CFAR PROCESSING WITH MULTIPLE EXPONENTIAL SMOOTHERS FOR NONHOMOGENEOUS ENVIRONMENTS

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

BY

BERK GÜRAKAN

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

DECEMBER 2010

Approval of the thesis:

CFAR PROCESSING WITH MULTIPLE EXPONENTIAL SMOOTHERS FOR NONHOMOGENEOUS ENVIRONMENTS

submitted by **BERK GÜRAKAN** 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. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. Tolga Çiloğlu
Supervisor, **Electrical and Electronics Engineering Dept., METU** _____

Assoc. Prof. Dr. Çağatay Candan
Co-supervisor, **Electrical and Electronics Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Mübeccel Demirekler
Electrical and Electronics Engineering Dept., METU _____

Assoc. Prof. Dr. Tolga Çiloğlu
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Mustafa Kuzuoğlu
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Aydın Alatan
Electrical and Electronics Engineering Dept., METU _____

Serkan Ceylan (M.Sc.)
ASELSAN _____

Date: _____

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, Last Name: BERK GÜRAKAN

Signature :

ABSTRACT

CFAR PROCESSING WITH MULTIPLE EXPONENTIAL SMOOTHERS FOR NONHOMOGENEOUS ENVIRONMENTS

Gürakan, Berk

M.Sc., Department of Electrical and Electronics Engineering

Supervisor : Assoc. Prof. Dr. Tolga Çiloğlu

Co-Supervisor : Assoc. Prof. Dr. Çağatay Candan

December 2010, 84 pages

Conventional methods of CFAR detection always use windowing, in the sense that some number of cells are investigated and the target present/absent decision is made according to the composition of the cells in that window. The most commonly used versions of CFAR detection algorithms are cell averaging CFAR, smallest of cell averaging CFAR, greatest of cell averaging CFAR and order-statistics CFAR. These methods all use windowing to set the decision threshold. In this thesis, rather than using windowed CFAR algorithms, a new method of estimating the background threshold is presented, analyzed and simulated. This new method is called the Switching IIR CFAR algorithm and uses two IIR filters to accurately estimate the background threshold. Then, using a comparison procedure, one of the filters is selected as the current threshold estimate and used. The results are seen to be satisfactory and comparable to conventional CFAR methods. The basic advantages of using the SIIR CFAR method are computational simplicity, small memory requirement and acceptable performance under clutter edges and multiple targets.

Keywords: CFAR, IIR Filter, Radar, Detection, Switching

ÖZ

RADAR HEDEF TESPİTİNDE FİLTRELEME YÖNTEMLERİNİN KULLANIMI

Gürakan, Berk
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Tolga Çiloğlu
Ortak Tez Yöneticisi : Doç. Dr. Çagatay Candan
Aralık 2010, 84 sayfa

Sabit yanlış alarm olasılığı (SYAO) yöntemleri genelde birden çok hücreye bakıp, bu hücrelerin belirli istatistiksel özelliklerine göre (ortalama, minimum, maksimum vs) tespit eşiğini belirlemektedirler. Bu da, hem hafıza gereksinimi açısından hem de işlem yükü açısından belirli problemleri beraberinde getirmektedir. Bu çalışmada, iki tane sonsuz dürtü cevaplı filtre kullanılarak tespit eşiği kestirilmiştir. Belirli bir karşılaştırma işleminden sonra hangi filtrenin seçileceğine karar veren bir algoritma geliştirilmiştir. Bu filtreler yalnızca tek bir hücreye bakarak çalıştıkları için işlem yükü ve hafıza gereksinimi açısından bilinen SYAO yöntemlerinden daha iyilerdir. Bu algoritmanın simülasyon sonuçlarının bilinen SYAO yöntemlerine yakın olduğu görülmüştür.

Anahtar Kelimeler: Radar, Filtre, Sonsuz Dürtü Cevabı

To My Family

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Tolga Çiloğlu for his guidance, advice, criticism, encouragements and insight throughout the research.

I would also like to thank my co advisor Assoc. Prof. Dr. Çağatay Candan for his valuable remarks and thoughtful comments. He has made his support available in a number of ways.

