Joint Multi-Target Detection in Clutter using RVM and AR Modelling

J.J.L. Kant

Joint Multi-Target Detection in Clutter using RVM and AR Modelling

Thesis report

by

J.J.L. Kant

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on November 21, 2024 at 13:00

Thesis committee:	
Chair:	Dr.ir. J.N. Driessen (Daily supervisor)
Internal examiner:	Dr. F. Fioranelli
External examiner:	Dr. M. Kok
Place:	Faculty of EEMCS, HB20.090, Delft
Project Duration:	February 2024 - November 2024
Student number:	4357493

Faculty of EEMCS · Delft University of Technology

Abstract

Radar technology has evolved into a versatile and robust tool for critical air traffic control, meteorology, surveillance, and defence applications. In surveillance radar, the need for continuous monitoring of large areas, often cluttered by ground or sea reflections, presents significant challenges for multitarget detection. This clutter can obscure true targets, complicating detection in environments where standard radar noise assumptions fall short.

This thesis introduces a novel implementation based on the relevance vector machine (RVM) to address the complexities of multitarget detection in cluttered environments. Unlike conventional approaches that assume white Gaussian noise, the proposed method jointly estimates a clutter covariance matrix, allowing it to adapt to the estimated clutter model over subsequent iterations. Performance evaluations using simulated data in one-dimensional (range or angle) and two-dimensional (range-angle) settings demonstrate that the framework achieves accurate AR parameter estimation. Results indicate a marked improvement in reducing false and missed detections compared to the white-noise-based model. Notably, the framework performs multitarget detection without prior knowledge of target locations and the need for guard cells, underscoring its adaptability to real-world scenarios.

Preface

Looking back, my journey through my studies has taken a path I would not have anticipated nearly ten years ago. Although it was not straightforward, I am deeply grateful for the experiences I have gained along the way. It has brought me valuable experience at multiple companies and different research topics, which has brought me to where I am today, and I wouldn't exchange the knowledge and perspective I have gained for anything.

I would like to express my deepest gratitude to Hans, my supervisor throughout this process, who first sparked my interest in Bayesian statistics and its applications in engineering, particularly in estimation and detection. His guidance has been invaluable, and his enthusiasm for these topics has continually inspired me to dive deeper and think more critically.

I would also like to thank Francesco and Manon for taking the time to be on my thesis committee amidst their demanding schedules.

Being part of the MS3 research group has been a rewarding experience. I am grateful for the valuable suggestions and insights offered by its members, which greatly enriched my understanding of the field and introduced me to new aspects of ongoing research.

I would also like to acknowledge my colleagues at Momo Medical, who have been a steadfast source of encouragement throughout my master's studies. Their support and camaraderie have kept me motivated and balanced, and I'm grateful to have had them by my side during this journey.

Finally, to my family and friends, thank you for your constant love and support. Your belief in me has been the foundation of my success through every challenge and triumph.

Jeroen Kant
Delft, November 2024

Contents

List of Figures

1
Introduction

Radar technology, initially developed in the years preceding World War II, advanced rapidly during the war, evolving into a vital tool across numerous applications, from air traffic control to meteorology. While various radar types and specific frequency bands, such as S-band radar, enable radar systems to operate reliably across diverse weather conditions [1], the whole has proven invaluable for applications requiring consistent performance under challenging environmental conditions. Beyond these roles, radar's ability to monitor large areas and penetrate adverse weather has made it a cornerstone technology in surveillance, defence, and scientific observation [2].

Surveillance radar, in particular, is designed to continuously monitor extensive areas, including land, sea, or a combination of both, where it must detect and track multiple targets in real time. This capability is critical in border security, airspace monitoring, and maritime safety, where an up-to-date awareness of numerous moving targets is essential. Yet, the environments where surveillance radar operates are often cluttered with unwanted reflections, known as clutter, caused by the ground, waves, vegetation, weather phenomena, and even artificial structures. This clutter complicates target detection by masking or distorting the radar returns from actual targets [3].

Despite these complex environments' challenges, radar scenes are typically sparse regarding actual targets, as relatively few objects of interest are usually present at any given time. Recognizing this sparsity allows us to improve detection methods by focusing only on relevant targets and limiting false positives. Recent developments in machine learning, particularly in sparse predictive modelling, have opened new avenues for multitarget detection. The support vector machine (SVM), a well-established algorithm in machine learning [4], led to the development of the relevance vector machine (RVM) technique [5], also referred to as sparse Bayesian learning (SBL) [6], which inherently promotes sparsity in estimations. These models have shown promise in multitarget detection tasks by efficiently modelling scenes with sparse target distributions [7].

However, a significant obstacle for surveillance radar systems is volume or surface clutter from precipitation or ground or sea reflections, which can vary dynamically with environmental changes. For example, wind-blown vegetation and moving sea waves create clutter, making it harder to differentiate actual targets from background noise. Standard RVM and SBL approaches commonly assume that noise is white and Gaussian, an assumption that does not align with the complex, non-Gaussian characteristics of surface clutter. Conversely, space-time adaptive processing (STAP) methods can handle non-Gaussian clutter by estimating clutter characteristics from secondary data considered free of targets. However, data with certainty free of targets can be scarce, leading to poor estimates of the clutter characteristics [8].

This thesis addresses these challenges by presenting a novel RVM-based framework for multitarget detection within cluttered environments. Specifically, the research focuses on applications in surveillance radar settings, such as land-based or downward-looking airborne radar systems, which encounter significant surface clutter from land or sea. The proposed framework will jointly estimate the clutter covariance matrix and detect multiple targets simultaneously, making it adaptable to environments varying over time without requiring pauses in radar operation. By utilizing the sparsity of RVM, the framework will effectively manage the low-density target distributions commonly encountered in wide-area surveillance, focusing on accurate detections while minimizing false positives.

This research will use simulated data to validate the framework's versatility and effectiveness. It will examine its performance across multiple scenarios, including one-dimensional (range and angle) and two-dimensional (range-angle) settings. This approach ensures feasibility while providing insights into the framework's potential applicability in real-world environments.

This research aims to achieve its goal by answering the following three research questions:

Research Question 1

How can a framework be developed to effectively perform joint multitarget detection and clutter covariance estimation in a cluttered environment, which is stationary within a single burst but can vary over time?

Research Question 2

Can incorporating clutter covariance estimation in the detection model reduce false and missed detections?

Research Question 3

How can this framework be applied across various scenarios, including both one-dimensional (range or angle) and two-dimensional (range-Doppler) cases?

1.1. Research contributions and outline

The main contributions of this research can be summarized as follows:

- The thesis also presents an efficient approach to simultaneous clutter covariance estimation and multitarget detection by imposing a structured model on the covariance matrix. This model reduces the number of required parameters to $P \times 2 + 1$, where P represents the order of the AR model, greatly enhancing computational efficiency for models where the measurement length $N \gg P$.
- One of the main contributions of this thesis is the novel derivation of Burg's method for autoregressive (AR) parameter estimation within an Expectation-Maximization (EM) framework. Integrating Burg's method into an EM setting allows for iterative estimation of the AR model parameters and other components in the system.

The following sections outline the work surrounding these contributions:

Chapter 2 introduces foundational concepts in radar detection, explicitly focusing on range detection and direction-of-arrival (DoA) estimation using uniform linear arrays. The chapter also briefly explores how the measurement model used for DoA estimation can be adapted for Doppler processing, providing flexibility across different radar applications. Additionally, it introduces autoregressive (AR) processes, which are crucial for clutter modelling in complex environments, setting the stage for later discussions on their integration within the proposed framework.

Chapter 3 explains the basics of the Relevance Vector Machine (RVM) framework, including the modifications needed to extend its capabilities for complex-valued data processing. This chapter addresses the limitations of the original framework's white noise assumption, explaining why a full covariance matrix for the clutter is essential in cluttered environments. It also highlights the benefits of AR clutter modelling, particularly in reducing the parameters necessary for accurate target detection. It motivates the need for adaptive clutter modelling within the RVM framework.

Chapter 4 presents the derivation of the clutter covariance matrix update within an Expectation-Maximization (EM) framework. The chapter details how the covariance update depends on the residual of the measurement vector minus the estimated model and the summation terms related to the basis functions. This derivation forms the core of the proposed approach for integrating clutter covariance estimation with target detection, preparing the ground for the AR parameter estimation discussed in the next chapter.

Chapter 5 develops the use of Burg's method for AR parameter estimation, along with an extension for handling multiple segments. This chapter explains why Burg's method is particularly well-suited for integration into the RVM-based detection framework, focusing on its computational efficiency and compatibility with the joint processing of target and clutter. The chapter thus provides a practical means of implementing AR parameter estimation to maximize the results derived in Chapter 4.

Chapter 6 presents simulations across various scenarios, including one-dimensional and two-dimensional settings, to demonstrate the general applicability and performance improvements of the proposed RVM-based framework over the original implementation. This chapter assesses the framework's effectiveness in enhancing detection accuracy and discusses potential pitfalls within the RVM framework, offering insights and suggestions on mitigating these challenges in future implementations.

The final chapter summarizes the research findings, emphasizing the framework's advantages in cluttered environments. It also outlines potential directions for future work, including the exploration of advanced detection methods to further reduce false detections and missed targets, enhancing the framework's robustness and adaptability for real-world radar applications.

2 Background

2.1. Pulse radar waveforms

In this thesis, we apply the proposed framework to radar data from a radar surveillance perspective. One possible operating mode of a surveillance radar is to transmit periodic pulses and, in between these pulses, listen to the reflections of the transmission coming back to the radar. This form of operation is called a pulsed radar system, which is contrary to continuous-wave radars, which are to receive their information simultaneously while transmitting.

Pulsed radar operates by emitting short, high-power pulses of radio waves and then listening for echoes that reflect off objects, allowing for the measurement of distance based on the time delay between pulse transmission and echo reception. A matched filter is typically used to maximise the signal-to-noise ratio (SNR), aligning the received signal with the transmitted pulse through cross-correlation to improve detection accuracy [9]. However, more extensive detection ranges generally require longer pulses, but this reduces the range resolution and is therefore undesired [10]. Pulse compression addresses this by modulating longer pulses with frequency or phase, enabling high-energy transmissions that, after compression, achieve finer range resolution similar to shorter pulses [11]. This technique thus enhances both range and resolution capabilities, making pulsed radar effective for detecting distant and closely spaced objects.

A standard pulse compression method sends a modulated waveform, which further sharpens the signal's response. This thesis will look at pulse-compressed linear frequency modulated (LFM) waveforms. These waveforms, also called chirps, are characterised by the frequency being either increased or decreased during the pulse transmission. The difference in frequency between the start and the end of the pulse is the bandwidth B.

For a pulse of duration T and bandwidth B surrounding the centre frequency f_c, we transmit the following signal

$$s(t) = \begin{cases} \text{exp}\left\{j2\pi\left(\left(f_c - \frac{B}{2}\right)t + \frac{B}{2T}t^2\right)\right\} & \text{if } 0 \le t < T \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

The reflected signal then incurs a delay due to the flight time on the return trip to and from the target. If we denote this return delay as τ, the return signal is

$$r(t) = \begin{cases} \text{exp}\left\{j2\pi\left(\left(f_c - \frac{B}{2}\right)(t-\tau) + \frac{B}{2T}(t-\tau)^2\right)\right\} & \text{if } \tau \le t < \tau + T \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

After taking the convolution of the transmitted and received signals, we end up with the pulse-compressed LFM waveform. Figure 2.1 shows an example of this waveform.

Within this work, we will consider a single pulse-compressed LFM waveform as a template response that we expect from a target located at a given distance.

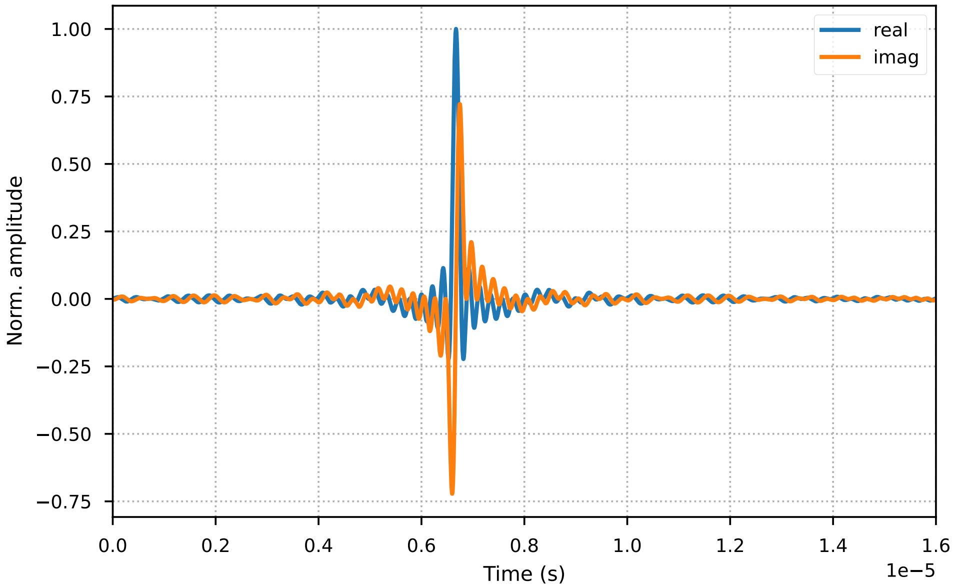

Figure 2.1: The pulse-compressed LFM waveform for a target at two kilometres from a pulse radar. The real and imaginary components are drawn in blue and yellow, respectively.

2.2. Direction-of-arrival estimation

Radar systems do not necessarily have to be composed of a single transmission and receive element. Array antennas, consisting of multiple antenna elements, can transmit or receive in parallel. Based on knowledge of the array structure, it is possible to perform a direction-of-arrival (DoA) estimation of a plane wave received by the array [12].

Based on the transmitted signal's carrier frequency, the return signal's arrival direction, and the distance between the antenna elements, you will experience a phase shift of the received signal from one element to the next. As this phase shift is relative, it is common to take one element as a reference element and express the difference relative to this reference element for each element in the array. For a sensor array of M elements, we can write the expected phase change vector **a** $\in \mathbb{C}^{1\times M}$ as [13]

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 & e^{jk} & e^{2jk} & \dots & e^{(M-1)jk} \end{bmatrix}^T \tag{2.3}$$

where the phase delay k is determined based on the interelement spacing and the carrier wavelength as

$$k = \frac{2\pi d}{\lambda} \sin\theta \tag{2.4}$$

Commonly, the interelement spacing is set at half of the carrier wavelength, as this is the maximum distance the elements can be separated by to still correctly estimate a signal perpendicular to the sensor array, similar to the Nyquist-Shannon sampling theorem for aliasing.

Figure 2.2 shows how the phase change per antenna element looks for three different arrival directions.

Whenever we examine DoA estimation in this thesis, we use the phase delay vector corresponding to a certain angle as an estimation template.

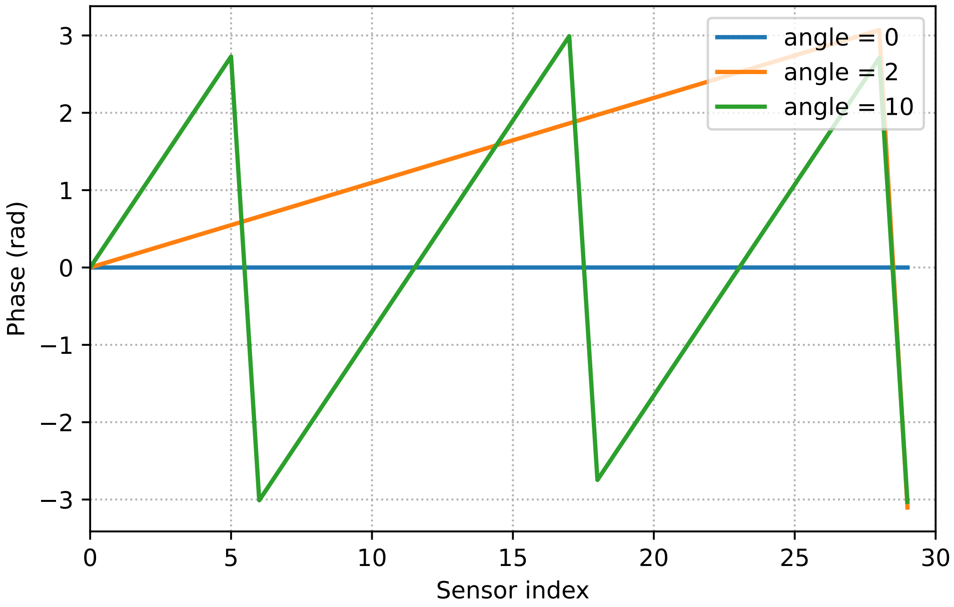

Figure 2.2: The phase difference experienced per sensor relative to the first sensor for some different angles of arrival

2.2.1. Translation to Doppler processing

In this work, we use the direction-of-arrival setting to show the framework's applicability to estimating the angles at which certain targets are present based on the data. However, the model in Equations 2.3 and 2.4 can be translated into Doppler processing by changing the phase delay definition to depend on the Doppler shift incurred due to the Doppler velocity. The relation of Doppler shift and Doppler velocity will be modelled through

$$e^{j2\pi v_D t} \tag{2.5}$$

where v_D is the Doppler shift incurred from a target. An example of how this can be expanded into basis functions is shown in [14].

2.3. Autoregressive models

An autoregressive (AR) model represents a type of random process that specifies the output to be dependent on a linear combination of previous values of the same process and noise. Autoregressive processes are considered to be of a certain order P, which describes the amount of previous values considered in the process. A discrete AR(P) model can be expressed as

$$x[n] = \sum_{i=1}^{P} a_i x[n-i] + \epsilon_n \tag{2.6}$$

where ϵ_n is white noise.

Therefore, generating an autoregressive process is as simple as synthesising a white noise sequence of the desired length and coherently weighting and summarising its values.

Alternatively, the AR(P) model can be written as a rational transfer function between the input and output sequences as an infinite impulse response (IIR) filter. Therefore, you can write the AR(P) model in the z-domain as

$$X(z) = \frac{1}{1 + a_1 z^{-1} + \cdots + a_P z^{-P}} E(z) \tag{2.7}$$

The definition above for the AR model and its AR coefficients a_p is the one we will use throughout the thesis whenever we reference the coefficients used to generate an AR process or when the coefficients have been estimated.

The likelihood of this autoregressive process, for a process vector **x** of length N, is given as [15]

$$p(\mathbf{x}) = \frac{1}{\pi^N \det\left(\mathbf{Q}_n\right)} \exp\left\{\mathbf{x}^H \mathbf{Q}_n^{-1} \mathbf{x}\right\} \tag{2.8}$$

3
Relevance vector machine

The relevance vector machine (RVM) framework, originally described by Tipping in [5], is a Bayesian treatment of the support vector machine (SVM). Both the SVM and the RVM make an estimation of a measured vector based on the weighted combination [16]:

$$y(\mathbf{x};\mathbf{w}) = \sum_{i=1}^{N} w_i K(\mathbf{x},\mathbf{x}_i) + w_0 \tag{3.1}$$

where $K(\mathbf{x},\mathbf{x}_i)$ is what we call a *basis function*. Basis functions are sample responses from a predefined location of interest. The w_i denotes the weight of how each basis function is applied to the estimation.

Although SVM has been successfully applied to many tasks, Tipping identified a couple of disadvantages in the methodology [16]. One of these was that SVM only outputs point estimates, where ideally, we desire an estimate of the conditional distribution $p(t|\mathbf{x})$, as this captures our uncertainty in the current estimation. Additionally, SVM results are relatively sparse but can still contain a fairly large set of basis functions to minimize the estimation error. Lastly, SVM imposes conditions that have to be satisfied on the specific kernel function.

Tipping proposed the RVM method as a fully probabilistic framework to mitigate these limitations. A desirable feature of RVM is that it achieves a comparable generalization performance to SVM while using dramatically fewer basis functions. This translates well to our radar-based problem, as any additional basis functions used correspond to additional false detections from the data.

3.1. Tipping's real-valued relevance vector machine

The main model behind the relevance vector machine considers a data set of input-target pairs $\{\mathbf{x}_n, t_n\}_{n=1}^{N}$ with real-valued target functions, and the assumption that the targets are samples from the following model with additive noise:

$$t_n = y(\mathbf{x}_n;\mathbf{w}) + \epsilon_n \tag{3.2}$$

Here, the ϵ_n are assumed to be independent samples from a zero-mean Gaussian noise process, with variance σ^2, i.e. $\mathcal{N}(0,\sigma^2)$. This results in the likelihood

$$p(t_n|\mathbf{x}) = \mathcal{N}(t_n|y(\mathbf{x}_n),\sigma^2) \tag{3.3}$$

where the function $y(\mathbf{x}_n)$ is as it has previously been defined in Equation 3.1.

For our relevance vector machine, however, we define our basis vectors by determining a discrete grid where possible targets are present and generating a sample response for when a target would be present at this point in the grid. A single basis function essentially represents a kernel function as defined previously: $\phi_i(\mathbf{x}) \equiv K(\mathbf{x}, \mathbf{x_i})$. From here, we arrive at the following form:

$$\mathbf{t} = \mathbf{\Phi w} + \mathbf{n} \tag{3.4}$$

where $\mathbf{t} \in \mathbb{R}^N$ is the target vector, $\mathbf{\Phi} \in \mathbb{R}^{N\times M}$ is our matrix consisting of M basis vectors, one for each grid point we modelled our target response for, $\mathbf{w} \in \mathbb{R}^M$ is the vector of weights for each of the basis functions, and the additional noise process $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. This then leads to a likelihood of the complete data set as [5], [16]

$$p(\mathbf{t}|\mathbf{w}, \sigma^2) = \left(2\pi\sigma^2\right)^{-\frac{N}{2}} \exp\left\{-\frac{1}{2}\left\|\mathbf{t} - \mathbf{\Phi w}\right\|^2\right\} \tag{3.5}$$

A prior probability distribution is defined over the weight vector **w** to prevent over-fitting in a model with many parameters. A preference for less complex functions is encoded by defining the prior as a zero-mean Gaussian distribution:

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=0}^{M-1} \mathcal{N}\left(w_i|0, \alpha_i^{-1}\right) \tag{3.6}$$

where $\boldsymbol{\alpha}$ is a vector of so-called *hyperparameters* associated independently with a single weight value. Another hyperparameter is added for the noise variance σ^2, with $\beta \equiv \sigma^{-2}$.

