RADAR RESOURCE ALLOCATION OPTIMIZATION IN PHASED ARRAY RADAR SYSTEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İZZET SERBEST

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2019

Approval of the thesis:

RADAR RESOURCE ALLOCATION OPTIMIZATION IN PHASED ARRAY RADAR SYSTEMS

submitted by **İZZET SERBEST** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Eng.** ________________

Prof. Dr. Mustafa Kuzuoğlu
Supervisor, **Electrical and Electronics Eng., METU** ________________

Examining Committee Members:

Prof. Dr. Umut Orguner
Electrical and Electronics Eng., METU ________________

Prof. Dr. Mustafa Kuzuoğlu
Electrical and Electronics Eng., METU ________________

Prof. Dr. Özlem Özgün
Electrical and Electronics Eng., Hacettepe University ________________

Prof. Dr. Asım Egemen Yılmaz
Electrical and Electronics Eng., Ankara University ________________

Assist. Prof. Dr. Emre Özkan
Electrical and Electronics Eng., METU ________________

Date: 09.09.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: İzzet Serbest

Signature:

ABSTRACT

RADAR RESOURCE ALLOCATION OPTIMIZATION IN PHASED ARRAY RADAR SYSTEMS

Serbest, İzzet
Master of Science, Electrical and Electronics Engineering
Supervisor: Prof. Dr. Mustafa Kuzuoğlu

September 2019, 74 pages

The demand for enhanced radar technologies has grown while mission requests have become more complex. Development of Active Electronically Scanned Array (AESA) Technologies has created enormous functional achievements. Development of radar platforms has led to the radar resource allocation issues and adaptive Radar Resource Management (RRM) studies for Multi-Function Radars. Combining the functionality of different tasks in one special device also makes resource allocation process more challenging due to its comprehensive capabilities. Such complicated systems have become an example of technology in which multiple tasks can share multiple resources in order to satisfy their requirements. Therefore, resource optimization strategy is becoming more crucial for radar systems.

This thesis is mainly focused on radar resource allocation in order to ensure optimization of radar resources in an efficient way. A proposed resource allocation approach described in [1] is applied in detail. Optimization-based measurement policies are studied for online beam scheduling in real-time. Radar tasks by which resource allocation is held are approached like series of independent tracking and searching subtasks in the system. Using the independent subtask approach makes optimization easier and converts it to a known general integer linear programming

problem. The optimization problem is modeled to maximize overall utility function based on tracking quality in real time while meeting resource constraints. Connection of radar tasks is handled via constraints of the resources, and the constraints are included in a resource allocation algorithm using Lagrange relaxation method.

In addition, different performance measures are used in optimization to reflect different aspects which are important at the slow time level. As an example, implementation and testing of tracking in clutter using Probabilistic Data Association is studied. Using PDA filter, control of the gating thresholds gives rise to a different optimization problem solution.

Keywords: Sensor Management, Resource Allocation, Optimization-based Scheduling, Lagrange Relaxation Method, Dynamic Programming

ÖZ

FAZ DIZILI RADAR SISTEMLERINDE KAYNAK TAHSISI OPTIMIZASYONU

Serbest, İzzet
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği
Tez Danışmanı: Prof. Dr. Mustafa Kuzuoğlu

Eylül 2019, 74 sayfa

Gelişerek artan ve karmaşık hale gelen görev ihtiyaçları nedeniyle gelişen radar teknolojilerine olan gereksinim artmaktadır. Özellikle elektronik tarama dizili radar teknolojisi, radarların fonksiyonel özelliklerini büyük ölçüde geliştirmiştir. Fakat bu gelişim ile artan görev yoğunluğu, radar kaynak tahsisi sorununu ortaya çıkarmıştır. Birden fazla ihtiyacın, tek bir kaynağı kullanması, radarlarda kaynak ihtisasının optimizasyonunu zorunlu hale getirmiştir.

Bu tezde, radarın huzme-zaman kaynak planlamasını verimli bir şekilde yapabilmek için geliştirilen gerçek zamanlı Lagrange rahatlatma modeli [1] gerçeklenmeye çalışılmıştır. Model, huzme planlaması yapabilmek için radarın takip ve arama görevlerini birbirinden bağımsız olacak şekilde tanımlamış ve optimizasyon problemini çözümü bilinen tamsayı doğrusal problemine dönüştürmüştür. Bu şekilde hedeflerin belirli bir zaman diliminde izleme kalitesini maksimuma çıkarmayı amaçlamaktadır. Görev paylaşımında kullanılan kısıtlamalar, Lagrange rahatlatma metodundan yararlanılarak optimizasyon problemine dahil edilmiştir.

Bu tezde ayrıca modele, farklı performans ölçümleri eklenerek farklı senaryolar oluşturulmuş optimizasyon modelinin gelişimi gözlenmiştir. Olasılıklı Veri İlişiklendirme filtresinin karmaşa içeren izlemede uygulanması ve test edilmesi örnek olarak verilebilir. Olasılıklı veri ilişiklendirmenin kullanılmasıyla birlikte kontrol edilen kapı eşikleri, farklı bir optimizasyon problemi çözümüne yol açar.

Anahtar Kelimeler: Sensör Yönetimi, Kaynak Tahsisi, Optimizasyon Tabanlı Planlama, Lagrange Rahatlatma Yöntemi, Dinamik Programlama

To my beloved family...

ACKNOWLEDGEMENTS

I would like to express that it was a great opportunity for me to study with Prof. Dr. Mustafa Kuzuoğlu during my thesis study. I had the chance to gain considerable experience in the area of radar applications. I am grateful to him for his guidance and enormous encouragements to complete this study.

I would like to thank to my beloved family members, my mother Semra Serbest, my father Fikri Serbest and my brother Murat Serbest for their continuous support during the period of study.

Pınar Altınkeser played a key role by advising new and unique ideas to solve the problems in my thesis. I would like to express my endless gratitude to her hereby.

I would like to express my gratitude to my dear friends who helped me a lot to concentrate on my study both mentally and physically, Mesut Göksu and Ali Kemal Kazar.

Finally, I would like to thank to my company ASELSAN Inc. for the equipments and facilities I was provided with and my team leader Muharrem Arık for his continuous support.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1. Overview of Radar Resource Management

Multi-function (MF) radar platforms are enhanced systems which must satisfy a wide range of on-demand services. These services, which can be considered as tasks, can be target tracking, sector scanning, missile guidance, terrain imaging (Synthetic-aperture radar - SAR), etc. Although these services seem to be extensive, they are strictly limited, and resource centers usually have limited capacity to host these data and applications. This leads to a strong motivation to design scheduling policies and optimize utilization of services. In radar, the scheduling policies of the tasks are managed by a central subsystem which is called Radar Resource Management (RRM). The subsystem, which is critical to the operational success of an MF Radar, optimizes the radar resource usage in order to achieve maximum performance where the optimality is defined according to pre-defined cost functions. A typical resource management model is stated in [2] and it is given in Figure 1.1.

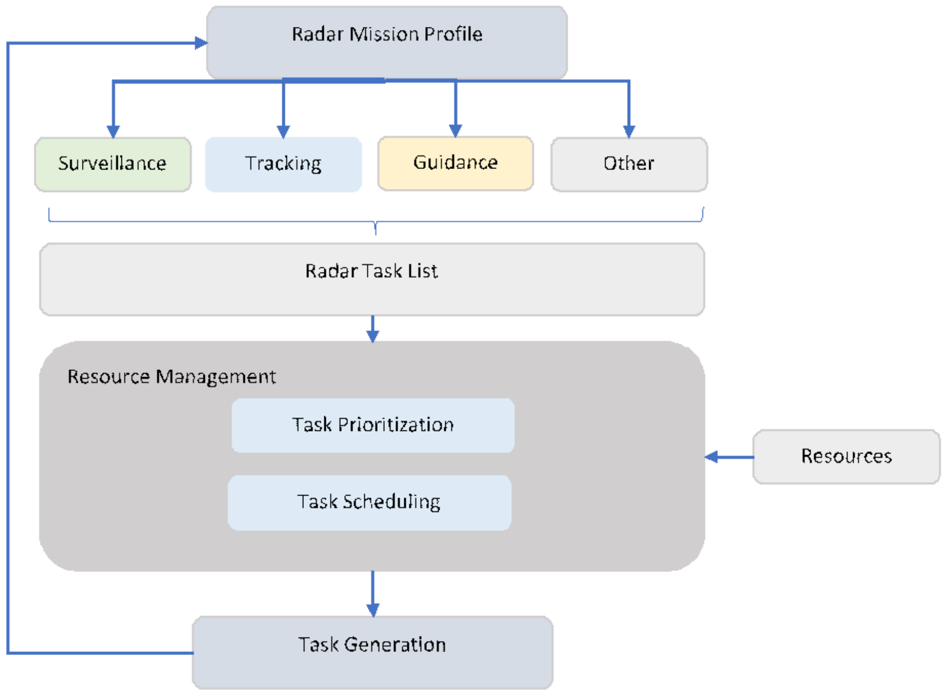

Figure 1.1: An MF Radar Resource Management Model [2]

In order to describe a resource management problem, system performance criteria and its resources must be well defined. The main goal of resource management is to increase the overall system performance, which usually depends on the quality of the radar output, while maintaining the radar resources below the acceptable level. The main resources of the radar are energy, time and processing budget. The challenge of resource management emerges when these resources are not sufficient to perform all the tasks. Some of the tasks might be done by degraded performance due to non-available resources or may not even be able to be performed. In these cases, to maintain acceptable quality of service (QoS), several resource management algorithms are used in the radar systems.

RRM algorithms are grouped into five following categories [2]:

- Artificial intelligence algorithms
- Quality of service (QoS) resource allocation management (Q-RAM) algorithms

- Waveform-aided algorithms
- Adaptive update rate algorithms
- Dynamic programming algorithms

All the algorithms stated above must perform two basic issues which are task prioritization and task scheduling. These algorithms are also essentially the elements of two classes which are entitled as “non-adaptive” or “adaptive”. In non-adaptive scheduling algorithm, the radar scheduler includes some heuristic rules and follows a pre-defined guideline. Therefore, the resource evaluation is not performed in real-time. In adaptive scheduling, algorithms are much more complicated than non-adaptive case and they have much more computational complexity, however they are more efficient. Because of high performance requirements particularly, advanced MF Radars mostly use adaptive scheduling algorithms.

Basic outlines of existing resource management algorithms are summarized as follows.

Artificial intelligence algorithms: These algorithms handle resource management problems by using neural network, expert system or fuzzy logic approaches. Neural network algorithms are used in not only task prioritization but also in task scheduling issues. RRM is handled by using multi-layer neural systems. Feature of task vector is the input to the multi-layer neural network, whose layers are linked together with specific weight values. In the testbed, training data is used to extract the weights of the neurons. In the radar operation, the trained neural network generates the priorities based on all pre-given target feature data and task scheduling is performed based on these results [3-5]. In expert systems, system control operations are decided by the database of the pre-determined rules. Other artificial intelligence algorithms, such as fuzzy logic approaches, resolve task priority issues in terms of an adaptive scheduler. The basic task of the adaptive scheduler is to decide the task prioritization by means of the fuzzy variables which are commonly tracking quality, friend or foe information and target behavior kinematics.

Quality of service (QoS) Algorithms: Q-RAM algorithms are based on the concept of quality of service (QoS). Originally, QoS algorithms have been introduced by International Telecommunication Union in 1994 in the field of internet services. Afterwards, similar algorithms have been utilized in radar applications [6]. These algorithms are similar to dynamic programming algorithms, but the aim of Q-RAM algorithms is to select an operation point for each task to maximize the global system utility in the system operation. Individual tasks in the system routine are combined in an appropriate way to describe the overall utility function, which is also called as quality space. Task scheduling and prioritization decisions are made by the operation setpoint where the compound utility function and resource constraints met. Mathematical representation of this approach is expressed below [7].

$$maximize: u(\emptyset_{1,}, \emptyset_{2,}, \ldots, \emptyset_n) = \sum_{i=1}^{n} u_i(\emptyset_i) \ \ (System\ Utility) \tag{1.1}$$

$$subject\ to: \forall_{1\le k\le m} \quad \textstyle\sum_{i=1}^{n} R_{ik} \le \ R_{ik}^{max}$$

$$\forall_{1\le k\le m, 1\le i\le n} \ \ R_{ik} = g_{ik}(\emptyset_i)$$

$$u_i\ is\ the\ utility\ function\ of\ i^{th}\ task$$

$$\emptyset_i\ set-point\ of\ the\ task\ i$$

$$R_{ik}\ is\ the\ k^{th}\ shared\ resource\ of\ task\ i$$

$$g\ is\ a\ function\ that\ specifies\ the\ amount\ of\ resources\ required$$

$$for\ a\ task\ to\ operate\ at\ each\ set-point$$

$$where\ n\ is\ the\ number\ of\ utility\ and\ m\ is\ the\ number\ of\ resource$$

The details of such formulations can be found in the literature [6] and [8].

Waveform-Aided Algorithms: The aim of these algorithms is to select the most effective waveforms from a fixed library while maintaining the high-level radar performance. In waveform-aided algorithms, waveform selection can be made by different approaches. One example to minimize the tracking errors is presented by

Scala [9]. In the paper, radar waveforms are selected for providing minimum target uncertainty and minimum detection time. In [10], waveforms are selected with the help of a neural network algorithm to minimize the clutter and jamming effect. Examples of such approaches exist in the literature [11-13].

Adaptive Update Rate Algorithms: These algorithms can be considered as extensions of traditional trackers having uniform update rates. The main purpose of such algorithms is to maintain a required level of tracking performance. Therefore, the update rate is closely related to the radar surroundings, target characteristics, and target maneuvering level. In contrast to waveform-aided algorithms, the adaptive update rate algorithms aim to minimize the target revisit time rather than minimizing environmental degradation. Spacing in track update tasks results less usage of the radar resources. Therefore, adaptive update rate algorithms are resource-aided algorithms. Adaptive update rate algorithms are described in [10] and [14].

Dynamic Programming Algorithms: Dynamic programming algorithms approach the resource management problem as if it is Multi-Armed Bandit Problem (MAP) involving Hidden Markov Models (HMM). The aim of these algorithms is to maximize radar performance selecting the best task decision based on a discrete time stochastic control process model. Since radar technologies are gradually becoming more complex, radar system operations and their constraints are also getting more complicated. Therefore, finding the optimal solution requires high computational workload because of the dimensionality problem of the states. In this approach, the most important workload is to set the system objective function and its constraints. Since MF Radar is one of the most complicated sensors in military platforms, it is hard to cover all the tasks and constraints because of the difficulty in computational implementation. Objective function is usually non-linear and the optimization problem is NP-hard. In addition to that, stochastic approach used for modeling also creates the complexity. Nevertheless, this research area has attracted attention recently, since these algorithms provide encouraging solutions by employing digital optimization techniques. Further researches in this direction are explained in [15] and [16].

In this thesis, one of the specialized dynamic programming algorithms is implemented.

1.2. Outline and Summary of Contributions

The approach given in [1] is taken as a basis and the method is further developed and implemented in the thesis. A study has been performed to find out how the algorithm developed in [1] performs according to changing system models and environmental situations. This thesis is not a direct implementation of [1], and the method is enhanced and modified in certain aspects. The main contributions and modifications can be summarized as follows

- In addition to the Kalman filter, we have employed a tracking algorithm by adding PDA filter.
- The approaches which are presented in [1] such as Markov decision process modelling, two timescales scheduling are also utilized in the thesis. However, unlike [1], a different approach is used to create the cost and utility functions.
- We have developed the simulation environment by adding track initiation, “tracking mixes” event, different kinematic models, missing targets and clutter in the trajectories.
- Target tracking techniques such as Track While Scan (TWS), adaptive tracking and Lagrange-based tracking have been compared with each other, and the performance of the algorithms are discussed.

1.2.1. Chapter 2

This chapter presents brief information about radar systems. Although basic working principles of the radar have stayed the same since its first release, there have been many developments and improvements on radar capabilities. Therefore, the basic principle of operation of radar is simple to understand, however, the system theory can be quite complex. The information described in this chapter will clarify overall concept of the study. In the chapter, fundamentals of radar principle are introduced. Radar block diagram and its operation are described briefly. This section also presents radar signal modelling, detection and estimation theory. After presenting the fundamentals of radar system, measurement and target motion model are also explained in this chapter.

1.2.2. Chapter 3

This section describes the structure of the radar resource management model on which the thesis is based. To describe the optimization problem, utility functions based on performance measures, target state dynamics, cost functions and constraints are explained in detail. Besides that, for the solution of optimization problem, to find the optimum policy, problem relaxation techniques based on Lagrange duality and the other assumptions are described further.

