Multi-target detection using Relevance Vector Machine

MSc Thesis

Sumeet Sharma
(4482875)

TU Delft

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

Multi-target detection using Relevance Vector Machine

MSc Thesis Report
by Sumeet Sharma

Dissertation to obtain the degree of Master of Science in Electrical Engineering
to be defended publicly on 23/02/2023

Thesis examination committee

Prof. DSc. A. Yarovoy (TU Delft)
Dr. H. Driessen (TU Delft)
Dr. R.Hendriks (TU Delft)

Abstract

The main purpose of a radar is to detect, recognize, and track objects of interest. When it is known that only a single target is present, the matched filter is proven to be optimal detector. However, in practice, a radar scene often consists of multiple targets. For example, in air surveillance and monitoring applications, multiple aircrafts might be in the airspace. When multiple targets are to be detected the matched filter is not guaranteed to give the best results. This can happen when a strong reflector masks the signals reflected from weak reflectors, thereby resulting in missed detections. Furthermore, when the sensor resolution is low, targets that are spaced closely together may only result in a single target actually being detected.

This research explores how the Relevance Vector Machine (RVM) framework may be used to achieve a better multi-target detector than the commonly used basic matched filter approach. RVM was selected to resolve the multi-target detection problem as it estimates the target locations iteratively. In this research it was shown how the RVM framework can be used to model the fluctuation of swerling I/II targets. Additionally, the RVM algorithm was modified to incorporate a notion of statistical thresholding. Simulations show that using RVM the false alarm rate can be reduced and target locations can be more accurately recovered compared to other existing methods in case of multiple swerling I/II fluctuating targets. Furthermore, the proposed approach is shown to have a much lower convergence time compared to a similar expectation-maximization based method, namely Enhanced Sparse Bayesian Learning.

Keywords: Multi-target detection, Relevance vector machine, Statistical thresholding

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to all those who have assisted me in one way or other in successful completion of this study. Firstly, I am thankful to my supervisor Dr. Hans Driessen for his excellent guidance, constructive comments and advice during weekly progress meetings and encouragement through out the research period. I learned a lot from his wide experience during this research.

Thanks are also due to Prof. Alexander Yarovoy and Dr. Richard Hendriks for their guidance and support as the Chair and member of examination committee. I am also indebted to the staff of MS3 group, Department of Microelectronics, Faculty of Electrical Engineering, Mathematics and Computer Science (TU Delft) for their direct and indirect support.

Finally, I am grateful to my family members for their understanding, encouragement, support and patience during this study.

Contents

1 Introduction

Radar technology is widely used in different sectors for detecting, recognizing and tracking objects of interest. Different techniques and approaches are available to process the data received. The matched filter is a widely used method in radar for the detection and estimation of a signal buried in noise. This works well when the targets are well separated, and the received echo can be considered a superposition of multiple distinct single targets. The problem then simplifies into multiple independent detection problems. In practice however, targets may be spaced closely together and influence each other. The basic matched filter does not take this into account, and as a result, some of the targets may go undetected in such a multitarget scenario. One way to mitigate this problem could be by using (a bank of) mismatched filters for sidelobe reduction [1]. However, this filter would have a lower processing gain and have a worse range resolution due to the wider main lobe and would thus be suboptimal. Additionally, weakly reflecting targets may still go undetected in the sidelobes of very strongly reflecting targets. An alternative approach would be to consider all possible likelihoods and to select the hypothesis that seems most likely. Such an approach would be very computationally intensive and would scale exponentially with the number of target parameters (range bin, doppler bin, angle bin). Instead, this approach may be approximated by estimating the number of targets and their parameters in an iterative scheme. The number of targets and their parameters are then updated according to what maximizes the likelihood during each iteration.

In traditional radar systems the entire scene is split into several sections. The energy sent in the direction of a particular section is limited to a certain time duration. A longer time duration means that more pulses are transmitted and therefore more measurements are obtained. This allows for the integration of the multiple measurements, which typically increases the probability of correctly detecting a target. The next time the same section is illuminated, targets in that section may have moved or rotated. Therefore, the orientations of the targets with respect to the radar could have changed which could result in a different RCS (radar cross section). These RCS fluctuations can be modelled statistically using for example

Swerling models [2]. In some radar systems, the modulation/operating frequency may change in each subsequent pulse. This would also result in a fluctuation of RCS, which can be characterized by the same Swerling models.

The Relevance Vector Machine (RVM) framework is an existing framework which iteratively determines the number of active components present and the associated parameters for each component by maximizing the likelihood. It also explicitly assumes a probabilistic model for a certain component to be active. Therefore, the RVM framework seems like a promising way to approximate a multi-target matched filter where the RCS of the targets follow certain statistics.

1.1 Research goal and objectives

The main goal of this research was to explore how the RVM framework can be used in a multitarget detection framework for radar applications. To achieve this goal, the specific objectives were

- Review of literature on multitarget detection
- Development of the algorithms for modifications for the Relevance Vector Machine
- Monte Carlo simulations to examine and compare performances of different detection algorithms using MATLAB

1.2 Outline of the thesis

This thesis is organized in 8 chapters. Chapter 2 provides background information on the matched filter, likelihood ratio tests, Swerling models, and some sparse recovery algorithms including the RVM framework. Chapter 3 gives an overview of the relationship between RVM and some common sparse recovery algorithms. Attempts to modify/improve on the standard RVM framwork are discussed in chapter 4. Chapter 5 describes the extension of the RVM framework for multiple measurement vectors. A comparison of the different algorithms discussed throughout the report is given in chapters 6 and 7. Finally, chapter 8 provides key conclusions of this research and makes some recommendations for future work.

2 Background

This chapter describes the background knowledge and fundamentals behind detection theory, Swerling fluctuation models and several sparse recovery algorithms. The first section explains the basic signal/measurement model that was used throughout this entire research. The second section presents the framework for a simple single target detection problem. Building upon this, the third section highlights the matched filter involving unknown parameters. The next section elaborates on how the matched filter may be extended for use in multitarget problems. Following this, the next two sections describe the matching pursuit (MP) and orthogonal matching pursuit (OMP) respectively. This is followed by an outline of Swerling fluctuation models. Finally, the last section gives an overview of the relevance vector machine (RVM).

2.1 Basic signal model

In practice, signals received from an observed scene by a radar are often high-dimensional. This leads to a large data-stream that has to be processed. However, the received signal generally only contains a small number of parameters of interest. Thus, it can be said that the received signal is sparse or compressible in a certain domain. Examples of commonly used sparse domains for radar are given in [3].

In this study, we specifically focus on the case where targets in the scene can be modelled as point targets. For simplicity we consider the received signal to be real instead of complex. The received signal $\mathbf{t}$ can then be expressed as a linear equation [4], namely

$$\mathbf{t} = \sum_{k=1}^{K} \boldsymbol{\phi}(\tau_k) w_k + \boldsymbol{\epsilon} \tag{2.1}$$

where $\mathbf{t} \in \mathcal{R}^{N\times 1}$, $\boldsymbol{\phi}(\tau_k) \in \mathcal{R}^{N\times 1}$, $\boldsymbol{\epsilon} \in \mathcal{R}^{N\times 1}$ and $\mathcal{R}$ represents the real coordinate space. When considering range, $\phi(\tau_k)$ may represent the response that arises due to a target at a range corresponding to delay τ_k. Similarly, during angle estimation

$\phi(\tau_k)$ may represent the steering vector corresponding to an angle τ_k [5]. K denotes the number of targets that are present in the observed scene. The noise/error term $\boldsymbol{\epsilon}$ is assumed to be normally distributed according to $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$,

Equation 2.1 does not have any constraints for the values of τ_k; its value can be anywhere on the continuous real number space. One approach to solve this is by reformulating the problem on a discretized grid wih a sparsity assumption. Assuming that the targets can only lie exactly on a grid point, equation 2.1 can then be written as

$$\mathbf{t} = \boldsymbol{\Phi}\mathbf{w} + \boldsymbol{\epsilon} \tag{2.2}$$

where $\boldsymbol{\Phi} \in \mathcal{R}^{M \times N}$ and $\mathbf{w} \in \mathcal{R}^{M \times 1}$. Here, $\mathbf{w}$ should be K-sparse: it contains only K non-zero entries. The problem that arises is then to recover $\mathbf{w}$ from the measurement $\mathbf{t}$.

2.2 Single target detection

The single target detection problem can be formulated as a decision problem where the goal is to decide which hypothesis $\mathcal{H}$ is most likely to have generated the measurement $\mathbf{t}$. Suppose that $\boldsymbol{\epsilon}$ in equation 2.2 is distributed according to $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ where $\mathbf{C}$ is a known covariance matrix. Assume additionally that τ and w are known and deterministic. Let $p(\mathbf{t}; \mathcal{H}_0)$ be the distribution when no target is present and $p(\mathbf{t}; \mathcal{H}_1)$ be the distribution when a target is present. Under the null hypothesis $\mathcal{H}_0$, the measurement $\mathbf{t}$ only consists of the noise $\boldsymbol{\epsilon}$, which means that $\mathbf{t} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$. Under the alternative hypothesis $\mathcal{H}_1$ the reflection due to the presence of a target now results in a non-zero mean. Since w is assumed to be deterministic, the variance of the measurement is not affected under $\mathcal{H}_1$. Thus, for this hypothesis, $\mathbf{t} \sim \mathcal{N}(\boldsymbol{\phi}_\tau w_\tau, \mathbf{C})$. Considering all vectors and matrices to be real-valued, the likelihoods under each hypothesis are then given by

$$p(\mathbf{t}; \mathcal{H}_0) = \frac{1}{\sqrt{|2\pi\mathbf{C}|}} \exp(-\frac{1}{2}\mathbf{t}^T\mathbf{C}^{-1}\mathbf{t}) \tag{2.3}$$

$$p(\mathbf{t};\mathcal{H}_1) = \frac{1}{\sqrt{|2\pi\mathbf{C}|}}\exp(-\frac{1}{2}(\mathbf{t}-\boldsymbol{\phi}_\tau w)^T\mathbf{C}^{-1}(\mathbf{t}-\boldsymbol{\phi}_\tau w)) \tag{2.4}$$

The following likelihood ratio test can then be defined

$$L(\mathbf{t}) = \frac{p(\mathbf{t};\mathcal{H}_1)}{p(\mathbf{t};\mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \theta \tag{2.5}$$

A hypothesis is then selected based on the value of this likelihood ratio test. In case likelihood ratio exceeds a certain threshold θ, the alternative hypothesis $\mathcal{H}_1$ is selected. Otherwise, $\mathcal{H}_0$ is selected.