For these hyperparameters, priors are chosen as Gamma distributions:

$$\begin{aligned} p(\boldsymbol{\alpha}) &= \prod_{i=0}^{M-1} \text{Gamma}\left(\alpha_i|a, b\right) \\ p(\beta) &= \text{Gamma}\left(\beta|c, d\right) \end{aligned} \tag{3.7}$$

where

$$\text{Gamma}\left(\alpha|a, b\right) = \Gamma(a)^{-1} b^a \alpha^{a-1} e^{-ba} \tag{3.8}$$

with $\Gamma(a)$ being the Gamma function. Tipping, however, advocates the use of uniform scale priors with $a = b = c = d = 0$, as this results in a uniform prior distribution to these hyperparameters, which creates a form of scale-invariance: estimations are independent of linear scaling of both **t** and the basis functions [16]. Formulating prior distributions like this is a form of automatic relevance determination (ARD) as previously described by MacKay [17] and Neal [18].

With the priors defined, the posterior over all unknowns given the data can then be defined using Bayes' theorem:

$$p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2|\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})}{p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)} \tag{3.9}$$

$$= (2\pi)^{-\frac{N}{2}} |\mathbf{\Sigma}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\mathbf{w} - \boldsymbol{\mu}\right)^T \mathbf{\Sigma}^{-1}\left(\mathbf{w} - \boldsymbol{\mu}\right)\right\} \tag{3.10}$$

where

$$\mathbf{\Sigma} = \left(\mathbf{\Phi}^T \mathbf{B} \mathbf{\Phi} + \mathbf{A}\right)^{-1} \tag{3.11}$$

$$\boldsymbol{\mu} = \mathbf{\Sigma}\mathbf{\Phi}^T\mathbf{B}\mathbf{t} \tag{3.12}$$

where $\mathbf{A} = \text{diag}(\alpha_0, \alpha_1, \ldots, \alpha_{M-1})$ and $\mathbf{B} = \sigma^{-2}\mathbf{I}$.

The specific formulation here is chosen in favour of the more regular form of the posterior formula, using $p(\mathbf{t})$ as the denominator, as the computation of

$$p(\mathbf{t}) = \int p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2) p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2) d\mathbf{w} d\boldsymbol{\alpha} d\sigma^2 \tag{3.13}$$

cannot be performed analytically.

By integrating away the weights in the formulation of Equation 3.10, we arrive at the marginal likelihood over the defined hyperparameters:

$$p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2) = (2\pi)^{-\frac{N}{2}} \left|\sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\mathbf{t}^T\left(\sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T\right)^{-1}\mathbf{t}\right\} \tag{3.14}$$

The relevance vector machine, just like sparse Bayesian learning in general, can now be formulated as a *type-II maximum likelihood* procedure, where the objective is to maximize the above marginal likelihood, or equivalently its logarithm $\mathcal{L}(\boldsymbol{\alpha}, \sigma^2)$, for the hyperparameters $\boldsymbol{\alpha}$ and β [19]:

$$\mathcal{L}(\boldsymbol{\alpha}) = \log p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2) = -\frac{1}{2}\left[N\log 2\pi + \log\ |\mathbf{C}| + \mathbf{t}^T\mathbf{C}^{-1}\mathbf{t}\right] \tag{3.15}$$

with $\mathbf{C} = \sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T$.

3.2. Fast marginal likelihood maximization

Tipping's original implementation [5] initializes with all M basis functions included in the model. By iteratively updating the hyperparameters, certain basis functions would be discarded as not being relevant to describe the measurement, which accelerated the subsequent iterations. However, at the start, the algorithm would still require to perform in the order of $\mathcal{O}\left(M^3\right)$ computations. However, due to the work of [19], [20] presented a fast RVM algorithm that initializes with an empty set of basis functions in the model and, with each iteration, tries to add basis functions to the model and/or update their respective weights to maximize the marginal likelihood.

This is done by writing the expression of Equation 3.15 into a decomposed form as

$$\mathcal{L}(\boldsymbol{\alpha}) = \mathcal{L}(\boldsymbol{\alpha}_{-i}) + \ell(\alpha_i) \tag{3.16}$$

in which the marginal likelihood with the basis function ϕ_i is excluded is contained in $\mathcal{L}(\boldsymbol{\alpha})$, and $\ell(\alpha_i)$ now isolates all the terms related to α_i for the basis function ϕ_i. The analysis in [19] also shows that there is a unique maximum for $\mathcal{L}(\boldsymbol{\alpha})$ with respect to α_i, as defined for as follows [19]:

$$\begin{cases} \alpha_i = \frac{s_i^2}{q_i^2 - s_i} & \text{if } q_i^2 > s_i \\ \alpha_i = \infty & \text{if } q_i^2 \leq s_i \end{cases} \tag{3.17}$$

where s_i, referred to as the 'sparsity factor', indicates if the basis function ϕ_i has a certain similarity with the functions already included in the model. q_i, referred to as the 'quality factor', indicates how well the basis function ϕ_i corresponds with the error of the current model with ϕ_i not included.

These quantities can be calculated from

$$s_i = \frac{\alpha_i S_i}{\alpha_i - S_i} \qquad q_i = \frac{\alpha_i Q_i}{\alpha_i - S_i} \tag{3.18}$$

where $s_i = S_i$ and $q_i = Q_i$ when $\alpha_i = \infty$. The quantities S_i and Q_i are defined as

$$\begin{aligned} S_i &= \phi_i^T\mathbf{B}_{-i}^{-1}\phi_i - \phi_i^T\mathbf{B}\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{B}\phi_i \\ Q_i &= \phi_i^T\mathbf{B}_{-i}^{-1}\mathbf{t} - \phi_i^T\mathbf{B}\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{B}\mathbf{t} \end{aligned} \tag{3.19}$$

with $\mathbf{C}_{-i}^{-1} = \sigma^2\mathbf{I} + \sum_{m\neq i} \alpha_m^{-1}\phi_m\phi_m^T$ as the matrix **C** without the contribution of basis function ϕ_i.

The quality and sparsity factors q_i and s_i can be calculated for all M available basis functions at any iteration. This allows us to determine which basis function improves the marginal likelihood the most and specifically add this to the model (or update its weight if it was already included). When adding a new basis function, the delta to the marginal likelihood can be calculated as [20]:

$$\Delta\mathcal{L} = \frac{1}{2}\left(\frac{Q_i^2 - S_i}{S_i} + \log\frac{S_i}{Q_i^2}\right) \tag{3.20}$$

When a basis function is already present in the model, its delta to the marginal likelihood can be calculated using the previous hyperparameter α_i and the updated hyperparameter $\tilde{\alpha}_i$. The formula then becomes:

$$\Delta\mathcal{L} = \frac{1}{2}\left(\frac{Q_i^2}{S_i + \left[\tilde{\alpha}_i^{-1} - \alpha_i^{-1}\right]^{-1}} - \log\left\{1 + S_i\left[\tilde{\alpha}_i^{-1} - \alpha_i^{-1}\right]\right\}\right) \tag{3.21}$$

3.3. Relevance vector machine for complex-valued input

The relevance vector machine as described by Tipping concerns itself with real-valued inputs and basis functions. However, in radar scenarios, we instead commonly work with data that is complex-valued. To allow for this, we need to update the previously defined formulas to accommodate this. Our model description of Equation 3.4 can stay the same, being

$$\mathbf{t} = \mathbf{\Phi}\mathbf{w} + \mathbf{n} \tag{3.22}$$

However, now $\mathbf{t} \in \mathbb{C}^N$ is our complex-valued target vector, $\mathbf{\Phi} \in \mathbb{C}^{N\times M}$ the basis matrix of complex-valued basis functions, $\mathbf{w} \in \mathbb{C}^M$ the complex-valued weight vector. We now also assume that our noise samples are drawn from a circularly symmetric central complex Gaussian distribution, i.e. $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \mathbf{Q}_n)$. With this, we now update the likelihood of the complete data set from Equation 3.5 as

$$p(\mathbf{t}|\mathbf{w}, \mathbf{Q}_n) = \frac{1}{\pi^N \det(\mathbf{Q}_n)} \exp\left\{(\mathbf{t} - \mathbf{\Phi}\mathbf{w})^H \mathbf{Q}_n^{-1} (\mathbf{t} - \mathbf{\Phi}\mathbf{w})\right\} \tag{3.23}$$

Similarly, for the prior probability distribution over the weight vector of Equation 3.6, we now also update this to be a circularly symmetric central complex Gaussian distribution:

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=0}^{M-1} \mathcal{CN}(w_i|\mathbf{0}, \alpha_i^{-1}\mathbf{I}) \tag{3.24}$$

In defining the posterior over all unknowns given the data, we essentially only need to update the regular matrix transposes in Equations 3.10, 3.11, 3.12 to use the Hermitian transposes. This results in the following relations for the posterior on the weights:

$$p(\mathbf{w}, \boldsymbol{\alpha}, \mathbf{Q}_n|\mathbf{t}) = (2\pi)^{-\frac{N}{2}} |\mathbf{\Sigma}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{w} - \boldsymbol{\mu})^H \mathbf{\Sigma}^{-1} (\mathbf{w} - \boldsymbol{\mu})\right\} \tag{3.25}$$

$$\mathbf{\Sigma} = \left(\mathbf{\Phi}^H\mathbf{B}\mathbf{\Phi} + \mathbf{A}\right)^{-1} \tag{3.26}$$

$$\boldsymbol{\mu} = \mathbf{\Sigma}\mathbf{\Phi}^H\mathbf{B}\mathbf{t} \tag{3.27}$$

where still $\mathbf{A} = \text{diag}(\alpha_0, \alpha_1, \ldots, \alpha_{M-1})$ but now $\mathbf{B} = \mathbf{Q_n^{-1}}$.

Similarly, we need to update our quality and sparsity measures Q_i and S_i from Equation 3.19.

Here, we now have

$$\begin{aligned} S_i &= \phi_i^H \mathbf{B}_{-i}^{-1} \phi_i - \phi_i^H \mathbf{B} \mathbf{\Phi} \mathbf{\Sigma} \mathbf{\Phi}^H \mathbf{B} \phi_i \\ Q_i &= \phi_i^H \mathbf{B}_{-i}^{-1} \mathbf{t} - \phi_i^H \mathbf{B} \mathbf{\Phi} \mathbf{\Sigma} \mathbf{\Phi}^H \mathbf{B} \mathbf{t} \end{aligned} \tag{3.28}$$

from which we can conclude that S_i will stay a real-valued quantity, even when using complex-valued data. Q_i, on the other hand, has now become a complex-valued quantity as well. This has some consequences on Equations 3.17, 3.20 and 3.21, where the square of either the q_i or Q_i term is used. Updating these equations gives us the following results for the maximum of the marginal likelihood with respect to α_i

$$\begin{cases} \alpha_i = \frac{s_i^2}{q_i q_i^* - s_i} & \text{if } q_i q_i^* > s_i \\ \alpha_i = \infty & \text{if } q_i q_i^* \leq s_i \end{cases} \tag{3.29}$$

and the deltas to the marginal likelihood of adding and updating a basis function

$$\Delta \mathcal{L} = \frac{1}{2} \left(\frac{Q_i Q_i^* - S_i}{S_i} + \log \frac{S_i}{Q_i Q_i^*} \right) \tag{3.30}$$

$$\Delta \mathcal{L} = \frac{1}{2} \left(\frac{Q_i Q_i^*}{S_i + \left[\tilde{\alpha}_i^{-1} - \alpha_i^{-1} \right]^{-1}} - \log \left\{ 1 + S_i \left[\tilde{\alpha}_i^{-1} - \alpha_i^{-1} \right] \right\} \right) \tag{3.31}$$

respectively.

3.4. Clutter estimation in RVM

The RVM model in Equation 3.4 assumes a white noise term in the measurement model. However, this assumption is unrealistic in a real-world surveillance radar setting, especially one directed downward. Alongside the target reflections, the radar will also pick up surface clutter; reflections from the ground or sea. Ground reflections may appear stationary when observed over land, but movement from wind in trees or crops introduces Doppler shifts in the signal. Similarly, sea clutter is influenced by surface waves, which also cause Doppler shifts in the reflections.

Our measurement model must drop the white noise assumption by incorporating the clutter covariance matrix to account for this surface clutter. This implies that in our complex-valued model (Equation 3.22):

$$\mathbf{t} = \mathbf{\Phi} \mathbf{w} + \boldsymbol{n}, \qquad \mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \mathbf{Q}_n) \tag{3.32}$$

we no longer assume $\mathbf{Q}_n = \sigma^2 \mathbf{I}$. Instead, we will work with a full clutter covariance matrix to better capture the characteristics of non-white, structured noise in our environment.

3.4.1. Full covariance estimation in Sparse Bayesian Learning

Several efforts have been made to include a clutter covariance matrix into RVM or other sparse Bayesian learning (SBL) implementations. However, clutter is not always defined in the same way. An example is [21] and [22], which have clutter defined as certain discrete scatterers on the same grid used for the targets in the SBL formulation. Also, [23] utilises some form of sparse clutter definition, although this is only sparse in the range domain as opposed to the range-angle domain of the other works. Alternatively, [24] also estimates a clutter covariance matrix. However, this is defined in a compressive sensing setting. For this work, however, we aim to look at the homogeneous clutter that occurs throughout a certain range bin and can (and is assumed to) span over multiple range bins.

In [25], the authors claim to do a structured covariance matrix recovery using SBL. Here, however, the noise is still assumed to be normally distributed, with the authors arguing that SBL still leads to a form of a Toeplitz matrix recovery.

Some works looked into providing an estimate for the homogeneous clutter covariance. For example, [26] provides an EM-based clutter covariance matrix estimate. However, this is based on the space-time-adaptive-processing (STAP) procedure, similar to [21], [22]. The problem with STAP-related implementations, however, is their need for pilot segments, in which the assumption is made that there is no target present in the data. This can, however, run into multiple problems. The covariance matrix is estimated based on the assumption that there was no target present in the pilot segment. However, if a target was present here, this would now be wrongfully seen as part of the clutter environment and most likely stay as missed detections during actual operation. Also, the clutter covariance is due to the movement of windswept trees or sea surface waves. This also implies that the clutter process is nonstationary [27]. As a result, even if the estimates could have been initially made without target presence, these estimations made during the pilot segment need to be updated over time, causing ’blind’ time for the radar, if even applicable, or even unable to perform another pilot due to target presence.

[28] and [29] also present methods to determine clutter, both based on a certain cell-under-test (CUT) principle. This makes the assumption that the neighbours of the cell under test are free of targets, which is an assumption that doesn’t necessarily need to hold, especially not when considering extended targets.

Therefore, this work aims to create a joint estimation algorithm for multi-target detection and clutter covariance estimation within the same scan. This allows us to perform a new estimate of the clutter covariance matrix in every scan separately, therefore not hindered by any nonstationarity of the clutter in subsequent scans. Within a single scan, the clutter process is assumed to be stationary. Also, the proposed framework can do clutter covariance estimation in scans with targets present and thus is not dependent on data segments where no targets are present, leading to poor estimations whenever this data is scarce. Also, by dropping the constraint that the adjacent cells need to be free of targets, we can provide an option for targets to be closely separated in range bins and not be considered in the clutter covariance estimation.

3.4.2. Benefit of autoregressive modelling of clutter

Most mentioned methods try to produce an estimate of the full covariance matrix directly without utilizing any knowledge of the internal structure of the covariance matrix.

As described by [27], radar clutter can often be represented as a sum of complex exponentials, translating well into its Fourier spectrum. Autoregressive (AR) processes can approximate such Fourier spectra efficiently, making them effective for modelling dynamic clutter. Previous studies [27], [30]–[32] have shown that ground clutter and sea clutter can be modelled using low-order AR processes (typically of order two or three).

Given our knowledge of our clutter process, we can leverage this known covariance matrix structure to simplify the covariance matrix estimation. Especially given the understanding of the low-order autoregressive process that can be used to model the clutter, we can drastically reduce the number of variables that need to be estimated. If no structure is imposed on the covariance matrix, this results in N^2 variables being estimated, where N is the length of your measurement. However, we only need to estimate seven parameters for a clutter process of length N and order 3, being the real and complex part of each of the AR coefficients and the residual prediction error. This reduced complexity enables more efficient use of this estimated covariance matrix in the iterative process of jointly performing multi-target detection and clutter covariance estimation.

Autoregressive (AR) parameter estimation models a signal as a function of its past values plus noise, helping capture temporal dependencies in time series data. AR models are crucial in fields like signal processing and radar, where accurately modelling temporal patterns aids in forecasting and noise reduction. Several key methods exist for estimating AR parameters:

- **Yule-Walker Equations**: This method uses autocorrelation of the observed signal to estimate coefficients, offering simplicity and efficiency, especially for stationary data, though it can be sensitive to noise and small samples. [33]
- **Burg's Method**: An iterative approach, Burg's method [34] minimizes both forward and backward prediction errors, providing high-resolution estimates even for short data sequences. Its stability and noise robustness make it ideal for applications like radar. [35], [36]
- **Least Squares Estimation**: By minimizing squared differences between observed and predicted values, this method provides precise estimates, particularly useful for large datasets. However, it can be sensitive to noise and outliers. [37]
- **Maximum Likelihood Estimation (MLE)**: MLE maximizes the likelihood of the data given the model, often yielding optimal estimates if the data is Gaussian. Though computationally intensive, it performs well under complex or noisy conditions. [38], [39]
- **Alternative implementations**: Other works, like, for example, [40], attempt to make other implementations, like linear prediction, perform similarly to the methods detailed above while limiting the computational complexity needed.

Each method offers distinct advantages based on data length, noise levels, and computational demands, with choices depending on application-specific requirements.

4

Updating hyperparameters using EM

The (re)estimation of the hyperparameters to maximise the marginal likelihood of the relevance vector machine is equivalent to an expectation-maximisation (EM) update, which is guaranteed to locally maximise $\mathcal{L}$ as defined in Equation 3.15 [16]. This means that we equivalently maximise

$$\mathbb{E}_{\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\mathbf{Q}_n}\left[\log p(\mathbf{t}|\mathbf{w},\mathbf{Q}_n)p(\mathbf{w}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(\mathbf{Q}_n)\right] \tag{4.1}$$

This expectation can then be split into two independent terms that can be equivalently maximised. For the hyperparameters for the RVM weights, we get

$$\mathbb{E}_{\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\mathbf{Q}_n}\left[\log p(\mathbf{w}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})\right] \tag{4.2}$$

for which [16] shows that this gives an update equation as:

$$\alpha_i = \frac{1+2a}{\mu_i^2 + \Sigma_{ii} + 2b} \tag{4.3}$$

Independently of the update of $\boldsymbol{\alpha}$, we have the expectation term concerning the clutter covariance matrix as

$$\mathbb{E}_{\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\mathbf{Q}_n}\left[\log p(\mathbf{t}|\mathbf{w},\mathbf{Q}_n)p(\mathbf{Q}_n)\right] \tag{4.4}$$

Based on this expectation maximisation, we will now derive an updating strategy for the clutter covariance matrix.

4.1. Derivation of clutter covariance matrix update

During this work, we focus on the maximisation of the conditional likelihood, the $p(\mathbf{t}|\mathbf{w},\mathbf{Q}_n)$ term. The full expectation maximisation in regards to $\mathbf{Q}_n$ should include maximization of $\log p(\mathbf{Q}_n)$ as well, the prior defined on the covariance matrix. This prior could potentially be set up in a similar noninformative way as with the original $p(\beta)$ definition in Equation 3.7, enabling scale-invariance. This could possibly be achieved by defining an inverse Wishart prior [41], but the consequences of defining this prior would have to be studied in further work.