Thanks to the examining committee members Prof. Dr. Mübeccel Demirekler, Prof. Dr. Mustafa Kuzuoğlu, Prof. Dr. Aydın Alatan and Serkan Ceylan for evaluating my work and for their valuable comments.

I would like to express my gratitude to Oktay Sipahigil for his excellent suggestions and support.

Deepest thanks to my family for their love, trust and understanding.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Classical hypothesis testing aims to decide between the null hypothesis and the alternative hypothesis. These hypotheses can be a statistical induction statement (such as ”all cars are red”) or in the context of a radar, a target present or absent decision. There are many different ways to develop statistical tests to decide which one of these hypotheses are correct. One of the first developments in hypothesis testing was initiated by Bayes [1] and Neyman-Pearson [2] have contributed significantly to the field later on.

If the hypotheses are regarded in terms of signal present or absent decision, the problem can be called binary signal detection. It is called binary in the sense that there are two possible hypotheses, either a 1 or a 0. In this thesis, only binary signal detection will be considered even though M-ary signal detection (more than two hypotheses) is also cited frequently in literature.

1.1 Scope

The aim of this work is to present a new algorithm for constant false alarm rate (CFAR) detection of radar targets under Gaussian noise, based on an IIR filtering and switching approach. The performance of this detector will be analyzed by simulations and theoretical calculations.

Signals are modeled as exponential random variables (the Swerling model) and the noise is modeled as a white Gaussian process.

1.2 Outline

In 1.3, a review of detection theory is made which will be used in later parts of the thesis. In Chapter 2, a literature review on constant false alarm rate (CFAR) processing is made. Commonly used CFAR methods are explained with examples, equations and figures. In Chapter 3, CFAR detection using IIR filters is explained and the proposed algorithm together with the modifications and reasoning is presented. In Chapter 4, the simulation results and comments about the results are given. In Chapter 5, the thesis is concluded with finishing remarks, summary and an outline of future work.

1.3 Detection Fundamentals

The primary functions of any surveillance system (radar, sonar etc) are detection, tracking and imaging. The main concern here is to decide whether any measurement falls into one of the two categories:

1. The measurement is the result of interference (noise) only.

2. The measurement is the combined result of interference and echoes from a target.

The first hypothesis is denoted as the *null hypothesis*, H_0 and the second as *alternative hypothesis* H_1. The detector must use some kind of logic that determines whether a measurement falls under H_0 or H_1. If H_0 best accounts for the data, the system declares that no target was present for that measurement. If H_1 best accounts for the data, the system declares that a target was present for that certain measurement. Since the signals can only be described statistically, the decision between the two hypotheses is a matter of statistical decision theory. One should start the analysis using a probability density function (pdf) describing the measurement to be tested under each of the two hypotheses. Let the sample to be denoted as y, then the following definitions are valid:

$p_y(y|H_0)$ = the pdf of y given that a target was not present (H_0 is true)
$p_y(y|H_1)$ = the pdf of y given that a target was present (H_1 is true)

More generally, the detection will be based on not a single sample but N number of samples of data y_n forming a vector $\mathbf{y}$

$$\mathbf{y} \equiv [y_0 \dots y_{N-1}]$$

The N dimensional joint pdfs $p_y(\mathbf{y}|H_0)$ and $p_y(\mathbf{y}|H_1)$ are then used. The following probabilities are important and must be defined here:

Probability of Detection (P_D): The probability that a target *is* declared(H_1 is chosen) when a target *is* in fact present.
Probability of False Alarm (P_{FA}): The probability that a target *is* declared (H_1 is chosen) when a target is in fact *not* present.
Probability of Miss (P_M): The probability that a target is *not* declared (H_0 is chosen) when a target *is* in fact present.

Since $P_M = 1 - P_D$, P_{FA} and P_D are sufficient to specify all of the probabilities of interest.

