\mathbf{x}_i)$ is an estimated and possibly censored likelihood would be ignored, the distribution of a product of random variables, albeit independent and normal distributed, is difficult to determine [24]. This makes even approximation of the probability of false alarm complicated.

Regardless of the complexity of determining the probability of false alarm, some value for the threshold γ_0 has to be set. Assume that all sensors use the same threshold $\gamma_i = \gamma$ in their decision rule (8). Furthermore, assume all sensors set the value associated with the censoring region as $\rho_i = \rho = \gamma$, as suggested in Chapter 3.3. A reasonable form for γ_0 would be

$$\gamma_0 = \alpha \rho^N, \tag{39}$$

where α is a factor depending on the number of sensors in the network, the probabilities of false alarms of the sensors, values of the censoring regions, and the desired probability of false alarm for the fusion center. Using the threshold above (39) in the decision rule of the fusion center (9) results in a decision rule that decides the target is present if the product of the likelihoods is α times larger than the product of censored likelihoods. It is not clear how to choose α to yield a certain probability of false alarm, but a suitable value for it can be found from a computer simulation. An example of such a simulation for Matlab [21] is presented in Appendix A.

4.4.2 Combining standard deviations

Determining parameter values for the maximum likelihood scheme discussed in Chapters 4.3.1 and 4.4.1 is a problem that typically requires computer simulations. To overcome this problem, another detection scheme for the fusion center is devised. This scheme is a heuristic approach based on classical detection theory. In this simpler scheme, likelihoods are forgotten, and sensors use a simpler function of the measurements to make a decision. For this setting, it is assumed that the sensors use function (35) as their decision rule and, when they send, they send their normalized measurements. This means that if sensor i decides the target is present it sends

$$y_i(\mathbf{x}_i) = \frac{\frac{1}{K}\sum_{k=1}^{K} x_i[k] - \mu_{0,i}}{\sqrt{\frac{\sigma_i^2}{K}}}, \tag{40}$$

which is the same as the transformed test statistic in equation (31). This value received by the fusion center, $y_i(\mathbf{x}_i)$, tells how many standard deviations away the set of measurements is from the average noise level. The fusion center could, for instance, sum all the received $y_i(\mathbf{x}_i)$, compare the sum to a threshold, and decide the target is present if the sum is large enough, i.e.,

$$\sum_{i=1}^{n} y_i(\mathbf{x}_i) > \gamma_0, \tag{41}$$

where n is the number of received transmissions from the sensors.

In addition to being a simpler scheme, setting a value for γ_0 is easier here than it is for the maximum likelihood scheme. In equation (34), $\mathrm{Q}^{-1}(P_{FA})$ tells how many standard deviations away the measurements must be from the average noise level before a sensor decides the target is present and sends information to the fusion center. Thus, if all sensor thresholds are set to yield the same P_{FA}, setting γ_0 twice as large as $\mathrm{Q}^{-1}(P_{FA})$ results in deciding the target is present if the $y_i(\mathbf{x}_i)$ received from a single sensor is large enough, or if two or more sensors send something. Notice that P_{FA} in $\mathrm{Q}^{-1}(P_{FA})$ means the probability of false alarm for a single sensor, not for the fusion center. If γ_0 is set to $\gamma_0 = 2 \cdot \mathrm{Q}^{-1}(P_{FA})$, as suggested, the probability of false alarm for the fusion center can be approximated quite easily. This approximation is derived next.

Probability that a single sensor would produce a value larger than γ_0 is very small, if P_{FA} for the individual sensors is set to a reasonably small value. For example, already for $P_{FA} = 10^{-3}$ and network size $N = 20$, the probability equals

$$\begin{aligned} \Pr\{y_i(\mathbf{x}_i) > \gamma_0\} &= \Pr\left\{ \frac{\frac{1}{K}\sum_{k=1}^{K} x_i[k] - \mu_{0,i}}{\sqrt{\frac{\sigma_i^2}{K}}} > 2\mathrm{Q}^{-1}(P_{FA}) \right\} && (42) \\ &= \mathrm{Q}\left(2\mathrm{Q}^{-1}(P_{FA})\right) && (43) \\ &= \mathrm{Q}\left(2\mathrm{Q}^{-1}(10^{-3})\right) && (44) \\ &\approx 3.2 \cdot 10^{-10}, && (45) \end{aligned}$$

where the use of Q-function follows from the fact that $y_i(\mathbf{x}_i)$ is a standard normal distributed variable. Since the measurements of the sensors are assumed to be independent, probability that one or more sensors in the network produce a value larger than γ_0 can be written as

$$\begin{aligned} 1 - \Pr\{\text{no large values}\} &= 1 - \left(1 - \Pr\{y_i(\mathbf{x}_i) > \gamma_0\}\right)^N && (46) \\ &\approx N \cdot \Pr\{y_i(\mathbf{x}_i) > \gamma_0\} && (47) \\ &= \mathrm{Q}\left(2\mathrm{Q}^{-1}(P_{FA})\right), && (48) \end{aligned}$$

where the approximation holds for reasonably small values of N and P_{FA}. Thus, for the example above of $P_{FA} = 10^{-3}$ and network size $N = 20$, the probability that one of the sensors in the network gives a value larger than γ_0 is approximately $20 \cdot \mathrm{Q}\left(2\mathrm{Q}^{-1}(10^{-3})\right) \approx 6.4 \cdot 10^{-9}$. This significantly smaller than the probability that the measurements of at least two sensors exceed their local threshold γ_i under H_0 (54). Therefore, the probability that a single sensor would produce a value larger than γ_0 is ignored in the approximation of the P_{FA} of the fusion center. Only the probability of two or more sensors giving a local alarm at the same time is considered. This probability is derived next.

Because γ_0 is set twice as large as the sensors' local thresholds, it does not matter how large $y_i(\mathbf{x}_i)$ the two sensors send to the fusion center since two of any magnitude will give an alarm at the fusion center. The number of sensors which give a false alarm at the same time, M, is a binomially distributed random variable, i.e., $M \sim \mathrm{Bin}(N, P_{FA})$. Therefore, the probability that two or more sensors give a false alarm at the same time can be obtained from the binomial distribution as

$$\Pr\{M \geq 2\} \qquad (49)$$

$$
\begin{aligned}
&= 1 - \Pr\{M=0\} - \Pr\{M=1\} && (50)\\
&= 1 - \binom{N}{0}(P_{FA})^0(1-P_{FA})^{N-0} - \binom{N}{1}(P_{FA})^1(1-P_{FA})^{N-1} && (51)\\
&= 1-(1-P_{FA})^N - NP_{FA}(1-P_{FA})^{N-1} && (52)\\
&= 1-\left(1-10^{-3}\right)^{20} - 20\cdot 10^{-3}\left(1-10^{-3}\right)^{19} && (53)\\
&\approx 1.9\cdot 10^{-4}, && (54)
\end{aligned}
$$

which is several orders of magnitude larger than $6.4\cdot 10^{-9}$. The probabilities given in equations (48) and (49) for $N=20$ as a function of P_{FA} are presented in Figure 3.

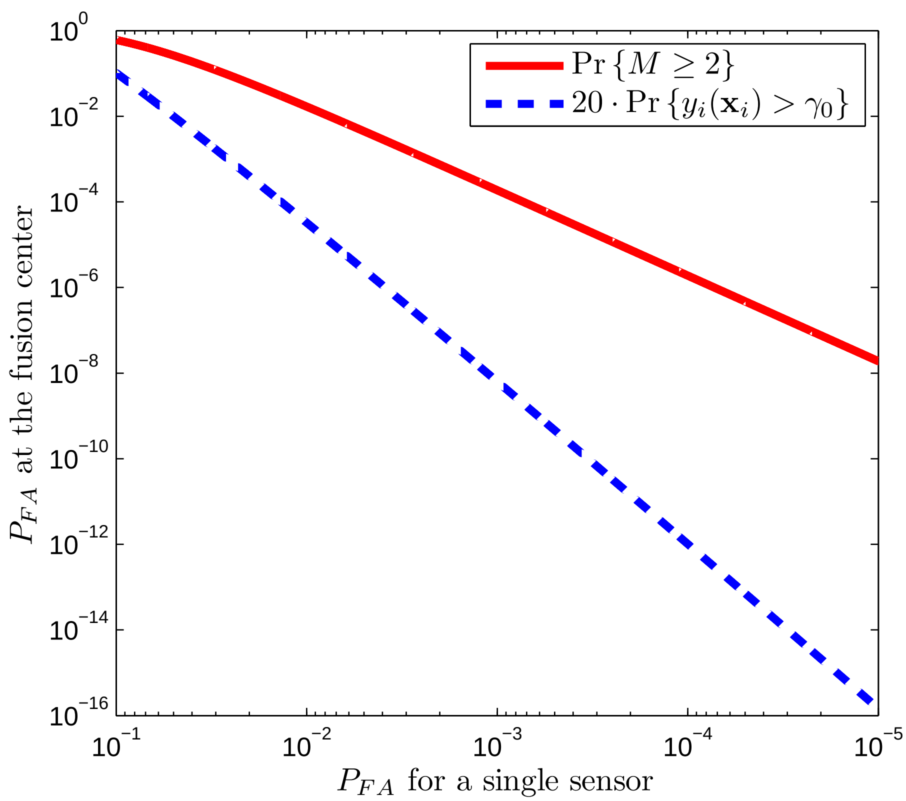

Figure 3: Probabilities of different false alarm types at the fusion center as a function of P_{FA} for network size $N=20$.

On the other hand, fixing the probability of false alarm to a value of $P_{FA}=10^{-3}$, and varying the number of sensors N from 5 to 100 shows that the approximation is valid also for other reasonable network sizes. The same probabilities of one sensor alarm (48) and multisensor alarms (49) of Figure 3 for different network sizes are presented in Figure 4.