Focussing on the conditional likelihood, we know from Equation 3.23 that it is defined as

$$p(\mathbf{t}|\mathbf{w},\mathbf{Q}_n) = \frac{1}{\pi^N \det(\mathbf{Q}_n)} \exp\left\{(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})^H \mathbf{Q}_n^{-1} (\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})\right\} \tag{4.5}$$

Taking the logarithm of this term yields the following expression

$$\log p(\mathbf{t}|\mathbf{w},\mathbf{Q}_n) = -N\log\pi - \log\ \det(\mathbf{Q}_n) + (\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})^H\mathbf{Q}_n^{-1}(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w}) \tag{4.6}$$

We can, therefore, write the expected value for the first term as

$$\begin{aligned} \mathbb{E}_{\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\mathbf{Q}_n}\left[\log\ \{p(\mathbf{t}|\mathbf{w},\mathbf{Q}_n)\}\right] = -N\log\pi \\ -\mathbb{E}\left[\log\ \det\left(\mathbf{Q}_n\right)\right] + \mathbb{E}\left[(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})^H\mathbf{Q}_n^{-1}(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})\right] \end{aligned} \tag{4.7}$$

where the $\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\mathbf{Q}_n$ subscript is left out from subsequent expectation operators for notational convenience. We then start to write the second expectation term into a more convenient form by first expanding and separating the expectation into three separate terms

$$\begin{aligned} &\mathbb{E}\left[(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})^H\mathbf{Q}_n^{-1}(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})\right] \\ &\quad= \mathbb{E}\left[\mathbf{t}^H\mathbf{Q}_n^{-1}\mathbf{t} - 2\mathbf{t}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{w} + \mathbf{w}^H\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{w}\right] \\ &\quad= \mathbb{E}\left[\mathbf{t}^H\mathbf{Q}_n^{-1}\mathbf{t}\right] - 2\mathbb{E}\left[\mathbf{t}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{w}\right] + \mathbb{E}\left[\mathbf{w}^H\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{w}\right] \end{aligned} \tag{4.8}$$

For the third expectation term, we rewrite the product by utilizing the cyclic property of a matrix trace

$$\begin{aligned} \mathbf{w}^H\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{w} &= \left(\boldsymbol{\Phi}\mathbf{w}\right)^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{w} \\ &= \operatorname{tr}\left(\mathbf{w}^H\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{w}\right) \\ &= \operatorname{tr}\left(\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{w}\mathbf{w}^H\right) \end{aligned} \tag{4.9}$$

From here, we can now simplify the expectation to

$$\begin{aligned} &\mathbb{E}\left[(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})^H\mathbf{Q}_n^{-1}(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})\right] \\ &\quad= \mathbf{t}^H\mathbf{Q}_n^{-1}\mathbf{t} - 2\mathbf{t}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbb{E}\left[\mathbf{w}\right] + \operatorname{tr}\left(\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbb{E}\left[\mathbf{w}\mathbf{w}^H\right]\right) \end{aligned} \tag{4.10}$$

For our determined weights, we have the following expectations:

$$\mathbb{E}\left[\mathbf{w}\right] = \hat{\boldsymbol{\mu}}, \qquad \mathbb{E}\left[\mathbf{w}\mathbf{w}^H\right] = \hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^H + \hat{\boldsymbol{\Sigma}} \tag{4.11}$$

where $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ are respectively the posterior mean and covariance of the estimated weights of the RVM. Note here that $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ are as large as the amount of basis functions included in the model, i.e. $\hat{\boldsymbol{\mu}} \in \mathbb{C}^{1\times K}$ and $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{K\times K}$, where K is the amount of basis functions included. Therefore we can update and further derive our expectation term as follows

$$\begin{aligned} &\mathbb{E}\left[(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})^H\mathbf{Q}_n^{-1}(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})\right] \\ &\quad= \mathbf{t}^H\mathbf{Q}_n^{-1}\mathbf{t} - 2\mathbf{t}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\hat{\boldsymbol{\mu}} + \operatorname{tr}\left(\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\left(\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^H + \hat{\boldsymbol{\Sigma}}\right)\right) \\ &\quad= \mathbf{t}^H\mathbf{Q}_n^{-1}\mathbf{t} - 2\mathbf{t}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\mu}}^H\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\hat{\boldsymbol{\mu}} + \operatorname{tr}\left(\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\hat{\boldsymbol{\Sigma}}\right) \\ &\quad= (\mathbf{t}-\boldsymbol{\Phi}\hat{\boldsymbol{\mu}})^H\mathbf{Q}_n^{-1}(\mathbf{t}-\boldsymbol{\Phi}\hat{\boldsymbol{\mu}}) + \operatorname{tr}\left(\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\hat{\boldsymbol{\Sigma}}\right) \end{aligned} \tag{4.12}$$

4.2. Derivation of trace to segments

We want to be able to jointly estimate the covariance matrix $\mathbf{Q}_n$ within the RVM framework. Therefore we rewrite the trace term in Equation 4.12 above to arrive at a more convenient expression for this estimation. We do this by decomposing our RVM posterior covariance matrix $\hat{\boldsymbol{\Sigma}}$ using the Cholesky decomposition, $\hat{\boldsymbol{\Sigma}} = \mathbf{L}\mathbf{L}^H$. As a result, we can now rewrite our trace term as follows

$$\begin{aligned}\text{tr}\left(\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\hat{\boldsymbol{\Sigma}}\right) &= \text{tr}\left(\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{L}\mathbf{L}^H\right)\\ &= \text{tr}\left(\mathbf{L}^H\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{L}\right)\end{aligned} \tag{4.13}$$

Using $\mathbf{C} = \boldsymbol{\Phi}\mathbf{L}$, this simplifies to

$$\text{tr}\left(\mathbf{L}^H\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{L}\right) = \text{tr}\left(\mathbf{C}^H\mathbf{Q}_n^{-1}\mathbf{C}\right) \tag{4.14}$$

The trace of the matrix dot product can also be written as the summation of inner products of the separate columns of the $\mathbf{C}$ matrix.

$$\text{tr}\left(\mathbf{C}^H\mathbf{Q}_n^{-1}\mathbf{C}\right) = \sum_{k=1}^{K}\mathbf{c}_k^H\mathbf{Q}_n^{-1}\mathbf{c}_k \tag{4.15}$$

Therefore, we can now write the full expectation for the complex-valued model as

$$\mathbb{E}\left[(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})^H\mathbf{Q}_n^{-1}(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})\right] = (\mathbf{t}-\boldsymbol{\Phi}\hat{\boldsymbol{\mu}})^H\mathbf{Q}_n^{-1}(\mathbf{t}-\boldsymbol{\Phi}\hat{\boldsymbol{\mu}}) + \sum_{k=1}^{K}\mathbf{c}_k^H\mathbf{Q}_n^{-1}\mathbf{c}_k \tag{4.16}$$

4.2.1. Alternative derivation using eigendecomposition

In the decomposition of the RVM posterior covariance matrix $\hat{\boldsymbol{\Sigma}}$ of Equation 4.13, we made use of the Cholesky decomposition. However, a similar result can also be obtained by using the eigendecomposition instead. In this case, we would decompose the covariance matrix as

$$\hat{\boldsymbol{\Sigma}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H \tag{4.17}$$

To obtain the same structure as we had in Equation 4.14, we need to split the diagonal eigenvalue matrix into its square root:

$$\hat{\boldsymbol{\Sigma}} = \mathbf{U}\sqrt{\boldsymbol{\Lambda}}\sqrt{\boldsymbol{\Lambda}}^H\mathbf{U}^H \tag{4.18}$$

From this, we end up at

$$\begin{aligned}\text{tr}\left(\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\hat{\boldsymbol{\Sigma}}\right) &= \text{tr}\left(\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{U}\sqrt{\boldsymbol{\Lambda}}\sqrt{\boldsymbol{\Lambda}}^H\mathbf{U}^H\right)\\ &= \text{tr}\left(\sqrt{\boldsymbol{\Lambda}}^H\mathbf{U}^H\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{U}\sqrt{\boldsymbol{\Lambda}}\right)\\ &= \text{tr}\left(\mathbf{R}^H\boldsymbol{\Phi}^H\mathbf{Q}_n^{-1}\boldsymbol{\Phi}\mathbf{R}\right)\\ &= \text{tr}\left(\mathbf{C}^H\mathbf{Q}_n^{-1}\mathbf{C}\right)\end{aligned} \tag{4.19}$$

where we used $\mathbf{R} = \mathbf{U}\sqrt{\boldsymbol{\Lambda}}$ and $\mathbf{C} = \boldsymbol{\Phi}\mathbf{R}$. This means we can construct our $\mathbf{C}$ matrix through either of the decompositions described above.

5

Applying Burg's method to EM

To solve our EM-update for the covariance matrix $\mathbf{Q}_n$, we need to find a way to maximise a term of the form $(\mathbf{t} - \mathbf{\Phi w})^H \mathbf{Q}_n^{-1} (\mathbf{t} - \mathbf{\Phi w})$. Attempting to maximise this term directly would lead to a nonlinear optimisation that needs to be performed [38]. Therefore, it is common to use recursive schemes to solve this maximisation [34], [38], [42]. Burg's method [34] is a favourable implementation to solve this maximisation, as this guarantees the estimated model to be both stable and have residual variances that lie close to the actual value [35], [36]. Compared to the maximum likelihood estimation (MLE) methods, performing as part of a joint processing step alongside the RVM on every iteration is also computationally feasible due to the lower computation complexity. This lower complexity in comparison is also shown by [42], which remarks that Burg's method does not consider the $\log$ det $(\mathbf{Q}_n)$ term of Equation 4.7, which we did not account for yet in our derivation. Given Burg's method being a favourable method for complexity, we will also therefore not consider the maximization of this $\log$ det $(\mathbf{Q}_n)$ term, and leave investigation of the feasibility of implementing this using the full MLE as possible future work. In this work, we therefore apply Burg's method to perform the maximization of the expectation of Equation 4.16.

Burg's method, also occasionally referred to as the maximum-entropy method (MEM), is described initially in [34]. It is a recursive algorithm that estimates the autoregressive parameters by determining reflection coefficients k. These reflection coefficients are later converted to the actual AR parameters by utilising the Levinson-Durbin recursion [43], [44].

5.1. Burg's maximum entropy method

The main idea of Burg's method is to maximise the entropy of the process in an information-theoretic sense, ensuring that no information is added as a result of the prediction process [31]. The algorithm bases itself on the forward and backward prediction errors for a pth order AR model, with data measurements $\{x(n)\}, n = 1, 2, \ldots, N$, defined as [45]

$$f_p(n) = x(n) + \sum_{i=1}^{P} \hat{a}_{i,p} x(n-i), \quad n = p+1, \ldots, N \tag{5.1}$$

$$b_p(n) = x(n-p) + \sum_{i=1}^{P} \hat{a}^*_{i,p} x(n-p+i), \quad n = p+1, \ldots, N \tag{5.2}$$

Burg's method recursively estimates the AR(p) model for $p = 1, \ldots, P$, utilising the $p-1$ estimated reflection coefficients $\hat{k}_1, \ldots, \hat{k}_{p-1}$ determined from the forward and backward prediction errors up to that point. The forward and backward prediction errors for stage p can then be calculated using:

$$f_p(n) = f_{p-1}(n) + \hat{k}_p b_{p-1}(n-1) \tag{5.3}$$

$$b_p(n) = b_{p-1}(n-1) + \hat{k}_p^* f_{p-1}(n) \tag{5.4}$$

The estimate for the reflection coefficient for the pth order is then calculated as:

$$\hat{k}_p = \frac{-2\sum_{n=p+1}^{N} f_{p-1}(n) b_{p-1}^*(n-1)}{\sum_{n=p+1}^{N}\left[|f_{p-1}(n)|^2 + |b_{p-1}(n-1)|^2\right]} \tag{5.5}$$

5.2. Extension to multiple segments

In our situation, however, we must not only maximise for the $(\mathbf{t} - \mathbf{\Phi w})^H \mathbf{Q}_n^{-1} (\mathbf{t} - \mathbf{\Phi w})$ term but also the summation $\sum_{k=1}^{K} \mathbf{c}_k^H \mathbf{Q}_n^{-1} \mathbf{c}_k$. This is a collection of vectors subject to the same covariance matrix $\mathbf{Q}_n$. These vectors are, however, all other realisations of this similar autoregressive process and can, therefore, not be concatenated into a single data vector to process using Burg's method. To still end up with a single estimate for $\mathbf{Q}_n$, we utilise the reformulation of Burg's method that can be applied to separate data segments [46]. This allows us to process all our vectors, both the $(\mathbf{t} - \mathbf{\Phi}\hat{\boldsymbol{\mu}})$ vector and the vectors from the $\mathbf{C}$ matrix, into a single estimate in one go. This updated estimate of $\mathbf{Q}_n$ is our variable update in the expectation-maximisation setting. After this updated estimate of $\mathbf{Q}_n$ has been computed, an update to the weight hyperparameters $\boldsymbol{\alpha}$ will be done again to iterate until convergence of both sets of parameters.

However, once we have S separate data segments, say $x^{[s]}(n), s = 1, \ldots, S$, all of the same length, we can apply the extension of [46] to the respective formulas above. Simply put, all the references x, f_p and b_p now get their respective superscripts denoting to which segment they belong: $x^{[s]}$, $f_p^{[s]}$ and $b_p^{[s]}$, respectively. The reflection coefficient for the pth order, based on all available segments, now becomes [46]

$$\hat{k}_p = \frac{-2\sum_{s=1}^{S}\sum_{n=p+1}^{N} f_{p-1}^{[s]}(n) b_{p-1}^{*[s]}(n-1)}{\sum_{s=1}^{S}\sum_{n=p+1}^{N}\left[\left|f_{p-1}^{[s]}(n)\right|^2 + \left|b_{p-1}^{[s]}(n-1)\right|^2\right]} \tag{5.6}$$

We will work with a variable total amount of segments S for our implementation inside the expectation-maximisation iterations. We will start with an empty set of included basis functions based on the fast marginal likelihood maximisation scheme defined in Section 3.2. As a result, we will not even have a $\mathbf{C}$ matrix as defined in Equation 4.14 to begin with. Once the iterations start and the amount of included basis functions increases, the size of the $\mathbf{C}$ matrix increases as well, increasing the total amount of segments being processed.

5.3. From reflection coefficients to AR parameters and covariance matrix

By utilising the Levinson-Durbin recursion, we can convert the estimated reflection coefficients for all the orders $p = 1, \ldots, P$ into their respective AR parameters [31], [44], [45]

$$\hat{a}_i^p = \begin{cases} \hat{a}_i^{p-1} + \hat{k}_p \hat{a}_{p-i}^{p-1,*} & 1 \leq i \leq p-1 \\ \hat{k}_p & i = p \end{cases} \tag{5.7}$$

Once we've obtained the AR parameters for the AR(p) model, we need to convert this information into our estimated clutter covariance matrix. This covariance matrix can be decomposed into a Toeplitz matrix of AR coefficients and a diagonal matrix containing the prediction error variances [42], [47]:

$$\mathbf{Q}_n^{-1}(\mathbf{a}, r_0, \boldsymbol{\sigma}^2) = \mathcal{A}_n \mathbf{D}_n^{-1} \mathcal{A}_n^H \tag{5.8}$$

where $\mathbf{a} = \begin{bmatrix} a_1^1 & a_1^2 & a_2^2 & \dots & a_{p-1}^p & a_p^p \end{bmatrix}^T$, the AR parameters up to order p. The Toeplitz $N \times N$ matrix $\mathcal{A}_n$ is then constructed based on these parameters as

$$\mathcal{A}_n = \left[\begin{array}{cccc|cccc}
1 & a_1^1 & \cdots & a_{p-1}^{p-1} & a_p^p & 0 & \cdots & 0 \\
0 & 1 & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & a_1^{p-1} & \vdots & & \ddots & 0 \\
\vdots & & \ddots & 1 & a_1^p & & & a_p^p \\
\cline{1-4}
 & & & & 1 & \ddots & & \vdots \\
 & & & & 0 & \ddots & \ddots & \vdots \\
\multicolumn{4}{c|}{\mathbf{0}} & \vdots & \ddots & \ddots & a_1^p \\
 & & & & 0 & \cdots & 0 & 1
\end{array}\right] \tag{5.9}$$

and $\mathbf{D}_n$ is the $N \times N$ diagonal matrix

$$\mathbf{D}_n = \frac{1}{\sigma_p^2} \text{diag}\left(\sigma_0^2, \sigma_1^2, \sigma_2^2, \dots, \sigma_p^2, \sigma_p^2, \dots, \sigma_p^2\right) \tag{5.10}$$

where ρ_i is the prediction error power for the AR model of order i. This prediction error is also computed using the Levinson recursion as [31]:

$$\sigma_p^2 = (1 - |k_p|^2)\sigma_{p-1}^2 \tag{5.11}$$

This means that to finish the construction of our clutter covariance matrix, we at least need the value of σ_0^2. This is the variance of the data measurement vector $\mathbf{x}$, calculated as [42]

$$\sigma_0^2 = \frac{1}{N} \mathbf{x}^H \mathbf{x} \tag{5.12}$$

In our multisegment cases, where there have been more basis functions included, we determine this value by taking the $(\mathbf{t} - \boldsymbol{\Phi}\hat{\boldsymbol{\mu}})$ vector for the $\mathbf{x}$ vector in Equation 5.12 above. This is done as $(\mathbf{t} - \boldsymbol{\Phi}\hat{\boldsymbol{\mu}})$ can be seen as the ”residual” of our measurement vector after subtracting our predicted model. This vector is consistently present in the multiple segments we supply using Burg’s method for segments.

6

Simulation-based performance analysis

Three different simulation scenarios have been created to evaluate how the proposed method based on the clutter covariance estimation works in relation to the original implementation with white noise assumption. The first scenario is a one-dimensional scenario where targets are distributed along the range axis. This is a fictitious example that helps to visualize the process and show some of the key concepts in the usage of the framework. The second scenario implements a direction-of-arrival (DoA) estimation problem based on a uniform linear array (ULA) of antenna elements. In the third scenario, we expand on the second scenario to implement a two-dimensional estimation of range and angle for a homogeneous clutter in range assumption.

All of the simulations in this chapter have been implemented in `Python 3.11.10`, utilizing the popular scientific computing packages `NumPy` [48] and `SciPy` [49].

6.1. Algorithm

All simulations performed in this chapter are the results of the same algorithm implementation. The following chapter will show the results of the white noise assumption and the proposed framework. These results have been made by executing the relevance vector machine on the same measurement vector twice, where once the **B** matrix is composed in the way Tipping does, and once it is composed by performing the EM-maximisation using Burg's method for segments as proposed in Chapter 5. To provide a fair comparison between the two implementations, for the original implementation using white noise, the estimated white noise variance is updated based on the residual in every iteration as well. For either implementation, the remainder of the algorithm remains the same. This is outlined in pseudocode in Algorithm 1.

6.2. One-dimensional range-only scenario

We work with multiple targets distributed along the range dimension for the initial simulated scenario. To simulate this, we start by defining an array of distances where we assume the target could be present. For each of these points, we then generate the expected return signal. Each return signal becomes a single basis function in our dictionary. For these return signals, we will use the pulse-compressed linear frequency modulated (LFM) waveform that was discussed in Section 2.1. These waveforms are stored as a separate basis function in the dictionary for each point. This could be seen as a parallel to the radar range resolution. In that case, a basis function is implemented for every range resolution cell of the radar.

The measurement is then simulated as a certain number of targets at different ranges from the radar, each with its own relative amplitude. The noiseless measurement vector can be created by simply summing over these basis functions. To generate our noisy measurement, which has been corrupted with noise, our autoregressive process is generated by filtering a white noise

Algorithm 1: RVM with EM-Burg implementation

```
Input: Measurement vector t, dictionary matrix $\Phi$, AR model order $P$
Initialize:
Estimate reflection coefficients k, prediction error $\sigma^2$ on t using (5.5) and (5.11)
Construct $\mathbf{B} = \mathbf{Q}_n^{-1}$ using (5.8)
Calculate which basis function maximalizes marginal likelihood using (3.30)
Set this function as initial basis
Calculate $\hat{\Sigma}, \hat{\mu}$ using (3.26) and (3.27)
Iterate until convergence:
for iteration $i \leftarrow 1$ to stopping criteria do
    Calculate which basis function maximalizes marginal likelihood using (3.30) and (3.31)
    Estimate reflection coefficients k, prediction error $\sigma^2$ on $\mathbf{t} - \Phi\hat{\mu}$ using (5.6) and (4.19)
    Construct $\mathbf{B} = \mathbf{Q}_n^{-1}$ using (5.8)
    Update included basis with additional basis function or updated weight
    Calculate $\hat{\Sigma}, \hat{\mu}$ using (3.26) and (3.27)
end
Output: $\mathbf{y} = \Phi\hat{\mathbf{w}}$
```

signal with the desired AR coefficients and added to the noiseless measurement vector.

This initial scenario is a fictitious example, implying some form of autoregressive clutter would be present in the range domain after pulse compression, for which no evidence could be found. It, therefore, simply serves to paint a picture that can be fairly easily understood in terms of basis functions, the defined grid points, and the implications of orders of the autoregressive process. More realistic scenarios will be discussed in Sections 6.3 and 6.4.

6.2.1. Simple AR(1) scenario

For our initial testing and visualisation, we implemented a first-order autoregressive noise process on top of the pulse-compressed LFM responses of four different targets. These targets are located 1 km, 2.5 km, 5 km, and 6 km from the radar, with amplitudes of 0.5, 0.3, 1.0, and 0.5, respectively. The distance between the grid points for the basis functions is 100 meters, placing the targets at the range bins 10, 25, 50 and 60. The autoregressive clutter process is a first-order AR process, generated using the complex AR coefficient $a_1^1 = -0.95 \times \left(\frac{1}{2\sqrt{2}} + \frac{1}{\sqrt{2}}j\right)$. The generated noisy vector is then stored and processed using the original white noise assumption implementation from [20] and the new proposed method of covariance estimation. The results of these simulations are shown in Figures 6.1 and 6.2.

Those figures show a very drastic difference between the two implementations. The total of 4 targets present in the measurement are described by the original RVM implementation using 21 basis functions. This effectively corresponds to 17 false detections being made. The weights of the false detections differ, for which one could argue that a weight below 0.1 might be viable to be discarded. However, the difference between the weight associated with the actual target located at 2.5 km and the false detection at 1.5 km becomes more difficult to decide between, as well as the false detection at 7.1 km. The proposed method, however, manages to correctly detect the 4 basis functions corresponding to the targets, not resulting in any false detections.

