Exploiting Sparse Structures in Source Localization and Tracking

Juhlin, Maria

2022

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Exploiting Sparse Structures in Source Localization and Tracking

by

MARIA JUHLIN

Thesis for the degree of Doctor of Philosophy

by due permission of the LTH at Lund University, to be publicly defended on Friday 25th of November 2022, at 9:00 in lecture hall MA 3 at Centre for Mathematical Sciences, Sölvegatan 20, Lund.

Faculty opponent: Prof. Joakim Jaldén, KTH Royal Institute of Technology, Stockholm

Organization
LUND UNIVERSITY

Centre of Mathematical Sciences
Mathematical Statistics
Box 118
SE-221 00 Lund, Sweden

Document name
DOCTORAL DISSERTATION

Date of disputation
2022-11-25

Sponsoring organization

Author(s)
Maria Juhlin

Title and subtitle
Exploiting Sparse Structures in Source Localization and Tracking

Abstract

This thesis deals with the modeling of structured signals under different sparsity constraints. Many phenomena exhibit an inherent structure that may be exploited when setting up models, examples include audio waves, radar, sonar, and image objects. These structures allow us to model, identify, and classify the processes, enabling parameter estimation for, e.g., identification, localisation, and tracking. In this work, such structures are exploited, with the goal to achieve efficient localisation and tracking of a structured source signal. Specifically, two scenarios are considered. In papers A and B, the aim is to find a sparse subset of a structured signal such that the signal parameters and source locations may be estimated in an optimal way. For the sparse subset selection, a combinatorial optimization problem is approximately solved by means of convex relaxation, with the results of allowing for different types of *a priori* information to be incorporated in the optimization. In paper C, a sparse subset of data is provided, and a generative model is used to find the location of an unknown number of jammers in a wireless network, with the jammers' movement in the network being tracked as additional observations become available.

DOKUMENTDATABLAD enl SIS 61 41 21

Key words
Source localization and tracking, Cramér-Rao lower bounds, convex optimization

Classification system and/or index terms (if any)

Supplementary bibliographical information

Language
English

ISSN and key title
1404-0034, Doctoral Theses in Mathematical Sciences 2022:10

ISBN
978-91-8039-413-0

Recipient's notes

Number of pages
154

ISRN
LUTFMS-1050-2022

Security classification

Signature ______________________ Date 2022-11-25

Exploiting Sparse Structures in Source Localization and Tracking

MARIA JUHLIN

LUND UNIVERSITY

LTH FACULTY OF ENGINEERING

Centre of Mathematical Sciences
Mathematical Statistics

Cover image by Maria Juhlin

Mathematical Statistics
Centre of Mathematical Sciences
Lund University
Box 118
SE-221 00 Lund
Sweden

http://www.maths.lth.se/

Doctoral Theses in Mathematical Sciences 2022:10
ISSN: 1404-0034
ISBN: 978-91-8039-413-0 (print)
ISBN: 978-91-8039-414-7 (pdf)
ISRN: LUTFMS-1050-2022

Printed in Sweden by Media-Tryck, Lund 2022

Every man takes the limits of his own field of vision for the limits of the world.

— Arthur Schopenhauer

Contents

Acknowledgements

Denna avhandling markerar slutet av mina fem år som doktorand. Det har varit en omtumlande och givande resa med höga toppar, djupa dalar, grusade planer, vilda glädjestunder och nya insikter. Saker och ting blir sällan exakt som vi föreställt oss, men det är just de tillfällen när förväntan möter verkligheten som är de mest utvecklande och lärorika. Såhär i efterhand så finns det mycket jag önskar att jag gjort annorlunda, men många gånger krävs det lite tidsperspektiv för att se sin egen utveckling och inse vad man har lärt sig.

Denna avhandling hade aldrig blivit till vore det inte för min handledare Prof. Andreas Jakobsson. Med din smittande entusiasm och förmåga att se möjligheter där andra ser begränsningar har du väglett mig under dessa år. I såväl medgång som motgång, har du funnits där, stöttat och peppat och med en ängels tålamod stått ut med alla mina sammanbrott. Viktigast av allt har dock varit att du alltid trott på mig, även vid de tillfällen jag inte ens gjort det själv. Jag skulle också vilja tacka min biträdande handledare Prof. Maria Sandsten för hennes stöd och uppmuntran. Ett särskilt tack till mina nära vänner och kollegor Isabella Reinhold och Carl Åkerlindh för deras osvikliga stöd och insiktsfulla kommentarer.

Ett stort tack riktas även till mina medförfattare, speciellt Jie Cheng, Filip Elvander, Johan Swärd, samt till den administrativa personalen James Hakim, Susann Nordqvist, Denise Bokander, Michael Macdougall, Maria Lövgren och Karin Serrano. Tack till mina föräldrar Peter och Vibeke och till min bror Karl för allt stöd och för att ni står ut med mig även i mina värsta stunder. Slutligen så vill jag tacka min man Johan, när allt är nattsvart och hemskt så är du ljuset som skänker värme och tröst och hopp om en bättre morgondag. Jag älskar dig.

Lund, 2022 *Maria Juhlin*

Abstract

This thesis deals with the modeling of structured signals under different sparsity constraints. Many phenomena exhibit an inherent structure that may be exploited when setting up models, examples include audio waves, radar, sonar, and image objects. These structures allow us to model, identify, and classify the processes, enabling parameter estimation for, e.g., identification, localisation, and tracking. In this work, such structures are exploited, with the goal to achieve efficient localisation and tracking of a structured source signal. Specifically, two scenarios are considered. In papers A and B, the aim is to find a sparse subset of a structured signal such that the signal parameters and source locations may be estimated in an optimal way. For the sparse subset selection, a combinatorial optimization problem is approximately solved by means of convex relaxation, with the results of allowing for different types of *a priori* information to be incorporated in the optimization. In paper C, a sparse subset of data is provided, and a generative model is used to find the location of an unknown number of jammers in a wireless network, with the jammers' movement in the network being tracked as additional observations become available.

Keywords: Source localization and tracking, Cramér-Rao lower bounds, convex optimization

Populärvetenskaplig sammanfattning

Denna avhandling behandlar hur man kan utnyttja signalers struktur för att fatta bra beslut trots att man har ofullständig information. Många signaler och vardagliga fenomen uppvisar någon form av struktur eller mönster, som vi medvetet eller omedvetet använder oss av vid olika typer av beslutsfattande.

Tal och musik kan väl beskrivas som en blandning av tonade och otonade ljud som därmed har en intern struktur som beror på vad som sägs eller spelas. Denna struktur avgör hur ljudet uppfattas av oss och där egenskaper som tonhöjd, klang och placering avgör hur vi uppfattas ljudet i dess sammanhang.

Ofta infinner sig situationer där vi har ofullständig information, men där vi kan utnyttja signalens struktur för att sluta oss till bra beslut. Det kan röra sig om ljud men även till exempel bilder där placering, färger och sammanhang avgör hur vi uppfattar bilden. Ibland kan den här typen av information vara svårtolkad och det infinner sig en osäkerhet. Trots denna osäkerhet kan sådan partiell information utnyttjas för att tolka informationen. Som ett exempel kan text och sammanhang hjälpa till med identifieringen av svårtolkade handskrivna bokstäver.

De två första artiklarna i avhandlingen handlar om hur man skall designa sina experiment givet partiell kunskap om signalerna man ämnar mäta. I den första artikeln har man en signalmodell för en ljudkälla och dess spridningsmönster, och man har en uppfattning kring dess position och frekvens, men kunskapen är behäftad med en viss osäkerhet. Målet är att placera ut mikrofoner på ett sådant sätt att signalmätningarna blir så bra som möjligt. Man har ett givet antal mikrofoner att placera ut, men antalet möjliga placeringar är långt fler. Metoden för att välja ut placeringar ger även en uppskattning på hur bra resultat man kan förvänta sig, och kan således vara till stöd vid experimentdesignen. Utnyttjandet av signalstruktur kan användas vid flera olika tillämpningar. I den andra artikeln utnyttjas kunskap om en signals utbredning och frekvens för att designa en radar på optimalt sätt.

I avhandlingens tredje artikel lokaliserades störkällor i ett trådlöst nätverk. Störkällor modelleras ofta som objekt vars utbredningsmönster kan beskrivas som delvis känt och som släcker ut all kommunikation inom utbredningsarean. Givet störstatusen för noderna i nätverket, samt generell information om utbredningsmönstret, lokaliserades störkällorna genom att anta modeller för utbredningsmönstret, och genom att anpassa parametrarna i modellerna så att de passade ihop med nodernas störstatus. Genom att låta störkällorna röra sig i nätverket kan man insamla mer information för att förbättra lokaliseringen. Ibland uppstår dock situationer när antalet störkällor förändras under tidens gång, till exempel för att en störkälla slås av. Avslutningsvis undersöktes metoder för att följa störkällor, beaktande att antalet objekt kan förändras under tiden.

Popular summary

The topic of this thesis is how signal structures may be exploited in decision making, enabling good decisions despite a lack of information. Many signals and everyday phenomena exhibit some sort of structure or pattern that we, intentionally or unintentionally, make use of in terms of decision making.

Speech and music may be well described as a mixture of voiced and unvoiced sounds that exhibit an internal structure depending on what is said and played. This structure determines how the sound is interpreted, where features such as key, timbre, and placement determine how we interpret the sound in its context.

Often, situations arise where we have incomplete information, but we may exploit the signal structure to make a good decision. This may concern audio, but also images, where placement, colours, and context determine how we perceive the image. Sometimes this information is difficult to interpret and an uncertainty arises. Despite this uncertainty, such partial information may be exploited when interpreting the image. For example, text and context may aid in the identification of blurry handwritten letters.

The first two articles in this thesis deal with experiment design, and more precisely, on how one should design and setup an experiment given partial information about the measurement signal of interest. The first article treats the case with an audio source with known signal model and spatial spread, and where there is partial knowledge about the position and frequency of the sound source. The goal is to place microphones in such a way that the signal measurements will be as good as possible. There is a given number of microphones to place, but the number of possible microphone placements is far greater. The method for selecting the optimal placement also gives an estimate of how good results that can be expected, and may thus be used as an aid in the experimental design. The exploitation of signal structure may be used in other applications, and in the second article an optimal radar was designed in terms of the selection of the used carrier frequencies, given partial information about the location of a potential reflector.

The third article in this thesis treats the localization of jammers in a wireless network. Jammers are often modeled as objects with jamming patterns that are partially known and that cancel all communication in the covered area. Given the jamming status of the nodes in the network, and general information about the jamming pattern, the jammers may be localized by assuming models for the jamming patterns and adapting the parameters in the models in such a way that they fit the nodes' jamming status.

By allowing the jammers to move in the network, more information may be gathered that may aid in improving the localization. Sometimes situations arise where the number of jammers to track change with time, for example if a jammer is switched off, or breaks. As a final task, different tracking methods were evaluated for the tracking of jammers, where the number of jammers may change with time.

List of publications

This thesis is based on the following publications:

A Maria Juhlin and Andreas Jakobsson, "Optimal Sensor Placement for Localizing Structured Signal Sources", *Elsevier Signal Processing*, Vol. 202, January 2023

which was also partially published as

M. Juhlin and A. Jakobsson, "Optimal Microphone Placement for Localising Tonal Sounds Sources", *28th European Signal Processing Conference*, Amsterdam, Jan. 18-22, 2020.

B Jie Cheng, Maria Juhlin, Andreas Jakobsson, Wen-Qin Wang, "Designing Optimal Frequency Offsets for Frequency Diverse Array MIMO Radar", *Submitted manuscript*

C Maria Juhlin and Andreas Jakobsson, "Efficient Tracking of Inhomogeneous Jammers in a Wireless Network", *Submitted manuscript*

which was also partially published as

Maria Juhlin and Andreas Jakobsson, "Localization of Multiple Jammers in Wireless Sensor Networks", *29th European Signal Processing Conference*, pp.1596-1600 , 2021

Publications not included in this thesis:

i T. Kronvall, M. Juhlin, J. Swärd, S I. Adalbjörnsson, and A. Jakobsson, "Sparse Modeling of Chroma Features", *Elsevier Signal Processing*, Vol. 130, pp. 105-117, Jan. 2017.

ii M. Juhlin and A. Jakobsson, "Identifying decaying sinusoidal modes using signed measurements", *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Guadeloupe, Dec. 15-18, 2019.

iii M. Juhlin, F. Elvander, and A. Jakobsson, "Defining Graph Signal Distances Using an Optimal Mass Transport Framework", *27th European Signal Processing Conference*, A Coruna, Sept. 2-6, 2019.

iv M. Juhlin, F. Elvander, J. Swärd, and A. Jakobsson, "Fast Gridless Estimation of Damped Modes", *2018 International Symposium on Intelligent Signal Processing and Communication Systems*, Okinawa, Nov. 27-30, 2018.

v M. Juhlin, J. Swärd, M. Pesavento, and A. Jakobsson, "Estimating Faults Modes in Ball Bearing Machinery using a Sparse Reconstruction Framework", *26th European Signal Processing Conference*, Rome, Sept. 3-7, 2018.

vi M. Juhlin, T. Kronvall, S. I. Adalbjörnsson, and A. Jakobsson, "Sparse Chroma estimation for Harmonic Non-stationary Audio", *23rd European Signal Processing Conference*, Nice, Aug. 31-Sept. 4, 2015.

vii T. Kronvall, M. Juhlin, S. I. Adalbjörnsson, and A. Jakobsson, "Sparse Chroma estimation for Harmonic Audio", *40th International Conference on Acoustics, Speech, and Signal Processing*, Brisbane, April 19-24, 2015.

Abbreviations and symbols

SNR	signal-to-noise ratio
FIM	Fisher Information Matrix
CRLB	Cramér-Rao lower bound
HCRLB	Hybrid Cramér-Rao lower bound
BCRLB	Bayesian Cramér-Rao lower bound
WSN	Wireless sensor network
MAP	Maximum *a posteriori*
ML	Maximum likelihood
GMM	Gaussian Mixture Model
EKF	Extended Kalman filter
MCMC	Markov Chain Monte Carlo
MVJR	Minimum volume jamming region
RMSE	Root mean squared error
ARMSE	Average root mean squared error
pdf	probability density function
$\ell(\theta)$	log-likelihood
$(\cdot)^T$	transpose
ω	angular frequency
τ	time-lag
$\mathcal{N}(\mu, \sigma^2)$	normal distribution with expected value μ and variance σ^2
$\int$	integrals without indicated limits go from $-\infty$ to ∞
$x^*(t)$	complex conjugate of $x(t)$
$f(t) * g(t)$	convolution between $f(t)$ and $g(t)$
$\mathbb{E}[\mathrm{X}]$	expected value of X
var[X]	variance of X
Cov[X]	covariance of X
$\Re(x)$	real part of x
$\Im(x)$	imaginary part of x

Introduction

Introduction

The aim of this thesis is to investigate how statistical methods can be used as a modeling tool to facilitate the placement, localization, and tracking of objects in situations where the data is structured, as well as sparse or uncertain. This introductory chapter gives a brief overview about the theory used in the enclosed papers, sets the work into context, and presents a research outlook for the studied problems.

Many forms of measurements exhibit a large degree of internal structure, such as periodically reoccurring patterns or a harmonic structure. Typically, methods striving to infer information about the measured signal can exploit these form of structures beneficially, improving the ability to, for example, detect abnormalities. However, it is not uncommon that these forms of structure are only approximately known, or only reasonably well modeled. The resulting mismatch may cause a substantial loss of performance if the actual inherent structure of the signal deviates too much from the assumed model. This work strives to exploit these partial structures, such that the assumed model is allowed to be determined with some degree of uncertainty. This endows the resulting methods the ability to work well over a range of models, typically making the resulting method more robust to model mismatch.

The rest of this chapter is organized as follows: section 1 describes the statistical models used in this thesis, and section 2 introduces some concepts from statistical decision theory. The tracking methods used in paper C are detailed in section 3, and section 4 describes the evaluation method used in papers A and B. Finally, section 5 gives an outline of the papers, and section 6 gives a brief outlook about possible further research.

1 Statistical models

This section introduces some of the statistical models used in the enclosed papers. All the enclosed papers deal with location estimates in 2- or 3-dimensional space.

Mathematical modeling is the task of describing the world, or a phenomena, by means of a model. Such models can take on many forms; they may, for example, be linear or non-linear, parametric or non-parametric, deterministic or stochastic, or they may be a mixture of these forms. Depending on the assumptions we make regarding the observed data, different models will be more suitable than others. As there are many possibilities when selecting a model, the question arises regarding which model is the best choice, a question that does not have a clear-cut answer. However, for the case of statistical models, one may possibly narrow and reformulate the question as one of finding the model parameters that best describe the data.

1.1 Maximum likelihood

For a given set of observed data $x_1, x_2, \dots, x_N$, the goal is often to estimate the joint density of a particular observation that is formed using the parameters θ, i.e., we wish to determine $p(x_1, x_2, \dots, x_N | \theta)$. However, the parameters detailing the joint density, θ, are typically unknown. The problem solved by the maximum likelihood (ML) estimator is that of determining the parameters, θ, assuming a known distribution, making our observed data the likeliest. For a set of N independent and identically distributed observations, $x_n \in f(x|\theta)$, the likelihood function may be formed as

$$\mathcal{L}(x_1, \dots, x_N | \theta) = \prod_{i=1}^{N} f(x_i | \theta) \tag{1}$$

The ML estimator thus seeks to find the parameters θ maximizing $\mathcal{L}(x_1, \dots, x_N | \theta)$. That is, given a set of observed data one strives to determine the parameters that have the highest probability of producing the observed data.

1.2 Maximum *a posteriori*

An alternative estimation method is maximum *a posteriori* (MAP), a Bayesian method for estimating the expected parameters in a model describing the observed data. One notable difference between the ML and the MAP estimator is that MAP allows for the incorporation of prior knowledge regarding the data models. Bayes' law states that the posterior probability of event A given B is

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{2}$$

where $p(A)$ and $p(B)$ are the probabilities of events A and B, respectively. In case $p(B)$ is used as a normalizing constant, $p(A|B)$ may thus be written as $p(A|B) \propto p(B|A)p(A)$. For the estimation problem, this implies that $p(\theta|x_1, \ldots, x_N) \propto p(x_1, \ldots, x_N|\theta)p(\theta)$. The MAP estimator thus forms an estimate of θ as

$$\max_{\theta} p(\theta|x_1, \ldots, x_N) \tag{3}$$

This estimate is very similar to the ML estimate, with the addition of the prior knowledge about the parameters detailed in $p(\theta)$. Both estimation techniques seek to find the joint probability density of the observed data, by determining the most likely parameters. Often, but not always, these estimates will also converge to the same solution.

1.3 Range-based and range-free methods

When dealing with signals with spatial spread, there are two modeling schools, focusing on range-based methods or range-free methods, with the main difference being if they account for signal power drop off with distance. Range-based methods incorporate information about the range between the signal source and the receiver by taking into account the fact that as the signal spreads in space, the signal power is spread over a larger area, such that the relative drop off in signal strength between two measuring points may be used to deduce the distance to the source. Range-based methods form the basis for the methods used in paper A and paper B, where audio waves and radar measurements are studied, respectively. In the papers, the signal spread is modeled as an exponentially decaying signal that depends on the distance r from the signal origin [1]

$$r^{-(d-1)/2} e^{-i\omega r/c} \tag{4}$$

where d is the dimensionality, c is the speed of the signal propagation, and the ω is the angular frequency of the signal. Thus, estimates of the decaying factor may be employed as a means to estimate the source location, and in radar also the velocity of the target.

With the range-free methods, no knowledge about the signal strength is available, or of the relative signal strength. This situation is common in scenarios such as when an unknown adversary is trying to corrupt or flood a communication channel, such as in a situation where a jammer is active in a wireless sensor network (WSN). This is the situation treated in paper C, where the signal of interest is a jamming signal of unknown location, signal signature, and signal strength, set in a WSN. This setup thus assumes that there is one, or more, jamming devices that completely blocks communication in a certain area, with the only known information being the position of all the nodes in the network and their binary state as being available or jammed.

When the range is not included as an available signal measurement, other signal structures have to be utilized in order to deduce the target location. In paper C, the observations are modeled as resulting from the following properties: the jamming patterns are isotropic, meaning that a node is either jammed or not, the jamming patterns may be modeled as convex parametric curves (or unions of such), and the location of all nodes are known. The number of jammers is not known, and their jamming patterns are allowed to overlap. These assumptions are common in the field of jammer location, but typically only a single jammer scenario is considered [2–6].

A common technique for localizing a jammer is to fit a circle that encloses the jammed nodes. Several different methods have been proposed for fitting such a circle, ranging form the minimal enclosing circle, maximum inscribed circle, and finding the convex hull of the jammed nodes and thus the diameter of the jamming region [2–6].

An alternative approach, that makes better use of the limited information provided, is to instead model the jammer's spatial spread as a 2-dimensional Gaussian distribution. As such, the model facilitates more general jamming patterns than provided by circular models, and it provides the possibility to consider overlapping jamming regions, thus making it more versatile than the single jammer location methods. The jammer location is then estimated as the

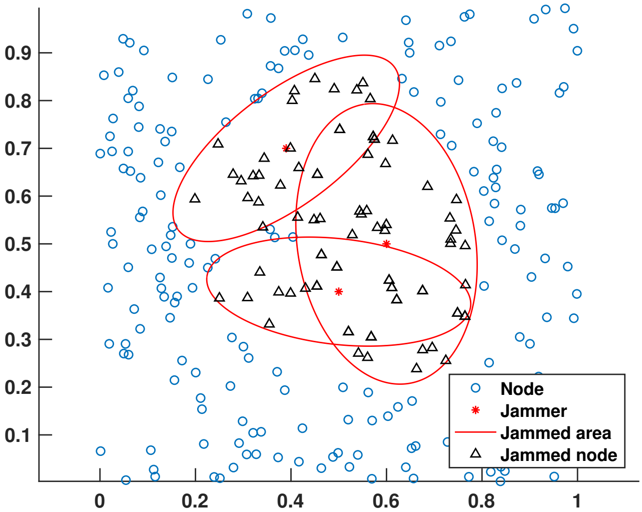


Figure 1: An example of a jamming scenario with overlapping jamming regions and elliptical jamming patterns. The unaffected nodes are represented by blue circles, and the jammed nodes are represented by black triangles.

mean of the distribution, and the spread is given by the standard deviation. Figure 1 describes an example of a jamming situation, where the outline of the jamming regions is marked in the red ellipses, the jammer locations is marked with red dots, the unaffected jammers are represented as blue circles, and the jammed nodes are represented as black triangles.

2 Statistical decision theory

Often, we want to use our observed data for making statistically grounded decisions, such as e.g., predictions or classifications. Statistical decision theory sets a framework for such questions. If we assume that our observed input data, x, may be well described as a set of random variables, we can form a statistical model $f(x)$, such that we may describe our observed output data, y, where it is assumed that this output data is also a set of random variables, with the joint distribution $p(x,y)$ [7]. To make a statement about how well our proposed model $f(x)$ fits the observed data y, a function detailing the similarity between $f(x)$ and y is needed. In the literature, such a function is known as a *loss-function*, $L(y, f(x))$. Two common loss-functions are the squared error loss

$$L_1(y, f(x)) = (y - f(x))^2 \tag{5}$$

and the absolute loss

$$L_2(y, f(x)) = |y - f(x)| \tag{6}$$

Typically, one wants to make a more general statement pertaining to all possible values of the observed input and output data. Thus, one forms the expectation with respect to the loss function, $\mathbb{E}(L(y, f(x))$, enabling general conclusions about the model fit of $f(x)$.

In paper C, the statistical decision to be made is that of classification, whereby we seek to classify the jammed nodes into different clusters, thus detailing the different jamming patterns. Typical statistical methods used for classification are k-nearest neighbour (k-nn) [7], linear discriminant analysis (LDA) [8], quadratic discriminant analysis (QDA) [8], logistic regression [9], the EM-algorithm [10], and the Gibbs algorithm [11]. The k-nn method approximates the statistical expectation by the average, and assumes that $f(x)$ is well approximated by its n nearest neighbours. LDA assumes that the different clusters can be well separated by a hyperplane; sometimes the dimensionality of the data has to be increased for this to be true. Similarly, QDA assumes that the data may be well separated by a quadratic function, sometimes eliminating the need to increase the dimension. Logistic regression uses the logit function to classify the different data points. For binary logistic regression the logit function becomes

$$\text{logit} = \frac{p}{1-p}, \text{ where } p \in \{0, 1\} \tag{7}$$

The three main variants of logistic regression are binary logistic regression, multinomial logistic regression, and ordinal logistic regression. In contrast, the EM and the Gibbs algorithms try to fit distributions to the observed data, thus classifying the data based on the posterior probability of class belonging.

Methods for clustering and classification are relevant in paper C of this thesis, where one aim was to identify and localize different jammers. The suggested modus operandi was to first classify the jammed nodes into different clusters, potentially corresponding to different jammers. The second step was then to form jamming patterns covering the different clusters, such that all jammed nodes were covered, and excluding all unjammed ones. In this thesis, we have opted to use the k-means method, the EM-algorithm, and Gibbs algorithm for forming the classification. The k-means method was selected mainly because of its computational simplicity, offering an intuitive clustering at a low computational cost. Moreover, the k-means algorithm is used as a substep in the EM-algorihtm, suggesting one may also use the algorithm as a stand alone method. The EM and Gibbs algorithms were selected because of their ability to incorporate the made assumptions about the geometry of the jamming regions when forming the clusters. Before using the algorithms, the assumption is made that the observed data is generated form a Gaussian Mixture Model (GMM), and that the geometry of Gaussians may be well modeled as ellipses, which is thus the assumed jamming pattern.

2.1 K-means and K-nn

There is often a misconception and some confusion about the difference between the k-nn method and the k-means method, in part probably due to the similarity between the names. Conceptually, the main difference between the two methods is that k-nn is a supervised learning algorithm for the classification problem, whereas k-means is an unsupervised method for the clustering problem [12–14]. The k-nn method classifies the data into different groups based on a set of predefined features. These features have to be predefined by the user, and the type of features used will differ depending on the application. Moreover, to determine good features and how to value the different features, the k-nn algorithm has to be trained, thus tuning the algorithm to the data, a process known as supervised learning. Although supervised learning may yield good results, the requirement of training the algorithm means that enough training data has to be available, and also that structural changes in the data may require the process of algorithm training to be repeated. In situations where clear features may be defined and enough data is available, the k-nn method is a simple yet effective method. Contrary to the k-nn method, the k-means method requires no features to be defined, and no algorithm training. The k-means algorithm clusters the data based solely on the raw data. The algorithm seeks to minimize the variance within the each cluster, while also maximizing the variance between the different clusters. The main appeal of the k-means algorithm is its inherent simplicity. It is intuitively easy to understand, and requires no training or feature selection. The jammer data in paper C offers no clear features, only raw data about the jamming status of the nodes, which was why the k-means algorithm was used.

2.2 EM-algorithm

The expectation-maximization (EM) algorithm [10], provides an iterative procedure for finding maximum likelihood estimates of the parameters in a statistical model with some form of hidden or latent state. In paper C, this is useful when estimating the parameters detailing the jammers' spatial Gaussian distributions, where the observed data, y, is the locations of the jammed nodes.

The idea of the EM algorithm is to extend the original maximum likelihood problem into two closely related problems that are easier to solve individually than the original problem. After providing an initial set of parameters, the algorithm proceeds by alternating between the E-step and the M-step until convergence is reached, e.g. when the change in parameters is small enough. In iteration k of the EM algorithm, these steps are defined as:

E-step: The first step of the algorithm is to obtain the objective function defined as the expectation

$$Q\left(\theta, \theta_k\right) = \mathbb{E}\left[\log \mathrm{pdf}_\theta\left(x_{1:n}, y_{1:n}\right) \mid y_{1:n}, \theta_k\right]. \tag{8}$$

where $x_{1:n}$ denotes the hidden state data, $y_{1:n}$ the observed data, and θ_k the parameters in the k:th iteration.

M-step: The second step then involves finding the parameters that maximizes this objective function,

$$\theta_{k+1} = \arg\max_{\theta \in \Theta} Q\left(\theta, \theta_k\right). \tag{9}$$

It can be shown that the log-likelihood $\ell(\theta_k)$, defined as the logarithm of the joint probabilities of θ_k, decided by the assumed distributions of the hidden state data $x_{1:n}$, is guaranteed to increase with each iteration of the algorithm [7]. The EM algorithm, however, does not guarantee convergence to the global maximum. Despite this, the EM algorithm is typically considered to be a simple and robust method for obtaining maximum likelihood estimates. For many distributions of the exponential family, the EM algorithm is particularly well suited, since the M-step can be solved analytically without the need for costly numerical optimization. Also, the EM-algoritm is well suited to unsupervised learning methods, limiting the need for user input and calibration. The two main disadvantages of the EM-algorithm are that the convergence is slow, and the method may converge to a local minimum, making the method sensitive to the starting values. One way to mitigate this is to rerun the algorithm several times with slightly different initial conditions.

2.3 Gibbs algorithm

An alternative method that may be used for obtaining the parameter estimates detailing the jammer's spatial spread is the Gibbs algorithm [11]. The Gibbs method is a Markov Chain Monte Carlo (MCMC) method [15,16], that samples from the assumed posterior distributions. The main difference between the Gibbs algorithm and the EM-agorithm is that where the EM-algorithm maximizes the conditional distribution, the Gibbs algorithm samples it instead [7]. The algorithm iterates between two steps, a computing step where the maximum likelihood estimates are computed, and a sampling step where new estimates are sampled. The iteration between the two steps is repeated until convergence is reached, i.e., when the change in parameters between consecutive iterations is small enough, or a predefined maximum number of iterations is reached. In iteration k of the Gibbs algorithm, the steps are defined as follows:

The first step of the Gibbs algorithm is the same as in the EM-algorithm, i.e., to obtain the objective function defined as the expectation

$$Q\left(\theta, \theta_k\right) = \mathbb{E}\left[\log \mathrm{pdf}_{\theta}\left(x_{1:n}, y_{1:n}\right) \mid y_{1:n}, \theta_k\right]. \tag{10}$$

The second step is to sample the new parameter estimates, where the estimates are sampled from the distributions defined by $Q\left(\theta, \theta_k\right)$ from the first step

$$\text{Sample } \theta_{k+1} \text{ from } Q\left(\theta, \theta_k\right) \tag{11}$$

The Gibbs algorithm is well suited to be used in combination with computationally efficient Monte Carlo methods, such as importance sampling and rejection sampling. Moreover, it is often easy to evaluate the conditional distributions, thus further increasing its appeal. Problems arise when the dimensionality is high as this causes slow convergence.

3 Tracking

As an object moves, additional information may be acquired as more measurement data is accumulated. In the jammer situation, the outlines of the jamming patterns may be more accurately estimated as new nodes are jammed, thus adding more measurements to the statistical model, and in the radar case the Doppler effect and velocity of the target may be better estimated by using more radar stations in the triangulation, reducing the number of ambiguities.

3.1 Kalman filter

The Kalman filter is a tracking method based on linear dynamic systems corrupted by an additive noise [17, 18]. Typically, the Kalman filter is used in situations when either the variable of interest may only be measured indirectly, or when measurements are available from various sensors but might be subject to noise. For the jammer localization problem in paper C, both of the above conditions apply. Consider a general system of the form

$$x_k = Ax_{k-1} + Bu_k + \eta_k \tag{12}$$

$$y_k = Cx_k + Du_k + \epsilon_k \tag{13}$$

with hidden state x_k, measurements y_k and deterministic input signal u_k. Here, A is a (potentially time varying) state transition matrix that determines how the hidden state x_k evolves, B and D are control input matrices that control the effect of the input vector u_k on the hidden and observed state, respectively, and C is a measurement matrix that defines the relationship between the hidden and observed states. The process noise, defined as the vector η_k, represents deviations from the ideal scenario, and the measurement noise is represented as the vector ϵ_k. It is often assumed that η_k is a zero mean Gaussian with covariance matrix Q, and ϵ_k is zero mean Gaussian with covariance matrix R.

The filter iterates between a prediction step and an update step, also known as a propagation step and a correction step. In each step, the filter equations provide an estimated mean and covariance, which in the case of Gaussian system are optimal. Using the notation where $(\hat{\cdot})$ denotes the estimate, $(\cdot)^-$ the predicted (prior) estimate, $(\cdot)^+$ the posterior (updated) estimate, and P denotes the error covariance matrix, the prediction step may be described as

$$\hat{x}_k^- = A\hat{x}_{k-1}^+ + Bu_{k-1} \qquad \text{predicted state estimate} \qquad (14)$$

$$P_k^- = AP_{k-1}^+ A^T + Q \qquad \text{predicted error covariance} \qquad (15)$$

In the update step, the measurement residual, $\tilde{z}_k$ is calculated and used to update the state estimate and the error covariance, such that

$$\tilde{z}_k = y_k - C\hat{x}_k^- - Du_k \qquad \text{measurement residual} \qquad (16)$$

$$K_k = P_k^- C^T (R + CP_k^- C^T)^{-1} \qquad \text{Kalman gain} \qquad (17)$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \tilde{z}_k \qquad \text{updated state estimate} \qquad (18)$$

$$P_k^+ = (I - K_k C)P_k^- \qquad \text{updated error covariance} \qquad (19)$$

where I denotes the identity matrix, and K denotes the Kalman gain. It may be noted that the Kalman gain is multiplied with the residual to provide the correction term, $K_k\tilde{z}_k$, to the predicted estimate $\hat{x}_k^+$, which determines how much weight to assign to the current estimate. This form of the Kalman filter makes the assumptions of model linearity as well as additive Gaussian noise, both being fairly strong assumptions. The Gaussanity of the processes is supposed to be reflected in the covariance matrices Q and R. Often, the statistics are not known or are not Gaussian. In such cases, the matrices Q and R may be used as tuning parameters that the user can adjust to get the desired performance.