1.3.1 The Neyman-Pearson Criterion

The next step in decision making is to find a rule for deciding what makes an optimal choice between the two hypotheses. The approaches follow directly from the theory of simple hypothesis testing problem. There are two main methods for dealing with this problem, the Neyman-Pearson criterion and the Bayesian risk approach. Sonar and radar systems typically use the Neyman-Pearson criterion [3]. The Neyman-Pearson criterion aims to find a decision process to maximize the probability of detection P_D under the constraint that the false alarm probability P_{FA} does not exceed a certain number. This criterion is motivated by the fact that for a fixed system design, increasing P_D implies increasing P_{FA} as well [4].

Each vector of measured data values $\mathbf{y}$ is a point in N dimensional space. To have a complete decision rule, every possible point in that space must be assigned to one of the hypotheses, H_0 (”target absent”) or H_1 (”target present”). Let R_1 denote the region for which if $\mathbf{y} \in R_1$ we denote target present (i.e, H_1). This means that the region R_1 is mapped to H_1. Then, the following definitions are valid:

$$\begin{aligned} P_D &= \int_{R_1} p_y(\mathbf{y}|H_1)\, d\mathbf{y} \\ P_{FA} &= \int_{R_1} p_y(\mathbf{y}|H_0)\, d\mathbf{y} \end{aligned} \tag{1.1}$$

Since probability density functions are nonnegative, it is seen from (1.1) that P_D and P_{FA} must rise or fall together when R_1 changes. Obviously when R_1 grows to include more of the possible observations $\mathbf{y}$, both of the integrals will be larger because of the nonnegative pdf. This leads to a very important result, in order to increase the detection probability, the false alarm probability must be allowed to increase as well.

1.3.2 The Likelihood Ratio Test

The Neyman-Pearson criterion aims to obtain the best possible detection performance while keeping a tolerable false alarm probability. This can be mathematically formulated as follows:

$$\text{choose } R_1 \text{ such that } P_D \text{ is maximized, subject to } P_{FA} \leq \alpha$$

where α is the maximum tolerable false alarm probability. One can use the method of Lagrange multipliers to solve this optimization problem. Let F be defined as

$$F = P_D + \lambda(P_{FA} - \alpha)$$

Now the problem is maximizing F and then choosing λ to satisfy $P_{FA} = \alpha$. Substituting from (1.1) we get:

$$\begin{aligned}\max F &= \max\{\int_{R_1} p_y(\mathbf{y}|H_1)\,d\mathbf{y} + \lambda(\int_{R_1} p_y(\mathbf{y}|H_0)\,d\mathbf{y} - \alpha)\} \\ &= \max\{-\lambda\alpha + \int_{R_1} p_y(\mathbf{y}|H_1) + \lambda p_y(\mathbf{y}|H_0)\,d\mathbf{y}\} \end{aligned} \tag{1.2}$$

The first term of (1.2) does not depend on the choice of R_1 therefore it can be omitted in the maximization process. To maximize F, one needs to maximize the value of the integral over R_1. So we should include $\mathbf{y}$ in R_1 if the integrand is positive for that value of $\mathbf{y}$. This means R_1 must be made of points $\mathbf{y}$ for which

$$p_{\mathbf{y}}(\mathbf{y}|H_1) + \lambda p_{\mathbf{y}}(\mathbf{y}|H_0) > 0$$

This directly leads to the decision rule

$$\frac{p_{\mathbf{y}}(\mathbf{y}|H_1)}{p_{\mathbf{y}}(\mathbf{y}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} -\lambda \tag{1.3}$$

(1.3) is known as the *likelihood ratio test* (LRT). This test is very useful in the sense that it gives an optimal rule for guessing, under the Neyman-Pearson criterion, whether a target is present or not based directly on the observed data $\mathbf{y}$ and the threshold $-\lambda$. Note that this threshold still needs to be computed. This threshold should be computed using the P_{FA} constraint. (1.3) tells us that the ratio of two pdfs, evaluated for the particular observation $\mathbf{y}$, must be computed and if that "likelihood ratio" exceeds some threshold, a target present decision must be made (choose H_1). If it does not exceed the threshold, declare no target is present (choose H_0). To carry out the LRT, it is obvious that the models of $p_y(\mathbf{y}|H_1)$ and $p_y(\mathbf{y}|H_0)$ are needed. As a convenient way, the LRT can be expressed in the following notation:

$$\Lambda(\mathbf{y}) \underset{H_0}{\overset{H_1}{\gtrless}} \eta \tag{1.4}$$

From (1.3), $\Lambda(y) = \frac{p_y(\mathbf{y}|H_1)}{p_y(\mathbf{y}|H_0)}$ and $\eta = -\lambda$. One can apply a monotone increasing operator to both sides of (1.4) and this will not change the performance. Most common way is to take natural logarithm to obtain the *log likelihood ratio test*.