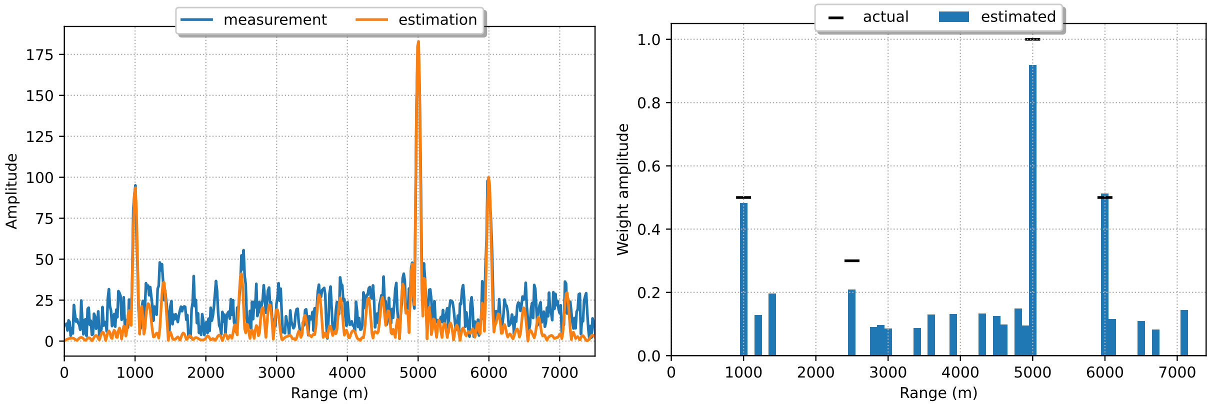

Figure 6.1: The results of using a relevance vector machine (RVM) with a white noise assumption for a scenario with four targets located at 1000, 2500, 5000, and 6000 meters (range bins 10, 25, 50, and 60, respectively). The left image shows the predicted target vector amplitude based on the estimated weights for the basis functions in the dictionary. The right image displays the amplitude of the weights with which each of the 21 basis functions is included in the estimation. Additionally, a clutter autoregressive (AR) process of order 1 with coefficient $a_1^1 = -0.95 \times \left(\frac{1}{2\sqrt{2}} + \frac{1}{\sqrt{2}}j\right)$ is added to the measurement. The RVM used these 21 basis functions to predict the measurement vector.

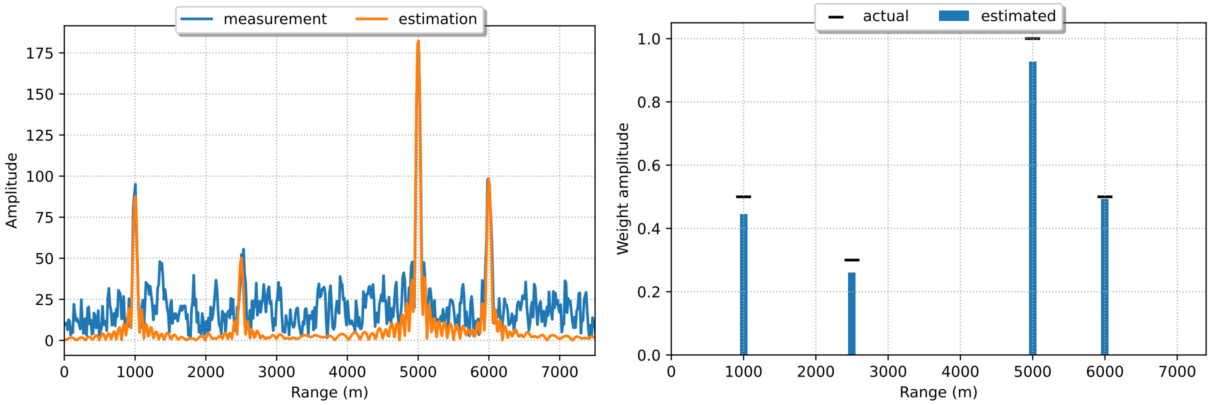

Figure 6.2: The results of using the proposed relevance vector machine (RVM) with clutter covariance matrix estimation based on Burg's method for a scenario with targets at 1000, 2500, 5000, and 6000 meters (range bins 10, 25, 50, and 60, respectively). The left image shows the predicted target vector amplitude based on the estimated weights for the basis functions in the dictionary, while the right image displays the amplitude of the weights for each basis function included in the estimation. Only 4 basis functions were required to generate this estimation. The clutter autoregressive (AR) process is the same realization of the first-order AR process as in Figure 6.1. The proposed method successfully predicted the targets in the measurement vector using these 4 basis functions.

The differences between the two implementations are not always as stark. In Figures 6.3 and 6.4, the results are shown for the case where the same target vector is now embedded into a clutter AR process with AR coefficient $a_1^1 = -0.65 \times \left(\frac{\sqrt{3}}{2} + \frac{1}{2}j\right)$. Still, the white noise assumption results in a estimation with additional vectors, but the amount has now significantly been reduced. The estimation based on the white noise assumption now utilizes 10 basis functions instead of the 4 still used by the proposed method.

There are even situations where there is no difference to be noted between the original and the proposed implementations. An example of this is when encountered while updating the clutter process to be generated using an AR coefficient of $a_1^1 = 0.4 \times \left(\frac{\sqrt{3}}{2} - \frac{1}{2}j\right)$. The figures

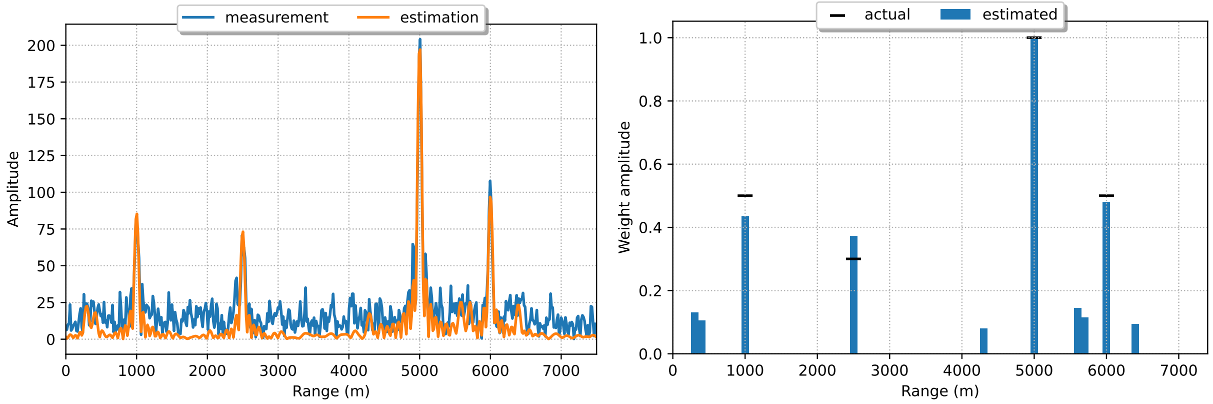

Figure 6.3: The results of using a relevance vector machine (RVM) with a white noise assumption for a scenario with the same target structure as in Figures 6.1 and 6.2. The left image shows the predicted target vector amplitude based on the estimated weights for the basis functions in the dictionary, while the right image displays the amplitude of the weights for each basis function included in the estimation. A total of 10 basis functions were used to create this estimation. In this case, the clutter process realization is based on an autoregressive (AR) coefficient of $a_1^1 = -0.65 \times \left(\frac{\sqrt{3}}{2} + \frac{1}{2}j\right)$. The RVM used these 10 basis functions to predict the measurement vector.

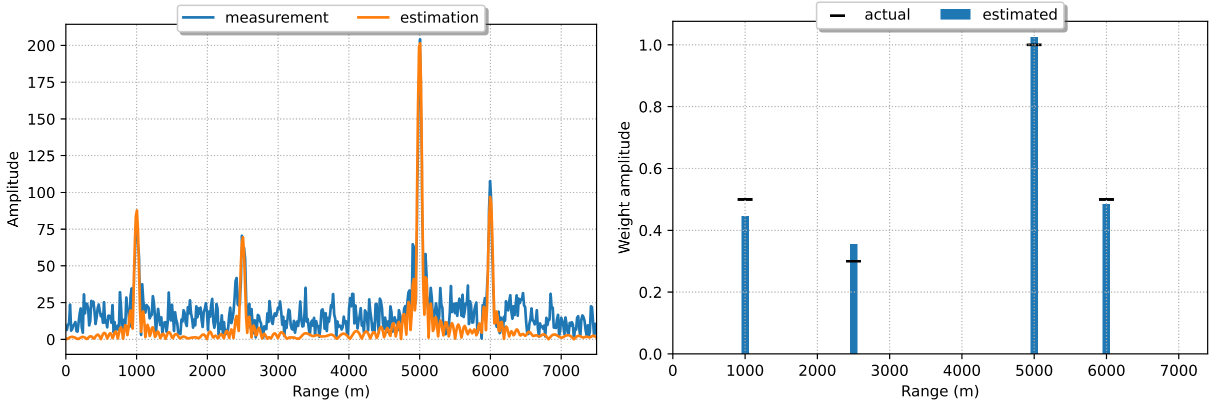

Figure 6.4: The results of using the proposed relevance vector machine (RVM) with clutter covariance matrix estimation based on Burg's method, applied to a scenario with the same target structure as Figures 6.1 and 6.2. The left image shows the predicted target vector amplitude based on the estimated weights for the basis functions in the dictionary, while the right image displays the amplitude of the weights for each of the 4 basis functions used in the estimation. The clutter autoregressive (AR) process is the same realization of the first-order AR process as in Figure 6.3. The proposed method successfully predicted the targets in the measurement vector using these 4 basis functions.

corresponding to this case, Figures A.1 and A.2, are included in Appendix A for reference. Here, both implementations were capable of perfectly reconstructing the expected target vector. This is most likely due to the quality factor being determined within the RVM framework, where the basis functions no longer provide a good enough quality match with the clutter to be considered a viable option. This happens due to the AR process ”changing shape” due to the difference in the AR coefficient. Therefore, the actual benefit of the proposed framework relative to the original implementation somewhat depends on the characteristics of the AR process that clutters the measurement.

6.2.2. Order selection

Using Burg's method, you can recursively estimate the AR coefficients up to a given model order. Based on the estimated reflection coefficients for every order, the coefficients are also determined recursively, as described in Section 5.3. In the following scenario, we investigate the consequence of setting up Burg's method with a higher predicted order than the actual process.

In this scenario, we again placed four targets at different ranges, this time at 1.2 km, 3.0 km, 4.8 km and 5.6 km, with amplitudes of 0.5, 1.0, 0.3 and 0.8, respectively. Added to the targets is an AR(1) process with again a coefficient of $a_1^1 = -0.95 \times \left(\frac{1}{2\sqrt{2}} + \frac{1}{\sqrt{2}}j\right)$.

Figure 6.5 shows the result of processing this signal using a 3rd order assumption in Burg's method. It misses the target located at 4.8 km of the radar but manages to match the other three targets closely without any false detections. The output of Burg's method for the different AR coefficients is as follows:

$$\begin{aligned} \hat{a}_1^3 &= -0.355 - 0.723j \\ \hat{a}_2^3 &= -0.056 + 0.042j \\ \hat{a}_3^3 &= 0.025 + 0.024j \end{aligned} \tag{6.1}$$

The actual AR(1) coefficient, using the same notation of complex numbers, is given as

$$a_1^1 = -0.336 - 0.672j \tag{6.2}$$

From this, we can see that Burg's method has fairly accurately managed to estimate the correct coefficient for the first lag, and also came to the conclusion that there is hardly any dependency on the second and third lags in the AR process, concluding from the $\hat{a}_2^3$ and $\hat{a}_3^3$ coefficients being close to zero.

However, when we use the knowledge that the AR process should be of the first order and therefore limit Burg's method only to predict the AR coefficient of the first order, we obtain the result as shown in Figure 6.6. Here, we can see that the RVM now resolved all four targets in the measurement. The predicted AR coefficient with the limited AR order in Burg's method is now given as

$$\hat{a}_1^1 = -0.328 - 0.691j \tag{6.3}$$

which is a closer representation of the actual AR coefficient when compared to $\hat{a}_1^3$ of Equation 6.1. The difference between these coefficients is explained by the recursion in Equation 5.7, where the residual reflection coefficients of the higher orders cause an update of all of the coefficients besides the last one.

The improved estimate of the AR coefficient helps better explain the experienced noise process, resulting in the RVM's ability to discern the additional target that was initially left buried in the noise. This advocates the benefit that the framework has by having a form of order selection built into it, which could be based on the Aikake information criterion (AIC) [50] or any alternative,

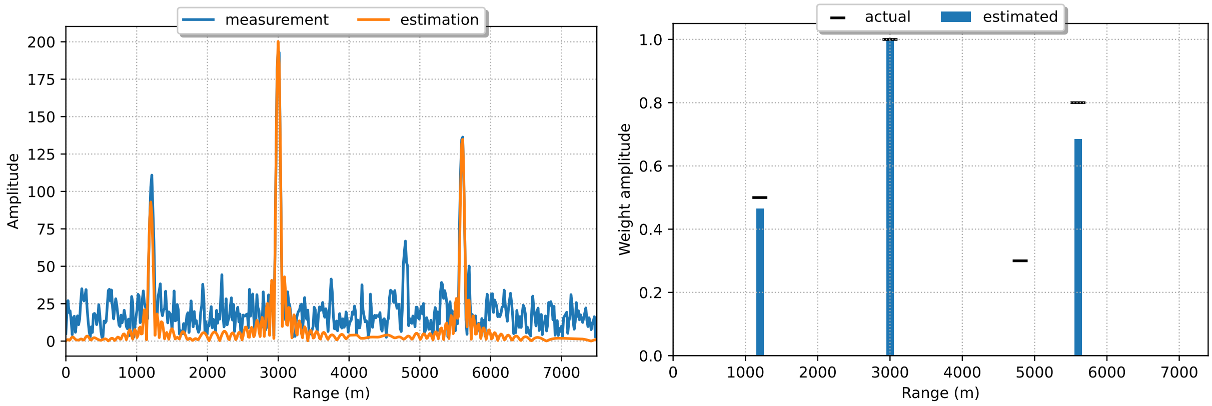

Figure 6.5: The results of using the proposed method with a higher order assumption in Burg's method than the actual AR process model order. The target vector is made out of four targets located at 1.2, 3.0, 4.8 and 5.6 km. The target vector is embedded in an autoregressive clutter process of order 1, with AR coefficient $a_1^1 = -0.95 \times \left(\frac{1}{2\sqrt{2}} + \frac{1}{\sqrt{2}}j\right)$. The expected model order given to Burg's method is set to 3. The figures above show how the proposed method with the wrong model order assumption is not able to resolve the small target at 4.8 km.

of which [36] presents an overview of in relation to Burg's method and certain model orders. This has not been implemented here and is left as possible future work. For the remainder of this work, we will assume that we have the correct knowledge of the AR model order available as an input into Burg's method.

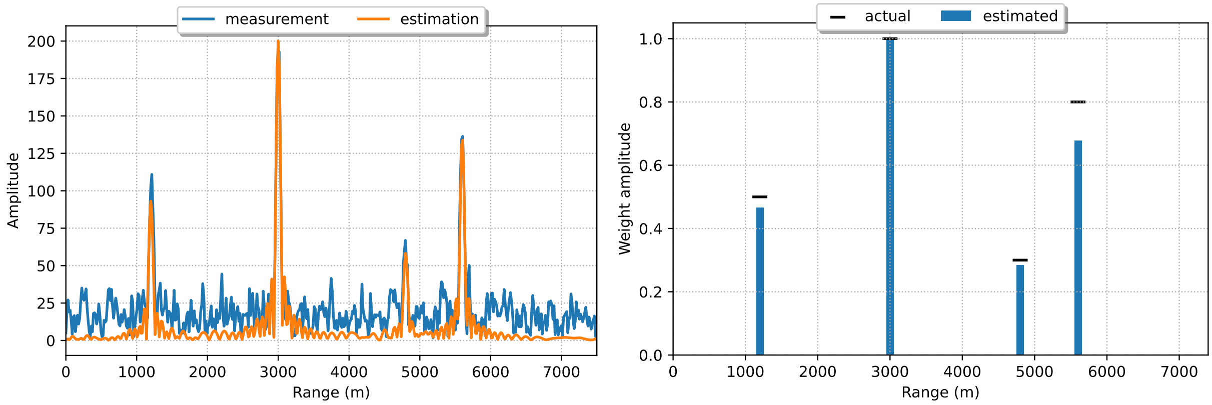

Figure 6.6: The results of using the proposed method with the correct order assumption in Burg's method based on the actual AR process model order. The measurement vector is the exact same realisation as shown in Figure 6.5. The expected model order given to Burg's method is now set to 1. The figures above show how the proposed method now is able to resolve the small target at 4.8 km due to having the correct order assumption.

6.2.3. Off-grid analysis in RVM

The relevance vector machine is inherently a grid-based technique due to its way of defining the basis functions on a pre-specified grid of possible locations. However, in any real measurement, there is no guarantee that the location of these targets perfectly align with the pre-defined grid. To show this situation, we have simulated a scenario where the actual target is in between two grid points. To make this easier, the spacing between the grid points is doubled, resulting in a grid with available locations for targets spaced 200 meters apart. Again, a 4-target measurement vector is created, with targets present at ranges of 800 meters, 3.05 km, 4.8 km and 6.3 km. These

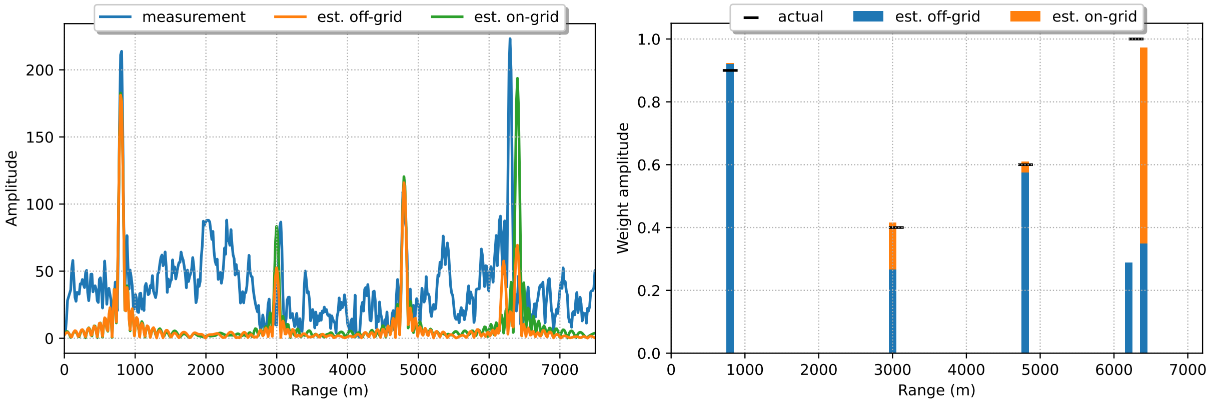

Figure 6.7: The figures above show the differences in the estimation made by the proposed framework whenever a target is located off-grid versus on-grid. The targets located at 3.05 km and 6.3 km do not align perfectly with the predefined grid. These targets were moved to 3.0 km and 6.4 km for the on-grid data shown. The figure clearly shows the destructive effect it can have on the estimation, especially when a target is in the middle of two grid points, like with the 4th target, which now only gets an amplitude of 0.3 assigned, while in reality, it was the strongest target present.

targets have been assigned an amplitude of 0.9, 0.4, 0.6 and 1.0, respectively. This implies that we have two targets that do not perfectly align with the grid: the second target is off by 50 meters, a quarter of the distance between grid points, and the fourth target is perfectly in the middle, 100 meters away from the nearest grid point. For the clutter process, a first-order AR process with AR coefficient $a_1^1 = -0.95$ is generated and added to the targets.

The results of this scenario are shown in Figure 6.7. From these figures, we can clearly see the effect that an off-grid target location has on the estimation. The first thing we can note is the difference in the weight assigned by the RVM to the basis function of the second target. It is still estimated correctly that there is a target present at around 3.0 km, but the estimated weight is seen as only around 0.3, where the actual value should be 0.4. For the last target, the results are even more extreme. As the target is placed precisely between the two grid points, the estimation is now made that there is a target present in both of these nearest range bins. As a result, less than half of the actual weight is assigned to the individual ranges, which makes the strongest reflection in the return signal be estimated on the same level as the other off-grid target at 3 kilometres.

To better visualize the difference that the off-grid location makes relative to an on-grid target, the comparison with the on-grid estimation is shown in the same figures of Figure 6.7. For these, the off-grid targets have been moved to exactly 3.0 km and 6.4 km, respectively. It can then clearly be seen how the weights are now assigned as would be expected from the targets that are present in the measurement vector.

In the remainder of this work, we will not consider the case of off-grid targets any further. Multiple works, for example, [25], [51], [52], have discussed the off-grid problem in different settings before, allowing us to compensate for these situations. As possible future work, the proposed framework could be extended to incorporate these solutions to ensure the performance stays satisfactory in these kinds of situations.

6.2.4. Higher order AR models

So far, we have only investigated the simple first-order AR(1) models. However, as shown in Section 3.4.2, surface clutter is often fairly accurately defined as an autoregressive process of the second or third order. Therefore, we will now also investigate some scenarios to see how the proposed method compares to the original method for that kind of clutter process.

Simple AR(2) scenario

For this test case, we have embedded our four targets in a second-order autoregressive clutter process, with the autoregressive coefficients

$$\begin{aligned} a_1^2 &= -0.95 \\ a_2^2 &= 0.4 \times \left(\frac{\sqrt{3}}{2} - \frac{1}{2} j \right) \end{aligned} \tag{6.4}$$

The targets are located at 800 m, 3.0 km, 4.8 km and 6.3 km, with amplitudes of 0.9, 0.4, 0.6 and 1.0, respectively. The results of the original implementation using white noise assumption are shown in Figure 6.8, and the proposed method is shown in Figure 6.9. From these results, we can see that the implementation using the white noise assumption had a great deal of difficulty in trying to find the correct basis functions. Especially the target located at 4.8 km, where we can see the RVM used a lot of basis functions from consecutive range bins with lower weights to attempt to match the measurement. Also, some false detections are present at around 2.4 km and around 7.0 km. In the end, the original implementation utilized 13 vectors to make the estimation of the measurement.