The assumption of model linearity limits the application scope of the standard Kalman filter as many processes are not linear. A popular method for mitigating this limitation is to instead consider the Extended Kalman filter (EKF). The EKF is used in many real world applications such as navigation systems and GPS [19]. In the EKF, the nonlinear model is linearized about the mean using a Taylor expansion.

Unfortunately, the Kalman filter cannot handle a varying number of states, and thus is makes tracking difficult in cases examined in paper C where objects may leave or enter the tracked area.

3.2 Particle filter

In many cases, the system model is non-linear or inaccurate, making the EKF an inappropriate modeling choice [20]. In such situations, it may be beneficial to turn to sequential Monte Carlo methods and employing a particle filter [15,16,19]. The particle filter assumes a probabilistic approach to the filtering process where the goal is to compute the posterior distributions of the states. The dynamical model considered may then be detailed as

$$x_k = A(x_{k-1})x_{k-1} + B(x_{k-1})u_k + \eta_k \tag{20}$$

$$y_k = C(x_{k-1})x_k + D(x_{k-1})u_k + \epsilon_k \tag{21}$$

which is the same as the model used for the Kalman filter. The difference between (12) and (20) is that in (20) no assumption about linearity or Gaussanity is made. As for the Kalman filter, it is assumed that the hidden states $x_0, x_1, ...,$ may be well modeled as a Markov process on $\mathbb{R}^{d_x}$ that evolves with a transition probability $p(x_k|x_{1:k-1}) = p(x_k|x_{k-1})$, i.e., the transition between states is dependent only on the current state, and not on the history. The observation process takes values on $\mathbb{R}^{d_y}$, and are assumed to be conditionally independent provided that $x_0, x_1, ...$ are known.

The goal of the particle filter is to estimate the distribution of x_k given y_k. For a general function $f(x)$ of x with pdf $p(x)$, this is equivalent to solving

$$\mathbb{E}(f(x)|y) = \int f(x)p(x|y)p(x)dx \tag{22}$$

In general, such expectations are difficult to solve analytically but may be well approximated by the use of Monte Carlo sampling. For the particle filter, the filtering steps consist of the state prediction step for estimating the distribution of the latent state process, i.e.,

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{k-1})dx_{k-1} \tag{23}$$

The measurement update then generates the posterior probability density

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{\int p(y_k|x_k)p(x_k|y_{1:k.1})dx_k} \tag{24}$$

To approximate the distributions, a number of particles is needed, hence the name particle filter. Unlike the Kalman filter, the particle filter is able to handle a varying number of states. The two main drawbacks of the particle filter are that convergence is not guaranteed for a finite number of particles, and that it is a computationally quite expensive method which performs poorly in high dimensions unless variance reduction (e.g., importance sampling, Rao-Blackwellization) can be utilized.

3.3 Viterbi tracking

An alternative method to the previous filters that is also able to handle non-linear models, non-Gaussian noise, and a varying number of states is the Viterbi algorithm. Like the particle filter, it is a probabilistic tracking algorithm that aims at maxmizing the *a posteriori* probability distribution of a set of events given a set of observations [21]. The algorithm assumes that the hidden states is defined using a Markov model, enabling the tracking problem to be efficiently solved by means of dynamic programming. In dynamic programming, one strives to break the original problem down into smaller subproblems that may be solved individually.

Unlike the particle filter that tries to estimate the expectation by means of sampling, the Viterbi algorithm instead aims at maximizing the *a posteriori* probability distribution. For the tracking problem in paper C this is the same as finding the most likely path. In order to rank the paths, they are assigned a score determined by a scoring function $g(\Lambda)$, where Λ is the path. The algorithm considers one step at a time, and at each time point evaluates which is the most likely event given the observations. If the event set consists of B possible events $\{e_1, e_2, \dots, e_B\}$, there will be B^T possible tracks of length T. For this reason, the number of events has to be quite small for the algorithm to be computationally feasible. Unlike the particle filter that does not guarantee convergence, the Viterbi algorithm guarantees that an optimal solution is found. Pseudo code for the overall algorithm is presented in paper C.

4 Cramér -Rao lower bound

There are many ways to evaluate the performance of an estimator, but two of the more desirable features of a good estimator are that it is unbiased (or at least asymptotically unbiased), which means that the estimate has a chance to reach the true value of the parameter, and that it has a low variance. Given a statistical model of some observed data, x, with the probability density function (pdf) $p(\mathbf{x};\boldsymbol{\theta})$, the amount of information that the data contains pertaining to the set of parameters, $\boldsymbol{\theta}$, is captured in the Fisher information matrix (FIM), $\mathbf{F}(\boldsymbol{\theta})$, defined as [22]

$$\mathbf{F}(\boldsymbol{\theta}) = -\mathbb{E}\left\{\nabla_{\boldsymbol{\theta}} logp(\mathbf{x};\boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}} logp(\mathbf{x};\boldsymbol{\theta})^H\right\} \tag{25}$$

Given the FIM, one is able to determine a lower bound on the achievable estimation accuracy, as specified in the following theorem:

Theorem 1 (Cramér-Rao lower bound) *If the pdf $p(\mathbf{x};\boldsymbol{\theta})$ satisfies the condition*

$$\mathbb{E}\left\{\nabla_{\boldsymbol{\theta}} log p(\mathbf{x};\boldsymbol{\theta})\right\} = 0, \forall \boldsymbol{\theta} \tag{26}$$

where the expectation is taken with respect to $p(\mathbf{x};\boldsymbol{\theta})$, then, the variance of any unbiased estimator $\hat{\boldsymbol{\theta}}$ must satisfy

$$var[\hat{\boldsymbol{\theta}}_{ii}] \geq [\mathbf{F}^{-1}(\boldsymbol{\theta})]_{ii} \tag{27}$$

That is, the Cramér-Rao lower bound (CRLB) forms a theoretical lower bound for the variance of any unbiased estimator satisfying (26). All estimators that attain this lower bound are said to be statistically efficient.

In papers A and B, we seek to find the set of microphones placements and carrier frequencies, respectively, such that the location parameters of the measurement signal may be estimated as efficiently as possible. In order to do so, we exploit a statistical model for the observed data. Using this model, we seek to determine the most suitable microphone placements and carrier frequencies, respectively. In order to do so, we strive to determine which of these placements and frequencies will minimize the corresponding CRLB for the problem, utilizing the available prior information of the observed signal sources. We thus seek to find the optimal set of microphone placements/carrier frequencies such that mean of the norm of the squared errors of the parameter estimates, $\mathbf{e} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$, is minimized. Formally, the objective function we seek to minimize is $\mathbb{E}\{||\mathbf{e}||_2^2\}$, which may be written as

$$\mathbb{E}\{||\mathbf{e}||_2^2\} = \mathbb{E}\{\text{trace}(\mathbf{e}\mathbf{e}^T)\} = \text{trace}\{\mathbf{R}_e\} \tag{28}$$

where $\mathbf{R}_e$ is the error covariance matrix. This may be formulated as the scalarized minimization problem

$$\begin{aligned} &\underset{v_1 \ldots v_J}{\text{minimize}}\ \text{trace}\left(\sum_{j=1}^{J} \lambda_j v_j v_j^T\right)^{-1} \\ &\text{subject to } \lambda \geq 0,\ \mathbf{1}^T\lambda = 1 \end{aligned} \tag{29}$$

where λ_j denotes the coefficients, and v_j is the dual variable of the error of the j:th parameter, J is the total number of parameters. This problem is known as the A-optimal design problem [23].

4.1 Other optimality criteria

Depending on the design requirements, different notions of optimality may be utilized, with other noteworthy examples being the D-optimal and the E-optimal design schemes. In the D-optimal scheme, the determinant of the error covariance matrix, $\mathbf{R}_e$, is minimized. Geometrically, this may be interpreted as minimizing the resulting confidence ellipsoid of the parameters [23]. Formally, the optimization problem to solve for the D-optimal case is

$$\begin{aligned} &\underset{v_1 \ldots v_J}{\text{minimize}} \log\det\left(\sum_{i=1}^{p} \lambda_i v_i v_i^T\right)^{-1} \\ &\text{subject to } \lambda \geq 0, \quad 1^T\lambda = 1, \end{aligned} \tag{30}$$

The D-optimal design is often used in situations where one seeks to determine which variables are the least significant.

In contrast, the E-optimal design scheme aims at minimizing the maximum eigenvalue of $\mathbf{R}_e$. Because the area of the confidence ellipsoid is related to the length of the norm of the largest eigenvalue, minimizing it may be geometrically interpreted as minimizing the maximum variance of the parameters [23]. The optimization problem for the E-optimal design is

$$\begin{aligned} &\underset{v_1 \ldots v_J}{\text{minimize}} \left\| \left(\sum_{t=1}^{p} \lambda_i v_i v_1^T\right)^{-1} \right\|_2 \\ &\text{subject to } \lambda \geq 0, \quad 1^T\lambda = 1 \end{aligned} \tag{31}$$

In this thesis, the selection problem is that of determining an optimal subset of parameters from a larger set of potential parameter candidates. Formally, this problem is a combinatorial problem, as one may only select a discrete number of candidates. The complexity of combinatorial problems grows exponentially, and thus finding the optimal solution quickly becomes infeasible even for cases with only a few parameters. As a way to proceed, one may form an approximative solution by performing a convex relaxation of the problem, whereby it is possible to select fractions of many different parameters at the same time. After the optimization, a rounding operation is performed such that only the candidates with the highest fractions are included.

4.2 Extensions of the Cramér-Rao bound

The CRLB provides a lower bound of the variance for each of the parameter estimates in $\boldsymbol{\theta}$. This bound will depend on the true values of these parameters, and will typically vary for different parameter values. This means that the CRLB for a given parameter value is indicating the expected variance of a statistically efficient estimator. However, it is often the case that the actual parameter values are unknown, as these are what is sought by the use of an estimator. One may estimate the parameters using the efficient estimator and then evaluate the CRLB for the found parameters, assuming that this bound will be reasonably close to the variance of the sought parameters, and that the resulting CRLB therefore can be used as an estimate of the variance of the found estimator. In paper A, the unknown microphone placements are sought, and in paper B the transmission frequencies, such that these settings minimize the CRLB for a given location of the signal source. Thus, in paper A, we seek to determine where sensors should be placed such that the expected variance of the location estimates will be minimized. As this bound will depend on the location of the sound source, this is generally not all that helpful. However, often we have some prior knowledge of the source location, such that we may know that the source can be expected to be in a specified area. We may thus assume that the source may be located at a finite number of positions, and evaluate the CRLB for each of these positions. By then selecting the microphone placement such that the worst case CRLB bound (WCRLB) is minimized, we determine the best sensor placements over the range of assumed locations. Similarly, in paper B, we consider a set of potential target locations and velocities when determining the most suitable carrier frequencies to use, i.e., the carrier frequencies minimizing the worst case variance of the location estimates, given the assumed uncertainty grids.

When evaluating the WCRLB, one thus forms a grid detailing the prior knowledge, around each of the parameters, and then evaluate (29) for each candidate on this grid. If there is more than one parameter that is uncertain, they may all be gridded, and (29) is then evaluated for each possible combination of the uncertain parameters. As such, the WCRLB does not necessarily provide the optimal selection in the sense of minimum variance, but rather it provides an optimal selection provided that there may be some uncertainty in the parameters. It may thus be seen as an estimator that minimizes the risk of the selection.

As may be noted, the resulting worst case estimator will need to be evaluated over a grid of possible parameters, which will substantially increase the complexity of the resulting design scheme. As an alternative, one may model the uncertainty in the parameters using an assumed probability distribution. Provided this information, one may then instead form what is known as the Bayesian Cramér-Rao Lower bound (BCRLB). For the BCRLB, the FIM consists of two parts, one pertaining to the data and one to the prior distributions of the parameters [24]. Thus, the Bayesian FIM (BFIM) may be decomposed as

$$\mathbf{F}_B(\theta) = \mathbf{F}_D(\theta) + \mathbf{F}_P(\theta) \tag{32}$$

If all the parameters are independent Gaussian variables, the second part of the FIM, $\mathbf{F}_P(\theta)$, pertaining to the prior, will be a diagonal matrix. Due to the stochastic nature of the parameters, the BFIM will then be a constant matrix [24]. When forming the BFIM, the MAP estimates of the parameters have to be computed. If the parameters are all linear functions, the MAP estimate of the parameters can be computed analytically. However, if the data model is non-linear in the parameters, computing the BFIM analytically is typically difficult, in which case the BFIM may instead be estimated using Monte Carlo methods.

Sometimes it may also be the case that a better model would be to describe some of the parameters as stochastic and some of the parameters as deterministic. If this is the case, one may form what is known as the Hybrid Cramér-Rao lower bound, (HCRLB), defined as [25]

$$\mathbf{F}_H = \begin{bmatrix} \mathbf{F}_\theta & \mathbf{F}_{\theta,\phi} \\ \mathbf{F}_{\theta,\phi} & \mathbf{F}_\phi \end{bmatrix} \tag{33}$$

where θ and ϕ details the deterministic and stochastic variables, respectively, with $F_{\theta,\phi}$ denoting the FIM with respect to both the stochastic and the deterministic parameters. As the name suggests, the resulting bound it is a hybrid of the classical CRLB and the BCRLB.

5 Outline of papers

Paper A: Optimal Sensor Placement for Localizing Structured Signal Sources
This work is concerned with determining optimal sensor placements that allow for an accurate location estimate of structured signal sources, taking into account the expected location areas and the typical range of the parameters detailing the signals. In the presentation, we illustrate the technique for tonal sound signals, exploiting the expected harmonic structure of such signals. To determine preferable sensor placements, we propose a computationally efficient scheme that minimizes theoretical lower bounds on the variance of the location estimate over the possible sensor placements, while taking into account the expected variability in the impinging signals, and introducing various forms of constraints on the optimization. Numerical examples and real measurements illustrate the performance of the proposed scheme. The work in paper A has been published in part as

> M. Juhlin and A. Jakobsson, "Optimal Microphone Placement for Localising Tonal Sounds Sources", *28th European Signal Processing Conference*, Amsterdam, Jan. 18-22, 2021.

and in full as

> M. Juhlin and A. Jakobsson, "Optimal Sensor Placement for Localizing Structured Signal Sources", Elsevier Signal Processing, Volume 202, January 2023.

Paper B: Designing Optimal Frequency Offsets for Frequency Diverse Array MIMO Radar
Frequency diverse array (FDA) radars provide a potential solution to target localisation along the slant range and azimuth angle due to the range-angle-dependent transmit beampattern caused by the used frequency increments. However, the S-shaped beampattern resulting from the standard FDA leads to multiple candidate location estimates, introducing ambiguity in the target localization. To make full use of the degrees of freedom (DOF) allowed by the frequency increments, we here propose an optimal FDA multiple-input multiple-output (MIMO) frequency design scheme based on the

Cramér-Rao lower bound (CRLB). The resulting system, here termed the optimal FDA-MIMO (OFDA-MIMO), is formed by optimizing the expected localization estimation accuracy, given the available prior knowledge of potential target locations. The used offsets are found as those minimizing the corresponding Bayesian CRLB (BCRLB), and may be iteratively refined as further information becomes available in a multi-pulse detection scenario. Both theoretical analysis and simulation results validate the preferable performance of the proposed system as compared to alternative frequency selection schemes.

Paper C: Efficient tracking of inhomogeneous jammers in a wireless network In this paper, we propose a novel approach for locating and tracking an unknown number of jammers in a wireless network. The area and shape covered by each jammer is assumed to be partly unknown, and may vary over time as the jammers move through the network. By assuming that the jammed region for each jammer may be described by one of a set of parametric methods, the appropriate shape as well as the number of jammers may be determined, even when the jammed regions overlap substantially. Employing a Viterbi-based tracking approach, we allow for an improved localization of moving jammers, including when new jammers emerge or leave the network. Numerical simulations illustrate the performance of the introduced framework as compared to recent alternative approaches. The work in paper C has been published in part as

> M. Juhlin and A. Jakobsson, "Localization of Multiple Jammers in Wireless Sensor Networks", *29th European Signal Processing Conference*, Dublin, Aug. 23-27, 2021.

The manuscript in this thesis will be submitted as a journal contribution.

6 Further outlook

The work in this thesis can be generalized in several different ways. In the following, we briefly mention some such extensions that closely relate to the presented works.

6.1 Paper A

The work in paper A may be further generalized in various ways. For example, currently, the model addresses the case of harmonic sources. A natural extension to this model would be to also allow for the presence of nearly-harmonic sources, possibly using models such as the one in [26]. One may also include envelope-based signal models, thereby also allowing for unvoiced or non-tonal sound sources. Moreover, the model may be improved by including information inferred by the surroundings such as the room impulse response, wall structures, as well as parametric reverberation models.

6.2 Paper B

As the model in paper B is currently stated, it deals with far field targets with no interfering surroundings. For a radar, this is typically an idealized and somewhat unrealistic scenario. A more realistic setting would include clutter from the surroundings as well as the presence of interference. One development of the model would be to make use of the area topology to deal with wideband disruptions. Another extension would be to consider wider targets that may not be approximated as point sources. Moreover, targets are often subject to rotation and acceleration, something currently not addressed by the model. Allowing for rotation and acceleration could be done by introducing a time varying reflexivity, or by the use of a parametric model, where these parameters are then included in the parameter vector.

6.3 Paper C

The work in paper C may also be extended in various ways. One such alternative is to extend the jammer patterns to more arbitrary patterns, e.g., described by a polygon that is then estimated as part of the tracking algorithm. Other research ventures would be to allow for fading effects from buildings and trees. This could possibly be done by including the topology of the area under study, inferred form e.g., maps. Further, the implementation of an online tracking algorithm would broaden the usability of the algorithm as adversaries may be tracked in real time. One may also envision that the ideas in the first two papers are employed to determine suitable locations for a subset of network nodes in order to allow for improved detection and localization performance of potential jammers.

References

[1] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*, Prentice Hall, Englewood Cliffs, N.J., 1993.

[2] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, pp. 28–34, 2000.

[3] J. Blumenthal, R. Grossman, F. Golatowski, and D. Timmermann, "Weighted Centroid Localization in Zigbee-based Sensor Networks," *IEEE International Symposium on Intelligent Signal Processing*, 2007.

[4] H. Huang, X. Hu, X. Zhang, P. Zhang, and Y. Li, "An Optimal Jammer Selection for Improving Physical-Layer Security in Wireless Networks with Multiple Jammers," in *International Wireless Communications and Mobile Computing Conference*, 2016.

[5] Y. Sun, R. Molva, M. Onen, X. Wang, and X. Zhou, "Catch the Jammer in Wireless Sensor Network," in *IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, 2011.

[6] H. Liu, W. Xu, Y. Chen, and Z. Liu, "Localizing jammers in wireless networks," in *IEEE International Conference on Pervasive Computing and Communications*, 2009.

[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, 2 edition, 2009.

[8] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[9] A. Agresti, *Categorical Data Analysis*, John Wiley & Sons, second edition, 2007.

[10] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Roy. Stat. Soc. Ser.*, vol. 39, pp. 1–38, 1977.

[11] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[12] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Pearson Education Inc., Upper Saddle River, New Jersey, 2003.

[13] R. Szeliski, *Computer Vision Algorithms and Applications*, Springer London Ltd, London, 2010.

[14] S.Shalev Shwartz and S. Ben-David, *Understanding Machine Learning from Theory to Algorithms*, Cambridge University Press, New York, USA, 2014.

[15] G. H. Givens and J. A. Hoeting, *Computational statistics*, John Wiley & Sons, Hoboken, New Jersey, 2013.

[16] N. Chopin and O. Papaspiliopoulos, *An Introduction to Sequential Monte Carlo*, Springer, Gewerbestrasse 11, 6330 Cham, Switzerland, 2020.

[17] R. E. Kalman, ""On the General Theory of Control Systems"," in *Proc. First IFAC Congress*, Moscow, 1960, vol. 1, pp. 481–492.

[18] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Systems*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 2000.

[19] A. Gustafsson, C. Lundquist, and Z. Sjanic, *Statistical sensor fusion*, Studentlitteratur AB, 2018.

[20] S. Särkkä, *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013.

[21] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[22] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, Prentice-Hall, Englewood Cliffs, N.J., 1993.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[24] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, John Wiley and Sons, Inc., 1968.

[25] H. Messer, "The Hybrid Cramer-Rao Lower Bound - From Practice to Theory," in *Fourth IEEE Workshop on Sensor Array and Multichannel Processing, 2006.*, 12–14 July, 2006, pp. 304–307.

[26] F. Elvander, J. Swärd, and A. Jakobsson, "Mismatched estimation of polynomially damped signals," in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Guadeloupe, Dec. 15-18 2019.

Scientific Publications

Author Contributions

Maria Juhlin's contributions to each of the papers included in the Ph.D thesis are mentioned in bold italics below each paper title

Paper A: Optimal Sensor Placement for Localizing Structured Signal Sources, Maria Juhlin and Andreas Jakobsson
Joint development of the main ideas and concepts. Extension of application scope to include spatial signals. Mathematical derivation, numerical simulations, and main responsible for the writing of the paper

Paper B: Designing Optimal Frequency Offsets for Frequency Diverse Array MIMO Radar, Jie Cheng, Maria Juhlin, Andreas Jakobsson, and Wen-Qin Wang
Joint development of the main ideas, concepts, and mathematical derivation. Co-wrote the paper and guided set-up of simulations

Paper C: Efficient tracking of inhomogeneous jammers in a wireless network, Maria Juhlin and Andreas Jakobsson
Developed ideas and concept, mathematical derivation, numerical simulations, and main responsible for the writing of the paper

Paper A

Paper A

Optimal Sensor Placement for Localizing Structured Signal Sources

Maria Juhlin and Andreas Jakobsson

Abstract

This work is concerned with determining optimal sensor placements that allow for an accurate location estimate of structured signal sources, taking into account the expected location areas and the typical range of the parameters detailing the signals. In the presentation, we illustrate the technique for tonal sound signals, exploiting the expected harmonic structure of such signals. To determine preferable sensor placements, we propose a computationally efficient scheme that minimizes theoretical lower bounds on the variance of the location estimate over the possible sensor placements, while taking into account the expected variability in the impinging signals, and introducing various forms of constraints on the optimization. Numerical examples and real measurements illustrate the performance of the proposed scheme.

Keywords: Sensor placement, structured signals, performance bounds, convex optimization

1 Introduction

Reliable localization, estimation, and/or detection of a partly known signal source is a problem of notable interest in many applications, ranging from telecommunications and metrology, to radar, sonar, and surveillance. For all these applications, the topic of sensor placement is of outmost importance in the resulting estimation problems, as it sets the boundary for the accuracy that may be achieved [1–5].

The problem of optimal sensor placement is a diverse and multi-faceted problem that allows for a rich plethora of different problem setups; [6] discusses the problem of optimal sensor placement for passive stationary sensors, [7] treats the optimal geometry for multi-static sensors using time of arrival (ToA) estimates, [8] explores optimal sensor configuration in a 2-D setting with mobile sensors/targets, and [9] presents an algorithm for a near optimal sensor placement under a minimum redundancy constraint. When discussing the problem of optimal sensor placement, one has to be precise in defining the premises of the problem formulation, as the setup will differ notably depending on factors such as if the sensors are passive or active, if the sensor environment is monostatic or multistatic, if the solution is grid-based, and if the sensors and/or target(s) are stationary or moving, to mention some of the aspects.

Another matter to take into account when formulating the optimal sensor placement scheme is the desired parameter estimates. Depending on what it is you try to estimate, the optimal sensor placement might differ. Consider, for example, the joint estimation of frequency and amplitude of a sinusoidal signal component. It is not obvious that the optimal sensor setup for estimating the amplitudes is the same as the optimal setup for estimating only the frequency. Often, the time difference of arrival (TDoA) is used to determine suitable locations, such as in [10], where optimal sensor placements were selected to minimize the corresponding Cramér-Rao lower bound (CRLB). Other sought parameters include the direction of arrival (DoA), which was employed in [11], the ToA as was used in [7], and the frequency difference of arrival (FDoA), which was used in combination with the TDoA in [8], where the latter also demonstrated that the optimal sensor setup differs depending on the desired parameters.

Yet another aspect of the sensor placement problem is the definition of optimal, as there are many different notions of optimality, which give rise to different optimization methods and frameworks. In many formulations, the criteria to be optimized relate in some notion to the signal strength/energy or the reconstruction error. Some widely used criteria of optimality are related to the Fisher information matrix (FIM) and its inverse; the uncertainty ellipse of an estimator is proportional to the determinant of the inverse FIM, the mean square error (MSE) is proportional to the trace of the inverse FIM, whereas E-optimality relates to the eigenvalues of the FIM. Another aspect to take into account is what robustness you require of your estimator. One might not be interested in maximizing the best case scenario, but rather in minimizing the error for the worst case scenario, as this will set an upper bound for the error of your estimation problem. The choice of optimality condition will affect the optimization scheme, and each of the notions offer different advantages and drawbacks. Moreover, when dealing with a set of different estimation parameters, it is unlikely that they are all of the same magnitude. Rather, the resulting optimization will be more dependent on the largest parameter, which might be a nuisance parameter. As such, a coveted feature of an optimization algorithm is the flexibility to weigh the impact of different parameters in the resulting optimization.

In [12], the authors formulated the sensor selection problem as a convex minimization, maximizing the power of the received signal. Later, in [13–15], the sensor placement was formulated as a minimization of the MSE of the localization estimates. In [3], sensor placement was considered from the perspective of a minimum variance distortionless response. Typically, the chosen algorithm will affect the solution and the computational complexity of the proposed method. Often, some sort of convex relaxation is required to allow for a computationally feasible solution, although other methods are employed, such as in [16], where a genetic algorithm utilizing the CRLB as the fitness function was presented. Generally, the sensor selection problem can be formulated as a grid-based combinatorial problem, but as the number of possible locations grows, such a solution quickly becomes unfeasible, and other methods or relaxations have to be utilized. An example of this is found in [12], wherein the authors formulate the optimization problem as one of maximizing the signal energy, and thereafter proceed to solve the non-convex optimization using a convex relaxation of the constraints.

One limitation of the noted works is that these do not exploit any available *a priori* knowledge of the expected source signals, such as signal structure or knowledge about the parameters detailing the signal. Often, the structure of the impinging signals is at least partly known, such as in the case of, for example, chirps and tonal audio, as well as several forms of sonar and vibrational signals [17, 18]. Such a structure may be exploited in determining a suitable sensor placement. Reminiscent to the optimal sampling scheme presented in [19], wherein the expected signal structure of spectroscopic signals was exploited to develop an optimal multi-dimensional sampling scheme allowing for the there typically partly known signals, we here propose an optimal sensor placement that exploits the expected signal structure of the signals, illustrating our development with tonal sources, such as voiced speech, sonar, or vibrational signals, as well as chirp signals.

In order to do so, we formulate an optimal sensor placement scheme that minimizes the CRLB of the source localization problem over the range of potential sensor placements, taking into account the spherical propagation of the waves, the expected location areas of the sources, the expected structure of the measured signals, in our initial illustration being the harmonic structure of tonal signals, as well as various forms of connection costs. In this sense, the work also expands on the source localization problem examined in [18], wherein the harmonic structure of tonal sounds was exploited to localize sound sources. Other than the signal structure, there are other types of *a priori* information that may be utilized, such as e.g. the nature of the parameters. For the case when the parameters to be estimated contain a mix of deterministic and stochastic components, it is more suitable to formulate the problem using the hybrid CRLB (HCRLB) [20], which allows for the integration of *a priori* information. In cases where such information exists for all parameters, it may be preferable to instead use the Bayesian CRLB (BCRLB) [21]. By comparing the optimal sensor placement when using the HCRLB, BCRLB, and the CRLB, we examine how the optimization criteria change as the amount of prior knowledge changes, and to what extent the accuracy and precision of the resulting estimator is affected. We also examine the robustness of the different estimators by exploring what happens when the prior information is incomplete or slightly incorrect. In summary, the main contributions of this work are

1. We formulate the problem of optimal sensor placement for structured signals as a convex minimization problem in the context of minimizing the worst case CRLB.

2. We illustrate and evaluate how the optimization problem and solution change when using the HCRLB and BCRLB instead of the worst case CRLB.

3. We present how different variations of the problem formulation, in terms of regularization terms and constraint functions, can be utilized to obtain solutions to real world applications.

4. By formulating robustness using prior distributions, in place of earlier gridding formulations, we substantially reduce the complexity of the resulting optimization problem.

The paper is structured as follows; initially, in the next section, the signal model is introduced together with the optimization problem that we aim to solve. In section 3, the different bounds used in the optimization procedure are detailed and derived. In section 4, numerical examples illustrate the results. Finally, in section 5, we conclude upon the work.

2 Signal Model

Consider a setup where M sensors measure the wavefronts impinging from K (near-field) sources. To better illustrate the methodology we will initially consider the sources to be tonal, but again note that other structures would be similarly applicable. Thus, we adopt a signal model where each of these sources have a harmonically related structure, such as may be expected in e.g., tonal audio, sonar, and vibrational signals, implying that the direct path [1] the kth sound source may be deemed to be well modeled as[2] [23, 24]

$$z_k(t) = \sum_{\ell=1}^{L_k} A_{k,\ell} \exp\{i\omega_{k,\ell}t + i\zeta_{k,\ell}\} \tag{1}$$

where $\omega_{k,\ell}$, $A_{k,\ell} \in \mathbb{R}$, and $\zeta_{k,\ell} \in [0, 2\pi)$ denote the angular frequency, amplitude, and phase for the ℓth harmonic component of the kth source, respectively, for

[1] For notational simplicity, we will here restrict our attention to the direct path wavefront, noting that details on the room acoustic response can be incorporated in the model, if they are (at least partially) known.

[2] For notational simplicity, we here use a complex-valued notation, noting that the complex-valued form of the measured signal may be formed using the discrete-time analytic signal [22].

$t = 0, \dots, N-1$. For tonal sounds, typically the ℓth frequency may be well modelled[3] as $\omega_{k,\ell} = \ell\omega_{k,0}$, where $\omega_{k,0}$ denotes the fundamental frequency, or pitch, of the kth sound source [23]. It should further be noted that for most sound sources, the number of overtones, L_k, is unknown, and typically varies over time. Assuming that the kth source is located at position

$$\mathbf{s}_k = [\; x_k^s \quad y_k^s \quad z_k^s \;]^T \tag{2}$$

the wavefront from source k impinging on the qth sensor, located at

$$\mathbf{m}_q = [\; x_q^m \quad y_q^m \quad z_q^m \;]^T \tag{3}$$

may then be expressed as

$$y_{k,q}(t) = \sum_{\ell=1}^{L_k} \chi_{\ell,k,q}(t) + n_k(t) \tag{4}$$

with the noise-free signal component defined as

$$\chi_{\ell,k,q}(t) = a_{k,q}(\omega_{k,\ell}) A_{k,\ell} \exp\{i\omega_{k,\ell}t + \zeta_{k,\ell}\} \tag{5}$$

and where $n_k(t)$ is an additive noise, here assumed to be well modelled as being zero-mean and circularly symmetric white Gaussian distributed. Furthermore, $a_{k,q}(\omega)$ details the spherical propagation gain and phase rotation of frequency ω for the signal impinging from the kth source onto the qth sensor, i.e., [26]

$$a_{k,q}(\omega) = r_{k,q}^{-(d-1)/2} e^{-i\omega r_{k,q}/c} \tag{6}$$

where $r_{k,q}$ is the distance between the kth source and the qth sensor, i.e.,

$$r_{k,q} = \left\| \mathbf{s}_k - \mathbf{m}_q \right\|_2 \tag{7}$$

and c is the propagation speed of the wavefront, for acoustic sources in air generally being $c = 343$ m/s, $\forall\omega$, whereas the propagation speed in water typically depends on depth and salination [27]. Finally, d denotes the dimensionality of the considered space, i.e., $d = 2$ in a flat space and $d = 3$ in 3-D.

[3] It is worth noting that many forms of tonal sources also exhibit inharmonicity, enabling further forms of structure [25].

In order to illustrate the performance for another structured signal, we will in the following also examine sources where the frequency is time dependent, such that a chirp model may be a better fit, i.e.,

$$z(t) = \sum_{k=1}^{K} A_k \sin(i\omega(t)) \tag{8}$$

where $\omega(t) = \omega_0 + ct$. Here, the frequency is defined as a base frequency, ω_0, with an additive drift term that depends on time, ct. This signal model well describes the sounds made by various animals, e.g., bats and whales.

Independent of the considered signal model, the problem may be formulated as that of determining the placement of M sensors such that the locations of the sources may be well estimated. To determine an optimal sensor placement, we propose minimizing a weighted form of the CRLB associated with the estimate of the K source locations given the M sensor locations. The M sensor locations are selected from a set of possible sensor placements, $\mathcal{M}$, consisting of a predefined grid of candidate sensor locations, the choice of which is dictated by computational as well as environmental considerations. In this work, we have mainly opted for the use of an equidistant cartesian grid, as well as a circular grid. Clearly, the CRLB for the source localization problem will depend on the locations of the M sensors, the source locations, and the source signals. Let the unknown parameters detailing the sources be

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_1 & \dots & \boldsymbol{\theta}_K & \sigma_n^2 \end{bmatrix}^T \tag{9}$$

where σ_n^2 denotes the variance of the additive noise $n_k(t)$ (which, for notational simplicity, is here assumed to be the same for each sensor), and with

$$\boldsymbol{\theta}_k = \begin{bmatrix} \mathbf{s}_k & \omega_{k,0} \end{bmatrix}^T \tag{10}$$

which thus implicitly makes an assumption that the number of sources, K, and the number of overtones for each source, L_k, for $k = 1, \dots, K$, are known. Typically, this cannot be assumed in many practical cases, and the sensor placement should thus be designed to allow for robustness to these assumptions. In this work, we will limit the discussion of robustness to the unknown parameters $\boldsymbol{\theta}_k$, formulating the minimization allowing for the expected source locations and range of pitches, noting that a similar robustness with respect to K and L_K may be implemented similarly, or using a formulation reminiscent to [28, 29].