$$\ln \Lambda(\mathbf{y}) \underset{H_0}{\overset{H_1}{\gtrless}} \ln \eta \tag{1.5}$$

To illustrate these cases better, consider a simple case where the presence or absence of a constant, in zero-mean Gaussian noise of variance β^2 is to be decided. This means distinguishing between two hypotheses:

$$H_0 : \mathbf{y} = \mathbf{w}$$
$$H_1 : \mathbf{y} = \mathbf{m} + \mathbf{w}$$

where w is a vector of independent identically distributed (i.i.d) zero mean Gaussian random variables and $\mathbf{m}$ is a vector of constants. When there is no signal (hypothesis H_0) the data vector $\mathbf{y}$ has an N-dimensional normal distribution with covariance matrix $\beta^2\mathbf{I}_N$ where $\mathbf{I}_N$ is the identity matrix of dimensions $N \times N$. When there is a signal (hypothesis H_1), the distribution is simply shifted to a nonzero mean.

$$H_0 : \mathbf{y} \sim N(\mathbf{0}_N, \beta^2\mathbf{I}_N)$$
$$H_1 : \mathbf{y} \sim N(\mathbf{m}_N, \beta^2\mathbf{I}_N) \tag{1.6}$$

The joint pdf of $\mathbf{y}$ under both hypothesis is therefore

$$p_{\mathbf{y}}(\mathbf{y}|H_0) = \prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\beta^2}} \exp\{-\frac{1}{2}(\frac{y_n}{\beta})^2\}$$
$$p_{\mathbf{y}}(\mathbf{y}|H_1) = \prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\beta^2}} \exp\{-\frac{1}{2}(\frac{y_n - m}{\beta})^2\} \tag{1.7}$$

The likelihood ratio $\Lambda(\mathbf{y})$ and the log likelihood ratio can be directly calculated.

$$\Lambda(\mathbf{y}) = \frac{p_y(\mathbf{y}|H_1)}{p_y(\mathbf{y}|H_0)} = \frac{\prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\beta^2}} \exp\{-\frac{1}{2}(\frac{y_n}{\beta})^2\}}{\prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\beta^2}} \exp\{-\frac{1}{2}(\frac{y_n-m}{\beta})^2\}} \tag{1.8}$$

$$\begin{aligned} \ln \Lambda(\mathbf{y}) &= \sum_{n=0}^{N-1} \{-\frac{1}{2}(\frac{y_n - m}{\beta})^2 + \frac{1}{2}(\frac{y_n}{\beta})^2\} \\ &= \frac{1}{\beta^2} \sum_{n=0}^{N-1} m y_n - \frac{1}{2\beta^2} \sum_{n=0}^{N-1} m^2 \end{aligned} \tag{1.9}$$

Using the log likelihood ratio and (1.5) we arrive at the following result.

$$\sum_{n=0}^{N-1} y_n \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\beta^2}{m} \ln(-\lambda) + \frac{Nm}{2} \tag{1.10}$$

It is worth noting that right hand side of (1.10) does not depend on y_n and can therefore be considered as a constant. It is specified from this equation that the available data samples, y_n, must be summed and then compared to a detection threshold, which still needs to be computed. This is an example of how LRT specifies the signal processing required on the data, to achieve optimal detection performance. In most cases, the LRT can be rearranged to isolate on one hand, only these terms including the data samples, moving all other constants to the other hand side. The term $\sum y_n$ is called the *sufficient statistic* for this example and denoted by $\Upsilon(\mathbf{y})$. The sufficient statistic is a function of the data $\mathbf{y}$ and has the property that the likelihood ratio can be written as a function of $\Upsilon(\mathbf{y})$. This means that no other statistic which can be calculated from the same sample, provides any additional information. This also means that knowing the sufficient statistic is as good as knowing the data itself [5]. Therefore, the LRT can now be changed to:

$$\Upsilon(\mathbf{y}) \underset{H_0}{\overset{H_1}{\gtrless}} T \tag{1.11}$$

Note that (1.11) is in the form of (1.10) with $\Upsilon(\mathbf{y}) = \sum y_n$ and $T = \frac{\beta^2}{m} \ln(-\lambda) + \frac{Nm}{2}$.