This shows equation (51) can be used to approximate the probability of false alarm at the fusion center when γ_0 is set to $\gamma_0 = 2\cdot \mathrm{Q}^{-1}(P_{FA})$. Same kind of approximation can be done when γ_0 is set a little smaller. If the multiplier of

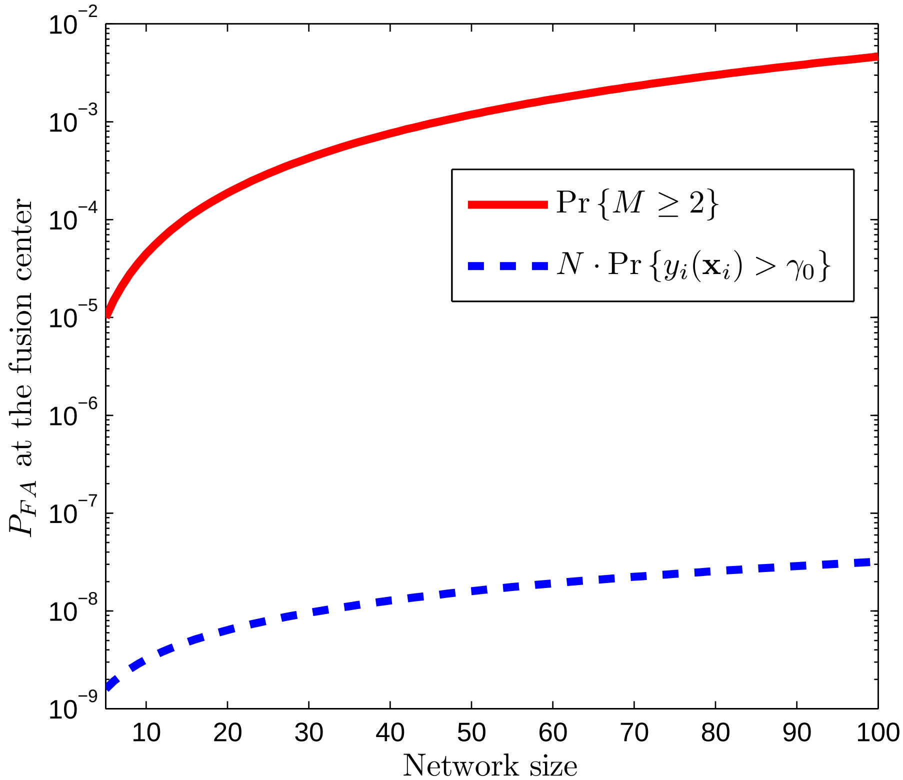

Figure 4: Probabilities of different false alarm types at the fusion center as a function of network size N with probability of false alarm of a single sensor set to $P_{FA} = 10^{-3}$.

$\mathrm{Q}^{-1}(P_{FA})$ is in the range $1 \ldots 2$, the situation is still the same: a large value from a single sensor, or any values from two or more sensors will set off the alarm. The difference is that the large value from a single sensor does not have to be so large as before. Now the probability of noise causing one of the sensors to give such a large value that it sets off the alarm at the fusion center is given by

$$N \cdot \Pr\{y_i(\mathbf{x}_i) > \gamma_0\} = N \cdot \mathrm{Q}\left(\beta \cdot \mathrm{Q}^{-1}(P_{FA})\right), \tag{55}$$

instead of equation (48). The probability of a single large value grows rapidly as the multiplier gets smaller than 2. For $N = 20$, $P_{FA} = 10^{-3}$, and multiplier, denoted by β, in the range $1.3 \ldots 1.7$, the probabilities of the single large values are presented in Figure 5. The probability that two or more sensors give false alarms at the same time for the same values of N and P_{FA} is $1.9 \cdot 10^{-4}$, so the approximation holds, for these values of N and P_{FA}, fairly well when the multiplier is in the range $1.5 \ldots 2$.

Approximating the probability of false alarm at the fusion center when $\gamma_0 > 2 \cdot \mathrm{Q}^{-1}(P_{FA})$ is more difficult. Restricting γ_0 this way is not actually very confining, since the probability of false alarm of the fusion center can be set to any desired value by adjusting the probability of false alarm of the sensors.

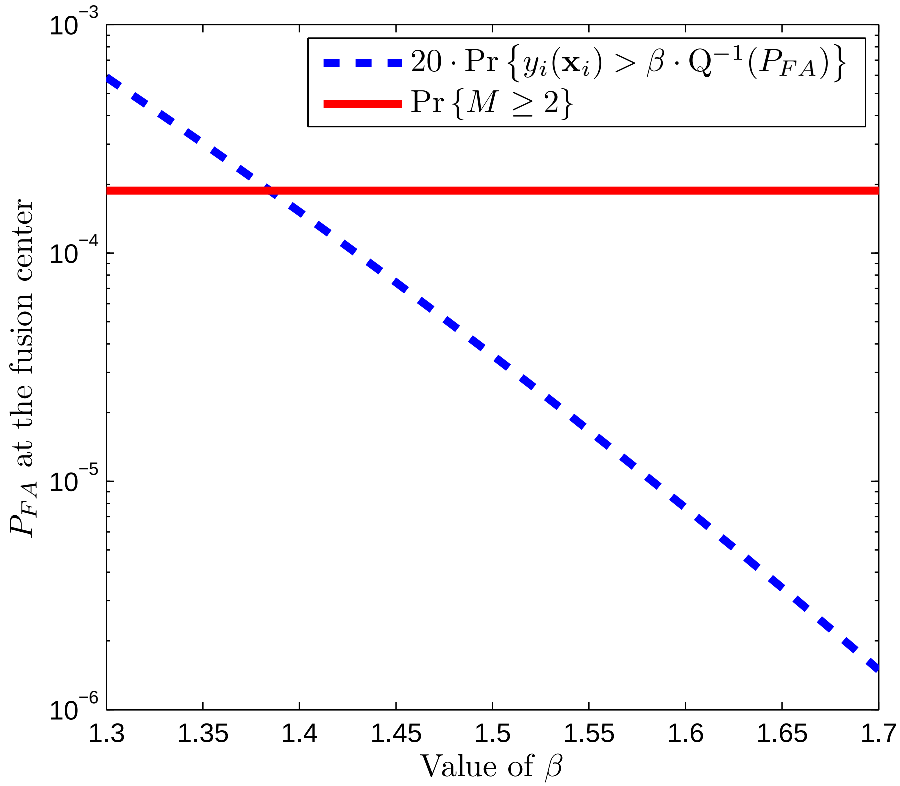

Figure 5: Probabilities of different false alarm types at the fusion center as a function of β for network size $N = 20$ and $P_{FA} = 10^{-3}$.

4.5 Further considerations

In addition to the form of the tests at the sensors and the fusion center, there is a question of whether to set the thresholds of the sensors low and limit the false alarms with a high fusion center threshold, or to set high sensor thresholds and a relatively low fusion center threshold. This is essentially a compromise between energy consumption and performance. If the sensors did not censor anything, the fusion center would have the most information available, and it would be able to make, at least in theory, the best decisions. However, this would quickly deplete the batteries of the sensors since they would have to send information in every time step. If the sensor thresholds were set higher than the fusion center threshold, energy consumption would be minimized, but performance would degrade and this would not be a decentralized setting anymore. In this case, a single transmission from a single sensor would set off the alarm, and fusion center would not be needed at all. Since there is essentially no fusion center to combine the information in this case, because all the sensors use only their own measurements, performance of the system drops. The value of the threshold at the fusion center does not affect the energy consumption of the network. It is a compromise between the probability of false alarm and probability of detection or the delay in detection. Setting a high threshold decreases both the probabilities of false alarm and detection, and vice versa, as discussed in Chapter 2.1.

In the previous chapters, each sensor is assumed to make a decision based on K previous measurements. The choice of K is a free design parameter. Taking several previous measurements into account each time step decreases the probability of the false alarms due to noise spikes, i.e., unusually large values of noise. If the normality assumption of the background noise does not hold, the distribution of the noise might be a long-tailed one, and noise spikes might be larger and more common than assumed. Choosing a K larger than one is especially useful in this situation. However, increasing K means averaging over more and more samples, which results in increasing delay in detection. The alarm at the fusion center will be set off a few time steps later than if there is no averaging. This is the price that is paid for mitigating the effects of noise spikes.

5 Special characteristics of other scenarios

Characteristics of other scenarios akin to the main scenario of Chapter 1.2 are considered in this chapter. Especially, the effects of these characteristics to detection schemes in these scenarios are discussed here.

5.1 Detecting impulses

In the main scenario, the sensor network is trying to detect a vehicle in a forest. The vehicle produces a continuous sound signal so taking measurements, e.g., once in a second is adequate for detection. If the sound produced by the sound source is an impulse, the sensor might miss the impulse if it takes measurements too infrequently. For example, a gunshot is such a short sound that it can well be classified as an impulse [25, figure 1]. In such a scenario, the sound pressure must be monitored very frequently. It might be more practical to measure the sound pressure continuously, instead of taking measurements at certain intervals. The downside is that monitoring continuously depletes the batteries of the sensors faster, which reduces the lifetime of the network. Also, in this type of impulse detection, averaging over several measurements, i.e., choosing K larger than one, might not be a good idea. Averaging smooths the signal, which is undesirable, as the objective is to detect abrupt impulse spikes.

5.2 Scenarios without a fusion center

As discussed in Chapter 4.5, letting a single sensor set off global alarm in the main scenario results in inferior performance. Even if the sensors are deployed far away from each other such that two sensors cannot detect the target at the same time, a fusion center can be useful. Although only one sensor at a time can detect the target, it may be beneficial to have information at the fusion center from the other sensors saying they do not detect any targets. Despite the sensors being quite distant from each other in this scenario, the sensors need to be within radio communication range of each other so that alarms can be transmitted across the network.

On the other hand, a detection system may be designed so that there is no need for a designated fusion center. This is called fully decentralized detection. In this type of system, sensors communicate between themselves to convey information about their measurements or local decision. The sensors try to find a consensus about presence or absence of the target. When enough nodes agree on the target being present, a global alarm is set off. Unlike a conventional decentralized system, a fully decentralized system is not vulnerable to fusion center breakdown. [26]

5.3 Direct communication with the fusion center

If the sensor setup is such that all sensors can communicate directly with the fusion center, there might not be need for large overheads in transmissions. In this case, the size of the transmissions may be heavily dependent on the quantization of the information the sensors send to the fusion center. While in the main scenario using, for example, 3 bits instead of 10 bits to describe the information might not save much energy, as discussed in Chapter 3.1, in the direct communication setup it might bring significant energy savings.

Additionally, if the sensors are arranged, e.g., into a linear or circular array with equal distances between the sensors, the system may give a good estimate on the direction of the target [27]. For this the sensors have to be so close to each other and the target has to be so close to the sensors that all sensors can hear the target.

5.4 Different types of sensors

Sensor types that can detect a target from as long distance as possible should be used for detection. However, other types of sensors may be more suitable for other purposes. E.g., it might make sense to deploy a few cameras to the network of the main scenario for classification purposes. These cameras could do automatic classification of targets [28, 29] or send images of the detected targets out of the network.

If the objective is to detect people, instead of vehicles, in a forest, infrared sensors are a viable option [30]. Unlike acoustic sensors, infrared sensors require line of sight with the target for detection. However, a single person walking in a forest might not make that much sound, so it is not obvious which sensor type would detect the target first. Especially in the case of heavy rain, the level of acoustic background noise can be so high that acoustic sensors might be practically useless in detecting people. It should also be taken into account that nearby thundering will cause all acoustic sensors to give alarms.

5.5 Target localization and tracking

In many scenarios, such as the main scenario, it is desirable to locate a target in addition of detecting it. Some short-range sensors, such as cameras, may be better than acoustic sensors for localizing a target inside the network. Since they have a short range, they cannot detect or localize a target that is at some distance from the network. Thus, it is reasonable to do localization with acoustic or other long-range sensors, too [31].

When localizing a target with acoustic or other long-range sensors, the detection thresholds of the sensors should be lowered from the one used in detection mode. Many localization algorithms likely have better accuracy with more data, albeit somewhat noise-corrupted data. It may not be of great importance anymore to save energy by reducing transmissions from the sensors when a target is detected. Instead, getting an accurate location estimate is far more important in many scenarios.