The following parts of the chapter provide the theoretical background of Lagrange relaxation and solution approach of the optimization problem. Furthermore, the model and algorithms that we used in this thesis are also detailed. The solution considers an online implementation of optimization problem, which operates in real-time.

1.2.3. Chapter 4

In the chapter, the simulated target tracking scenarios with different performance measures are performed. The experimental results used to make a comparison on different performance models are also discussed. Comments are provided for each scenario of the model, which lead to recommendations for future studies.

CHAPTER 2

FUNDAMENTALS OF THE RADAR PRINCIPLE

2.1. Introduction to Radar

A radar is a system that uses electromagnetic waves in order to determine the velocity, location (distance), elevation (altitude) of the objects and the direction of travel. Radar systems, like other complex electronics systems, are composed of several major subsystems and many individual circuits. Although modern radar systems are quite complicated, their operation can be easily understood by using a basic block diagram of the system which is shown in Figure 2.1. Most radar systems can be considered as a variation of this system. In order to better understand the radar structure, elements of the block diagram are described briefly in the following sections.

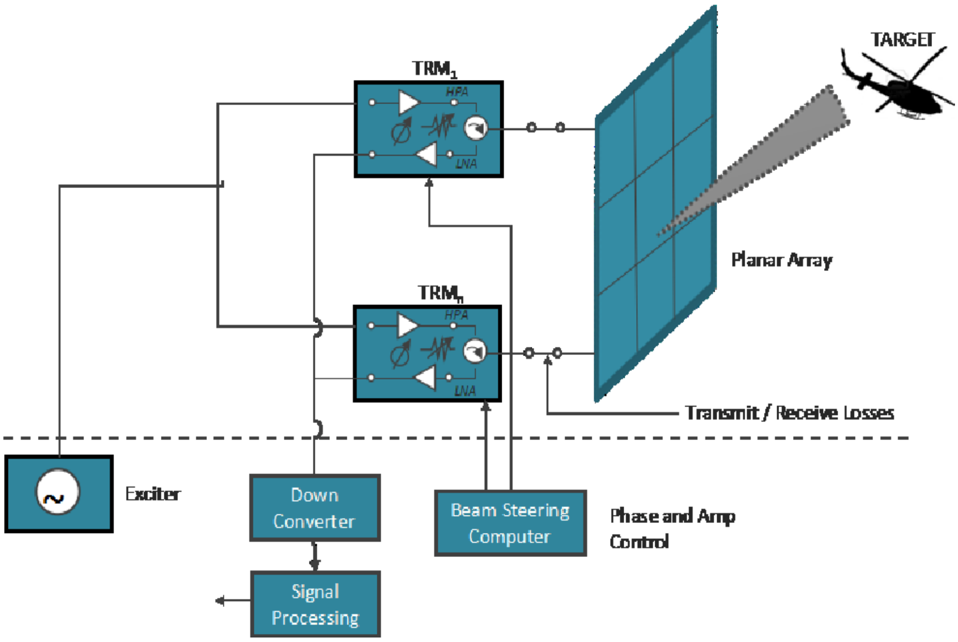

Figure 2.1: A Typical Block Diagram of Multi-function Radar [17]

2.1.1. Subsystems of Multi-Function Radar

2.1.1.1. Transmitter Subsystem

MF radar steers the signal in a different way than traditional radars produced in earlier years. In contrast to traditional radars, MF Radar uses a large number of Transmitter/Receiver blocks, named as TR modules, just behind the planar array (composed of many small antennas on a flat panel) for signal amplification and transmission purposes. Using this type of modules enables more efficient usage of energy as well as steering the beam electronically. This method also allows radar to accelerate beam position movements.

In order to detect the objects / targets surrounding the radar, a pulsed signal is generated by the RF synthesizer, which is commonly called as Direct Digital Synthesis. Then, the signal is transmitted to the input of TR modules under processor control. Each TR module changes the phase of the signal in a unique way in order to achieve spatial selectivity of the signal which is known as beamforming. The signals from each element are added together in the desired direction, and beamforming technique cancel out other signal directions known as interference. This capability obviates the need for motors and moving parts, which increases the reliability and agility but also it can decrease the cost of the system. Final stage of the transmitter path, planar array antenna steers the radar beam.

2.1.1.2. Receiver Subsystem

Reflected signal from the target, which is time delayed, Doppler shifted, attenuated version is received by T/R modules. Commonly, T/R modules convert the signal to IF frequency in the first step. Secondly, signal is delivered to analog-digital converters (ADC) and first signal processing step, which includes digital down conversion and range gating, occurs in these time intervals. Furthermore, the received signals from each element are added together in order to realize the digital beamforming operation. After receiver subsystem, the digitized signal is sent to the radar processing subsystem for advanced signal processing operations.

2.1.1.3. Radar Processing Subsystem

In general, Radar Processing Subsystem have two basic functions. These are:

- performing, controlling and scheduling radar tasks called as radar system management function.
- processing all incoming received radar data, preparing and servicing them properly for users called as radar signal processing function.

The most important issue to be mentioned is that radar resource management is performed in this subsystem.

2.1.2. Radar Signal Modelling

For comprehensive information, radar signal modelling also needs to be explained. This section presents the basic signal model used in the radar operations. In MF Radar, all incident signals are summed coherently with a specific time delay (or phase shift) at each antenna element.

To define the signal, let us denote the antenna element position $\boldsymbol{r} = (x, y,\ z)^T$ in terms of a Cartesian coordinate system. In collaboration with the antenna position, the angle of incidence of an incoming plane wave is also expressed by $\boldsymbol{u}$ which is the unit direction vector in the antenna coordinate system. $\boldsymbol{u}$ is also named as "direction of cosines" and is shown in Figure 2.2.

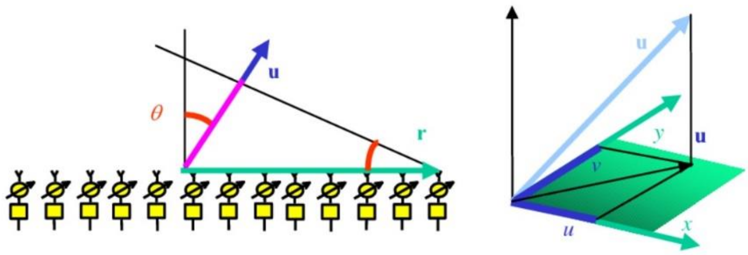

Figure 2.2: Direction Cosines [18]

The path length difference between the antenna element at position $\boldsymbol{r}$ and the origin is described by (2.1).

$$\boldsymbol{r}^T\boldsymbol{u} = xu + yv + zw = |r|.|u|_{=1}.\cos(r, u) \qquad (2.1)$$

After describing spatial information, a transmitted signal at each element is expressed as follows [18],

$$s_r(t, \boldsymbol{u}) = b.e^{j2\pi f t}.e^{j2\pi f \boldsymbol{r}^T\boldsymbol{u}/\boldsymbol{c}} \qquad (2.2)$$

where f is the transmit frequency and c is the velocity of light. Correspondingly, the designated beam in direction u_0 with N antenna elements are expressed as (2.3).

$$s_t(\boldsymbol{u}, \boldsymbol{u_0}) = \sum_{k=1}^{N} \underbrace{e^{-\frac{j2\pi f \boldsymbol{r_k}^T \boldsymbol{u_0}}{c}}}_{\bar{a}_k(\boldsymbol{u_0})} . s_{\boldsymbol{r_k}}(t, \boldsymbol{u}) = \boldsymbol{a}^{\boldsymbol{H}}(\boldsymbol{u_0}) . \boldsymbol{s}(t, \boldsymbol{u}) \tag{2.3}$$

The superscript $\boldsymbol{H}$ indicates conjugate transpose transformation. $\boldsymbol{a}^{\boldsymbol{H}}(\boldsymbol{u_0})$ is the steering vector. Finally, for the case of a linear antenna with equidistant elements where $x_k = \frac{kd\lambda}{2}$ (distance of the elements is $\frac{d\lambda}{2}$) results in the well-known function

$$\begin{aligned} s_t(\boldsymbol{u}, \boldsymbol{u_0}) &= \sum_{k=1}^{N} s_k(\boldsymbol{t}, \boldsymbol{u}) . e^{-\frac{j2\pi f x_k \boldsymbol{u_0}}{c}} \\ &= \sum_{k=1}^{N} b . e^{j2\pi f t} . e^{\frac{j2\pi}{\lambda} . \mathrm{x}_k (\boldsymbol{u} - \boldsymbol{u_0})} \\ &= b . e^{j2\pi f t} . \sum_{k=1}^{N} e^{\frac{j2\pi}{\lambda} . \mathrm{x}_k (\boldsymbol{u} - \boldsymbol{u_0})} \end{aligned} \tag{2.4}$$

In addition to the transmitted signal, each returning signal can be expressed as a set of reflections and each reflected signal is a time delayed, Doppler shifted, phase shifted and attenuated version of the transmitted signal which come back to the radar. Assume that the reflection process is linear and frequency independent within the bandwidth of the transmitted pulse. The returned signal from a target is then modelled as:

$$s_r(\boldsymbol{u}, \boldsymbol{u_0}) = \left\{ \sum_{k=1}^{N} b . G \, e^{j2\pi f \left(t + \frac{2V_r}{c} t\right)} . e^{\frac{j2\pi}{\lambda} . \mathrm{x}_k (\boldsymbol{u} - \boldsymbol{u_0})} \right\} \left\{ \sum_{i} g_i . e^{j2\pi f (-\tau_i) + \theta_i} \right\} \tag{2.5}$$

where g_i is the radar cross section of a reflector i, θ_i is the phase shift, V_r is the radial velocity between the antenna and the target creating a Doppler frequency shift, and τ_i is the time delay. Path losses, antenna gain, and other losses are included in G.

In equation 2.5, radar cross section (RCS) can be considered as the ratio of backscatter power density from the target to the power density intercepted by the target. Hence, as RCS gets larger, targets are detected more easily. In general, the RCS of a target depends on the orientation and target kinematics which is relative to the line of sight of the radar. The shape of the target and the line of sight of the radar cause the amplitude and phase of the signal to fluctuate. Therefore, RCS needs to be modeled in terms of a probability density function. Peter Swerling developed statistical representations of RCS referred to as the Swerling models. Models stated in [19] are as follows.

Swerling I Target

In Swerling I model, it is assumed that magnitude of the backscattered signal is relatively constant in each coherent processing interval, but it varies independently from scan to scan. Coherent Processing Interval fluctuations in Swerling I targets can be modelled by the Rayleigh-Function which is,

$$P(\sigma) = \frac{1}{\sigma_{average}} . exp\left(\frac{-\sigma}{\sigma_{average}}\right) \tag{2.6}$$

$$where\ \sigma_{average}\ is\ arithmetic\ mean\ of\ RCS\ values$$

Swerling II Target

This model is similar to Swerling I, except that the RCS changes from pulse to pulse in each dwell time. The Swerling cases I and II assumes that the target is built with many independent scatterers of roughly equal areas.

Swerling III Target

In Swerling III, the RCS varies according to a Chi-squared probability density function with four degrees of freedom ($m = 2$). The RCS is constant for a single scan just as in Swerling I type. However, the scattering scheme is different from Swerling I and that causes the target to be modelled with a different probability distribution. Probability distribution of the model can be viewed as:

$$P(\sigma) = \frac{4\sigma}{(\sigma_{average})^2}.exp\left(\frac{-2\sigma}{\sigma_{average}}\right) \tag{2.7}$$

Swerling IV Target

This model is similar to Swerling III, but there is one exception that the RCS changes from pulse to pulse in each dwell time.

Swerling V Target

The Swerling case V is a reference value with constant radar cross-section. In this case, it is assumed that magnitude and phase information of the backscattered signal remain the same all target scan.

2.2. Target Detection

After MF Radar receives the signal, it must be determined whether it is generated by a target or not. The received signal goes directly to analog to digital converter after the band pass filter. By the help of [20], down-converted IF signal can be modelled as:

$$v(t) = v_I(t)\, cos\omega_0 t + v_Q(t) \sin \omega_0 t = r(t) cos\big(\omega_0 t - \varphi(t)\big) \tag{2.8}$$

$$v_I(t) = r(t) cos\big(\varphi(t)\big) \tag{2.9}$$

$$v_Q(t) = r(t) \sin\big(\varphi(t)\big) \tag{2.10}$$

where $\omega_0 = 2\pi f_0$ is the carrier frequency, $r(t)$ is amplitude information of the $v(t)$ and $\varphi(t)$ is the phase information. At the same time, noise is added to the signal after passing through the receiver subsystem. Noise is assumed to be additive zero mean white gaussian distribution $n(t)$ with variance Ψ^2. But IF filter output can be viewed as complex random variable so the noise is also composed of quadrature components $(n_I(t), n_Q(t))$. In addition to that, noise is spatially incoherent and uncorrelated with the received signal. The joint Probability Density Function (pdf) of the two random variables are given as,

$$f(n_I, n_Q) = \frac{1}{2\pi\Psi^2} exp\left(-\frac{{n_I}^2 + {n_Q}^2}{2\Psi^2}\right) \tag{2.11}$$

If we assume that our target return signal is sinusoidal with amplitude A, the output of the IF filter can be described by the joint probability density function of noise and sine wave which is (detailed information can be found in [21]),

$$f(r, \varphi) = \frac{1}{2\pi\Psi^2} exp\left(-\frac{r^2 + \mathrm{A}^2}{2\Psi^2}\right) exp\left(\frac{rAcos\ \varphi}{\Psi^2}\right) \tag{2.12}$$

If we want to extract the amplitude information, we need to integrate the pdf (2.12) over $\varphi(t)$. Thus,

$$\begin{aligned} f(r) &= \int_0^{2\pi} f(r, \varphi) d\varphi \\ &= \frac{r}{\Psi^2} \exp\left(-\frac{r^2 + A^2}{2\Psi^2}\right) \frac{1}{2\pi} \int_0^{2\pi} exp\left(\frac{rAcos\ \varphi}{\Psi^2}\right) d\varphi \end{aligned} \tag{2.13}$$

where the integral inside Eq. (2.13) is known as the modified Bessel function of zero order,

$$I_0(\beta) = \frac{1}{2\pi} \int_0^{2\pi} e^{\beta cos\theta}\, d\theta \tag{2.14}$$

As a result, pdf of r(t) is given in (2.15) which is the Rician probability density function. If $\frac{A}{\Psi^2}$ is equal to zero, then the probability model turns into Rayleigh. However, if $\frac{A}{\Psi^2}$ is too large, then it becomes Gaussian density.

$$f(r) = \frac{r}{\Psi^2} I_0\left(\frac{rA}{\Psi^2}\right) \exp\left(-\frac{r^2}{2\Psi^2}\right) \tag{2.15}$$

Since the signal is modeled by a probabilistic distribution, we need to do a test to check whether the target exists. The choice is related to hypothesis testing, which is widely used in detection theory. In this case, the hypothesis is based on the decision which is evaluated according to the position of the received signal level and detection threshold. So, the target is detected whenever it exceeds the threshold value.

$$H_1 \colon v(t) = Acos(2\pi f_0 n + \varphi) + w(n) \tag{2.16}$$

$$H_0 \colon v(t) = w(n) \tag{2.17}$$

H_1 hypothesis shows that the target is detected, and H_0 shows that target is not detected. In addition to these, there are four possible outcomes which are based on this hypothesis:

- H_1 given that H_1 is true, correct detection.
- H_1given that H_0 is true, false alarm.
- H_0given that H_0 is true, correct dismiss.
- H_0 given that H_1 is true, missed detection

The efficiency of the radar detection is described with the probability of these outcomes, most often given by probability of detection and probability of false alarm. If the target is present and signal r(t) will exceed the threshold V_T then the probability of detection is found by integrating pdf of r(t) over the interval $\{V_T\ ,\ \infty\}$ which is,

$$P_D = \int_{V_T}^{\infty} \frac{r}{\Psi^2} I_0\left(\frac{rA}{\Psi^2}\right) \exp\left(\frac{r^2 + A^2}{2\Psi^2}\right) dr \tag{2.18}$$

In the case of the target not being present, the detection is incorrect, signal r(t) will exceed the threshold V_T and it is called as false alarm. In this case, amplitude of the received signal is zero and the equation 2.15 turns into,

$$f(r) = \frac{r}{\Psi^2} \exp\left(-\frac{r^2}{2\Psi^2}\right) \tag{2.19}$$

Corresponding probability of false alarm is represented by,

$$P_{fa} = \int_{V_T}^{\infty} \frac{r}{\Psi^2} \exp\left(\frac{r^2}{2\Psi^2}\right) dr = \exp\left(\frac{-{V_T}^2}{2\Psi^2}\right) \tag{2.20}$$