To make computations easier, the log of the likelihood ratio is often taken instead of the likelihood ratio itself. Using equations 2.3-2.4, the log likelihood ratio test can then be found to be

$$\begin{aligned}
&\ln L(\mathbf{t}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \ln(\theta) \\
&\ln(p(\mathbf{t};\mathcal{H}_1)) - \ln(p(\mathbf{t};\mathcal{H}_0)) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \ln(\theta) \\
&-\frac{1}{2}\ln|2\pi\mathbf{C}| - \frac{1}{2}(\mathbf{t}-\boldsymbol{\phi}_\tau w)^T\mathbf{C}^{-1}(\mathbf{t}-\boldsymbol{\phi}_\tau w) + \frac{1}{2}\ln|2\pi\mathbf{C}| + \frac{1}{2}\mathbf{t}^T\mathbf{C}^{-1}\mathbf{t} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \ln(\theta) \\
&\mathbf{t}^{\mathbf{T}}\mathbf{C}^{-1}\boldsymbol{\phi}_\tau w - \frac{1}{2}w\boldsymbol{\phi}_\tau^T\mathbf{C}^{-1}\boldsymbol{\phi}_\tau w \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \ln\theta
\end{aligned} \tag{2.6}$$

If $\mathbf{C}$, $\boldsymbol{\phi}_\tau$ and w are known a priori, only the first term of equation 2.6 is dependent on the measurement $\mathbf{t}$. Thus, whether the null hypothesis is rejected depends on the magnitude of $\mathbf{t}^{\mathbf{T}}\mathbf{C}^{-1}\boldsymbol{\phi}_\tau w$. This type of detector can be interpreted as something that measures the ”match” between between the template signal $\boldsymbol{\phi}_\tau w$ and the whitened measurement vector $\mathbf{C}^{-1}\mathbf{t}$.

For white/uncorrelated noise we have $\mathbf{C} = \sigma^2\mathbf{I}$. The likelihoods under hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ are then given by

$$p(\mathbf{t};\mathcal{H}_0) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp(-\frac{1}{2\sigma^2}\mathbf{t}^T\mathbf{t}) \tag{2.7}$$

$$p(\mathbf{t};\mathcal{H}_1) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp(-\frac{1}{2\sigma^2}(\mathbf{t}-\boldsymbol{\phi}_\tau w)^T(\mathbf{t}-\boldsymbol{\phi}_\tau w)) \tag{2.8}$$

Following the same process as in equation 2.6 the loglikelihood ratio test for the white noise case can be found to be

$$\ln L(\mathbf{t}) = \frac{\mathbf{t}^{\mathbf{T}}\boldsymbol{\phi}_\tau w}{\sigma^2} - \frac{w\boldsymbol{\phi}_\tau^T\boldsymbol{\phi}_\tau w}{2\sigma^2} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \ln\theta \tag{2.9}$$

2.3 The matched filter

In most practical radar detection problems the location and reflectivity of a target are not known beforehand. This means that τ and w are unknown and hence the loglikelihood ratio tests described in equations 2.6 and 2.9 can not directly be used. Such detection problems involving more than one unknown parameters are also known as composite hypothesis problems. There exist two main (suboptimal) approaches for solving such composite hypothesis problems. If there is some prior knowledge about any of the unknown parameters, a so-called Bayesian likelihood ratio can be formed by marginalizing over the unknown parameters for each hypothesis [6]. This method will yield the optimal result in case the prior knowledge is actually correct. Another approach would be by obtaining the maximum likelihood estimates for all of the unknown parameters for each hypothesis and plugging these values into the likelihood ratio test. The test formed using this latter approach is also known as the Generalized Likelihood Ratio Test (GLRT) [7]. For the single target detection problem described in section 2.2 the GLRT is given by

$$\frac{p(\mathbf{t},\hat{w},\hat{\tau};\mathcal{H}_1)}{p(\mathbf{t};\mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \theta \tag{2.10}$$

where $\hat{w}$ and $\hat{\tau}$ are the maximum likelihood estimates for w and τ.

These maximum likelihood values for the unknown parameters can be found by setting the derivative w.r.t the unknown parameter to 0. As the $\log$ function is monotonic, this is equivalent to setting the derivative of the loglikelihood w.r.t the parameter to 0. Doing this for equations 2.4 and 2.8 we get the following:

$$\begin{aligned}\hat{\tau} &= \arg\max_{\tau} = \mathbf{t}^T\mathbf{C}^{-1}\boldsymbol{\phi}_\tau(\boldsymbol{\phi}_\tau^T\mathbf{C}^{-1}\boldsymbol{\phi}_\tau)^{-1}\boldsymbol{\phi}_\tau^T\mathbf{C}^{-1}\mathbf{t}\\ \hat{w} &= (\boldsymbol{\phi}_{\hat{\tau}}^T\mathbf{C}^{-1}\boldsymbol{\phi}_{\hat{\tau}})^{-1}\boldsymbol{\phi}_{\hat{\tau}}^T\mathbf{C}^{-1}\mathbf{t}\end{aligned} \tag{2.11}$$

and

$$\begin{aligned}\hat{\tau} &= \arg\max_{\tau} = \mathbf{t}^T\boldsymbol{\phi}_\tau(\boldsymbol{\phi}_\tau^T\boldsymbol{\phi}_\tau)^{-1}\boldsymbol{\phi}_\tau^T\mathbf{t}\\ \hat{w} &= (\boldsymbol{\phi}_{\hat{\tau}}^T\boldsymbol{\phi}_{\hat{\tau}})^{-1}\boldsymbol{\phi}_{\hat{\tau}}^T\mathbf{t}\end{aligned} \tag{2.12}$$

respectively.

The expression given by equation 2.6 produces different values depending on the exact shape of $\boldsymbol{\phi}_\tau$ and the corresponding amplitude w. We can normalize and rewrite the expression to

$$\frac{\mathbf{t}^T\mathbf{C}^{-1}\boldsymbol{\phi}_\tau}{\sqrt{\boldsymbol{\phi}_\tau^T\mathbf{C}^{-1}\boldsymbol{\phi}_\tau}} > \theta' \tag{2.13}$$

The probability of detection P_d and the probability of false alarm P_{fa} for the detector proposed in equation 2.13 are given by

$$\begin{aligned}P_d &= Q(\theta' - w\sqrt{\boldsymbol{\phi}_\tau^T\mathbf{C}^{-1}\boldsymbol{\phi}_\tau})\\ P_{fa} &= Q(\theta')\end{aligned} \tag{2.14}$$

where Q denotes the right-tail normal cumulative distribution function. The detector described in equations 2.6-2.13 maximizes the probability of detection for a given probability of false alarm and are considered optimal according to the Neyman-Pearson criterion [8].

2.4 Multitarget usage of matched filter

The matched filter is known to be the optimal detector for a single target with known impulse response in white noise. One way of going about the multi-target problem is by extending the maximum likelihood estimates $\hat{w}$ in equations 2.11 and 2.12 to include estimates for all possible weights $\mathbf{w}$ at once. Thus, we get

$$\begin{aligned}\hat{\boldsymbol{\tau}} =& diag((\boldsymbol{\Phi}^T\mathbf{C}^{-1}\boldsymbol{\Phi}))^{-1}\boldsymbol{\Phi}^T\mathbf{C}^{-1}\mathbf{t} \\ =& \begin{bmatrix} \boldsymbol{\Phi}_1^T\mathbf{C}^{-1}\boldsymbol{\Phi}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{\Phi}_2^T\mathbf{C}^{-1}\boldsymbol{\Phi}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{\Phi}_M^T\mathbf{C}^{-1}\boldsymbol{\Phi}_M \end{bmatrix}^{-1} \begin{bmatrix} |\boldsymbol{\Phi}_1^T\mathbf{C}^{-1}\mathbf{t}|^2 \\ |\boldsymbol{\Phi}_2^T\mathbf{C}^{-1}\mathbf{t}|^2 \\ \vdots \\ |\boldsymbol{\Phi}_M^T\mathbf{t}|^2 \end{bmatrix}\end{aligned} \tag{2.15}$$

and

$$\begin{aligned}\hat{\mathbf{w}} =& (diag(\boldsymbol{\Phi}^T\mathbf{C}^{-1}\boldsymbol{\Phi}))^{-1}\boldsymbol{\Phi}^T\mathbf{C}^{-1}\mathbf{t} \\ =& \begin{bmatrix} \boldsymbol{\Phi}_1^T\mathbf{C}^{-1}\boldsymbol{\Phi}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{\Phi}_2^T\mathbf{C}^{-1}\boldsymbol{\Phi}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{\Phi}_M^T\mathbf{C}^{-1}\boldsymbol{\Phi}_M \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}_1^T\mathbf{C}^{-1}\mathbf{t} \\ \boldsymbol{\Phi}_2^T\mathbf{C}^{-1}\mathbf{t} \\ \vdots \\ \boldsymbol{\Phi}_M^T\mathbf{C}^{-1}\mathbf{t} \end{bmatrix}\end{aligned} \tag{2.16}$$

where each $\boldsymbol{\Phi}_i$ corresponds to $\boldsymbol{\phi}$ centered at τ_i. Subsequently, we can filter out the $\hat{w}_i$'s for which the corresponding $\hat{\tau}_i$'s are below a certain threshold to obtain a sparse solution. It is important to note that in this multitarget matched filter approach the entries in the matrices in equations 2.15 and 2.16 that are not on the main diagonal are set to 0. This implies that $\boldsymbol{\Phi}_i^T\mathbf{C}^{-1}\boldsymbol{\Phi}_j$ should be negligible when $i \neq j$. The major downside of this method is that each entry of $\hat{w}$ is estimated independent from its other entries; the influence that a certain estimate $\hat{w}_i$ may have on $\hat{w}_j$ is not taken into account. Therefore, while the multi-target matched filter might be a suitable approach for targets that are spaced sufficiently far apart such that $\boldsymbol{\Phi}_i^T\mathbf{C}^{-1}\boldsymbol{\Phi}_j$ is sufficiently small, it will likely fail for targets that are spaced closely to each other. Instead, for more accurate target detection for such cases an iterative solution is required.

Similarly, for uncorrelated noise ($\mathbf{C} = \sigma^2\mathbf{I}$), the expressions in equations 2.15 and 2.16 simplify to

$$\begin{aligned}\hat{\boldsymbol{\tau}} &= diag((\mathbf{\Phi}^T\mathbf{\Phi}))^{-1}\mathbf{\Phi}^T\mathbf{t} \\ &= \begin{bmatrix} \mathbf{\Phi}_1^T\mathbf{\Phi}_1 & 0 & \dots & 0 \\ 0 & \mathbf{\Phi}_2^T\mathbf{\Phi}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{\Phi}_M^T\mathbf{\Phi}_M \end{bmatrix}^{-1} \begin{bmatrix} |\mathbf{\Phi}_1^T\mathbf{t}|^2 \\ |\mathbf{\Phi}_2^T\mathbf{t}|^2 \\ \vdots \\ |\mathbf{\Phi}_M^T\mathbf{t}|^2 \end{bmatrix}\end{aligned} \tag{2.17}$$

and

$$\begin{aligned}\hat{\mathbf{w}} &= diag((\mathbf{\Phi}^T\mathbf{\Phi}))^{-1}\mathbf{\Phi}^T\mathbf{t} \\ &= \begin{bmatrix} \mathbf{\Phi}_1^T\mathbf{\Phi}_1 & 0 & \dots & 0 \\ 0 & \mathbf{\Phi}_2^T\mathbf{\Phi}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{\Phi}_M^T\mathbf{\Phi}_M \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Phi}_1^T\mathbf{t} \\ \mathbf{\Phi}_2^T\mathbf{t} \\ \vdots \\ \mathbf{\Phi}_M^T\mathbf{t} \end{bmatrix}\end{aligned} \tag{2.18}$$

Here, the term $\sigma^2\mathbf{I}$ cancels out and is no longer present in the expressions. Hence, the model is only valid when $\mathbf{\Phi}_i^T\mathbf{\Phi}_j$ is negligble for $i \neq j$.