3 Placement Scheme

Our goal is to find a sensor placement scheme that minimizes the estimation error of the parameters, $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$. By utilizing the assumed structure of the signal, one may form the FIM, which can be used to calculate the covariance matrix related to the maximum likelihood estimator. Let $\mathbf{Y}_{\boldsymbol{\theta}}(m)$ denote N samples measured by sensor m, with probability density function $p(\mathbf{Y}_{\boldsymbol{\theta}}(m))$, parameterised by $\boldsymbol{\theta} \in \mathbb{R}^{P \times 1}$, where P denotes the number of unknown parameters in (9). The FIM for the localization problem at hand, denoted $\mathbf{F}(\mathbf{Y}_{\boldsymbol{\theta}}(m))$, may then be formed as

$$\mathbf{F}(\mathbf{Y}_{\boldsymbol{\theta}}(m)) = \mathbb{E}\left\{\nabla_{\boldsymbol{\theta}} \log\left(p(\mathbf{Y}_{\boldsymbol{\theta}}(m))\right) \nabla_{\boldsymbol{\theta}} \log\left(p(\mathbf{Y}_{\boldsymbol{\theta}}(m))\right)^{H}\right\} \tag{11}$$

where $\mathbb{E}\{\cdot\}$ denotes the statistical expectation, in this case taken with respect to the distribution of $\mathbf{Y}_{\boldsymbol{\theta}}(m)$, which in turn is parametrised by $\boldsymbol{\theta}$, and $\nabla_{\boldsymbol{\theta}}$ denotes the gradient with respect to $\boldsymbol{\theta}$, respectively. There are many different optimality notions related to the error covariance matrix. In this work, we use A-optimality, which amounts to minimizing the trace of the resulting error covariance matrix. Since each of the diagonal elements of the inverse FIM relates to the variance of the associated parameter, the trace of the error covariance matrix will give an estimate of the variance of the overall estimate. From [30], the trace minimization problem can be formulated as

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{trace}\Big(\big(\sum_{m=1}^{\Upsilon} w_m \mathbf{F}(\mathbf{Y}_{\boldsymbol{\theta}}(m))\big)^{-1}\Big) \\ \text{subject to} \quad & \mathbf{1}^T \mathbf{w} \leq M, \\ & w_m \in \{0,1\}, \quad m = 1, 2, \ldots, \Upsilon = |\mathcal{M}| \end{aligned} \tag{12}$$

where $\mathbf{w}$ is the Υ dimensional weight vector indicating if a candidate sensor placement in the set of potential candidate placements, $\mathcal{M}$, is used or not. Here, w_m denotes the m:th element in $\mathbf{w}$. Because the weight vector $\mathbf{w}$ only takes the values 0 or 1, the optimisation problem in (12) amounts to a non-convex combinatorial problem. To form a computationally tractable approximation to (12), one may instead use the convex relaxation

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{trace}\Big(\big(\sum_{m=1}^{\Upsilon} w_m \mathbf{F}(\mathbf{Y}_{\boldsymbol{\theta}}(m))\big)^{-1}\Big) \\ \text{subject to} \quad & \mathbf{1}^T \mathbf{w} \leq M, \\ & w_m \in [0,1], \quad m = 1, 2, \ldots, \Upsilon \end{aligned} \tag{13}$$

where the constraints on $\mathbf{w}$ have been relaxed by allowing it to take on values in the interval $[0,1]$. The resulting relaxed solution will result in a sensor placement, formed by rounding the determined $\mathbf{w}$ values to either 0 or 1, yielding close to the optimal (combinatorial) solution. An actual comparison to the achievable optimal solution is typically unfeasible for non-trivial cases, but a small scale comparison is included in the numerical section, together with observations on the achieved duality gap for the considered examples. Fortunately, as shown in [19, 30], this problem can be relaxed to allow it to be implemented as a semidefinite program, using for example, the projected Newton's method with computational complexity $\mathcal{O}(\dim(\theta)\Gamma^3)$ per iteration [31]. Here, we extend upon the work in [19], which focused on the temporal sampling of (structured) decaying sinusoidal signals. Our formulation is reminiscent to the one in this work, although we extend the formulation to consider both the temporal and spatial sampling of structured near-field sound sources. Furthermore, we introduce several forms of additional constraints as well as computationally efficient formulations for allowing uncertainty in the assumed signal parameters that reduce the computational complexity of the optimization substantially. For completeness and to introduce our notation, we next summarize the steps presented in [19]. Initially, note that for every invertible $P \times P$ matrix, it holds that

$$\text{trace}\left(\mathbf{B}^{-1}\right) = \sum_{p=1}^{P} \mathbf{e}_p^T \mathbf{B}^{-1} \mathbf{e}_p \tag{14}$$

where $\mathbf{e}_p$ denotes the p:th canonical basis vector, i.e., a vector containing a 1 at the p:th position and zeros elsewhere. Moreover, using the Schur complement, it holds that for any positive definite matrix $\mathbf{B}$, scalar μ, and arbitrary vector $\mathbf{a}$,

$$\mu - \mathbf{a}^T \mathbf{B}^{-1} \mathbf{a} \geq 0 \iff \begin{bmatrix} \mathbf{B} & \mathbf{a} \\ \mathbf{a}^T & \mu \end{bmatrix} \succeq 0 \tag{15}$$

where $\mathbf{X} \succeq 0$ denotes that the matrix is positive semidefinite. Utilizing (15), the same matrix, $\mathbf{B}$, minimizes both

$$\underset{\mathbf{B}\succ 0}{\text{minimize}}\ \mathbf{a}^T \mathbf{B}^{-1} \mathbf{a} \tag{16}$$

and

$$\underset{\mu,\mathbf{B}\succ 0}{\text{minimize}}\ \mu \tag{17}$$

$$\text{subject to} \begin{bmatrix} \mathbf{B} & \mathbf{a} \\ \mathbf{a}^T & \mu \end{bmatrix} \succeq 0 \quad , w_m \in [0,1] \tag{18}$$

Hence, (13) can be reformulated as

$$
\begin{aligned}
\underset{\mu,\mathbf{w}}{\text{minimize}} \quad & \sum_{p=1}^{P} \mu_p \\
\text{subject to} \quad & \begin{bmatrix} \sum_{m=1}^{\Upsilon} w_m \mathbf{F}(\mathbf{Y}_\theta(m)) & \mathbf{e} \\ \mathbf{e}^T & \mu_p \end{bmatrix} \succeq \mathbf{0}, \\
& \sum_{m}^{\Upsilon} w_m \mathbf{F}(\mathbf{Y}_\theta(m)) \succ \mathbf{0} \\
& \mathbf{1}^T \mathbf{w} \leq M, \quad w_m \in [0,1], \quad m = 1 \ldots \Upsilon
\end{aligned}
\tag{19}
$$

where $p = 1, \ldots, P$. The minimization in (19) constitutes a semi-definite program that may be solved efficiently using standard convex solvers, such as SeDuMi [32] or SDPT3 [33]. Accounting for imperfect signal information can be done in many different ways. Inspired by [30] and [19], one approach to induce robustness is to consider the worst case scenario. In this setting, the parameter space is partitioned into nonempty sets of possible parameter values, and the minimization is done over all different parameter sets, i.e., $\mathbf{\Theta} = [\theta^1 \cdots \theta^J]$, where J is the cardinality of the set $\mathbf{\Theta}$. The result is that the optimization is formed over all parameters in the set, thereby yielding the sensor selection that is the most robust to all these parameter values. Clearly, if the true parameters are known, and this knowledge used in the optimization, the resulting sensor placement will be preferable to one formed over such a set of parameters. However, in case the assumed prior knowledge is imperfect, this may result in a loss of performance. Optimizing over a set of potential parameters, thereby minimizing the potential discrepancy for all parameters in the set, results in the selection optimizing the worst case CRLB. The placement scheme presented thus far assumes that all parameters are of the same magnitude, a setup that is highly unrealistic in many real world settings. To broaden the usefulness of the proposed placement estimation scheme, it is necessary to introduce some robustness to account for this mismatch. To address the case when the different parameters in the error covariance matrix are of different importance, or of significantly different magnitudes, one may introduce a weighting parameter, as was done in [19]. In this setup, a linear weighting matrix $\mathbf{A}(\theta)$ is introduced yielding the weighted FIM $\tilde{\mathbf{F}}(\mathbf{Y}_\theta(m)) = \mathbf{A}(\theta)\mathbf{F}(\mathbf{Y}_\theta(m))\mathbf{A}(\theta)^T$, which is then used in the optimization instead. An example of when this type of weighting is needed is mentioned in [19], where the CRLB of a damped sinusoid will be dominated by the amplitude estimate. The resulting worst case minimization becomes

$$
\begin{aligned}
\underset{\mu,\mathbf{w}}{\text{minimize}} \quad & \sum_{p=1}^{P} \psi_p \mu_p \\
\text{subject to} \quad & \begin{bmatrix} \sum_{m=1}^{\Upsilon} w_m \mathbf{F}(\mathbf{Y}_\theta(m)) & \mathbf{e} \\ \mathbf{e}^T & \mu_p \end{bmatrix} \succeq \mathbf{0}, \forall p \\
& \forall \theta \in \mathbf{\Theta}, \\
& \sum_{m=1}^{\Upsilon} w_m \mathbf{F}(\mathbf{Y}_\theta(m)) \succ \mathbf{0} \\
& \mathbf{1}^T \mathbf{w} \leq M, \quad \mu_p \leq \lambda_p \quad m = 1 \dots \Upsilon
\end{aligned}
\tag{20}
$$

where ψ_p denotes the weighting of parameter p, and we have also included an upper tolerance on the weights $\mu_p \leq \lambda_p$.

Comparing equations (19) and (20), one may note that the weighing parameter ψ_p has been introduced in (20). The linear weighting matrix $\mathbf{A}(\theta)$ enables reweighting of the FIM by linear transformation, but to ensure invertibility $\mathbf{A}(\theta)$ has to have full rank. Thus, in order to completely disregard a parameter in the FIM, the weighting parameter ψ_p has to be introduced as well. If $\psi_p = 1$ for all values of p, no weighting is introduced and one recovers (19), and by putting $\psi_p = 0$ for some values of p makes it possible to completely disregard those parameters in the minimization.

The reason for including the upper tolerance parameter is to account for possible performance constraints on the parameters, more specifically if there is some upper tolerance constraint on the CRLB of one or more of the parameters. In this formulation, the placement scheme is robust to several performance measures, improving the control of the optimization procedure. One notable drawback with the worst case scenario is tractability. The accuracy of the estimator becomes better the finer the gridding of the parameter space is. However, the finer the gridding, the larger the dimensionality of the problem. For example, the dimensionality of using a parameter space of three parameters, with each grid consisting of 100 points, would mean that the optimisation has to be evaluated at 1 000 000 grid points, for each time point. In order to reduce this complexity, we here introduce the assumption that the parameters have some known distribution, instead of discretizing the parameter set with a grid. In the case when all the parameters are stochastic, one may then instead form the optimisation using the Bayesian FIM (BFIM) [34]. Adapting the notation from [21], the FIM in the Bayesian case consists of two parts,

$$\mathbf{F}_B(\mathbf{Y}_\theta) = \mathbf{F}_D(\mathbf{Y}_\theta) + \mathbf{F}_P(\mathbf{Y}_\theta) \tag{21}$$

where the first part, $\mathbf{F}_D(\mathbf{Y}_\theta)$, comes from the data and the second part, $\mathbf{F}_P(\mathbf{Y}_\theta)$, comes from the *a priori* knowledge. Denote the likelihood of the observed signal by $L_\mathbf{Y}(\theta)$. Then, the (i,j)th element of $\mathbf{F}_D(\mathbf{Y}_\theta)$ is

$$[\mathbf{F}_D(\mathbf{Y}_\theta)]_{i,j} = -\mathbb{E}\left[\frac{\partial^2 L_\mathbf{Y}(\theta)}{\partial\theta_i\partial\theta_j}\right] \tag{22}$$

with the expectation over both the data and θ. Moreover,

$$[\mathbf{F}_P(\mathbf{Y}_\theta)]_{i,j} = -\mathbb{E}\left[\frac{\partial^2 log(p(\theta))}{\partial\theta_i\partial\theta_j}\right] \tag{23}$$

where the expectation is over θ. In this case, the minimization problem in (13) may be reformulated as

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{trace}\Big(\big(\sum_{m=1}^{\Upsilon} w_m \mathbf{F}_B(\mathbf{Y}_\theta(m))\big)^{-1}\Big) \\ \text{subject to} \quad & \mathbf{1}^T\mathbf{w} \le M, \\ & w_m \in [0,1], \quad m = 1,2,\dots,\Upsilon \end{aligned} \tag{24}$$

where $\mathbf{F}_B(\mathbf{Y}_\theta(m))$ denotes the BFIM, with the minimized bound being the Bayesian Cramér-Rao lower bound (BCRLB).

Similar to the worst case bound investigated above, the BFIM forms a minimax bound. The difference is that the CRLB utilized in (12) yields a lower bound for all *unbiased* estimators; in contrast, the BCRLB bound yields a lower bound for all estimators, both *biased* or *unbiased*. Furthermore, the BFIM has the advantage of computational tractability. Given that the distributions are known and computable, the dimensionality of the problem only grows with the number of parameters, as opposed to the worst case estimator considered above, where the dimensionality depended not only on the number of parameters, but also on their gridding. An obvious drawback of the BFIM is that the prior distribution is assumed to be known[4].

[4] The distributions also have to be Lipschitz [35].

In many practical applications, the parameter set is best modeled as a mixture of both *deterministic* and *stochastic* parameters. In such cases, it makes more sense to instead minimize the hybrid Cramér-Rao lower bound (HCRLB) (see e.g. [20] and [36]). To form the HCRLB, assume we have U deterministic and Q stochastic parameters, and let $\boldsymbol{\psi} = [\boldsymbol{\theta}^T, \boldsymbol{\phi}^T]^T$, where $\boldsymbol{\theta} \in \mathbb{R}^U$ and $\boldsymbol{\phi} \in \mathbb{R}^Q$ are the deterministic and stochastic parameters detailing the signals, respectively. Introduce the joint probability density function $f_{\chi,\boldsymbol{\phi}}(\chi, \boldsymbol{\phi}; \boldsymbol{\theta})$. The hybrid Fisher information (HFIM) is then formed as

$$\mathbf{F}_H(\boldsymbol{\theta}) = \mathbb{E}\left\{\frac{\partial \log f_{\chi,\boldsymbol{\phi}}(\chi, \boldsymbol{\phi}; \boldsymbol{\theta})}{\partial \boldsymbol{\psi}^T} \frac{\partial \log f_{\chi,\boldsymbol{\phi}}(\chi, \boldsymbol{\phi}; \boldsymbol{\theta})}{\partial \boldsymbol{\psi}}\right\} \tag{25}$$

$$= \begin{bmatrix} \mathbf{F}_{\boldsymbol{\theta}} & \mathbf{F}_{\boldsymbol{\theta},\boldsymbol{\phi}} \\ \mathbf{F}^T_{\boldsymbol{\theta},\boldsymbol{\phi}} & \mathbf{F}_{\boldsymbol{\phi}} \end{bmatrix} \tag{26}$$

The fundamental difference between (25) and (11) is that in (25) the matrix has been divided into one deterministic, one stochastic, and two joint blocks yielding the HCRLB

$$\mathbb{E}\{(\hat{\boldsymbol{\psi}} - \boldsymbol{\psi})(\hat{\boldsymbol{\psi}} - \boldsymbol{\psi})^T\} \geq \mathbf{F}_H^{-1}(\boldsymbol{\theta}) \tag{27}$$

The deterministic FIM may be derived using the marginal probability density of the measurement vector, i.e.,

$$f_\chi(\chi; \boldsymbol{\theta}) = \int_{\mathbb{R}^Q} f_{\chi,\boldsymbol{\psi}}(\chi, \boldsymbol{\psi}; \boldsymbol{\theta}) d\boldsymbol{\psi} \tag{28}$$

yielding

$$\mathbf{F}(\boldsymbol{\theta}) = \mathbb{E}\left\{\frac{\partial \log f_\chi(\chi; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \frac{\partial \log f_\chi(\chi; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right\} \tag{29}$$

The deterministic CRLB is asymptotically tight for high signal to noise ratios (SNRs) or large sample sizes. This property is typically not enjoyed by the HCRLB [20, 37]. As with the BCRLB, one assumption for using the HCRLB is that the distributions are known and Lipschitz. As is suggested by the name, the HCRLB is a tradeoff between the BCRLB and the deterministic CRLB, offering the traceability of the Bayesian bound, while not being limited to unbiased estimators only. Significantly, the computational complexity of the problem reduces as the amount of *a priori* knowledge is increased. For each stochastic parameter the complexity decreases by a factor P, where the complexity of the worst case CRLB with b parameters is of the order $(\Upsilon P)^b$.

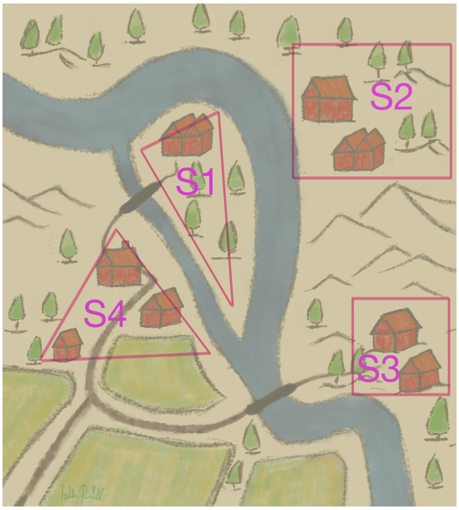


Figure 1: In a real world sensor placement scenario, not all sensor placements have the same cost.

In many applications, the cost of placing sensors is also not equal for all candidate positions. This is illustrated in Figure 1, where placing sensors on top of a mountain or in the river is probably more difficult/expensive than placing the sensors in the open field. Moreover, in many settings the cost of placing the first sensor in a region is often more expensive than placing the rest of the sensors, a notion familiar from infrastructure where the cost of laying the wire is expensive but connecting further sensors to the wire when it is already in place is less expensive. This, and other design features, can be incorporated either as a penalty that assigns extra cost to a sensor placement scheme in which the sensors are scattered, or alternatively as constraints. To exemplify, suppose the terrain is divided into R regions $S_1, \dots, S_R$, as illustrated in Figure 1. Moreover, assume that one sensor is placed in region S_1, such that the wiring is already in place in this region. Then, placing an extra sensor in region S_1 will be cheaper than placing a sensor in any of $S_2, \dots, S_R$, since $S_2, \dots, S_R$ all come with the extra cost of laying the cable. As such, the resulting design will prefer a sensor placement scheme that places the sensors in the same region, employing a new region only in the case when the benefit outweighs the cost. Such a penalty can be implemented in terms of a group penalty, $V(\mathbf{w}) = \sum_{j=1}^{S_R} ||\mathbf{w}_j||_2$, where the weight vector $\mathbf{w}$ has been partitioned into the different groups and $\mathbf{w}_j$ represents the parts of the weight vector belonging to group j. This yields

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \operatorname{trace}\Big(\big(\sum_{m=1}^{\Upsilon} w_m \mathbf{F}(\mathbf{Y}_\theta(m))\big)^{-1}\Big) + \nu V(\mathbf{w}) \\ \text{subject to} \quad & \mathbf{1}^T \mathbf{w} \leq M, \\ & w_m \in [0,1], \quad m = 1, 2, \dots, \Upsilon \end{aligned} \tag{30}$$

where the parameter ν is a scalar governing the amount of total variation regularization. It is worth noting that in this setup, the number of selected sensors is still M. Alternatively, sometimes one wishes to study only a subset of the possible sensor positions, for example to allow for efficient wiring or hardware restrictions. One such constraint example could be to force all the sensors to be in the same predefined region. If the region of interest is a square, the optimization problem becomes

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \operatorname{trace}\Big(\big(\sum_{m=1}^{\Upsilon} w_m \mathbf{F}(\mathbf{Y}_\theta(m))\big)^{-1}\Big) \\ \text{subject to} \quad & \mathbf{1}^T \mathbf{w} \leq M, \\ & \ell \leq m_m \leq u, \quad m = 1, 2, \dots, \Upsilon \\ & w_m \in [0,1], \quad m = 1, 2, \dots, \Upsilon \end{aligned} \tag{31}$$

where the extra variables ℓ and u have been introduced, representing the borders of the square in either, some, or all dimensions. Other forms of regions can be included similarly, by introducing a corresponding function indicating if the placement is in the region or not. The major difference between applying a penalty function or a constraint is that a constraint has to be met, whereas with a penalty function the sensor placement can deviate if the benefit of deviating is greater than the cost. Thus, depending on if the constraints are fixed or if there is some flexibility decides if the design should be implemented as a penalty function or as a set of constraints.

In contrast to the previous design setup, one might also want to enforce some separation among the selected sensor placements, to avoid clustering. Sensor separation may be a desired feature especially in terms of robustness. Placing all the sensors close to each other, although possibly being more optimal in terms of CRB, may make the positioning sensitive to environmental impact, as well as detrimental effort such as the possible destruction of the sensors, or topological differences that have not been addressed here, potentially making it desirable to separate the sensors by some minimum distance. This too can be incorporated in the design, but in terms of constraints on the variables. Introduce the two-diagonal matrix

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & -1 & 1 \end{bmatrix} \tag{32}$$

The constraints enforcing the separation between the sensors can then be implemented as

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{trace}\Big(\big(\sum_{m=1}^{\Upsilon} w_m \mathbf{F}(\mathbf{Y}_\theta(m))\big)^{-1}\Big) \\ \text{subject to} \quad & \mathbf{1}^T\mathbf{w} \le M, \\ & w_m \in [0,1], \quad m = 1,2,\dots,\Upsilon \\ & ||\mathbf{D}m|| \ge \eta \end{aligned} \tag{33}$$

where η is the minimum separation required. Clearly, these two designs may be combined by introducing both constraints. So far, we have considered problems of minimising the trace for a given number of sensors. This problem can naturally be reformulated to another problem whereby instead of fixing the number of sensors to be placed, we fix a budget, B, which is added to the optimisation problem in terms of a constraint. The cost of placing a sensor at position i is then associated with the cost c_i, with the total cost for all sensors collected in the vector $\mathbf{c} = [\ c_1, \dots, c_\Upsilon\ ]$. The associated problem then becomes

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{trace}\Big(\big(\sum_{m=1}^{\Upsilon} w_m \mathbf{F}(\mathbf{Y}_\theta(m))\big)^{-1}\Big) \\ \text{subject to} \quad & w_m \in [0,1], \quad m = 1,2,\dots,\Upsilon \\ & \mathbf{c}^T\mathbf{w} \le B \end{aligned} \tag{34}$$

Analogous to the previous formulation, we might seek a sensor placement where the cost of connecting the different sensors, measured as the distance between them, is minimized. Such a formulation might arise when the sensors are to be connected by a cable, and one seeks a design that minimizes the length of the cable. Typically such a design feature is incorporated as a penalty function where one assigns a cost to each arc connecting two sensors, such that

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{trace}\Big(\Big(\sum_{m=1}^{\Upsilon} w_m \mathbf{F}(\mathbf{Y}_\theta(m))\Big)^{-1}\Big) + \sum_{i,j=1}^{\Upsilon} k_{i,j}||w_i - w_j||_q \\ \text{subject to} \quad & \mathbf{1}^T\mathbf{w} \leq M, \\ & w_m \in [0,1], \quad m = 1, 2, \dots, \Upsilon \end{aligned} \tag{35}$$

where $k_{i,j}$ is a weight associated with each arc and $|| \cdot ||_q$ denotes the q-norm. Different choices for q yield different sensor placement setups. Another important class of penalty functions that may be utilized is the deadzone-linear penalty function. Such a penalty assigns no cost when the sensors are within the deadzone with width a, but the cost increases linearly as the sensors deviate from a. Naturally, one may combine the different types of design feature such as minimum cost together with a group constraint.

We conclude this section with a remark regarding the introduced grid of potential candidate sensor placements. Clearly, the use of a grid impose some restrictions on the available sensor placements. This is done partly to allow for that sensors typically occupies some space, making it unrealistic to allow sensors to be be placed arbitrarily close to each other; furthermore, the used grid allows the problem to be solved in a computationally efficient manner, with the size of the sensor grid having a direct impact on the computational complexity of the optimization as the problem needs to be evaluated at each grid point. In this work, we have limited our attention to equidistant cartesian grids as well as circular grids. Other grids, more suitable to the environment of considered applications are of course possible.

4 Numerical

To illustrate the proposed sensor placement scheme, we examine the results using both simulated and experimental data.

4.1 Simulated data - Tonal model

Initially, consider a hypothetical setup where two speakers are expected to be located at coordinates (0,0) and (3,3), respectively, such as could be expected, for instance, in an auditorium. The fundamental frequencies of the two speakers are assumed to be 200 ± 5 Hz and 180 ± 5 Hz, respectively, although the exact

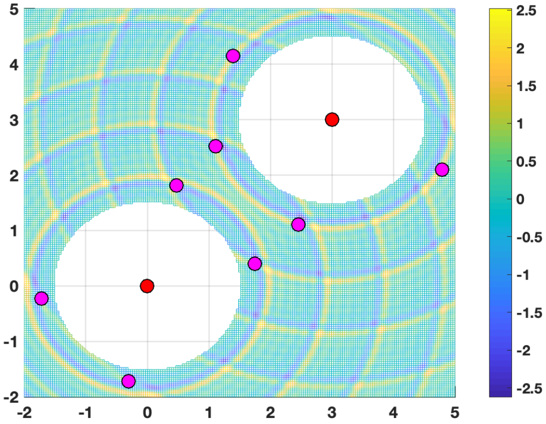


Figure 2: The figure illustrates the found optimal sensor placement (magenta) for two tonal sources (red). Shown in the background is the expected signal strength for the possible positions. As seen, the sensor placement scheme has selected sensor placement where the signal strength is expected to be strong, while spread around the expected source locations.

pitches are unknown. The sources have $L_1 = L_2 = 5$ overtones. For simplicity, we limit our attention to the planar scenario, so that $d = 2$. We consider an equidistant grid of potential sensor locations consisting of $|\mathcal{M}| = 9820$ candidate locations in the region $x, y \in [-2, 5]$, except, for presentational clarity, within a ≤ 1.5 m distance from the sources. Figure 2 illustrates the setup, where each pixel represents a candidate sensor location (except in the vicinity of ≤ 1.5 form each source), and the actual source locations are marked in red. The magenta marks indicate the optimal placements if placing $M = 8$ sensors, together with the expected strength of the sound field at each location. As can be seen from the figure, the optimization has selected sensor placements where the expected signal is strong. Because of the superposition of the signals, the regions with high signal power are spread out, and so in this case there is no need to enforce sensor separation. Here, we have used $M = 8$, $\lambda = 0.10$, $\psi_p = 1$, $\forall p$. The uncertainty in the pitches has been accounted for by adding an equally fine gridding for both frequency parameters, consisting of 20 gridpoints around the proposed frequencies. The simulated data in the numerical section used a square

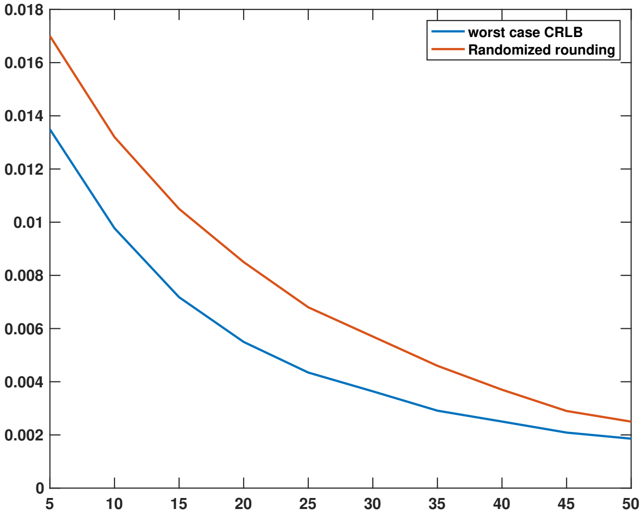


Figure 3: The expected variation of the source position estimates as a function of number of sensors, comparing the worst case estimator to the randomized rounding method.

equidistant set of gridpoints, with the signals sampled at 13.8 kHz. The simulated data length was $N = 500$, and the noise variance was set to $\sigma^2 = 0.01$.

Figure 3 illustrates how the variance of the source location decreases with the number of placed sensors, when placing the sensor optimally for each number of sensors using (19), as compared to the randomized rounding method presented in [31]. The figure shows the minimum expected variance over the x- and y-directions for a setup with one source, with 5 harmonics, placed at the origin as the number of sensors grow, also illustrating how one may determine the minimum required number of sensors to use to allow for a desired localization accuracy. As can be expected, the marginal gain of adding a sensor decreases as the number of sensors increases, with the amount of added useful information available in the localization problem decreasing for each additional sensor. It should be noted that for each setup, the sensors are here placed anew, meaning that when placing five sensors all five sensors are placed simultaneously, instead of keeping the optimal sensor setup for four sensors and adding one.

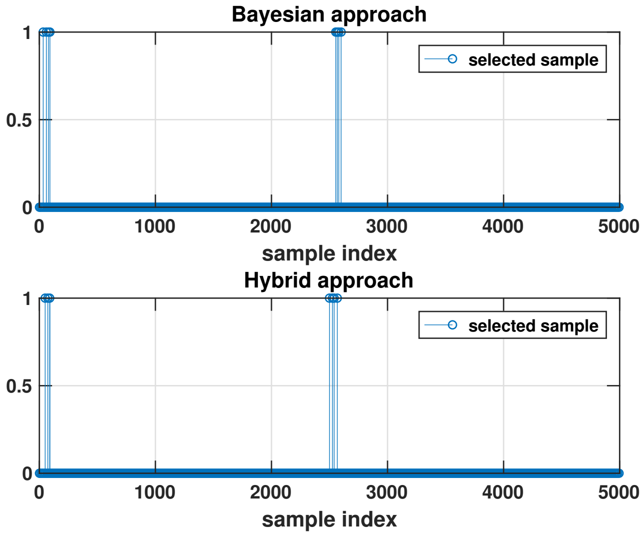


Figure 4: The figure shows the proposed optimal sensor selection for the chirp signal for the Bayesian and hybrid cases, respectively.

4.2 Chirp data

As an illustration of how the method works for signals with time dependent frequency, we also present the resulting sensor placement for the chirp signal consisting of one sinusoid with base frequency $\omega_0 = 180$ Hz and a drift term of $c = 0.5$. The results are displayed in Figure 4, for both the Bayesian and Hybrid approach. In the setup, the sensors were placed on a grid, and each grid point assigned a linear index. The figure shows which sensors were selected with respect to this linear index. It may be noted that the selected indices for the Bayesian and the hybrid are quite similar, although the resulting variance of the Bayesian estimate is somewhat lower than the hybrid version, 0.0053 and 0.0064, respectively.

4.3 Real data

Next, we consider a real data experimental setup, using the data set reported on in [18]. This data was collected in an anechoic chamber of approximate dimensions $4 \times 4 \times 3$ meters. The sensor/source setup can be seen in Figures 5 and 6, where the sources are located at $\mathbf{s}_1 = [-0.4866, 1.0420]$ and $\mathbf{s}_2 = [0.3548, -1.2818]$. The sources each play the same signal, a SQAM violin signal. As a first step, we

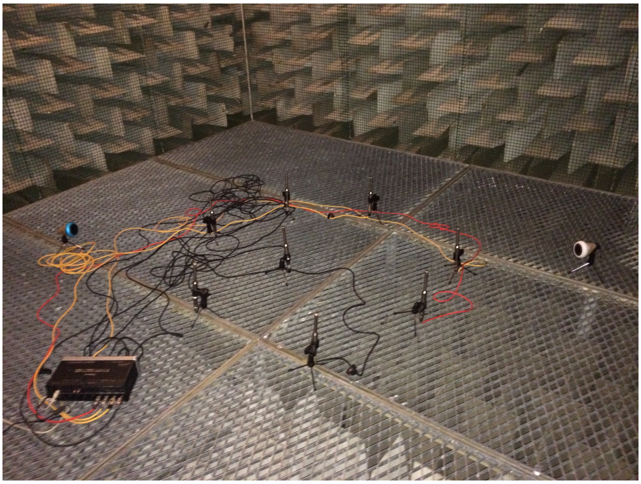


Figure 5: The setup of sources and sensors in the anechoic chamber.

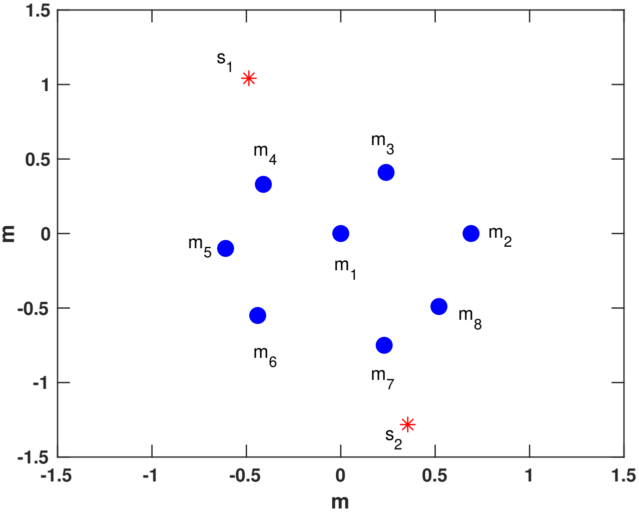


Figure 6: The setup of sources and sensors used in the real experiment.