2.3. Formation of Scans

The main task of radar is to search targets of interest as well as providing the target parameters which are range and angular position. This task is accomplished by radiating multiple pulses called as scan. In order to investigate how radar takes information of scans, it would be better to start with the basic one pulse form of the radar range equation, which is stated in [21],

$$SNR = \frac{Energy\ of\ signal\ after\ matched\ filter}{noise\ variance} = \frac{E_t G_t A_r RCS}{4\pi^2 r^4 k T_s B_n L} \tag{2.21}$$

$$E_t\ is\ the\ energy\ of\ the\ transmitted\ pulse$$

$$G_t\ is\ the\ gain\ of\ the\ transmitting\ antenna$$

$$A_r\ is\ the\ effective\ aperture\ of\ the\ receiving\ antenna$$

$$r\ is\ the\ target\ range$$

$$kT_sB_n = 2\sigma^2\ is\ the\ noise\ power, where\ k\ is\ the\ Boltzmann's\ constant$$

$$T_s\ is\ the\ system\ noise\ temperature$$

$$B_n\ is\ the\ receiver\ noise\ bandwith$$

$$L\ is\ a\ generic\ loss\ term$$

If we formulate the equation in a simpler form, the equation turns into (2.22).

$$SNR = \left(\frac{r_0}{r}\right)^4 . G_{int}. RCS \tag{2.22}$$

In dB scale,

$$SNR_{dB} = 40log\left(\frac{r_0}{r}\right) + G_{int,dB} + RCS_{dB} \tag{2.23}$$

where G_{int} is a radar system performance variable. But this is the case when the target is in the line of sight of the radar. Most of the time, targets do not remain in the beam sufficiently, so range equation is needed to update by using antenna beamwidth position of the target, then updated range equation is as follows.

$$SNR_{dB} = 40log\left(\frac{r_0}{r}\right) + G_{int,dB} + RCS_{dB} - 10log_{10}(\cos^n(\varphi_b)) - 6\left(\frac{(\varphi_b - \varphi_t)^2}{\left(\frac{B}{2}\right)^2}\right) \tag{2.24}$$

where B is the double sided bandwith, φ_b is the azimuth direction of the beam. The detailed equation is given in [22] and [23]. As mentioned above, in search scan operations, amplitude and phase information of the receiving signal does not remain constant because the target does not remain sufficiently in the beam. In addition, the RCS fluctuations will also affect the detection performance. At this situation, SNR has a close relationship with the probability of detection, and this can be expressed by approximations and one of the very accurate approximation is described by Norton in [23], which is given by,

$$P_D \approx 0.5erfc\left(\sqrt{-\ln P_{fa}} - \sqrt{SNR + 0.5}\right) \tag{2.25}$$

where $erfc(z) = 1 - \frac{2}{\sqrt{\pi}}\int_0^z e^{-v^2} dv$ is the Gauss error function.

Furthermore, environmental factors such as clutter, atmospheric effects or electronic countermeasures, have an impact on the radar operation so radar should adjust itself to such external effects. In order to overcome these problems and improve the probability of detection as well as the SNR, pulse integration must be performed. Pulse integration is a technique to use multiple pulses in one horizon glance. In radar theory, two different pulse integration techniques are performed, coherent and non-coherent integration. In coherent case, pulses preserve the phase relationship in each received pulse and this method is also called as coherent processing integration (CPI). The other method called as the non-coherent integration is handled by summing all sequential pulses without considering the phase information. With the help of pulse integration, the effect of environmental factors such as clutter is reduced, and SNR is increased to the level where it can detect the targets. In [21], it is explained how pulse integration effect SNR and P_d over radar equation.

- **Coherent Case**

  In Coherent Case, the phase information is preserved as stated before. Then integrating n_p pulses could improve the SNR by the same factor, which is described as,

$$y_m(t) = s(t) + n_m(t) \tag{2.26}$$

$$s(t) = received\ signal$$

$$n_m(t) = gaussian\ white\ noise\ in\ pulse\ number\ m$$

$$m = pulse\ number$$

  Adding phase stabile n_p pulses will yield,

$$z(t) = \frac{1}{n_p}\sum_{m=1}^{n_p} y_m(t) = \sum_{m=1}^{n_p}\frac{1}{n_p}[s(t) + n_m(t)] \tag{2.27}$$

$$= s(t) + \sum_{m=1}^{n_p}\frac{1}{n_p}n_m(t) \tag{2.28}$$

  At each pulse integration step, the noise sequence would be uncorrelated and independent. So total noise power is degraded by the factor n_p. Hence, the total noise power of the coherent integration will be equal to the single pulse noise power. As a result, SNR will be improved by the factor n_p.

$$SNR_{n_p} = n_p . SNR_1 \tag{2.29}$$

Although coherent integration gives better results, since it leads to much more computational complexity, most radars do not perform coherent integration during search scans. They utilize non-coherent integration instead, which is explained below.

- **Non-Coherent Case**

  In non-coherent case, preserving the phase information is not a necessity. However, adding pulses in different phases may cause amplitude degradation. To prevent this, non-coherent integration is performed whenever signal passes through the envelope detector. Non-coherent integration is expressed conveniently by the following formula.

$$z = \sum_{n=1}^{n_p} x_n \tag{2.30}$$

x_n is the detector output of the received signal $r(t)$. Since $x_n's$ are uncorrelated, the joint pdf is calculated by taking convolution of the signals.

$$f(z) = f(x_1) \cdot f(x_2) \cdot \ldots f\left(x_{n_p}\right) \tag{2.31}$$

Without further explanation, improvement factor in non-coherent integration is defined as,

$$I(n_p) = \frac{SNR_1}{SNR_{n_p}} \leq n_p \tag{2.32}$$

Details of this formulation can be found in [21]. An empirically derived expression for the improvement factor involved with P_d and P_{fa} is reported in [22] and can be seen in (2.25).

$$[I(n_p)]_{dB} = 6.79(1 + 0.235P_d)\left(1 + \frac{\log\left(\frac{1}{P_{fa}}\right)}{46.6}\right)\log(n_p) \qquad (2.33)$$

As can be seen from the above equations, to resolve the range and Doppler ambiguities, radar uses formation of scans which utilize pulse integration.

2.4. Formation of Tracking

Search scan is not enough to describe the evaluation of target classification. Trajectory of the specific targets is also needed for the quality of detection and the target classification. In addition to search scans, radar use tracking techniques for clarification of the target identity.

In this section, system model and tracking methods used in the thesis are discussed. This subject has enormous number of application fields; therefore, this section only aims to introduce the topics planned for future usage.

2.4.1. Target Motion Models

Setting up the correct target motion modeling is one of the important issues for target tracking. Whenever there is any lack of knowledge about the state estimation, it causes uncertainty to increase sharply. The knowledge of the states allows to predict the future target dynamics clearly. Hence, it is required to describe a proper target behavioral model. In [23], target motion models are classified into categories which are stated below.

- **Non-maneuver Models**

  These types of motions are described by the state vector $\dot{x}(t) = 0, where\ x = [\dot{x},\ \dot{y},\ z]^T$ are the Euclidian coordinate system elements. Motions in the model are expressed as straight-line segments and remain at constant velocity. We should note that when describing the target motion, the velocity is approached as a vector and the speed as a magnitude. Non-maneuver models are also referred as uniform motion models.

- **Coordinate-Uncoupled Maneuver Models**

  In these types of motions, maneuver input is modeled as a specific random process, so target maneuvers randomly in each specific direction. This model is divided into three groups which are White Noise Model, Markov Process Model and Semi-Markov Jump Process model.

- **2D Horizontal Motion Models**

  This model is proposed for tracking of a target moving in the horizontal plane. So, the targets maneuver with a known turn-rate in 2D plane. The maneuvering input modelled in are generally based on specific target kinematics, the others, which is stated before, are based on random processes.

- **3D Motion Models**

  The model is designed for tracking highly maneuver and agile targets. These models are mostly used in air defense systems.

The specified models except the non-maneuver CV model are out of scope for our subject which are surveyed detail in [23]. The main idea of the model is explained below.

It is well known that, in tracking scenarios, a target is treated as a point object. In order to describe a point target, a state vector is defined in the cartesian coordinate system which is

$$[x,\ \dot{x}, y, \dot{y}, z, \dot{z}]$$

(x, y, z) denote the position and $(\dot{x}, \dot{y}, \dot{z})$ the velocity, respectively. In the CV model, it is assumed that $\dot{x}(t) = 0, where\ x = [\dot{x},\ \dot{y},\ z]^T$. For non-maneuvering motion, target maneuvers are in the x-y plane, so z direction is treated differently. Furthermore, in most cases, the acceleration of each direction is assumed as a random variable ($\dot{x}(t) = w(t)\ \approx 0$, where $w(t)$ is white noise) and this approach accounts for uncertain motion of the target trajectory. The corresponding state-space model is described by,

$$\xi(t+T) = F(T)\xi(t) + w(t,T)$$
$$= \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \xi(t) + w(t,T) \quad (2.34)$$

where $w(t,T)$ is the driving input for the target maneuvering in the interval from t to $t+T$. $w(t,T)$ is gaussian white noise for describing uncertain motion. In addition to the state model, the observation can be modeled as a function of the state transition matrix. Let y be an observation state at time t, it is expressed by

$$y(t,T) = H(T)\xi(t) + v(t,T)$$
$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \xi(t) + v(t,T) \quad (2.35)$$

This model stated in 2.35 describes 2D CV motion model. States of the system are the target position and the velocity. Additionally, $w(t,T)$ and $v(t,T)$ used in process and observation steps are Gaussian white noise.

After building up the target motion model, proper choice of tracking techniques must be used. Selection of tracking techniques is based on estimation theory. That is why, tracking methods are classified by stochastic properties of estimation variable. In the following chapters, tracking algorithms used in the thesis are discussed briefly.

2.4.2. Tracking Filters

2.4.2.1. Kalman Filter

The Kalman filter is a type of recursive filter used to estimate the state of a linear dynamic system from a series of noisy measurements. The word “filter” is used, because if all noise in the system is Gaussian, Kalman filter minimizes the mean square error of the estimations, in other terms, Kalman filter filters out the noise. Main theory of Kalman filter is based on estimation of a joint probability distribution of the states for each time frame. The joint probability distribution of the states is used to minimize the uncertainty of the trajectory.

Assume that the estimation is made by the kinematic state model assumption which are 2.34 and 2.35 and $\xi(t)$ denote the measurement data up to time t, the prediction step is described as follows:

State estimation

$$\hat{\xi}_{n|n-1} = F(T_n)\hat{\xi}_{n-1|n-1} \tag{2.36}$$

$$P_{n|n-1} = F(T_n)P_{n-1|n-1}F(T_n)^T + Q(T_n) \tag{2.37}$$

Measurement Estimation

$$\hat{y}_{n|n-1} = H\hat{\xi}_{n|n-1} \tag{2.38}$$

$$S_{n|n-1} = HP_{n|n-1}H^T + R(T_n) \tag{2.39}$$

where $\hat{\xi}$ is the estimation of state, P is state variance matrix, Q is the covariance of the process noise, R is the covariance of the observation noise and S is the innovation covariance.

- After state estimation process, the measurement information is received by the radar and the measurement residual is given by

$$\tilde{y}_n = y_n - \hat{y}_{n|n-1} \tag{2.40}$$

- To update the state distribution and minimize the error, it should be calculated the joint probability density of the measurement and distribution matrices named as Kalman Gain Matrix which is expressed as K.

$$K_n = P_{n|n-1}H^T\left(HP_{n|n-1}H^T + R_n\right)^{-1} \tag{2.41}$$

$$= P_{n|n-1}H^T S^{-1}{}_{n|n-1}$$

- Finally, the estimate of the states and Kalman Error Covariance Matrix must be corrected using Kalman Gain.

$$\hat{\xi}_{n|n} = \hat{\xi}_{n|n-1} + K_n\left(y_n - \hat{y}_{n|n-1}\right) \tag{2.42}$$

$$P_{n|n} = (1 - K_n H)P_{n|n-1} \tag{2.43}$$

2.4.2.2. PDAF Filter

The second filter used in the thesis is the probabilistic data association filter. Basically, The PDA filter is based on Kalman filter theory. However, the use of Bayesian theorem in the model distinguishes PDAF from Kalman filter. PDAF is specifically used for removing measurement uncertainty rather than assignment of measurements to a target which is the goal of Joint PDAF. Main idea of PDAF algorithm is to calculate the probability that each validated measurement is attributable to the target of interest.

The outline of the algorithm described in [24] is explained below:

- State estimation stated by equations 2.36-2.39 is applied.
- Different from Kalman approach, the measurement validation region is calculated by the following equations,

$$v(n,\gamma) = \{y: [y - \hat{y}(n|n-1)]^{-1} S^{-1}(n)[y - \hat{y}(n|n-1)] \leq \gamma\} \tag{2.44}$$

$$v(n) = c_{n_z}|\gamma S(n)|^{\frac{1}{2}} = c_{n_z}\gamma^{\frac{n_z}{2}}|S(n)|^{\frac{1}{2}} \tag{2.45}$$

γ is the gate threshold corresponding the gate probability P_G.

c_{n_z} is the volume of the n_z-dimensional unit hypersphere depends on the dimension. For example, $c_1 = 2$, $c_2 = \pi$ and $c_3 = \frac{4\pi}{3}$.

- Using validation region, data association probability for each validated measurement is given by the following equations,

$$\beta_i(n) = \begin{cases} \dfrac{\mathcal{L}_i(n)}{1 - P_D P_G + \sum_{j=1}^{m(n)} \mathcal{L}_i(n)}, i = 1, \dots, m(k) \\ \dfrac{1 - P_D P_G}{1 - P_D P_G + \sum_{j=1}^{m(k)} \mathcal{L}_i(k)}, i = 0 \end{cases} \quad (2.46)$$

$$\mathcal{L}_i(n) \triangleq \frac{\mathrm{N}[y_i(n); \hat{y}(n|n-1)\ ), S(n)] P_D}{\gamma} \quad (2.47)$$

Where $i = 0$ indicates that no measurement is correct, P_D is the target detection probability, P_G is the gate probability and $\mathcal{L}_i(k)$ is the likelihood ratio of the measurement $y_i(k)$ originating from the target rather than clutter

- After the association probabilities are calculated, the effect of measurement uncertainty on states and its covariances also need to be updated

State Update Equation

$$\hat{\xi}_{n|n} = \hat{\xi}_{n|n-1} + K_n v(n) \quad (2.48)$$

Combined Innovation

$$v(n) = \sum_{i=1}^{m(k)} \beta_i(n). v_i(n) \quad (2.49)$$

Gain is the same as Kalman Filter

$$K_n = P_{n|n-1} H^T S^{-1}{}_{n|n-1} \quad (2.50)$$

Covariance Updated State

$$P_{n|n} = \beta_0(n)P_{n|n-1} + \left(1 - \beta_0(n)\right)P_{n|n,Kalman} + P_n \qquad (2.51)$$

$$P_n = K_n \left[\sum_{i=1}^{m(k)} \beta_i(n).\, v_i(n)v_i(n)^T - v_i(n)v_i(n)^T \right] {K_n}^T \qquad (2.52)$$

In the light of the foregoing statements, it can be said that, when the measurement step is not progressed, the error covariance matrix $P_{n|n}$ is increased by weighting $\beta_0(n)$, and whenever new measurement step is made and the measurement is correct, the error covariance matrix is decreased by weighting $\left(1 - \beta_0(n)\right)$. However, if none of the validated measurements $m(k)$ is correct, this case would increase the error covariance matrix.

To summarize, when the clutter or other effects are dominant on the target tracking, PDAF is used to decrease measurement uncertainty. To do this, it calculates the association probability for each measurement and by the help of these weighting coefficients, the state information matrices are updated.

CHAPTER 3

RESOURCE MANAGEMENT IMPLEMENTATION

3.1. Radar Resource Management

MF Radars must be able to operate in overload situations. For this reason, it is critical for MF Radar to provide updated target database which satisfies Quality of Service (QoS) requirements. In intensive target density conditions, MF Radar cannot preserve all targets in tracking state because it does not have enough radar resources to perform all required tasks simultaneously. In this case, efficient usage of resources becomes significant. The radar resource management system must evaluate the allocation of resources and select the tasks to provide the best performance. Therefore, radar resource management must assign the allocation of limited resources in an optimal way to satisfy the task requirements. In the light of this information, RRM has two different missions: task prioritization and task scheduling.