In addition to running only a localization algorithm, it is usually desirable run a tracking algorithm on top of that. While localization algorithms typically consider only the measurements from the current time instant, tracking algorithms can be used to exploit the time dependence between consecutive location estimates. In the tracking algorithms, the location estimate is based on prediction derived from the previous estimates in addition to measurements. The statistical performance of the refined location estimates given by a tracking algorithm is generally better than that of the raw location estimates given by the localization algorithm. Examples of this type of tracking algorithms are the Kalman filter [20] and its extensions [32].

6 Simulations

A set of simulations was run in the main scenario and in an alternative scenario (see Chapter 6.7.3) to compare the performance of the two proposed schemes. In addition, the effect of different parameters of the schemes were studied. The simulations were run with Matlab R2010a software [21], which is a technical computing environment focusing on numerical computing. Simulation settings and results of the simulations are described in this chapter.

6.1 Simulation scenario

A number of simulations were run to compare the performance of the suggested detection schemes. The simulations are based on a scenario in which a vehicle comes to a network of acoustic sensors located in a forest, stays there for a while, and leaves the network via a different route.

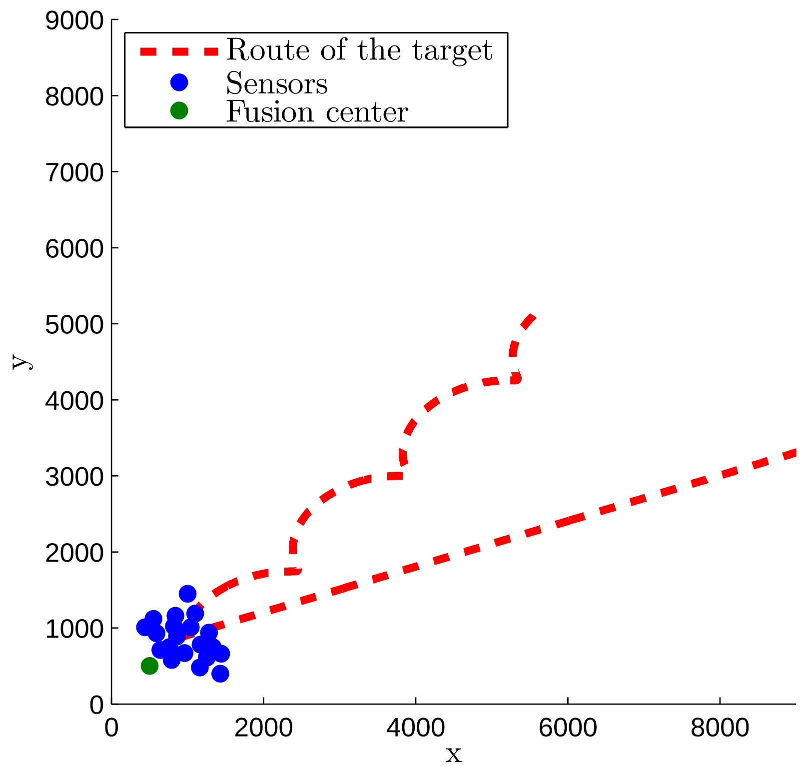

Figure 6: The route of the target in the simulation scenario.

The simulation scenario is shown in Figures 6 and 7. Figure 6 shows the whole route of the target in the simulation scenario. Figure 7 is the same figure zoomed in near the network to better illustrate the grouping of the sensors. There are 20 sensors and a fusion center grouped into an area of ca. 1 km x 1.5 km along a road. In the coordinate system of Figures 6 and 7, one unit corresponds to one meter in

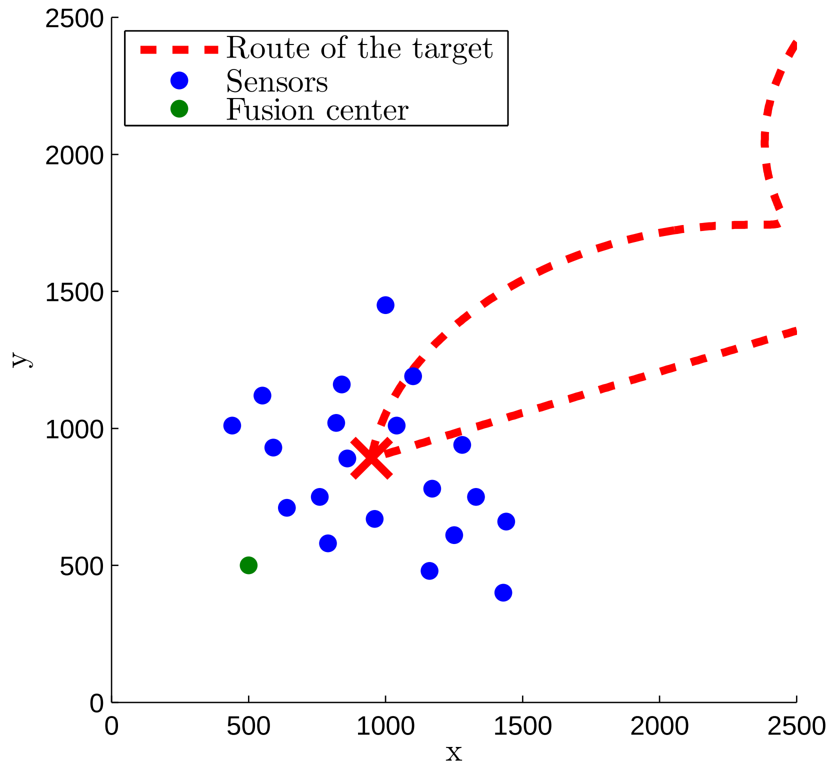

Figure 7: The sensor network of the simulation scenario with the route of the target near the network.

the simulation. The locations of the sensors are marked with blue dots, the location of the fusion center is marked with a green dot, and the route of the target is marked by the red curve. The target enters the network via the upper curve, stops for 1 min 40 s in the place denoted by the red cross, and leaves the network via the lower curve. The simulation lasts for 2000 seconds, i.e., approximately 34 min, and the speed of the target as it moves is ca. 30 km/h.

6.2 Modeling sound propagation in a forest

A model for sound propagation in a forest is needed for the simulations. The model used in the simulations is a simplified version of an ISO standard [36] describing sound propagation outdoors.

To calculate the sound pressure level at a sensor, the distance from the target and the sound power level of the target are needed. In the ISO standard, the following equation is given for the sound pressure level:

$$L_{fT}(DW) = L_W + D_C - A, \tag{56}$$

where $L_{fT}(DW)$ is the continuous downwind octave-band sound pressure level relative to a 20 μPa reference sound pressure, L_W is the sound power level of the sound source relative to a 1 pW reference sound power, D_C is the directivity correction in decibels, and A is the attenuation, in decibels, that occurs during propagation from the target to the sensor. For the simulation, it is assumed that the spectrum of the sound produced by the target is such that most of the energy is concentrated on a narrow frequency band so that only one octave band from $L_{fT}(DW)$ can be taken into account. In general, the attenuation of sound depends on frequency, so this assumption makes the calculation of attenuation simpler. The target is assumed to be an unidirectional sound source, so D_C is set to $D_C = 0$. Although the standard is defined for a sound propagating downwind, wind is not taken into account in the simulations, and the same propagation model is used for sound propagating in all directions. Therefore, the sound pressure level at a sensor due to the target is simplified into

$$L_P = L_W - A. \tag{57}$$

The attenuation A is divided into components in the standard:

$$A = A_{div} + A_{atm} + A_{gr} + A_{bar} + A_{misc}, \tag{58}$$

where A_{div} is the attenuation due to geometrical divergence, A_{atm} is the attenuation due to atmospheric absorption, A_{gr} is the attenuation due to the ground effect, A_{bar} is the attenuation due to a barrier, and A_{misc} is the attenuation due to miscellaneous other effects.

Besides the forest, no other obstacles between a sensor and the target are taken into account, so $A_{bar} = 0$. The miscellaneous other factors are also assumed to be zero, i.e., $A_{misc} = 0$.

Since the objective is to detect a vehicle in the simulation, most of the energy of the sound is assumed to be around the frequency of 100 Hz [37, 38]. This approximation might not be valid with all vehicles, but it is deemed an appropriate simplification. With this assumption, the atmospheric absorption and ground effect can be calculated for the frequency of 100 Hz, and their frequency dependence can be neglected.

In the standard atmospheric absorption is calculated as

$$A_{atm} = \alpha_s d/1000, \tag{59}$$

where α_s is a parameter depending on the frequency of the sound, and d is the distance the sound propagates in the air. For frequencies around 100 Hz, α_s is approximately 0.2, so for, e.g., the distance of 1 km, $A_{atm} = 0.2$ dB. This is such a small value that it is lost under the other inaccuracies of the model. Thus, in the simulation, the atmospheric absorption is set to $A_{atm} = 0$.

For the geometric divergence, there is an equation in the standard that is directly applied in the simulation:

$$A_{div} = 20 \log_{10}\left(\frac{d}{d_0}\right) + 11, \tag{60}$$

where d is the distance the sound propagates, and d_0 is the reference distance $d_0 = 1$ m. The constant 11 in the equation relates the sound power level of an omnidirectional sound source to the sound pressure level at a distance of 1 m [39, p. 88].

Several different methods for calculating the attenuation due to ground effect are given in the standard. In addition of frequency, the ground effect depends on the height of the sensors and the target. However, based on actual measurements in a Finnish forest [40, figure 3 (left)], the attenuation due to ground effect in simulations is set to $A_{gr} = 8$ dB.

As a summary, the sound pressure level L_P[dB] at a sensor due to the target is calculated from the sound power level of the target, and its distance from the sensor as

$$L_P = L_W - A = L_W - (A_{div} + A_{gr}) = L_W - 20\log_{10}(d) - 19. \tag{61}$$

6.3 Wireless communication model

Usually the research on decentralized detection and the related simulations do not consider the communication aspect of the sensor network. As discussed in Chapter 1.1, the sensors are typically connected by a wireless communication channel. To make the simulations more realistic, a model for the communication between sensors is implemented in the simulations. The fundamental function of this model in the simulations is discarding some of the information sent from the sensors to the fusion center.

There are essentially two major reasons why the measurement information from sensors might not received by the fusion center:

- In the physical layer, there may be so low a signal-to-noise ratio (SNR) that the receiving sensor or fusion center cannot properly decode the information from a transmission.
- In the medium access control (MAC) layer, there may be congestion in the transmission medium.

Both of these are taken into account in the model for wireless communication in the network. Since the lifetime of the battery-powered sensors depend on energy consumption, energy-efficiency is of paramount importance in sensor networks. Energy can be saved by limiting the transmit power of the sensors, which results in quite a low SNR at the receiver. Thus, a relatively high percentage of the information sent by the sensors may be lost because of this. Sleep cycles [34] may also be utilized by the sensors to save energy. During sleep periods, a sensor's radio is completely turned off, so it cannot transmit or receive anything during that period. The active periods, when the radio is actually on, may be short, and there may be a lot of traffic in the network during these periods. The information from all the sensors may not make it to the fusion center during an active period in case of congestion in the network.