For the proposed method, we set up the AR parameter estimation using Burg's method to now estimate a second order, assuming the correct knowledge of the model order in advance as discussed in Section 6.2.2. The output of the proposed method, as shown in Figure 6.9, only consists of the expected 4 basis functions, having a false detection at 500 meters. Also, the target present at 3 km happens to be predicted in the adjacent range bin instead of its correct one. We also see this come into effect in the sense that the estimated weight of the corresponding basis function is a fair way off of the intended value of 0.4. However, even with these slight deviations from a perfect estimation, the proposed method manages to drastically reduce the amount of false detections in the output. The AR coefficients that were estimated by the proposed method were

$$\begin{aligned} \hat{a}_1^2 &= -0.893 - 0.079j \\ \hat{a}_2^2 &= 0.305 - 0.141j \end{aligned} \tag{6.5}$$

which is a decent approximation of the expected values of $\mathbf{a}^2 = \begin{bmatrix} -0.95 & 0.346 - 0.200j \end{bmatrix}$. Even though the estimation of the AR coefficients is not perfect, we do see that it helps the relevance vector machine almost completely remove all of the false detections, as well as provide more certainty of the estimated targets.

AR(2) scenario with low SCR

To also investigate how well the proposed method could work under more extreme clutter conditions, the following tests were performed with a signal-to-clutter ratio (SCR) of -6 dB. Again, we take a second-order AR clutter process, this time with different AR coefficients

$$\begin{aligned} a_1^2 &= 0.95 \times \left(\frac{1}{2\sqrt{2}} + \frac{1}{\sqrt{2}} j \right) \\ a_2^2 &= -0.5 \end{aligned} \tag{6.6}$$

and the targets were distributed at 1.1 km, 2.4 km, 5.0 km and 5.6 km, with amplitudes of 0.5, 0.7, 0.3 and 1.0. The different estimations made by the two implementations can be seen in Figures 6.10 and 6.11. Again, the original implementation is able to correctly recognise the four targets in the clutter. However, it has some trouble in explaining the rest of the noise, leading to an estimated model in which 15 basis functions are included, leading to false detections around

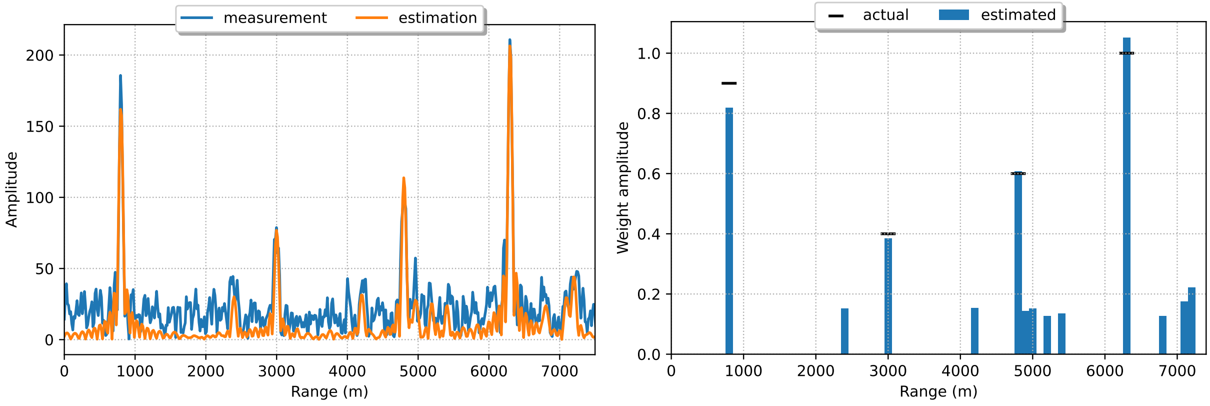

Figure 6.8: The results of processing an AR(2) process using the original RVM implementation with white noise assumption. The targets are located at 800 m, 3.0 km, 4.8 km and 6.3 km. The clutter process is generated using the AR coefficients $a_1^2 = -0.95$ and $a_2^2 = 0.4 \times \left(\frac{\sqrt{3}}{2} - \frac{1}{2}j\right)$. We see the predicted target vector amplitude based on the estimated model on the left. On the right, we see the amplitude of the weights that compose the estimated model. The RVM used 13 basis functions to predict the target vector based on the measurement.

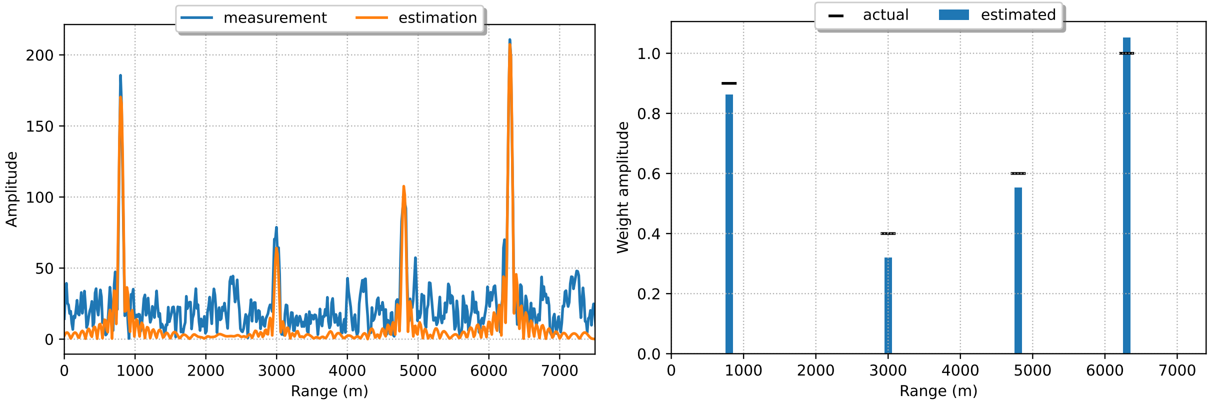

Figure 6.9: The results of processing the same AR(2) process as in Figure 6.8 using the proposed RVM implementation based on AR modelling. Again, we see the predicted target vector amplitude based on the estimated model on the right and the amplitude of the weights composing the model on the right. The proposed RVM implementation used 4 basis functions to predict the target vector based on the measurement.

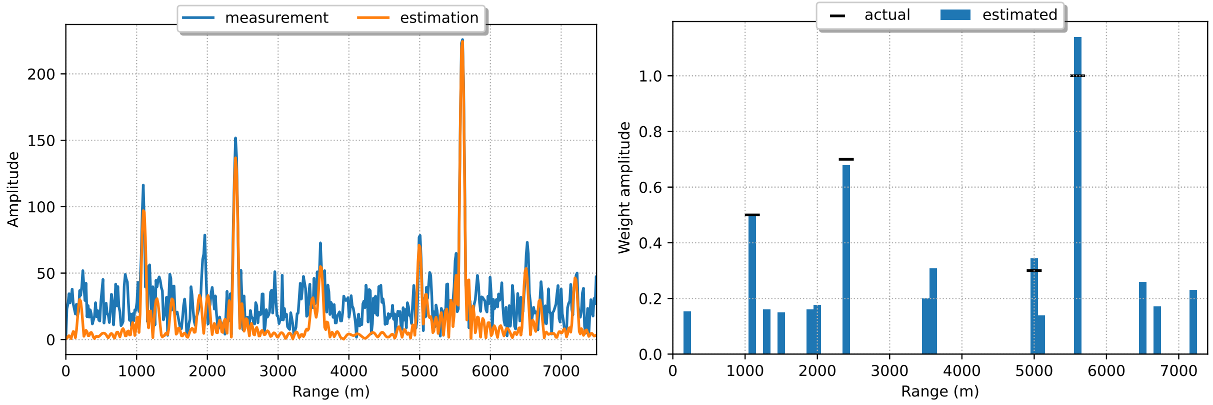

Figure 6.10: The results of processing an AR(2) process using the original RVM implementation with white noise assumption. Here the signal to clutter ratio (SCR) is reduced to -6dB. The targets are located at 1.1 km, 2.4 km, 5.0 km and 5.6 km. The clutter process is generated using the same AR coefficients as before, $a_1^2 = -0.95$ and $a_2^2 = 0.4 \times \left(\frac{\sqrt{3}}{2} - \frac{1}{2}j\right)$. We see the predicted target vector amplitude based on the estimated model on the left. On the right, we see the amplitude of the weights that compose the estimated model. The RVM used 15 basis functions to predict the target vector based on the measurement.

3.5 km and around 7.0 km with similar amplitudes as the target at 5.0 km. The proposed method, however, manages to create an estimated model without any of these false detections. The predicted AR coefficients are

$$\begin{aligned}\hat{a}_1^2 &= 0.381 + 0.596j \\ \hat{a}_2^2 &= -0.408 + 0.001j\end{aligned} \tag{6.7}$$

which are a fair approximation of the actual coefficients $\mathbf{a}^2 = \begin{bmatrix}0.336 + 0.672j & -0.5\end{bmatrix}$. Burg's method correctly estimates that the second AR coefficient is a purely real parameter, although the magnitude is off somewhat. It is not a perfect reflection of the autoregressive clutter process, but it enables the relevance vector framework to find enough quality in certain basis functions to include only the correct targets in the estimation.

To see how well both methods fare under even worse SCR conditions, another simulation was performed where the SCR was further reduced to -12 dB. The AR coefficients for the clutter were defined as

$$\begin{aligned}a_1^2 &= 0.8 \times \left(-\frac{\sqrt{3}}{2} + \frac{1}{2}j\right) = -0.693 + 0.4j \\ a_2^2 &= 0.3\end{aligned} \tag{6.8}$$

These results are shown in Figures 6.12 and 6.13. From these figures, we see how the original implementation is now unable to detect two of the targets, only providing an estimate for two of the four targets. For the proposed method, we see this is able to uncover one additional target by using the estimated clutter, leaving only one target as a missed detection. The AR coefficients estimated by the proposed method are

$$\begin{aligned}\hat{a}_1^2 &= -0.651 + 0.432j \\ \hat{a}_2^2 &= 0.217 - 0.001j\end{aligned} \tag{6.9}$$

which is an accurate estimate of the known AR coefficients.

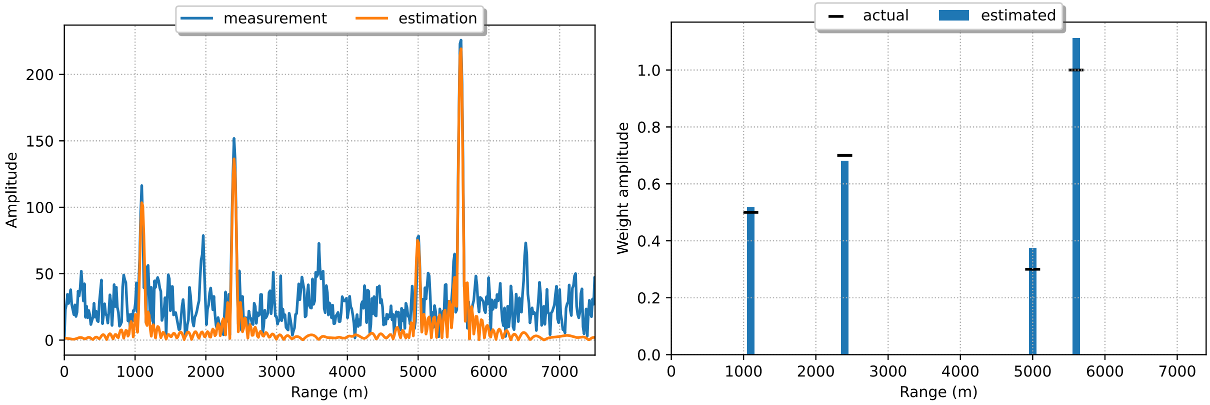

Figure 6.11: The results of processing the same AR(2) process as in Figure 6.10 using the proposed RVM implementation based on AR modelling. Again, we see the predicted target vector amplitude based on the estimated model on the right and the amplitude of the weights composing the model on the right. The proposed RVM implementation used 4 basis functions to predict the target vector based on the measurement.

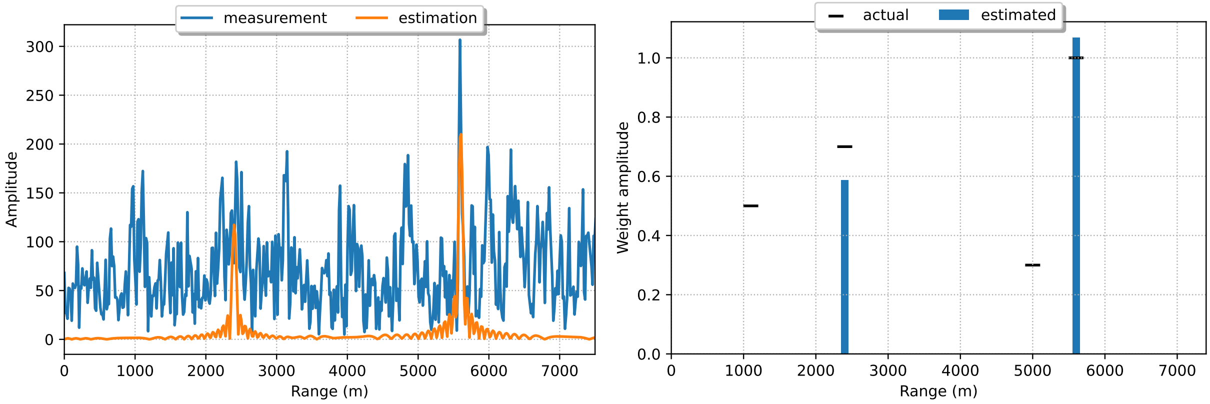

Figure 6.12: The results of processing 4 targets buried in AR(2) clutter using the original RVM implementation, where the signal-to-clutter (SCR) ratio is -12 dB. The targets are still located at 1.1 km, 2.4 km, 5.0 km and 5.6 km. The AR(2) clutter process was generated using the AR coefficients $a_1^2 = 0.8 \times \left(-\frac{\sqrt{3}}{2} + \frac{1}{2}j\right)$ and $a_2^2 = 0.3$. From the figures above, we see that the RVM used 2 basis functions to predict the target vector based on the measurement, leading to two missed detections.

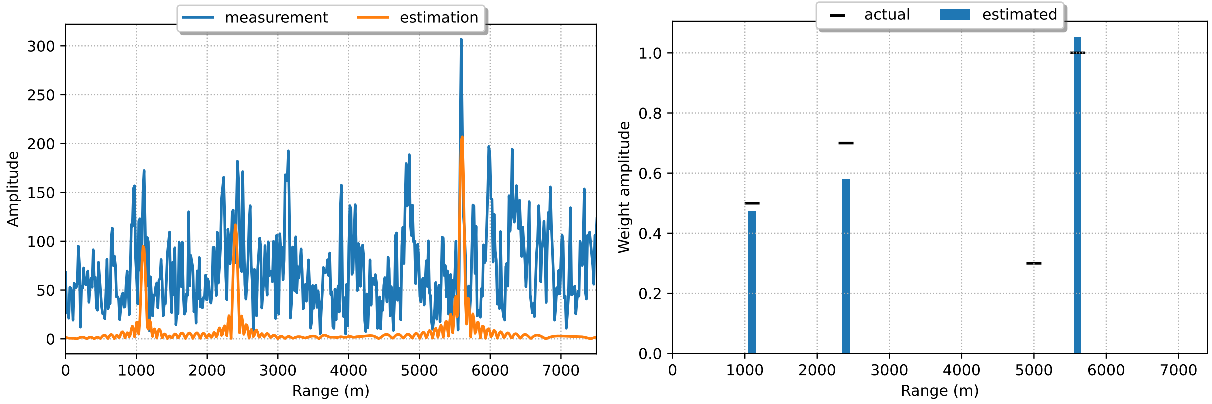

Figure 6.13: The results of processing the same AR(2) process as in Figure 6.12 using the proposed RVM implementation based on AR modelling. Again, we see the predicted target vector amplitude based on the estimated model on the left and the amplitude of the weights composing the model on the right. The proposed RVM implementation used 3 basis functions to predict the target vector based on the measurement, leading to a missed detection of the lowest amplitude target.

Low SCR scenario with AR(3)

To complete the tests with higher-order autoregressive clutter processes, the same SCR of -12 dB as before is used in conjunction with a third-order autoregressive clutter. For this test, the clutter parameters are taken as

$$\begin{aligned} a_1^3 &= -0.5 \times \left(\frac{\sqrt{3}}{2} - \frac{1}{2}j\right) = -0.433 + 0.5j \\ a_2^3 &= -0.15 \\ a_3^3 &= 0.05 \end{aligned} \tag{6.10}$$

with the same target distribution and amplitudes as in the previous low SCR AR(2) test. The results of this test case are presented in Figures 6.14 and 6.15. Again, we managed to reduce the number of false detections and return the actual targets embedded in the measurement vector. For the autoregressive process, the following set of coefficients was estimated:

$$\begin{aligned} \hat{a}_1^3 &= -0.451 + 0.499j \\ \hat{a}_2^3 &= -0.127 + 0.014j \\ \hat{a}_3^3 &= 0.042 - 0.013j \end{aligned} \tag{6.11}$$

This shows us that Burg's method is able to provide a reasonable estimate of the autoregressive parameters, at least up to an autoregressive model order of 3. By using this estimate, the proposed framework is able to reduce the amount of false detections and/or recover targets that are otherwise buried in the clutter.

6.3. One-dimensional direction-of-arrival scenario

As we already eluded to in Section 6.2, the previous examples have been fairly fictitious ones to simplify the visualisation of certain aspects of the relevance vector machine framework in general, as well as the consequences the proposed method can have on the estimation being formed. However, we also want to show the framework's applicability to a common surveillance radar-related problem that actually incurs a clutter that can be modelled as an autoregressive process.

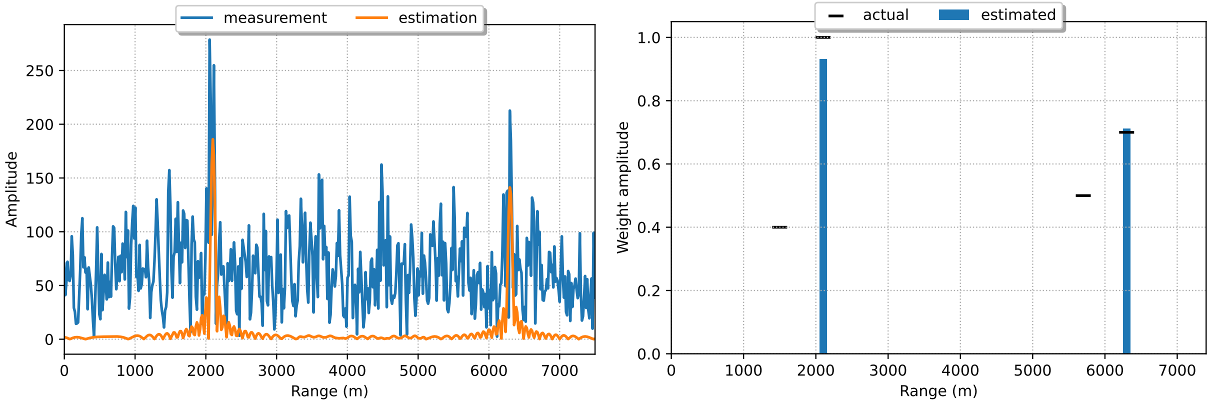

Figure 6.14: The results of processing 4 targets using the original RVM implementation, this time buried in AR(3) clutter. The targets are located at 1.5 km, 2.1 km, 5.7 km and 6.3 km, with amplitudes of 0.4, 1.0, 0.5 and 0.7, respectively. The signal-to-clutter ratio (SCR) is still -12 dB. The AR(3) clutter process was generated using AR coefficients $a_1^3 = -0.5 \times \left(\frac{\sqrt{3}}{2} - \frac{1}{2}j\right)$, $a_2^3 = -0.15$ and $a_3^3 = 0.05$. From the figures above, we see that the RVM used 2 basis functions to predict the target vector based on the measurement.

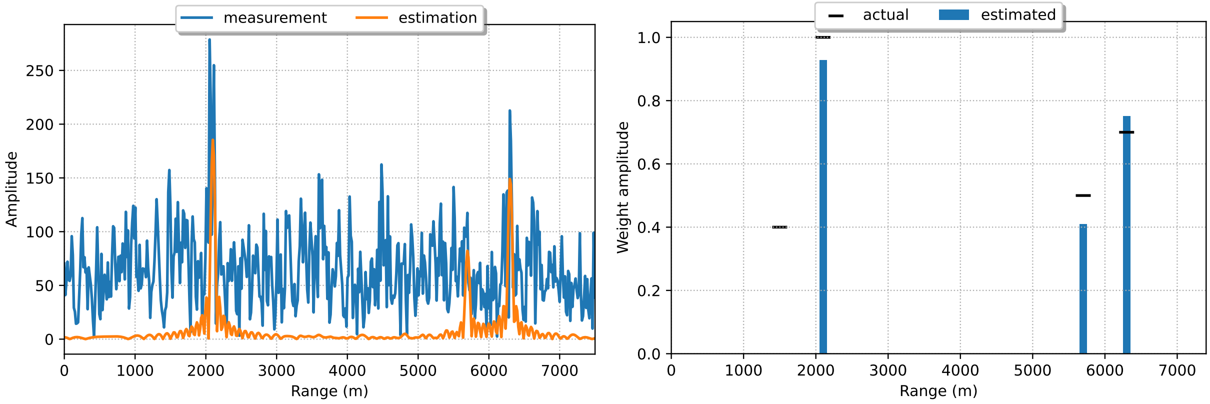

Figure 6.15: The results of processing the same AR(3) process as in Figure 6.14 using the proposed RVM implementation based on AR modelling. Again, we see the predicted target vector amplitude based on the estimated model on the left and the amplitude of the weights composing the model on the right. The proposed RVM implementation used 3 basis functions to predict the target vector based on the measurement, leading to a missed detection of the lowest amplitude target.