In the previous section we tried to explain the building blocks of the radar system and how it works. In order to accomplish the resource management, radar performance model and the optimization problem must be well-defined. Among the resource allocation problems, it is proposed that use of adaptive methods for tracking targets is an efficient way to manage resource allocation in radar systems. Therefore, in [1], a novel method is proposed for optimization-based scheduling of the tracked targets to maximize the utility functions individually in the finite time horizon. The method assumes that there are quantified performance measures which describe the track-wise quality properties such as track accuracy and track continuity. To generate the radar objective function, aggregation should be made over three entities: time, tracks, and the tracking accuracies. After defining the radar objective function, the method aims to maximize the overall performance measure (track-wise) over a finite time interval by using dynamic programming. In some studies, dynamic programming-based stochastic optimization of scheduling has also been presented [25-26]. However, these approaches are too restrictive because of the dimensionality of the stochastic optimization problem and they do not include the sector scan utilities in the optimization problem. To overcome the high dimensionality of the state transitions, usage of an alternative sub-optimal model is suggested. The model which is given in [1] is based on following aspects.

i) Performance Calculations

The model calculates the radar performance by adding a set of target-wise individual utility functions (track continuity, track performance etc.) between future time intervals. To find the performance of the radar in the future, target performance measurements are modeled as a finite state stochastic discrete event system. Therefore, future measurement decisions can be made by using a Markov chain constructed with performance predictions.

ii) Scheduling Approach

In [1], radar scheduling is divided into two-time scales, slow and fast time scheduling. In slow time scale, radar decides the measurement operations for the next measurement batch interval. Decision contains specific tasks like sequential track update request on a target which is the output of the resource management. Fast time scheduler applies the tasks based on the slow time decisions while the radar is fetching the measurements. In other words, fast time scheduler prepares the output of the slow time decisions to minimize the ambiguity resolution. In this thesis, fast time scheduling is not included in the resource allocation problem. The optimization problem may involve fast time scheduling, but this will lead to not only computational complexity but also has negligible effect in the solution. For this reason, we did not perform fast time scheduling in resource allocation step. As a result, computational complexity is reduced considerably. Scheduling approach and its operations can be seen in Figure 3.1.

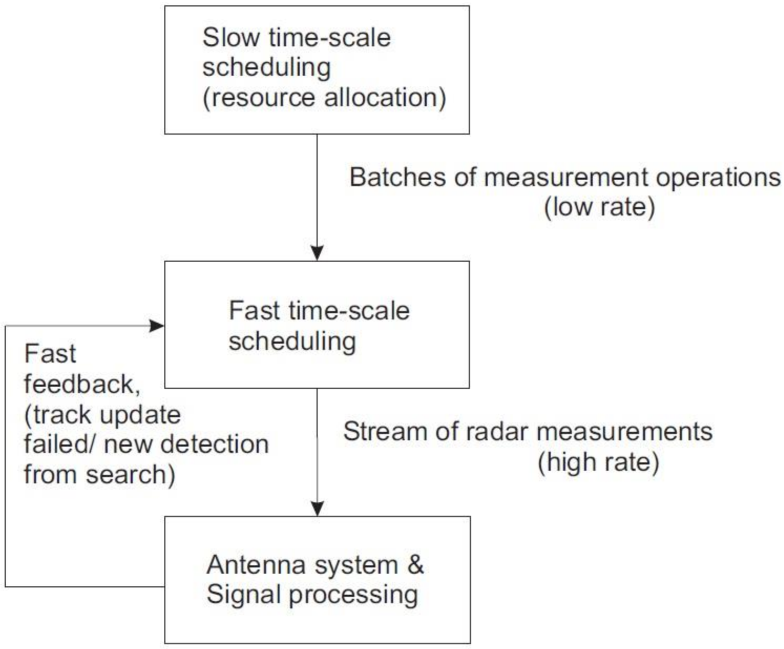

Figure 3.1: Two-Time Scale Scheduling [1]

If we call the starting time of resource allocation t_0, maximum performance, which is based on tracking accuracy, will be predicted by using partially observable Markov decision process for each target until time reaches $t_0 + t_{batch\ interval}$. This time interval is called as slow time scale scheduling. At the end of this time interval, the predicted best policies of targets are generated. The policies are applied in each fast time interval time which is also called as the measurement batch interval. The new predicted policies are generated at the slow-time iterations, namely for each total batch of measurement intervals.

iii) Resource Constraints

As stated above, due to the nature of radar measurements and target dynamics, performance measurements of the MFR are not affected dramatically in slow time scheduling. But there is a necessity that, the stream of radar measurement operations must be performed within a certain time interval. In [1], it is stated that there is only one resource constraint in the optimization model which is "each fast time operations that occurs within slow time scheduling shall not exceed a specific time interval". Hence, we will try to comply with this statement in our optimization problem.

3.1.1. Performance Calculations of the Model

Radar resource management aims to maximize the QoS level. And QoS components can differ according to the radar operation. In optimization-based scheduling, QoS components may generally consist of track continuity, track accuracy, total target database and target drop rate. In this part of the chapter, we aim to describe the overall objective function which is based on the approach given in [1].

3.1.1.1. Utility Function

System utility function of the control system in time interval Δt is generally defined as,

$$Overall\ Utility\ Function = \int_{\Delta t} \sum_{i} U_i(t)dt = \sum_{i} \int_{\Delta t} U_i(t)dt \tag{3.1}$$

where i is the number of performance measurements and $\Delta\mathrm{t}$ is the measurement batch interval. To make a decision in resource management systems, we have to take both present and future performance outputs into consideration instead of calculating instantaneous performance. Therefore, the overall utility function is defined by integrating the expected instantaneous utility over a time horizon. Then, we represent the overall utility function as,

$$J\big(x(t_0\,)\big) = E\left\{\sum_{k=0}^{N-1} \int_{t_0+k\Delta_t}^{t_0+(k+1)\Delta_t} \sum_{i=1}^{M} U_i\big(x_i(t)\big)\, dt | x(t_0\,)\right\} \tag{3.2}$$

$$\approx E\left\{\sum_{k=0}^{N-1}\sum_{i=1}^{M}U_i\big(x_i(t_0+k\Delta_t)\big)\Delta_t\Big|x(t_0\ )\right\}$$

$x(t)$ is the augmented state matrix which covers all performance measurement and N is the horizon of decision-making. $U_i\big(x_i(t)\big)$ is the tracked-wise utility function of the radar system and defined as,

$$U_i\big(x_i(t)\big) = U_{nom,i}Q_{acc,i}(x_i(t))x_{i,tracked} \tag{3.3}$$

$U_{nom,i}$ is constant value and it can be modified by the user according to the priority of the targets. $Q_{acc,i}$ is the performance function and needed to be normalized because of the Bellman conditions that should be satisfied. $x_i(t)$ is the target performance state and $x_{i,tracked}$ is the state variable that takes value 1 when target is tracked, 0 otherwise.

As it can be seen in (3.2), the expected utility function is built independently for each target. This will provide flexibility to convert the resource allocation problem to a known optimization problem by reducing number of state dimensions.

3.1.1.2. Cost Functions

There may exist certain events in radar operations which decrease the radar performance. For example, track mixing, target drop in tracking, track re-initialization will cause radar to reorganize its operational procedure. Therefore, the cost function, which affects the objective function negatively, should be included in radar system, since the radar must ensure the identity of the target, and try to guess the intentions of the target. Cost functions added to overall objective function are described below.

3.1.1.2.1. Track Re-initialization Event

The cost function of re-initiation events is straightforward which is stated as,

$$C_{reinit}(t_n) = C_{reinit,i}x_{i,tracked} \tag{3.4}$$

$C_{reinit,i}$ is the constant value that appears in the track re-initiations event. This cost function is added in the objective function when the target is dropped and then detected again after a certain time.

3.1.1.3. Track Mix Event

Track mixing event occurs depending on various situations, such as target interactions. To find the track mixes event, multivariate Gaussian distribution covariance matrix is extracted for each tracking target. When the Mahalanobis distance of two targets falls below a certain limit, track mixes event is triggered which is described as,

$$C_{mix,i}(t_n) = C_{mix,i} x_{i,tracked} \tag{3.5}$$

$C_{mix,i}$ is the constant value that appears in the track mixing event.

$x_{i,tracked}$ is state variable that it takes value 1 when target is tracked else 0.

3.1.1.4. Objective Function

In (3.2) interval from t_0 to $t_0 + k\Delta_t$ is referred as the slow time- total measurement batch interval. In this interval, radar will perform its own duties such as TWS or adaptive tracking and searching between determined sectors. According to [1], sensitivity of tracking performance in surveillance radar is low with respect to these observation time variations, and there will be many sequences of scans yielding almost the same performance. Therefore, the essential dynamics of tracking performance are assumed to be captured on the slow timescale, given as one second time intervals. Consequently, the time between updates of the Markov chains for QoS prediction can be adjusted to the slow timescale without affecting prediction performance considerably. This fact should be utilized to reduce computational demand. Optimization problems involving this type of objective functions stated in (3.2) can be solved by many possible ways, one of which is backwards recursion DP algorithm. In this case, “value to go” objective function will be used as follows.

$$\mathrm{J}_k(\mathrm{x}_k) = \mathrm{E}\left\{\sum_{n=k}^{N} U(\mathrm{x}_k) | \, x_k\right\} \tag{3.6}$$

The DP backwards recursion is expressed as,

$$J_k^*(x_k) = max_{d_k}\left(U(x_k) + \sum_{x_{k+1}\in X} J^*_{k+1}(x_{k+1})P\left(x_{k+1}\middle|x_k, d_{k,}, \xi_k\right)\right) \tag{3.7}$$

where $J_k^*(x_k)$ is the optimal value of $J_k(x_k)$.

If the Markov chain is not stationary, multi-armed bandit model cannot be used. As an example, let $p_{x_{i,k}}$ be the state probability vector of $x_{i,k} = x_i(k\Delta t)$ and let $x_{i,k}$ as a Markov chain which is affected by control action d_k. Then, state probability evaluation can be modelled by,

$$p_{x_{i,k+1}} = P_{tr,i}\left(d_k, \xi_{i,k}\right)p_{x_{i,k}} \tag{3.8}$$

where $P_{tr,i}$ is the probability transition matrix of the Markov decision process, and $\xi_{i,k}$= ξ_i $(k\Delta t)$ is the kinematic state of target i. Overall state probability matrix can be expressed as follows.

$$p_{x_k} = p_{x_{1,k}} \otimes p_{x_{2,k}} \otimes p_{x_{3,k}} \otimes ... \otimes p_{x_{M,k}} \tag{3.9}$$

where $\otimes$ denotes Kronecker (tensor) product. In addition to that, the aggregated state variables can be expressed as,

$$X = X_1 \times X_2 \times ... \times X_M \tag{3.10}$$

Where $\times$ denotes Cartesian product. Since tracking performance values are independent of each other, total state transition matrix can be composed as in the equation (3.11)

$$P_{tr}(d_k, \xi_i) = P_{tr,1}(d_k, \xi_{1,k}) \otimes P_{tr,2}(d_k, \xi_{2,k}) \otimes \ldots \otimes P_{tr,M}(d_k, \xi_{M,k}) \tag{3.11}$$

The augmented state probability vector p_{x_k} evolves according to,

$$p_{x_{k+1}} = P_{tr,k}(d_k, \xi_k) p_{x_k} \tag{3.12}$$

After defining state transitions, calculation steps for DP are as follows. Let n be the time horizon that we try to maximize the expected aggregate reward. Decision process starts from time $t = 1$ and lasts until $t = n - 1$. To find the best policies that maximize the reward, the transition matrix and the corresponding reward matrix are multiplied, and this augmented matrix should be used recursively. This method stated in [9] is explained as follows:

$r_i^{(k)}$ is the instantaneous reward at $t = 1$ and state i, $P_{ij}^{(k)}$ is transition probability from state i to state j with control input $\boldsymbol{k}$. At $t = 2$ the objective function will be,

$$J_i^*(2, u) = max_k \left\{ r_i^{(k)} + \sum_j P_{ij}^{(k)} u_j \right\} \tag{3.13}$$

The recursive expression for $t = n$ is as follows.

$$J_i^*(n, u) = max_k \left\{ r_i^{(k)} + \sum_j P_{ij}^{(k)} J_j^*(n-1, u) \right\} \tag{3.14}$$

Using recursive expression in each time, DP algorithm extracts the optimal policies $\boldsymbol{k}$ for a final reward vector u between the time horizon $\boldsymbol{n}$.

While solving the optimization problem stated in (3.2) using DP, if the performance evolution model is a nonstationary Markov Decision Process, complexity of the state transition might be too high due to the overwhelming state matrices. The main concern about this calculation is the complexity level due to the increasing number of state variables over time and including timing constraints in the problem. As a result, the problem becomes unsolvable in a reasonable time. That is why, the problem should be simplified or relaxed. The following sections include the simplified approaches against this concern.

The first approach to simplify the problem is to create a target-wise state performance matrix by separating the overall state performance matrix. In this way, global radar task will be decomposed into subtasks and target-wise utility functions will be used to make optimization problem tractable. In radar operations, target performance estimations are assumed independent of each other. Therefore, the system state transition matrix will not get larger, but instead of this, state transition matrix will be used for each target separately. The new target-wise state probability vector can be expressed as the following.

$$p_{xi_{,t_0+t_h}} = P_{tr,i}\left(d_{i,t_0}, \xi_{i,t_0}\right) p_{x_{i,t_0}} \tag{3.15}$$

where $P_{tr,i}$ is the probability transition matrix for target i and ξ_{i,t_0} is the target position. Target-wise overall utility function at time t_0 will be as follows.

$$U(x(t_0)) = \sum_{i=1}^{M} U_i(x_i(t_0)) \tag{3.16}$$

where $x_{t_0} = \{x_i(t_0)\}_{i=1}^{M}$, M is the number of targets.

Continuous objective function for single target i is expressed by (3.17)

$$J_i\big(x_i(t_0\,)\big) = E\left\{\int_{t_0}^{t_0+t_h} x_{i,tracked}(t)U_{nom,i}\big(x_i(t\,)\big)Q_{acc}\big(x_i(t\,)\big)\,dt\right.$$

$$- \sum_{n|t_n\in[t_0,t_0+t_h]} C_{reinit,i}x_{i,tracked}(t_n) \qquad (3.17)$$

$$\left. +\; C_{mix,i}x_{i,tracked}(t_n)\mid x(t_0)\right\}$$

As stated before, performance value for each target can be considered as independent, however, this assumption may be invalid if targets are too close to each other. This case requires neighboring or interacting targets to be added to the objective function as new costs. Therefore, in equation (3.17), track mixing and re-initiation cost functions are added to the objective function. Consequently, independence assumption becomes valid for all cases. Moreover, (3.17) is the case when the radar fetches a single scan measurement. For this reason, time between t_0 and $t_0 + t_h$ is called the measurement batch interval. In addition, if sector scan time does not exceed one second, it is assumed that utility functions or performance values are not affected dramatically in the measurement interval, integration terms in (3.17) can be disregarded and the equation should converge to summation of target-wise utility functions. In resource allocation problem, to determine the best policy, the performance of the upcoming measurements should also be predicted. Since radar measurements are not deterministic, future performance predictions will also be probabilistic. Hence, radar resource allocation problem should be approached as a stochastic control problem. Considering the future decisions of the radar, future utilities must be included so that target dynamics should be modelled as Partially Observed Semi-Markov Decision Processes (POMDP). However, the optimization problem arises when the target estimations have continuous infinite state space. Since states are growing after each iteration, the solution will be intractable. To overcome this problem, target estimation should be modelled in a form of a generalized finite-state semi-Markov decision process (GSMP). In GMSP, a decision is made in time t, the model parameters are determined as a result of the decision, and another decision will be made in the next iteration. Due to Semi-Markov property, the time until the next decision epoch and the state at that epoch depend only on the present state and the subsequently chosen action and are independent of the past history of the system property. In the chapter [3.3], we aim to build a stochastic system based on performance measurements which is in a form of a generalized semi-Markov decision process (GSMP).