To represent the loss of detection information from low SNR, a simple approach is applied. Each value a sensor sends to the fusion center is lost with some probability P_{SNR}. This is a baseline for the probability of packet loss, where one packet constitutes one value a sensors sends. In the simulations, it is assumed the sensors are set to transmit at quite a low power and, thus, the value for P_{SNR} is set quite high, $P_{SNR} = 0.02$. This means that 2% of packets are lost regardless of the traffic intensity at MAC layer.

To model the packet loss resulting from congestion in the network, a method from article [35] is used. A contention-based MAC scheme is considered where the sensors, which have a packet to send, compete for the used of the communication channel. The active period, discussed above, is split into S time slots for analysis. Denote the number of sensors, which have a packet to send, as B. At each time slot, each of the B sensors transmit with probability p. If two or more sensors try to send during the same time slot, all their transmissions fail, and they have to try again at later time slots. A packet is lost if it is not successfully transmitted during the active period, i.e., during the S time slots. According to [35], the probability that a packet is lost due to congestion in the communication channel, for each packet individually, can be approximated by

$$\Pr\{\text{packet lost}\} = \left((1-p) + p(1-(1-p)^{B-1})\right)^S. \tag{62}$$

The optimal value for p under the assumptions in [35] is approximately

$$p_{opt} \approx \frac{2}{B+2}. \tag{63}$$

However, the sensors do not know the number of sensors that have something to send, B. In the simulations, sensors assume the worst-case scenario and use the number of sensors in the network, N, instead of B, in equation (63). The number of time slots, S, used in simulations is $S = 100$. The shape of function (62) with $S = 100$, network size $N = 20$, and $p = \frac{2}{N+2} = \frac{2}{20+2} = \frac{1}{11}$ is shown in Figure 8.

This approach to modeling the nonideal communication between sensors is actually appropriate only for sensors communicating directly with the fusion center, and the networking aspect, different transmit power levels, different number of hops of different packets in the network, or different distances between the sensors are not modeled. A more elaborate scheme would have to be devised to account for these. This simple model used in simulations is justified by that fact that the primary interest is on the percentage of lost packets. It is not relevant, which packets are dropped and which are received.

6.4 Default settings for the simulation

A sound power level for the target was set from [4, figure 4]. The value chosen for the sound power level is 115 dB which corresponds to a sound pressure level of 104 dB at a distance of 1 m according to the rule of thumb [39]

$$L_{P1m}\ [\text{dB relative to } 20\ \mu\text{Pa}] = L_W\ [\text{dB relative to } 1\ \text{pW}] - 11, \tag{64}$$

where L_{P1m} denotes the sound pressure level at a distance of 1 m from the source, and L_W denotes the sound power level of the source.

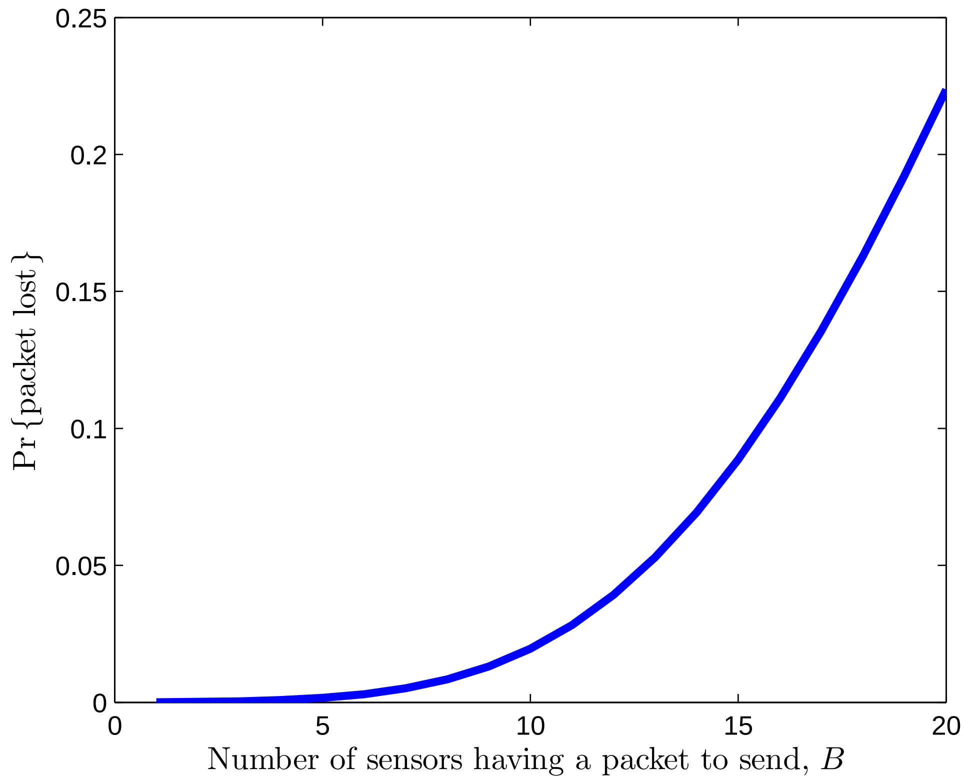

Figure 8: Probability of packet loss for a single packet as a function of the number of sensors having something to send for the settings of simulation scenario described in Chapter 6.1.

The acoustic background noise is modeled by a normal distribution, see Chapter 4.2. Therefore, two parameters have to be set for the noise, mean and variance. The mean was set to 1.1 mPa, which corresponds to a sound pressure level of 35 dB, and the variance was set to $10^{-7}\,\mathrm{Pa}^2$. It is assumed the sensors have very good estimates of the mean and the variance of the background noise, i.e., the estimation errors are ignored, and the sensors use the true values instead.

All sensors were set to the same level of probability of false alarm in the simulations. For the combining standard deviations scheme, the probability of false alarm for the fusion center is fully determined by the false alarm rate of the sensors. Multiplier of the fusion center threshold for the combining standard deviations scheme, β, is set to its default value of two. For a desired level of P_{FA} at the fusion center, the P_{FA} of the sensors was searched by iteration from function (52).

For the maximum likelihood scheme, the thresholds for the sensors and the fusion center were searched by iteration with the Matlab script of Appendix A. First, the threshold for the sensors, ρ, was set to yield approximately the same P_{FA} at the sensors as in the combining standard deviations scheme. After ρ was set, the threshold for the fusion center, α, was searched which yields approximately the desired P_{FA}.

K is set to one for all sensors, i.e., sensors use only their current measurement at each time instant in their decision-making. The sensors take measurements and run their detection algorithm at intervals of one second. The fusion center also makes a decision of presence or absence of the target once in a second. Some of the settings presented here or in Chapter 6.1 are changed in some of the simulations. The changes are discussed in the chapters discussing results for the corresponding simulation.

6.5 Simulation 1 — Performance at different rates of P_{FA}

In this simulation, the detection performance of the combining standard deviations scheme and the maximum likelihood scheme is compared. As described in Chapter 6.4, in both schemes the sensors were set to the same level of P_{FA}, and the same level of P_{FA} was set to the fusion center. The simulation was run with three different rates of P_{FA} at the fusion center, 10^{-4}, 10^{-5}, and 10^{-6}. These rates were chosen since they represent a reasonable compromise between number of false alarms and detection sensitivity for the main scenario. The detection thresholds corresponding to these rates in both schemes are listed in Table 2.

Table 2: Settings of the fusion center and the sensors in simulation 1.

Setting number	1	2	3
P_{FA} of fusion center	10^{-4}	10^{-5}	10^{-6}
P_{FA} of sensors	$7.29 \cdot 10^{-4}$	$2.30 \cdot 10^{-4}$	$7.27 \cdot 10^{-5}$
ρ	301	878	2640
α	117	295	900

A thousand simulations were run with each false alarm rate for both schemes. The average number alarms was computed from the results for each false alarm rate and for both schemes. A larger number of alarms is considered to be better, although there probably are false alarms included in these alarms. However, these are ignored for two reasons:

- Most of the alarms are actual detections of the target.
- Simulations are run to compare the relative performance of the two schemes, and the fusion center is set to the same P_{FA} for both schemes. The expected number of false alarms is same for both schemes. Thus, the false alarms do not play a large role in comparison of the two schemes.

The results of this simulation are shown in Figure 9. In all three levels of P_{FA}, the combining standard deviations scheme performs significantly better than the maximum likelihood scheme. Although the maximum likelihood scheme is semi-optimal in a more ideal scenario, i.e., where all the sensors are at an equal distance

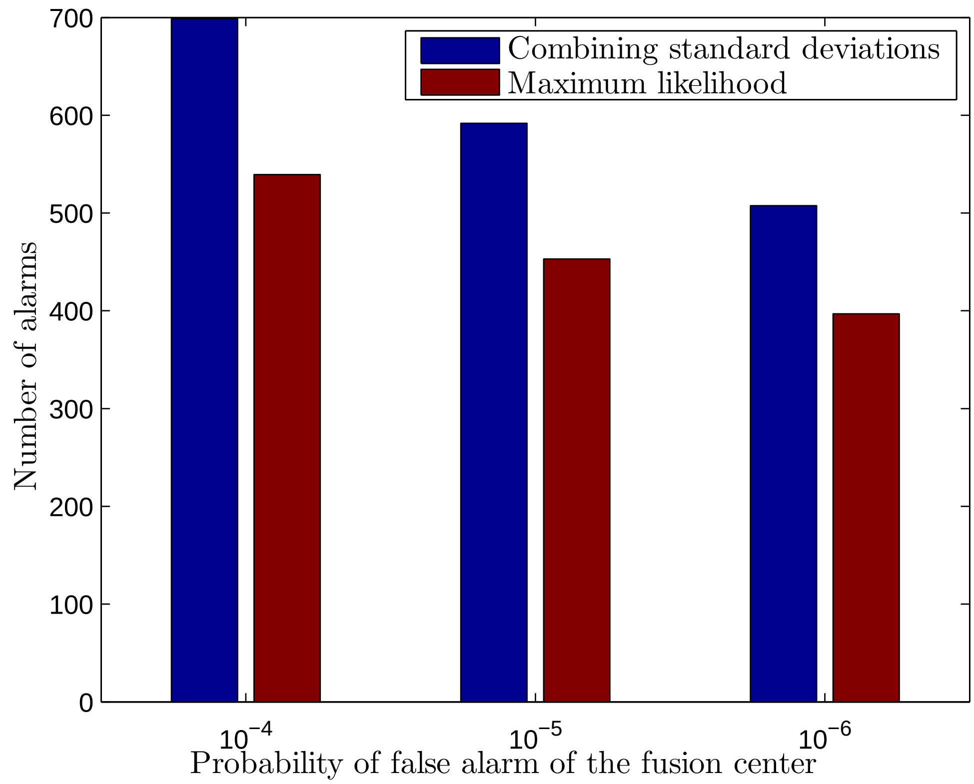

Figure 9: Number of alarms in the simulation scenario for three different rates of P_{FA} for both detection schemes.

from the target, its performance is not good in this type of more practical scenario. This shows it is not robust against deviations from assumptions since it performs clearly worse than an heuristic scheme without optimality properties.