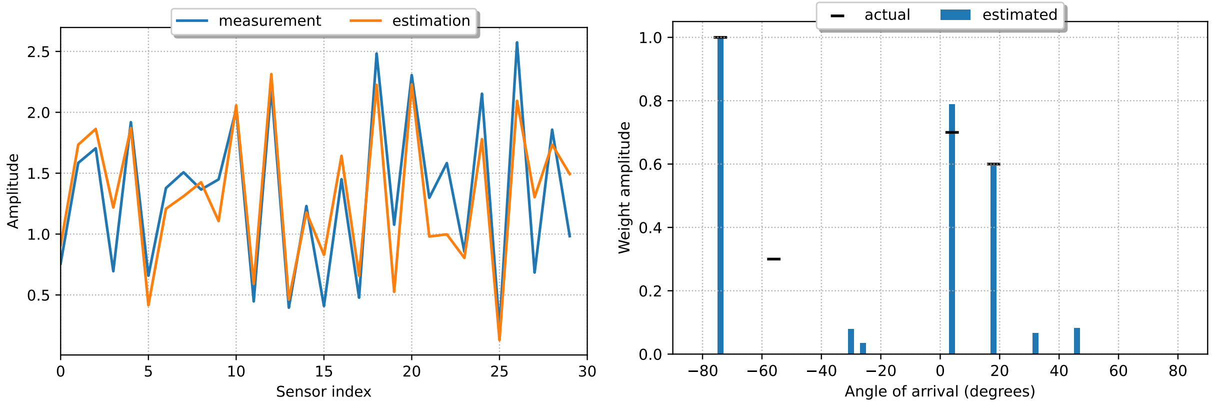

Figure 6.16: The results of the original implementation for DoA-estimation in a measurement vector containing 4 targets embedded in AR(1) clutter. The measurement contains 4 targets at 4°, 18°, -56°and -74°. The measurement is embedded in an AR(1) clutter process with AR coefficient $a_1^1 = -0.95$. From the figures above, we can see the original RVM implementation fails to properly resolve one of the four angles present, so it chose a low-weight combination of four other angles instead. In total, the model consists of 7 angles.

For this reason, we implement a multi-target direction-of-arrival (DoA) detection problem. Instead of a measurement vector corresponding to different points in time, we will work with a vector where every value corresponds to the received signal at every antenna element on a uniform linear array (ULA) at a single point in time. We use the expected phase change vectors described in Section 2.2 as our basis functions. The dictionary is then composed by generating these basis functions for all angles in the range $\theta_l \in [-90°, 90°]$ with the grid points being spaced by $\Delta\theta = 2°$.

For these tests, a random selection would be made of 4 angles available in the dictionary, and a measurement would be generated by taking a constant definition of amplitude and phase delay for four different targets with randomly assigned directions of arrival. This results in a vector of target directions defined as

$$\boldsymbol{\theta}_t = \begin{bmatrix} 0.7\,\text{exp}\left\{j\frac{1}{2}\pi\right\}\text{exp}\left\{jk_0\right\} \\ 0.6\,\text{exp}\left\{j2\pi\right\}\text{exp}\left\{jk_1\right\} \\ 0.1\,\text{exp}\left\{j2\pi\right\}\text{exp}\left\{jk_2\right\} \\ 1.0\,\text{exp}\left\{j1\pi\right\}\text{exp}\left\{jk_3\right\} \end{bmatrix} \tag{6.12}$$

where $k_i, i = 0, \ldots, 3$ is determined using Equation 2.4 based on a random angle in the dictionary. This target vector is then embedded in the first-order autoregressive clutter process vector, with AR coefficient $a_1^1 = -0.95$.

Figures 6.16 and 6.17 show the result for a measurement where the angles of the the targets were generated to be $\phi = \begin{bmatrix} 72 & 26 & -10 & -16 \end{bmatrix}$, in degrees.

We can note that both implementations could provide an estimate of the directions of arrival with varying accuracy. The original implementation based its estimation on 7 angles while missing the actual targets that were located at 26 and -10 degrees, therefore having 5 false detections and two missed detections. The proposed framework could estimate 3 angles, which corresponded correctly to the different targets, but also had a missed detection for the target located at -10 degrees. The AR(1) coefficient estimated by Burg's method was $\hat{a}_1^1 = -0.927 + 0.015j$, which is a good estimate given the few samples that are available.

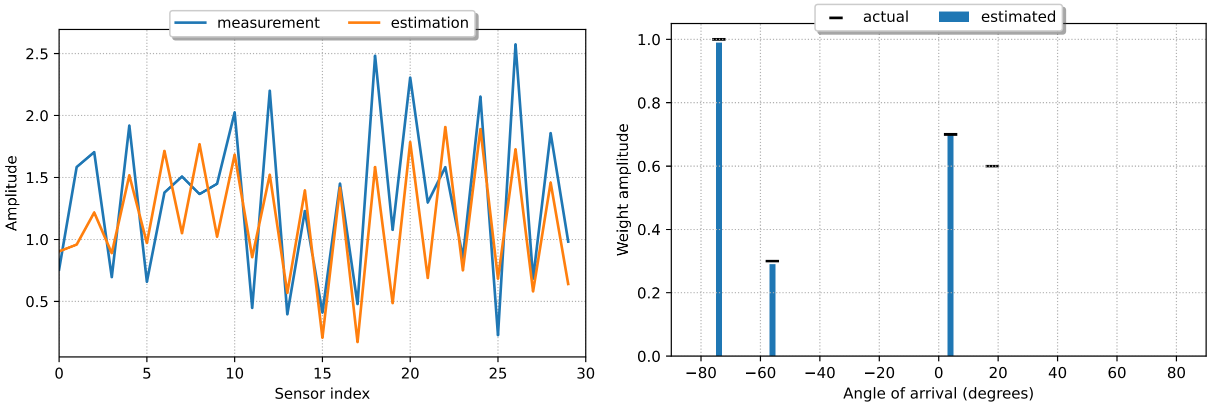

Figure 6.17: The results of the proposed method for the DoA-estimation shown in Figure 6.16. Here, we see that the model contains three correctly predicted targets, but a missed detection for the target located at 18°. In total, only three angles were included in the model.

In this first example, the proposed method did a fairly decent job of performing an estimate. However, we did encounter that it was hardly ever the case in general. More commonly, the proposed method would only predict a single target to be present in the data and then leave the estimation untouched over the next iterations of the algorithm. An example of this is shown in Figures 6.18 and 6.19. In this case, the angle vector used to create the measurement was $\phi = \begin{bmatrix} -18^\circ & -68^\circ & 44^\circ & 36^\circ \end{bmatrix}$. A different realisation of the same AR process as before was used, so the AR(1) coefficient still was $a_1^1 = -0.95$.

From the Figures 6.18 and 6.19, we see that both implementations fail to handle this specific case well. However, the original implementation at least manages to create an estimation based on the basis functions that resemble the measurement in terms of amplitude, albeit by using 13 vectors. The 13 basis functions comprise three perfect estimations for the targets -18°, 36°and 44°. The last target, at -68°, is missed. The remaining signal is then estimated using some angles with very extreme weights, way exceeding the expected amplitude of 1.0.

On the other hand, the proposed method only estimated a single direction. After selecting this angle and estimating the autoregressive parameter, the estimation would no longer be updated by adding other angles for the entire iteration process. We can also see this effect in the estimated autoregressive coefficient:

$$\hat{a}_1^1 = 0.353 - 0.607j \tag{6.13}$$

which is not even remotely close to the clutter process's actual autoregressive coefficient. This occurs very commonly, independently of the exact locations where the targets would be simulated. The same problem also persists for the higher-order AR(2) and AR(3) clutters that were tested.

The problem at hand can be described by the autoregressive nature of the basis functions we are processing in this scenario. Figure 6.20 shows the autocorrelation function of the basis functions for three different angles for lags 0 through 5. From this, we can see that we can commonly expect a fairly high autocorrelation value at low lags. This also implies that at least an approximation of the basis function can be made by trying to model it as an autoregressive process. Especially when we are supplied a summation of these basis functions alongside an already present autoregressive process, this causes issues where the autoregressive parameters are incorrectly estimated due to the influence of the targets in the measurement vector.

To mitigate this problem, we propose an extension to handle multiple measurement vectors. The idea is that the amount of targets present is sparse, therefore there will be plenty of range bins present with no target presence. Using this information as additional segments in the Burg's

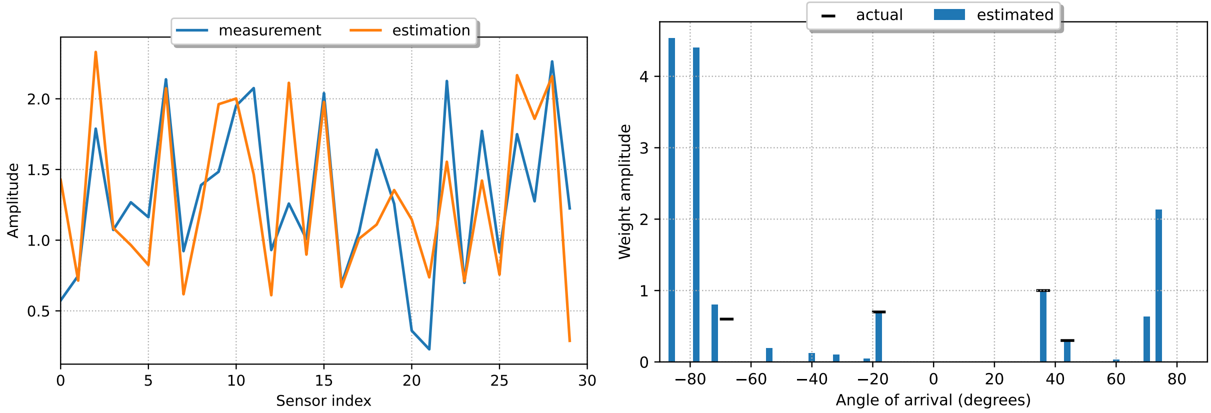

Figure 6.18: DoA-estimation using the original implementation for another measurement vector containing 4 targets embedded in AR(1) clutter, similar to the measurement in Figure 6.16. The targets are located at -18°, -68°, 44°and 36°. The original implementation manages to estimate 3 angles correctly but also includes very strong false detections in an attempt to estimate the large target.

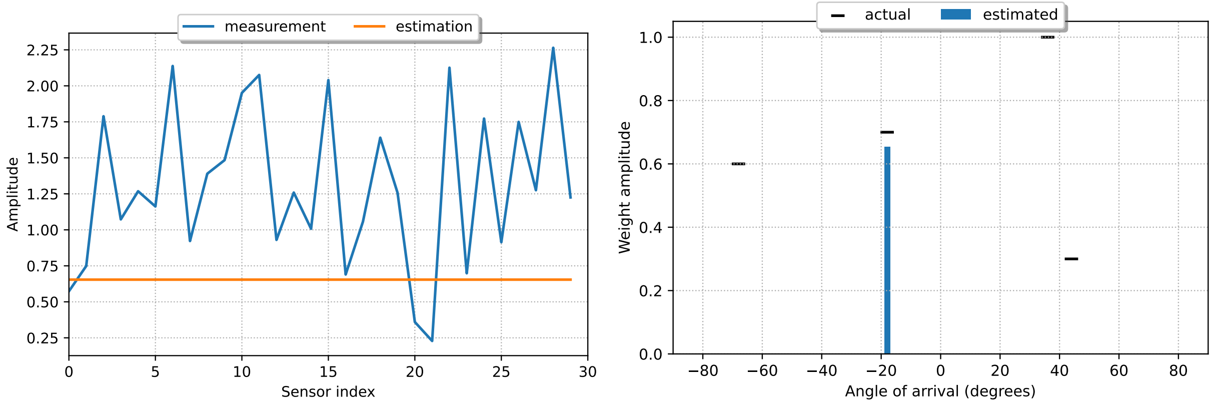

Figure 6.19: The same DoA-estimation as Figure 6.18 for a measurement vector containing 4 targets embedded in AR(1) clutter, using the proposed method. We see how the proposed method completely fails to resolve three out of four targets, classifying them as the AR clutter process instead.

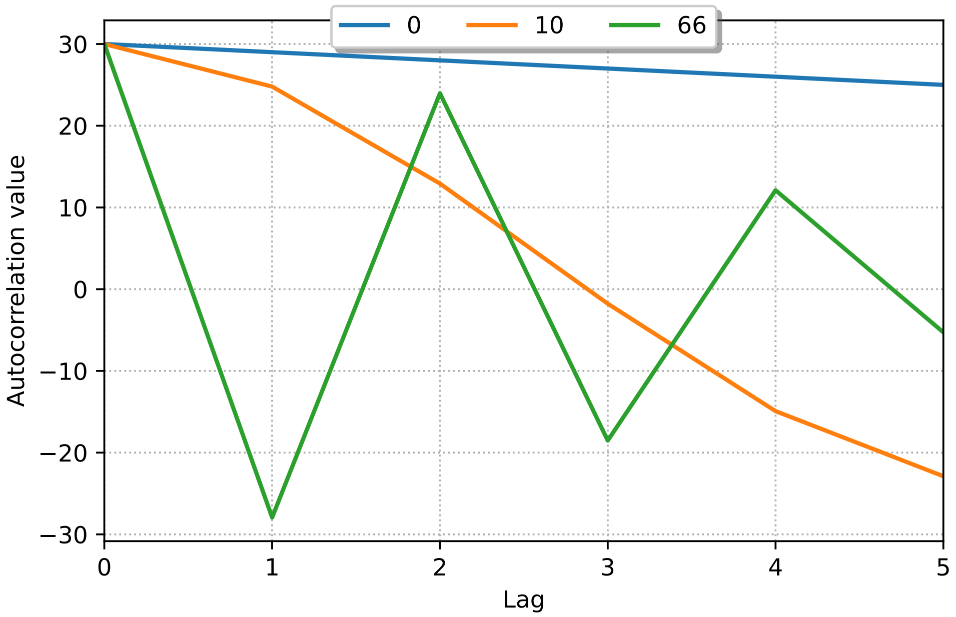

Figure 6.20: The values of the autocorrelation function for three different basis functions corresponding to angles of 0°, 10°and 66°.

method for segments algorithm, we can provide a better estimate of the clutter, mitigating the case where the target itself is classified as clutter.

6.4. Two-dimensional range-angle scenario

For the multiple measurement vector setup, we create a single measurement vector as per Section 6.3 above. We also generate 19 separate clutter realisations using the same AR parameters, which holds for the assumption that the clutter is homogeneous over the range bins considered. The complete measurement vector is made by stacking all these vectors into a single column vector.

Beforehand we are unaware at which range we expect the targets to be present. Therefore, to allow for all of these options, the dictionary used by RVM is expanded. We start off by taking the additional dictionary, as also used in the single measurement case before, but extend each basis function with zeros to match the now longer length of the measurement vector. To allow for any arbitrary range bin to contain a measurement from a certain angle, we also replicate the zero-padded dictionary as many times as we have measurement vectors, shifting the samples to make the basis function correspond to a target in a different range bin. As a result, our dictionary that used to be $\mathbf{\Phi} \in \mathbb{C}^{M\times L}$, with M the number of antenna elements in the array and L the number of angles being considered, now has become $\mathbf{\Phi} \in \mathbb{C}^{(MN)\times(LN)}$, where N is the number of measurement vectors available.

This setup allows us to apply the RVM without any prior knowledge of the presence of a target in any specific range bin, as the dictionary contains basis functions for all range bins. One change needs to be made to the processing, however, in regards to how the clutter AR parameters are being estimated. The original implementation supplied the entire measurement vector minus the estimated model, $\mathbf{t} - \mathbf{\Phi}\hat{\mu}$ to Burg's method, along with the additional segments from the $\mathbf{C}$ matrix. In the current case, however, with the measurement vectors being stacked, our vector now spans multiple shorter clutter process realisations. Within these separate range-bin-specific realisations, the same clutter covariance matrix is valid. But this does not apply to the large stacked vector. Therefore, to ensure the clutter AR parameters are estimated correctly, the vector is separated

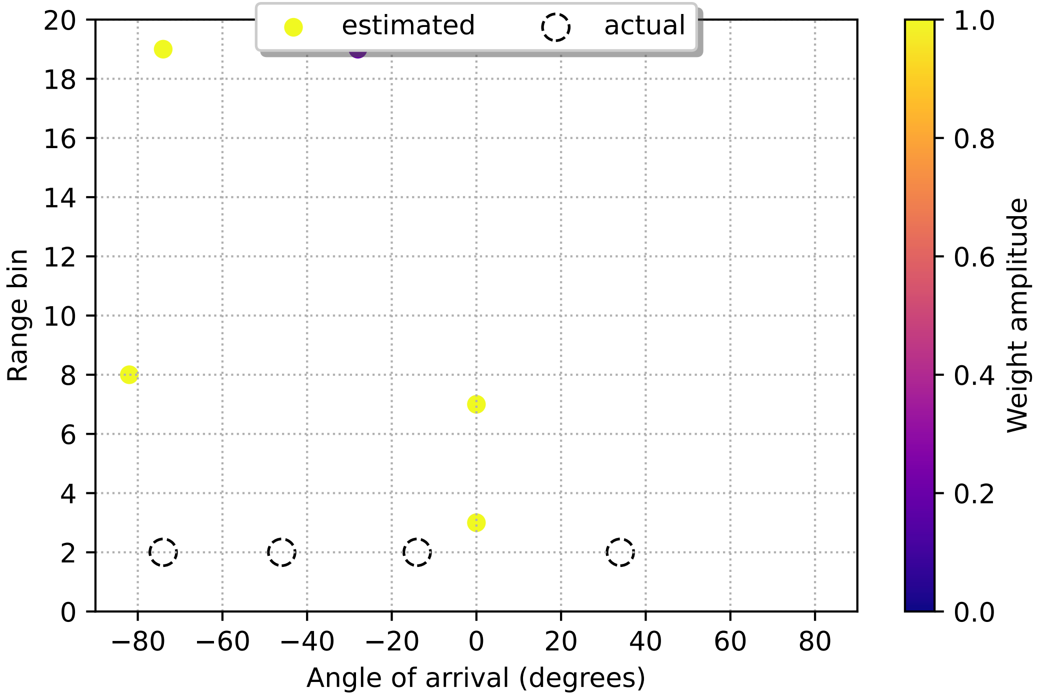

Figure 6.21: The estimation of targets by the original implementation. Four well-separated targets are present in the range bin with index 2. The original implementation failed to resolve any of the targets, yielding 5 false detections and 4 missed detections.

into separate vectors per range-bin before it is being processed by Burg's method for multiple segments, which now has N times as many segments to process.

The initial results of using this framework to process multiple range bins are shown in Figures 6.21 and 6.22. In this case, four targets are still contained within the measurement vector, similar to the single vector cases in 6.3. Here, we have our targets present in the range bin with index 2, with targets present at angles -74°, -46°, 34°and -14°, with amplitudes of 0.7, 0.6, 0.3 and 1.0, respectively. The clutter process in which the targets are embedded, as well as the clutter-only realisations for the other range bins, is an AR(1) process with AR coefficient $a_1^1 = -0.95$.

In Figure 6.21 we see that the original implementation fails to find the targets in the second range bin, and ends up estimating 5 targets in total distributed across the available range bins. The proposed method ends up estimating even more targets, 11 in total, which is shown in Figure 6.22. However based on the additional knowledge of the clutter process, the proposed method now is able to perfectly resolve the targets present in the second range bin. The weights have not been estimated perfectly, for example the target at -14°should have been bright yellow, but their presence and approximate values are estimated correctly. The proposed method does end up with seven false detections in the other range bins however. This is most likely due to the same similarity between the basis functions and the clutter, which leads to the RVM optimizing the error between the actual clutter parameter and the estimated clutter parameter using the available basis functions. Most notably, the 0°basis function can be used as a bias term, and other low angle basis functions can be utilised to 'detrend' the data, which is possibly why RVM ends up adding these to the estimated model. However, the estimated AR coefficient for this case was fairly accurate, being estimated at $\hat{a}_1^1 = 0.951 - 0.015j$.