3.1.1.5. Constraints

Radar measurements are accomplished by sending a signal consisting of consecutive pulses and receiving the echoes of the pulses. This is the main task of the radar. These processes are repeated periodically and must be completed within the specific time interval. Therefore, we will refer about a single resource constraint in here that measurement batch interval must not exceed a specific time. $I_{batchInterval,j}(t)$ is a function which shows that target number j is scanned in the batch interval.

$$I_{batchInterval,j}(t) = \begin{cases} 1, If\ target\ number\ j\ is\ processed\ at\ time\ t \\ 0,\ \ otherwise \end{cases} \tag{3.18}$$

and the constraints are expressed as the inequality of process time of the tasks.

$$\int_{k\Delta t}^{(k+1)\Delta t} \sum_{j} I_{batchInterval,j}(t)dt \ \leq\ \Delta t \tag{3.19}$$

Δt Each batch interval time

j Target Task Number

We define two variables, which are the allocated time of a target at each scan attempt and the number of target scans, and express these as $u_j(k)$ and s, respectively. Then, $l_{s,k}$ is described as the total load for a scan at measurement batch time interval k.

$$l_{s,k} = \sum_{\{j|\ j \in stream\ of\ \ s\}} \frac{u_j(k)}{\Delta t} \tag{3.20}$$

However, the time load of the task sequences is random because of environmental effects on target kinematics. Therefore, the total load time in a batch interval will be clearly random. To overcome this restriction, the constraints in (3.20) should be modified with constraints based on the expected processing time defined as (3.21).

$$E_{u_{upd},i,k|x_{i,k},d_{upd,i,k}\forall i}\{l_k\} \leq c_k \tag{3.21}$$

3.1.1.6. Lagrangian Relaxation Model

In optimization model, which is based on (3.6), if the constraints (3.21) are added into the objective function like cost functions with specific weights then Lagrange relaxation method can be applicable. Consequently, this relaxed problem can be solved more easily. So, the Lagrangian is defined as,

$$\begin{aligned} L_k\left(x_{k,}d_{k,}\lambda_k\right) = U(x_k) + \lambda_k\left(c_k - E_{u_{upd},i,k|x_{i,k},d_{upd,i,k}\forall i}\{l_k, d_k\}\right) + \\ E_{x_{k+1}|x_k,d_k}\left\{max_{d_{k+1}} L_{k+1}\left(x_{k+1,}d_{k+1,}\lambda^*{}_{k+1}(x_{k+1})\right)\right\} \end{aligned} \tag{3.22}$$

where λ_k is the Lagrange multiplier at time k and $\lambda^*{}_{k+1}(x_{k+1})$ is the expected Lagrange multiplier that represents a penalty that satisfy the constraint at time $k+1$. The optimal Lagrangian at time k, $L_k\left(x_{k,}d^*{}_k(x_k)\lambda^*{}_k(x_k)\right)$ is equal to the optimal value-to-go function $J_k^*(x_k)$. At the end of decision time interval N, Lagrangian is equal to utility which is $L_N(x_N) = U(x_N)$.

This equation is computationally intractable because of nested expectations. To simplify the equations for dynamic programming, we must get rid of the nested expectations and separate our utility function independently. A separation of the problem requires evaluating all the target utility by itself. Therefore, the Lagrangian is recomposed as follows,

$$L_k(x_{k,}d_{k,}\lambda_k) = \sum_s U_s(x_{s,k}) + \lambda_k \left(c_k - \sum_s l_{s,k}(x_{s,k,}d_{s,k})\right)$$
$$+ E_{x_{k+1}|x_k,d_k}\left\{max_{d_{k+1}} \sum_s U_s(x_{s,k+1})\right.$$
$$+ \lambda^*{}_{k+1}(x_{k+1})\left(1 - \sum_s l_{s,k+1}(x_{s,k+1,}d_{s,k+1})\right) \tag{3.23}$$
$$\left.+ E_{x_{k+2}|x_{k+1},d_{k+1}}\{max_{d_{k+2}} L_{k+2}(x_{k+2,}d_{k,+2},\lambda^*_{k+2})\}\right\}$$

Another problem arises while separating utility functions independently. $\lambda^*{}_k(x_k)$ is a function of global state, therefore the algorithm is getting complex to be solved. To overcome this problem, it is assumed that the variance of $\mathrm{E}_{\mathrm{x_k|x_0}}\{\lambda^*{}_{\mathrm{k}}(\mathrm{x_k})\}$ is small because there will be many independent subtasks. So estimated $\lambda^*{}_{\mathrm{k}}$ will be available where the estimates are chosen such that,

$$E_{x_k|x_0} E_{u_{upd},i,k\forall i|x_k,d^*{}_k(x_k)} \left\{l_k\left(x_{k,}d^*{}_k(x_k)\right)\right\} = 1 \tag{3.24}$$

Finally, we rearrange equation again with respect to $\lambda^*{}_{\mathrm{k}}$, the Lagrangian function turns into the following.

$$\begin{aligned} &L_{s,k}(x_{s,k}d_{s,k}\lambda_k) \\ &= U_s(x_{s,k}) - \lambda_k l_{s,k}(x_{s,k}, d_{s,k}) \\ &+ max_{d_{s,k+1}} E_{x_{s,k+1}|x_{s,k},d_{s,k}} \{L_{s,k+1}(x_{s,k+1}d_{s,k+1,}\hat{\lambda}^*{}_{k+1})\} \end{aligned} \tag{3.25}$$

$$L(x_k d_k \lambda_k) = \sum_s L_{s,k}(x_{s,k}d_{s,k}\lambda_k) + \lambda_k + \sum_{n=k+1}^{N-1} \hat{\lambda}^*{}_n \tag{3.26}$$

In the following sections, auxiliary models and simplifications that will help to solve this problem with dynamic programming will be discussed.

3.1.1.7. Finite-State Markov Modeling

For computational flexibility in (3.26), the optimization solution should be based on track-wise rewards using Partially Observed Markov Decision Processes (POMDP). By this way, performance state evaluation will be updated according to the generalized finite-state semi-Markov decision process at each future decision step. We now formulate a dynamic model for the radar, where the instantaneous tracking performance is a function of the state of this dynamic model. As stated before, in the system scenario, trace of the Kalman Error Covariance Matrix (P) indicates the accuracy of tracking. In the tracking scenario, this matrix is predicted using (2.31) and updated using (2.37). If the trace of the process covariance matrix is not changing in time, we can calculate the prediction of steady-state covariance $P_{steadystate}$ exactly, by solving Ricatti equations. In case of continuous track update demand by the radar resource allocation, Kalman error covariance matrix gets the minimum value and remains constant. Inverse of the trace value of this matrix can be used as the **maximum utility value** that the target can reach in the tracking process. This value can be changed by depending on the controlled input, which can be one of two possibilities, “update target tracking” or “do not update target tracking” demand by radar control center. If the target decision is “do not update”, error covariance matrix will not be updated by Kalman filter, and error covariance matrix will get larger. After many successive “do not update” decision, validation gate level in tracking process gets higher and tracking process may confront the target drop, which will be the worst state of the Markov chain. Successive “update” demands will lead to the best state of the Markov chain however will bring higher resource demand. When the target is dropped in tracking process, to re-initiate the target, radar must detect the target in successive measurements. Typical state diagram in radar system can be seen in Figure 3.2.

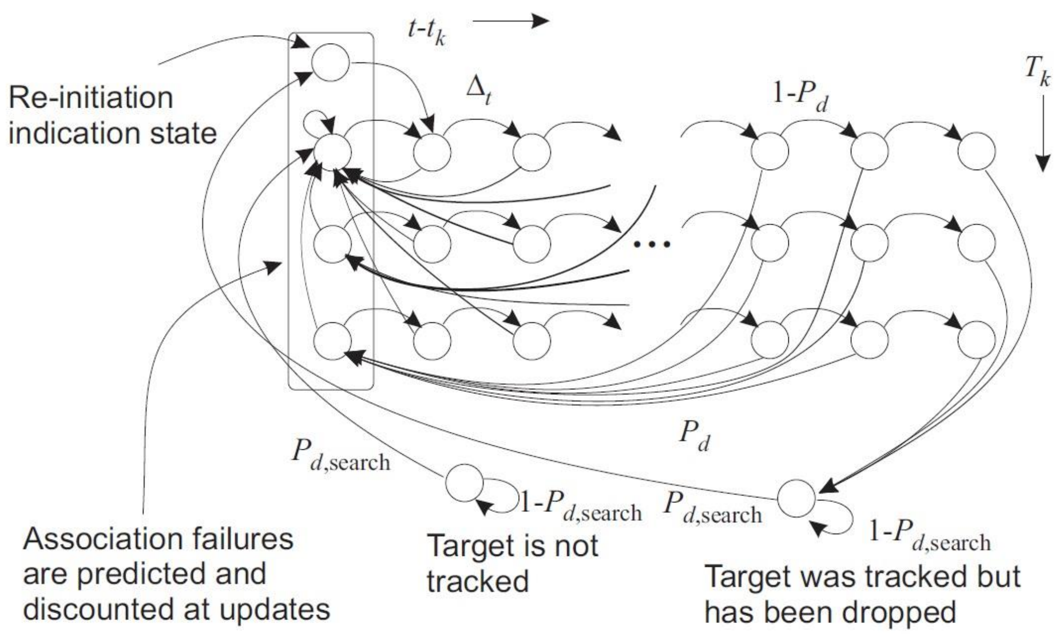

Figure 3.2: Typical State Diagram in Radar System [1]

According to target kinematics, trace of error covariance matrix may be infinite. To simplify the optimization solution, Markov chain related to the performance value should be converted to finite-state MDP. As stated before, trace of steady state Kalman error covariance matrix is the maximum performance value. Besides, the minimum performance value must also be found because of the discretization needs. To find the minimum performance value, we perform the Kalman filter recursion up to the horizon of decision-making interval and ignoring the measurement update equations, we can find the minimum performance value by taking trace of error covariance at the end of decision-making interval. Discretized covariance step of our state diagram is stated in (3.28).

$$P_{discretizationstep} = \frac{trace(P_{max}) - trace(P_{min})}{Total\ Number\ of\ States - 1} \tag{3.28}$$

Once it has been found the trace of error covariance matrices for each Markov states, the utility values will be like the following.

$$U_i\big(x_i(t)\big) = U_{nom,i} * \frac{1}{Trace(State\ Error\ Covariance\ Matrix\ x_i)} * x_{i,tracked} \qquad (3.29)$$

In addition to that, there is a radar target scanning time which is called as look time in here corresponding to each performance state to satisfy quality of tracking. If the target is the state n, total target look time should be calculated as the following and stated in [27].

$$t_{totalLook} = t_{look}(1.P_d + 2P_d(1-P_d) + 3(1-P_d)^2 + \ldots + mP_d(1-P_d)^{m-1} + \cdots + m(1-P_d)^m) \qquad (3.30)$$

where $m = \lceil(\text{n/5})\rceil$, n is the target state and t_{look} is time spent for one look. Equation (3.30) comes from cumulative probability of detection equation. In [6], it is stated that in each CPI trial, probability of detection evolves according to (3.31)

$$P_{cumulative,d} = 1 - \prod_{i=1}^{total\ \#\ of\ CPI} \big(1 - P_{d,i}\big) \qquad (3.31)$$

Example of state topology can be seen in Figure 3.3. In the state diagram, state number 1 indicates the minimum error case (maximum utility, steady state error) and state 26 indicates that the maximum error case where the target drops. Five successive "don't update" decisions cause state to target drop state (state 26).

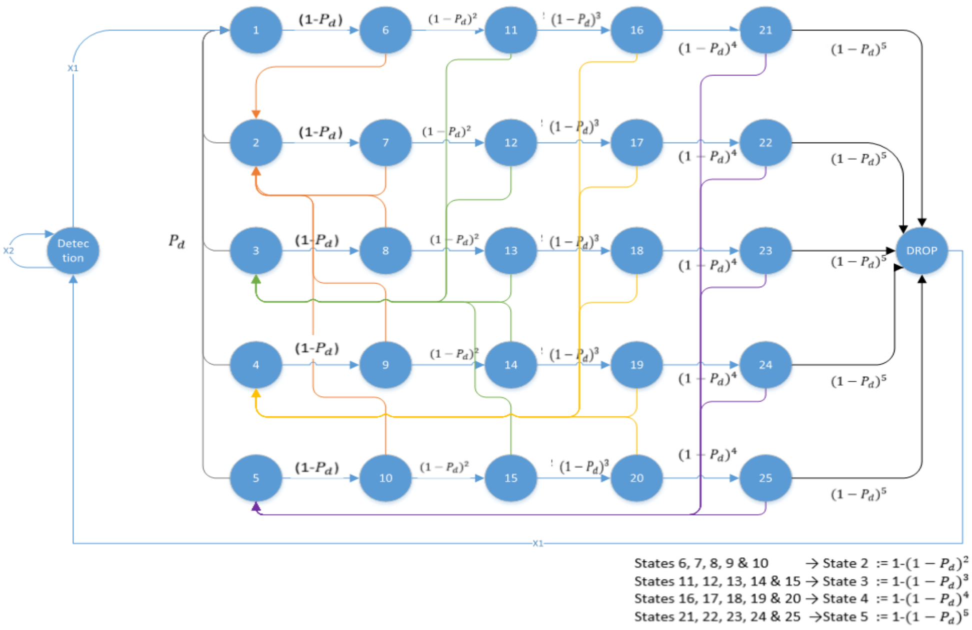

Figure 3.3 Markov Chain Topology

Each target is represented by a twenty-six state Markov chain as shown in Figure 3.3. Markov chain states are numbered by considering their quantized track quality. First state (1) is the best state that has the least trace of the error covariance matrix. Last state (25) corresponds to the highest uncertainty and state 26 is the drop state that the tracker lost the target. The uncertainty depends on the state that the last update has occurred and the duration between two consecutive update instances. If a target is updated, the Markov chain jumps to one of the leftmost states depending on the time.

3.1.1.8. Lagrangian Duality

After clarifying the model assumptions, we solve the optimization problem. Our objective function and constraints are stated in (3.25) and (3.26). This is in the form of known shortest path problem with transient time restrictions. The shortest path optimization problems are easy to solve but adding transient time restrictions to our problems converts the optimization problem to an NP-Hard problem. Therefore, we relax the complicating constraint, benefiting the theory of Lagrangian relaxation. Relaxing the complex constraints refers removing them and putting them into the objective function with specific weights. These weights (λ) called as Lagrange multipliers are used to penalize non satisfied constraints by the help of the objective function. Getting rid of the complex constraints decreases the computational complexity. Our objective function is similar to the form in (3.32).

$$Z(\lambda) \coloneqq \max u^T + \lambda^T(b - Ax) \text{ (3.32)} \tag{3.32}$$

where u^T is utility function and $b \geq Ax$ is constraint. We extracted the dual of the optimization problem stated in (3.26) and then the dual problem can be written as (3.33)

$$\begin{gathered} Z_D \coloneqq \min Z\,(\lambda) \\ where\ \lambda^T \geq 0 \end{gathered} \tag{3.33}$$

$$Z(\lambda^*) = \min\left\{ max \left\{ \sum_{t=0}^{Time\ Horizon} \sum_{n=1}^{Number\ of\ Targets} (\boldsymbol{Utility_{n,t}} + \boldsymbol{\lambda_t}(\mathbf{1} - \mathbf{load}_{\boldsymbol{n,t}})) \right\} \right\} \tag{3.34}$$

Equation 3.34 is solved with the help of dynamic programming. However, how to predict λ values is another important issue. The method for finding optimal value of λ called as Subgradient method steps is stated in [28].

1. At iteration **t = 0**, choose randomly large value for λ_t
2. Compute objective function for this $\boldsymbol{\lambda_t}$
3. Find the subgradient as $\mathbf{g}_{\boldsymbol{t}} = (\mathbf{load}_{\boldsymbol{n,t}} - \mathbf{1})$
4. If $\mathbf{g_t} = \mathbf{0}$, then stop, the optimal solution is $L(\boldsymbol{\lambda_t})$,
5. If $\mathbf{g}_{\boldsymbol{t}} \neq 0$, compute $\lambda_{k+1} = \lambda_k + Q_k^T.\mathbf{g}_{\boldsymbol{k}}$ where $Q_k = \frac{J_{k+1} - J_k}{g_{k+1} - g_k}$ the step size at this iteration.
6. Increment t and go to step 2 with λ_{k+1}.
7. After finding the optimal **λ**, the decision vector x_2 is found from our set of solutions.

While solving the dual of the optimization problem which is stated at 3.20, we ignore the duality gap, because we only want to find optimal policy of targets. It can be left to future studies.

CHAPTER 4

EXPERIMENTAL EVALUATION

4.1. Scenario Models

In this chapter, we present the performance evaluation of the radar resource analysis problem which is based on Lagrange relaxation model. To observe the accuracy of the model, we have run different tracking scenarios which can be considered as the problems that radar will encounter. Besides, we want to observe how the system works by adding a different tracking filter (namely, PDAF) to the model. After confirming Lagrangian-based tracking optimization model, we compared it to Track While Scan and Adaptive Tracking model, then we commented on these three tracking models.