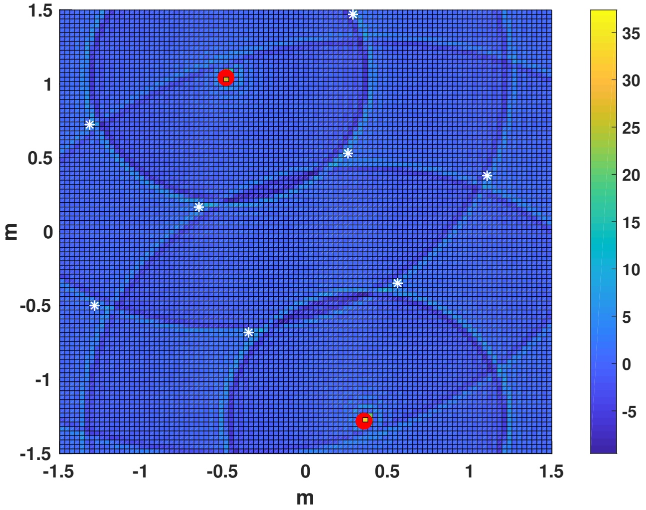


Figure 7: The figure illustrates the found optimal sensor placement (white) for two audio sources (red) for the real data case. Shown in the background is the expected signal strength for the possible positions.

used the HALO method presented in [18] to estimate the signal parameters and locations. The estimates indicated that the violin signal may be well described as having fundamental frequency $\hat{\omega}/2\pi = 198.0$ Hz with $\hat{L} = 14$ overtones. Next, we proceeded to use these estimates in the sensor placement scheme to find an optimal sensor placement. The result for placing 8 sensors is shown in Figure 7, where the uncertainties in pitch have been allowed to be $\pm 5Hz$, again gridded using 20 gridpoints in the uncertainty area for each parameter. Again the selected sensor placement is in locations where the signal is strong. Given the low dimensionality of the problem, we also compared which of the sensors that yielded most information, for varying number of sensors, comparing this to the combinatorial solution. We found that the sensor selections were the same for both approaches, indicating that the optimization scheme yields the same sensor placement as the combinatorial search.

4.4 Worst case estimator vs stochastic estimators

To illustrate how the worst case sensor placement in (19) compares with the Bayesian and hybrid optimization schemes, we simulate data from one sinusoid with frequency $f = 180$ Hz, and $L = 5$, located at [0,0]. We introduce an uncertainty of ± 0.25m for the x and y coordinates, and a frequency uncertainty of $\pm$10Hz. For the worst case placement scheme, we grid the parameter space around the frequency with 40 points for each grid, i.e., forming the parameter grid $f = [170, 190]$, and a grid of 50 points around the source coordinates. For the hybrid placement scheme, we assume that the frequency has a Gaussian distribution, $f \in \mathcal{N}(180, 0.32)$, and place a grid around the source coordinates similar to the one employed for the worst case scheme. Finally, for the Bayesian setup, we assume the same distribution for the frequency as in the hybrid case, and further assume that both source coordinates follow a normal distrbution, $x, y \in \mathcal{N}(0, 0.08)$.

It is worth noting that if selecting a uniform prior for the stochastic placement scheme, this will coincide with the worst case placement scheme as J_P reduces to zero, and the second part, J_D, simplifies to the Fisher information for the midpoint of the interval corresponding to the uniform prior. Figures 8-11, illustrating the performance of the placement schemes, are generated by a two-step procedure. First, the optimal sensor placement is found by minimizing the optimization problem associated with the worst case CRLB, the HCRLB, and the BCRLB. Given the sensor positions found, the resulting RMSE curves for the different parameters were formed. Here, the MSE estimates are formed using the corresponding maximum likelihood (ML) and maximum a posteriori (MAP) estimators, depending on the assumed problem formulation, i.e., when designing the optimal sensor placement under the Bayesian framework, using (24), we then estimate the corresponding sensor location using the corresponding MAP estimator. The simulation was repeated for different number of sensors and different SNRs. The performance is also compared to a compressed sensing approach, employing a random sensor placement, as well as the randomized rounding method introduced in [31]. For these cases, the corresponding location estimator is the corresponding maximum likelihood estimator. The figures show the results of averaging 100 Monte Carlo simulations. As can be seen, the Bayesian placement scheme in combination with the corresponding MAP estimator gives the lowest variance, as expected given that it is able to efficiently exploit the available *a priori* knowledge. The compressed sensing approach performs the worst, which is also to be expected given that it exploits no signal or structure information, either for the placement or the estimation problem. Also displayed in the figure is the theoretical lower bound for the Bayesian approach, and one may note that it coincides with its corresponding simulated curve. It should be stressed that the different placement schemes will yield different sensor placements, and as a result different estimation bounds. These bounds are illustrated in Figure 12, indicating the notable performance limits resulting from the used optimality criteria. One may note that the theoretical bound is the lowest for the Bayesian approach and the highest for the compressed sensing case. The behaviour of these bounds shows a similar behaviour as a function of SNR. Finally, a note on the number of evaluations needed for the different estimators. The minimization using the BCRLB requires one evaluation for each grid point, yielding a total number of optimization rounds of Υ. For the hybrid estimator, the same problem required $\vartheta\Upsilon$ evaluations of the optimization problem, and for the worst case estimator the minimization required $\vartheta\Upsilon \times \vartheta\Upsilon$ evaluations of the optimization problem, where ϑ denotes the size of the used grid. In cases where the expectation in the FIM can not be estimated analytically the MAP has to first be estimated. This may be done using Monte Carlo methods When comparing the proposed methods to alternative methods, it may be noted that the proposed methods have a better performance in terms of MSE for both range and frequency. This performance gap stems form the fact that the proposed methods utilize the (partial) *a priori* knowledge of the signal structure. The computational burden will depend on how fine the gridding is, an aspect where random methods may offer a lower complexity at the cost of lower performance.

4.5 Penalised worst case optimisation

We proceed to consider how incorporating a total variation penalty on the optimisation changes the sensor placement. Again, we utilize the above simulated example, but now we also enforce a sensor separation of at least 0.2 (m) to avoid clustering. We consider the case when the regularization parameter is $\nu = 0.7$. The result is illustrated in Figure 13. As can be seen the sensor separation is much smaller as compared to the case with no penalty enforced. The parameter ν is a design parameter related to the cost of sensor separation. We also compare the case in (31), when the sensors are constrained to lie in a small region, $2.5 \leq x \leq 4.5$, $-1.5 \leq y \leq 0.5$, and when the same constraint is applied as a penalty. The results when placing 4 sensors are shown in Figure 14. For this case, we chose the hyper-parameter ν to have the value 0.2. As can be seen in the figure, the sensor placement complies with the constraints. Finally, we conclude with an example of the group penalty in (30). In this example, we use a signal with frequency 200 Hz and 5 overtones, located at position (0,0). All other simulation variables are

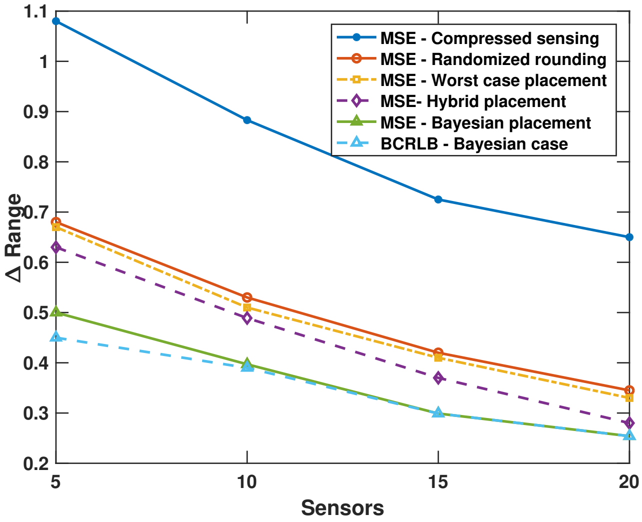


Figure 8: The RMSE of the range vs the number of sensors, given sensor locations derived using five different methods.

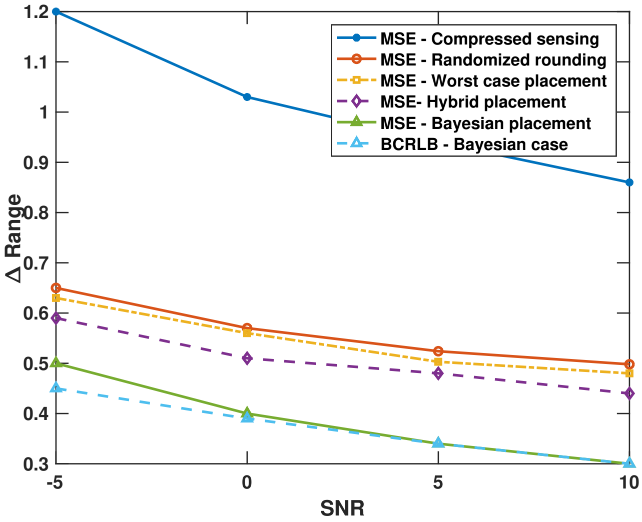


Figure 9: The RMSE of the range vs the SNR of sensors, given sensor locations derived using five different methods.

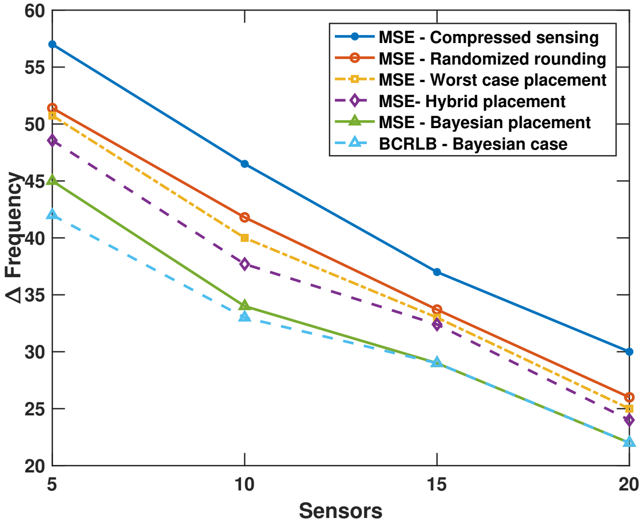


Figure 10: The RMSE of the frequency vs the number of sensors, given sensor locations derived using five different methods.

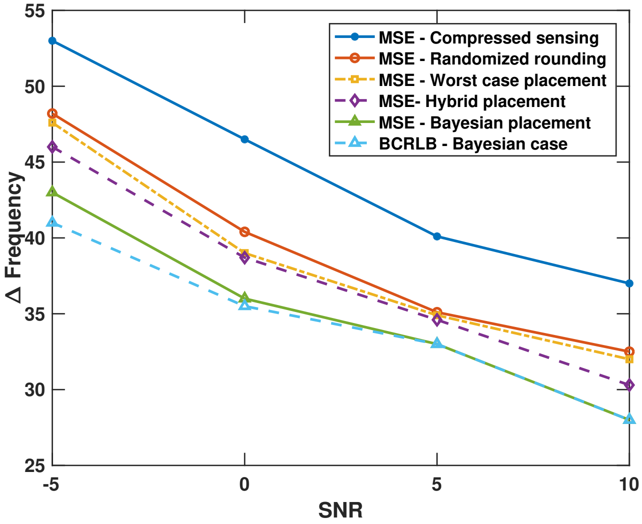


Figure 11: The RMSE of the frequency vs the number of sensors, given sensor locations derived using five different methods.

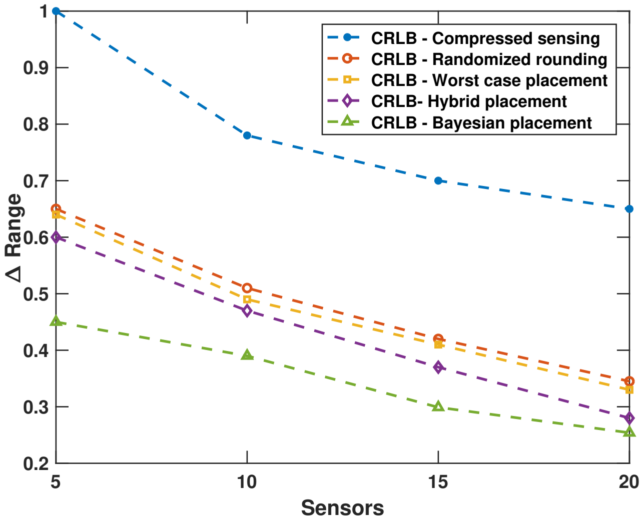


Figure 12: The theoretical lower bounds for the different placement approaches.

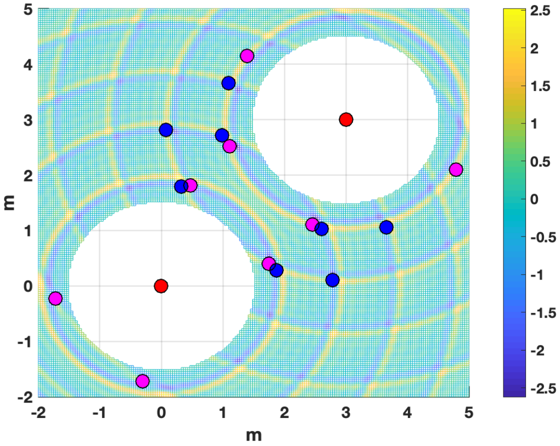


Figure 13: The proposed sensor placement of eight sensors when using the penalized worst case optimization. The proposed sensor placement using the penalty is illustrated in pink circles, whereas the unpenalized sensor placement is illustrated in white.

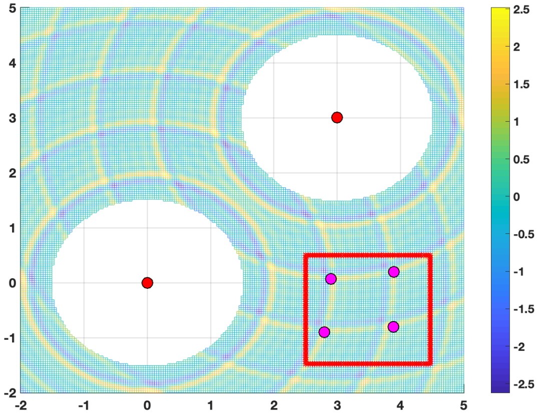


Figure 14: The proposed sensor placement of five sensors when using the penalized worst case optimization and the box constrained optimization.

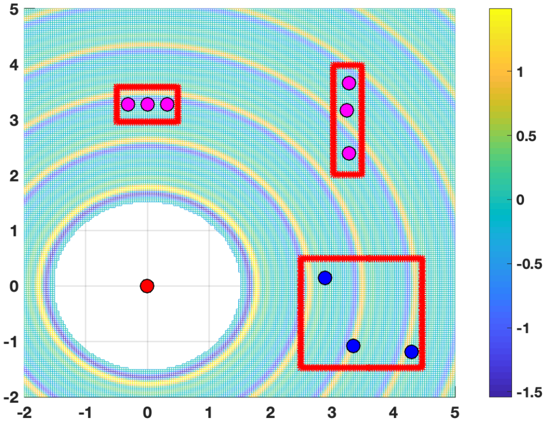


Figure 15: The proposed sensor placement of three sensors when using the group penalty.

as before. Consider a set of 9 potential sensors, divided into three groups, out of which $M = 3$ sensors should be selected. Here, the parameter ν has been set to 3. The setup is presented in figure 15, where it may be noted that the selected sensors all belong to the same group, marked blue in the picture. Although the signal values are higher for some individual sensors in the other groups, the group penalty selects the sensors so that they belong to the same group, which is what was desired.

5 Conclusion

In this paper, we have proposed a method for determining a sensor placement that minimizes a formulation of the CRLB for the corresponding localization problem. Exploiting the expected signal model, the scheme minimizes a relaxation of the resulting A-optimality formulation of the localization problem, and can be efficiently implemented as a semi-definite program. We studied how the optimization problem changes when using performance measures such as the worst case CRLB, the Bayesian CRLB, and the hybrid CRLB. Noting the amount of a priori knowledge needed, and the reduction in complexity associated with more a priori information. For each known parameter, the optimization problem reduces in complexity by a factor equal to the grid size. As an alternative, one could initially use a coarse candidate grid, make an initial sensor placement, and then repeat the optimization with a finer grid around the selected points. Given that the sensor placement is a problem that may typically be solved offline, and that the required computations may be done using parallel implementations, the computational burden may be reduced allowing for the efficient use of the method. Additionally, we have extended the problem of finding an optimal sensor placement to also include different design features, as may be expected in real applications, such as minimum sensor separation and sensor placement cost. The proposed methods have been evaluated by numerical experiments for both simulated and experimental data.

A The Fisher Information Matrix

Introduce the signal at sensor q and the signal derived from source k as

$$y_{:,q}(t) = \sum_{k=1}^{K} y_{k,q}(t) \quad \text{and} \quad y_{k,:}(t) = \sum_{q=1}^{M} y_{k,q}(t)$$

and let $\tilde{\mathbf{y}}_t$ denote the measured data vector at time t, i.e.,

$$\tilde{\mathbf{y}}_t = \left[\; y_{:,1}(t) \quad \dots \quad y_{:,M}(t) \;\right]^T = \tilde{\mathbf{u}}_t + \tilde{\mathbf{n}}_t$$

where $\tilde{\mathbf{n}}$ is defined similar to $\tilde{\mathbf{y}}$, and sensor

$$\tilde{\mathbf{u}}_t = \left[\; \Upsilon_1(t) \quad \dots \quad \Upsilon_M(t) \;\right]^T$$

with the noise-free measurements at sensor q defined as

$$\Upsilon_q(t) = \sum_{k=1}^{K} \sum_{\ell=1}^{L_k} \chi_{\ell,k,q}$$

Then, according to Slepian-Bangs formula (see, e.g., [38]), when $d = 2$, the i, jth element in $\mathbf{F}(\mathbf{Y}_\theta(m))$ may then be expressed as

$$\left[\mathbf{F}(\mathbf{Y}_\theta(m))\right]_{i,j} = \frac{2}{\sigma_n^2} \Re \left\{ \sum_{t=0}^{N-1} \frac{\partial \tilde{\mathbf{u}}_t^H}{\partial \theta_i} \frac{\partial \tilde{\mathbf{u}}_t}{\partial \theta_j} \right\}$$

where $\Re(\cdot)$ denotes the real part, with

$$\frac{\partial \tilde{\mathbf{u}}(t)}{\partial x_k^s} = \sum_{q=1}^{M} \sum_{\ell=1}^{L_k} \left(\frac{\kappa_{q,k}}{2\xi_{q,k}^2} + \frac{i\omega_{k,0}\kappa_{q,k}}{c\xi_{q,k}} \right) \chi_{\ell,k,q}$$

$$\frac{\partial \tilde{\mathbf{u}}(t)}{\partial \omega_{k,0}} = \sum_{q=1}^{M} \sum_{\ell=1}^{L_k} \ell i \left(t - \frac{r_{k,q}}{c} \right) \chi_{\ell,k,q}$$

where $\xi_{q,k} = ||\mathbf{m}_q - \mathbf{s}_k||_2$ and $\kappa_{q,k} = (x_q^m - x_k^s)$ for the qth sensor and kth source, respectively. The derivatives with respect to the y-coordinate is defined analogously. For the hybrid and Bayesian bounds, we utilize the formula given in (21), with one part denoting information from the data and the other part denoting information from the prior knowledge. Assuming independent Gaussian distributions, the components become $J_P = \Sigma_\theta^{-1}$, where Σ_θ has the form of a diagonal matrix with $[\Sigma_\theta]_{i,i} = \sigma_{\theta_i}^2$, where $\sigma_{\theta_i}^2$ is the variance for parameter i. The components in J_D are given by [35]

$$\left[\mathbf{F}(\mathbf{Y}_\theta(m))\right]_{i,j} = \mathbb{E}_{\theta,\tilde{\mathbf{y}}_t} \left\{ \frac{2}{\sigma_n^2} \Re \left\{ \sum_{t=0}^{N-1} \frac{\partial \tilde{\mathbf{u}}_t^H}{\partial \theta_i} \frac{\partial \tilde{\mathbf{u}}_t}{\partial \theta_j} \right\} \right\}$$

References

[1] V. Kekatos, G. B. Giannakis, and B. Wollenberg, "Optimal Placement of Phasor Measurement Units via Convex Relaxation," *IEEE Trans. on Power Systems*, vol. 27, no. 3, pp. 1521–1530, Aug 2012.

[2] M. Galetto and B. Pralio, "Optimal sensor positioning for large scale metrology applications," *Precision Engineering*, vol. 34, no. 3, pp. 563–577, 7 2010.

[3] J. Zhang, S. P. Chepuri, R. C. Hendriks, and R. Heusdens, "Microphone subset selection for MVDR beamforming based noise reduction," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 26, no. 3, pp. 550–563, 2018.

[4] G. Ciavarrini, M. S. Greco, and A. Vecchio, "Geolocation of Internet hosts: Accuracy limits through Cramér-Rao lower bound," *Computer Networks*, vol. 135, pp. 70–80, April 2018.

[5] C. Jiang, Z. Chen, R. Su, and Y. C. Soh, "Group Greedy Method for Sensor Placement," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2249–2262, 5 2019.

[6] J. Neering, M. Bordier, and N. Maizi, "Optimal Passive Source Localization," *Int. Conference on Sensor Technology and Applications*, 2007.

[7] N. H. Nguyen and K. Dogancay, "Optimal Geometry Analysis for Multistatic TOA Localization," *IEEE Transactions on Signal Processing*, vol. 64, pp. 4180 – 4193, 2016.

[8] M. Hamdollahzadeh, S. Adelipour, and F. Behnia, "Optimal Sensor Configuration for Two Dimensional Source Localization Based on TDOA/FDOA Measurements," *17th Intern. Radar Symposium*, 2016.

[9] D. Pearson, S. U. Pillai, and Y. Lee, "An Algorithm for Near -Optimal Placement of Sensor Elements," *IEEE Transactions on Information Theory*, vol. 36, pp. 1280 – 1284, 1990.

[10] B. Yang, "Different Sensor Placement Strategies for TDOA Based Localization ," *IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP*, 2007.

[11] J. R. Jensen, M. G. Christensen, and S. H. Jensen, "Nonlinear Least Squares Methods for Joint DOA and Pitch Estimation," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 21, no. 5, pp. 923–933, 2013.

[12] S. Joshi and S. Boyd, "Sensor Selection via Convex Optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, 2 2009.

[13] S. Liu, S. P. Chepuri, M. Fardad, E. Masazade, and G. Leus P. K. Varshney, "Sensor Selection for Estimation with Correlated Measurement Noise," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3509–3522, July 2016.

[14] H. Jamali-Rad, A. Simonetto, X. Ma, and G. Leus, "Distributed Sparsity-Aware Sensor Selection," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 5951–5964, 11 2015.

[15] S. P. Chepuri and G. Leus, "Sparsity-Promoting Sensor Selection for Non-Linear Measurement Models," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 684–698, 2 2015.

[16] L. Pengzheng, H. Xiaotao, W. Jian, and M. Xile, "Sensor Placement of Multistatic Radar System by using Genetic Algorithm," *IEEE International Geoscience and Remote Sensing Symposium*, 2012.

[17] T. van Waterschoot and M. Moonen, "Comparison of Linear Prediction Models for Audio Signals," *EURASIP Journal on Audio, Speech, and Music Processing*, 2008.

[18] S. I. Adalbjörnsson, T. Kronvall, S. Burgess, K. Åström, and A. Jakobsson, "Sparse Localization of Harmonic Audio Sources," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 117–129, January 2016.

[19] J. Swärd, F. Elvander, and A. Jakobsson, "Designing Sampling Schemes for Multi-Dimensional Data," *Elsevier Signal Processing*, vol. 150, pp. 1–10, 9 2018.

[20] H. Messer, "The Hybrid Cramer-Rao Lower Bound - From Practice to Theory," in *Fourth IEEE Workshop on Sensor Array and Multichannel Processing, 2006.*, 12–14 July, 2006, pp. 304–307.

[21] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part IV, Optimum Array Processing*, John Wiley and Sons, Inc., 2002.

[22] S. L. Marple, “Computing the discrete-time “analytic” signal via FFT,” *IEEE Trans. Signal Process.*, vol. 47, no. 9, pp. 2600–2603, 9 1999.

[23] M. Christensen and A. Jakobsson, *Multi-Pitch Estimation*, Morgan & Claypool, San Rafael, Calif., 2009.

[24] Jun Ling, Hao He, Jian Li, W. Roberts, and P. Stoica, “Covert underwater acoustic communications,” *J. Acoust. Soc. Am.*, vol. 128, no. 5, pp. 2898–2909, 2010.

[25] F. Elvander and A. Jakobsson, “Defining Fundamental Frequency for Almost Harmonic Signals,” *submitted to IEEE Transactions on Signal Processing*, 2020.

[26] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*, Prentice Hall, Englewood Cliffs, N.J., 1993.

[27] D. A. Abraham, *Underwater Acoustic Signal Processing*, Springer Nature Switzerland, 2019.

[28] M. G. Christensen, A. Jakobsson, and S. H. Jensen, “Joint High-Resolution Fundamental Frequency and Order Estimation,” *IEEE Trans. Audio, Speech, Language Process*, vol. 15, no. 5, pp. 1635–1644, July 2007.

[29] S. I. Adalbjörnsson, A. Jakobsson, and M. G. Christensen, “Multi-Pitch Estimation Exploiting Block Sparsity,” *Elsevier Signal Processing*, vol. 109, pp. 236–247, April 2015.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[31] S. P. Chepuri, *Sparse Sensing for Statistical Inference - Theory, Algorithms, and Applications*, Ph.D. thesis, Delft University of Technology, 2015.

[32] J. F. Sturm, “Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones,” *Optimization Methods and Software*, vol. 11-12, pp. 625–653, August 1999.

[33] R. H. Tutuncu, K. C. Toh, and M. J. Todd, “Solving semidefinite-quadratic-linear programs using SDPT3,” *Mathematical Programming Ser. B*, vol. 95, pp. 189–217, 2003.

[34] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, John Wiley and Sons, Inc., 1968.

[35] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, John Wiley and Sons, New York, 1971.

[36] Y. Rockah and P. M. Schultheiss, "Array Shape Calibration Using Sources in Unknown Locations – Part I: Far-Field Sources," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 3, pp. 286–299, March 1987.

[37] Y. Naim and H. Messer, "Notes on the Tightness of the Hybrid Cramér-Rao Lower Bound," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2074–2084, Jun. 2009.

[38] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Prentice Hall, Upper Saddle River, N.J., 2005.

Paper B

Paper B

Designing Optimal Frequency Offsets for Frequency Diverse Array MIMO Radar

Jie Cheng, Maria Juhlin, Andreas Jakobsson, Wen-Qin Wang

Abstract

Frequency diverse array (FDA) radars provide a potential solution to target localisation along the slant range and azimuth angle due to the range-angle-dependent transmit beampattern caused by the used frequency increments. However, the S-shaped beampattern resulting from the standard FDA leads to multiple candidate location estimates, introducing ambiguity in the target localization. To make full use of the degrees of freedom (DOF) allowed by the frequency increments, we here propose an optimal FDA multiple-input multiple-output (MIMO) frequency design scheme based on the Cramér-Rao lower bound (CRLB). The resulting system, here termed the optimal FDA-MIMO (OFDA-MIMO), is formed by optimizing the expected localization estimation accuracy, given the available prior knowledge of potential target locations. The used offsets are found as those minimizing the corresponding Bayesian CRLB (BCRLB), and may be iteratively refined as further information becomes available in a multi-pulse detection scenario. Both theoretical analysis and simulation results validate the preferable performance of the proposed system as compared to alternative frequency selection schemes.

Keywords: Frequency diverse array multiple-input multiple-output (FDA-MIMO), MIMO radar, frequency offset optimization, exploiting prior knowledge

1 Introduction

Frequency diverse array (FDA) radars have recently attracted notable attention in the literature [1–3]. This interest is partly due to that, in contrast to phased-array (PA) radars with angle-dependent transmit beampatterns and MIMO radars with conventional joint angular-Doppler processing, FDA can generate a range-angle-time-dependent response due to the existence of the different carrier frequencies used by adjacent antenna elements. In 2006, Antonik *et al.* introduced the concept of FDA [4] and showed the potential for multipath interference suppression and joint time-frequency control [5, 6]. Subsequently, the radiation pattern of FDA has been studied theoretically [7–9], using electromagnetic simulations [10, 11], as well as physical FDA antennas [12–14]. These results show that, in addition to the range dependence, the range-angle factors in the standard (uniform) FDA (SFDA) transmit beampattern are coupled with each other. This coupling leads to an S-shaped beampattern, which cannot be focused at a specific range-angle region, creating a range-angle ambiguity in the target localization. In effect, the decoupling of range and angle cause the resulting beampattern to differ for non-uniform arrays and for the case of using different carrier frequencies, as is done in an FDA.

For non-uniform arrays, Sammartino *et al.* introduced a non-uniform array element spacing arrangement for the transmit FDA of a bistatic system, creating a range-dependent transmit beampattern [15]. Combined with the angle-dependent beamforming of the receive phased array, the joint transmit-receive beampattern of the entire bistatic system is then decoupled in range and in angle, forming a point-like distribution. Sparsity has also been introduced in non-uniform FDA systems to achieve range-angle decoupling with the aim to obtain a well-focused point beampattern [16]. Nevertheless, it is often difficult to modify the arrangement of the antenna elements due to constraints in the array structure, whereas the choice of used carrier frequencies is often more flexible. Therefore, many FDA transmit beam-decoupling methods are based on nonlinear frequency increments. For example, Khan *et al.* [17] proposed an FDA with logarithmically increasing frequency carriers to decouple the range-angle transmitting beampattern, which produces a single maximum at the area of interest. Gao *et al.* [18] instead used a nonlinear frequency increment according to square and cubic functions, but the resulting beam side flap is relatively high. Additionally, non-uniform frequency offset functions, such as using a logarithmic frequency offset scheme [19], a sinusoidal function [20], a Hamming window-based [21], a tangent hyperbolic circular function [22], and a

Costas-sequence modulated code [23] have also been proposed for FDA systems. Further alternatives include an FDA range-angle-dependent decoupling transmit beamforming method based on random frequency offsets [24] and a discrete frequency increment scheme design based on subspace orthogonality [25]. However, for an FDA radar, these approaches still preserve the time-varying characteristics, and there is no optimal way to match the range-angle pairs in the transmitted signal, causing potential performance degradation. To overcome these drawbacks, there has been notable interest in FDA-MIMO radars, which can provide greater degrees of freedom (DOF) for range-angle control.

Employing an orthogonal frequency shift, the transmit aperture of an FDA-MIMO radar can be recovered at the receiver to jointly determine the range and angle parameters of the target [3]. In [26], the Cramér-Rao lower bound (CRLB) was derived for the resulting estimation problem, and both MUSIC [27] and maximum likelihood [28] estimators, using an unstructured model, were proposed to estimate the joint range, angle, and Doppler parameters. In [29], a unitary tensor MUSIC (UTMUSIC) method was developed for multiple-target localization using a frequency diverse subaperturing MIMO (FDS-MIMO) radar, whereas a one-dimensional MUSIC algorithm for the joint angle and range estimation was proposed in [30]. A joint angle and range estimator for low-elevation FDA-MIMO was also proposed in [31]. In all these works, linear frequency offsets are employed, limiting the resulting DOF, and thereby the achievable performance.

In this work, we formulate an optimal frequency selection scheme for an FDA-MIMO radar that minimizes the CRLB of the target localization problem given the available *a priori* knowledge of potential target locations. In order to do so, the OFDA-MIMO constructs a (possibly non-uniform) grid of potential carrier frequencies covering the available bandwidth. Then, for each transmit antenna, the scheme determines which of these candidate carriers that should be used for the transmission pulse, such that the collection of offsets used over the array maximize the expected localization performance. For a given (possible) target location, the resulting optimization determines the set of offset candidates that minimize the corresponding CRLB. More realistically, when one only has some knowledge of the regions of interest where a target may appear, such partial information may be included by instead minimizing either the worst-case CRLB (WCRLB) or the Bayesian CRLB (BCRLB), in a manner reminiscent of the microphone placement scheme developed in [32]. The main contributions of this work are

- We propose an optimal transmit frequency selection scheme for an FDA-MIMO radar, which selects the used carrier frequencies for each transmitter in order to optimize the location performance given the available prior knowledge.
- We introduce an efficient updating scheme of the used carriers that exploits the additional information obtained as a result of each pulse transmission.
- We illustrate and evaluate the proposed schemes for varying levels of available prior knowledge, illustrating the preferable performance obtained as compared to alternative frequency selection schemes.

The remainder of the paper is organized as follows: Section II formulates the OFDA-MIMO radar model. The different bounds used in the optimization and the proposed frequency selection scheme are derived in Section III. Section IV discusses the corresponding maximum likelihood (ML) and maximum *a posteriori* (MAP) estimates. Numerical results are presented in Section V, and, in Section VI, we conclude upon the work.