Since Λ and Υ are functions of the data vector $\mathbf{y}$, they have their own pdfs. This means we can compute the LRT threshold in terms of Λ and Υ to compute the detection threshold. One can use the following expressions:

$$P_{FA} = \int_{\eta=-\lambda}^{\infty} p_{\Lambda}(\Lambda|H_0)\, d\Lambda = \alpha$$
$$P_{FA} = \int_{T}^{\infty} p_{\Upsilon}(\Upsilon|H_0)\, d\Upsilon = \alpha \tag{1.12}$$

Continuing from the same example, we saw that the sufficient statistic $\Upsilon(\mathbf{y})$ is just the sum of the data samples y_n, $\Upsilon = \sum_{i=0}^{N-1} y_n$. Under H_0, $y_n \sim N(0, \beta^2)$ and therefore $\Upsilon \sim N(0, N\beta^2)$. Using the definitions in (1.12) we arrive at the following results.

$$\begin{aligned} P_{FA} &= \int_{T}^{\infty} p_{\Upsilon}(\Upsilon|H_0)\, d\Upsilon = \alpha \\ &= \int_{T}^{\infty} \frac{1}{\sqrt{2\pi N\beta^2}} e^{-\frac{\Upsilon^2}{2N\beta^2}}\, d\Upsilon \end{aligned} \tag{1.13}$$

Define the *error function*, $\mathrm{erf}(x)$ and *complementary error function*, $\mathrm{erfc}(x)$ as:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2}\, dt \tag{1.14}$$
$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_{x}^{\infty} e^{-t^2}\, dt \tag{1.15}$$

It can be seen that $\mathrm{erfc}(x) = 1 - \mathrm{erf}(x)$. One also needs to define the inverse error and inverse complementary error functions, and these are defined as $\mathrm{erf}^{-1}(x)$ and $\mathrm{erfc}^{-1}(x)$. It can be shown that $\mathrm{erfc}^{-1}(x) = \mathrm{erf}^{-1}(1 - x)$.

To evaluate the integral in (1.13), make the change of variables $t = \dfrac{\Upsilon}{\sqrt{2N\beta^2}}$, then $dt = \dfrac{d\Upsilon}{\sqrt{2N\beta^2}}$

$$\begin{aligned} P_{FA} &= \frac{1}{\sqrt{\pi}} \int_{T/\sqrt{2N\beta^2}}^{\infty} e^{-t^2}\, dt = \alpha \\ &= \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{T}{\sqrt{2N\beta^2}}\right)\right] \end{aligned} \tag{1.16}$$

Solving for T, we get:

$$T = \sqrt{2N\beta^2}\,\text{erf}^{-1}(1 - 2P_{FA}) \tag{1.17}$$

Equations (1.16) and (1.17) shows how to compute P_{FA} given T and T given P_{FA}. Now, we need to calculate P_D using the threshold computed from (1.17). Note that, under H_1, $y_n \sim N(m, \beta^2)$ therefore $\Upsilon \sim N(Nm, N\beta^2)$. Then,

$$\begin{aligned} P_D &= \int_T^\infty p_\Upsilon(\Upsilon|H_1)\, d\Upsilon \\ &= \int_T^\infty \frac{1}{\sqrt{2\pi N\beta^2}} e^{-\frac{(\Upsilon - Nm)^2}{2N\beta^2}}\, d\Upsilon \end{aligned} \tag{1.18}$$

Using a change of variables, $t = \dfrac{\Upsilon - Nm}{\sqrt{2N\beta^2}}$ one arrives at the following result:

$$P_D = \frac{1}{2}\left[1 - \text{erf}\left(\frac{T - Nm}{\sqrt{2N\beta^2}}\right)\right] \tag{1.19}$$