6.6 Simulation 2 — Different noise distribution

The sensors assume in their detection algorithms that the background acoustic noise is normal distributed. In this simulation, the performance of the proposed schemes is studied if this assumption does not hold. The background noise in this simulation is gamma distributed [41] with the same expected value and variance as the normal distributed noise in simulation 1, i.e., expected value $\mu = 1.1\,\mathrm{mPa}$ and variance $\sigma^2 = 10^{-7}\,\mathrm{Pa}^2$. The gamma distribution has two parameters, shape parameter $k_G > 0$, and scale parameter $\theta_G > 0$. Its expected value equals $k_G\theta_G$, and its variance equals $k_G\theta_G^2$. Setting the same expected value and variance for the background noise as in simulation 1 means setting k_G and θ_G such that

$$\begin{cases} k_G\theta_G &= \mu \\ k_G\theta_G^2 &= \sigma^2. \end{cases} \tag{65}$$

Solving for k_G and θ_G yields

$$\begin{cases} k_G &= \frac{\mu^2}{\sigma^2} \\ \theta_G &= \frac{\sigma^2}{\mu}. \end{cases} \tag{66}$$

The probability density function of a gamma distributed variable X is of the form

$$f_G(x) = x^{k_G-1}\frac{e^{-x/\theta_G}}{\Gamma(k_G)\theta_G^{k_G}}, \qquad x \in [0, \infty), \tag{67}$$

where $\Gamma(k_G)$ is the so-called gamma function, $\Gamma(k_G) = \int_0^\infty t^{k_G-1}e^{-t}\mathrm{d}t$. The gamma distribution has a longer tail than the normal distribution, i.e., it is relatively likely to draw values from gamma distribution that are quite far from the expected value. Thus, for a fixed variance, false alarms are more likely in the case where the background noise is gamma distributed than in the case where it is normal distributed. The probability density functions of a normal distribution and a gamma distribution with the parameters used in the simulations are plotted in Figure 10. In the x-axis, μ is the mean value and σ is the standard deviation of the normal distribution. To illustrate better the long tail of the gamma distribution, Figure 11 shows the same two distributions zoomed into an interval $\mu + 3\sigma \ldots \mu + 6\sigma$. There it can be seen that the probability of the background noise causing measurement values that are several standard deviations over the mean is clearly larger under the gamma distribution than under the normal distribution. Setting, e.g., $P_{FA} = 10^{-4}$ for a single sensor in the combining standard deviations scheme and considering only a single measurement corresponds to making a local alarm if the measurement $x > \mu + Q^{-1}(10^{-4})\sigma = \mu + 3.72\sigma$. This is the type of measurement that causes false alarms, so P_{FA} is larger under the gamma distribution.

The simulation was run with the same three settings of Table 2 for both detection schemes as in Simulation 1. The only difference is the gamma distributed noise. The sensors and the fusion center still assume, now incorrectly, the same P_{FA} as before. The true rates of P_{FA} for the sensors and the fusion center were estimated with Matlab script of Appendix A. The settings and the probabilities of false alarm are presented in Tables 3 and 4.

Table 3: Settings and the probabilities of false alarm of the fusion center and the sensors in simulation 2 for the combining standard deviations scheme.

Setting number	1	2	3
Assumed P_{FA} of fusion center	10^{-4}	10^{-5}	10^{-6}
True P_{FA} of fusion center	$3.5 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$
Assumed P_{FA} of sensors	$7.29 \cdot 10^{-4}$	$2.30 \cdot 10^{-4}$	$7.27 \cdot 10^{-5}$
True P_{FA} of sensors	$4.3 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$

As anticipated, the false alarm rates grow significantly in both schemes when the background noise is changed from normal distributed to gamma distributed. There

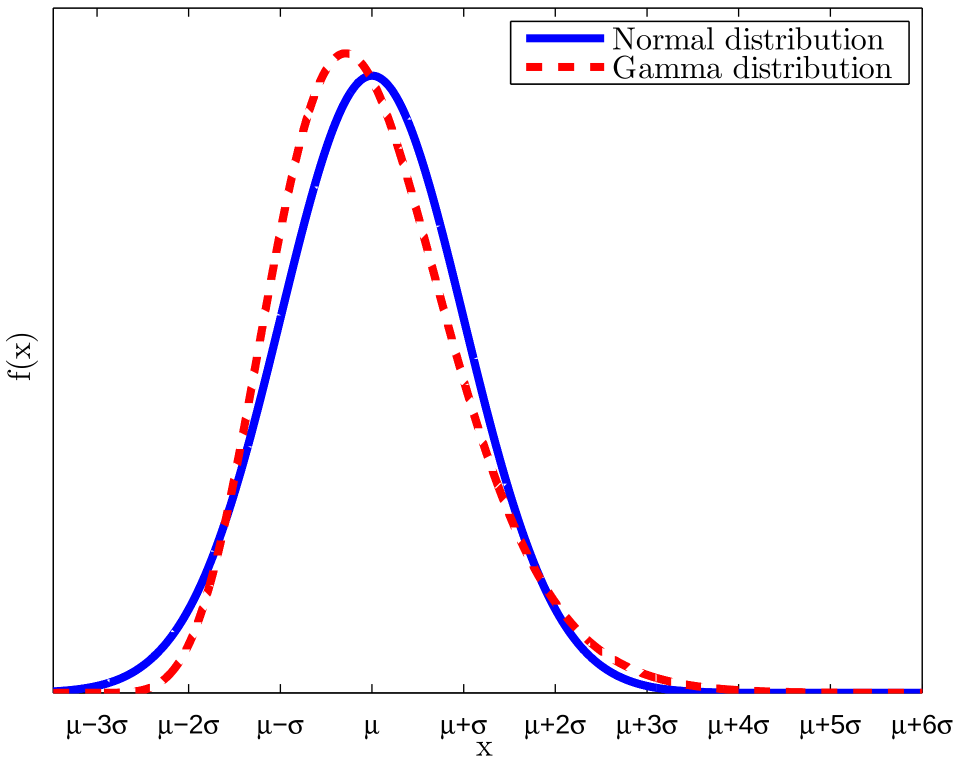

Figure 10: The probability density functions of a normal distribution and a gamma distribution with the parameters used in the simulations.

Table 4: Settings and the probabilities of false alarm of the fusion center and the sensors in simulation 2 for the maximum likelihood scheme.

Setting number	1	2	3
ρ	301	878	2640
α	117	295	900
Assumed P_{FA} of fusion center	10^{-4}	10^{-5}	10^{-6}
True P_{FA} of fusion center	$3.0 \cdot 10^{-3}$	$1.9 \cdot 10^{-3}$	$7.6 \cdot 10^{-4}$
Assumed P_{FA} of sensors	$7.29 \cdot 10^{-4}$	$2.30 \cdot 10^{-4}$	$7.27 \cdot 10^{-5}$
True P_{FA} of sensors	$5.3 \cdot 10^{-3}$	$1.6 \cdot 10^{-3}$	$9.1 \cdot 10^{-4}$

seems to be no large difference in the change of the false alarm rate between the two detection schemes, i.e., both schemes seem to be equally robust or sensitive to this change in the noise distribution. Both schemes are robust in the sense that they still work as intended but, on the other hand, sensitive in the sense that their false alarm rates change significantly from the intended ones.

The same scenario as in simulation 1, now with a different background noise

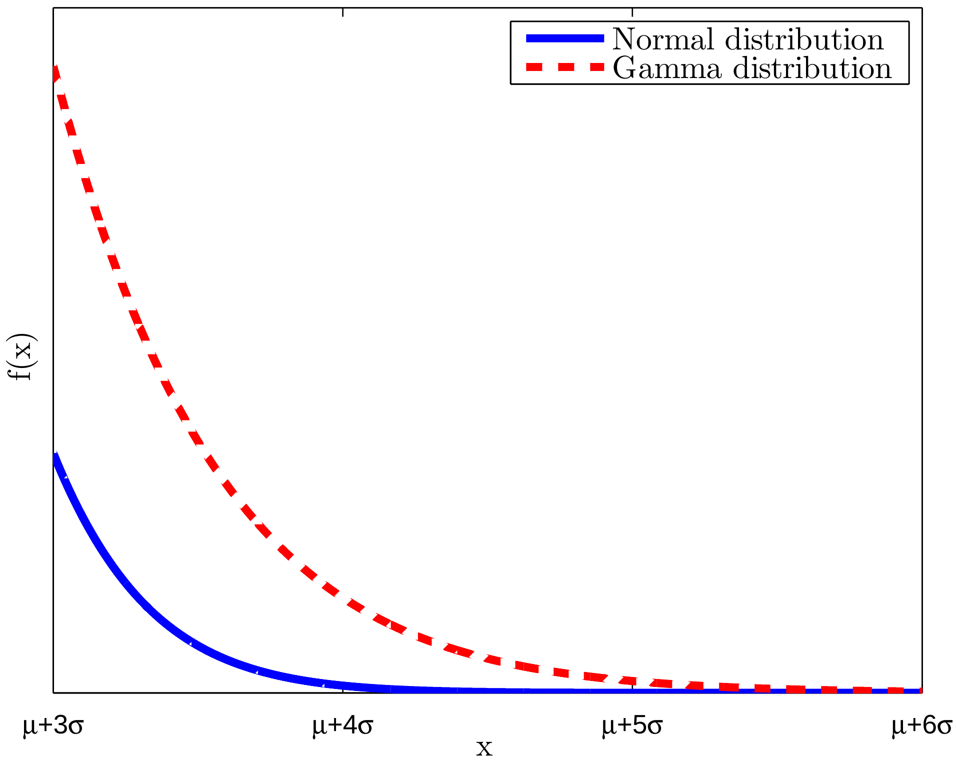

Figure 11: The probability density functions of a normal distribution and a gamma distribution with the parameters used in the simulations in an interval $\mu+3\sigma \ldots \mu+6\sigma$.

distribution, was run a thousand times for each setting for both schemes, and the average number of alarms in each case was computed. The results of simulation 2 are shown in Figure 12. Although there are more false alarms now, the vast majority of the alarms are still valid detections of the target. Therefore, detection performance can be compared by looking at the total number of alarms. It can be seen by comparing Figures 9 and 12 that the number of alarms went up for all settings in both detection schemes as the background noise distribution was changed to a gamma distribution. Also, as can be seen from Tables 3 and 4, the probability of false alarm increased because of the long tail of the noise distribution. There is essentially no change in the relative performance of the two schemes with the corresponding settings. Thus, changing the background noise distribution from normal to gamma distribution resembles the situation of decreasing the thresholds in order to have a larger probability of detection at the cost of having a larger P_{FA}. However, ratio of the number of alarms and P_{FA} is worse in the case of gamma distributed noise. This means that detecting the target in gamma distributed noise is a more difficult problem than detecting it in normal distributed noise.