2.5 Matching pursuit

The matching pursuit algorithm was first introduced in [9] as a way of decomposing a signal into structures. It attempts to obtain the sparse solution for $\hat{\mathbf{w}}$ iteratively. In each iteration, the algorithm computes the projection for each $\mathbf{\Phi}_i$ onto the residual. For the first iteration, this residual is equal to the measurement itself. After this step, the component i which has the largest projection onto the residual is selected. Using the corresponding basis function $\mathbf{\Phi}_i$, an estimate is made for $\hat{w}_i$. The estimated contribution $\mathbf{\Phi}_i\ \hat{w}_i$ is then subtracted from the residual. This process is repeated as long as the projection onto the residual is above a certain predefined threshold θ. The pseudocode for the matching pursuit algorithm is given in algorithm 1.

Algorithm 1 Matching pursuit

```
$\mathbf{w} \leftarrow 0$
$\mathbf{r} \leftarrow \mathbf{t}$
while $\|\mathbf{r}\|_2 \geq \theta$ do
    $\mathbf{p} \leftarrow (\mathbf{\Phi}^T \mathbf{r})$
    $k \leftarrow \arg\max_{M \geq i \geq 1}(\|p_i\|_2)$
    $w_k = w_k + p_k$
    $\mathbf{r} = \mathbf{r} - p_k \mathbf{\Phi}_k$
end while
```

Upon completion, the matching pursuit algorithm will have decomposed the signal/measurement $\mathbf{t}$ into the strongest components present. As the algorithm selects one i in each iteration, the number of selected components will be equal to the number of iterations.

2.6 Orthogonal Matching Pursuit

The Orthogonal Matching Pursuit (OMP) algorithm is an algorithm that improves on some of the limitations of Matching Pursuit. For example, in Matching Pursuit, it may happen that the same component i is selected more than once. Furthermore, the estimation of a new weight w_i may have an impact the weight of a neighbouring cell which was selected and computed in a previous iteration. OMP overcomes most of these issues by obtaining $\hat{\mathbf{w}}$ through a least squares estimation in each iteration. Because of this, if the support of the true underlying signal is estimated correctly, OMP is able to correctly identify the number of targets and accurately reconstruct the signal with a much higher probability [10]. The main disadvantage of OMP compared to MP is that OMP requires a pseudo-inverse to be calculated in each step, and is therefore much more computationally intensive than MP. The pseudocode for OMP can be seen in algorithm 2.

Algorithm 2 Orthogonal Matching Pursuit

```
$\mathbf{w} \leftarrow 0$
$\mathbf{r} \leftarrow \mathbf{t}$
$k \leftarrow 0$
$\Lambda = \emptyset$
**while** $\|\mathbf{r}\|_2 \geq \theta$ **do**
    $\mathbf{p} \leftarrow (\mathbf{\Phi}^T \mathbf{r})$
    $\tau \leftarrow \arg\max_{M \geq i \geq 1}(\|p_i\|_2)$
    $\Lambda \leftarrow \Lambda \cup \tau$
    $\mathbf{w} \leftarrow (\mathbf{\Phi}_\Lambda^T \mathbf{\Phi}_\Lambda)^{-1} \mathbf{\Phi}_\Lambda^T \mathbf{t}$
    $\mathbf{r} \leftarrow \mathbf{t} - \mathbf{\Phi}_\Lambda \mathbf{w}$
**end while**
```

2.7 Radar Cross Section and the Swerling models

The power reflected from a target is typically characterized by the Radar Cross Section (RCS). It is defined as

$$\text{RCS} = 4\pi \frac{\text{power reflected towards the source per unit angle}}{\text{incident power density}} \tag{2.19}$$

The RCS of a target depends on various factors such as its electrical properties, its orientation relative to the radar, the radar's transmitting frequency and its shape and features [2]. Because of this, the RCS of a target generally shows some fluctuation in several consecutive measurements. For simple targets it is possible to explicitly calculate the dependence of the targets' RCS on these parameters. For complex real-world targets, however, this is not the case.

The Swerling models are a set of 5 simple models that are commonly used to approximate the RCS of a complex target [2]. These models distinguish the RCS of a reflecting target based on probability distributions and RCS decorrelation time.

The Swerling 0/V model models the ideal case that a target's RCS remains constant. In the Swerling I model, the RCS is assumed to remain constant within a burst of pulses, but varies from scan to scan. The complex RCS can then be modelled with a chi-squared distribution with 2 degrees of freedom (Rayleigh distribution) , which means that the I- and Q-channels can be assumed to follow a zero-mean Gaussian distribution. This model generally applies to (large) targets which consist of a large number of scattering surfaces. The Swerling II model assumes the same probability distribution as the Swerling I model, only now the RCS varies from pulse to pulse as well. The Swerling III model is similar to the Swerling I model, in that the RCS remains constant from pulse to pulse. However, the RCS now follows a chi-squared distribution with 4 degrees of freedom. It represents the sum of the reflection of a dominant scatterer plus many smaller scatterers. Finally, the Swerling IV model is for similar targets to the Swerling III model but where the RCS also varies from pulse to pulse.

In this research, the focus was real-valued RCS. The formulation for complex-valued RCS can be found by extending equations 2.20-2.22 to complex-valued numbers. The detection performance is expected to be better for complex-valued RCS as they can be seen as two independent real-valued RCS.

2.8 The Relevance Vector Machine

In the RVM framework, a probabilistic model is set over the weights $\mathbf{w}$. Specifically, each w_i is assumed to be zero-mean with variance α_i^{-1} (note: this term must always be positive). Tipping's original paper [11] involved an iterative method to find the relevant basis functions. In that paper, the updates for the estimates for α and σ^2 were derived through expectation maximization and converged rather slowly. In his other paper [12], it is suggested to sequentially add/delete basis functions. This significantly speeds up the process. The main idea behind the second paper is that the marginal (log)likelihood can be split up into an expression that depends on α_i, and an expression that does not explicitly depend on α_i. Thus, by maximizing this first expression w.r.t. α_i at each iteration, the estimates for the α_i's are improved. Specifically, the marginal loglikelihood is given by equation 2.20

$$L(\boldsymbol{\alpha}) = -\frac{1}{2}[N \log(2\pi) + \log |\mathbf{C}| + \mathbf{t}^T \mathbf{C}^{-1} \mathbf{t}] \tag{2.20}$$

where $\mathbf{C} = \sigma^2\mathbf{I} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^T$. Here, $\mathbf{A}$ is a diagonal matrix consisting of all the α_i's.

From the definition of $\mathbf{C}$ above it follows that the covariance matrix $\mathbf{C}$ can be decomposed into a term that depends on the i th component, and a term that doesn't as follows:

$$\mathbf{C} = \mathbf{C}_{-i} + \alpha_i^{-1}\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i \tag{2.21}$$

After plugging this back into equation 2.20 and after simplifying the following expression is obtained:

$$L(\boldsymbol{\alpha}) = L(\boldsymbol{\alpha}_{-i}) + \frac{1}{2}[log(\alpha_i) - log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}] \tag{2.22}$$

where

$$s_i = \boldsymbol{\phi}_i^T\mathbf{C}_{-i}^{-1}\boldsymbol{\phi}_i \tag{2.23}$$

$$q_i = \boldsymbol{\phi}_i^T\mathbf{C}_{-i}^{-1}\mathbf{t} \tag{2.24}$$

The term q_i provides an indication of how well $\boldsymbol{\phi}_i$ increases $L(\boldsymbol{\alpha})$ by helping to explain the data, while the term s_i provides a measure of how much the inclusion of ϕ_i serves to decrease $L(\boldsymbol{\alpha})$ through 'inflating' C and hence adding to the normalising factor. Taking the first and second derivative of $l(\alpha_i)$ shows that $L(\alpha)$ has a unique maximum with respect to α_i:

$$\begin{aligned} \alpha_i &= \frac{s_i^2}{q_i^2 - s_i} \quad if \quad \frac{q_i^2}{s_i} > 1 \\ \alpha_i &= \infty \quad if \quad \frac{q_i^2}{s_i} \leq 1 \end{aligned} \tag{2.25}$$

Plugging this value back into $l(\alpha_i)$ we obtain:

$$\begin{aligned} l(\alpha_i) &= \frac{q_i^2}{s_i} - log(\frac{q_i^2}{s_i}) - 1 \quad if \quad \frac{q_i^2}{s_i} > 1 \\ l(\alpha_i) &= 0 \qquad\quad if \quad \frac{q_i^2}{s_i} \leq 1 \end{aligned} \tag{2.26}$$

Therefore, the α_i's corresponding to components that have a $\frac{q_i^2}{s_i} \geq 1$ are included in the model.

The estimated α_i's can then be used to find the posterior estimate for $\mathbf{w}$. Specifically, we then have $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{t} \tag{2.27}$$

$$\boldsymbol{\Sigma} = (\mathbf{A} + \sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi}) \tag{2.28}$$

In case the noise variance σ^2 is not known, an estimate is provided by

$$\hat{\sigma}^2 = \frac{(\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu})^T(\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu})}{N - M + \sum_m \alpha_m \boldsymbol{\Sigma}_{mm}} \tag{2.29}$$

where $\boldsymbol{\Sigma}_{mm}$ is the entry of the mth row and mth column of $\boldsymbol{\Sigma}$. This estimate can be derived using an expectation maximization approach [11].

During each iteration of RVM, for each α_i, 3 types of actions are considered: adding, deleting and re-estimating. The adding action refers to adding a new α_i to the model and is only applicable to the components that are not yet in the model which exceed the threshold described in equation 2.25. Executing this action would be equivalent to including i in the support of the reconstructed signal. The deleting action refers to removing an α_i that is already present in the model which is beneath the threshold of equation 2.25, and thus corresponds to removing the corresponding bin i from the reconstructed signal. Finally, the re-estimating action refers to updating α_i for the components that exceed the threshold in equation 2.25 but which are already present in the model. After evaluating the change in marginal loglikelihood for each of these quations for each α_i, the action that provides the largest increase in marginal likelihood is then executed. A schematic overview of the RVM algorithm is provided in figure 1.