Using (1.17), the threshold T can be eliminated from (1.19). This will be the relationship between P_{FA} and P_D.

$$\begin{aligned} P_D &= \frac{1}{2}\left[1 - \text{erf}\left(\frac{\sqrt{2N\beta^2}\,\text{erf}^{-1}(1 - 2P_{FA}) - Nm}{\sqrt{2N\beta^2}}\right)\right] \\ &= \frac{1}{2}\left[1 - \text{erf}\left(\text{erf}^{-1}(1 - 2P_{FA}) - \frac{\sqrt{N}m}{\sqrt{2\beta^2}}\right)\right] \\ &= \frac{1}{2}\left[1 - \text{erf}\left(\text{erfc}^{-1}(2P_{FA}) - \frac{\sqrt{N}m}{\sqrt{2\beta^2}}\right)\right] \\ P_D &= \frac{1}{2}\,\text{erfc}\left(\text{erfc}^{-1}(2P_{FA}) - \frac{\sqrt{N}m}{\sqrt{2\beta^2}}\right) \end{aligned} \tag{1.20}$$

The sufficient statistic $\Upsilon(\mathbf{y})$ consists of two parts, the signal part and the noise part. Nm is considered to be the signal component, since the goal is to detect its presence. The signal component has power $(Nm)^2$. The noise component has a power of $N\beta^2$. Therefore the signal-to-noise ratio (SNR) χ for this problem, is defined as $\chi = \frac{(Nm)^2}{N\beta^2}$ and $\sqrt{\chi} = \frac{m\sqrt{N}}{\beta}$. (1.20) can then be reduced to,

$$P_D = \frac{1}{2}\,\text{erfc}\left[\,\text{erfc}^{-1}(2P_{FA}) - \sqrt{\frac{\chi}{2}}\,\right] \tag{1.21}$$

(1.21) is very useful in the sense that one can analytically determine the P_D for a given P_{FA} and SNR. Also, it can be seen that P_D and P_{FA} must rise or fall together and the only way to increase P_D for a given P_{FA} is to increase the SNR for such an optimal detector. All these calculations lead to the following (optimal) detector setup.

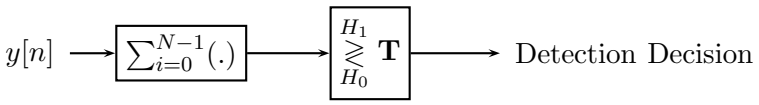

Figure 1.1: Detector setup for a known constant in Gaussian noise

1.3.3 Coherent Detection

Even though the Gaussian problem considered so far is useful for introducing and explaining the major elements of detection such as Neyman-Pearson criteria or the likelihood ratio test, it is not a very good example for modeling real life situations such as radar or sonar detection. Most radar systems use coherent detection, which produces complex-valued measurements. The approach considered so far is valid for real valued data only. Also, the Gaussian approach does not account for unknown parameters. The target amplitude and noise variance, for example, are mostly unknown and must be estimated.

An appropriate model for the noise at the output of a coherent receiver is developed in [4]. The I and Q channels will contain independent, identically distributed zero-mean white Gaussian noise with power $\beta^2/2$. A complex noise process, for which the real and imaginary parts are independent, identical and zero mean is called a *circular symmetric complex normal distribution*. The joint pdf of N complex samples of this distribution is:

$$p_y(\mathbf{y}) = \frac{1}{\pi^N \det(\mathbf{\Gamma})} \exp\{-(\mathbf{y}-\mathbf{m})^H \Gamma^{-1}(\mathbf{y}-\mathbf{m})\} \tag{1.22}$$

Here $\mathbf{m}$ is the $N \times 1$ vector of means of the signal $\mathbf{y} = \mathbf{m} + \mathbf{w}$, $\mathbf{\Gamma}$ is the $N \times N$ covariance matrix of $\mathbf{y}$ and $\mathbf{\Gamma} = \mathrm{E}\{\mathbf{y}\mathbf{y}^H\} - \mathbf{m}\mathbf{m}^H$. Since the noise samples are i.i.d, $\mathbf{\Gamma} = \beta^2 \mathbf{I}_N$, $\det\{\mathbf{\Gamma}\} = \beta^{2N}$ and