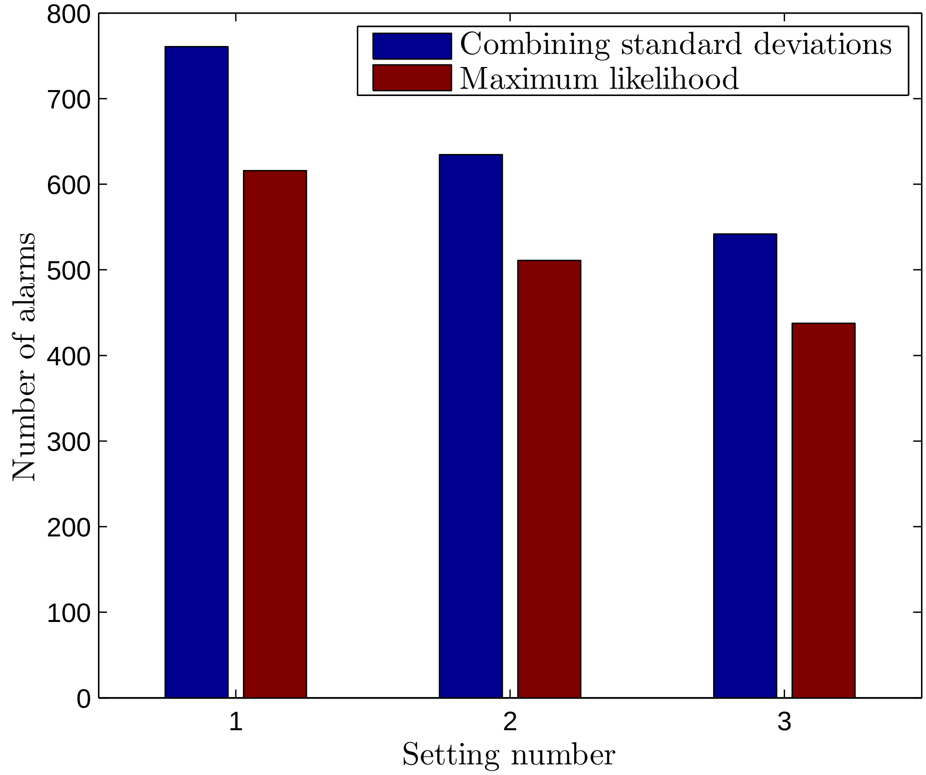

Figure 12: Number of alarms in simulation 2 for three different settings for both detection schemes.

6.7 Simulation 3 — Binary sensors

6.7.1 Introduction

Quantization of the data the sensors send to the fusion center is discussed in Chapter 3.1. It is argued there that the energy savings achieved by quantizing the measurements are likely to be minor. In this simulation, the performance of the combining standard deviations scheme is studied in the situation, where the sensors quantify their measurements to a single bit, i.e., the sensors give only a local yes/no decision of presence of the target.

Consider the combining standard deviations scheme without quantization. As explained in Chapter 4.4.2, the probability of a single sensor giving a large enough value to set off the fusion center threshold by itself is negligibly small. Also, two sensors making a local decision the target is present and giving any value always sets off the alarm. Therefore, if a scheme with extreme quantization, i.e., binary sensors, is considered, setting the fusion center to decide the target is present when two sensors give an alarm yields approximately the same P_{FA} at the fusion center. Thus, the two situations can be compared and the effect of binary sensors on the detection performance can be seen. Although the P_{FA} at the fusion center is the same in both situations, the detecion performance is expected to be worse with binary sensors. This stems from the fact that a single sensor cannot produce a large value which would set off the alarm at the fusion center, even if the target were right next to the sensor.

6.7.2 Main scenario

The main scenario, described in Chapter 6.1, was run with both the basic combining standard deviations scheme and binary sensors. In both cases, P_{FA} of the fusion center was set to 10^{-4}. In case of the binary sensors, the fusion center sets off the alarm when it recieves local alarms from at least two sensors.

For both setups, the simulation was run a thousand times, and the average number of alarms in the basic combining standard deviations scheme was 698.2, and in the binary sensors setup, the average number of alarms was 698.2. In both setups, the performance is the same within rounding error. This is due to the network topology in the main scenario. The sensors are grouped so close to each other that many sensors detect the approaching target almost at the same time. When one sensor has so strong a signal from the target that it would cause the fusion center to set off the alarm by itself, the measurement of at least one other has risen above its detection threshold, and it sends a local alarm to the fusion center, too.

6.7.3 Linear array

To illustrate the possible degredation of detection performance when the sensors give binary information instead of more accurate information to the fusion center, another simulation scenario is considered. In this scenario, there are only 5 sensors, and they are grouped to a linear equidistant array. The distance between sensors is 400 m, whereas in the main scenario the distance is ca. 200 m. The sensors are grouped along the y-axis of the coordinate system and the target approaches along the y-axis as well. The speed of the target is ca. 30 km/h and it moves 8 km during the simulation. This scenario is illustrated in Figure 13.

The sensitivity of the network is set to the same level as in the binary sensor simulation of the main scenario, i.e., P_{FA} of the fusion center is set to 10^{-4}. As shown in Table 2, setting P_{FA} of the sensors to $7.29 \cdot 10^{-4}$ results in P_{FA} of 10^{-4} at the fusion center in the main scenario. Now the network has a different size, N, so according to equation (52), a different sensitivity has to be set to the sensors to get the desired P_{FA} of 10^{-4} at the fusion center. By iteration, it was found that setting the sensors to $P_{FA} = 3.18 \cdot 10^{-3}$ yields approximately $P_{FA} = 10^{-4}$ at the fusion center.

This simulation was run one thousand times with both basic and binary sensors in the combining standard deviations scheme. The average number of alarms in the basic combining standard deviations scheme was 270.4, and in the binary sensors setup, the average number of alarms was 268.0. Unlike in the main scenario with binary sensors, discussed in Chapter 6.7.2, now there is a notable difference in the number of alarms between the basic and the binary setup. This difference would likely grow larger if the distance between sensors were increased. On the other hand,

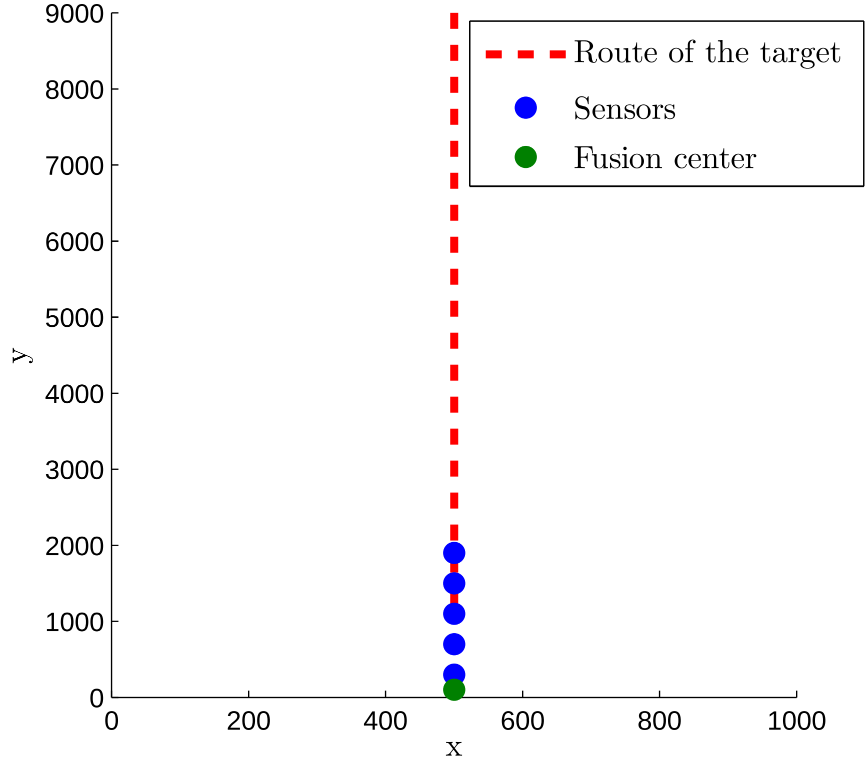

Figure 13: Sensor setup and the route of the target in simulation 3b: linear array.

the difference would likely become smaller if the target approached from some other direction. This scenario, where the target approaches from the same direction along which the sensors are grouped, is essentially a worst-case scenario from the viewpoint of the detection performance of the binary sensors setup.

6.8 Simulation 4 — Effect of β

6.8.1 Introduction

As discussed in Chapter 4.4.2, the fusion center threshold parameter, β, of the combining standard deviations scheme can be adjusted a little lower than its default value of 2. Adjusting beta this way allows a single sensor to set off easier the global alarm at the fusion center. If β is adjusted by a sufficiently small amount, this can be done with a very small increase in the P_{FA} of the fusion center. In this simulation, the effect of β on the detection performance of the combining standard deviations scheme is studied.

6.8.2 Main scenario

The main scenario was run with the combining standard deviations scheme and with two different values of β. In both cases, P_{FA} of the fusion center was set to approximately 10^{-4}. The default value of β is 2, so this is the baseline for comparison with a lower value of β. An increase in P_{FA} that is two orders of magnitude smaller than P_{FA} was deemed a negligible change. According to equation (55), setting $\beta = 1.67$ in the main scenario increases P_{FA} by $1.06 \cdot 10^{-6}$, so $\beta = 1.67$ was used as the lower value.

For both settings, the simulation was run a thousand times, and the average number of alarms was computed. With $\beta = 2$, the average number of alarms was 698.2 and with $\beta = 1.67$, the average number of alarms was 698.5. The increase in the number of alarms is very small and may be caused only by the tiny increase in the number of false alarms. This is the same result as in the simulation with binary sensors, described in Chapter 6.7.2, and the reasons are the same, too. The sensors are grouped relatively close to each other, and the target approaches from such a direction, that the measurement of the sensor closest to the target does not have the time to rise to a level in which it would set off the alarm of the fusion center by itself, even though the threshold is now lower. Other nearby sensors give their alarms before that happens and, thus, a lower value of β does not help here.

6.8.3 Linear array

To show that decreasing β might be beneficial in some cases, the linear array scenario with 5 sensors, described in Chapter 6.7.3, was run with the same P_{FA} of the fusion center and two different values of β, the default value of 2 and a lower value. Now there is a different number of sensors and a different P_{FA} of the sensors, so a new value for β has to be found. According to equation (55), setting $\beta = 1.85$ increases P_{FA} by $1.12 \cdot 10^{-6}$, so $\beta = 1.85$ was used as the lower value.

As before, for both settings, the simulation was run one thousand times, and the average number of alarms was computed. The results are shown in Figure 14, and parameters for the different settings are summarized in Table 5. With $\beta = 2$, the average number of alarms was 270.4 and with $\beta = 1.85$, the average number of alarms was 271.8. Now there is a tangible difference in the detection performance between the setups with different values of β. This goes to show that detection performance can be improved in some cases by adjusting β slightly lower than its default value of 2.

Table 5: Summary of the parameters for the different settings in the linear array scenario.