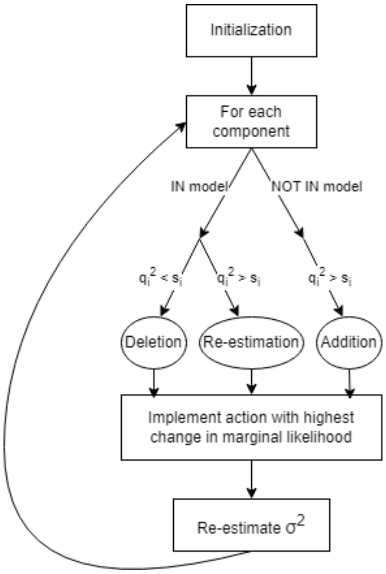

Figure 1: Schematic overview of the relevance vector machine algorithm

The RVM algorithm meets its stopping criteria once no more potential actions indicate an increase in the marginal loglikelihood, or when this increase is negligible. In case the noise variance is unknown, the change in noise variance must be beneath a certain threshold as an additional stopping criteria.

As described in section 2.3, there are two main ways for solving composite hypothesis problems, namely marginalizing out the unknown variables or by plugging in the maximum likelihood estimates for the unknown parameters/variables. In the RVM framework, $\mathbf{w}$ is marginalized out while α is estimated using maximum likelihood estimates. Hence, the RVM framework can be considered a hybrid between these two main approaches for composite hypothesis testing.

3 Multi-target detection

This chapter explores the sparse signal recovery algorithms discussed in section 2 more in depth for a multi-target setting. First, the relationship between the MP and OMP algorithms is described in more detail. Then, it is shown that the maximum likelihood estimate for the variance corresponds to [12]. Finally, the relationship between RVM and OMP, and their advantages and disadvantages are discussed.

3.1 Relation between MP and OMP

In subsection 2.2 it was mentioned that an iterative approach is required for multi-target detection. A greedy way of doing this could be by testing whether an additional target is present during each iteration [13]. For example, in the first iteration, the hypotheses to be tested would be

$$\begin{aligned} \mathcal{H}_1 &: \quad \text{One target present} \\ \mathcal{H}_0 &: \quad \text{No targets present} \end{aligned} \tag{3.1}$$

The maximum likelihood estimate for the location and the amplitude of the target would then be given by equation 2.12. It can be seen that these expressions correspond to the solution provided by the Matching Pursuit algorithm in the first iteration for normalized basis vectors. Similarly, in the next iteration, the two hypothesis to be tested would be

$$\begin{aligned} \mathcal{H}_2 &: \quad \text{Two targets present} \\ \mathcal{H}_1 &: \quad \text{One target present} \end{aligned} \tag{3.2}$$

Note the two new hypotheses implicitly assume that the target detected in the previous step is definitely present in the measurement. Under the assumption that the parameters (location, amplitude) of the first target were correctly detected, the response due to the first target could be substracted from the measurement, and the resulting problem could be reformulated as in hypothesis 3.1. Such an iterative approach would be identical to the MP algorithm. The OMP algorithm can be seen as an extension to the same solution approach where we assume the the estimated amplitude w_i in each iteration might contain some error, while the location of the target (the support of $\mathbf{w}$) is assumed to be correctly estimated. This error may be for instance due to the presence of noise.

3.2 Derivation maximum likelihood for variance

Consider having the following two hypothesis

$$\begin{aligned} \mathcal{H}_0 : \qquad & \mathbf{t} = \boldsymbol{\epsilon} \\ \mathcal{H}_1 : \qquad & \mathbf{t} = w_1 \boldsymbol{\phi}_\tau + \boldsymbol{\epsilon} \end{aligned} \tag{3.3}$$

where $\boldsymbol{\Phi}_\tau$ is known but w_1 is unknown and Gaussian distributed with zero mean and variance α_1^{-1}, and $\boldsymbol{\epsilon}$ is Gaussian distributed with zero mean and known covariance matrix $\mathbf{C}_0$. The covariance matrix of $\mathbf{t}$ for $\mathcal{H}_1$ is then given by $\mathbf{C}_1 = \mathbf{C}_0 + \alpha_1^{-1} \boldsymbol{\phi}_\tau \boldsymbol{\phi}_\tau^T$ .Using properties of the determinant and the woodbury identity we can then derive the following:

$$|\mathbf{C}_1| = |\mathbf{C}_0|(\alpha_1^{-1} \boldsymbol{\Phi}_\tau^T \mathbf{C}_0^{-1} \boldsymbol{\Phi}_\tau + 1) \tag{3.4}$$

$$\mathbf{C}_1^{-1} = \mathbf{C}_0^{-1} - \mathbf{C}_0^{-1} \boldsymbol{\Phi}_\tau (\boldsymbol{\Phi}_\tau^T \mathbf{C}_0^{-1} \boldsymbol{\Phi}_\tau + \alpha_1) \boldsymbol{\Phi}_\tau \mathbf{C}_0^{-1} \tag{3.5}$$

$$\mathbf{t}^T \mathbf{C}_1^{-1} \mathbf{t} = \mathbf{t}^T \mathbf{C}_0^{-1} \mathbf{t} - \frac{|\boldsymbol{\Phi}_\tau^T \mathbf{C}_0^{-1} \mathbf{t}|^2}{\boldsymbol{\Phi}_\tau^T \mathbf{C}_0^{-1} \boldsymbol{\Phi}_\tau + \alpha_1} \tag{3.6}$$

The loglikelihood under $\mathcal{H}_1$ is then given by

$$\begin{aligned} \log(p(\mathbf{t}; \mathcal{H}_1)) &= \log(\frac{1}{\sqrt{|2\pi \mathbf{C}_1|}} \exp(-\frac{1}{2} \mathbf{t}^T \mathbf{C}_1^{-1} \mathbf{t})) \\ &= -\frac{1}{2} \log |2\pi \mathbf{C}_1| - \frac{1}{2} \mathbf{t}^T \mathbf{C}_1^{-1} \mathbf{t} \\ &= -\frac{N}{2} \log |2\pi| - \frac{1}{2} \log(|\mathbf{C}_0|(1 + \alpha_1^{-1} s_1)) - \frac{1}{2} \mathbf{t}^T \mathbf{C}_0^{-1} \mathbf{t} + \frac{1}{2} \frac{q_1^2}{\alpha_1 + s_1} \end{aligned} \tag{3.7}$$

This term is identical to the expression for the marginal loglikelihood in [12].

After computing the first and second order derivative [14], we find that it results in the maximum likelihood estimate that was described in equation 2.25.

3.3 Relation between RVM and OMP

In [13] it is suggested that such a multitarget detector can be approximated through the Orthogonal Matching Pursuit (OMP) algorithm. However, in OMP, the support is only increased in successive iterations, and the effect of removing a component that is already present in your support is never evaluated. OMP also does not estimate the noise precision β, whereas RVM does. Another big difference between OMP and RVM is that if any prior information is known (e.g. the locations of targets estimated from a previous pulse), that information can be more easily incorporated into RVM than in OMP. In [15] it is shown that RVM generally outperforms OMP it terms of estimation accuracy. Only when the number of targets is known a priori, the performance of OMP approaches the performance of RVM. Additionally, RVM naturally allows us to model fluctuation loss of targets. This is especially effective when considering multiple measurement vectors.

In [16] a Bayesian version of OMP (BOMP) is proposed. This algorithm includes an uniform prior for each of the components present. In other words, instead of having a prior for each component w_i, it has the same prior for all components in $\mathbf{w}$. The resulting expression has a similar form to the estimates for $\boldsymbol{\mu}$ in the RVM framework. Hence, this indicates that the RVM algorithm can be seen as a Bayesian formulation of the OMP algorithm, where each component w_i has its own unique prior α_i.

4 Multi-target detection schemes based on RVM

The RVM framework discussed so far selects components only based on a positive change in marginal loglikelihood. In practice, this usually results in a relatively high fixed false alarm rate. This chapter of this report is dedicated to the modifications that were made to the RVM to include a better notion of thresholding.

4.1 Enhanced Sparse Bayesian Learning algorithm

The ESBL (Enhanced Sparse Bayesian Learning) algorithm was first proposed in [17]. It provides an alternate, but similar, framework to the RVM to solve the sparse recovery problem. Similar to [11] it uses the expectation-maximization framework to iteratively retrieve the support of the signal. This results in the algorithm having a slow convergence rate, compared to the RVM using fixed-point derivates as described in [12]. Another big difference is that whereas RVM is capable of adding and deleting components, the ESBL algorithm can only delete components. This means that the ESBL algorithm has to initialize with all α's being positive and finite, which slows down the convergence rate even more. One of the main contributions of [17] is that a notion of statistical thresholding is used to determine whether a component is relevant or not. In the next section it is shown how this may be used to introduce a notion of statistical thresholding for the RVM.

4.2 Statistical thresholding

The basic RVM algorithm generally results in a high number of false alarms. In practice, we would like to minimize the number of false alarms w.r.t. the number of correct detections. The derivation for the false alarm detector can be done in a similar approach as in [17]. Let k^{-1} define the factor that was used for thresholding in [17]. Rewriting this factor to Tipping's terms, we have:

$$k^{-1} = 1 - \alpha_i^{-1} \boldsymbol{\phi}_i^T \mathbf{C}^{-1} \boldsymbol{\phi}_i \tag{4.1}$$

Using the relationship between $\mathbf{C}$ and $\mathbf{C}_{-i}$ we can simplify the following:

$$\begin{aligned}\boldsymbol{\phi}_i^T\mathbf{C}^{-1}\boldsymbol{\phi}_i &= \boldsymbol{\phi}_i^T\mathbf{C}_{-i}^{-1}\boldsymbol{\phi}_i - \frac{\boldsymbol{\phi}_i^T\mathbf{C}_{-i}^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^T\mathbf{C}_{-i}^{-1}\boldsymbol{\phi}_i}{\alpha_i + \boldsymbol{\phi}_i^T\mathbf{C}_{-i}^{-1}\boldsymbol{\phi}_i} \\ &= s_i - \frac{s_i^2}{\alpha_i + s_i} \\ &= \frac{\alpha_i s_i}{\alpha_i + s_i}\end{aligned} \tag{4.2}$$

The expression for k^{-1} is then given by

$$k^{-1} = 1 - \frac{s_i}{\alpha_i + s_i} = \frac{\alpha_i}{\alpha_i + s_i} \tag{4.3}$$

For the components that are not considered relevant, we have $\alpha_i = \infty$ which results in $k = 1$. For the components that are considered relevant, we have $\alpha_i = s_i^2/(q_i^2 - s_i)$. For these relevant components we can rewrite the following:

$$\begin{aligned}k^{-1} =& \frac{\alpha_i}{\alpha_i + s_i} \\ =& \frac{s_i^2}{q_i^2 - s_i}/(\frac{s_i^2}{q_i^2 - s_i} + s_i) \\ =& \frac{s_i^2}{q_i^2 - s_i}/\frac{s_i^2 + s_i q_i^2 - s_i^2}{q_i^2 - s_i} \\ =& \frac{s_i^2}{s_i q_i^2} \\ =& \frac{s_i}{q_i^2}\end{aligned} \tag{4.4}$$

Thus, $k = \frac{q_i^2}{s_i}$.