2 OFDA-MIMO Radar Design

For notational simplicity, we consider a linearly colocated FDA-MIMO radar with M transmit antennas and N receive antennas, using a range of carrier frequencies, as illustrated in Fig. 1. The inter-element spacing of the transmitters and receivers are, again for notational simplicity, assumed to be uniform, and is denoted by d. Each antenna element is connected to a narrowband transmitter, with the overall available bandwidth for all of the transmitters being denoted B_{max}. Initially, only considering a single pulse, we proceed to form a (potentially non-uniform) grid of P candidate carrier frequencies, which here, for notational convenience, is formed, such that the mth candidate carrier takes the form

$$f_m = f_0 + \Delta f_m = f_0 + \mathbf{w}_m^T \begin{bmatrix} \delta \\ 2\delta \\ \vdots \\ P\delta \end{bmatrix} = f_0 + \mathbf{w}_m^T \delta_P \tag{1}$$

for $m = 0, 1, \ldots, M-1$, where $(\cdot)^T$ denotes the transpose, $\mathbf{w}_m$ is a P-dimensional selection vector containing a single non-zero element, which is set to one, δ is the

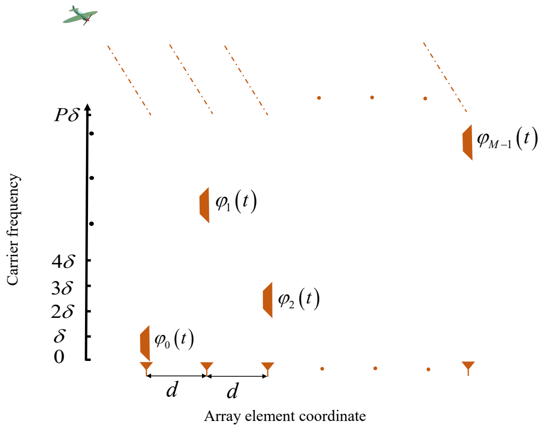


Figure 1: System sketch of an OFDA-MIMO radar. As shown, each transmitter use a frequency band selected among P candidate frequency bands. Each transmitter emits a waveform with envelope $\varphi_m(t)$, centered at the corresponding carrier frequency, f_m=f_0+Δf_m, for $m = 0, 1, \ldots, M-1$.

frequency increment between the P candidate offsets, and f_0 denotes the reference carrier frequency. For an FDA system, the narrowband signal emitted from the mth element can be expressed as

$$s_m(t) = \sqrt{\frac{E}{M}} \varphi_m(t)\, e^{j2\pi \mathbf{w}_m^T \boldsymbol{\delta}_P t}, 0 \leq t \leq T_P \tag{2}$$

where $\varphi_m(t)$ is the complex envelope, j denotes the imaginary unit, E is the transmitted energy, T_P is the radar pulse duration, and t is the (fast) time index within the radar pulse. The transmitted complex envelopes are selected so that they are (at least approximately) orthogonal to each other [3, 26], such that

$$\int_{T_P} \varphi_k(t) \varphi_m^*(t-\tau)\, dt \approx 0, k \neq m, \forall \tau \tag{3}$$

where τ and * denote the time delay and complex conjugate operator, respectively. Consider a far-field point target with range-angle pair (r, θ), where r is the slant range and θ is the azimuth angle, as measured from the target direction to the radar boresight. The target is assumed to be moving at constant velocity v towards the radar. Therefore, using the far-field approximation, the noise-free radar return at the nth receive antenna transmitted from the mth antenna, may be well detailed

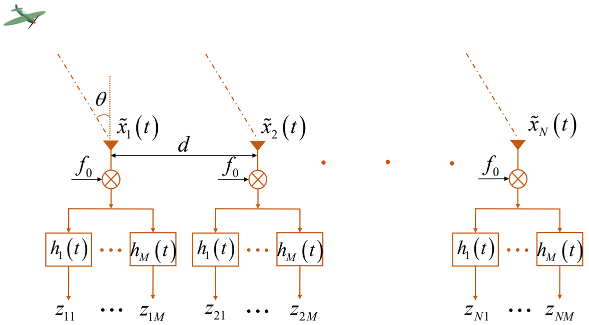


Figure 2: Receive structure of an OFDA-MIMO radar.

as [34]

$$\tilde{x}_{mn}\left(t,r,\theta\right) = \tilde{\xi}\varphi_m\left(t-\tau_{mn}\left(t,r,\theta\right)\right)e^{j2\pi f_m\left(t-\tau_{mn}(t,r,\theta)\right)} \tag{4}$$

where $\tilde{\xi}$ denotes the target's complex reflection coefficient accounting for the two-way path loss and the target reflexivity. Here, $\tau_{mn}\left(t,r,\theta\right)$ denotes the round-trip delay for the signal from the mth transmit element to the nth receive element, such that

$$\tau_{mn}\left(t,r,\theta\right) = \tau_0(r) + \tau_D\left(t,r,\theta\right) + \tau_{T,m}(\theta) + \tau_{R,n}(\theta) \tag{5}$$

where

$$\tau_0(r) = 2r/c \tag{6a}$$
$$\tau_D\left(t\right) = -2vt/c \tag{6b}$$
$$\tau_{T,m}(\theta) = -md\sin\theta/c \tag{6c}$$
$$\tau_{R,n}(\theta) = -nd\sin\theta/c \tag{6d}$$

with $\tau_0\left(r\right)$, $\tau_D\left(t\right)$, $\tau_{T,m}\left(\theta\right)$, and $\tau_{R,n}\left(\theta\right)$ denoting the propagation delays experienced by the slant range, target motion, inter-element spacings for the transmitting elements, and the receiving elements, respectively. Considering that $\delta \ll f_0$ and employing the narrowband assumption, i.e., $\varphi_m\left(t-\tau_{mn}\left(t,r,\theta\right)\right) \approx \varphi_m\left(t-\tau_0(r)\right)$, the approximative (noise-free) radar return, at receiver n, from transmitter $m = 0\ldots M-1$, may be obtained by inserting (5) into (4), yielding

$$\tilde{x}_n\left(t,r,\theta\right) \approx \tilde{\xi} e^{-j2\pi f_0 \tau_{R,n}(\theta)} e^{-j2\pi f_0 \tau_D(t)} \sum_{m=0}^{M-1} e^{-j2\pi f_0 \tau_{T,m}(\theta)} e^{j2\pi f_m (t-\tau_0(r))} \varphi_m\left(t-\tau_0(r)\right) \tag{7}$$

For notational brevity, we will in the following use the notation $\tilde{x}_n\left(t\right) = \tilde{x}_n\left(t,r,\theta\right)$, $\tau_0 = \tau_0(r)$, $\tau_D = \tau_D(t)$, $\tau_{T,m} = \tau_{T,m}(\theta)$, and $\tau_{R,n} = \tau_{R,n}(\theta)$. The Doppler shift within the pulse may be neglected assuming that the individual pulse durations satisfy $T_{PR} D_f \ll 1$, where $D_f = \frac{2v}{c} f_0 T_{PR}$, i.e., (see also [45])

$$\nu\left(t, D_f\right) = e^{-j2\pi f_0 \tau_D(t)} \approx e^{-j2\pi k D_f} = \nu\left(D_f\right) \tag{8}$$

where T_{PR} denotes the pulse repetition interval (PRI). In the initial processing of an FDA-MIMO radar, the received signal (7) is demodulated with $e^{-j2\pi f_0 t}$, yielding

$$\bar{x}_n\left(t\right) = \xi \nu\left(D_f\right) e^{j2\pi f_0 \tau_{R,n}} \sum_{m=0}^{M-1} \varphi_m\left(t-\tau_0\right) e^{j2\pi f_0 \tau_{T,m}} e^{j2\pi \mathbf{w}_m^T \delta_P (t-\tau_0)} e^{j2\pi f_0 \tau_D} \tag{9}$$

where $\xi = \tilde{\xi} e^{-j2\pi f_0 \tau_0}$. After this step, the received signal is processed through a bank of M matched filters, i.e., $h_l\left(t\right) = \varphi_l^*\left(t\right) e^{j2\pi \mathbf{w}_l^T \delta_P t}$, for $l = 0, \ldots, M-1$, such that the filter output of the lth filter may be expressed as

$$\begin{aligned}
z_{nl} &= \int_{-\infty}^{\infty} \xi \nu\left(D_f\right) e^{j2\pi f_0 \tau_{R,n}} \sum_{m=0}^{M-1} \varphi_m\left(t-\tau_0\right) e^{j2\pi f_0 \tau_{T,m}} e^{j2\pi \mathbf{w}_m^T \delta_P (t-\tau_0)} h_l(t-\tau') dt \\
&= \int_{-\infty}^{\infty} \xi \nu\left(D_f\right) e^{j2\pi f_0 \tau_{R,n}} \sum_{m=0}^{M-1} \varphi_m\left(t-\tau_0\right) e^{j2\pi f_0 \tau_{T,m}} e^{j2\pi \mathbf{w}_m^T \delta_P (t-\tau_0)} \varphi_l^*\left(t-\tau'\right) e^{-j2\pi \mathbf{w}_l^T \delta_P t} dt \\
&= \xi \nu\left(D_f\right) e^{j2\pi f_0 \tau_{R,n}} \sum_{m=0}^{M-1} e^{j2\pi f_0 \tau_{T,m}} e^{-j2\pi \mathbf{w}_m^T \delta_P \tau_0} \int_{-\infty}^{\infty} \varphi_m\left(t\right) \varphi_l^*\left(t-\Delta\tau\right) e^{-j2\pi\left(\mathbf{w}_m^T - \mathbf{w}_l^T\right)\delta_P t} dt \\
&= \xi \nu\left(D_f\right) e^{j2\pi f_0 \tau_{R,n}} e^{-j2\pi \mathbf{w}_m^T \delta_P \tau_0} \sum_{m=0}^{M-1} e^{j2\pi f_0 \tau_{T,m}} \chi_{m,l}\left(\Delta\tau, \left[\mathbf{w}_m^T - \mathbf{w}_l^T\right] \delta_P\right)
\end{aligned} \tag{10}$$

where

$$\chi_{m,l}\left(\Delta\tau, \left[\mathbf{w}_m^T - \mathbf{w}_l^T\right] \delta_P\right) = \int_{-\infty}^{\infty} \varphi_m\left(t\right) \varphi_l^*\left(t-\Delta\tau\right) e^{-j2\pi\left(\mathbf{w}_m^T - \mathbf{w}_l^T\right)\delta_P t} dt \tag{11}$$

with τ' denoting the time delay corresponding to the cell under test (CUT) and $\Delta\tau$ the relative delay corresponding to the first array element, i.e., $\Delta\tau = \tau' - \tau_0$. For an FDA-MIMO, (11) may be well approximated as [33, 34]

$$\begin{aligned}\chi_{m,l}\left(\Delta\tau, \left[\mathbf{w}_m^T - \mathbf{w}_l^T\right]\delta_P\right) \approx & \chi_{m,l}\left(0, \left[\mathbf{w}_m^T - \mathbf{w}_l^T\right]\delta_P\right) \\ & = \int_0^{T_P} \varphi_m(t)\varphi_l^*(t)\, e^{j2\pi\left(\mathbf{w}_m^T - \mathbf{w}_l^T\right)\delta_P t} dt \\ & \approx \int_0^{T_P} \varphi_m(t)\varphi_l^*(t)\, dt = R_{m,l}\end{aligned} \tag{12}$$

with the approximations holding for those pulses whose cross ambiguities are Doppler tolerant [33]. Here, $R_{m,l} = \int_0^{T_P} \varphi_m(t)\varphi_l^*(t)\, dt$ denotes the output of the lth filter matched to the mth transmitter waveform. In particular, because of the orthogonality assumption in (3), the synthesized output of the nth received signal with the lth matched filter for one pulse, including the corresponding additive noise, may be expressed as

$$\begin{aligned}y_{nl} &= z_{nl} + \omega_{nl} \\ &= \xi\nu\left(D_f\right) e^{j2\pi f_0 \tau_{R,n}} e^{j2\pi \mathbf{w}_l^T \delta_P \tau_0} e^{j2\pi f_0 \tau_{T,l}} + \omega_{nl}\end{aligned} \tag{13}$$

where $\omega_{nl} \sim \mathcal{CN}\left(0, \sigma_\omega^2\right)$ is assumed to be well modelled using a zero-mean circularly symmetric Gaussian distribution, and where we have used the assumption that $R_{l,l} \approx 1$, and $R_{m,l} \approx 0$ if $m \neq l$.

3 Frequency offset design via CRLB minimisation

To derive the frequency offset scheme that minimizes the estimation error for the range, angle, and Doppler shift estimates, we introduce a vector $\boldsymbol{\gamma} \in \mathbb{R}^{L\times 1}$ detailing the unknown parameters of the target, where L denotes the number of unknown parameters, initially considering these parameters to be given, such that

$$\boldsymbol{\gamma} = \left[\begin{array}{ccccc} \Re\{\xi\} & \Im\{\xi\} & r & \theta & D_f \end{array}\right]^T \tag{14}$$

where $\Re\{\cdot\}$ and $\Im\{\cdot\}$ indicates the real and imaginary parts, respectively. Assuming that a statistically efficient estimator of the unknown parameters, $\boldsymbol{\gamma}$, is feasible, we seek to determine the set of frequency offsets such that the corresponding CRLB is minimized, thus minimizing the achievable root mean squared error (RMSE) of the $\boldsymbol{\gamma}$ estimate. The CRLB is formed as the inverse of the Fisher Information Matrix (FIM). Let $y_{nl}\left(\boldsymbol{\gamma}\right)$ denote the received signal at receiver n and filtered through the matched filter l, as defined in (13), for the parameters $\boldsymbol{\gamma}$, and $\mathbf{F}\left(y_{nl}\left(\boldsymbol{\gamma}\right)\right)$ the corresponding FIM. In order to minimize the CRLB, one should determine which form of optimality that is sought; in this work, as is also commonly done in the literature [32, 42, 43], we employ an A-optimality formulation, minimizing the trace of the inverse FIM, yielding the minimization

$$\begin{aligned} \underset{\mathbf{w}_l}{\text{minimize}} \quad & \text{trace}\left(\left(\sum_{n=0}^{N-1}\sum_{l=0}^{M-1}\mathbf{F}\left(y_{nl}\left(\boldsymbol{\gamma}\right)\right)\right)^{-1}\right) \\ \text{subject to} \quad & 0 < ||\mathbf{w}_l||_0 \leq 1, \quad l = 0, \dots, M-1 \end{aligned} \tag{15}$$

where the $\left(i,j\right)$th element of $\mathbf{F}\left(y_{nl}\left(\boldsymbol{\gamma}\right)\right)$ is formed as

$$\left[\mathbf{F}\left(y_{nl}\left(\boldsymbol{\gamma}\right)\right)\right]_{i,j} = \mathbb{E}\left\{\nabla_{\gamma_i}\log\left(pdf\left(y_{nl}\left(\boldsymbol{\gamma}\right)\right)\right)\nabla_{\gamma_j}\log\left(pdf\left(y_{nl}\left(\boldsymbol{\gamma}\right)\right)\right)^*\right\} \tag{16}$$

where $\mathbb{E}\left\{\cdot\right\}$, ∇_{γ_i}, and $pdf\left(y_{nl}\left(\boldsymbol{\gamma}\right)\right)$ denote the statistical expectation, gradient with respect to γ_i, i.e., the ith element of $\boldsymbol{\gamma}$, and the probability density function (PDF) of $y_{nl}\left(\boldsymbol{\gamma}\right)$, respectively, and where the (pseudo)-norm constraint, $||\mathbf{w}_l||_0$, ensures that only one element in the selection vector $\mathbf{w}_l$ is chosen, for each l. This constraint causes the optimization problem to be a non-convex combinatorial problem, which implies that it is infeasible to solve in any practically relevant scenario. Fortunately, the problem may be approximated by relaxing the constraint by allowing the elements of $\mathbf{w}_l$ to instead take on values in the interval $[0,1]$, relaxing the (pseudo)-norm by instead using the 1-norm, enabling the problem to be approximated by a semidefinite program [35], [36]. The resulting relaxed solution, as detailed below, leads to an estimated selection vector for each transmitting antenna, with each element of the vector indicating the suitability of each candidate. By only retaining the maximum value in the vector, the selected candidate may then be determined.

In order to form the relaxed optimization problem, we initially note that the identity

$$\text{trace}\left(\mathbf{B}^{-1}\right) = \sum_{g=0}^{G-1}\mathbf{e}_g^T\mathbf{B}^{-1}\mathbf{e}_g \tag{17}$$

where $\mathbf{e}_g$ denotes the gth canonical basis vector, i.e., a vector that is 1 at the gth element and zero elsewhere, holds for every invertible $G \times G$ matrix [35]. Employing the Schur's complement

$$\mu - \mathbf{c}^T \mathbf{B}^{-1} \mathbf{c} \geq 0 \Leftrightarrow \begin{bmatrix} \mathbf{B} & \mathbf{c} \\ \mathbf{c}^T & \mu \end{bmatrix} \succeq 0 \tag{18}$$

where μ is scalar, $\mathbf{c}$ is an arbitrary vector, and $\mathbf{X} \succeq 0$ denotes that the matrix is positive semidefinite. Therefore, it follows that the minimization problem

$$\underset{\mathbf{B} \succ 0}{\text{minimize}} \qquad \mathbf{c}^T \mathbf{B}^{-1} \mathbf{c} \tag{19}$$

may be reformulated as the equivalent problem

$$\begin{aligned} &\underset{\mu, \mathbf{B} \succ 0}{\text{minimize}} && \mu \\ &\text{subject to} && \begin{bmatrix} \mathbf{B} & \mathbf{c} \\ \mathbf{c}^T & \mu \end{bmatrix} \succeq 0 \end{aligned} \tag{20}$$

Therefore, (15) can be reformulated as the relaxed optimization problem

$$\begin{aligned} &\underset{\{\mu_g\}, \{\mathbf{w}_l\}}{\text{minimize}} && \sum_{g=0}^{M-1} \mu_g \\ &\text{subject to} && \begin{bmatrix} \mathbf{F}\left(\mathbf{y}_M(\boldsymbol{\gamma})\right) & \mathbf{e}_g \\ \mathbf{e}_g^T & \mu_g \end{bmatrix} \succeq 0 \\ & && \mathbf{F}\left(\mathbf{y}_M(\boldsymbol{\gamma})\right) \succ 0 \\ & && 0 < ||\mathbf{w}_l||_1 \leq 1 \end{aligned} \tag{21}$$

where

$$\mathbf{F}\left(\mathbf{y}_M(\boldsymbol{\gamma})\right) = \sum_{n=0}^{N-1} \sum_{l=0}^{M-1} \mathbf{F}\left(y_{nl}(\boldsymbol{\gamma})\right) \tag{22}$$

The minimization in (21) may be solved as a semidefinite program with standard convex solvers, such as SeDuMi [40] or SDPT3 [37], although such a solution is computationally somewhat cumbersome if evaluable for multiple candidate target locations. The resulting optimization assumes perfect signal information in terms of the unknown parameters, $\boldsymbol{\gamma}$, and it is assumed that parameters have the same impact on the minimization, assumptions which are highly unrealistic. In order to address the case of varying importance of parameters in the minimization

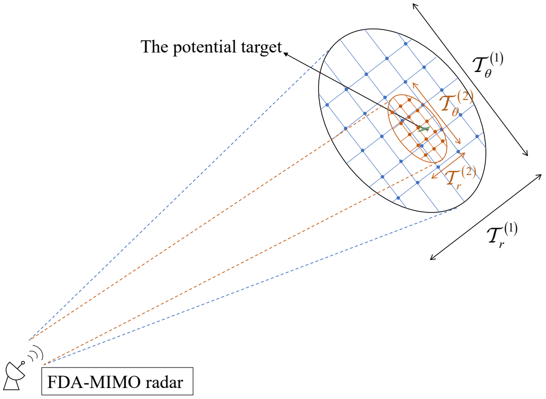


Figure 3: The discrete range and angle grid of the detection area for the worst-case formulation in (24). Here, the considered range and angle regions are formed over a grid $\mathcal{T}_r^{(i)} = \left\{r_1^{(i)}, \ldots, r_{T_r}^{(i)}\right\}$ and $\mathcal{T}_\theta^{(i)} = \left\{\theta_1^{(i)}, \ldots, \theta_{T_\theta}^{(i)}\right\}$, respectively. For each such location, a grid of T_D Doppler shifts is considered.

problem, a weighting matrix may be introduced to yield the weighted FIM [32, 35]. Introducing the weighting parameters $\left\{\psi_g\right\}$, (21) may be generalized as

$$\begin{aligned} \underset{\{\mu_g\},\{\mathbf{w}_l\}}{\text{minimize}} \quad & \sum_{g=0}^{M-1} \psi_g \mu_g \\ \text{subject to} \quad & \begin{bmatrix} \mathbf{F}\left(\mathbf{y}_M\left(\boldsymbol{\gamma}\right)\right) & e_g \\ e_g^T & \mu_g \end{bmatrix} \succeq 0 \\ & \mathbf{F}\left(\mathbf{y}_M\left(\boldsymbol{\gamma}\right)\right) \succ 0 \\ & 0 < ||\mathbf{w}_l||_1 \leq 1 \end{aligned} \tag{23}$$

Comparing (21) and (23), it may be noted that the weight parameter, ψ_g, allows the user to put different emphasis on different components in the vector $\boldsymbol{\gamma}$. If $\psi_g = 0$ for some values of g, this implies that the parameter $\boldsymbol{\gamma}_g$ will be disregarded in the minimization.

Typically, the precise location of a potential target is not known *a priori*, although regions of interest may well be. To allow for this, we proceed to modify the design scheme such that this form of information may be taken into account. In order to do so, we examine two different approaches to allow for a wider range of target locations. Initially, we examine an extension of (23), wherein we form a worst-case optimization over a grid of potential target locations and Doppler shifts. Let $\mathcal{T}_r^{(1)} = \left\{r_1^{(1)}, \ldots, r_{T_r}^{(1)}\right\}$, $\mathcal{T}_\theta^{(1)} = \left\{\theta_1^{(1)}, \ldots, \theta_{T_\theta}^{(1)}\right\}$, and $\mathcal{T}_D^{(1)} = \left\{D_1^{(1)}, \ldots, D_{f_{T_D}}^{(1)}\right\}$ denote the sets of considered range and angle locations, as well as Doppler shifts of a potential target for the initial pulse, as illustrated in Fig. 3, where T_r, T_θ, and T_D denote the number of range, angle, and Doppler candidates, respectively. Thus, in order to determine the worst-case optimal solution for the initial pulse, the optimization in (23) is generalized to

$$\begin{aligned}
&\underset{\{\mu_g\},\{\mathbf{w}_l\}}{\text{minimize}} && \sum_{g=0}^{M-1} \psi_g \mu_g \\
&\text{subject to} && \begin{bmatrix} \mathbf{F}\left(\mathbf{y}_M\left(\boldsymbol{\gamma}\right)\right) & \mathbf{e}_g \\ \mathbf{e}_g^T & \mu_g \end{bmatrix} \succeq 0 \\
& && \mathbf{F}\left(\mathbf{y}_M\left(\boldsymbol{\gamma}\right)\right) \succ 0 \\
& && 0 < ||\mathbf{w}_l||_1 \leq 1 \\
& && r \in \mathcal{T}_r^{(1)},\ \theta \in \mathcal{T}_\theta^{(1)},\ D_f \in \mathcal{T}_{D_f}^{(1)}
\end{aligned} \tag{24}$$

We term the resulting worst-case optimal frequency selection scheme OFDA-MIMO-W. For a single pulse, the computational complexity of forming (24) is mainly affected by the size of the used parameter grids. Forming the FIM over all P possible combinations of carriers, for the M transmitters, over the three parameter grids, thus requires $O(P^M T_r T_\theta T_D L^3)$ operations, where the L^3 results from the cost of forming the inverse FIM. Fortunately, the optimization in (24) may be formed in a greedy manner, fixing all but one of the transmitter grids and only optimizing for the sole remaining transmitter. By then doing this for all transmitters, the overall complexity may be reduced to $O(MPT_r T_\theta T_D L^3)$.

In a second formulation, we instead consider that the target parameters follow some known distribution. In this way, the range and angle of a potential target may be considered stochastic, allowing for a solution that incorporates this uncertainty. To incorporate the location uncertainty, we therefore instead optimize the localization performance using the Bayesian FIM (BFIM) [37]. In this case, the BFIM, $\mathbf{F}_B\left(\mathbf{y}_M\left(\boldsymbol{\gamma}\right)\right)$, will consist of two parts [38], such that

$$\mathbf{F}_B\left(\mathbf{y}_M\left(\boldsymbol{\gamma}\right)\right) = \mathbf{F}_D\left(\mathbf{y}_M\left(\boldsymbol{\gamma}\right)\right) + \mathbf{F}_P\left(\mathbf{y}_M\left(\boldsymbol{\gamma}\right)\right) \tag{25}$$

where $\mathbf{F}_D\left(\mathbf{y}_M\left(\boldsymbol{\gamma}\right)\right)$ denotes the contribution from the data, as defined in (16), whereas $\mathbf{F}_P\left(\mathbf{y}_M\left(\boldsymbol{\gamma}\right)\right)$ details the available knowledge of the stochastic prior [37,38], with the (i, j)th element of $\mathbf{F}_P(\mathbf{y}_M(\boldsymbol{\gamma}))$ being

$$\left[\mathbf{F}_P(\mathbf{y}_M(\boldsymbol{\gamma}))\right]_{i,j} = -\mathbb{E}\left[\frac{\partial^2 pdf(\boldsymbol{\gamma})}{\partial \gamma_i \partial \gamma_j}\right] \tag{26}$$

where $pdf(\boldsymbol{\gamma})$ denotes the prior distribution of the L-dimensional parameters $\boldsymbol{\gamma}$. In order to form $\mathbf{F}_B(\mathbf{y}_M(\boldsymbol{\gamma}))$, the expected value of the Jacobian in $\mathbf{F}_D(\mathbf{y}_M(\boldsymbol{\gamma}))$ has to be estimated. As further discussed in the numerical section, this can be done using Monte Carlo simulations given the assumed prior. This allows the frequency design problem to be reformulated as the one instead minimizing the Bayesian CRLB (BCRLB), such that

$$\begin{aligned} \underset{\{\mu_g\},\{\mathbf{w}_l\}}{\text{minimize}} \quad & \sum_{g=0}^{M-1} \psi_g \mu_g \\ \text{subject to} \quad & \begin{bmatrix} \mathbf{F}_B(\mathbf{y}_M(\boldsymbol{\gamma})) & \mathbf{e}_g \\ \mathbf{e}_g^T & \mu_g \end{bmatrix} \succeq 0 \\ & \mathbf{F}_B(\mathbf{y}_M(\boldsymbol{\gamma})) \succ 0 \\ & 0 < ||\mathbf{w}_l||_1 \leq 1 \end{aligned} \tag{27}$$

Different from (24), (27) only need to be solved once, without the use of grid over r, θ, and D_f, resulting in a computational cost of $O(P^M L^3)$, which, by the use of a greedy implementation as noted above, reduces to $O(PML^3)$. However, in order to form the BFIM one has to form $\mathbf{F}_D(y_M(\boldsymbol{\gamma}))$, which typically has to be done using Monte Carlo simulations, increasing the complexity by $O(C^L)$, where C denotes the number of Monte Carlo samples.

4 Estimation

We proceed to examine the problem of estimating the location of a target using the used frequency offset selection. Consider a target at location $\Phi = [r, \theta]$, moving with velocity v. The filtered signal samples for the kth pulse, $y_{nl}^{(k)}$, as defined in (13), may be stacked in matrix form as

$$\mathbf{Y}_k = \begin{bmatrix} \mathbf{y}_0^{(k)} & \dots & \mathbf{y}_{M-1}^{(k)} \end{bmatrix} \in \mathbb{C}^{N \times M} \tag{28}$$

$$\mathbf{y}_\ell^{(k)} = \begin{bmatrix} y_{0\ell}^{(k)} & \dots & y_{(N-1)\ell}^{(k)} \end{bmatrix}^T \tag{29}$$

where $\mathbf{y}_{\ell}^{(k)}$ denotes the ℓth filtered signals for all receivers, for pulse k. This may succinctly be expressed as

$$\mathbf{Y}_k = \xi \nu_k \left(D_f\right) \mathbf{A}_{TR|k}\left(r, \theta\right) + \boldsymbol{\Omega}_k \tag{30}$$

where

$$\mathbf{A}_{TR|k}\left(r, \theta\right) = \mathbf{a}_R\left(\theta\right) \mathbf{a}_{T|k}^T\left(r, \theta\right) \tag{31}$$

$$\mathbf{a}_{T|k}\left(r, \theta\right) = \begin{bmatrix} e^{-j2\pi\left(\mathbf{w}_{0|k}^T \boldsymbol{\delta}_P \tau_0 + f_0 \tau_{T,0}\right)} \\ e^{-j2\pi\left(\mathbf{w}_{1|k}^T \boldsymbol{\delta}_P \tau_0 + f_0 \tau_{T,1}\right)} \\ \vdots \\ e^{-j2\pi\left(\mathbf{w}_{M-1|k}^T \boldsymbol{\delta}_P \tau_0 + f_0 \tau_{T,M-1}\right)} \end{bmatrix} \tag{32}$$

$$\mathbf{a}_R\left(\theta\right) = \begin{bmatrix} e^{-j2\pi f_0 \tau_{R,0}} & \dots & e^{-j2\pi f_0 \tau_{R,N-1}} \end{bmatrix}^T \tag{33}$$

$$\boldsymbol{\Omega}_k = \begin{bmatrix} \boldsymbol{\omega}_0^{(k)} & \dots & \boldsymbol{\omega}_{M-1}^{(k)} \end{bmatrix} \tag{34}$$

$$\boldsymbol{\omega}_{\ell}^{(k)} = \begin{bmatrix} \omega_{0\ell}^{(k)} & \dots & \omega_{(N-1)\ell}^{(k)} \end{bmatrix}^T \tag{35}$$

with $\nu_k(D_f)$ representing the Doppler shift of the kth pulse, as defined in (8), and $\omega_{nl}^{(k)} \sim \mathcal{CN}\left(0, \sigma_{\omega}^2\right)$. The ML estimate of the unknown parameters using K pulses, each possibly having different sets of carrier frequencies, may then be formed as (see also [34, 41])

$$\left\{\hat{\boldsymbol{\Phi}}, \hat{D}_f\right\} = \arg\max_{\boldsymbol{\Phi}, D_f} \left| \sum_{k=0}^{K-1} \mathbf{a}_R^H\left(\theta\right) Y_k \mathbf{a}_{T|k}^*\left(r, \theta\right) \right|^2 \tag{36}$$

In the case when considering Bayesian priors, one may instead form the MAP estimate as [38, 44]

$$\left\{\hat{\boldsymbol{\Phi}}, \hat{D}_f\right\} = \arg\max_{\boldsymbol{\Phi}, D_f} \operatorname{pdf}\left(\boldsymbol{\Gamma} \left| \boldsymbol{\Phi}, D_f\right.\right) \operatorname{pdf}\left(\boldsymbol{\Phi}\right) \operatorname{pdf}\left(D_f\right) \tag{37}$$

When forming the estimates, one initially has only a coarse prior knowledge of potential target locations, or perhaps no such knowledge is available at all. In the latter case, as is also considered in the following, we employ a standard linear frequency offset scheme, i.e., using the standard FDA-MIMO (SFDA-MIMO) [26], for the initial pulse. The accuracy of the resulting location and velocity estimates may be reasonably well approximated by the

Table 1: Simulation parameters of the studied FDA-MIMO radar

Parameter	Value
Transmit elements	6
Receive elements	4
Reference carrier frequency, f_0	10 GHz
Wavelength, λ	0.03 m

corresponding CRLB, allowing this accuracy to be employed as prior knowledge when computing the suitable frequency offsets for the following pulse, thereby allowing for an iteratively refined localization estimate. In this way, the optimal frequency offset schemes are found for the consecutive pulses by minimizing the optimization problem associated with the worst-case CRLB or the BCRLB, respectively. The resulting frequency offsets are then employed for the following pulse, and the location and velocity estimates updated using the ML or MAP estimates, respectively, for the new offset scheme. The resulting search grids for the second pulse, $\mathcal{T}_r^{(2)}$, $\mathcal{T}_\theta^{(2)}$, and $\mathcal{T}_D^{(2)}$, are then, as illustrated in Fig. 3, formed over a smaller uncertainty region, reflecting the improved prior information available for the second pulse. This in turn allows the CRLB estimates to be improved, further improving the offset scheme used for the following pulse, and so on, until the desired resolution has been achieved. As the velocity estimates are typically quite poor until the uncertainty in r and θ has been reduced, it is often preferable to restrict the size of the grid of D_f to a single or a few elements for the initial pulses, and first include a finer grid for later pulses.

5 Numerical Evaluation

We proceed to evaluate the performance of the proposed frequency selection scheme using simulated data with parameters as listed in Table I. When doing so, we design the transmission carrier frequencies using different approaches and illustrate both the resulting bounds and the RMSE achieved when employing the ML or MAP estimators, for the corresponding deterministic and Bayesian settings, respectively. The ML or MAP estimators are in these cases given data simulated using the corresponding carrier design scheme, illustrating the achievable performance for each design scheme. In the simulations, we examine the following FDA-MIMO radar configurations: (i) the SFDA-MIMO radar

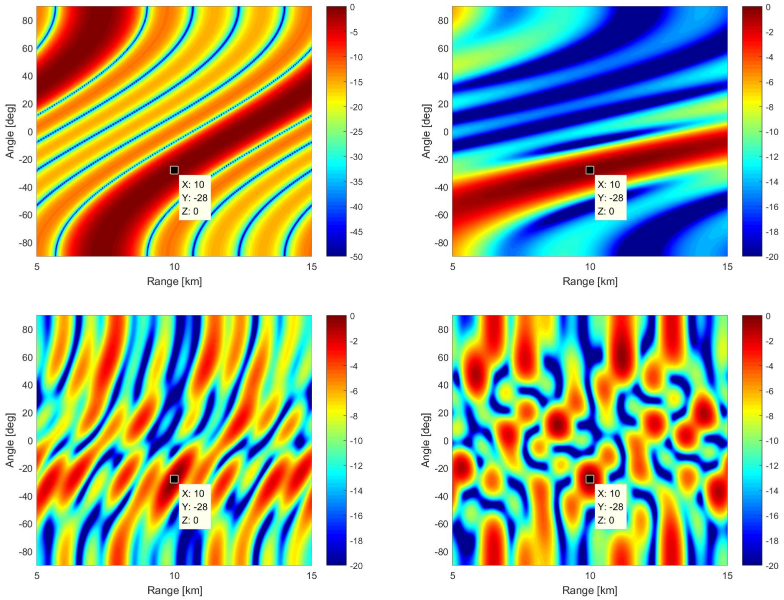


Figure 4: The resulting FDA-MIMO radar transmit beampatterns for the discussed frequency selection schemes.

with linear increasing frequency offset, using $\Delta f_m^{SFDA-MIMO} = m \cdot B_{\max}/M$ as frequency offset for the mth carrier; (ii) the logarithmic FDA-MIMO (LFDA-MIMO) [19]: FDA-MIMO radar using logarithmically increasing frequency offset, where the frequency offset for the mth element is $\Delta f_m^{LFDA-MIMO} = \log(m+1) \cdot B_{\max}/\log(M)$; (iii) the proposed worst-case OFDA-MIMO selection scheme (OFDA-MIMO-W), i.e., the FDA-MIMO radar using $\Delta f_m^{OFDA-MIMO}$ minimizing (24) for a given target location; (iv) the proposed Bayesian OFDA-MIMO selected scheme (OFDA-MIMO-B): the FDA-MIMO radar minimizing the BCRLB in (27). For the OFDA-MIMO- B, the BFIM is computed using $C = 10^3$ Monte Carlo simulations.