Setting	β	Sensor P_{FA}	Approximate fusion center P_{FA}
Basic	2	$3.18 \cdot 10^{-3}$	10^{-4}
Alternative 1	1.85	$3.18 \cdot 10^{-3}$	$10^{-4} + 10^{-6}$
Alternative 2	2	$3.19 \cdot 10^{-3}$	$10^{-4} + 10^{-6}$

The increase in the number of alarms by adjusting β in this scenario is larger than the one achieved by directly increasing P_{FA} of the sensors. To show this, a

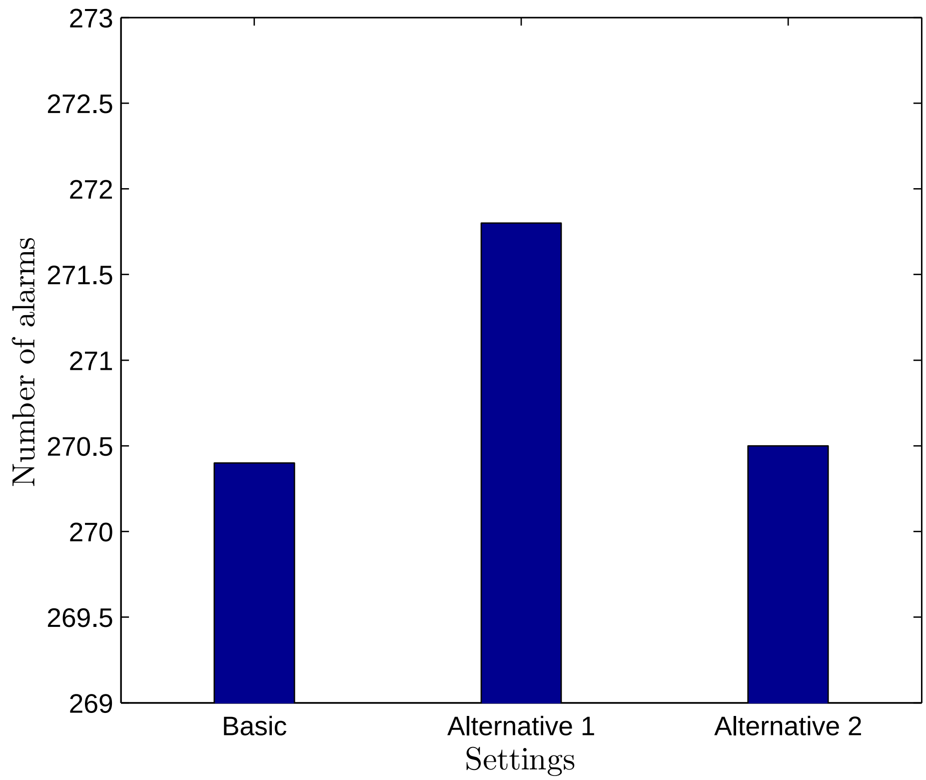

Figure 14: Average number of alarms from a thousand simulation runs in the linear array scenario with 5 sensors using three different settings in the combining standard deviations scheme.

batch of simulations was run where the P_{FA} of the fusion center was increased by approximately the same amount as decreasing β to 1.85 yielded. As stated above, setting $\beta = 1.85$ increases P_{FA} by $1.12 \cdot 10^{-6}$. According to equation (52), increasing P_{FA} of the sensors from $3.18 \cdot 10^{-3}$ to $3.19 \cdot 10^{-3}$ increases the P_{FA} of the fusion center by approximately $1.11 \cdot 10^{-6}$. A simulation was run one thousand times with $\beta = 2$ and $P_{FA} = 3.19 \cdot 10^{-3}$. The result of the simulation is shown in Figure 14 and the parameters discussed above are listed in Table 5. The average number of alarms was 270.5, which is smaller than the number of alarms obtained by setting $\beta = 1.85$, even though the increase in P_{FA} is the same. This shows that, in some cases, adjusting β is results in a better detection performance than directly adjusting the P_{FA} of the sensors. However, β can be adjusted only by a little. For larger modifications to the sensitivity of the sensors, their P_{FA} has to be adjusted directly.

6.9 Discussion of simulation results

Results of the simulations conducted in Simulink indicate that the combining standard deviations scheme performs better in the main scenario. Simulations also show that the proposed schemes work appropriately under a long tailed background noise distribution. Although performance decreases for both schemes, it still remains at an acceptable level. The difference in the performance of the two schemes is approximately the same under the long tailed noise distribution, i.e., the combining standard deviations scheme still performs considerably better than the maximum likelihood scheme.

It is quite surprising that the heuristic combining standard deviations scheme works so much better in the main scenario than the maximum likelihood scheme, which semi-optimal in a more idealistic scenario. The motivation for developing the combining standard deviations scheme was to devise a somewhat simpler scheme with easier setup of parameters. In addition these benefits, performance in the main scenario is significantly better than that of the maximum likelihood scheme. Also, the combining standard deviations scheme is equally robust against a long tailed noise distribution as the maximum likelihood scheme.

7 Conclusions and future work

A summary of the thesis is given is this chapter. Also, usefulness of the schemes are discussed here, and suggestions are given for future research.

7.1 Conclusions

Decentralized detection is an area of research which focuses on detection of targets with a network of sensors. The research conducted in this area often concentrates on theoretical issues and ideal situations, whereas this thesis discusses the problems in a realistic scenario. The main scenario of this thesis is detecting a vehicle in a forest using acoustic sensors.

The main problem with many of the detection schemes developed for ideal scenarios is the requirement that all sensors observe the exactly same phenomena. The fact that the target is at a different distance from the sensors is typically not taken into account. Another assumption, which is often made, is the assumption of normal distributed background noise. In the derivations of the detection schemes of this thesis, the same assumption is made, but one of the simulations is conducted with another noise distribution. This thesis studies the performance of two schemes in the aforementioned main scenario. One of the schemes is the maximum likelihood scheme which has some optimality properties under certain assumptions. The other one is a heuristic scheme named combining standard deviations, based on classical detection theory, with no optimality properties.

The combining standard deviations scheme offers easier setup of threshold values and the approximation of false alarm rates for given thresholds using simple equations. Based on simulation results, it also seems to offer better performance. The main objective of this thesis was to find a practically viable detection scheme for the realistic main scenario. While both presented schemes can be used for this scenario, the combining standard deviations scheme offers better performance and easier setup in the type of scenario considered here.

7.2 Directions of future work

Neither scheme properly utilizes information from sensors, which send nothing, and the geometry of the network. Consider a situation in which sensors are placed so densely that their detection regions overlap, and only a sensor in the center of the network gives a local alarm. This alarm is almost certainly a false alarm since none of the other sensors nearby detect anything. On the other hand, if a sensor in the edge of the network gives a local alarm, and none of the nearby sensors detect anything, it is much more likely to be real detection of a target approaching the network. The fusion center does not use this information any way in its decision rule. A viable direction for future work could be to try to find a way to use this information in the decision rule of the fusion center.

Another natural step would be to use actual measurements as the acoustic background noise in simulations, or better yet, employ a network of sensors in a forest, and conduct a live experiment. It would be of interest to try different detection schemes in different background noise conditions, for example, in a windy or rainy environment. To realistically assess the performance in these type of conditions, the analysis would have to be based on actual measurements in the desired conditions.

References

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. Wireless Sensor Networks: A Survey. *Computer Networks,* March 2002, vol. 38, no. 4, pp. 393–422.

[2] Viswanathan, R. and Varshney, P. K. Distributed Detection with Multiple Sensors: Part 1 — Fundamentals. *Proceedings the IEEE,* January 1997, vol. 85, no. 1, pp. 54–63.

[3] Arampatzis, T., Lygeros, J. and Manesis, S. A Survey of Applications of Wireless Sensors and Wireless Sensor Networks. *Proceedings of the 2005 IEEE International Symposium on Intelligent Control, 2005, Mediterrean Conference on Control and Automation,* 2005, pp. 719–724.

[4] Becker, G. and Güdesen, A. Passive Sensing with Acoustics on the Battlefield. *Applied Acoustics,* 2000, vol. 59, no. 2, pp. 149–178.

[5] Kay, S. M. *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory.* New Jersey, Prentice Hall PTR, 1998.

[6] Neyman, J. and Pearson, E. S. On the Problem of the Most Efficient Tests of Statistical Hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character,* 1933, vol. 231, pp. 289–337.

[7] Chamberland, J.-F. and Veeravalli, V. V. Wireless Sensors in Distributed Detection Applications — An Alternative Framework Tailored to Decentralized Detection. *IEEE Signal Processing Magazine,* May 2007, vol. 24, no. 3, pp. 16–25.

[8] Appadwedula, S., Veeravalli, V. V. and Jones, D. L. Decentralized Detection with Censoring Sensors. *IEEE Transactions on Signal Processing,* April 2008, vol. 56, no. 4, pp. 1362–1373.

[9] Warren, D. and Willett, P. Optimal Decentralized Detection for Conditionally Independent Sensors. *Proceedings of the 1989 American Control Conference,* June 1989, pp. 1326–1329.

[10] IEEE 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. 2007 revision. New York, IEEE Standards Association. 1184 p.

[11] IEEE 802.15.1. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs). New York, IEEE Standards Association, 2005. 580 p.

[12] Cho, S. and Chandrakasan, A. P. Energy Efficient Protocols for Low Duty Cycle Wireless Microsensor Networks. *ICASSP Proceedings 2001,* vol. 4, Salt Lake City, Utah, pp. 2041–2044.

[13] Appadwedula, S. Energy-efficient Sensor Networks for Detection Applications. Ph.D. Thesis. University of Illinois, USA, 2003.

[14] Tsitsiklis, J. N. and Athans, M. On the Complexity of Decentralized Decision Making and Detection Problems. *IEEE Transactions on Automatic Control,* May 1985, vol. 30, no. 4, pp. 440–446.

[15] Yan, Q. and Blum, R. S. On Some Unresolved Issues in Finding Optimum Distributed Detection Schemes. *IEEE Transactions on Signal Processing,* December 2000, vol. 48, no. 12, pp. 3280–3288.

[16] Tenney, R. R. and Sandell, N. R., Jr. Detection with Distributed Sensors. *IEEE Transactions on Aerospace and Electronic Systems,* July 1981, vol. 17, no. 4, pp. 501–510.

[17] Rago, C., Willett, P. and Bar-Shalom, Y. Censoring Sensors: A Low-Communication-Rate Scheme for Distributed Detection. *IEEE Transactions on Aerospace and Electronic Systems,* April 1996, vol. 32, no. 2, pp. 554–568.

[18] Le Cam, L. The Central Limit Theorem Around 1935. *Statistical Science,* 1986, vol. 1, no. 1, pp. 78–91.

[19] Quach, A. and Lo, K. Automatic Target Detection Using a Ground-based Passive Acoustic Sensor. *IEEE Proceedings of Information, Decision and Control,* 1999, Pyrmont, Australia, pp. 187–192.

[20] Kay, S. M. *Fundamentals of Statistical Signal Processing: Estimation Theory.* New Jersey, Prentice Hall PTR, 1993.

[21] Matlab website, July 2011, http://www.mathworks.com/products/matlab/

[22] Tse, D. and Viswanath, P. *Fundamentals of Wireless Communication,* Cambridge University Press, 2005.

[23] Adams, R. A. *Calculus: A Complete Course.* 6th ed. Toronto, Pearson Education Canada Inc., 2006.

[24] Springer, M. D. and Thompson, W. E. The Distribution of Products of Independent Random Variables. *SIAM Journal on Applied Mathematics,* May 1966, vol. 14, no. 3, pp. 511–526.