In the iterative SBL paper [17] it is shown that $0 < k^{-1} \leq 1$. This would mean that $1 \leq k$. This implies that all of the components selected by the iterative SBL algorithm satisfy Tipping's test ($s_i \leq q_i^2$), and thus increase the marginal loglikelihood.

Similarly to [17], a threshold can be selected based on a fixed false alarm rate:

$$\theta = F_{\chi_1^2}^{-1}(1 - P_{FA}) \tag{4.5}$$

where χ_1^2 indicates the chi-square distribution with one degree of freedom.
The probability of detection for a certain component is then given by

$$P_d = 1 - F_{\chi_1^2}(k^{-1}\theta) \tag{4.6}$$

As the strength of a component (k) increases, $F_{\chi_{2D}^2}(k^{-1}\theta)$ approaches 0, and thus P_d approaches 1. It is important to note that these probabilities only hold when the estimate for α is accurate. In practice, for a single realization, this estimate will not be accurate and therefore the performance of the detector will be worse. For multiple measurement vectors, which are considered in section 5, it is expected that the accuracy for the probability of detection and false alarm improves.

4.3 GLRT-based algorithm

Another likelihood ratio can be formed by considering the influence of a certain component i on the marginal (log)likelihood. This likelihood ratio is inspired by the GLRT and is given by

$$GLRT - RVM_i = \frac{\max_{\alpha_1 \ldots \alpha_n} p(\mathbf{t}|\alpha_1, ..., \alpha_N)}{\max_{\alpha_1 \ldots \alpha_n \backslash \alpha_i} p(\mathbf{t}|\alpha_1, ..., \alpha_N \backslash \alpha_i)} \tag{4.7}$$

The numerator of equation 4.7 is the loglikelihood that is obtained when all possible α's, including α_i (the precision of the component of interest) are included in the model. The numerator denotes the marginal loglikelihood that can be achieved when the component is not present. The change in loglikelihood generated by this component can then be thresholded and gives us the GLRT-RVM detector. The pseudocode for the algorithm is given in 3. Here, Λ is the support (relevant vectors) currently in the model and $\mathcal{L}$ represents the loglikelihood .The component α_i can be excluded by removing the basis from the input dictionary when running the RVM algorithm. Just like in Tipping's original algorithm, the α's are constrained to be larger than or equal to 0. While this algorithm tries to determine the effect of excluding a certain α_i from the model, there are certain cases where the result provided by this algorithm may not be indicative of the relevance of a certain component. An example of such a scenario is when pruning a component α_i results in α_{i-1} and α_{i+1} being falsely added to the model and the likelihood increases. The probability of this occurring increases as the coherency between the columns of $\mathbf{\Phi}$ increases.

Algorithm 3 GLRT-RVM

```
$\hat{\boldsymbol{\alpha}}, \mathcal{L}_0 \leftarrow$ RVM($\mathbf{\Phi}$,$\mathbf{t}$)
$\Lambda = \emptyset$
for each $\alpha_i$ in $\hat{\boldsymbol{\alpha}}$ do
    $\tilde{\mathbf{\Phi}} = \mathbf{\Phi}/\{\boldsymbol{\phi}_i\}$
    $\gamma, \mathcal{L}_i \leftarrow$ RVM($\tilde{\mathbf{\Phi}}$,$\mathbf{t}$)
    if $\mathcal{L}_0/\mathcal{L}_i \geq \theta$ then
        $\Lambda = \Lambda \cup i$
    end if
end for
```

4.4 Relevance Likelihood Ratio Test

Similarly, we can define $\hat{\alpha}_1, ..., \hat{\alpha}_N = arg\max_{\alpha_1,...,\alpha_N} p(z|\alpha_1, ..., \alpha_N)$

$$\frac{p(\mathbf{t}|\hat{\alpha}_1, ..., \hat{\alpha}_N)}{p(\mathbf{t}|\hat{\alpha_1}, ..., \hat{\alpha}_N, \alpha_i = 0)} \tag{4.8}$$

where the numerator is the loglikelihood obtained through the RVM, and the denominator is the loglikelihood when the variance of that component is set to 0 (precision to infinity). The major difference between this algorithm and the GLRTRVM is that in RLRT the RVM algorithm is only executed once. Thus, there is no longer a maximization over α. The expression in equation 4.8 can also be expressed as

$$\frac{p(\mathbf{t}|\hat{\alpha}_i \neq 0, A_{-i})}{p(\mathbf{t}|\hat{\alpha}_i = 0, A_{-i})} = \frac{L(\boldsymbol{\alpha}_{-i}) + l(\alpha_i)}{L(\boldsymbol{\alpha}_{-i})} \tag{4.9}$$

and provides an upper bound for equation 4.7. The bound is only met when component α_i is completely independent from all other relevant/selected components. The ratio in equation 4.9 is identical to the loglikelihood ratio that follows from Tipping's ratio. The pseudo-code for the RLRT is given by algorithm 4.

Algorithm 4 Relevance Likelihood Ratio Test (RLRT Algorithm)

```
$\hat{\boldsymbol{\alpha}}, \mathcal{L}_0 \leftarrow$ RVM($\boldsymbol{\Phi}$,$\mathbf{t}$)
$\Lambda = \emptyset$
for each $\alpha_i$ in $\hat{\boldsymbol{\alpha}}$ do
    $\mathcal{L}_i \leftarrow \alpha_i = \infty$
    if $\mathcal{L}_0/\mathcal{L}_i \geq \theta$ then
        $\Lambda = \Lambda \cup i$
    end if
end for
```

5 Extension to multiple measurement vectors

This chapter describes how the Relevance Vector Machine framework may be extended to include multiple measurement vectors (MMV). First, an overview is given of the MMV model. Secondly, the expression for the marginal loglikelihood is derived for in a MMV setting.

In case of multiple measurement vectors we can concatenate each measurement (column-vector) into a measurement matrix $\mathbf{T}$. Let D denote the number of measurements/the number of transmitted pulses. The associated weights for each measurement is then represented by $\mathbf{W}$. Here, we assume that the targets location does not change throughout the multiple measurements. This implies that the matrix $\mathbf{W}$ is row sparse. We then have that $\mathbf{T} \in \mathcal{R}^{M \times D}$, $\mathbf{W} \in \mathcal{R}^{M \times D}$. The measurement equation is then given by

$$\mathbf{T} = \mathbf{\Phi}\mathbf{W} + \mathbf{E} \tag{5.1}$$

where $\mathbf{\Phi} \in \mathcal{R}^{N \times M}$, the same as for the single measurement case, and $\mathbf{E} \in \mathcal{R}^{N \times D}$.

The joint likelihood for $\mathbf{T}$ can then be expressed as

$$\begin{aligned} p(\mathbf{T}|\mathbf{W}) = p(\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_D|\mathbf{w}_1, \mathbf{w}_2, ...\mathbf{w}_D) = \\ \prod_{d=1}^{D}[(2\pi)^{-N/2}\sigma^{-N} \exp(-\frac{||\mathbf{t} - \mathbf{\Phi}\mathbf{w}_d||^2}{2\sigma^2})] \end{aligned} \tag{5.2}$$

Furthermore, the logposterior

$$ln(p(\mathbf{W}|\mathbf{T}, \boldsymbol{\alpha}, \boldsymbol{\epsilon})) = ln(p(\mathbf{T}|\mathbf{W}, \sigma^2)) + ln(p(\mathbf{W}|\boldsymbol{\alpha})) - ln(p(\mathbf{T}|\boldsymbol{\alpha}, \sigma^2)) \tag{5.3}$$

is then maximized by

$$L(\boldsymbol{\alpha}) = -\frac{D}{2}[Nlog(2\pi) + log|\mathbf{C}|] + \mathbf{T}^T\mathbf{C}^{-1}\mathbf{T} \tag{5.4}$$

Similarly to [12] we can then split this expression in a term that is dependent on

α_i and a term that does not explicitly depend on α_i. Specifically, we then have

$$L(\boldsymbol{\alpha}) = L(\boldsymbol{\alpha}_{-i}) + \frac{D}{2}[log(\alpha_i) - log(\alpha_i + s_i)] + \sum_{d=1}^{D} \frac{q_{i,d}^2}{\alpha_i + s_i} \tag{5.5}$$

This expression is similar to the one derived in [18]. However, in [18] the assumption is made that that ϕ_i varies over multiple measurements/tasks. In our case, we do not have this constraint, and hence our expression significantly simplifies.
Building upon the model described above, the MMV equivalent estimates for μ and $\boldsymbol{\Sigma}$ are then given by

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma\Phi}\mathbf{T} \tag{5.6}$$

and

$$\boldsymbol{\Sigma} = (\mathbf{A} + \sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1} \tag{5.7}$$

Note that $\boldsymbol{\mu}$ is now a matrix, but that the expression for $\boldsymbol{\Sigma}$ is exactly the same as in the single measurement vector case.

6 Simulations for detector metrics

Monte Carlo simulations were ran in MATLAB to compare the performances of the various algorithms discussed in the previous section. This chapter describes the simulation in more detail. Firstly, an overview of the different parameters and variables that were used in the simulations are elaborated. Secondly, an overview is presented of the detector performance metrics that were used.

6.1 Simulation parameters

N_{seed} denotes the number of Monte Carlo iterations that were simulated for each scenario. This number should be high enough such that the results are accurate even when using different values for the random numbers. However, having a high number would also mean that the simulation takes a long time duration to complete. Based on these two factors, N_{seed} was set to 2000.

All of the algorithms determine whether a target is present in a certain range cell based on whether the associated ratio exceeds a certain threshold. In simple detection problems, this threshold can be set such that a desired probability of false alarm (P_{fa}) is achieved. For multitarget detection problems it is difficult to find the explicit relationship between P_{fa} and the threshold because it depends on a lot of parameters such as distance between the targets, and the SNR (signal-to-noise ratio) of each target. In our simulations, the thresholds were chosen such that the performance curves of the different algorithms could easily be compared. Empirical estimates for the P_{fa}'s and P_d (probability of detection) were then obtained from the simulations.

To properly analyze the performance of the detection algorithms, two types of measurement generation models were considered. In the first type of generation model, the RCS is assumed to be constant whereas in the second type of generation model the RCS follows a zero-mean Gaussian distribution. The latter generation model is equivalent to the real or imaginary component of a complex swerling I/II target, as was mentioned in chapter 2.

The simulations were ran for different SNR's for each target. Here, the SNR is defined as

$$\begin{aligned} SNR = & \frac{P_{reflection}}{P_{noise}} = \\ E[\frac{||\mathbf{\Phi}_i w_i||^2}{||\boldsymbol{\epsilon}||^2}] = & \frac{\mathbf{\Phi}_i^T \mathbf{\Phi} E[w_i^2]}{N\sigma^2} \end{aligned} \tag{6.1}$$

For the first generation model, the term $E[w_i^2]$ in equation 6.1 is equal to w_i^2 as w_i is a deterministic constant. However, for the second generation model the term simplifies to $E[w_i^2] = \alpha_i^{-1}$.