5.1 Example 1: FDA-MIMO radar transmit beampatterns

Initially, we illustrate the beampatterns of the different FDA-MIMO radar configurations for a single target located at $(10 \text{ km}, -28^\circ)$, with a velocity $v = 324$ km/h, yielding a Doppler shift $D_f = 0.4$ Hz. In the figures, we have

Table 2: Used frequency offset schemes (in kHz) of the discussed FDA-MIMO radar for the example in Fig. 5

	Δf_1	Δf_2	Δf_3	Δf_4	Δf_5	Δf_6
SFDA-MIMO	0	100	200	300	400	500
LFDA-MIMO	0	193.43	306.57	386.85	449.12	500
OFDA-MIMO-W	0	100	400	500	400	0
OFDA-MIMO-B	200	100	500	0	0	100

used $P = 50$ carrier candidates, for a grid spacing $\delta = 100$ kHz, such that $B_{max} = 5$ MHz. Fig. 4(a) shows the beampattern of the SFDA-MIMO radar, illustrating how the energy of its beampattern in space has the noted S-shape, being scattered over a large region and with high sidelobe ridges. Fig. 4(b) shows the corresponding beampattern of the LFDA-MIMO radar, illustrating how the energy of the beampattern is even more scattered than for the SFDA-MIMO, although having lower energy in the sidelobe. Fig. 4(c) shows the beampattern of the proposed OFDA-MIMO-W radar, assuming accurate information of the target location. As can be seen in the figure, the resulting beampattern yields a well focused pattern centered at the target location. Allowing for an uncertainty in the target location, Fig. 4(d) shows the beampattern of the proposed OFDA-MIMO-B radar, which is seen to also offer a well focused beampattern, even though the radar now only assumes the target to be located as detailed by the priors. Here, the parameter vector $\boldsymbol{\gamma}$ is modelled with $\xi = 1$, $r \in \mathcal{N}(10000, 200)$, $\theta \in \mathcal{N}(-28, 0.5)$, and $D_f \in \mathcal{N}(0.4, 0.05)$.

5.2 Example 2: localisation performance analysis for target

We proceed to illustrate how the used frequency offset scheme affects the performance of the location estimate considering a target located at $r_0 = 10220$ m, $\theta_0 = -28^\circ$, with $D_f = 0.4$ Hz and $\xi = 1$. For the OFDA-MIMO-W, we consider an initial gird of candidate locations $\mathcal{T}_r^{(1)} = \{1, 1.02, 1.04, \ldots, 15\}$ km, $\mathcal{T}_\theta^{(1)} = \left\{-40^\circ, -37^\circ, -34^\circ, \ldots, 11^\circ\right\}$, and $\mathcal{T}_D^{(1)} = \{0.25, 0.35, 0.45\}$. For the OFDA-MIMO-B, we assume that the range, angle, and Doppler shift have Gaussian distributions, $r_s \in \mathcal{N}\left(r_0, \sigma_r\right)$, $\theta_s \in \mathcal{N}(\theta_0, \sigma_\theta)$, and $D_f \in \mathcal{N}\left(D_f, \sigma_{D_f}\right)$. To allow the different forms of priors to be roughly the same, we set $\sigma_r = 15$ m, $\sigma_\theta = 1.5^\circ$, and $\sigma_{D_f} = 0.05$. The signal-to-noise ratio (SNR) in decibel is here defined as $SNR = -10\log_{10}\frac{|\xi|^2}{\sigma_\omega^2}$, where σ_ω^2 is the noise power at the receiver (for notational simplicity, we here assume that all receivers experience the same SNR).

In this example, the initial pulse of the proposed scheme is formed using SFDA-MIMO, after which the resulting location estimates are used to estimate the corresponding CRLB, which is then in turn used to improve the frequency selection scheme, as discussed above. As the prior knowledge of the location and velocity of a target will initially be poor, we proceed to illustrate the proposed updating procedure taking this into account. In the first pulse, striving to initially detect the presence of a target, we are primarily interested in determining the range to a potential reflector. For this reason, we use a weighting of the parameters in $\boldsymbol{\gamma}$ such that the frequency offset scheme will emphasise only the range parameter, setting $\psi_3 = 1$, while letting all other weights, $\psi_l = 0$, for $l = 1, \ldots, L$, $l \neq 3$. This will cause the offset design scheme to prioritize minimizing the CRLB corresponding to the range estimate. The corresponding uncertainties are set correspondingly to reflect the lack of knowledge for the other parameters. For the second pulse, the measured signal offers some information of the range to the target, thereby allowing us to reduce the size of the corresponding uncertainty region, using the estimated range to determine the CRLB and update the used prior. To begin to also consider the angle information, we now include the weighting for θ, setting $\psi_4 = 10$, while retaining $\psi_3 = 1$ (note the significant difference in scale for the range and angle estimates, indicating that we are still giving the range parameter a weighting roughly 100 times higher than the angle estimate.) After the second pulse, we have further information of the range and angle estimates, allowing us to increase the weighting of the θ parameter, setting $\psi_4 = 10^3$, although the location information is still deemed to be insufficient to accurately estimate the velocity, updating the frequency selection scheme accordingly. In the fourth pulse, we start including a focus also on the velocity estimate; letting $\psi_3 = 1$, $\psi_4 = 10^4$, and $\psi_5 = 100$, now giving equal weighting to the range and angle estimates, but a lower weighting to the velocity estimate. Finally, in the fifth pulse, and onwards, we allow an equal weighting of the sought parameters (reflecting their difference in size), setting $\psi_3 = 1$, $\psi_4 = 10^4$, and $\psi_5 = 10^5$. Figs. 5-9 show the corresponding performance for the first 5 pulses, as a function of SNR, illustrating how the performance of the corresponding estimates improve given

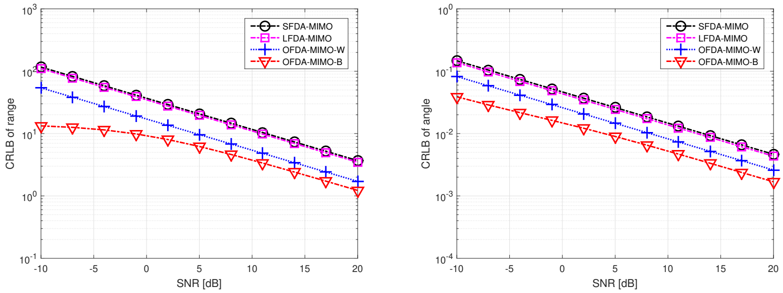


Figure 5: The expected achievable performance for the initial pulse as measured by the corresponding CRLB when employing the discussed frequency selection schemes using $\psi_3 = 1$ and $\psi_l = 0$, for $l = 1, \dots, 5$, $l \neq 3$, as a function of SNR, for the (a) range and (b) angle estimates.

the increasing information. As can be seen in the figures, the optimal schemes allow for notably improved location estimates as compared to the standard and logarithmic frequency selection schemes, for all configurations. It may be noted that the performance difference for these initial pulses are rather similar. To better illustrate how the performance improves over the pulses, Fig. 10 illustrates the performance as a function of the pulse number, for SNR $= -10\ dB$. It may be noted that OFDA- MIMO-B yields the best performance in both angle, range, and Doppler, whereas the SFDA-MIMO radar offers the worst. Fig. 11 compares the RMSE as a function of the number of used pulses, K, of FDA-MIMO radars for range, angle and Doppler shift, respectively. The results are here averaged over 1000 Monte Carlo (MC) trials at $SNR = -10$ dB. It is observed that, in the case of larger K, the range, angle and Doppler estimators of both FDA-MIMO radars have MSEs close to the corresponding CRLBs.

Fig. 12 illustrates the expected performance as a function of the used bandwidth B_{max}, for $SNR = -10$ dB and $\delta = 100$ kHz. As may be seen in the figure, the proposed frequency selection schemes improve on the efficiency of the used bandwidth, enabling accurate estimates using a substantially lower overall bandwidth. It should be stressed that, for the OFDA-MIMO-B, the bandwidth does not notably affect the performance as the exploited information is the assumed prior information. When accurate prior information of target is given, the bandwidth does not have a significant effect on the estimation of the target. Interestingly, it can also be seen in Fig. 12, that the accuracy of the angle

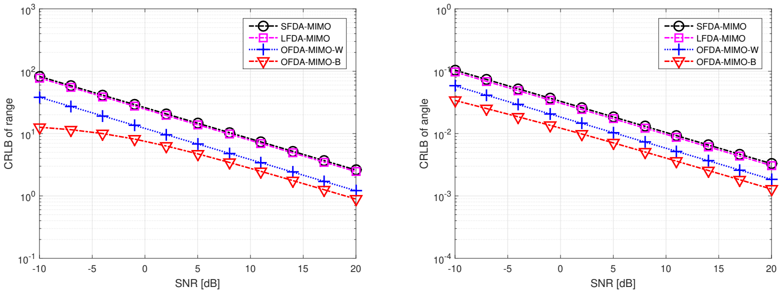


Figure 6: The expected achievable performance for the second pulse as measured by the corresponding CRLB when employing the discussed frequency selection schemes for $\psi_3 = 1$, $\psi_4 = 10$, and $\psi_l = 0$, $l = 1, \ldots, 5, \ \neq 3, 4$, as a function of SNR, for the (a) range and (b) angle estimates.

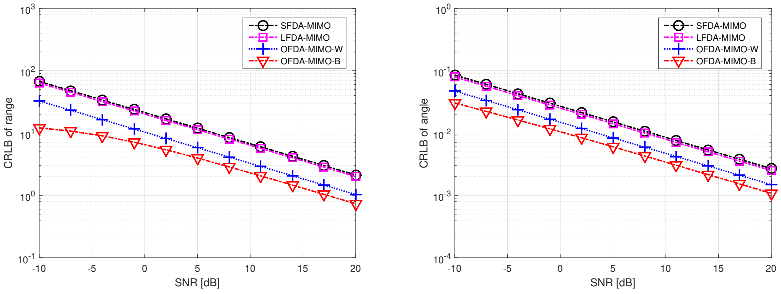


Figure 7: The expected achievable performance as measured by the corresponding CRLB when employing of the discussed frequency selection schemes for $\psi_3 = 1$, $\psi_4 = 10^3$, and $\psi_l = 0$, for $l = 1, 2, 5$, as a function of SNR, for the (a) range, (b) angle.

estimate are almost not affected by the changes of B_{max}, a result which is consistent with [26].

6 Conclusion

In this paper, we formulate two frequency selection design schemes based on minimizing the worst-case and the Bayesian CRLB, incorporating the available prior knowledge of potential target locations in the design scheme. The proposed scheme form an A-optimal minimization of the corresponding CRLB

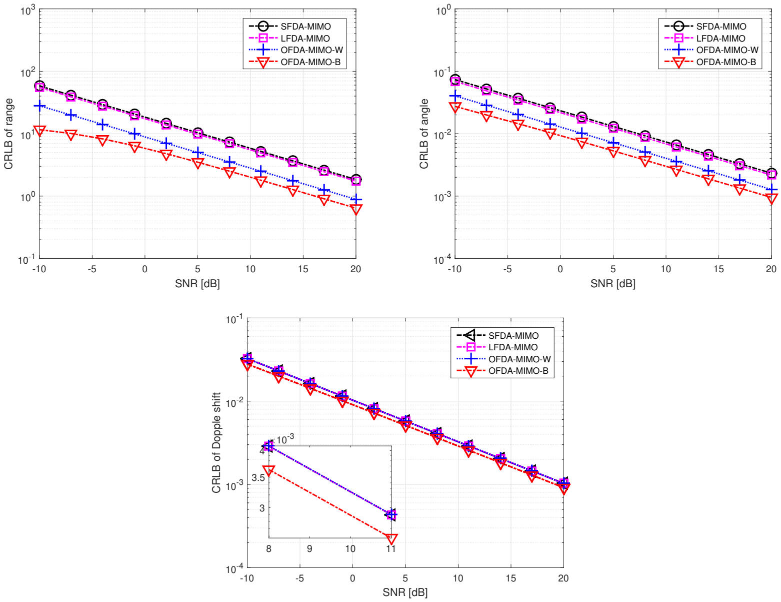


Figure 8: The expected achievable performance for the fourth pulse as measured by the corresponding CRLB when employing the discussed frequency selection schemes using $\psi_3 = 1$, $\psi_4 = 10^4$, $\psi_5 = 100$, and $\psi_l = 0$, for $l = 1, 2$, as a function of SNR, for the (a) range, (b) angle, and (c) Doppler shift estimates.

over a range of candidate frequency offsets, thereby obtaining the set of carriers most suitable for locating a target given the prior information. Comparative simulation results show that the proposed selection schemes significantly improve the radar performance as compared with existing frequency selection approaches, in both focusing the transmit energy to the desired target position, as well as for the resulting estimates.

References

[1] Y. Liao, H. Tang, W. Q. Wang and M. Xing, ”A Low Sidelobe Deceptive Jamming Suppression Beamforming Method With a Frequency Diverse Array,” in IEEE Transactions on Antennas and Propagation, vol. 70, no. 6, pp. 4884-4889, June 2022.

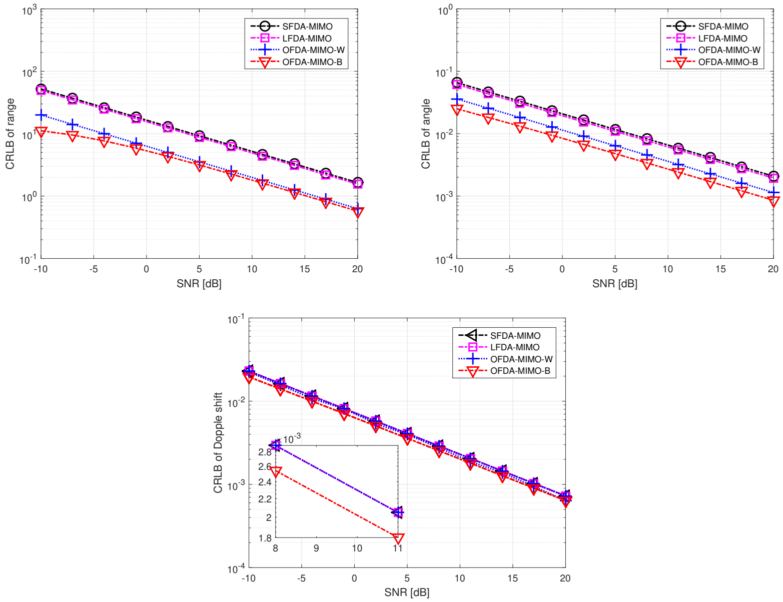


Figure 9: The expected achievable performance for the fifth pulse, and onwards, as measured by the corresponding CRLB when employing the discussed frequency selection schemes using $\psi_3 = 1$, $\psi_4 = 10^4$, $\psi_5 = 10^5$, and $\psi_l = 0$, for $l = 1, 2$, as a function of SNR, for the (a) range, (b) angle, and (c) Doppler shift estimates.

[2] W. Wang, ”Frequency Diverse Array Antenna: New Opportunities,” in IEEE Antennas and Propagation Magazine, vol. 57, no. 2, pp. 145-152, April 2015.

[3] J. Xu, G. Liao, S. Zhu, L. Huang and H. C. So, ”Joint Range and Angle Estimation Using MIMO Radar With Frequency Diverse Array,” in IEEE Transactions on Signal Processing, vol. 63, no. 13, pp. 3396-3410, July, 2015.

[4] P. Antonik, M. C. Wicks, H. D. Griffiths and C. J. Baker., ”Frequency diverse array radars”, IEEE Radar Conference, Verona, 2006, 215-217

[5] P. Antonik, M. C. Wicks, H. D. Griffiths and C. J. Baker, ”Multi-mission multi-mode waveform diversity” IEEE Radar Conference, Verona, 2006, 3-7

[6] P. Antonik. An investigation of a frequency diverse array. London: University College London, 2009, 1-20.

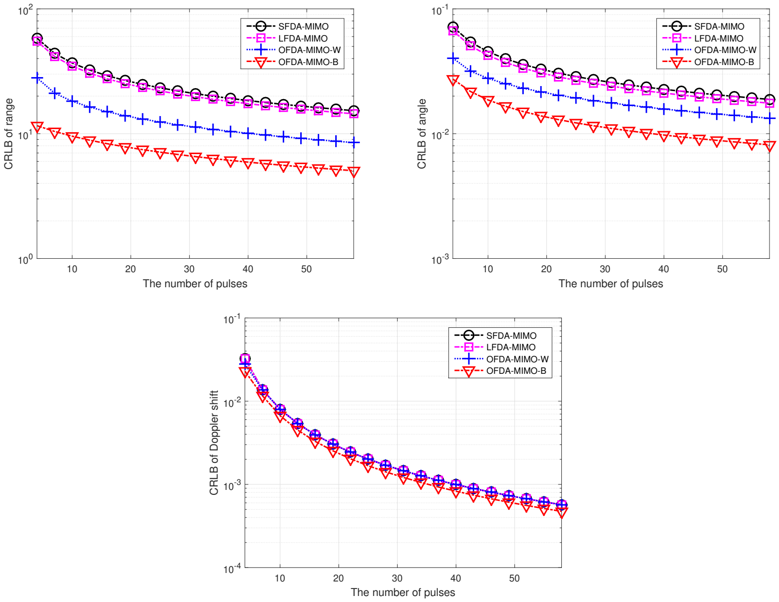


Figure 10: The expected achievable performance as measured by the corresponding CRLB when employing the discussed frequency selection schemes, as a function of the number of pulses, for the (a) range, (b) angle, and (c) Doppler shift estimates.

[7] T. Higgins and S. D. Blunt, ”Analysis of range-angle coupled beamforming with frequency-diverse chirps,” 2009 International Waveform Diversity and Design Conference, 2009, pp. 140-144.

[8] Y. Wang, W. Wang and H. Chen, ”Linear Frequency Diverse Array Manifold Geometry and Ambiguity Analysis,” in IEEE Sensors Journal, vol. 15, no. 2, pp. 984-993, Feb. 2015.

[9] T. Liao, Y. Pan and W. Q. Wang, ”Generalized Linear Frequency Diverse Array Manifold Curve Analysis,” in IEEE Signal Processing Letters, vol. 25, no. 6, pp. 768-772, June 2018.

[10] M. Secmen, S. Demir, A. Hizal and T. Eker, ”Frequency diverse array antenna with periodic time modulated pattern in range and angle”, IEEE Radar Conference, Waltham, 2007, 427-430

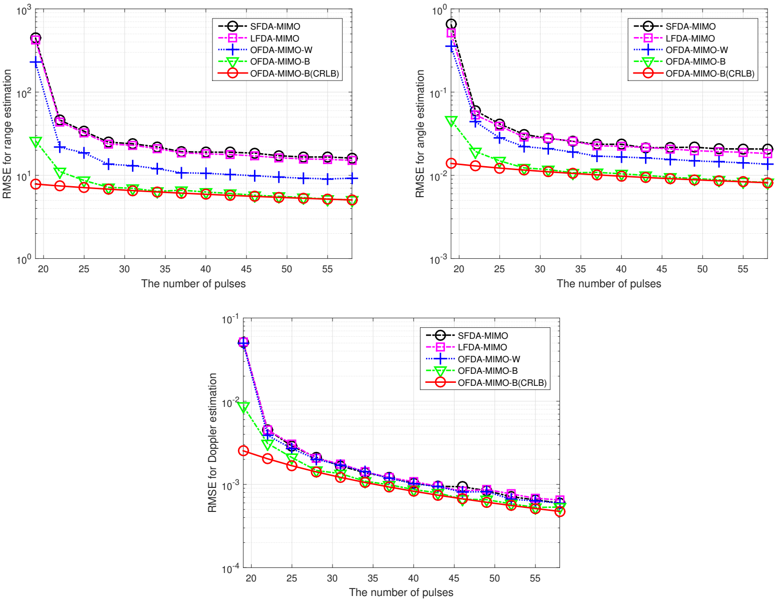


Figure 11: The expected achievable performance as measured by the corresponding RMSE when employing the discussed frequency selection schemes, as a function of the number of pulses, for the (a) range, (b) angle, and (c) Doppler shift estimates.

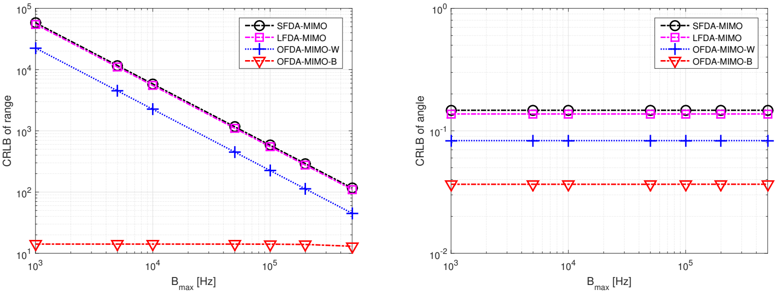


Figure 12: The expected achievable performance as measured by the corresponding CRLB for the discussed frequency selection schemes, as a function of B_{max}, for the (a) range and (b) angle estimates.

[11] J. Shin, J. H. Choi, J. Kim, J. Yang, W. Lee, J. So and C. Cheon, "Full-wave simulation of frequency diverse array antenna using the FDTD method," 2013 Asia-Pacific Microwave Conference Proceedings (APMC), 2013, pp. 1070-1072.

[12] J. Huang, K-F. Tong and C .J. Baker, "Frequency diverse array with beam scanning feature," 2008 IEEE Antennas and Propagation Society International Symposium, 2008, pp. 1-4.

[13] J. Huang, K. Tong, K. Woodbridge and C. Baker, "Frequency diverse array: Simulation and design", IEEE Radar Conference, Pasadena, 2009, 1-4.

[14] K. Wan, W. -Q. Wang, H. Chen and S. Zhang, "Space-Time Modulated Wideband Array Antenna", in IEEE Antennas and Wireless Propagation Letters, vol. 18, no. 6, pp. 1081-1085, June 2019.

[15] P. F. Sammartino, C. J. Baker and H. D. Griffiths, "Frequency Diverse MIMO Techniques for Radar", in IEEE Transactions on Aerospace and Electronic Systems, vol. 49, no. 1, pp. 201-222, Jan. 2013, doi: 10.1109/TAES.2013.6404099.

[16] H. Chen, H. Shao and W. Wang, "Joint Sparsity-Based Range-Angle-Dependent Beampattern Synthesis for Frequency Diverse Array", in IEEE Access, vol. 5, pp. 15152-15161, 2017.

[17] W. Khan, I. M. Qureshi, A. Basit and W. Khan, "Range-Bins-Based MIMO Frequency Diverse Array Radar With Logarithmic Frequency Offset", in IEEE Antennas and Wireless Propagation Letters, vol. 15, pp. 885-888, 2016.

[18] K. Gao, W. -Q. Wang, J. Cai and J. Xiong, "Decoupled Frequency Diverse Array Range- Angle- Dependent Beampattern Synthesis Using Non-linearly Increasing Frequency Offsets", in IET Microwaves, Antennas and Propagation, vol. 10, pp. 880-884, 2016.

[19] W. Khan, I. M. Qureshi and S. Saeed, "Frequency Diverse Array Radar With Logarithmically Increasing Frequency Offset", in IEEE Antennas and Wireless Propagation Letters, vol. 14, pp. 499-502, 2015.

[20] Y. Wang, G. Huang and W. Li, "Transmit Beampattern Design in Range and Angle Domains for MIMO Frequency Diverse Array Radar", in IEEE Antennas and Wireless Propagation Letters, vol. 16, pp. 1003-1006, 2017.

[21] A Basit, I. M. Qureshi, W. Khan, S. U. Rehman and M. M. Khan, ”Beam Pattern Synthesis for an FDA Radar with Hamming Window-Based Nonuniform Frequency Offset”, in IEEE Antennas and Wireless Propagation Letters, vol. 16, pp. 2283-2286, 2017.

[22] S. Saeed, I. M. Qureshi, W. Khan, A. Salman, ”Tangent Hyperbolic Circular Frequency Diverse Array Radars”, in the Journal of Engineering, 2016, 2016(3): 23-28.

[23] Z Wang, W. Q. Wang, H. Shao, ”Range-azimuth Decouple Beamforming for Frequency Diverse Array with Costas-sequence Modulated Frequency Offsets”, in EURASIP Journal on Advances in Signal Processing, 2016(1), 1-9.

[24] Y. Liu, H. Ruan, L. Wang and A. Nehorai, ”The Random Frequency Diverse Array: A New Antenna Structure for Uncoupled Direction-Range Indication in Active Sensing”, in IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 2, pp. 295-308, March 2017.

[25] J. Li, S. Ouyang, K. Liao, and X. Sun, ”Identifying unambiguous frequency patterns for two-target localization using frequency diverse array”, in IEEE Signal Processing, Vol. 171, pp. 107452, 2020.

[26] J. Xiong, W. Wang and K. Gao, ”FDA-MIMO Radar Range-☒☒Angle Estimation: CRLB, MSE, and Resolution Analysis”, in IEEE Transactions on Aerospace and Electronic Systems, vol. 54, no. 1, pp. 284-294, Feb. 2018.

[27] X. Chen, B. Chen, J. Guan, Y. Huang and Y. He, ”Space-Range-Doppler Focus-Based Low-observable Moving Target Detection Using Frequency Diverse Array MIMO Radar”, in IEEE Access, vol. 6, pp. 43892-43904, 2018.

[28] J. Xu, W. Wang, C. Cui and R. Gui, ”Joint Range, Angle and Doppler Estimation for FDA-MIMO Radar”, 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM), 2018, pp. 499-503.

[29] X. Li, D. Wang, W. -Q. Wang, W. Liu and X. Ma, ”Range-Angle Localization of Targets With Planar Frequency Diverse Subaperturing MIMO Radar”, in IEEE Access, vol. 6, pp. 12505-12517, 2018.

[30] M. Feng, Z. Cui, Y. Yang and Q. Shu, ”A Reduced-Dimension MUSIC Algorithm for Monostatic FDA-MIMO Radar”, in IEEE Communications Letters, vol. 25, no. 4, pp. 1279-1282, April 2021.

[31] G. Zheng and Y. Song, ”Signal Model and Method for Joint Angle and Range Estimation of Low-Elevation Target in Meter-Wave FDA-MIMO Radar”, in IEEE Communications Letters, vol. 26, no. 2, pp. 449-453, Feb. 2022.

[32] M. Juhlin and A. Jakobsson, ”Optimal Sensor Placement for Localizing Tonal Sound Sources”, in Elsevier Signal Processing, vol. 202, pp. 108679, 2023.

[33] L. Lan, A. Marino, A. Aubry, A. De Maio, G. Liao, J. Xu and Y. Zhang, ”GLRT-Based Adaptive Target Detection in FDA-MIMO Radar”, in IEEE Transactions on Aerospace and Electronic Systems, vol. 57, no. 1, pp. 597-613, Feb. 2021.

[34] R. Gui, W. Wang, C. Cui and H. C. So, ”Coherent Pulsed-FDA Radar Receiver Design With Time-Variance Consideration: SINR and CRB Analysis”, in IEEE Transactions on Signal Processing, vol. 66, pp. 200-214, 2018.

[35] J. Swärd, F. Elvander, A. Jakobsson, ”Designing Sampling Schemes for Multi-dimensional Data”, in IEEE Signal Processing, Vol. 150, pp. 1-10, 2018.

[36] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[37] H. L. Van Trees, Detection, Estimation, and Modulation Theory, Part I, John Wiley and Sons, Inc., 1968.

[38] H. L. Van Trees, Detection, Estimation, and Modulation Theory, Part IV, Optimum Array Processing, John Wiley and Sons, Inc., 2002.

[39] P. Stoica, and R. L. Moses, Spectral analysis of signals (Vol. 452, pp. 25-26). Upper Saddle River, NJ: Pearson Prentice Hall, 2005.

[40] J. F. Sturm, Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones, Optimization Methods and Software, 11:1-4, 625-653.

[41] J. Li and P. Stoica, MIMO Radar Signal Processing. New York, USA: John Wiley & Sons, Inc., 2009.

[42] F. Elvander, J. Swärd, and A. Jakobsson. ”An efficient solver for designing optimal sampling schemes.” arXiv preprint arXiv:2111.05579 (2021).

[43] S. P. Chepuri, Sparse Sensing for Statistical Inference: Theory, algorithms, and applications, Delft University of Technology, 2015 Ph.D.Thesis.

[44] H. L. Van Trees, K. L. Bell, Bayesian bounds for parameter estimation and nonlinear filtering/tracking. Wiley-Interscience, John Wiley & Sons and IEEE Press, 2007.

[45] R. Gui, W. -Q. Wang, and Z Zheng, ”Low-complexity GLRT for FDA radar without training data”, Digital Signal Processing 107 (2020): 102861.

Paper C

Paper C

Efficient Tracking of Inhomogeneous Jammers in a Wireless Network

Maria Juhlin and Andreas Jakobsson

Abstract

In this paper, we propose a novel approach for locating and tracking an unknown number of jammers in a wireless network. The area and shape covered by each jammer is assumed to be partly unknown, and may vary over time as the jammers move through the network. By assuming that the jammed region for each jammer may be described by one of a set of parametric methods, the appropriate shape as well as the number of jammers may be determined, even when the jammed regions overlap substantially. Employing a Viterbi-based tracking approach, we allow for an improved localization of moving jammers, including when new jammers emerge or leave the network. Numerical simulations illustrate the performance of the introduced framework as compared to recent alternative approaches.

Keywords: Localization and tracking of jammers, wireless networks, model order estimation, detecting jamming patterns

1 Introduction

Many everyday phenomena may be considered as different forms of networks, with examples ranging from railway networks and shipping routes, to wireless LANs and sensor networks. In most cases, the communication in the networks are safe-guarded against adversarial attacks by means of predefined communication protocols, such as e.g. MACs, and other security measures. Apart from eavesdropping, denial-of-service (DoS) is a common method used to interfere with the communication in a network. There are many ways to instigate a DoS attack, e.g., with the DoS attacker flooding the network with data forcing parts of it to shut down due to overload. Another way to interfere with a network is to use physical jamming devices, making communication in the jammed area impossible.

When a successful jamming attack does occur, some form of action must be taken to fend off the attack in order to allow the communication to recommence. To counter jamming attacks, it is generally critical to accurately locate the jammer's location, whereafter it may be neutralized in a suitable way. Thus, an important problem is to detect and accurately estimate the location of any jammers in the network. Given the prevalence of wireless sensor networks (WSNs), jammer localization in WSNs is a problem currently attracting notable interest. In the literature, there are two main approaches to jammer localization, range-based methods and range-free methods [1]. Range-based methods utilize the fact that the signal strength of the jamming signal decreases with distance, whereas range-free methods do not make this assumption. When range measurements are available, range-based methods are often superior. Regrettably, in most cases, range estimates are not available or reliable, generally making range-free techniques the methods of choice. One of the first methods presented for jammer localization is found in [2], where a single jammer with a circular jamming region is located on a grid of sensor nodes. The setup of sensors and jammers on a grid as done in [2] is a somewhat idealized setup, and a more general approach was considered in [3, 4], where the jammer location was found as the weighted average of the jammed nodes. The method in [3] may be used both with range measurements and without. An alternative method is found in [1], where physical constraints have been applied to find the jammer location. The localization algorithm is iterative and at each time point the jammer location is updated so as to incorporate all the jammed nodes while still maximizing the distance to the unaffected ones. Yet another method was presented in [5] whereby the convex hull of the jammed nodes was formed, with the (single) jammer being assumed to be located at the center of the circle enclosing this hull. The common feature for all the noted methods is that they all treat the somewhat ideal setup of a single stationary jammer with a circular jamming region. A more realistic scenario considering multiple possibly overlapping jammers was presented in [6], where two methods were presented for jammer localization, the first method exploiting fuzzy C-means, and the other making use of a skeleton algorithm from image analysis for finding bifurcation points. Despite this extension, the jammers were still assumed to be circular and stationary, moreover the jammers were only allowed a maximum overlap of 50%, making the problem rather idealized for real world applications. Addressing these concerns have partially been done by studying jammer tracking in e.g. [7, 8], where the first method allows for the tracking of one circular jammer, and the other two methods use different variants of the Kalman filter for tracking multiple jammers, albeit all assuming circular jamming regions. Apart from only allowing for a circular jamming region, another drawback with the proposed methods is that they assume that the number of jammers is known *a priori*, an often unrealistic assumption, but one that is required by the Kalman filter. A potential workaround to this problem could be to estimate the number of jammers present at several prescribed time instances and then update the Kalman algorithm accordingly, although such an approach will quickly become computationally wasteful, and might suffer significant performance degradations.