Using this method, however, the proposed method cannot perfectly reconstruct the target vector every time. This is shown in Figures 6.23 and 6.24. Here, the targets are located at range bin index 5, with the targets present at angles 60°, 52°, 6°and 88°. The amplitudes are distributed

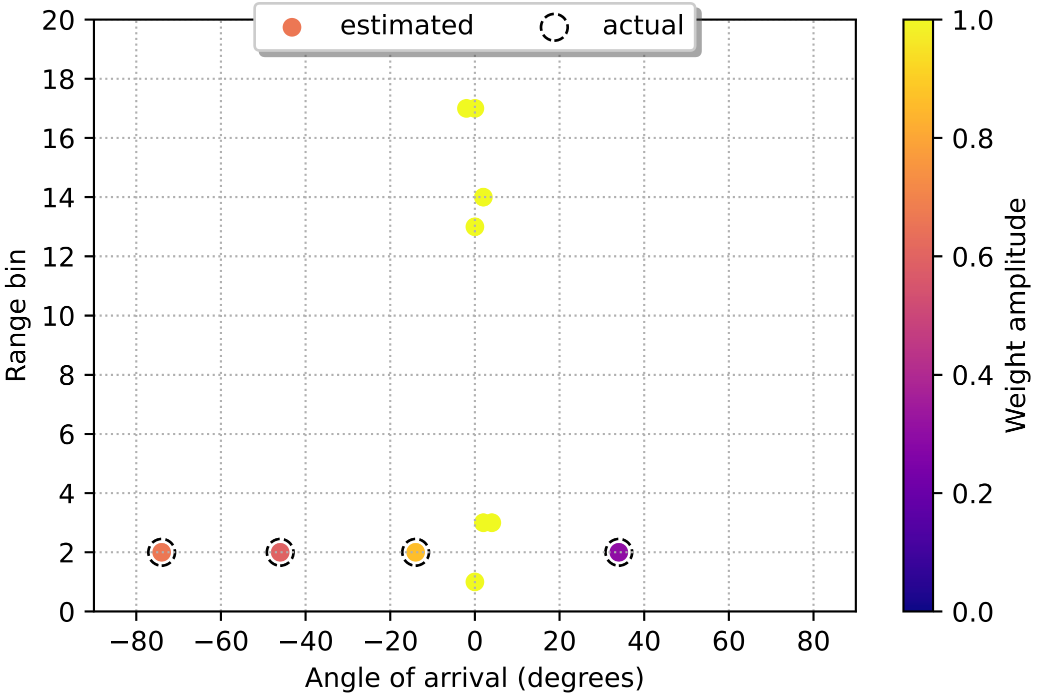

Figure 6.22: The estimation of targets by the proposed method. It perfectly reconstructs the four targets in the second range bin. It, however, also estimates some false detections at other ranges, resulting in an estimated model consisting of 11 targets.

in the same order as before: 0.7, 0.6, 0.3 and 1.0, respectively. The clutter process is still the same, an AR(1) process with coefficient $a_1^1 = -0.95$. Again, the original implementation fails to resolve any of the targets correctly. In the proposed framework, we do get 4 targets being estimated in the 5th range bin, however only the two at 52°and 60°actually were correct. The targets at 6°88°have been placed at angles close to their actual values. This is probably due to the error that is made in the estimation of the AR coefficient, which is estimated at $\hat{a}_1^1 = -0.900 + 0.008j$. As a result of this mismatch, it probably attempts to resolve some of the clutter alongside the target and estimates a target at an angle near the actual angle.

Similar problems arise when there are targets very closely separated in angles. In Figures 6.25 and 6.26, there are 3 targets in the range bin index 2, located at angles of -16°, -18°and -78°, with amplitudes of 0.7, 1.0 and 0.3, respectively. The clutter process remains the same AR(1) process as for the previous multiple range-bin cases. Again, the original implementation seems unable to locate the actual angles of arrival in the dataset. The proposed method correctly manages to estimate the angle of the target at -78°. The two targets at -16°and -18°, however, pose a problem, where the proposed method actually estimates targets on either side of the two intended angles but not exactly on either one of them.

6.4.1. Range-angle with targets in multiple range bins

One nice feature of the framework is that it does not need any prior information regarding the target presence in a certain range bin, nor needing any guard cells around a cell under test with the assumption that the remainder of the ranges are free of targets. Therefore, in Figures 6.27 and 6.28, the results are shown of placing two targets at different ranges. We still use $N = 20$ range bins, resulting in 18 target-free clutter realisations alongside the measurements.

In this scenario, the two targets are present in the range bins 6 and 15. The target in range bin 6 is located at an angle of -82°, with an amplitude of 1.0. The target in range bin 15 is located

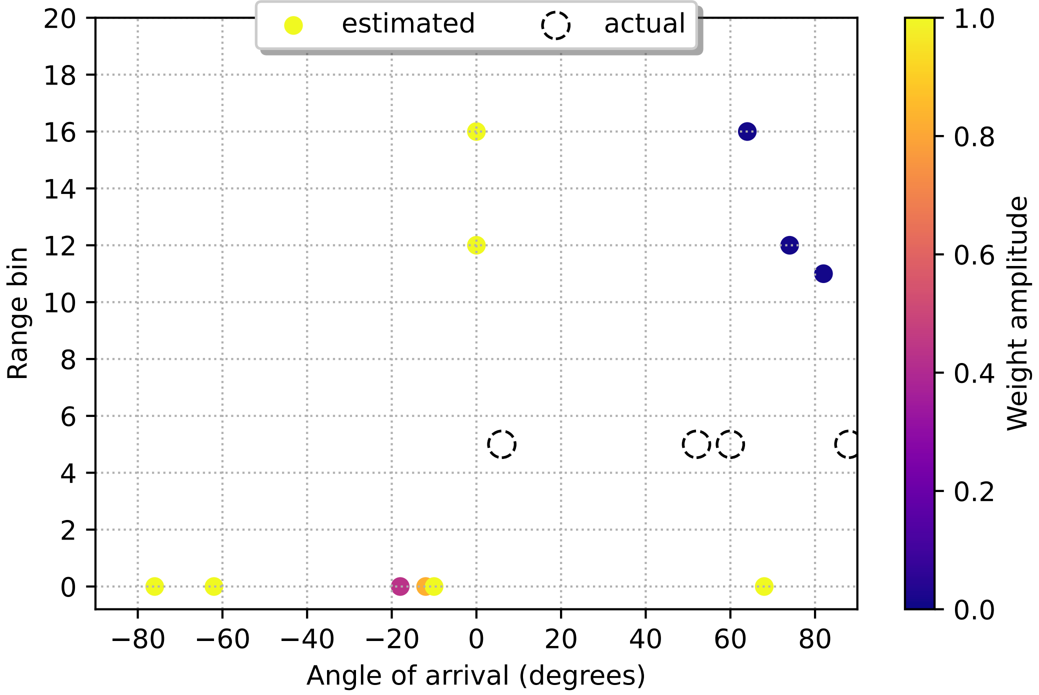

Figure 6.23: The estimation of the second multiple vector case using the original implementation. Four targets are present in the 5th range bin. They are located at 60°, 52°, 6°and 88°. The clutter present across all ranges was the same AR(1) process as in Figures 6.21 and 6.22. Again, the original implementation could not resolve any of the actual targets.

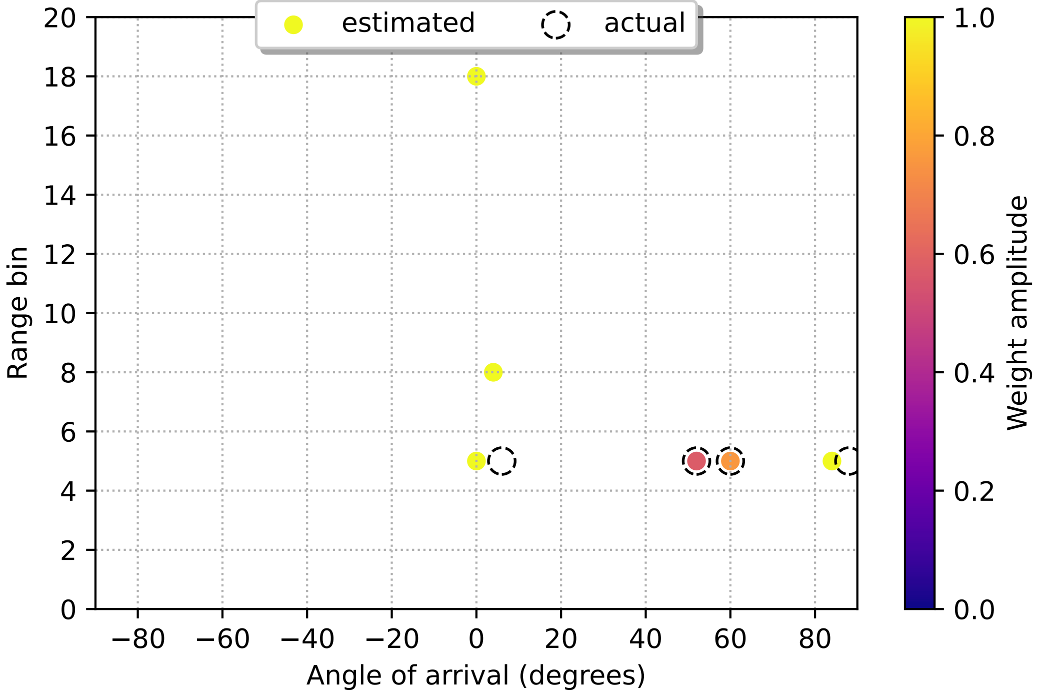

Figure 6.24: The estimation as performed by the proposed framework on the same setup as Figure 6.23. The proposed method manages to estimate four targets in the 5th range bin but is not precisely accurate on the angle estimates for the targets at 6°and 88°. Besides the actual targets, 2 additional false detections were estimated.

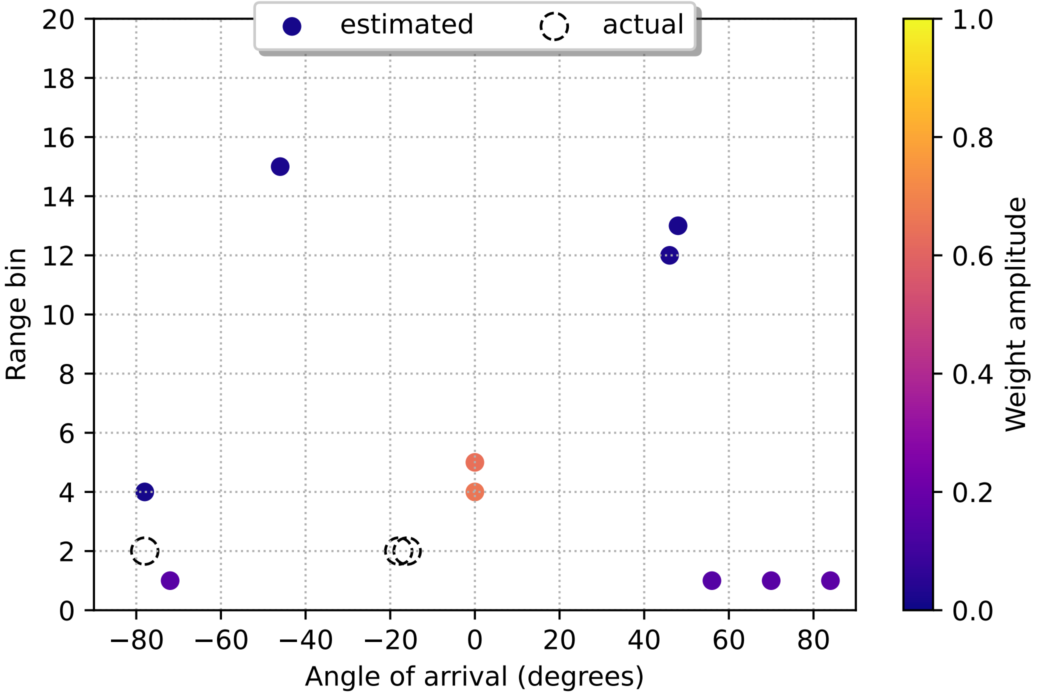

Figure 6.25: The estimation result of the original implementation when faced with three targets present in the 2nd range bin, at -16°, -18°and -78°. Again, the original implementation fails to resolve any of the present targets correctly.

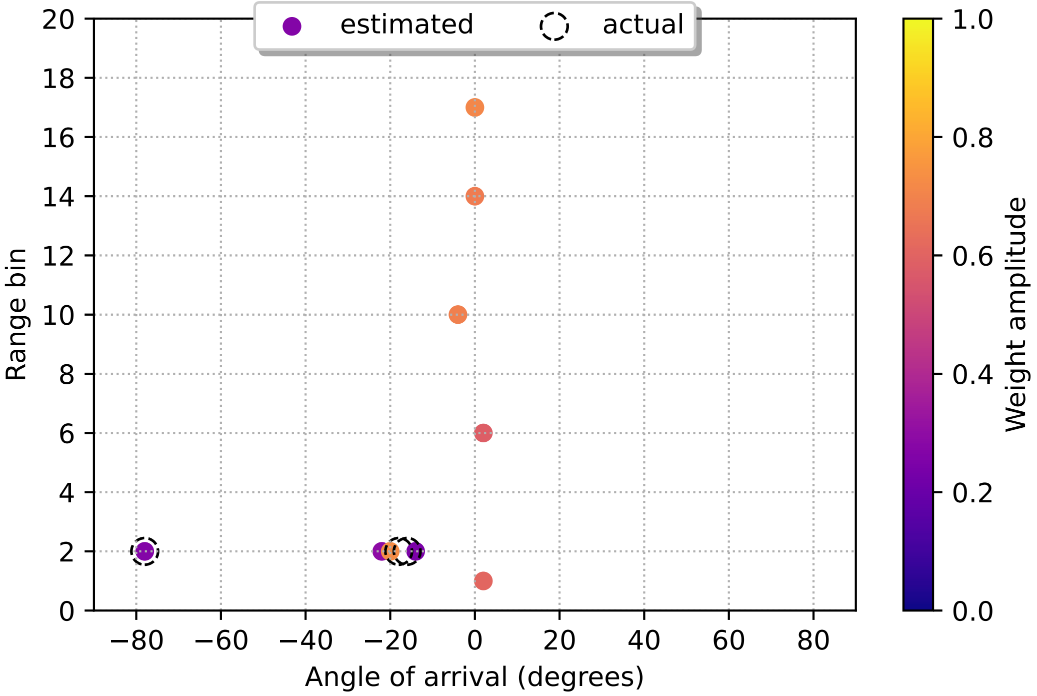

Figure 6.26: The estimation by the proposed framework for the same scenario as in Figure 6.25. The isolated target at -78°is resolved well, but the framework has troubles resolving the two targets in adjacent angle cells correctly. The final estimation consists of three angles surrounding the two actual values. Besides this, the proposed method also estimates another 5 targets in other range bins.

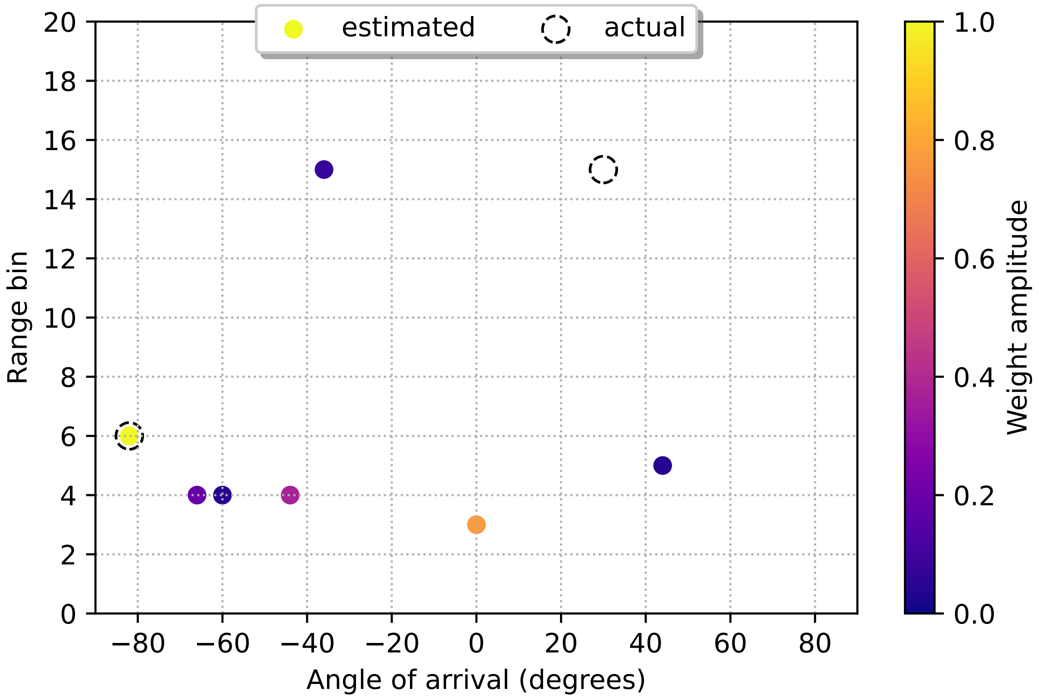

Figure 6.27: The estimation by the original implementation when given two targets separated in two different range bins. The targets are located in the 6th range bin, at -82°, and at the 15th range bin, at 30°. The targets have an amplitude of 1.0 and 0.7, respectively. The original implementation manages to resolve the target in the 6th range bin correctly but misses the target located in the 15th range bin. Additionally, 6 false detections were present in the estimation.

at an angle of 30°, with an amplitude of 0.7. Both implementations were able to correctly resolve the single target present in the 6th range bin. The original RVM however did not manage to correctly place the target at the 15th range bin, estimating a weaker target at a different angle instead. The proposed method did manage to locate both targets perfectly. The estimation of the AR coefficient resulted in an estimate of $\hat{a}_1^1 = -0.874 + 0.012j$, which led to 5 additional false detections being present in the estimated result.

As the process is agnostic of the target's position relative to each other but simply relies on enough clutter-only range bins being present to allow for the AR parameter estimation, it is also possible to perform processing on targets being present in subsequent range bins without any additional difficulty. This scenario is shown in Figures 6.29 and 6.30. Now, the targets are all in range bins 10 and 11. In the 10th range bin, three targets are present at angles 50°, 74 °and -62°, with amplitudes of 0.7, 1.0 and 0.3, respectively. In the 11th range bin, two targets are present at 46°and -62°. These have amplitudes of 0.7 and 0.6, respectively. The original framework is able to resolve only one of these targets. The proposed method is able to accurately resolve all 5 targets, with an estimated AR coefficient of $\hat{a}_1^1 = -0.916 - 0.011j$.

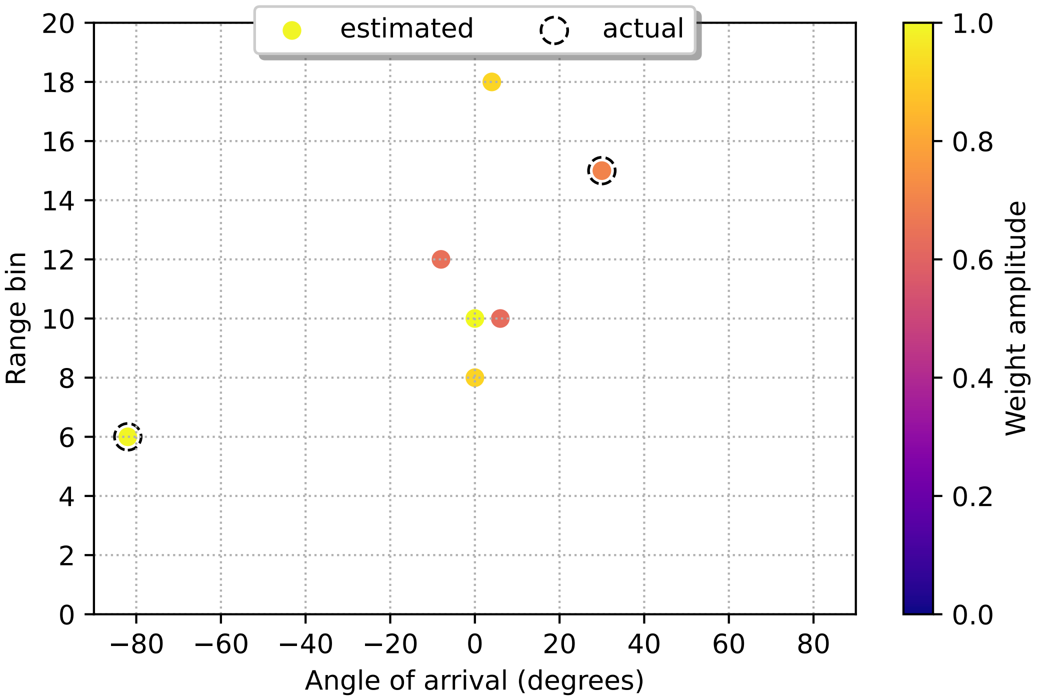

Figure 6.28: The estimation as performed using the proposed implementation for the two separated target case of Figure 6.27. The proposed method correctly estimates both targets in the two different range bins. The estimate contains 5 additional false detections.

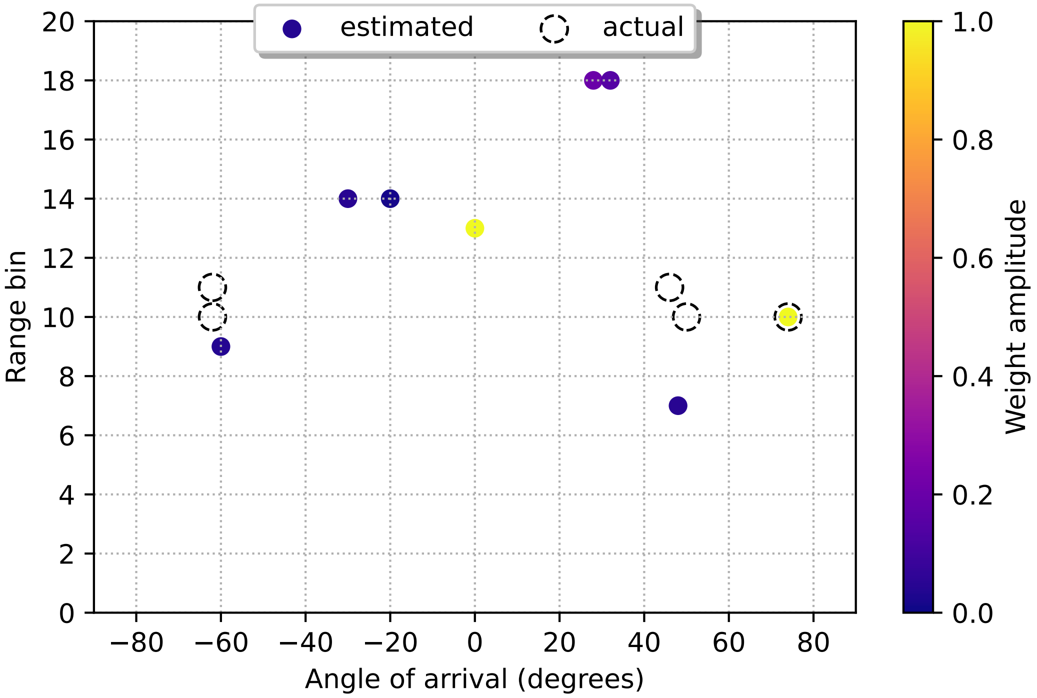

Figure 6.29: The estimation performed by the original implementation for the case where there are 5 targets present in two adjacent range bins. The targets in the 10th range bin are located at 50°, 74°and -62°. The two targets in the 11th range bin are located at 46°and -62°. The original implementation managed to correctly resolve the target at 74°but missed the remaining 4 targets. 6 additional false detections were included in the estimation.