4.1.1. Model Validation Studies

4.1.1.1. Effect of Constraints on the Optimization Model

To see that the model is working correctly, we run the simulation at each time by changing the constraints. In this way, when no constraint exists, optimization model sends continuous tracking command to maximize the objective function. In the other case, in order to satisfy the constraints, the model does not make tracking decisions for specific targets in some measurement batch time interval.

4.1.1.2. Effect of Lagrange Coefficients on the Optimization Model

To see the effects of the coefficient, we run the simulation at each time by changing Lagrange coefficients. Thus, we observe that, the selection of initial Lagrange coefficient has an important role in the optimization problem. When we choose the initial coefficients too high, the optimization model will spend extra effort to fulfill the constraints, the importance of the constraints will become dominant on the dual function and the utility functions will be less effective on the dual. When the initial coefficients are chosen too low, utility function maximization will be dominant, however the model will ignore the constraints, so optimal solution might be infeasible

4.1.2. Effect of Different Objective Functions on the Model

We assign different priority, velocity preference and track mixing cost on each target to see the effect of different objective functions on the model. Preference of the velocity is dependent on the velocity values of the targets. As velocity increases, preference also increases. Preference is added to the utility function as a multiplier.

4.1.3. Effect of Tracking Filter on the Model

The simulation is reperformed with Kalman filter and PDAF to observe the tracking filter effect. Then different filter types are compared.

4.1.4. Comparison of TWS, Adaptive and Lagrangian-based Tracking Model

In the scenario, we compare the performance of LRM with two heuristic tracking methods: Adaptive Tracking (AT), and Track While Scan (TWS). The results of this comparison may be used when selecting the scheduling model.

4.2. Model Description

Our simulation model has the following properties.

- Different target trajectories have been simulated.
- Clutter and target misses have been generated in the measurement operation by adding random scattering and gaps to target trajectories respectively.
- No multipath reflections are included.
- For track initiation, 3 successive target detections are required. In case of 5 successive unsuccessful detections (misses), target is dropped.
- Track correlation is performed by using Mahalanobis distance.
- After tracking starts for all the targets, the best decisions about tracking in the next steps will be determined using Lagrange Duality method.

- In the simulation, we manage to track slowly maneuvering targets by adding certain assumptions to the model for different scenarios. CV model cannot track highly maneuvering targets; however, process covariance matrix values are set to a suitable level for slowly maneuvering targets. Unlike highly maneuvering targets, track missing, which occurs when the target measurement position exceeds the validation gate level, is not observed for slowly maneuvering targets, because measurement of target positions is in the validation gate range.

4.3. Simulation Results

4.3.1. Model Validation Studies

4.3.1.1. Effect of Constraints on the Optimization Model

To verify the Lagrange-based optimization, firstly, the two targets are moved in different trajectories without any resource constraints. As a result, optimization model makes update decision continuously for tracking these targets. Additionally, Kalman covariance matrices of the targets preserve their steady state value while total loads of the targets are reaching 23.4 msec in the measurement batch interval. This can be seen in Table 4.1. In the tables,

- **t** is the number of slow time intervals passed.
- **Decision** information is 0 and 1 for “Do not Update” and “Update” decisions respectively.
- **λ Coeff** is the optimal Lagrange Coefficient which maximizes the objective function at the specific interval.
- **Trace (Pi)** is the trace of the Kalman error covariance matrix for target i, where i=1, 2, 3, 4
- **Time Spent** is the total batch time for all targets at each slow time. Time Occupancy show usage of the timing resources in each slow time interval.

- ***When constraints are large*** (Time limit for 2 Target is 50 msec)

Table 4.1. *Effect of Constraints on the Optimization Model 1*

	t=1	***t=2***	***t=3***	***t=4***	***t=5***	***t=6***	***t=7***	***t=8***	***t=9***
Decisions (Target 1)	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
Decisions (Target 2)	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
λ Coeff	*1000*	*1000*	*1000*	*1000*	*1000*	*1000*	*1000*	*1000*	*1000*
Trace (P1)	*10.5*	*10.5*	*10.5*	*10.5*	*10.5*	*10.5*	*10.5*	*10.5*	*10.5*
Trace (P2)	*10.5*	*10.5*	*10.5*	*10.5*	*10.5*	*10.5*	*10.5*	*10.5*	*10.5*
Time Spent (s)	*0.023*	*0.023*	*0.023*	*0.02*	*0.023*	*0.023*	*0.023*	*0.023*	*0.023*
Time Occupancy (%)	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Secondly, the same simulation is re-performed by changing the time constraint to 20msec. In this case, the optimization model does not make update decision continuously for some targets in order to satisfy the resource constraints.

- ***When constraints are limited*** (Time limit for 2 Target is 20 msec)

Table 4.2. *Effect of Constraints on the Optimization Model 2*

	t=1	***t=2***	***t=3***	***t=4***	***t=5***	***t=6***	***t=7***	***t=8***	***t=9***
Decisions (Target 1)	*1*	*1*	*1*	*1*	*0*	*0*	*1*	*0*	*1*
Decisions (Target 2)	*1*	*1*	*1*	*0*	*1*	*0*	*0*	*1*	*0*
λ Coeff	*1000*	*1000*	*0.171*	*0.137*	*0.088*	*0.070*	*0.07*	*0.05*	*0.045*
Trace (P1)	*10.5*	*10.5*	*10.5*	*10.5*	*30.0*	*73.5*	*12.6*	*31.3*	*11.9*
Trace (P2)	*10.5*	*10.5*	*10.5*	*30.0*	*11.5*	*30.4*	*73.6*	*12.7*	*31.5*
Time Spent (s)	*0.019*	*0.020*	*0.020*	*0.012*	*0.008*	*0*	*0.012*	*0.008*	*0.012*
Time Occupancy (%)	**95**	**100**	**100**	**60**	**40**	**0**	**60**	**40**	**60**

In the second case, it can be seen in Table 4.2 that some of timing loads exceed the constraint. This case will occur when SNR of the target's changes drastically in time (velocity of the target, maneuvering case etc.). To prevent this, decision time interval should be decreased.

4.3.1.2. Effect of Lagrange Coefficients on the Optimization Model

Lagrange coefficients have an important role on the main objective function. When the Lagrange coefficients are too high, the constraints dominate, and optimization model tends to satisfy the constraints only, ignoring the main objective. On the other hand, when the Lagrange coefficients are too low, optimization model tends to maximize the utility function and cost functions while ignoring the constraints.

In Lagrange Duality, the first Lagrange coefficient should be given as high as possible for the utility function to be maximum. But this is a non-feasible situation where the total timing loads have exceeded the constraints and there is a desire to track each target. Therefore, the coefficients which exceed the time limit are reduced to the limit at each iteration. Since coefficient is changed, a new objective value is calculated, and a new time load is obtained. If the initial coefficient is small, the utilities does not reach the maximum value, so the optimum value will not be found. This can be seen in Table 4.3.

- ***When the starting point of the coefficients is 10,***

  In this case, trace error increases for each target (duality → minimization), so the optimization does not obey the constraints and only the main objective is to minimize the overall utility.

Table 4.3 *Effect of Lagrange Coefficients on the Optimization Model 1*

	t=1	***t=2***	***t=3***	***t=4***	***t=5***	***t=6***	***t=7***	***t=8***	***t=9***
Decisions (Target 1)	*0*	*0*	*1*	*0*	*0*	*1*	*0*	*0*	*1*
Decisions (Target 2)	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*1*	*0*
Trace (P1)	*30.00*	*73.50*	*12.67*	*31.37*	*75.93*	*12.79*	*31.64*	*76.59*	*12.80*
Trace (P2)	*30.00*	*73.50*	*153.0*	*13.93*	*33.01*	*80.23*	*167.5*	*13.99*	*33.12*
Time Spent (s)	*0*	*0*	*0.012*	*0.008*	*0*	*0.012*	*0*	*0.008*	*0.012*
Time Occupancy (%)	**0**	**0**	**60**	**40**	**0**	**60**	**0**	**40**	**60**

- ***When the starting point of the coefficients is 200,***

  In this case, trace error decreases for each target, so optimization tries to fit the constraints, and time load is getting high but still remaining below the limit. We can see the effect of duality, trace errors are getting low, and utilities are getting high.

Table 4.4 *Effect of Lagrange Coefficients on the Optimization Model 2*

	t=1	***t=2***	***t=3***	***t=4***	***t=5***	***t=6***	***t=7***	***t=8***	***t=9***
Decisions (Target 1)	*1*	*0*	*1*	*0*	*1*	*0*	*0*	*1*	*0*
Decisions (Target 2)	*0*	*1*	*0*	*1*	*0*	*0*	*1*	*0*	*0*

Table 4.4 (cont'd)									
Trace (P1)	*10.50*	*30.00*	*11.56*	*30.48*	*11.78*	*30.89*	*74.74*	*12.77*	*31.57*
Trace (P2)	*30.00*	*11.56*	*30.48*	*11.78*	*30.89*	*74.74*	*12.77*	*31.57*	*76.74*
Time Spent (s)	*0.012*	*0.008*	*0.012*	*0.008*	*0.012*	*0*	*0.008*	*0.012*	*0*
Time Occupancy (%)	**60**	**40**	**60**	**40**	**60**	**0**	**40**	**60**	**0**

In our algorithm (to find the coefficient), we must set the coefficient limit above 10 and we must decrease it to find the optimal value.

4.3.2. Effect of Different Utility Functions on the Model

4.3.2.1. Effect of Tracking Priorities on the Model

In order to see the effect of priority, we create three targets with different priority, which follows the same trajectory. It can be seen in Table 4.5 that there are different tracking update decisions of the targets according to their priorities.

Target 1: Priority 3

Target 2: Priority 2

Target 3: Priority 1

Initial Case (Three approaching targets): T = 0 sec, Vel= 400 m/s

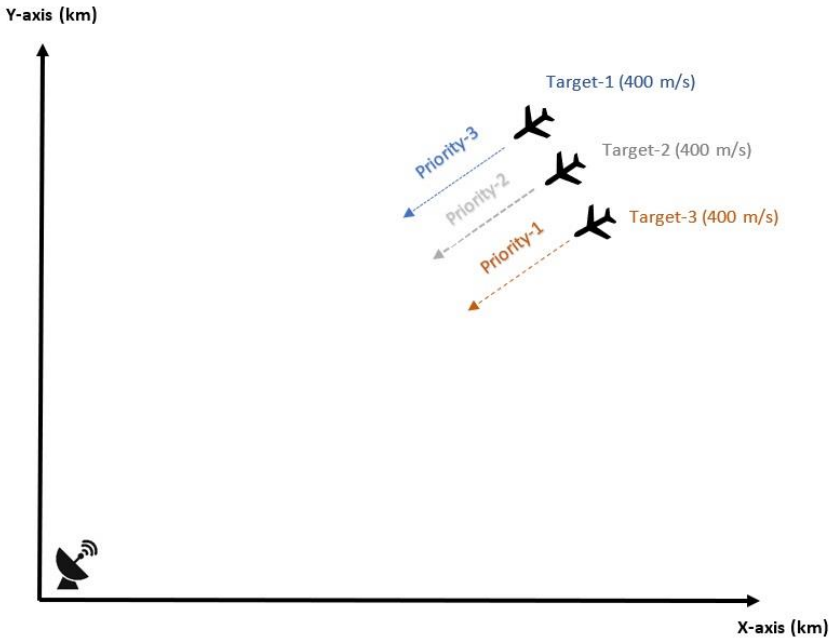

Figure 4.1: Different Target Priorities Scenario

Table 4.5 *Different Target Priorities Scenario*

	t=1	***t=2***	***t=3***	***t=4***	***t=5***	***t=6***	***t=7***	***t=8***	***t=9***
Decisions (Target 1)	*1*	*0*	*0*	*0*	*0*	*1*	*0*	*1*	*0*
Decisions (Target 2)	*0*	*0*	*1*	*0*	*1*	*0*	*1*	*0*	*1*
Decisions (Target 3)	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
λ Coeff	*3.35*	*2.14*	*2.14*	*2.14*	*2.68*	*3.35*	*4.19*	*4.19*	*3.35*
Trace (P1)	*11.81*	*30.93*	*74.80*	*155.4*	*284.7*	*15.32*	*35.18*	*12.10*	*31.70*
Trace (P2)	*31.25*	*75.57*	*156.8*	*13.99*	*33.13*	*80.52*	*12.81*	*31.67*	*11.93*
Trace (P3)	*10.50*	*10.50*	*10.50*	*10.50*	*10.50*	*10.50*	*10.50*	*10.50*	*10.50*
Time Spent (s)	*0.026*	*0.013*	*0.026*	*0.013*	*0.026*	*0.026*	*0.026*	*0.026*	*0.026*
Time Occupancy (%)	**100**	**50**	**100**	**50**	**100**	**100**	**100**	**100**	**100**

4.3.2.2. Different Target Velocities

In order to observe how the model responds to different targets velocities, 3 different trajectories have been created. Two of them are approaching the radar with velocities 1000 m/s and 400 m/s. The other is moving away from radar with the velocity 300 m/s. It can be seen from Table 4.6 that the model makes different tracking update decisions according to the velocities.

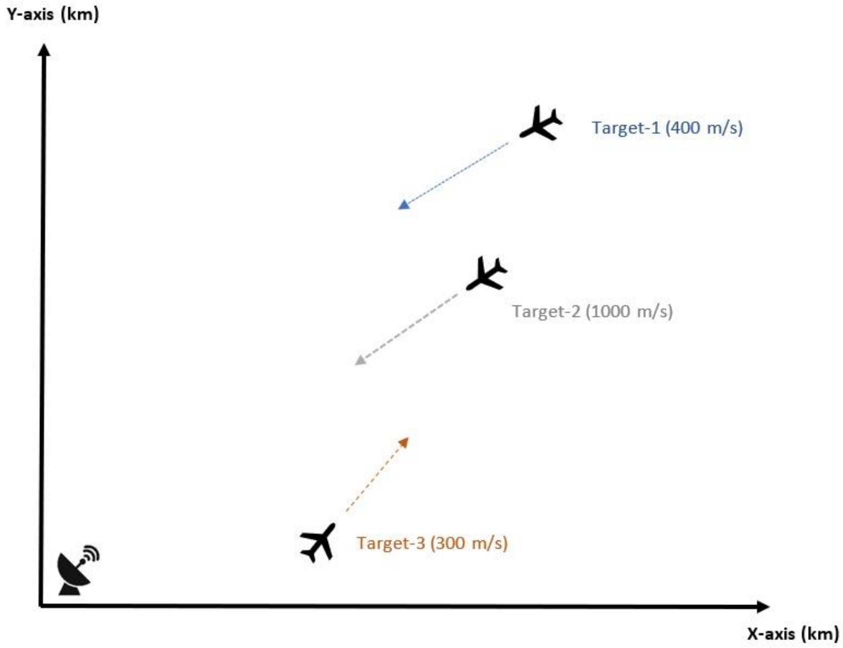

Figure 4.2 Different Target Velocities Scenario

Table 4.6 *Different Target Velocities Scenario*

	t=1	*t=2*	*t=3*	*t=4*	*t=5*	*t=6*	*t=7*	*t=8*	*t=9*
Decisions (Target 1)	*1*	*1*	*1*	*1*	*0*	*1*	*1*	*1*	*1*
Decisions (Target 2)	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
Decisions (Target 3)	*0*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*0*
λ Coeff	*1.09*	*0.70*	*0.56*	*0.56*	*0.56*	*6.55*	*6.55*	*6.55*	*6.55*
Trace (P1)	*10.50*	*10.50*	*10.50*	*10.50*	*30.00*	*11.56*	*10.96*	*10.68*	*10.54*
Trace (P2)	*10.50*	*10.50*	*10.50*	*10.50*	*10.50*	*10.50*	*10.50*	*10.50*	*10.50*
Trace (P3)	*30.00*	*73.50*	*153.0*	*280.5*	*15.29*	*35.12*	*85.68*	*178.9*	*327.0*
Time Spent (s)	*0.026*	*0.026*	*0.026*	*0.026*	*0.018*	*0.026*	*0.026*	*0.026*	*0.026*
Time Occupancy (%)	**100**	**100**	**100**	**100**	**70**	**100**	**100**	**100**	**100**

4.3.2.3. Track Mixing Cost Function

In order to observe how the model behaves in case of track mixing, two targets were moved in trajectories that are close to each other.