Other than the SNR, the PSF's -3 dB width was also varied. Increasing the width of the PSF results in the columns of $\mathbf{\Phi}$ being more correlated with each other. This is also known as an increase in coherency. It is well known that a higher coherence makes accurate detections for sparse signal recovery algorithms such as OMP more difficult [19]. It is therefore expected that the same will hold for the RVM-based algorithms.

Targets spaced closely together are likely to reduce P_d. To investigate this effect in more detail, the simulations included several scenarios with different number of range bins between the targets. When targets are spaced sufficiently apart, they will no longer influence each other's P_d. In such a case, the algorithms are expected to have similar performance metrics to each other.

Finally, two types of PSF's were used: a $Gaussian$ PSF and a $sinc$ PSF. The $Gaussian$ PSF was used as it is simple and provides basic insights into the performances of the different algorithms. To make it possible to analyze the effects of a weak target being hidden in the sidelobe of a strong target, the $sinc$ PSF had to be used as well.

6.2 Performance metrics

One way to compare the performance of different detectors is by plotting their empirical ROC-curves (Receiver Operating Characteristic-curves). Such curves illustrate how P_d and P_{fa} for the detector vary for different thresholds. A curve that approaches the top-left corner (high P_d for low P_{fa}) is considered better. In the multitarget case, there are different ways to define the P_{fa}. In this research, similar to [13], P_{fa} was determined to be the false alarm rate over the whole observation window. The false alarm rate over the whole observation window can be found by averaging out the false alarm rates for each cell ($P_{fa,i}$). Let us define $N_{fa,i}$ to be the number of false alarms that occur in range cell i throughout the Monte Carlo iterations and let $N_{tn,i}$ be the number of times no target was present and no target was detected in range cell i. The false alarm rate for a range cell i is obtained using

$$P_{fa,i} = \frac{N_{fa,i}}{N_{fa,i} + N_{tn,i}} \tag{6.2}$$

Similarly, let $N_{d,i}$ denote the number of times a target was correctly detected in range cell i, and let $N_{md,i}$ be the number of times a missed detection occured in range cell i. The probability of detection for a single cell is defined by

$$P_{d,i} = \frac{N_{d,i}}{N_{d,i} + N_{md,i}} \tag{6.3}$$

During the simulation, $N_{fa,i}$,$N_{tn,i}$,$N_{d,i}$ and $N_{md,i}$ are kept track of and updated in each Monte Carlo iteration. Once all N_{seed} iterations are complete, $P_{fa,i}$ and $P_{d,i}$ are computed. An example showing $P_{d,i}$ and $P_{fa,i}$ for each range cell, with 2 targets spaced 1 range cell apart is provided in figure 2.

In this research, the definitions for $P_{fa,i}$ and $P_{d,i}$ are quite stringent. In practice, we may not necessarily be interested in finding the precise locations of the targets. In that case we may relax the definitions for $P_{d,i}$ to include detections in neighbouring cells from the correct target locations as well. However, by using such a modification, it may be difficult to accurately determine the number of targets present in the radar scene.

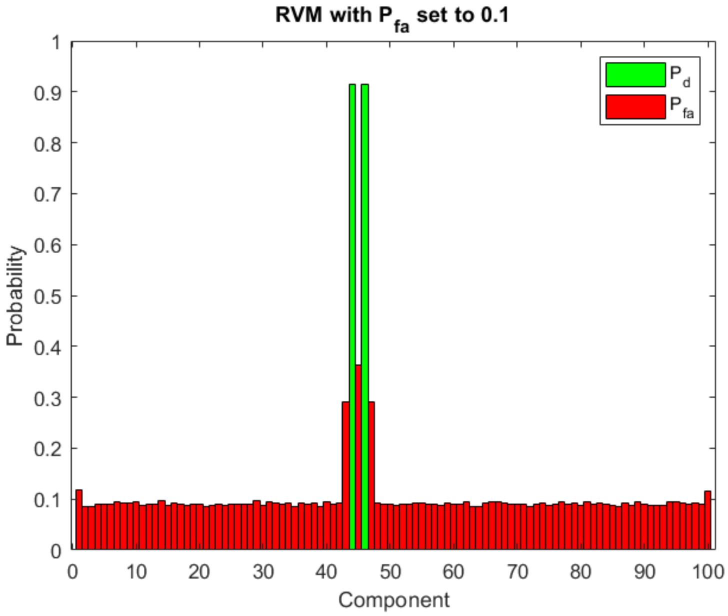

Figure 2: False alarm rate and hit rate per cell

Another method that is often used to characterize the performance of an algorithm in reconstruction problems is the MSE (mean squared error). It can be defined as

$$MSE(\mathbf{w}, \hat{\mathbf{w}}) = E[(\mathbf{w} - \hat{\mathbf{w}})^T(\mathbf{w} - \hat{\mathbf{w}})] \tag{6.4}$$

where $\hat{\mathbf{w}}$ is the estimate for $\mathbf{w}$. A downside to using this metric is that it penalizes a target being detected in a neighbouring incorrect range cell very heavily. Similarly to the previously discussed metrics, an empirical estimate for the MSE is obtained by averaging out the MSE computed using each Monte Carlo iteration.

7 Results

This chapter of the report discusses the results that were obtained using the simulator discussed in chapter 6. First, an overview of the runtime for the different algorithms is provided. Next, the effect of modifying each parameter in the simulation is discussed.

7.1 Comparison of runtimes

To compare the runtime of the different detection algorithms, the execution time for 2000 Monte Carlo iterations was kept track of for each algorithms. The results can be seen in table 1.

The table shows that the multitarget matched filter has the lowest execution time. This is as expected, since the mutlitarget matched filter is less computationally intensive compared to the other algorithms.

Another important observation is that the runtime for the modified RVM is significantly reduced compared to the basic RVM. This is because the modified RVM has a higher threshold compared to the basic RVM. Consequently, less weaker components are allowed into the model. Once the α_i's for the stronger components have been accepted, the modified RVM algorithm then terminates which leads to this speedup.

The execution time for the ESBL algorithm turned out to be much slower than expected. This can be attributed to the estimated parameters being updated to fixed-point derivatives which have a very slow convergence rate. The runtimes provided in table 1 are only for a single threshold. Thus, due to time restrictions, it was not possible to accurately evaluate the performance of the ESBL algorithm in terms of P_d and P_{fa}. However, inspection during the simulating process showed that the P_d and P_{fa} were similar to that of the modified RVM algorithm.

Algorithm	**Runtime [s]**
Basic RVM	129.5
Modified RVM	56.1
ESBL	1033.95
Matched filter	1.80
OMP	12.9
GLRT	27.4
RLRT	29.7

Table 1: Execution time for the detection algorithms

7.2 Setting the false alarm rate for the modified RVM

As a first scenario, consider the case where there are no targets present in the radar scene. The probability of detection P_d is then 0. The false alarm rate P_{fa} depends on the threshold θ that was used. For the modified RVM algorithm, the relationship between P_{fa} and the threshold was given in equation 4.5. The relationship between the desired P_{fa} and the empirical P_{fa} evaluated from the simulation can be seen in figure 3.

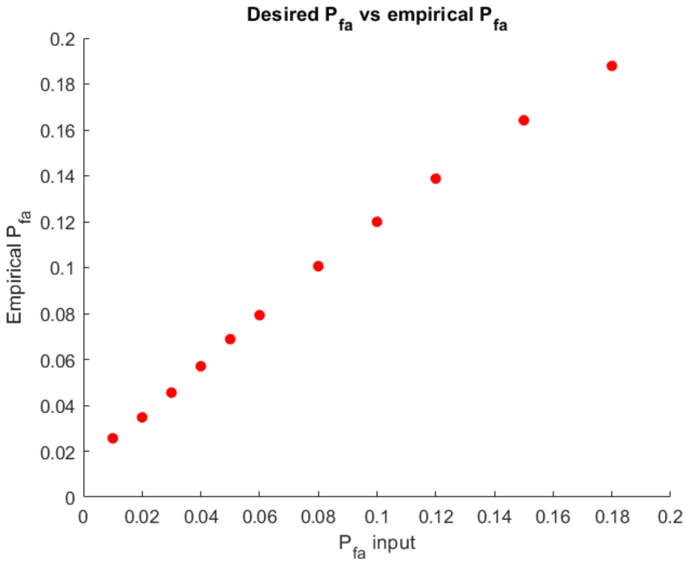

Figure 3: Desired P_{fa} vs Empirical P_{fa}

Figure 3 indicates that the desired P_{fa} doesn't perfectly match with the empirical P_{fa}. The empirical P_{fa} is always slightly higher than the desired P_{fa}. This behaviour can likely be attributed to the RVM algorithm not removing a component because it lowers the marginal loglikelihood despite that component not meeting the statistical threshold.

7.3 Single target detection

Consider the case where there is only a single target present in the radar scene.In this specific case the target has a SNR of 25.1 dB and the response from the target results in a Gaussian PSF that reaches its -3 dB point exactly at the neighbouring range bins. Figure 4 indicates that the performance of all the algorithms, except for the multitarget matched filter, are very similar. This does not correspond with our expectations, as it was expected that all of the algorithms would have very similar ROC curves. An explanation for this discrepency is that because the SNR is very high, a signal component is detected in the neighbouring bins for the multitarget matched filter. A practical way to reduce this effect is by using a clustering-based algorithm which can detect the neighbouring components jointly as a single target.

7.4 Increasing the targets spacing for Gaussian PSF

As the next scenario, consider the case where there are two targets with equal SNR that are spaced in consecutive range bins ($\Delta = 1$). Additionally we have that the amplitude of the PSF reaches its -3 dB point at the center of the neighbouring range bins ($L = 1$). The resulting ROC curves for each algorithm are displayed in figure 5. It can be observed that the curves for the RLRT and GLRT seems to coincide for the curve of the modified RVM algorithm. Furthermore, the curves for all 3 algorithms intersect with the operation point of the basic original RVM. Recalling that the original RVM algorithm has the lowest possible threshold that still increase the loglikelihood, this implies that the RVM-based algorithms will not be able to have a false alarm rate that exceeds the false alarm rate of the original RVM. Finally, we see that the RVM based algorithms outperform OMP. This is likely due to the substraction step of OMP, where two neighbouring targets are

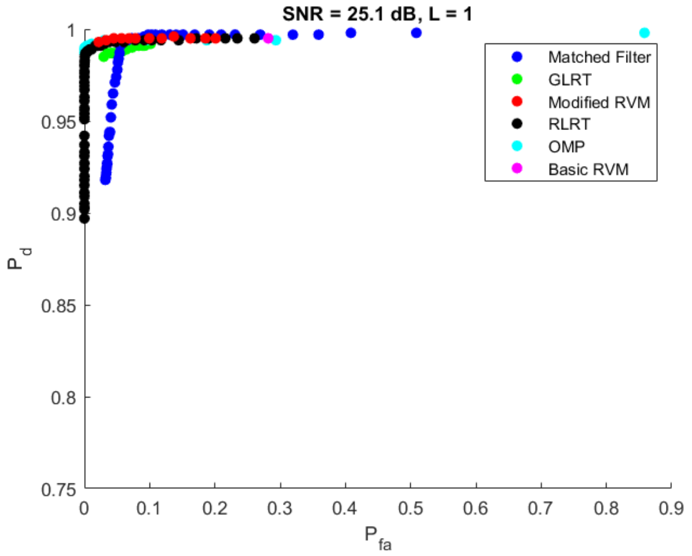

Figure 4: ROC curve for a single target, SNR = 25.1 dB, L = 1

seen as one target with a larger RCS, and subsequently too much signal is substracted. Thus, this is due to OMP not being able to remove a component from its support once it has been added. For a low false alarm rate, the RVM algorithm also has a higher probability of detection than the matched filter approach. However, the matched filter outperforms all the other algorithms at high P_{fa}'s.