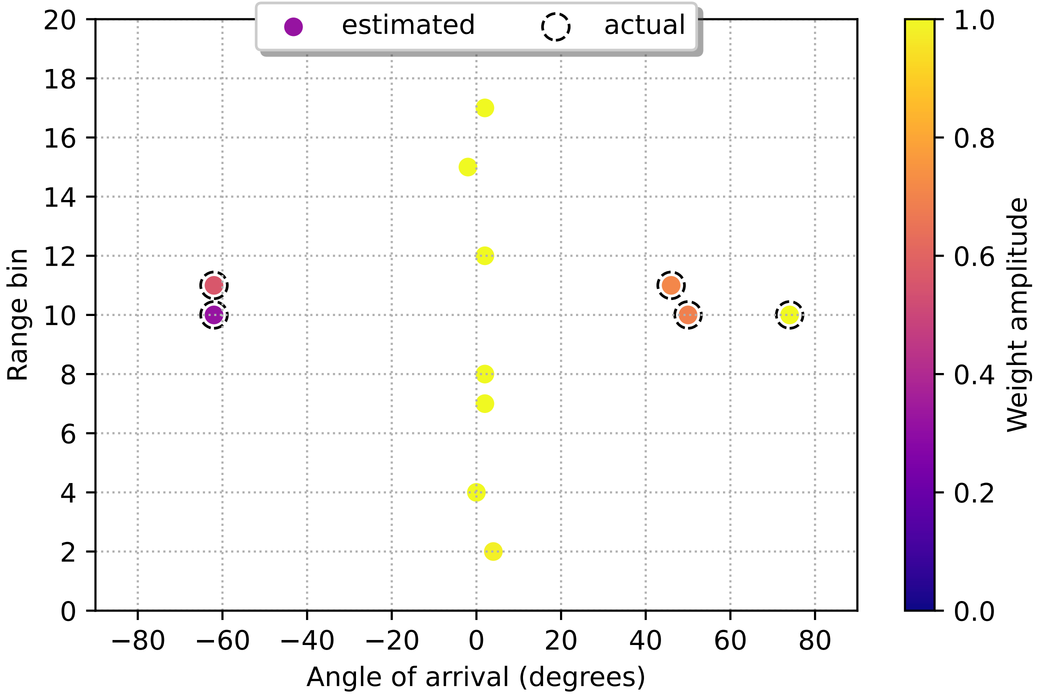

Figure 6.30: The estimation by the proposed framework for the adjacent range-bin scenario of Figure 6.29. The proposed method correctly resolved all 5 targets across the two different range bins. Additionally, there are 7 false detections included, however, across the range bins that didn't contain any targets.

7 Conclusions

In conclusion, this work presents an enhanced RVM-based framework for multitarget detection in cluttered environments, leveraging autoregressive parameter estimation to adapt effectively to dynamic clutter. The proposed method accurately determines AR parameters within a low number of iterations, enabling it to function efficiently as a joint process within the RVM framework. Through this joint estimation of target presence and clutter covariance, the framework demonstrates significant improvements in reducing false and missed detections compared to the traditional model based on white noise assumption.

This framework presents a promising step forward for multitarget detection in variable and cluttered environments, demonstrating potential applicability across various surveillance and defence radar systems.

7.1. Conclusions

The conclusions of this thesis will be drawn by reiterating and answering the research questions posed in Section 1.

Research Question 1

How can a framework be developed to effectively perform joint multitarget detection and clutter covariance estimation in a cluttered environment, which is stationary within a single burst but can vary over time?

By deriving the expectation-maximization update for a full covariance matrix in the RVM framework and applying Burg's method to perform this maximization, a joint framework is created that iteratively updates the estimated targets present in the data along with the clutter covariance matrix that is estimated from the residual. This enables the framework to assess the characteristics of the clutter experienced within that specific measurement and utilize this to explain the experienced noise better when performing an update of the estimated targets in the next iteration. By leveraging Burg's method, which is guaranteed to provide a stable autoregressive model and a good approximation of the autoregressive parameters, efficient joint processing is created. This results in a joint processing framework that generally only needs roughly as many iterations as the amount of included targets in the estimation.

Research Question 2

Can incorporating clutter covariance estimation in the detection model reduce false and missed detections?

The proposed framework is able to improve the amount of missed detections as well as false detections in multiple situations. As demonstrated through multiple simulated scenarios, the original model of Tipping using a white noise assumption fails to correctly resolve the correct targets from the measurement, leading to many missed detections and false detections, depending on the exact scenario. Although still not always perfect, the proposed method showed a significant improvement in the number of faults that were made in estimating the model, lowering the amount of missed detections in scenarios with very bad signal-to-clutter ratio and improving the number of false detections when the signal to clutter ratio was more favourable. The improvement that could be achieved also differed per different autoregressive clutter models and, therefore, deserves to be investigated further to express this into quantifiable results.

Research Question 3

How can this framework be applied across various scenarios, including both one-dimensional (range or angle) and two-dimensional (range-angle) cases?

As the proposed method leverages the RVM framework for estimation, it can be directly applied to diverse scenarios depending on the model definition. In this thesis, we've shown its applicability to a more fictitious range-only example and a realistic range-angle scenario. As also mentioned in Chapter 2, an update can be made to the phase delay expression to allow Doppler shifts to be estimated, making the framework also directly applicable to range-Doppler processing.

In this work, however, we also saw how the similarity of the provided basis functions and the autoregressive process can pose problems in the angle-only scenario. This can cause the intended framework to attempt to estimate autoregressive parameters based on the received signals with additional clutter, causing faulty estimations. However, we've also shown how these situations can be mitigated by supplying multiple measurement vectors with the same clutter process. Without the need for any knowledge of which vectors contain actual targets and which are pure clutter, the proposed method is able to perform a decent approximation in both estimated targets as well as autoregressive parameters. This allows for great flexibility in scenarios where it is known that only a few measurements will contain targets, but no knowledge beforehand is present as to which measurements will remain target-free.

7.2. Future work

Based on the work performed in this thesis, some future work can also be noted. First and foremost, additional investigation can be done to quantify the gain of the proposed method and the criteria for when this gain can be experienced. The initial results shown in this thesis showed very promising results but also showed big differences in how much better the proposed method performs compared to the original implementation, depending on AR parameters and the definition of the basis functions. Extensive simulations, possibly using Monte Carlo simulation, could provide insight into both the quantifiable gain the proposed method can yield and the bounds on the AR process regarding model order or parameter values for where to expect these gains.

Another topic that was shown to be relevant in the simulation was the assumed model order used in Burg's method to perform the expectation maximization. As an incorrect assumption of the model order can lead to missed results, the framework would benefit from having some form of order selection implemented to ensure it can perform the maximization properly without needing to know this model order beforehand.

Another suggestion for future work is to incorporate a proper detector into the framework to reduce false detections further. Especially in the range-angle simulations, it became apparent that while the proposed framework was very good at resolving the targets in the different range bins, it would also come up with additional targets to minimize the error between the measurement and the estimated autoregressive clutter process. While improving the accuracy of the clutter parameter estimate will probably also aid in reducing this error, implementing a detector, for example, based on a likelihood ratio test, could also prove valuable in assigning if the estimated targets are likely or not.

References

[1] K. L. S. Gunn and T. W. R. East, “The microwave properties of precipitation particles,” *Quarterly Journal of the Royal Meteorological Society*, vol. 80, no. 346, pp. 522–545, 1954.

[2] M. Skolnik, “Role of radar in microwaves,” *IEEE Transactions on microwave theory and techniques*, vol. 50, no. 3, pp. 625–632, 2002.

[3] J. Jao, “Amplitude distribution of composite terrain radar clutter and the κ-distribution,” *IEEE Transactions on Antennas and Propagation*, vol. 32, no. 10, pp. 1049–1062, 1984.

[4] R. G. Brereton and G. R. Lloyd, “Support vector machines for classification and regression,” *Analyst*, vol. 135, no. 2, pp. 230–267, 2010.

[5] M. Tipping, “The relevance vector machine,” *Advances in neural information processing systems*, vol. 12, 1999.

[6] D. P. Wipf and B. D. Rao, “Sparse bayesian learning for basis selection,” *IEEE Transactions on Signal processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[7] H. Liu, B. Jiu, H. Liu, and Z. Bao, “Superresolution isar imaging based on sparse bayesian learning,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 5005–5013, 2013.

[8] P. Stoica, J. Li, X. Zhu, and J. R. Guerci, “On using a priori knowledge in space-time adaptive processing,” *IEEE transactions on signal processing*, vol. 56, no. 6, pp. 2598–2602, 2008.

[9] C. Cook, *Radar signals: An introduction to theory and application*. Elsevier, 2012.

[10] H. Said, A. El-Kouny, and A. El-Henawey, “Design and realization of digital pulse compression in pulsed radars based on linear frequency modulation (lfm) waveforms using fpga,” in *2013 International Conference on Advanced ICT and Education (ICAICTE-13)*, Atlantis Press, 2013, pp. 803–808.

[11] C. E. Cook, “Pulse compression-key to more efficient radar transmission,” *Proceedings of the IRE*, vol. 48, no. 3, pp. 310–316, 1960.

[12] L. C. Godara, “Application of antenna arrays to mobile communications. ii. beam-forming and direction-of-arrival considerations,” *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1195–1245, 1997.

[13] S. Qin, Y. D. Zhang, and M. G. Amin, “Generalized coprime array configurations for direction-of-arrival estimation,” *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1377–1390, 2015.

[14] B. Demissie and C. R. Berger, “High-resolution range-doppler processing by coherent block-sparse estimation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 2, pp. 843–857, 2014.

[15] A. N. Pettitt, I. S. Weir, and A. G. Hart, “A conditional autoregressive gaussian process for irregularly spaced multivariate data with application to modelling large sets of binary data,” *Statistics and Computing*, vol. 12, no. 4, pp. 353–367, 2002.

[16] M. E. Tipping, “Sparse bayesian learning and the relevance vector machine,” *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001.

[17] D. J. C. MacKay, “Bayesian methods for backpropagation networks,” *Models of Neural Notworks*, pp. 211–254, 1994.

[18] R. M. Neal, “Bayesian learning for neural networks,” Ph.D. dissertation, University of Toronto, 1995.

[19] A. Faul and M. Tipping, “Analysis of sparse bayesian learning,” *Advances in neural information processing systems*, vol. 14, 2001.

[20] M. E. Tipping and A. C. Faul, “Fast marginal likelihood maximisation for sparse bayesian models,” in *International workshop on artificial intelligence and statistics*, PMLR, 2003, pp. 276–283.

[21] K. Duan, Z. Wang, W. Xie, H. Chen, and Y. Wang, “Sparsity-based stap algorithm with multiple measurement vectors via sparse bayesian learning strategy for airborne radar,” *IET Signal Processing*, vol. 11, no. 5, pp. 544–553, 2017.

[22] Y. Sun, X. Yang, T. Long, and T. K. Sarkar, “Robust sparse bayesian learning stap method for discrete interference suppression in nonhomogeneous clutter,” in *2017 IEEE Radar Conference (RadarConf)*, IEEE, 2017, pp. 1003–1008.

[23] K. P. Rajput, M. B. Shankar, K. V. Mishra, M. Rangaswamy, and B. Ottersten, “Cofar clutter channel estimation via sparse bayesian learning,” in *2023 IEEE Radar Conference (RadarConf23)*, IEEE, 2023, pp. 1–5.

[24] S. Salari, F. Chan, Y.-T. Chan, I.-M. Kim, and R. Cormier, “Joint doa and clutter covariance matrix estimation in compressive sensing mimo radar,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 1, pp. 318–331, 2018.

[25] R. R. Pote and B. D. Rao, “Maximum likelihood-based gridless doa estimation using structured covariance matrix recovery and sbl with grid refinement,” *IEEE Transactions on Signal Processing*, vol. 71, pp. 802–815, 2023.

[26] D. Wang, T. Wang, W. Cui, and X. Zhang, “A clutter suppression algorithm via enhanced sparse bayesian learning for airborne radar,” *IEEE Sensors Journal*, vol. 23, no. 10, pp. 10 900–10 911, 2023.

[27] S. Haykin, R. Bakker, and B. W. Currie, “Uncovering nonlinear dynamics-the case study of sea clutter,” *Proceedings of the IEEE*, vol. 90, no. 5, pp. 860–881, 2002.

[28] E. Conte, A. De Maio, and G. Ricci, “Covariance matrix estimation for adaptive cfar detection in compound-gaussian clutter,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 415–426, 2002.

[29] H. Wensink, “On parametric detection of small targets in sea clutter,” in *Proceedings of the Third International Conference on Information Fusion*, IEEE, vol. 1, 2000, MOC1–17.

[30] J. B. Billingsley, A. Farina, F. Gini, M. V. Greco, and L. Verrazzani, “Statistical analyses of measured radar ground clutter data,” *IEEE Transactions on Aerospace and electronic Systems*, vol. 35, no. 2, pp. 579–593, 1999.

[31] S. Haykin, B. W. Currie, and S. B. Kesler, “Maximum-entropy spectral analysis of radar clutter,” *Proceedings of the IEEE*, vol. 70, no. 9, pp. 953–962, 1982.

[32] M. Greco, F. Bordoni, and F. Gini, “X-band sea-clutter nonstationarity: Influence of long waves,” *IEEE Journal of Oceanic Engineering*, vol. 29, no. 2, pp. 269–283, 2004.

[33] B. Friedlander and B. Porat, “The modified yule-walker method of arma spectral estimation,” *IEEE Transactions on Aerospace and Electronic Systems*, no. 2, pp. 158–173, 1984.

[34] J. P. Burg, *Maximum entropy spectral analysis.* Stanford University, 1975.

[35] M. J. de Hoon, T. H. van der Hagen, H. Schoonewelle, and H. van Dam, “Why yule-walker should not be used for autoregressive modelling,” *Annals of nuclear energy*, vol. 23, no. 15, pp. 1219–1228, 1996.

[36] P. Broersen, “The abc of autoregressive order selection criteria,” *IFAC Proceedings Volumes*, vol. 30, no. 11, pp. 245–250, 1997.

[37] W. X. Zheng, “Autoregressive parameter estimation from noisy data,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 1, pp. 71–75, 2000.

[38] S. Kay, “Recursive maximum likelihood estimation of autoregressive processes,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 1, pp. 56–65, 1983.

[39] Ö. Çayır and Ç. Candan, “Maximum likelihood autoregressive model parameter estimation with noise corrupted independent snapshots,” *Signal Processing*, vol. 186, p. 108 118, 2021.

[40] Ç. Candan, “Making linear prediction perform like maximum likelihood in gaussian autoregressive model parameter estimation,” *Signal Processing*, vol. 166, p. 107 256, 2020.

[41] S. M. Hurtado Rúa, M. Mazumdar, and R. L. Strawderman, “The choice of prior distribution for a covariance matrix in multivariate meta-analysis: A simulation study,” *Statistics in medicine*, vol. 34, no. 30, pp. 4083–4104, 2015.

[42] M. L. Vis and L. L. Scharf, “A note on recursive maximum likelihood for autoregressive modeling,” *IEEE transactions on signal processing*, vol. 42, no. 10, pp. 2881–2883, 1994.

[43] J. Durbin, “The fitting of time-series models,” *Revue de l’Institut International de Statistique*, pp. 233–244, 1960.

[44] J. Makhoul, “Stable and efficient lattice methods for linear prediction,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 5, pp. 423–428, 1977.

[45] P. Stoica, R. L. Moses, *et al.*, *Spectral analysis of signals*. Pearson Prentice Hall Upper Saddle River, NJ, 2005, vol. 452.

[46] S. De Waele and P. M. Broersen, “The burg algorithm for segments,” *IEEE Transactions on Signal Processing*, vol. 48, no. 10, pp. 2876–2880, 2000.

[47] M. Deriche, “Ar parameter estimation from noisy data using the em algorithm,” in *Proceedings of ICASSP’94. IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, vol. 4, 1994, pp. IV–69.

[48] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: `10.1038/s41586-020-2649-2`. [Online]. Available: `https://doi.org/10.1038/s41586-020-2649-2`.

[49] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: `10.1038/s41592-019-0686-2`.

[50] H. Akaike, “A new look at the statistical model identification,” *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716–723, 1974.

[51] H. Zhu, G. Leus, and G. B. Giannakis, “Sparsity-cognizant total least-squares for perturbed compressive sampling,” *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2002–2016, 2011.

[52] Z. Yang, L. Xie, and C. Zhang, “Off-grid direction of arrival estimation using sparse bayesian inference,” *IEEE transactions on signal processing*, vol. 61, no. 1, pp. 38–43, 2012.

A

Additional figures

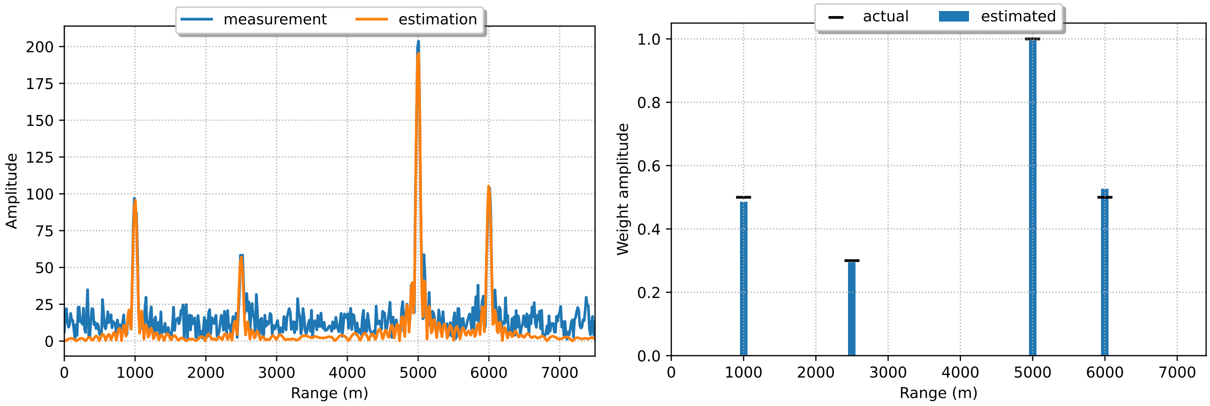

Figure A.1: The results of using a relevance vector machine (RVM) with a white noise assumption for a scenario with four targets located at 1000, 2500, 5000, and 6000 meters (range bins 10, 25, 50, and 60, respectively). The left image shows the predicted target vector amplitude based on the estimated weights for the basis functions in the dictionary. The right image displays the amplitude of the weights with which each of the 18 basis functions is included in the prediction. Additionally, a clutter autoregressive (AR) process of order 1 with coefficient $a_1^1 = -0.95 \times \left(\frac{1}{2\sqrt{2}} + \frac{1}{\sqrt{2}}j\right)$ is added to the measurement. The RVM used these 18 basis functions to predict the measurement vector.

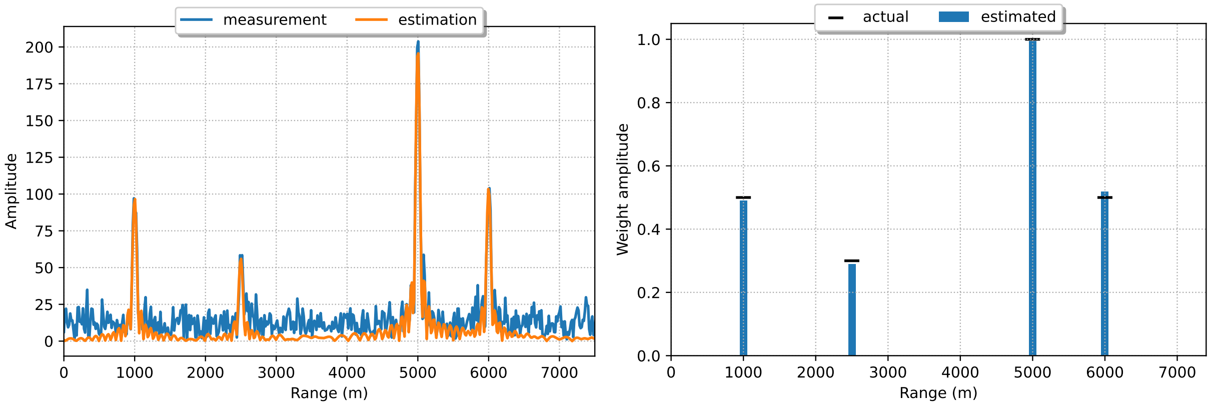

Figure A.2: The results of using a relevance vector machine (RVM) with a white noise assumption for a scenario with four targets located at 1000, 2500, 5000, and 6000 meters (range bins 10, 25, 50, and 60, respectively). The left image shows the predicted target vector amplitude based on the estimated weights for the basis functions in the dictionary. The right image displays the amplitude of the weights with which each of the 18 basis functions is included in the prediction. Additionally, a clutter autoregressive (AR) process of order 1 with coefficient $a_1^1 = -0.95 \times \left(\frac{1}{2\sqrt{2}} + \frac{1}{\sqrt{2}}j\right)$ is added to the measurement. The RVM used these 18 basis functions to predict the measurement vector.