To address the issues outlined above, this work focuses on the more realistic setup of moving jammers with potentially overlapping non-circular jamming regions, with an unknown number of jammers being present in the network. By allowing for non-stationary jammers, one may extract information about their locations and jamming patterns that would not have been possible from a single snapshot. As the jammer moves through the network, information regarding the jammer's jamming pattern can be accumulated, thus potentially allowing for a better jammer location estimate, as opposed to the case when only a single snapshot is used. To address this, we examine the problem when one observes a series of snapshots of a jamming scenario, wherein the network suffers from the effects of an unknown number of non-stationary, possibly overlapping, jammers with unknown locations and jamming patterns, with the aim of developing a method for tracking each such jammer as time progresses. Because the jamming regions are allowed to overlap, the problem resembles that of the cell tracking problem studied in [9], from which we draw inspiration. There are two main methods for tackling this form of tracking problems, namely tracking-by-model evaluation or tracking-by-detection [9]. In this work, we have opted to form our estimate using tracking-by-detection, treating the jammer detection step in terms of an image segmentation problem. The image analysis literature contains a multitude of methods to segment overlapping objects (see e.g. [10]). What sets the jammer problem apart from previously studied image segmentation problems is that the contour of the jammed region is unknown and has to be inferred from the surroundings. This problem is associated with a high degree of uncertainty, as illustrated in Figure 1, and [11]. The three curves in Figure 1 depict the maximal and minimal jamming regions given the observed nodes, along with the actual jamming region. The dark blue circle is the result when estimating the jamming region as the maximally inscribed circle of the unjammed nodes, and the red circle is the result when using the minimally enclosing circle of the jammed nodes. As may be seen, the true jamming region is here in between these two extremes. In this work, in the segmentation step, we seek not only to estimate the number of jammers, and their respective position, but also to identify their respective jamming patterns. We present a greedy algorithm for iteratively estimating the number of jammers and their locations. The jamming patterns are deemed to be unknown, but limited to a set of possible patterns found in a predefined bank of parametric pattern models, or to a combination of these patterns. Due to the complexity of the problem, we also study approximation methods to allow for a faster convergence of the estimator, reducing the overall computational complexity.

Analogous to [9], we propose to use the Viterbi algorithm [12] for the trajectory linking. The Viterbi algorithm is a probabilistic algorithm used in a variety of applications (see, e.g., [13–16]). The choice of the Viterbi algorithm is motivated by the algorithm's ability to handle a changing number of states, thus allowing jammers and nodes to leave and enter the studied area. This feature is necessary in the case of overlapping jamming regions, where it may be difficult or impossible to discern the correct number of jammers present from one snapshot. Another advantage is that the Viterbi algorithm's probabilistic nature enables path determination of the most likely path anchored in the problem setting by using the *a priori* information included in the form of the chosen distributions. The Viterbi algorithm is suitable for batch implementations, thus allowing one to find the overall most likely path within each batch, but lends itself poorly to on-line implementation, although good approximative methods are available [17].

In summary, the contributions in this work are the following:

- We present three new methods, depending on the assumptions made, for determining the locations of an unknown number of jammers, allowing for partly unknown jamming patterns. The presented segmentation methods allow for several forms of jamming regions, as well as for combinations of different jamming patterns.
- We track the jammer's movement over time by linking sequential jammer segmentations using the Viterbi algorithm. The use of the Viterbi algorithm allows for both complete and partial overlap of the jamming regions, and the introduction/loss of jammers over time.

The remainder of the paper is structured as follows: in the next section, we formalize the jammer localization problem, splitting the overall procedure into a series of subproblems aiming to determine the number of jammers and the shape of their jammed regions. Then, in section 3, we examine how jammers may be tracked over time as they move through the network. In section 4, we evaluate the performance of the proposed method as compared to alternative approaches. Finally, section 5 contains our conclusion.

2 Problem Formulation

Consider a network containing N sensors and K jammers. The positions of the sensors are here assumed to be known, whereas both the number of jammers, K, and their positions are unknown. The sensors are considered to be in either of two possible states, being jammed or not, here represented as 1 or 0, respectively. Thus, the state of being jammed is considered to be a binary state. Each present jammer is considered to have a jamming region that may be represented by a predefined pattern, collected as a parametric template in a bank of such patterns. The jamming regions are assumed to be isotropic, implying that all nodes within the jammer's range are jammed, as opposed to a degree of jamming that decreases with the distance from the jammer, reflecting the fact that nodes typically are either able to communicate or not, but that they typically do not provide information about the radio environment at the sensor. Furthermore, the jammers may have overlapping jamming regions. The unknowns in this problem setup are thus the number of jammers and their positions, the form of their respective jamming regions, as well as their motions over time. An example of the setup is shown in Figure 2, showing a network with $K = 3$ jammers

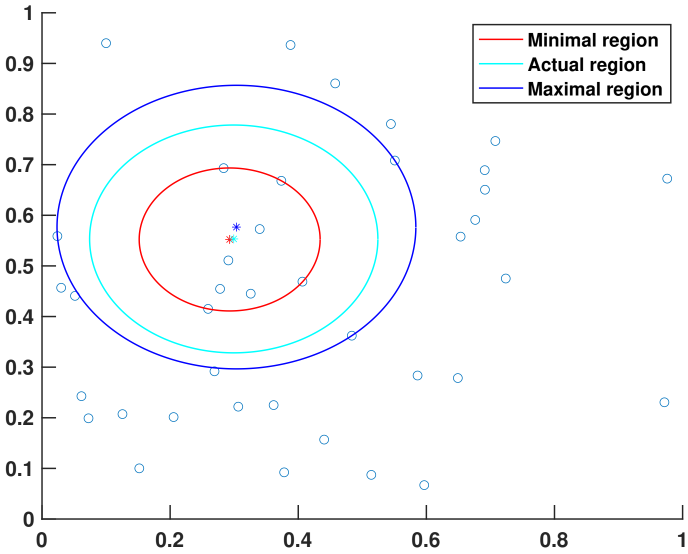


Figure 1: The resulting jamming region using a maximally inscribed circle (dark blue), and minimally enclosing circle (red), along with the true region (cyan). The network dimensions are for simplicity normalized.

with partly overlapping jamming regions. The problem of locating and tracking multiple jammers is non-trivial, and one that typically has to be broken down into several subproblems. In this work, we have opted for a tracking-by-detection approach, which means that the overall algorithm may be divided into two subproblems that are solved in sequence. This procedure is then updated for each time point. The first of these subproblems is that of jammer localization, for which the aim is to find the position of each of the jammers, whereas the second problem is that of jammer tracking, where the goal is to find the path each of the jammers have moved between two consecutive time points. An overview of the overall procedure is presented in Algorithm 1, for a batch of T timepoints.

To formalise the problem further, consider a set of N sensor nodes $\mathcal{S} = \{\mathcal{U}, \mathcal{V}\}$, composed of J jammed nodes, $\mathcal{U} = (u_1, \ldots, u_J)$, and $N - J$ unaffected nodes, $\mathcal{V} = (v_1, \ldots, v_{N-J})$, with $\mathcal{U}, \mathcal{V} \in \mathbb{R}^d$, where d denotes the dimensionality of the scenario, typically 2 or 3. The coordinates of the j:th jammed node are represented by the tuple $u_j = [u_{j,1}, u_{j,2}, \ldots, u_{j,d}]^T$, with v_j being defined similarly. At each

Algorithm 1 Overview

```
*Input:* $K = 1$
**for** $t \leftarrow 1$ to $T$ **do** *$T$ is the batch size*
    Solve the segmentation problem using algorithm 2
    noSolutionFound = True
    **while** noSolutionFound **do**
        *Estimate the number of jammers, $K$*
        **for** $k \leftarrow 1$ to $K$ **do**
            Using either of algorithms 3-5, for each jammer $k$, find its
            jamming region and estimate the jammer location, $\ell_k(t)$
        **end for**
        **if** Estimated jamming regions cover all jammed and no
           unjammed nodes **then**
            noSolutionFound = False
        **else**
            $K = K + 1$
        **end if**
    **end while**
    *Check if pattern merging should be done*
    *Track the $K$ jammers using the Viterbi algorithm*
    **for** $k \leftarrow 1$ to $K$ **do**
        Update the trajectory of jammer $k$,
        using algorithm 6
    **end for**
**end for**
```

point in time, each of the K jammers will give rise to a minimum volume jammed region $P_k(t)$, for $k = 1, \ldots, K$, such that the total jammed area may be modeled as

$$P(t) = \bigcup_{k=1}^{K} P_k(t) \tag{1}$$

For notational simplicity, we will for now omit the time index, later reintroducing it in the tracking steps. The aim of the estimation problem is to, for each point in time, find the number of jammers, K, and deduce which of the jammed nodes that are jammed by each of the jammers. Thus, we simultaneously wish to estimate K, partition the jammed nodes into K jamming regions, and find the

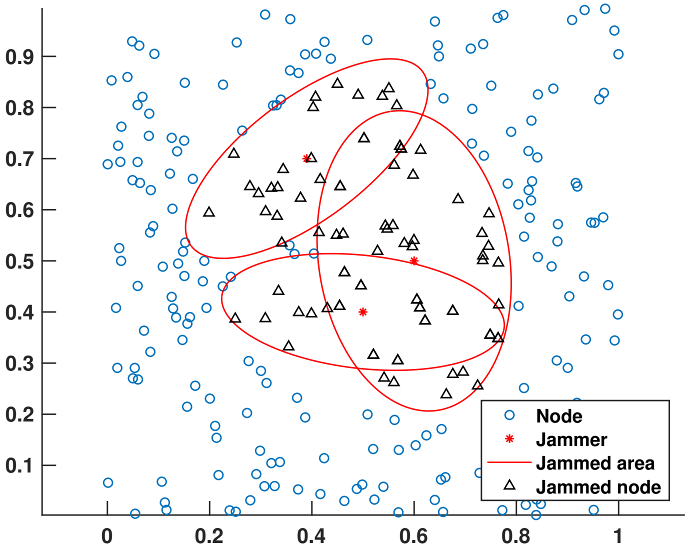


Figure 2: An example setup with unjammed nodes, jammed nodes, jammers, and jamming regions marked. The dimensions of the network are normalized.

parameters describing each jamming region, θ_k, and its respective center point, $c_k = (c_{k,1}, c_{k,2}, \dots, c_{k,d})$.

For each $k = 1, 2, \dots, K$, we therefore seek to find the jammed nodes $U_k = (u_1, u_2, \dots, u_q)$, $q \leq J$, constituting the jammed region P_k, and the transformation matrix $A_k(\theta_k)$ detailing the jamming region, and its parameters θ_k, such that each P_k may be described as

$$P_k = \left\{u \in U_k : \left\|A_k^{-1}(\theta_k)(u - c_k)\right\| \leq 1\right\} \tag{2}$$

for every u in U_k. The outline of the jamming region, $A_k(\theta_k)$, is thus parameterized by θ_k, with some examples being a circle, an ellipse, and an oval, all of which may be specified as

$$A_k(\theta_k) = \begin{bmatrix} \theta_{k,1} & 0 \\ 0 & \theta_{k,2} \end{bmatrix} \tag{3}$$

where $\theta_{k,\ell} = t(u)/a_\ell$, with $t(\cdot)$ denoting a function of the coordinates, and a_ℓ is the length of the ℓ:th semi axis. Thus, using $\theta = [1, 1]$, $\theta = [1/a_1, 1/a_2]$, and

$\theta = [\frac{1}{a_1}, \frac{e^{0.2u_{j,1}}}{a_2}]$, where $u_{j,1}$ is the first coordinate of the j:th jammed node, results in a circular, an elliptical, and an oval jamming pattern, respectively [18].

Using this form of pattern, one may also form more complicated patterns, such as a flower pattern, common in beamforming, by forming a collection of ellipses overlapping at one end. Rewriting (2) as

$$\begin{aligned}1 &\geq \left\| A_k^{-1}(\theta_k)(u - c_k) \right\|^2 \\ &= \left(A_k^{-1}(\theta_k)(u - c_k)\right)^{\mathrm{T}} \left(A_k^{-1}(\theta_k)(u - c_k)\right) \\ &= (u - c_k)^{\mathrm{T}} \left(A_k^{-1}(\theta_k)\right)^{\mathrm{T}} A_k^{-1}(\theta_k)(u - c_k) \\ &= (u - c_k)^{\mathrm{T}} Q_k^{-1}(\theta_k)(u - c_k), \end{aligned} \tag{4}$$

indicates that one may specify the closed curve describing the jamming region P_k using the symmetric positive definite matrix

$$Q_k(\theta_k) = A_k^T(\theta_k) A_k(\theta_k) \tag{5}$$

The volume covered by this curve is found as the determinant of $Q_k(\theta_k)$. The optimization problem to be solved may therefore be reformulated as that of finding the areas of minimum volume covering the jammed nodes, i.e.,

$$\min_K \bigcup_{k=1}^{K} P_k \tag{6}$$

To solve this problem, one has to determine the number of jammers present, K, and for each jammer fit a jamming pattern P_k of minimum volume, such that the union of all jamming patterns covers the jammed nodes $\mathcal{U}$, and none of the unaffected nodes $\mathcal{V}$. Using (5) as a parameterization of P_k, the resulting optimization problem in (6) may be expressed as

$$\begin{aligned} \min_{K, \{U_k\}, \{Q_k\}, \{\theta_k\}, \{c_k\}} \quad & \bigcup_{k=1}^{K} \det Q_k(\theta_k) \\ \text{s.t.} \quad & \left(u - c_k\right)^{\mathrm{T}} Q_k^{-1}(\theta_k) \left(u - c_k\right) \leq 1, \forall u \in U_k, \forall k \end{aligned} \tag{7}$$

The formulation thus strives to minimize the total jamming volume, with respect to the number of jammers, the different possible node sets U_k, and the parameterization of the individual jamming regions. The optimization in (7) is a non-convex problem, but allows for a solution if the problem is approximated as a series of smaller convex subproblems. To do so, we initially consider the situation for a given (and thus known) number of jammers, K. The solution process can thus be separated into the following subproblems, $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$, detailed as follows:

- Subproblem $\mathcal{P}_1$: The first obstacle consists of finding a partitioning of the set of jammed nodes $\mathcal{U}$ into K (potentially overlapping) sets.
- Subproblem $\mathcal{P}_2$: Given a partitioning, obtained as the solution to $\mathcal{P}_1$, the next problem is to find the minimum volume jamming region (MVJR) enclosing the nodes in the partitioning.
- Subproblem $\mathcal{P}_3$: As a final step, one needs to determine if the union of the proposed MVJRs covers any unaffected nodes. If that is the case, the solution process has to be reiterated, finding a new partitioning and/or finding a new MVJR (thereby changing the estimated pattern).

The solution process iterates between subproblems $\mathcal{P}_1$-$\mathcal{P}_3$. In case no solution can be found that does not cover any unjammed nodes, while still covering all the jammed nodes, the method deems it necessary to increase the number of assumed jammers by 1, and the process begins anew. An overview of the segmentation is presented in Algorithm 2.

Solving subproblem $\mathcal{P}_1$

The first problem that needs to be solved is that of finding a partitioning of the jammed nodes into possibly overlapping clusters, and to find the cluster center of each such cluster. This cluster center will then be used as an estimate of the MVJR center in the next section. There are various methods for finding a partitioning of nodes (see e.g. [19]). In this work, we present three different methods used to find a good initial partitioning depending on the assumptions that can be made, namely (i) the k-means with soft thresholding [20–22], (ii) using the Expectation Maximization algorithm (EM-algorithm) [23, 24] for fitting a Gaussian Mixture Model (GMM), and (iii) classification using Gibbs sampling from an underlying GMM [23,24] . These three methods are tailored for different assumptions on the jamming regions. Notably, the k-means method imposes no assumptions on the clustering, whereas the EM and Gibbs methods both make the assumption that

Algorithm 2 Segmentation

```
noSolutionFound = True
while noSolutionFound do
    Partition the jammed nodes into $K$ sets $\mathcal{U} = \{U_1, \ldots, U_K\}$,
    using either of algorithms 3-5
    for each $U_k$ in $\mathcal{U}$ do
        Find the minimum jamming region, $P_k$, by solving (18)
    end for
    if $\bigcup P_k$ does not cover any unaffected nodes then
        noSolutionFound = False
    else
        $K \leftarrow K + 1$
    end if
end while
```

the regions may be well modeled using ellipsoids. In the k-means algorithm, we cluster the data into K sets, for a given value of K, such that the average squared distance between the jammed nodes and the cluster center is minimized. The use of a soft threshold ensures that the clusters do not have to be disjoint, but are allowed to overlap, in accordance with the possibility of overlapping jamming regions. To each jammed node u, we assign a weight, $w_k(u)$, that signifies its degree of belonging to cluster k. For the regular k-means algorithm, this weight is binary and thus limited to $w_k(u) \in \{0, 1\}$, but for the soft threshold version, this is relaxed to allow for weights in the interval between 0 and 1, i.e., $w_k(u) \in [0, 1]$. The algorithm finds the weight matrix $W \in \mathbb{R}^{J \times K}$, that describes the degree of belonging of all nodes and all clusters, $w_{j,k} = w_k(u_j)$, as well as the cluster centers for the K clusters, $\mu = [\mu_1, \ldots, \mu_K]$. It is worth noting that the cluster centers, μ_k, may be different from the center of the MVJRs, c_k. Formally, the minimization forming the partitioning is

$$\min_{\mu} \sum_{j=1}^{J} \sum_{k=1}^{K} w_{j,k}^m ||u_j - \mu_k||^2 \tag{8}$$

where the weights are defined as

$$w_{j,k}^m = \frac{1}{\sum_{\ell=1}^{K} \left(\frac{||u_j - \mu_k||}{||u_j - \mu_\ell||} \right)^{\frac{2}{m-1}}} \tag{9}$$

Algorithm 3 k-means clustering

```
Initialize: $w \leftarrow w_0$
repeat
    for $k \leftarrow 1$ to $K$ do
        Compute the cluster centers $\mu_k = \sum_{j=1}^{J} w_{j,k}^m u_j$
    end for
    for $k \leftarrow 1$ to $K$ do
        Update the weights $w_{j,k}$ using (9)
    end for
until The cost function in (8) is below a given threshold, $T_w$, or the maximum
    number of allowed iterations is reached
```

with $m > 1$ being a user-defined parameter that governs the allowed degree of overlap. For $m = 1$, the algorithm degenerates and the k-means segmentation without any overlap is recovered, in which case the weights are instead proportional to the number of class members. It should be noted that the partitioning given by the k-means algorithm is only a local solution to the partitioning problem that will depend on the provided initial weights.

As an alternative to the k-means algorithm, one may instead employ the second method, i.e., the EM-algorithm, which may be viewed as a probabilistic way of doing clustering with soft thresholding. As opposed to the k-means algorithm, the EM-algorithm assumes that each cluster results from a generative model. We here assume that the data may be described by a GMM. That is, we assume that the different clusters may be well modeled using d-dimensional Gaussian distributions, thus taking into consideration that we are looking for elliptical, or somewhat elliptical, patterns, where the mean is an estimate of the cluster center, and where the covariance matrix may be used to determine the d semiaxises of the ellipse. The GMM allows for the clusters to overlap, meaning that some data points may belong to more than one cluster. The use of the EM-algorithm is to find the local maximum likelihood estimates of the parameters in a given statistical model, in our case the mean, μ_k, and covariance, Σ_k, of the underlying data, Z, as well as the prior class probabilities, π_k. The jammed nodes $\mathcal{U} = (u_1, \ldots, u_J)$ constitute the observations of the underlying data $Z = (z_1, \ldots, z_J)$, and the prior class probabilities are formed as $p(z_i = k) = \pi_k$ for $i = 1, \ldots, J$. The conditional distributions of the observed variables are then formed as

$$p(u_i|z_i = k) \in \mathcal{N}_d(\mu_k, \Sigma_k)$$

where d denotes the dimensionality of the considered scenario. The Gaussian mixture model for node u_i may then be specified as

$$p(u_i) = \sum_{k=1}^{K} p(u_i|z_i = k)\pi_k$$

where the density for all of $\mathcal{U}$ becomes

$$p(\psi; u_1, \dots, u_J, Z) = \prod_{i=1}^{J} \sum_{k=1}^{K} p(u_i|z_i = k)\pi_k \tag{10}$$

giving rise to the log-likelihood function

$$\ell(\psi; u_1, \dots, u_J, Z) = \sum_{i=1}^{J} \log\left(\sum_{k=1}^{K} p(u_i|z_i = k)\pi_k\right) \tag{11}$$

where $\psi = [\mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_k, \pi_1, \dots, \pi_K]$ is the set of parameters we wish to estimate. Numerical evaluation of (11) is generally difficult, due to the sum of terms within the logarithm [19]. This is the problem that is solved iteratively by the EM-algorithm. The method alternates between first forming the conditional expectation (E) of an observation given the current parameter estimates, and then a maximization (M) of the expectation function to find new parameter estimates. The expectation at iteration m may be specified as

$$\Gamma^{(m)}(\psi_k^{(m)}, u_i, k) = \mathbf{E}\left(\ell(\psi; \mathcal{U}, Z)|\mathcal{U}, \psi^{(m)}\right) = \sum_{i=1}^{J} \sum_{k=1}^{K} \gamma_i \ell(\psi_k^{(m)}, u_i, k) \tag{12}$$

where $\psi_k = (\mu_k, \Sigma_k, \pi_k)$ are the parameters for distribution k. Using Bayes' formula, it may be concluded that

$$\gamma_i = p(z_i = k|u_i; \psi^{(m)}) = \frac{\pi_k^{(m)} p(u_i|\mu_k, \Sigma_k)}{\sum_{t=1,t\neq k}^{K} \pi_t p(u_i|\mu_t, \Sigma_t)} \tag{13}$$

Algorithm 4 EM-clustering

```
Initialize: $\psi \leftarrow \psi_0$, $m \leftarrow 1$
repeat
    Compute $\Gamma^{(m)}(\psi_k^{(m)}, u_i, k)$ using the expectation step in (12)
    Compute $\psi^{(m+1)}$ using the maximization step in (14)
    $m = m + 1$
until diff($\psi^{(m+1)}, \psi^{(m)}$) $<$ tol, or max iterations
```

The maximization step is then completed by maximizing (12) with respect to the parameters in ψ, setting

$$\psi^{(m+1)} = \underset{\psi}{\operatorname{argmax}} \, \Gamma^{(m)}(\psi_k^{(m)}, u_i, k) \tag{14}$$

The method then iterates between the E-step and the M-step until convergence. The method assumes that the number of clusters to estimate, K, is known, and starts out by randomly selecting the parameters for the K distributions (if *a priori* knowledge exists, this may be used when initiating the parameters, to get a faster convergence). In our case, we consider different Gaussian distributions specified by their mean and covariance. As an initial guess for the prior class probabilities, we set $\pi_i = 1/K$, for $i = 1, \dots, K$.

Finally, one may employ the third approach based on Gibbs sampling, forming a Markov Chain Monte Carlo (MCMC) method, sampling from the assumed posterior distributions, as summarized in algorithm 5. The main difference between Gibbs sampling and the EM-algorithm is that where the EM-algorithm maximises the conditional distributions, the Gibbs sampling method samples them instead [19]. The method starts in the same way as the EM-algorithm, with randomly selecting the parameters for the K initial distributions, in our case Gaussian distributions, parametrized by $\psi = [\mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K]$. The method then continues by sampling from the posterior distributions of the class belongings in (13). The algorithm terminates when the posterior distributions do not change significantly in the consecutive iterations.

Algorithm 5 Clustering with Gibbs sampling

```
Initialize: $\psi \leftarrow \psi_0$
repeat
    for $i \leftarrow 1$ to $K$ do
        Compute $\gamma_i$ using (13)
    end for
    Compute estimates of $\hat{\psi}$ using (14)
    Sample new estimates of $\mu_k, \Sigma_k \in N(\hat{\mu}_k, \hat{\Sigma}_k)$
until convergence, or maximum number of allowed iterations reached.
```

Solving subproblem $\mathcal{P}_2$

After determining the partitioning, the next step is to find the MVJR for each of the clusters. This may be solved independently and in parallel for each cluster, $U_k = (u_{q_1}, u_{q_2}, \dots, u_{q_j})$, where $u_{q_1}, \dots u_{q_j} \in J$. In this work, we consider a set of different predefined jamming patterns, or combinations thereof, collected in a predefined bank of pattern templates. For each cluster, we proceed to fit a jamming pattern from the set of potential template patterns. Even though the patterns differ, the methodology of finding the MVJR is similar for most patterns. It is initially assumed that the center of the MVJR, c_k is the same as the cluster center, μ_k. This assumption implies a centered solution (which may well be incorrect). For each cluster, the MVJR problem may then be formulated as the convex problem

$$\begin{aligned} \min_{Q_k(\theta_k), c_k} \quad & \det Q_k(\theta_k)^{-1} \\ \text{s.t.} \quad & \left(u_i - c_k\right)^{\mathrm{T}} Q_k(\theta_k) \left(u_i - c_k\right) \leq 1, \quad i = 1, \dots, q \end{aligned} \tag{15}$$

Noting that taking the logarithm of the objective function does not change the location of the minimum, one may reformulate the problem as

$$\begin{aligned} \min_{\tilde{Q}_k(\theta_k), c_k} \quad & -\log\det \tilde{Q}_k(\theta_k) \\ \text{s.t.} \quad & \left(\hat{u}_i - c_k\right)^{\mathrm{T}} \tilde{Q}_k(\theta_k) \left(\hat{u}_i - c_k\right) \leq 1, \quad i = 1, \dots, q \end{aligned} \tag{16}$$

where $\tilde{Q}_k(\theta_k) \in \mathbb{R}^{d+1 \times d+1}$ and $\hat{u}_i = (u_i, 1)$, implying that the problem has been augmented with an additional dimension (see also [25]). In order to form the dual of (15), one may introduce the Lagrangian multipliers $\lambda = (\lambda_1, \dots, \lambda_q)$, forming the Lagrangian, such that

$$L(\tilde{Q}_k(\theta_k),\lambda) = -\log\det\tilde{Q}_k(\theta) + \sum_{i=1}^{q}\lambda_i\left(\left(\hat{u}_i - c_k\right)^{\mathrm{T}}\tilde{Q}_k(\theta_k)\left(\hat{u}_i - c_k\right) - 1\right) \quad (17)$$

This problem is strictly convex and satisfies Slater's condition, implying that the dual and primal problems have the same solution [25]. The dual problem may be formulated as

$$\begin{aligned}&\underset{\lambda}{\text{argmax}}\ \log\det(\sum_{i=1}^{q}\lambda_i(u_i - c_k)(u_i - c_k)^T) - \mathbf{1}\lambda + d + 1\\&\text{s.t. } \lambda_i \geq 0, \quad i = 1, \dots, q\end{aligned} \quad (18)$$

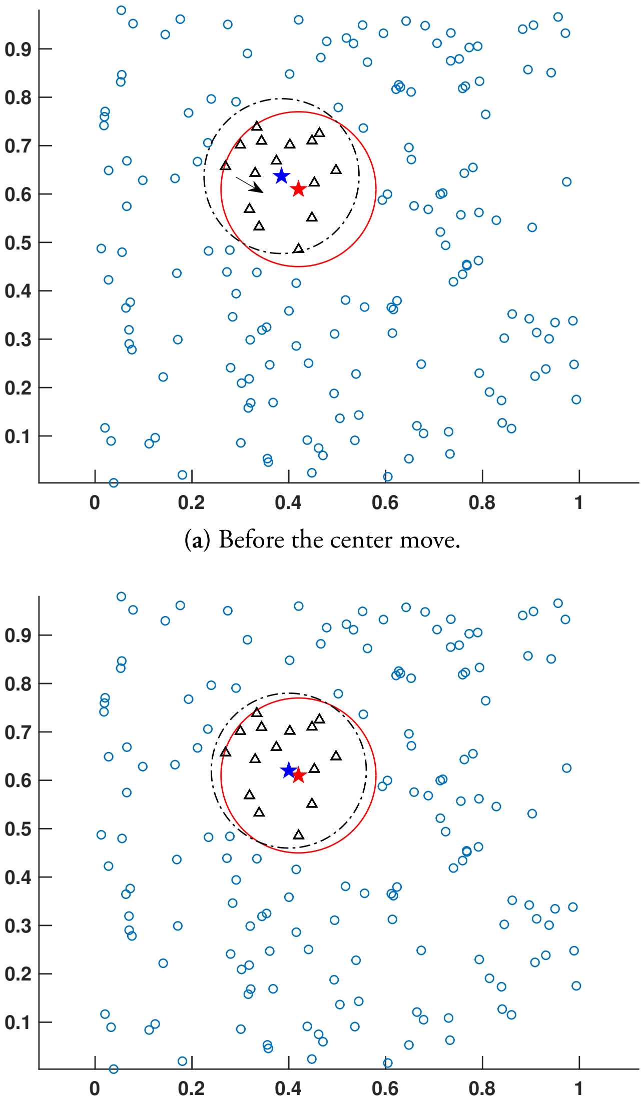
where **1** denotes a vector of ones. This problem is convex and may be efficiently solved by standard optimizers, such as, e.g. cvx [26].

Solving subproblem $\mathcal{P}_3$

After determining the clusters and their centers, we proceed to determine if the found solution explains the observed nodes, i.e., if the formed jamming patterns cover all jammed nodes and none of the unjammed nodes. This may be violated for three reasons.

- Reason $\mathcal{R}_1$: The assumption that the jammer locations are the same as the cluster centers, $\mu_k = c_k$, may not hold.
- Reason $\mathcal{R}_2$: The partitioning may not be optimal.
- Reason $\mathcal{R}_3$: The number of jammers may need to be increased.

Considering each of these reasons in turn, for $\mathcal{R}_1$, the assumption that $c_k = \mu_k$ is fairly strong, and will, in general, not hold. One way to deal with this issue is to compute the farthest distance between the cluster center and the nodes, and then move the proposed center point a prespecified distance, Δ_d, in the direction of the point that is farthest from the center point, as illustrated in figure 3. After this move of the center, the steps in subproblems $\mathcal{P}_2$-$\mathcal{P}_3$ are reiterated. If moving the center point does not solve the problem, we proceed to contemplate the second reason, $\mathcal{R}_2$, i.e., that we ought to consider a different partitioning of the sets [27].



(a) Before the center move.

(b) After the center move.

Figure 3: Illustrating the center move operation whereby the blue center is moved closer to the red (true) center, thus ensuring a better fit of the jamming patterns.

Because the inclusion/exclusion of q points in a set is a problem of dimensionality 2^q, an exhaustive partitioning search quickly becomes intractable. An alternative route to finding a good candidate set is to study the concept of active sets, in combination with the weight matrix from the partitioning, W. The active set, $\mathcal{Z}_k$, of a point set U_k is the subset of nodes in U_k that are on the boundary of the MVJR, as illustrated in figure 4. The points in the active set are the points that determine the shape of the MVJR, whereas the points that are not part of the active set are found in the interior of the MVJR, and will have no direct effect on its shape. Thus, when changing the partitioning, only changes to the active set will have an effect on the MVJR. Having identified the active set, the next step is to determine which points should be included or excluded from the set. Let the thresholding matrix from the partitioning, W, denote the magnitude of the set belonging for each point, where W is a dense matrix. The procedure to decide which points to include or exclude is as follows:

- When the partitioning is made, the set belonging for all points and all sets are gathered in the thresholding matrix W. A point is considered as part of the set if its set belonging for that set is above a predefined threshold, κ. Each partitioning has an associated active set that consists of the points in the set with more than one set belonging. For each such active set, the points are ranked based on their magnitude of belonging to the current set, as specified in the thresholding matrix W.
- Next, a second ranking is done whereby the points are ranked based on the relative difference between the set belonging to the current set and the set belonging to the second set. In practice, this means that the points are ranked based on their difference in set belonging. The relative distances are then normalized, and the ranking is done in descending order.

Based on the above feature criteria, a ranking is formed of which nodes to include or exclude. If the points to be included are only interior points of the other sets, the procedure in subproblem $\mathcal{P}_2$ is repeated for the current set alone, and if the points to be included are active points of a different set, subproblem $\mathcal{P}_2$ is repeated for the current set as well as the set from which the included points came from. If the steps taken above do not solve the problem, the assumed number of jammers is insufficient to explain the observed jammed nodes and the number of jammers should be increased by one, after which the entire procedure begins anew.

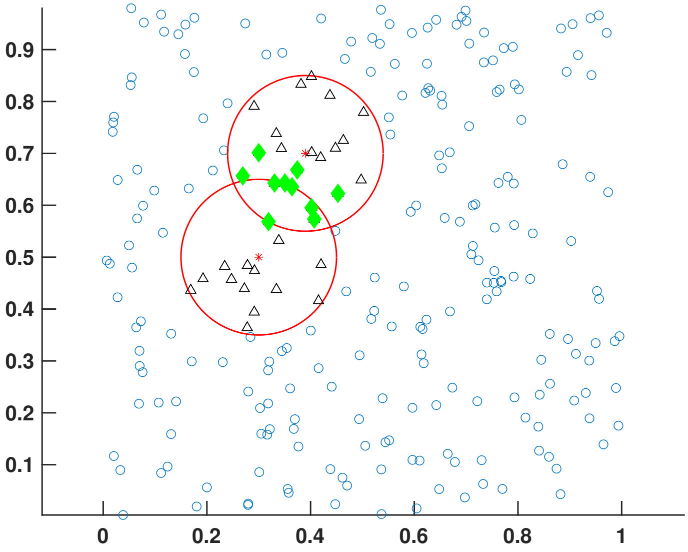


Figure 4: An illustration of two jammers where the active set of the jammer located at (0.3,0.5) is marked with green diamonds.

3 Jammer tracking

It is not uncommon that jamming sources are moving over time, causing the jamming pattern to be nonstationary. As the jammers move through the network, the set of jammed nodes will then differ, providing additional information of the jammers' locations and movements. The problem of tracking moving objects via indirect measurements occurs in numerous applications and the topic has attracted notable attention in the literature [7, 8, 28, 29]. A common feature in many tracking algorithms, such as the popular Kalman filter and its derivatives, is that the number of objects to track is assumed to be known *a priori*, or can at least be assumed to be constant over the tracking period. This assumption notably simplifies the problem, and often improves the performance of such algorithms, but also poses a severe restriction for problems that do not meet this condition. One potential solution when this assumption is unrealistic is to also estimate the number of states in every frame, and each time an object is added or removed, a new tracking model is initiated, combined with some method of fusing the tracks between the different tracking models. Although this approach can work, it is reasonable to expect a notable loss of performance due to the addition of the often difficult model order estimation procedure, and it is generally better to find alternative solutions instead.