Next, we use the exact same simulation parameter settings, except that there is now one range bin between the two targets ($L = 2$). The results are visualized in figure 6. In this scenario, we see that the performance of all the algorithms, except for the matched filter, are similar. These algorithms outperform the multi-target matched filter as the curves are similar for a high P_{fa}, but the matched filter performs much worse for a low P_{fa}.

Increasing the width even more, we get the data shown in figure 7. The results seem very comparable to figure 6. Recalling that our -3dB width is 1 range cell away, this would mean that the two targets barely influence each other at $\Delta = 2$, which explains these similar results.

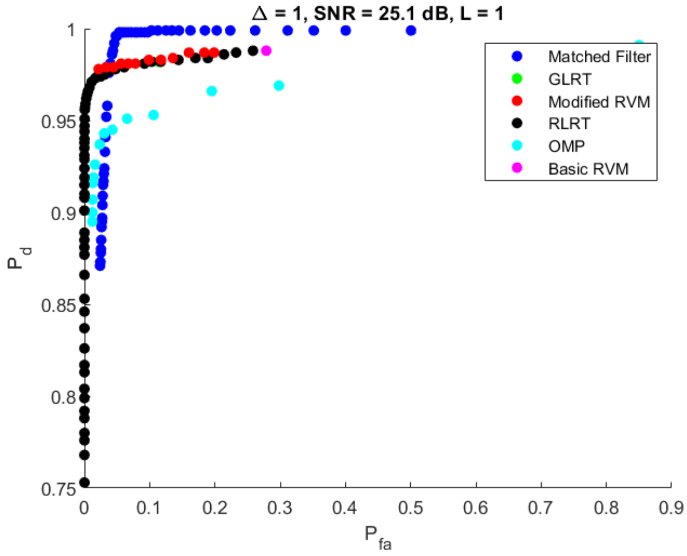

Figure 5: ROC curve for Δ =1, SNR = 25.1 dB, L=1

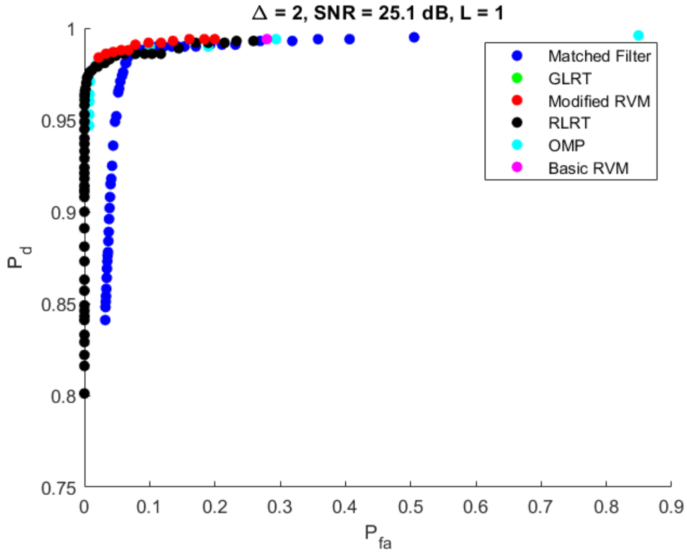

Figure 6: ROC curve for Δ =2, SNR = 25.1 dB, L=1

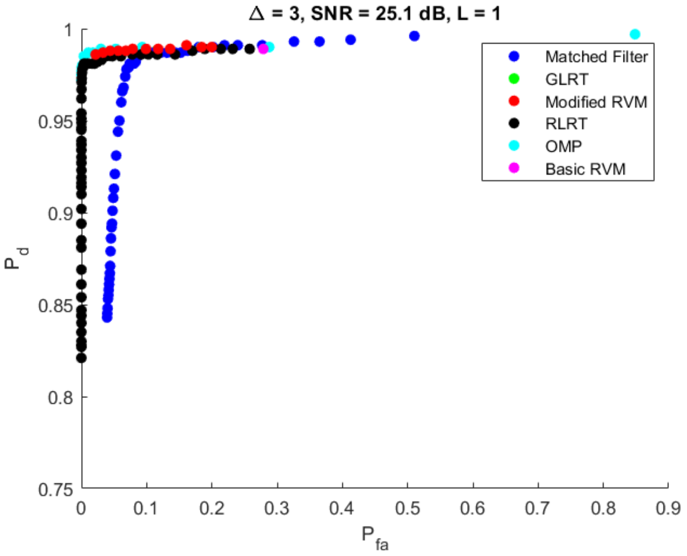

Figure 7: ROC curve for Δ =3, SNR = 25.1 dB, L=1

7.5 Detecting a weak target in a sidelobe

We now consider the case with $sinc$ PSF's, where there is a strong target with a SNR of 26.3 dB. A weaker target with a SNR of 4.8 dB is located in the sidelobe of the stronger target. The resulting ROC curves are displayed in figure 8. Here, the P_d on the y-axis represents the probability of detection for the weak target.

We can repeat this process for multiple different SNRs for the small target. Using these ROC curves, and taking a sice at around $P_{fa} = 0.147$ we obtain the plot in figure 9. From this figure it can be concluded that the RVM algorithm has a better performance than the matched filter approach. This difference seems to increase as the SNR of the weak target increases. This is also why sparse signal processing algorithms such as OMP are often used instead of the multitarget matched filter for estimating targets hidden in sidelobes [13].

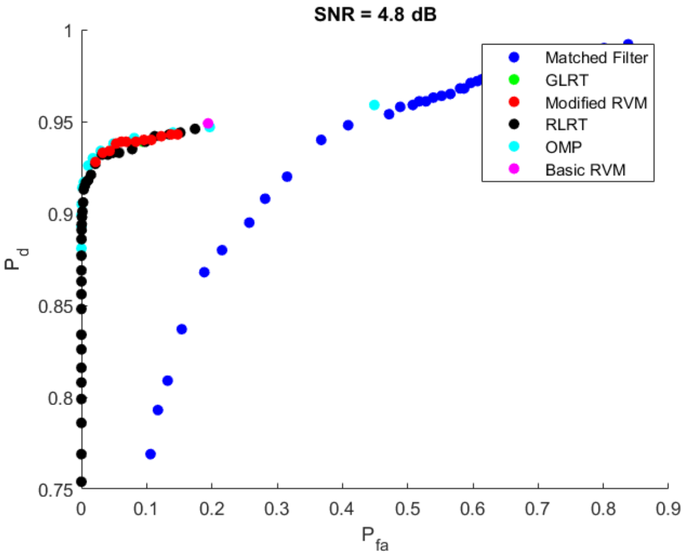

Figure 8: ROC curve, $SNR_{weak} = 4.8dB$, $SNR_{strong} = 26.3dB$

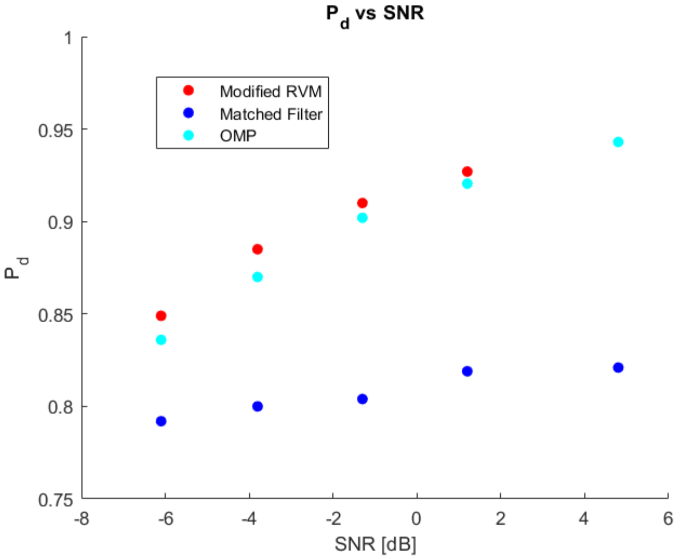

Figure 9: P_d vs SNR_{weak} , $SNR_{strong} = 26.3dB$

7.6 Effect of the PSF width

In this section, we verify the effect that the PSF has on the performance of the different algorithms. The -3 dB widths were set to 1.5, 2, 2.5 and 3. The results can be seen in figures 10-13. From these figures it can be observed that the matched filter performs much better than the RVM based algorithms and OMP when the PSF is increased. However, the RLRT seems to outperform the modified RVM and the GLRT algorithm.