[25] Chacon-Rodriguez, A., Julian, P., Castro, L., Alvarado, P. and Hernandez, N. Evaluation of Gunshot Detection Algorithms. *IEEE Transactions on Circuits and Systems — I: Regular Papers,* February 2011, vol. 58, no. 2, pp. 363–373.

[26] Nyberg, T. Fully Decentralized Sequential Detection in an Underwater Setting. M.Sc. Thesis. Stockholm University, Sweden, 2008.

[27] Krim, H. and Viberg, M. Two Decades of Array Signal Processing Research: The Parametric Approach. *IEEE Signal Processing Magazine,* July 1996, vol. 13, no. 4, pp. 67–94.

[28] Gupte, S., Masoud, O., Martin, R. F. K. and Papaniikolopoulos, N. P. Detection and Classification of Vehicles. *IEEE Transactions on Intelligent Transportation Systems,* March 2002, vol. 3, no. 1, pp. 37–47.

[29] Lipton, A. J., Fujiyoshi, H. and Patil, R. S. Moving Target Classification and Tracking from Real-Time Video. *Fourth IEEE Workshop on Applications of Computer Vision, WACV '98, Proceedings,* October 1998, pp. 8–14.

[30] Bertozzi, M., Broggi, A., Caraffi, C., Del Rose, M., Felisa, M. and Vezzoni, G. Pedestrian Detection by Mean of Far-Infrared Stereo Vision. *Computer Vision and Image Understanding Journal,* May 2007, vol. 106, no. 2–3, pp. 194–204.

[31] Sheng, X. and Hu, Y.-H. Maximum Likelihood Multiple-Source Localization Using Acoustic Energy Measurements with Wireless Sensor Networks. *IEEE Transactions on Signal Processing,* 2005, vol. 53, no. 1, pp. 44–53.

[32] Ristic, B., Arulampalam, S. and Gordon, N. *Beyond the Kalman Filter — Particle Filters for Tracking Applications.* Artech House, 2004.

[33] Simulink website, December 2011, http://www.mathworks.se/products/simulink/

[34] Hsin, C.-f., Liu, M. Network Coverage Using Low Duty-cycled Sensors: Random & Coordinated Sleep Algorithms. *ISPN '04: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks,* 2004, pp. 433–442.

[35] Rabi, M., Stabellini, L., Proutiere, A. and Johansson, M. Networked Estimation under Contention-based Medium Access. *International Journal of Robust and Nonlinear Control,* January 2010, vol. 20, no. 2, pp. 140–155.

[36] ISO 9613-2. Acoustics — Attenuation of Sound During Propagation Outdoors — Part 2: General Method of Calculation. The International Organization for Standardization, 1996. 49 p.

[37] Mellin, V. and Antila, M. Henkilöauton moottoriäänen muokkausjärjestelmä. Akustiikkapäivät 2007, Espoo, Finland.

[38] Antila, M. and Lankila, A. Kuorma-auton ohjaamon aktiivinen melunvaimennus. Akustiikkapäivät 2009, Vaasa, Finland.

[39] Rossing, T. D. *The Science of Sound.* 2nd ed. Addison-Wesley Publishing, 1990.

[40] Björk, E. and Toivonen, T. Äänen vaimeneminen metsämaastossa. Akustiikkapäivät 2009, Vaasa, Finland.

[41] Bolstad, W. M. *Introduction to Bayesian Statistics.* 2nd ed. Wiley-Interscience, USA, 2007.

Appendix A — Matlab simulation script

The listing below presents the Matlab script that was used in the simulations. If no target is used, outputs are estimated probabilities of false alarm for a single sensor and the fusion center. If a target is included in the simulation, the script prints out the average number of alarms at the fusion center. This script works only for sensors which take one measurement into account at a time, i.e., for setting $K = 1$. Also, all sensors are assumed to be set to have the same P_{FA}.

```
%% The simulation

% This simulation works only for the case K = 1!

clear all;


%% Parameters

scenario = 0; % Select scenario: 0:main, 1: linear array
drop_packets = 1; % Dropping some of the packets: 0:no, 1:yes
use_target = 1;
scheme = 0; % Select detection scheme: 0:CSD, 1:ML, 2:binary-CSD
noise = 0; % Select noise distribution: 0:normal, 1:gamma
I = 1000; % Number of simulation runs

mu = 0.0011; % Mean of the background noise
var = 10^(-7); % Variance of the background noise
spower = 115; % Sound power level of the target

P_FA = 7.29*10^(-4); % P_FA which the sensors assume in CSD scheme
% Threshold of the sensors in the CSD scheme
threshold = sqrt(var)*qfuncinv(P_FA) + mu;
beta = 2; % FC threshold parameter in CSD scheme

rho = 2640; % Sensor threshold of the ML scheme
alpha = 900; % Fusion center threshold of the ML scheme

if noise == 1
    % Gamma distribution parameters
    k = mu^2/var;
    theta = var/mu;
end


%% Locations, target route, sound pressures & communication ...
    parameters

% Locations of the sensors
if scenario == 0
    sensors = [1000 1450; 1100 1190; 840 1160; 550 1120; 440 ...
        1010; 820 1020;...
        1040 1010; 590 930; 1280 940; 860 890; 1170 780; 760 ...
```

```
            750; 1330 750;...
        640 710; 960 670; 1440 660; 1250 610; 790 580; 1160 480; ...
            1430 400];
    T = 2000; % Number of time instances in simulation
    N = 20; % Number of sensors in the network
else
    sensors = [500 1900; 500 1500; 500 1100; 500 700; 500 300];
    T = 1000; % Number of time instances in simulation
    N = 5; % Number of sensors in the network
end

% Genrate route of the target
location = zeros(T,2);
if scenario == 0
    for time = 1:T
        if time < 1000
            location(time,2) = ...
                (3000-20*0.2*(time-500))+200*sin(0.1*0.2*(time-500));
            location(time,1) = ...
                (3500-23*0.2*(time-500))+300*cos(0.1*0.2*(time-500));
        elseif time ≥ 1000 && time ≤ 1100
            location(time,2) = ...
                (3000-20*0.2*500)+200*sin(0.1*0.2*500);
            location(time,1) = ...
                (3500-23*0.2*500)+300*cos(0.1*0.2*500);
        else
            location(time,2) = 3*(time-1100) + ...
                (3000-20*0.2*500)+200*sin(0.1*0.2*500);
            location(time,1) = 10*(time-1100) + ...
                (3500-23*0.2*500)+300*cos(0.1*0.2*500);
        end
    end
else
    for t = 1:T
        location(t,1) = 500;
        location(t,2) = 9100 - 8*t;
    end
end

% Compute distance of the target from all sensors,
% sound pressure levels [dB] and sound powers [Pa] at all time ...
    instances
distances = zeros(T,N);
spl = zeros(T,N); % Sound pressure levels
sp = zeros(T,N); % Sound pressures
for t = 1:T
    for n = 1:N
        distances(t,n) = distance(location(t,1),location(t,2),...
            sensors(n,1),sensors(n,2));
        if distances(t,n) < 1
            distances(t,n) = 1;
        end
        spl(t,n) = soundpressure(spower,distances(t,n));
        sp(t,n) = 20*10^(-6)*10^(spl(t,n)/20);
```

```
    end
end

% Wireless communication model parameters
S = 100; % Number of time slots in the MAC scheme
p = 2/(N+2); % Probability of sending data in one time slot
P_SNR = 0.02; % Probability of packet loss due to low SNR


%% Data generation & processing

sensor_alarms = 0;
FCalarms = 0;
lost_packets = zeros(T,1);

for i = 1:I
    % Noise of the sensors
    if noise == 0
        x = mu + sqrt(var).*randn(T,N); % Normal distributed noise
    else
        x = gamrnd(k,theta,T,N); % Gamma distributed noise
    end

    % Generate the data for sensors
    if use_target == 0
        sensor_data = x;
    else
        sensor_data = sp + x;
    end

    % Sensor processing
    sent_data = zeros(T,N); % The data for fusion center
    if scheme == 0 || scheme == 2 % CSD scheme (binary included)
        for t = 1:T
            for n = 1:N
                if sensor_data(t,n) > threshold
                    sent_data(t,n) = ...
                        (sensor_data(t,n)-mu)/sqrt(var);
                    sensor_alarms = sensor_alarms + 1;
                end
            end
        end
    else % ML scheme
        for t = 1:T
            for n = 1:N
                mu1 = sensor_data(t,n);
                x0 = (sensor_data(t,n)-mu)^2;
                x1 = (sensor_data(t,n)-mu1)^2;
                Lambda = exp(-1/(2*var).*x1)./exp(-1/(2*var).*x0);
                if Lambda > rho
                    sent_data(t,n) = Lambda;
                    sensor_alarms = sensor_alarms + 1;
                else
                    sent_data(t,n) = rho;
```

```
                end
            end
        end
    end

    % Wireless communication simulation, i.e. dropping packets
    if drop_packets == 1
        B = zeros(T,1); % Number of transmissions at a given time
        if scheme == 0 || scheme == 2 % CSD and binary-CSD schemes
            for t = 1:T
                B(t) = nnz(sent_data(t,:));
                for n = 1:N
                    if sent_data(t,n) > 0
                        if rand < ((1-p)+p*(1-(1-p)^(B(t)-1)))^S
                            sent_data(t,n) = 0;
                            lost_packets(t) = lost_packets(t) + 1;
                        elseif rand < P_SNR
                            sent_data(t,n) = 0;
                            lost_packets(t) = lost_packets(t) + 1;
                        end
                    end
                end
            end
        else % ML scheme
            for t = 1:T
                B(t) = nnz(sent_data(t,:)-rho);
                for n = 1:N
                    if sent_data(t,n) > rho
                        if rand < ((1-p)+p*(1-(1-p)^(B(t)-1)))^S
                            sent_data(t,n) = rho;
                            lost_packets(t) = lost_packets(t) + 1;
                        elseif rand < P_SNR
                            sent_data(t,n) = rho;
                            lost_packets(t) = lost_packets(t) + 1;
                        end
                    end
                end
            end
        end
    end

    % Fusion center processing
    if scheme == 0 % CSD scheme
        for t = 1:T
            if sum(sent_data(t,:)) > beta*qfuncinv(P_FA)
                FCalarms = FCalarms + 1;
            end
        end
    elseif scheme == 1 % ML scheme
        for t = 1:T
            if prod(sent_data(t,:)) > alpha*rho^N
                FCalarms = FCalarms + 1;
            end
        end
```

```
    else % Binary—CSD scheme
        for t = 1:T
            % Modify the sent data into binary form
            binary_sent_data = (sent_data > 0);
            if sum(binary_sent_data(t,:)) ≥ 2
                FCalarms = FCalarms + 1;
            end
        end
    end

    i
end

if use_target == 0
    % Estimated probability of false alarm of one sensor
    P_FA_sensors = sensor_alarms/(I*T*N)
    % Estimated probability of false alarm of the fusion center
    P_FA_FC = FCalarms/(I*T)
else
    % Average number of fusion center alarms
    FCalarms_average = FCalarms/I
end

% Average number of lost packets
lost_packets_average = lost_packets/I;
```