The here considered problem necessitates that the segmentation pattern is estimated for each time snapshot, while also allowing for overlapping jammers, as well as for jammers that migrate in or out of the tracking region over time. The possibility of overlapping jamming regions provides further challenges to the localization problem, but also new possibilities, such as allowing for the occurrence of new jamming sources over time. A natural solution to the resulting problem is to use a tracking approach based on the Viterbi algorithm, as was also done in the related problem of cell tracking proposed in [9].

Similar to the Kalman algorithm, the Viterbi algorithm may be applied to situations where we are provided with a set of observations of an underlying unobservable event variable [12, 30]. The Viterbi algorithm has been recommended for tracking point like objects [31], such as a jammer location. It is a probabilistic algorithm, based on dynamic programming [32], that seeks to maximize the *a posteriori* probability of the possible events given the set of observations. In the discussed scenario, this is the same as finding the most likely path taken by the jammers, given observations of the set of jammed nodes over time. Here, the observation set consists of the jammed nodes, and the thus inferred jammer locations, and the set of events consists of six possible events, collected in Table 1. For each time frame, the segmentation step provides information about the location of the jammers, and the aim is to link the jammers between consecutive time frames, thus tracking the progression of the jammers over time, often referred to as tracking by detection. Figure 5 illustrates the event when two overlapping jamming regions divide and appear as two separate jammers, known as a splitting event, where in the first frame there appears to be only one jammer present, whereas in the second frame both jammers can be observed; such a scenario may happen when two jammers overlap, and then move in different directions. Figure 6 illustrates the different possible paths created between frame 1 and frame 2, along with their most likely paths. The events represent the different scenarios that may happen between two consecutive frames, and when tracking a batch of T frames, the path consists of a sequence of events of length $T - 1$. The Viterbi algorithm is a global optimization algorithm, and the dynamic programming implementation of the algorithm means that every possible path is investigated. Denote the set of events as E_B, with with $E_B = \{e_1, \dots, e_B\}$, where $B = |E_B|$. Considering a batch of T frames, the number of possible paths that have to be evaluated is B^T, i.e., the number of paths grows exponentially with the batch size T, necessitating the use of relative low values of B, especially when T is large.

Formally, the problem formulation when running the Viterbi algorithm is that of finding the most likely path, given the set of observations. Considering a batch of T frames, the algorithm starts at the first frame with the observation O_0, and finds the most likely path between O_0 and O_1, the second observation. The algorithm then proceeds in this way, finding the most likely path between observation O_{t-1} and O_t, for $t = 1, \ldots, T$. This procedure of finding the global optimum by dividing the problem into smaller parts and solving them individually is what is known as dynamic programming. The assumption enabling the use of dynamic programming is known as the Markov property, which assumes that the path history leading up to observation O_{t-1} has no bearing on the most likely path between O_{t-1} and O_t.

In an ideal scenario, the probability distributions of the different events are all unimodal, and finding the most likely path may be done by considering only the forward path between O_{t-1} and O_t, for $t = 1, \ldots, T$. However, often the probabilities are not unimodal, and considering the forward path is not guaranteed to find the optimal path. Fortunately, the Viterbi algorithm is able to handle also the multimodal case [31]. By also considering the backward propagation, i.e., the most likely path between O_t and O_{t-1} if traversing in the reverse order, multimodal probability distributions may be handled. Hence, the Viterbi algorithm is known as a forward-backward algorithm as is considers both the forward and the backward path.

In order to rank the different paths, a scoring function has to be defined, that assigns different scores to different paths. The Viterbi algorithm is a probabilistic algorithm, and thus the scoring function is associated to the *a posteriori* probabilities of the different paths. As noted, a path may be described as a time indexed set of events, $\Lambda_t = \{e_b^t\}$, with $e_b \in E_B$, and $t = 1, \ldots, T$. For example, the optimal path in Figure 6 may be described as $\Lambda = \{e_C^1, e_S^2, e_C^3\}$. In this work, we adopt the same scoring function used in [9], i.e.,

$$g(\Lambda) = \sum_{t=1}^{T} \sum_{b=1}^{B} \log(p(\Lambda_t = \Lambda_{e_b^t})) \tag{19}$$

Table 1: The possible events taken by the jammer(s). *Count* refers to the case when the number of jammers is the same in two consecutive frames, *splitting* is when jammers that previously overlap are identified, *quiet* is when a jammers is suddenly switched off, *migration in* is when an additional jammer is observed without a splitting event, *migration out* is when a jammer leaves the observation area, and *pattern merging* is when simpler jamming patterns are merged to form a more complicated pattern, such as the one seen in, e.g., beamforming.

Event description	Abbreviation
Count	C
Splitting	S
Quiet	Q
Migration in	MI
Migration out	MO
Pattern merging	PM

where p denotes the assumed probability function[5]. Here Λ is the overall optimal path, and Λ_t is the event of the optimal path in frame t, and $\Lambda_{e_b^t}$ is the candidate path that is in event e_b at time t, an example of which is illustrated in Figure 6. Formally, the optimization problem we solve when using the Viterbi algorithm is that of

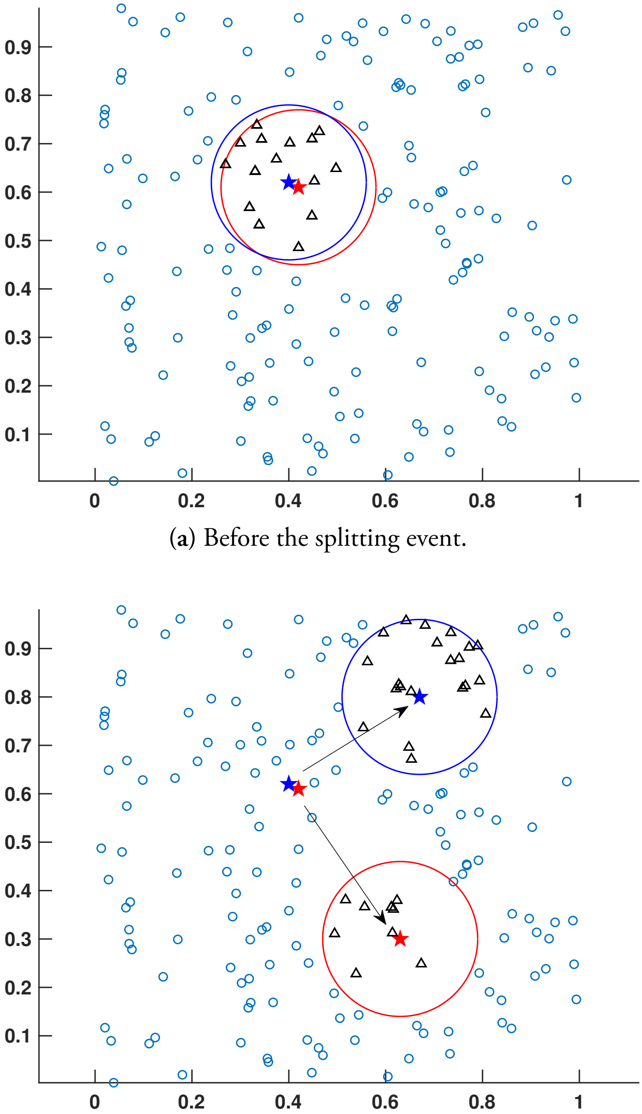
$$g_{\max}(\Lambda) = \max_{e_1,\dots e_B} \sum_{t=1}^{T} \sum_{b=1}^{B} \log(p(\Lambda_t = \Lambda_{e_b^t})) \tag{20}$$

The overall Viterbi algorithm is illustrated in algorithm 6, where ϕ is a function to keep track of the second to last event in the highest scoring path.

4 Numerical Experiments

We proceed to evaluate the performance of the proposed methods using Monte-Carlo simulations, examining both cases when jammers leave and enter the area of interest. We begin with evaluating the different segmentation methods in terms of their jammer localization performance. We examine two scenarios, one with circular jamming regions, as is commonly assumed in the

[5] The usage of the logarithmized probabilities is merely for numerical convenience, as the sum offers better numerical performance than the multiplication.



(a) Before the splitting event.

(b) After the splitting event.

Figure 5: An illustration of a splitting event, wherein two jammers move apart.

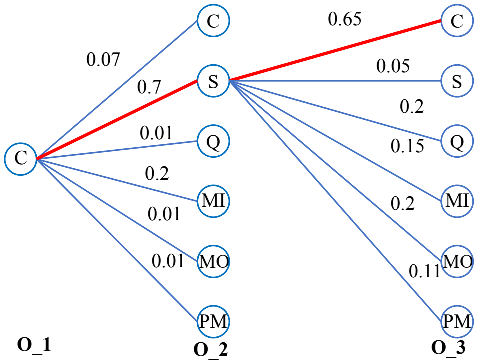


Figure 6: An illustration of the possible paths between the observations in Figure 5. Example probabilities of each path are marked on each edge, with the nodes in the path with highest probability being highlighted.

literature, and one with oval jamming regions. It is worth noting that most of the state-of-the-art methods for jammer localization, such as X-ray [6], CJ [5], and WCL [3], are based both on the assumption that the jamming regions are circular, and for CJ and WCL also that the number of jammers is known *a priori*. We then proceed to evaluate the jammer tracking method, considering two different scenarios, namely that of tracking jammers with a fixed orientation as well as tracking jammers with changing orientation. The results illustrate how tracking may be used to improve jammer localisation, and the effect that knowledge about the orientation can have on the accuracy of the localisation as well as on the convergence rate of the localisation. Moreover, we illustrate the tracking performance of our method when *splitting*, *migration in*, and *migration out* occur, all cases when the number of objects to track change over time. The tracking performance is evaluated with respect to state-of-the-art-methods, clearly illustrating the need for a tracking method that can handle a variable number of states.

Finally, the methods are evaluated on a scenario with a so-called *pattern merging* event. In this case, the algorithm first identifies the number of states to track, but as time progresses it becomes clear that the pattern does not stem from individual jammers, but rather from one jammer with a complicated jamming pattern, that may be viewed as composed of many simpler jamming patterns. This situation may be viewed as the merging of states, forcing the algorithm to backtrack and

Algorithm 6 Viterbi algorithm

```
In each forward step we seek to maximize (19), by finding $g_{max}(\Lambda^f)$.
Initialize: $g_{max}(O_0)$
*Forward algorithm*
for $t \leftarrow 1$ to $T$ do *Loop over the frames*
    for $e_b = e_1, \dots, e_B$ do *Loop over the events*
        Compute $g_{max}(\Lambda^f_{e^t_b})$ using (19)
        Compute $\phi(\Lambda^f_{e^t_b})$ as the second to last state in the highest scoring path.
        *This is needed in the backward path.*
    end for
end for
In each backward step we seek to find the most likely path $\Lambda^{back}$.
Initialize: $E_{max} = O_T$ *The algorithm starts at $O_T$.*
$\Lambda^{back} \leftarrow \{E_{max}\}$ *Assign the likeliest event to the path*
for $i \leftarrow T$ to 1 do *Loop over the frames*
    $E_{max} \leftarrow \phi(E_{max})$ Find the next step in the backward search by using $\phi$
        from the forward algorithm.
    $\Lambda^{back} \leftarrow \{E_{max}, \Lambda^{back}\}$ *Add the likeliest state to the path*
end for
```

update its pervious jammer localisations. For the simulations in this section we have used threshold value of set belonging set to $\zeta = 0.3$, and when a center move occurs, the predefined distance is set to $\Delta_d = 0.01$ standard units. For the initial parameters estimated in the Gibbs and EM methods the means were chosen as the mean of the points and the standard deviations were all set to 0.01.

4.1 Stationary jammers

Circular jammers

Initially, we examine a case with K jammers, all having circular jamming regions, albeit with different radii, r_i, for $i = 1, \dots, K$. The area under consideration is a 1×1 square in normalized units, and with the jammer radii all selected uniformly in the interval $[0.05, 0.2]$, $r_i \in U[0.05, 0.2]$, $\forall i$. The jammers initial locations are selected uniformly at random, with the constraint that their initial jamming areas may not overlap more than 50%, in order to allow for a comparison with methods with such restrictions. For a fixed maximum overlap, the two main external parameters affecting the localisation accuracy are the number of jammers present, K, and the node density. To control for these two factors, two simulation studies were performed, one with a fixed K but variable node densities, and one with fixed node density but a variable number of jammers. Additionally, the effect of oracle knowledge of K was studied in both simulations. The performance was evaluated with respect to the overall average localisation error, here defined as the average root mean squared error (ARMSE) of all jammer localizations, i.e.,

$$ARMSE = \frac{1}{K} \sum_{k=1}^{K} \sqrt{(x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2} \tag{21}$$

where $\hat{x}_k$ and $\hat{y}_k$ are the estimated jammer coordinates of the k:th jammer, (x_k, y_k), respectively, and for the cases where K was unknown, also with the respect to the percentage of correct detections, here defined as the percentage of Monte Carlo simulations where the correct number of jammers were identified. For the variable node density scenario, we consider the case $K = 3$, and three node densities: 250, 350, and 450 nodes. In the variable jammer scenario, the node density is fixed to 350 nodes, whereas the number of jammers present varies from $K = 1, \ldots, 4$. The results are displayed in Figures 7-9. In cases where a method is unable to form an estimate, such as for WCL and CJ when $K > 1$, the results have been omitted from the figures. Studying the case with variable node density, one may note that the localisation error decreases when the node density increases. This should be expected as a higher node density will result in more data points, thus increasing the accuracy of the localisation. Furthermore, one may note that of the proposed methods, the EM-algorithm and the Gibbs method perform on par with the state of the art X-ray method, whereas the k-means algorithm has a slightly lower accuracy, which in turn may be compensated by its lower computational complexity. For $K = 1$, it may be noted that CJ and WCL estimates perform well in case the jamming region is circular, as these methods assume, but this performance degrades notably if this is not the case. Also, it should be stressed that both fail to yield estimates when $K > 1$. Moreover, one may note that the localisation accuracy decreases as the number of jammers increase, a behaviour that may be attributed to all the proposed methods. This is as expected, as the number of unambiguous data points can only increase or remain the same with an increasing number of overlapping jammers, causing the marginal information gained for each data point to be bigger for small values of K.

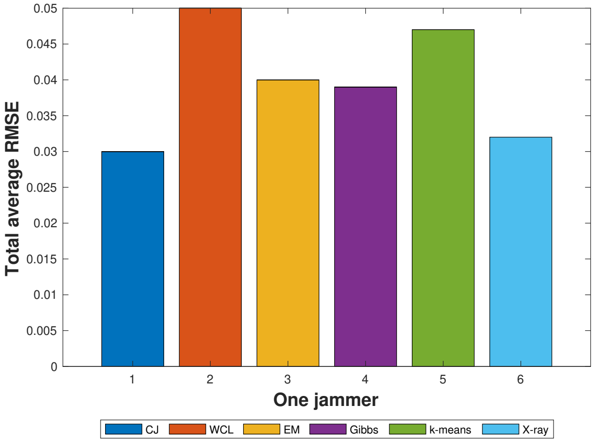


Figure 7: Total ARMSE localization estimation error for the single jammer case.

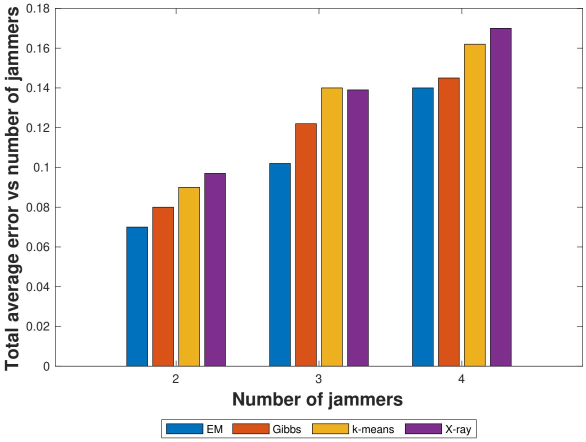


Figure 8: Total ARMSE localization estimation error with respect to all jammers vs number of jammers.

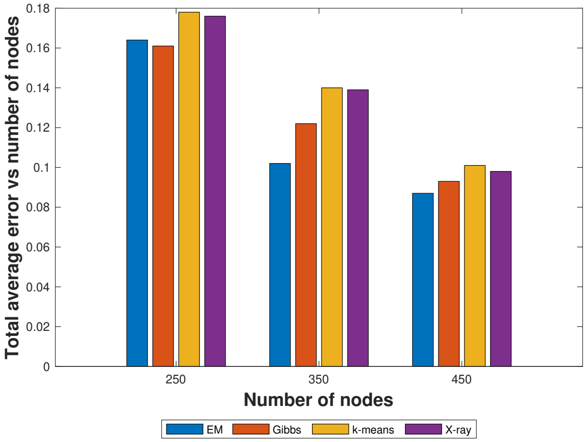


Figure 9: Total ARMSE localization estimation error with respect to all jammers vs number of nodes.

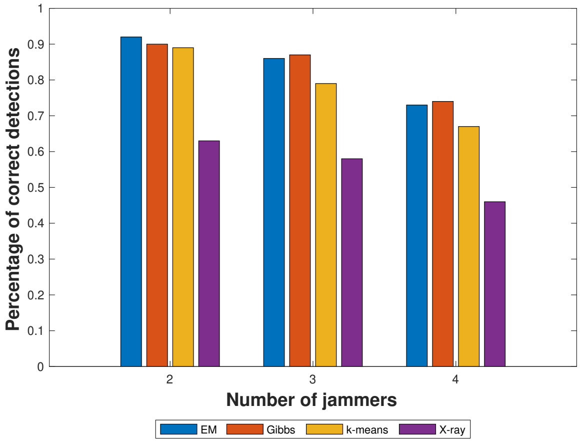


Figure 10: Percentage of number of correct detections over all Monte Carlo simulations.

Oval jammers

In many real situations, jamming patterns are often directed, meaning that the degree of jamming is larger in some direction than in other. A simple way to model this is to consider oval jamming patterns instead of circular ones. To investigate how this affects the performance of the discussed methods, we proceed to modify the earlier simulations to instead employ elliptical jamming patterns, with the semiaxes of the ellipses, r_i^a and r_i^b, being randomly generated as independent uniform stochastic variables from $r_i^a, r_i^b \in \mathcal{U}[0.05, 0.2]$, $\forall i$. The rotation of the jamming regions were similarly drawn as having rotations $\alpha \in U[-\pi, \pi]$. An example of the resulting setup is shown in figure 2. The resulting localization performance results are shown in figures 11-13, wherein one may note that for the elliptical case, all of the proposed segmentation methods outperform other methods with the CJ and WCL again failing to yield estimates for $K > 1$. Notably, the X-ray method can now be seen to suffer a significant performance loss. This happens as the X-ray method specifically assumes a circular jamming region. Moreover, it can be noted that the X-ray method suffers significant performance losses in the elliptical case, as illustrated in figure 10. This is because of the assumption the method makes about circular jamming regions.

4.2 Moving jammers

Proceeding, we examine the performance of the tracking algorithm by considering the case of $K = 2$ jammers moving in a network with 350 nodes. The jammers under consideration have elliptical jamming patterns with $r_i^a, r_i^b \in U[0.05, 0.2]$, $\forall i$, and $\alpha \in U[-\pi, \pi]$. As before, the initial locations of the jammers were selected at random, with the constraint that the jamming regions should overlap of at least 5 %. In this scenario, no upper bound on the amount of overlap allowed was imposed. The jammers are modeled as moving in different directions, along linear paths, $y_1(t) = 0.5x_1(t) + 2$, and $y_2(t) = -0.3x_2(t) + 5$, where $t = 1, \dots, T = 20$ denotes the time steps.

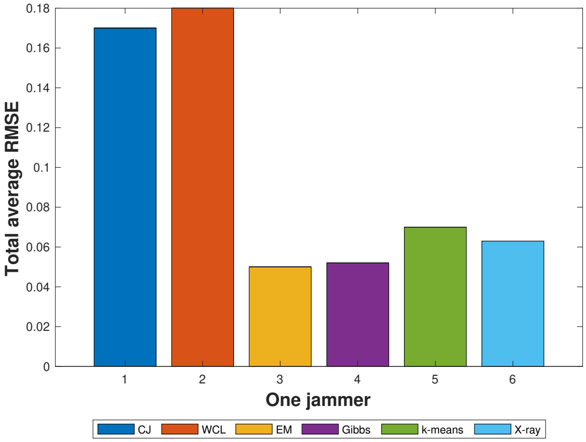


Figure 11: Total ARMSE localization estimation error for the single jammer elliptical case.

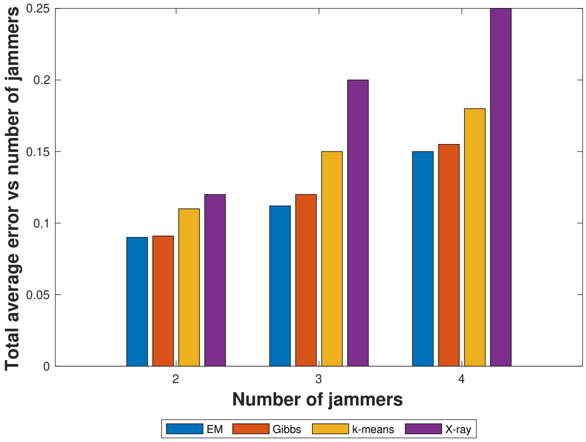


Figure 12: Total ARMSE localization estimation error with respect to all jammers vs number of jammers for the elliptical case.

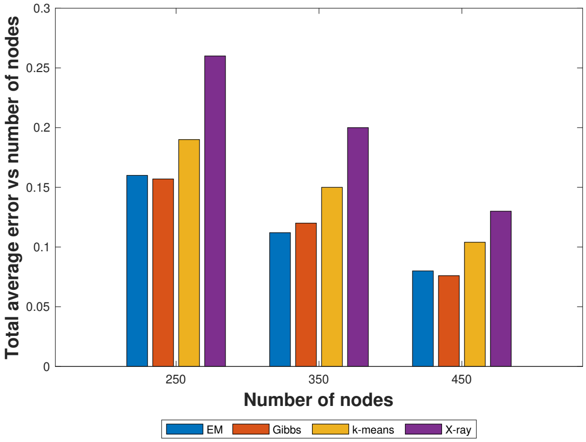


Figure 13: Total ARMSE localization estimation error with respect to all jammers vs number of nodes for the elliptical case.

Fixed orientation

As time progressed, the jammers' orientation here remained fixed, i.e., α does not vary over time, this in order to illustrate how a time-varying orientation affects the results (as shown below). The proposed method here employs the EM segmentation algorithm. The tracking was evaluated for unknown K where the value of K was obtained from the segmentation step, and the performance of the different tracking algorithms is presented in Table 2. The tracking was evaluated for the Kalman filter (KF) [33], and a particle filter [24, 34]. It should be noted that the Kalman filter fails to produce accurate estimates when the number of jammers change. Such estimates have been excluded from the ARMSE calculation. The proposed method employs the estimated pattern to improve the localization estimates.

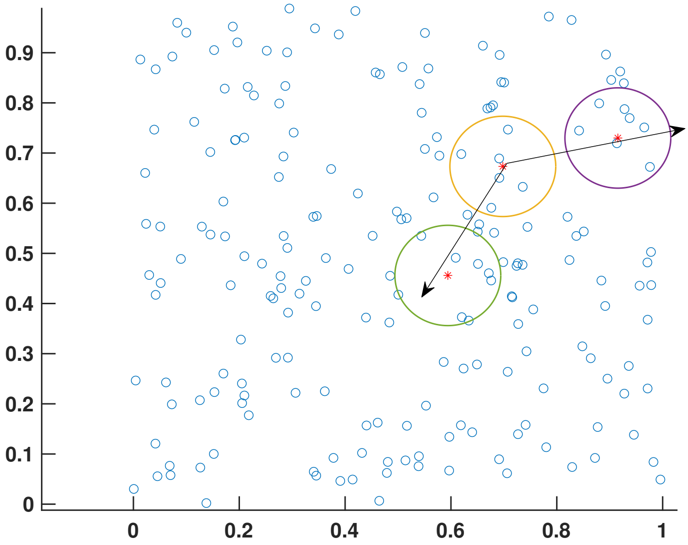


Figure 14: An illustration of a tracking scenario where the purple circle has been split up in the blue and green circle.

Table 2: Path errors for the three different methods tested.

Method	Final ARMSE
Proposed	**0.08**
KF	0.20
Particle filter	0.22

Table 3: Path errors for the three different methods tested.

Method	ARMSE fixed orientation	ARMSE variable orientation
Proposed	**0.0175**	**0.0195**
KF	0.05	0.07
Particle filter	0.07	0.09

Variable orientation

Proceeding, we examine how the performance of the methods are affected if the jammer's orientation is allowed to change over time. It may well happen that jammers (or persons operating such devices) aims at disrupting a specific target of interest and thus may be interested in changing the orientation of its jamming in order to maximize the jamming effect. This makes the detection step more difficult as the learning curve effectively has a smaller rate of change than in the case when the orientation may be assumed to be fixed, thus making the convergence slower. To investigate this effect, we examine a scenario in which a single jammer moves through the network. The simulation setup was the same as before, following the path given for $y_1(t)$ but the studied cases now also include a variable orientation with the orientation α drifting with time, such that $\alpha(t) = \alpha + n(t)$, where $n(t) \in \mathcal{N}(0,1)$. The resulting localization error of the proposed method is shown in Table 3, along with comparison methods.Figure 15 illustrates the learning rate for the first 7 steps for the fixed and variable case, respectively. One may notice that the localisation error decreases with time, illustrating how knowledge about the orientation helps to improve the estimate, as may be expected, but also that this gain diminishes somewhat over time as the method is able to utilize the estimated jamming pattern to accurately locate the orientation.

4.3 Merged patterns

Finally, we examine how more complicated jamming patterns may be constructed by merging simple jamming patterns together. If one only has one observation, it is impossible to determine if the observed pattern is from multiple overlapping jammers, or if it is one jammer with a complicated jamming pattern. However, if several consecutive observations are available, it may be possible to tell the scenarios apart. If the pattern is made up of individual jammers, it is reasonable to believe that as time progresses they will move independently of each other, thus changing the relative distance between their centra. In contrast, if the pattern is made up of just one jammer, it is likely that the relative distance between their centra should remain unaltered. To setup a simulation study for this case, we considered a jamming pattern consisting of 4 joined jammers, all with elliptical jamming regions, all sharing a common merging point such that the entire pattern moves as one, with their shared center points $y(t)$ moving as $y(t) = 0.5x(t)$, $t = 1, \ldots, 20$. When the tracking starts, the method tracked 4 individual jammers, however, as time progressed and the

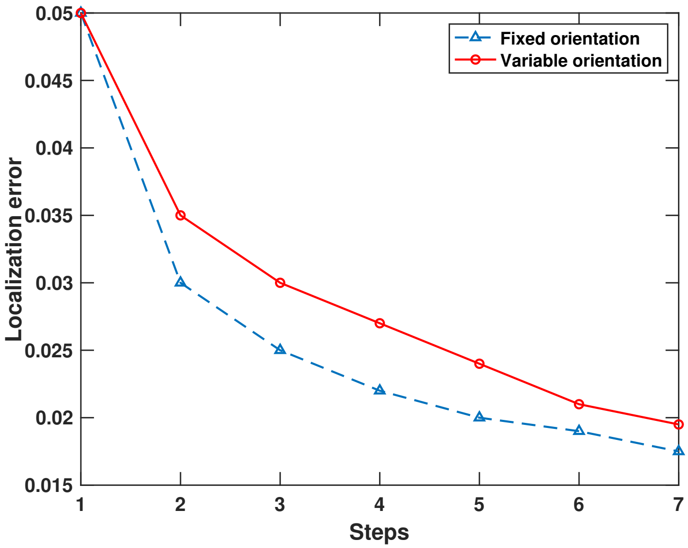


Figure 15: Convergence of the jammer localization problem with fixed and variable orientation, illustrated for the proposed method.

relative distances between the jammers remained unaltered, the method instead started to track the entire entity as one object, updating the previous track as well. The methods was evaluated for three different node densities, 250, 250 and 450. The results are presented in Table 4, only results for the proposed method are presented, due to the inability of the other methods to capture the scenario.

Table 4: RMSE for the merged pattern.

Node density	Final RMSE	Number of steps before merged pattern detected
250	0.20	10
350	0.18	7
450	0.15	4

5 Conclusion

To conclude, this article presents a procedure for the localisation of multiple, overlapping jammers, with general jamming patterns. We evaluate three methods for the segmentation and localisation step, and their resulting localisation performances are compared to other state of the art jammer localisation techniques, illustrating the improved performance by using the proposed techniques. The proposed methods are evaluated for different jamming scenarios, including different types of jammers, different amount of overlap, and different network settings. Additionally, a jammer tracking method is presented to track the jammers' movement in the network. The proposed algorithm is flexible regarding the number of tracked objects, as opposed to other state of the art tracking methods the require knowledge about the number of agents to track.

References

[1] H. Liu, W. Xu, Y. Chen, and Z. Liu, "Localizing jammers in wireless networks," in *IEEE International Conference on Pervasive Computing and Communications*, 2009.

[2] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, pp. 28–34, 2000.

[3] J. Blumenthal, R. Grossman, F. Golatowski, and D. Timmermann, "Weighted Centroid Localization in Zigbee-based Sensor Networks," *IEEE International Symposium on Intelligent Signal Processing*, 2007.

[4] H. Huang, X. Hu, X. Zhang, P. Zhang, and Y. Li, "An Optimal Jammer Selection for Improving Physical-Layer Security in Wireless Networks with Multiple Jammers," in *International Wireless Communications and Mobile Computing Conference*, 2016.

[5] Y. Sun, R. Molva, M. Onen, X. Wang, and X. Zhou, "Catch the Jammer in Wireless Sensor Network," in *IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, 2011.

[6] T. Cheng, P. Li, and S. Zhu, "Multi-jammer localization in wireless sensor networks," in *2011 Seventh International Conference on Computational Intelligence and Security*, 2011, pp. 736–740.

[7] W. Aldosari, M. Zohdy, and R. Olawoyin, "Tracking the Mobile Jammer in Wireless Sensor Networks using Extended Kalman Filter," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2019, pp. 0207–0212.

[8] L. Pang, P. Guo, X. Chen, and Z. Xue, "Tracking the mobile jammer continuously in time by using moving vector," in *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, 2017, vol. 1, pp. 43–48.

[9] K. Magnusson, J. Jaldén, P. Gilbert, and H. Blau, "Global linking of cell tracks using the Viterbi algorithm," *IEEE Transactions on Medical Imaging*, vol. 34, no. 4, pp. 911–929, 2015.

[10] S. Zafari, T. Eerola, J. Sampo, H. Kälviäinen, and H. Haario, "Segmentation of Overlapping Elliptical Objects in Silhouette Images," *IEEE Transactions on Image Processing*, 2015.

[11] T. Cheng, P. Li, and S. Zhu, "An algorithm for jammer localization in wireless sensor networks," in *IEEE 26th International Conference on Advanced Information Networking and Applications*, 2012, pp. 724–731.

[12] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[13] Y. Liu, "Research on intelligent digital printmaking printing system based on viterbi algorithm," in *2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI)*, 2022, pp. 1055–1058.

[14] D. Liu, W. Fang, H. Zhou, and Y. Li, "Bigram chinese word segmentation by viterbi algorithm," in *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, 2009, vol. 5, pp. 364–368.

[15] N. Zhang, S. Hao, and Y. Li, "An improved fast viterbi algorithm for track-before-detect," in *2010 3rd International Congress on Image and Signal Processing*, 2010, vol. 7, pp. 3125–3128.

[16] V. N. Hulyalkar, Priyatamkumar, and R. Joshi, "Hard and soft decision decoding using viterbi algorithm," in *2018 3rd International Conference on Inventive Computation Technologies (ICICT)*, 2018, pp. 417–421.

[17] R. Sramek, "The Online Viterbi Algorithm," Master's thesis, Comenius University, Bratislava, 2007.

[18] E. H. Lockwood, *Book of Curves*, Cambridge University Press, 1961.

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, 2 edition, 2009.

[20] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Pearson Education Inc., Upper Saddle River, New Jersey, 2003.

[21] S.Shalev Shwartz and S. Ben-David, *Understanding Machine Learning from Theory to Algorithms*, Cambridge University Press, New York, USA, 2014.

[22] R. Szeliski, *Computer Vision Algorithms and Applications*, Springer London Ltd, London, 2010.

[23] G. H. Givens and J. A. Hoeting, *Computational statistics*, John Wiley & Sons, Hoboken, New Jersey, 2013.

[24] N. Chopin and O. Papaspiliopoulos, *An Introduction to Sequential Monte Carlo*, Springer, Gewerbestrasse 11, 6330 Cham, Switzerland, 2020.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[26] Inc. CVX Research, "CVX: Matlab Software for Disciplined Convex Programming, version 2.0 beta," `http://cvxr.com/cvx`, Sept. 2012.

[27] L. Källberg, *Minimum Enclosing Balls and Ellipsoids in General Dimensions*, Ph.D. thesis, Mälardalens högskola, 2019.

[28] W. Aldosari and M. Zohdy, "Tracking a jammer in wireless sensor networks and selecting boundary nodes by estimating signal-to-noise ratios and using an extended kalman filter," *Journal of Sensor and Actuator Networks*, vol. 7, no. 4, 2018.

[29] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, 1993.

[30] H. Ardö, K. Åström, and R. Berthilsson, "Real time viterbi optimization of hidden markov models for multi target tracking," in *IEEE Workshop on Motion and Video Computing (WMVC'07)*, 2007.

[31] A. Kräußling, *Tracking Multiple Objects Using the Viterbi Algorithm*, pp. 233–247, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[32] R. Bellman, "On the theory of dynamic programming," in *Proceedings of the National Academy of Sciences of the United States of America*, 1952, vol. 38.

[33] R. E. Kalman, ""On the General Theory of Control Systems"," in *Proc. First IFAC Congress*, Moscow, 1960, vol. 1, pp. 481–492.

[34] S. Särkkä, *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013.