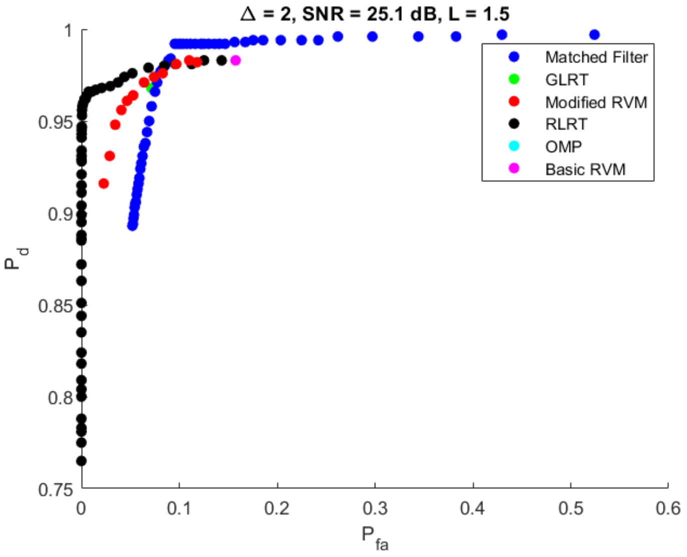

Figure 10: ROC curve, $L = 1.5$

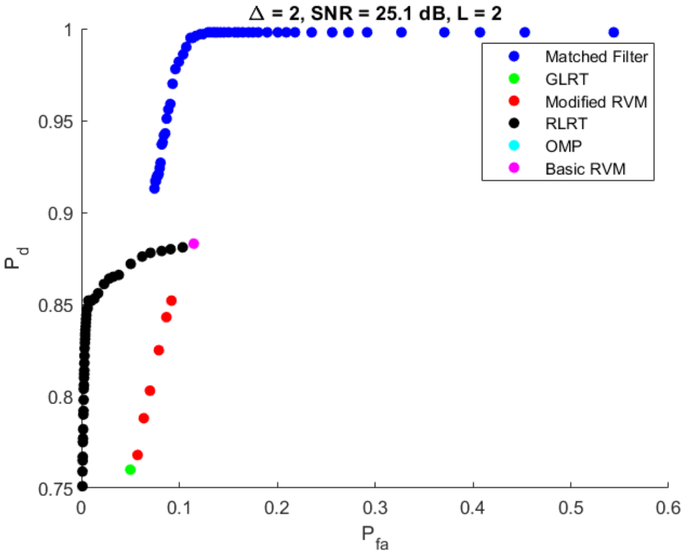

Figure 11: ROC curve, $L = 2$

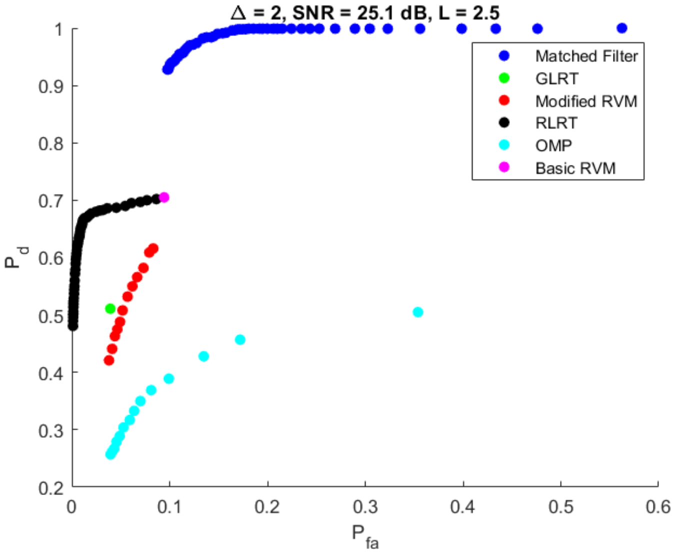

Figure 12: ROC curve, $L = 2.5$

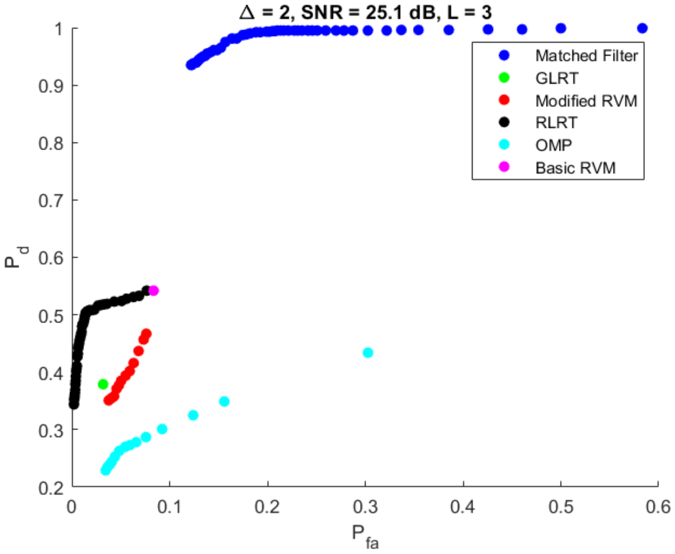

Figure 13: ROC curve, $L = 3$

8 Conclusions and future work

8.1 Conclusions

Based on the simulation results conducted during this study, it can be concluded that for low probabilities of false alarms, the RVM outperforms the multitarget matched filter approach and OMP. The LRLT and GLRT generally tend to follow similar metrics as the modified RVM algorithm. Furthermore, based on the simulations where we varied the SNR of the target, it was found that the same thresholds gave the same probability of false alarms. This implies that the expressions derived from the ESBL algorithm transfer properly to the RVM framework. However, in the case of very low SNR the RVM generally performs poorer compared to the matched filter. In these cases, the constant false alarm rate (CFAR) property of the modified RVM framework also ceases to function. In practice, we see that the empirical false alarm probability does not completely correspond to the theoretical false alarm probability.

One of the reasons that the RVM performs better compared to the other sparse signal processing algorithms examined in this study (namely MP and OMP) is because the signal support is iteratively updated based on the received measurement data. In each iteration of the RVM algorithm, it is possible that a signal support gets removed, re-estimated or deleted. Such steps are not present in the multi-target matched filter solution or in OMP.

The ESBL algorithm initializes with all components included in the support, whereas the RVM based algorithm initialize with an empty support set. Due to this, the ESBL algorithm scales poorly as the number of bins N in the measurement increases. Additionally, the ESBL has a much slower convergence rate compared to the other algorithms discussed such as OMP and RVM.

The thresholds for the GLRT-RVM algorithm are very much dependent on the measurement and specific reconstruction problem. Consequently, it is hard to tune the threshold for this algorithm in order to achieve a specific probability of false alarm. The reason that the GLRT-RVM algorithm does not provide robust results for a certain threshold may be attributed to the algorithm selecting neighbouring components α_{i-1} and α_{i+1} when the component corresponding to α is temporarily pruned.

The results obtained also showed that all the algorithms perform much worse when the -3 dB width of the PSF is increased. This is due to the increased coherency between the columns of $\mathbf{\Phi}$, which generally reduces the probability of accurate recovery. This seems to impact the multitarget matched filter less compared to the other algorithms.

8.2 Future work

It was expected that the empirical ROC-curves would coincide for the single target detection case. However, the matched filter seems to perform poorer in such scenarios when considering low threshold values/ low probabilities of false alarm. The exact reason behind this is not exactly known and should be explored in future research.

The RLRT and GLRT-RVM algorithms currently perform worse than the modified RVM algorithm in most scenarios. This is likely because the RLRT and GLRT-RVM algorithm only suppress one component at a time. Future work may explore how suppressing multiple components at once may increase the performances of these algorithms.

This research specifically focused on detecting targets based on range measurements only. This was done by having a basis function $\boldsymbol{\phi}_i$ for each range bin. In future research, this approach could be extended to consider angular and doppler velocities as well by associating a basis function $\boldsymbol{\phi}_i$ to each range-angle bin. Additionally, more complicated basis functions may also be considered.

One of the original applications that RVM was developed for was to solve standard regression problems. In such problems, the assumption that the amplitude w_i associated with a basis function $\boldsymbol{\phi}_i$ follows a Gaussian distribution with a certain mean and variance, is generally not correct. Despite this, RVM still manages to outperform various other sparse signal processing techniques and is generally able to provide acceptable results. Although the model would not be perfectly matched to the problem, future work may explore the use of the RVM framework for swerling III/IV target models.

In [17] an example is provided on how measurements from different polarization channels may be used in a multiple measurement vector (MMV) framework. A similar extension could be done for the modified RVM model. In that case, each polarization channel can be modelled by its own α_i.

In this research, the focus was specifically on processing individual measurements, and processing measurements that arrive in batch. In a more practical setting, information from previous measurement (e.g. velocity estimate) may be used as prior into the next measurement. As RVM is built upon a Bayesian framework, it can be more easily extended to include such priors. In [20] a framework has been provided in which both the weight parameters and prior hyperparameters are updated as measurement data arrives sequentially.

Finally, this study specifically focused on scenarios where the targets are located exactly on a grid point at the center of the range bin. In a practical radar scene targets do not have such restrictions. Future work may focus on how the marginal loglikelihood may be used to find the correct location of a target in a continuous parameter space using grid search.

References

[1] R. McAulay and J. Johnson. Optimal mismatched filter design for radar ranging, detection, and resolution. *IEEE Transactions on Information Theory*, 17(6):696–701, 1971.

[2] Mark A. Richards, James A. Scheer, and William A. Holm. Principles of modern radar: Basic principles. 2013.

[3] G. Li. Advanced sparsity-driven models and methods for radar applications. pages 20–21, 2020.

[4] Oguzhan Teke, Ali Cafer Gurbuz, and Orhan Arikan. A robust compressive sensing based technique for reconstruction of sparse radar scenes. *Digital Signal Processing*, 27:23–32, 2014.

[5] Matthias Weiß. Chapter 6 - sparsity-based radar technique. In Rama Chellappa and Sergios Theodoridis, editors, *Academic Press Library in Signal Processing, Volume 7*, pages 277–316. Academic Press, 2018.

[6] Valen E Johnson. *Bayes' Factors*, pages 1–6. John Wiley & Sons, Ltd, Chichester, UK, November 2016.

[7] Jianqing Fan, Chunming Zhang, and Jian Zhang. Generalized likelihood ratio statistics and wilks phenomenon. *The Annals of Statistics*, 29, 08 2000.

[8] Steven M. Kay. *Fundamentals of Statistical Signal Processing*. Prentice Hall, 2018.

[9] S.G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[10] T. Tony Cai and Lie Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, 57(7):4680–4688, 2011.

[11] Michael Tipping. The relevance vector machine. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

[12] Michael E. Tipping and Anita C. Faul. Fast marginal likelihood maximisation for sparse bayesian models. In Christopher M. Bishop and Brendan J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, volume R4 of *Proceedings of Machine Learning Research*, pages 276–283. PMLR, 03–06 Jan 2003. Reissued by PMLR on 01 April 2021.

[13] Olivier Rabaste, Laurent Savy, and Guy Desodt. Approximate multitarget matched filter for mimo radar detection via orthogonal matching pursuit. In *2014 International Radar Conference*, pages 1–6, 2014.

[14] Anita Faul and Michael Tipping. Analysis of sparse bayesian learning. volume 14, pages 383–389, 01 2001.

[15] Visa Tapio, Mubarak Umar Aminu, Janne Lehtomäki, and Markku Juntti. Channel estimation algorithms for hybrid antenna arrays: Performance and complexity. In *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, pages 293–297, 2019.

[16] Cédue Herzet and Angélique Drémeau. Bayesian pursuit algorithms. In *2010 18th European Signal Processing Conference*, pages 1474–1478, 2010.

[17] Martin Hurtado, Carlos H. Muravchik, and Arye Nehorai. Enhanced sparse bayesian learning via statistical thresholding for signals in structured noise. *IEEE Transactions on Signal Processing*, 61(21):5430–5443, 2013.

[18] Shihao Ji, David Dunson, and Lawrence Carin. Multitask compressive sensing. *IEEE Transactions on Signal Processing*, 57(1):92–106, 2009.

[19] Ehsan Miandji, Mohammad Emadi, Jonas Unger, and Ehsan Afshari. On probability of support recovery for orthogonal matching pursuit using mutual coherence. *IEEE Signal Processing Letters*, 24(11):1646–1650, 2017.

[20] N. Nikolaev and P. Tino. Sequential relevance vector machine learning from time series. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 1308–1313 vol. 2, 2005.