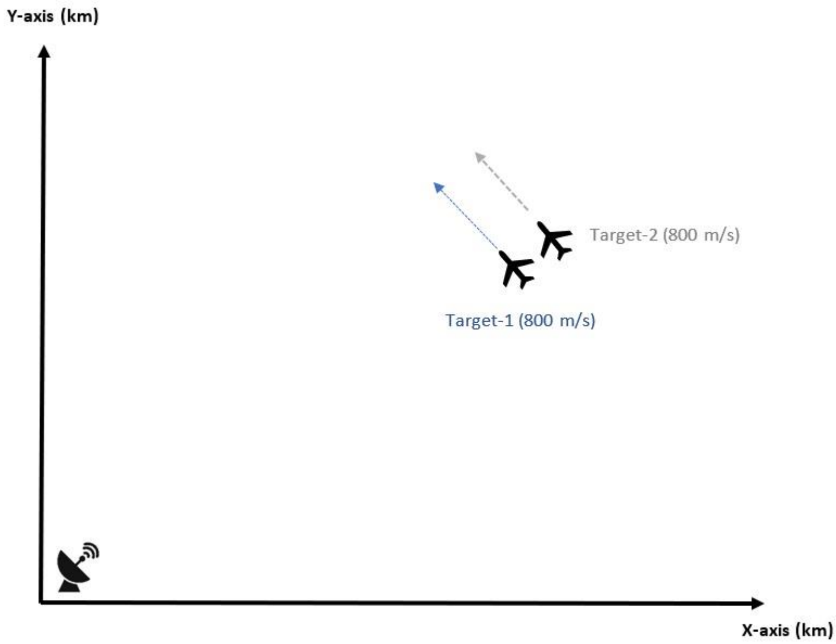

Figure 4.3 Track Mixing Cost Function Scenario

Table 4.7 *Track Mixing Cost Function Scenario*

	t=1	***t=2***	***t=3***	***t=4***	***t=5***	***t=6***	***t=7***	***t=8***	***t=9***
Decisions (Target 1)	*1*	*0*	*1*	*0*	*1*	*0*	*1*	*1*	*1*
Decisions (Target 2)	*0*	*1*	*0*	*1*	*0*	*0*	*1*	*1*	*1*
Trace (P1)	*10.50*	*30.00*	*11.56*	*30.48*	*11.78*	*30.89*	*11.79*	*11.04*	*10.69*
Trace (P2)	*30.00*	*11.56*	*30.48*	*11.78*	*30.89*	*74.74*	*12.77*	*11.53*	*10.82*
Time Spent (s)	*0.013*	*0.013*	*0.013*	*0.013*	*0.013*	*0*	*0.026*	*0.026*	*0.026*
Time Occupancy (%)	**50**	**50**	**50**	**50**	**50**	**0**	**100**	**100**	**100**

The cost function has been added to the objective function of target 1. As a result, the model tends to make update decision more for this target to increase the target performance.

4.3.3. Effect of Tracking Filter on the Model

Our optimization model uses the inverse of the error covariance matrix trace for utility function in each decision time step. To make a difference on this case, we have added PDA filter to our model which is based on Kalman filter and we observe its contribution. At first, we thought we would minimize the error using PDA filter, but it is seen in the trials that the decision process is not affected as expected. Scheduling decisions does not change in both situations because Markov chain step does not reflect the effect of clutter. If the resolution of the performance state increases, the effect of high clutter situation can be seen on optimization model.

Scenario: 15 Target (4 Tracked)

Initial Lagrange: 1000

Time Limit for 4 Target: 40 msec; in unrestricted mode time is reaching 60 msec

Time Forward: 4 sec

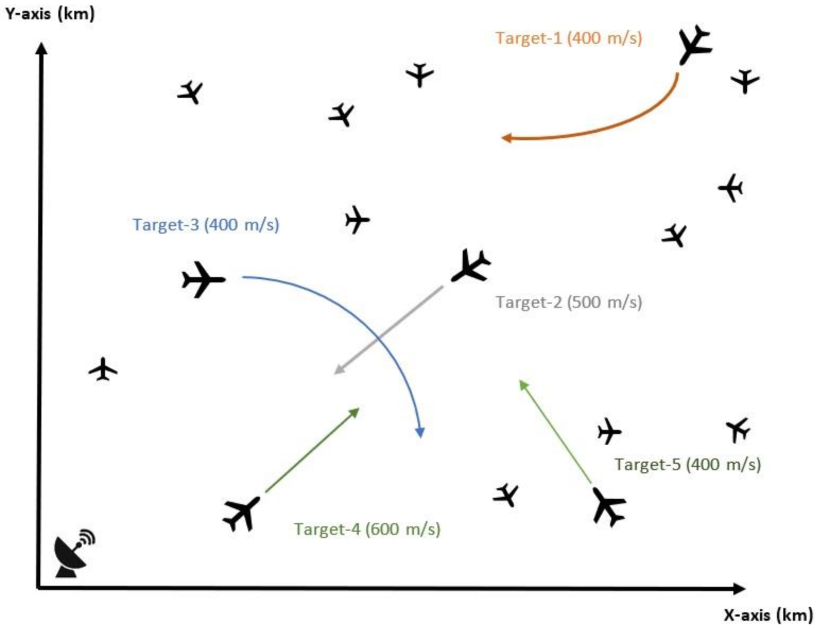

Figure 4.4 Filter Effect Scenario

Table 4.8 *Scenario with Kalman Filter*

	t=1	***t=2***	***t=3***	***t=4***
Decisions (Target 1)	*0*	*1*	*0*	*1*
Decisions (Target 2)	*1*	*0*	*1*	*0*
Decisions (Target 3)	*0*	*1*	*0*	*0*
Decisions (Target 4)	*1*	*0*	*1*	*0*
Trace (P1)	*30.00*	*11.56*	*30.48*	*11.78*
Trace (P2)	*10.50*	*30.00*	*11.56*	*30.48*
Trace (P3)	*30.00*	*11.56*	*30.48*	*73.68*

Table 4.8 (cont'd)				
Trace (P4)	*10.50*	*30.00*	*11.56*	*30.48*
Time Spent (s)	*0.026*	*0.017*	*0.026*	*0.011*
Time Occupancy (%)	**100**	**65**	**100**	**42**

Table 4.9 *Scenario with PDAF and Low Clutter*

	t=1	***t=2***	***t=3***	***t=4***
Decisions (Target 1)	*0*	*1*	*0*	*1*
Decisions (Target 2)	*1*	*0*	*1*	*0*
Decisions (Target 3)	*0*	*1*	*0*	*0*
Decisions (Target 4)	*1*	*0*	*1*	*0*
Trace (P1)	*30.00*	*15.68*	*38.63*	*32.46*
Trace (P2)	*11.73*	*32.75*	*81.68*	*165.68*
Trace (P3)	*30.00*	*18.41*	*44.02*	*96.54*
Trace (P4)	*19.67*	*43.70*	*25.56*	*55.16*
Time Spent (s)	*0.026*	*0.017*	*0.026*	*0.011*
Time Occupancy (%)	**100**	**65**	**100**	**42**

Table 4.10 *Scenario with PDAF and High Clutter*

	t=1	***t=2***	***t=3***	***t=4***
Decisions (Target 1)	*0*	*1*	*0*	*1*
Decisions (Target 2)	*1*	*0*	*1*	*0*
Decisions (Target 3)	*0*	*1*	*0*	*0*
Decisions (Target 4)	*1*	*0*	*1*	*0*
Trace (P1)	*379.41*	*42.26*	*87.76*	*83.48*
Trace (P2)	*26.16*	*64.93*	*88.04*	*168.48*
Trace (P3)	*30.00*	*20.98*	*49.10*	*105.12*
Trace (P4)	*16.56*	*41.33*	*46.57*	*99.67*
Time Spent (s)	*0.026*	*0.017*	*0.026*	*0.011*
Time Occupancy (%)	**100**	**65**	**100**	**42**

As can be seen from Table 4.8, Table 4.9 and Table 4.10, trace of P is higher for PDA Filter than Kalman under same circumstances. The reason is that, when PDA Filter is applied, probability β_0 used to update Error Covariance Matrix extends the validation gate range in high clutter returns. For PDA Filter, probability β_0 is evolved according to the difference between the current measurement error and the previous one. If the returned signal remains in high clutter density or is originated from false alarm, β_0 increases the Kalman error covariance matrix. In other words, at each update iteration, error covariance matrix is updated according to the error. However, for Kalman Filter, error covariance matrix always gets the value according to the system model. Hence, updated β also updates validation gate range. This update process reduces the risk of target drop at the track correlation step. Unlike Kalman Filter, in PDA Filter gate level does not change linearly with respect to system dynamics.

4.3.4. Comparison of TWS, Adaptive and Lagrangian-based Tracking Model

Firstly, we mention about the working principle of these methods. In TWS and adaptive cases, performance of the tracking is only considered at the measurement time, so when the allocated time resource is not enough, tracking algorithm must select targets which have the lowest priorities to be omitted in the measurement batch interval. Hence, overall performance analysis of the tracking operation is performed in the fast time scheduling which is also called as myopic case. So, optimization of tracking policy is made in this interval which is different from Lagrangian case. To see the differences of these methods and make a comparison, we performed adaptive, TWS and Lagrangian tracking simulations in a specific scenario.

In the first case, TWS allocates the specific time interval in the radar resources while scanning the horizon. If the allocated time is not enough to track all the targets, the radar resource management must negotiate in tracking targets to drop a couple of them. As can be seen in Table 4.11, two targets with lowest SNR are dropped because of limited resources and error values for these targets are getting high in each measurement time. In addition, at slow time 4, Target 1 is detected and tracked again because timing resources become enough to track one more target in measurement batch time.

In adaptive case, tracking decision is based on uncertainty threshold value. If the tracking accuracy of any target falls below the threshold value, tracking decision is made for these targets, the others does not need to be updated. The tracking diagram of adaptive tracking can be seen in Figure 4.5. If we look at the graph of the adaptive tracking for future time zones, adaptive tracking behaves as if it uses greedy algorithm to track the targets in slow time. The adaptive tracking is useful when limited radar resource is not enough to track desired number of targets. But it does not guarantee that the time constraints are satisfied in resource allocation strategy. Adaptive tracking results can be seen in Table 4.12. Trace error of the targets does not exceed the error threshold until measurement time 4. At measurement time 4, however, tracking decisions are made for all targets and total batch time is reached to 58.5 msec which is too high for timing constraint.

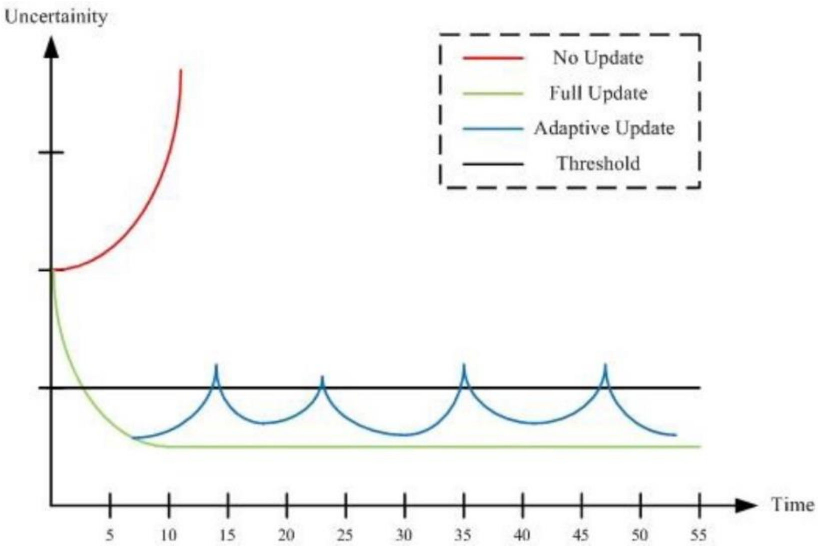

Figure 4.5 An Example of an Adaptive Update Strategy [27]

On the other hand, in Lagrangian case, tracking algorithm calculates its performance value by looking at the utilities in future time zones and it tries to maximize their profit. The strategy is useful when the resources are limited, and target density is high. These results can be seen in Table 4.13. Trace error for each target were kept at a certain safety level while satisfying the constraints. But when the optimization time interval is kept too long, Lagrangian-based tracking will not work properly because the predicted performance accuracy will decrease dramatically as the time interval increases.

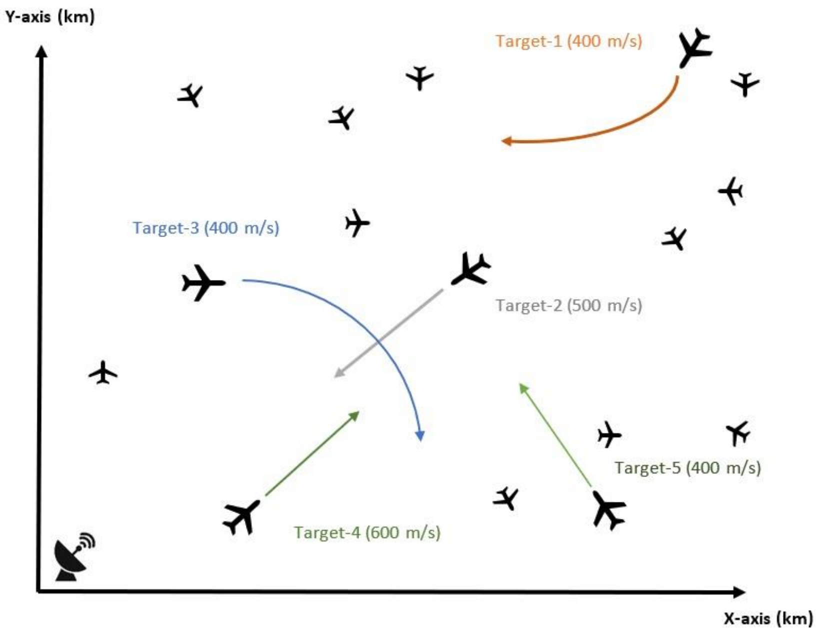

Figure 4.6 Comparison of Tracking Methods Scenario

Table 4.11 *Online Trace Error Values for TWS Tracking*

	Batch Time	**Trace of Kalman Error Covariance Matrix**				
Measurement #	All Targets	Target 1	Target 2	Target 3	Target 4	Target 5
1	0.0306	42650.00	42650.00	10.53	10.53	10.53
2	0.0306	62777.50	62777.50	10.50	10.50	10.50
3	0.0306	86965.00	86965.00	10.50	10.50	10.50
4	0.0306	115224.50	115224.50	10.50	10.50	10.50
5	0.0306	147568.00	147568.00	10.50	10.50	10.50
6	0.0306	184007.50	184007.50	10.50	10.50	10.50
Slow Time 1						
1	0.0316	224555.00	224555.00	10.50	10.50	10.50
2	0.0308	269222.50	269222.50	10.50	10.50	10.50
3	0.0310	318022.00	318022.00	10.50	10.50	10.50
4	0.0312	370965.50	370965.50	10.50	10.50	10.50
5	0.0314	428065.00	428065.00	10.50	10.50	10.50
Slow Time 2						
1	0.0322	489332.50	489332.50	10.50	10.50	10.50
2	0.0317	554780.00	554780.00	10.50	10.50	10.50
3	0.0318	624419.50	624419.50	10.50	10.50	10.50
4	0.0320	698263.00	698263.00	10.50	10.50	10.50

Table 4.11 (cont'd)						
5	0.0321	776322.50	776322.50	10.50	10.50	10.50
Slow Time 3						
1	0.0327	858610.00	858610.00	10.50	10.50	10.50
2	0.0323	945137.50	945137.50	10.50	10.50	10.50
3	0.0324	1035917.00	1035917.00	10.50	10.50	10.50
4	0.0326	1130961.00	1130961.00	10.50	10.50	10.50
5	0.0326	1230280.00	1230280.00	10.50	10.50	10.50
Slow Time 4						
1	0.0318	1333888.00	1333888.00	10.50	10.50	10.50
2	0.0328	1441795.00	1441795.00	10.50	10.50	10.50
3	0.0329	1554015.00	1554015.00	10.50	10.50	10.50
4	0.0330	62.59	1670558.00	10.50	30.00	10.50
5	0.0317	17.36	1791438.00	10.50	73.50	10.50
Slow Time 5						
1	0.0321	11.47	1916665.00	10.50	153.00	10.50
2	0.0318	10.64	2046253.00	10.50	280.50	10.50
3	0.0319	10.54	2180212.00	10.50	468.00	10.50
4	0.0320	10.52	2318556.00	10.50	727.50	10.50
5	0.0320	10.50	2461295.00	10.50	1071.00	10.50
Slow Time 6						
1	0.0323	10.50	2608443.00	10.50	1510.50	10.50
2	0.0321	10.50	2760010.00	10.50	2058.00	10.50
3	0.0322	10.50	2916010.00	10.50	2725.50	10.50
4	0.0322	10.50	3076453.00	10.50	3525.00	10.50
5	0.0323	10.50	3241353.00	10.50	4468.50	10.50
Slow Time 7						
1	0.0326	10.50	3410720.00	10.50	5568.00	10.50
2	0.0324	10.50	3584568.00	10.50	6835.50	10.50
3	0.0324	10.50	3762907.00	10.50	8283.00	10.50
4	0.0325	10.50	3945751.00	10.50	9922.50	10.50
5	0.0325	10.50	4133110.00	10.50	11766.00	10.50
Slow Time 8						
1	0.0328	10.50	4324998.00	10.50	13825.50	10.50
2	0.0326	10.50	4521425.00	10.50	16113.00	10.50
3	0.0326	10.50	4722405.00	10.50	18640.50	10.50
4	0.0327	10.50	4927948.00	10.50	21420.00	10.50
5	0.0327	10.50	5138068.00	10.50	24463.50	10.50

Table 4.11 (cont'd)						
Slow Time 9						
1	0.0329	10.50	5352775.00	10.50	27783.00	10.50
2	0.0328	10.50	5572083.00	10.50	31390.50	10.50
3	0.0328	10.50	5796002.00	10.50	35298.00	10.50
4	0.0329	10.50	6024546.00	10.50	39517.50	10.50
5	0.0329	10.50	6257725.00	10.50	44061.00	10.50
Slow Time 10						
1	0.0331	10.50	6495553.00	10.50	48940.50	10.50
2	0.0330	10.50	6738040.00	10.50	54168.00	10.50
3	0.0330	10.50	6985200.00	10.50	59755.50	10.50
4	0.0330	10.50	7237043.00	10.50	65715.00	10.50
5	0.0330	10.50	7493583.00	10.50	72058.50	10.50

Table 4.12 *Online Trace Error Values for Adaptive Tracking*

	Batch Time	**Trace of Kalman Error Covariance Matrix**				
Measurement #	All Targets	Target 1	Target 2	Target 3	Target 4	Target 5
1	0	153.80	153.80	153.80	153.80	153.80
2	0	278.62	278.62	278.62	278.62	278.62
3	0	462.11	462.11	462.11	462.11	462.11
4	0	16.83	16.83	16.83	16.83	16.83
5	0.0558	37.75	37.75	37.75	37.75	37.75
6	0	92.44	92.44	92.44	92.44	92.44
Slow Time 1						
1	0	192.89	192.89	192.89	192.89	192.89
2	0	351.11	351.11	351.11	351.11	351.11
3	0	15.44	15.44	15.44	15.44	15.44
4	0.0568	35.32	35.32	35.32	35.32	35.32
5	0	86.20	86.20	86.20	86.20	86.20
Slow Time 2						
1	0	180.09	180.09	180.09	180.09	180.09
2	0	328.98	328.98	328.98	328.98	328.98
3	0	15.39	15.39	15.39	15.39	15.39
4	0.0574	35.26	35.26	35.26	35.26	35.26
5	0	86.04	86.04	86.04	86.04	86.04
Slow Time 3						
1	0	179.74	179.74	179.74	179.74	179.74

Table 4.12 (cont'd)						
2	0	328.36	328.36	328.36	328.36	328.36
3	0	15.39	15.39	15.39	15.39	15.39
4	0.0578	35.25	35.25	35.25	35.25	35.25
5	0	86.04	86.04	86.04	86.04	86.04
Slow Time 4						
1	0	179.74	179.74	179.74	179.74	179.74
2	0	328.35	328.35	328.35	328.35	328.35
3	0	15.39	15.39	15.39	15.39	15.39
4	0.0581	35.25	35.25	35.25	35.25	35.25
5	0	86.04	86.04	86.04	86.04	86.04
Slow Time 5						
1	0	179.74	179.74	179.74	179.74	179.74
2	0	328.35	328.35	328.35	328.35	328.35
3	0	15.39	15.39	15.39	15.39	15.39
4	0.0583	35.25	35.25	35.25	35.25	35.25
5	0	86.04	86.04	86.04	86.04	86.04
Slow Time 6						
1	0	179.74	179.74	179.74	179.74	179.74
2	0	328.35	328.35	328.35	328.35	328.35
3	0	15.39	15.39	15.39	15.39	15.39
4	0.0585	35.25	35.25	35.25	35.25	35.25
5	0	86.04	86.04	86.04	86.04	86.04
Slow Time 7						
1	0	179.74	179.74	179.74	179.74	179.74
2	0	328.35	328.35	328.35	328.35	328.35
3	0	15.39	15.39	15.39	15.39	15.39
4	0.0586	35.25	35.25	35.25	35.25	35.25
5	0	86.04	86.04	86.04	86.04	86.04
Slow Time 8						
1	0	179.74	179.74	179.74	179.74	179.74
2	0	328.35	328.35	328.35	328.35	328.35
3	0	15.39	15.39	15.39	15.39	15.39
4	0.0586	35.25	35.25	35.25	35.25	35.25
5	0	86.04	86.04	86.04	86.04	86.04
Slow Time 9						
1	0	179.74	179.74	179.74	179.74	179.74
2	0	328.35	328.35	328.35	328.35	328.35

Table 4.12 (cont'd)						
3	0	15.39	15.39	15.39	15.39	15.39
4	0.0586	35.25	35.25	35.25	35.25	35.25
5	0	86.04	86.04	86.04	86.04	86.04
Slow Time 10						
1	0	179.74	179.74	179.74	179.74	179.74
2	0	328.35	328.35	328.35	328.35	328.35
3	0	15.39	15.39	15.39	15.39	15.39
4	0.0586	35.25	35.25	35.25	35.25	35.25
5	0	86.04	86.04	86.04	86.04	86.04

Table 4.13 *Online Trace Error Values for Lagrangian-Based Tracking*

	Batch Time	**Trace of Kalman Error Covariance Matrix**				
Measurement #	All Targets	Target 1	Target 2	Target 3	Target 4	Target 5
1	0.0551	10.53	10.53	10.53	10.53	10.53
2	0.0554	10.50	10.50	10.50	10.50	10.50
3	0.0557	10.50	10.50	10.50	10.50	10.50
4	0.0559	10.50	10.50	10.50	10.50	10.50
5	0.0561	10.50	10.50	10.50	10.50	10.50
6	0.0561	10.50	10.50	10.50	10.50	10.50
Slow Time 1						
1	0.027	30.00	10.50	30.00	10.50	30.00
2	0.0296	11.56	30.00	11.56	30.00	11.56
3	0.0269	30.48	11.56	30.48	11.56	30.48
4	0.0245	11.78	30.48	73.68	30.48	11.78
5	0.0268	30.89	11.78	153.18	11.78	30.89
Slow Time 2						
1	0.0268	74.74	11.04	280.98	11.04	74.74
2	0.0304	12.77	30.91	15.31	30.91	12.77
3	0.0268	31.57	11.60	35.16	11.60	31.57
4	0.0243	11.92	30.56	85.80	30.56	11.92
5	0.0267	31.25	11.78	179.22	11.78	31.25
Slow Time 3						
1	0.0267	75.56	11.04	327.42	11.04	75.56
2	0	156.85	30.91	542.41	30.91	156.85
3	0.0311	13.99	75.79	16.83	75.79	13.99
4	0.0266	33.13	12.68	37.71	12.68	33.13

Table 4.13 (cont'd)						
5	0	80.52	31.38	92.34	31.38	80.52
Slow Time 4						
1	0.0265	168.18	11.92	192.73	11.92	168.18
2	0	308.10	31.24	350.88	31.24	308.10
3	0.0316	15.35	75.55	15.43	75.55	15.35
4	0.0265	35.20	12.78	35.32	12.78	35.20
5	0	85.91	31.58	86.20	31.58	85.91
Slow Time 5						
1	0.0264	179.46	11.92	180.08	11.92	179.46
2	0	327.85	31.25	328.96	31.25	327.85
3	0.0319	15.39	75.57	15.39	75.57	15.39
4	0.0263	35.25	12.78	35.26	12.78	35.25
5	0	86.04	31.58	86.04	31.58	86.04
Slow Time 6						
1	0.0262	179.73	11.92	179.74	11.92	179.73
2	0.0321	14.02	31.25	14.02	31.25	14.02
3	0.0262	33.14	11.80	33.14	11.80	33.14
4	0	80.53	30.93	80.53	30.93	80.53
5	0.0238	12.81	74.79	168.19	74.79	12.81
Slow Time 7						
1	0.0261	31.68	12.77	308.13	12.77	31.68
2	0.0324	11.93	31.58	15.35	31.58	11.93
3	0.0260	31.25	11.92	35.21	11.92	31.25
4	0.0237	11.80	31.25	85.91	31.25	11.80
5	0	30.93	75.56	179.46	75.56	30.93
Slow Time 8						
1	0.0259	74.80	12.78	327.85	12.78	74.80
2	0.0326	12.77	31.58	15.39	31.58	12.77
3	0.0259	31.58	11.92	35.25	11.92	31.58
4	0.0236	11.92	31.25	86.04	31.25	11.92
5	0	31.25	75.57	179.73	75.57	31.25
Slow Time 9						
1	0.0258	75.56	12.78	328.35	12.78	75.56
2	0.0328	12.78	31.58	15.39	31.58	12.78
3	0.0257	31.58	11.92	35.25	11.92	31.58
4	0.0235	11.92	31.25	86.04	31.25	11.92
5	0	31.25	75.57	179.74	75.57	31.25

Table 4.13 (cont'd)						
Slow Time 10						
1	0.0256	75.57	12.78	328.35	12.78	75.57
2	0	156.86	31.58	543.89	31.58	156.86
3	0	287.14	76.45	838.33	76.45	287.14
4	0.0330	15.33	159.40	18.32	159.40	15.33
5	0.0255	35.20	13.99	40.35	13.99	35.20

CHAPTER 5

CONCLUSION

5.1. Conclusion and Future Works

In this thesis, the online resource allocation problem in MF Radar systems is presented. A detailed research of a proposed resource management approach called Lagrangian-based optimization is improved. A simulation model has been constructed with the aim of the online operation feasibility. Besides the Kalman filter, implementation of Probabilistic Data Association filter is also performed. Also, track mixing and track re-initiation events are included in the optimization model. In addition to these, TWS, adaptive tracking and Lagrange based algorithms are compared. TWS algorithm allocates a certain time interval for the whole tracking process, and this time interval may not satisfy needs for all targets. In this case, some of the targets with low priority are dropped. In adaptive case, targets are tracked or not tracked according to a certain threshold value. In Lagrange-based optimization, a certain level of tracking quality is guaranteed. The main difficulty about Lagrange-based optimization is the possibility of performance variables to change dramatically in slow time scheduling interval. If this interval, namely resource allocation interval, gets higher, estimation for high-velocity targets becomes unrealistic.

Further research could be done by improving target kinematics model. Besides, searching tasks can be added to the optimization problem. Another future work direction might be including algorithms to select the best Lagrange coefficients in Lagrangian Relaxation. Moreover, Joint Probabilistic Data Association Filter or Multi Hypothesis Tracking Filter can be added in tracking scenario and effect of these filter can be observed. Improving duality gap in the optimization is another valuable future research area.

REFERENCES

[1] J. Wintenby and V. Krishnamurthy, "Hierarchical resource management in adaptive airborne surveillance radars," in IEEE Transactions on Aerospace and Electronic Systems, vol. 42, no. 2, pp. 401-420, April 2006.

[2] Zhen Ding, "A survey of radar resource management algorithms," 2008 Canadian Conference on Electrical and Computer Engineering, Niagara Falls, ON, 2008, pp. 001559-001564.

[3] W. Komorniczak, T. Kuczerski and J. F. Pietrasinski, "The priority assignment for detected targets in multi-function radar," 13th International Conference on Microwaves, Radar and Wireless Communications. MIKON - 2000. Conference Proceedings (IEEE Cat. No.00EX428), Wroclaw, Poland, 2000, pp. 244-247 vol.1.

[4] A. Izquierdo-Fuente and J. R. Casar-Corredera, "Optimal radar pulse scheduling using a neural network," Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), Orlando, FL, USA, 1994, pp. 4588-4591 vol.7.

[5] S. Miranda, C. Baker, K. Woodbridge and H. Griffiths, "Knowledge-based resource management for multifunction radar: a look at scheduling and task prioritization," in IEEE Signal Processing Magazine, vol. 23, no. 1, pp. 66-76, Jan. 2006.

[6] R. Rajkumar, C. Lee, J. Lehoczky and D. Siewiorek, "A resource allocation model for QoS management," Proceedings Real-Time Systems Symposium, San Francisco, CA, USA, 1997, pp. 298-307.

[7] S. Ghosh, J. Hansen, R. Rajkumar, and J. Lehoczky. Adaptive qos optimizations with applications to radar tracking, Technical Report 18-03-04, Institute for Complex Engineering Systems, Carnegie Mellon University, 2004.

[8] S. Ghosh, R. Rajkumar, J. Hansen, and J. Lehoczky, “Integrated QoS-aware resource management and scheduling with multi-resource constraints” in Real-Time Syst., vol. 33, pp. 7-46, July 2006.

[9] Bellman, R., Dynamic Programming, Princeton, NJ: Princeton University Press, 1957.

[10] F. Daum and R. Fitzgerald, "Decoupled Kalman filters for phased array radar tracking," in IEEE Transactions on Automatic Control, vol. 28, no. 3, pp. 269-283, March 1983.

[11] D. J. Kershaw and R. J. Evans, "Waveform selective probabilistic data association," in IEEE Transactions on Aerospace and Electronic Systems, vol. 33, no. 4, pp. 1180-1188, Oct. 1997.

[12] B. F. La Scala, W. Moran and R. J. Evans, "Optimal adaptive waveform selection for target detection," 2003 Proceedings of the International Conference on Radar (IEEE Cat. No.03EX695), Adelaide, SA, Australia, 2003, pp. 492-496.

[13] Haykin, S., Currie, B. and Kirubarajan, T., Literature search on adaptive radar transmit waveforms (DRDC Ottawa CR-2003-094), McMaster University, Hamilton, Ontario, 2003.

[14] G. van Keuk and S. S. Blackman, "On phased-array radar tracking and parameter control," in IEEE Transactions on Aerospace and Electronic Systems, vol. 29, no. 1, pp. 186-194, Jan. 1993.

[15] Scala, Barbara & Moran, Bill. (2006). Optimal target tracking with restless bandits. Digital Signal Processing. 16. 479-487.

[16] J. M. Butler, A. R. Moore and H. D. Griffiths, "Resource Management For A Rotating Multi-function Radar," Radar 97 (Conf. Publ. No. 449), Edinburgh, UK, 1997, pp. 568-572.

[17] National Severe Storms Library, MULTI-FUNCTION PHASED ARRAY RADAR (MPAR), http://www.nssl.noaa.gov/tools/radar/mpar/

[18] Nickel, U., "Fundamentals of signal processing for phased array radar," in Advanced Radar Signal and Data Processing, Educational Notes RTO-EN-SET-086, Paper 1, pp.1-1ł1-22. Neuilly-sur-Seine, France: NATO Research and Technology Organisation (RTO), 2006.

[19] P. Swerling, "Probability of detection for fluctuating targets," in IRE Transactions on Information Theory, vol. 6, no. 2, pp. 269-308, April 1960.

[20] Mahafza, B.R.. (2016). Radar systems analysis and design using MATLAB: Third edition.

[21] Skolnik, M.. (1980). Introduction to radar systems /2nd edition/. New York, McGraw Hill Book Co., 1980. 590 p.. -1.

[22] S. Kingsley and S. Quegan, Understanding Radar Systems, McGraw-Hill, 1992.

[23] R. Curry, Radar Systems Performance Modelling, ArtechHouse, 2001

[24] Y. Bar-Shalom, F. Daum and J. Huang, "The probabilistic data association filter," in IEEE Control Systems Magazine, vol. 29, no. 6, pp. 82-100, Dec. 2009.

[25] R. B. Washburn, M. K. Schneider and J. J. Fox, "Stochastic dynamic programming-based approaches to sensor resource management," Proceedings

of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997), Annapolis, MD, USA, 2002, pp. 608-615 vol.1.

[26] V. Krishnamurthy and R. J. Evans, "Hidden Markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking," in IEEE Transactions on Signal Processing, vol. 49, no. 12, pp. 2893-2908, Dec. 2001.

[27] Ç. Uzun and M. Demirekler, "Stochastic dynamic programming based resource allocation for multi target tracking for electronically steered antenna radar," 2015 23nd Signal Processing and Communications Applications Conference (SIU), Malatya, 2015, pp. 867-870.

[28] R. K. Ahuja, T. L. Magnanti and J. B. Orlin. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993