$\mathbf{\Gamma^{-1}} = \frac{1}{\beta^2}\mathbf{I}_N$. Then, (1.22) reduces to:

$$p_y(\mathbf{y}) = \frac{1}{\pi^N \beta^{2N}} \exp\{-\frac{1}{\beta^2}(\mathbf{y}-\mathbf{m})^H(\mathbf{y}-\mathbf{m})\} \tag{1.23}$$

One can then write the LRT as:

$$\begin{aligned}
\Lambda = \frac{p_{\mathbf{y}}(\mathbf{y}|H_1)}{p_{\mathbf{y}}(\mathbf{y}|H_0)} &= \frac{\frac{1}{\pi^N\beta^{2N}} \exp\{-\frac{1}{\beta^2}(\mathbf{y}-\mathbf{m})^H(\mathbf{y}-\mathbf{m})\}}{\frac{1}{\pi^N\beta^{2N}} \exp\{-\frac{1}{\beta^2}\mathbf{y}^H\mathbf{y}\}} \\
&= \exp\{-\frac{1}{\beta^2}(\mathbf{y}^H\mathbf{y} - \mathbf{y}^H\mathbf{m} - \mathbf{m}^H\mathbf{y} + \mathbf{m}^H\mathbf{m} - \mathbf{y}^H\mathbf{y})\} \\
&= \exp\{\frac{1}{\beta^2}(2\,\mathrm{Re}\{\mathbf{m}^H\mathbf{y}\} - \mathbf{m}^H\mathbf{m})\} \\
\ln\Lambda &= \frac{1}{\beta^2}[2\,\mathrm{Re}\{\mathbf{m}^H\mathbf{y}\} - \mathbf{m}^H\mathbf{m}] \\
&= \frac{2}{\beta^2}\underbrace{\mathrm{Re}\{\sum_{n=0}^{N-1} m * y_n\}}_{\text{Matched Filter}} - \frac{1}{\beta^2}\underbrace{N|m|^2}_{\text{Energy of the } m \text{ vector}}
\end{aligned} \tag{1.24}$$

(1.24) can be interpreted in the following way. The first term involves a dot product of two complex vectors **m** and **y**. This represents an FIR filtering operation and since the impulse response of this filter is identical to the signal to be detected under H_1, it can be named as a matched filter. The second term, is obviously the energy of the signal **m**. Denote this energy as $E = N|m|^2$.

Obviously, looking at (1.24) one can deduce that the sufficient statistic $\Upsilon(\mathbf{y}) = \mathrm{Re}\{\mathbf{m}^H\mathbf{y}\}$. Expressing the LRT in terms of sufficient statistic yields:

$$\Upsilon = \mathrm{Re}\{\mathbf{m}^H\mathbf{y}\} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\beta^2}{2}\ln(-\lambda) + \frac{E}{2} = T \tag{1.25}$$

Now we need to determine the pdf of the sufficient statistic Υ under both H_0 and H_1. Let $z = m^H y$. Since z is just the sum of independent Gaussian random variables, it will also be Gaussian. Under H_0, y_n is zero-mean noise and so is z. The variance is found from:

$$z = m^H y = \sum_{n=0}^{N-1} m_n * y_n$$

$$\begin{aligned} \text{var}(z) &= \text{var}(\sum_{n=0}^{N-1} m_n * y_n) \\ &= \sum_{n=0}^{N-1} \text{var}(m_n * y_n) = \sum_{n=0}^{N-1} |m_n|^2 \underbrace{\text{var}(y_n)}_{\beta^2} \\ &= \underbrace{\beta^2 \sum_{n=0}^{N-1} |m_n|^2}_{\mathbf{E}} = E\beta^2 \end{aligned} \tag{1.26}$$

Under H_0, $z \sim N(0, E\beta^2)$. Similarly under H_1, $z \sim N(E, E\beta^2)$. We need to find the pdf of $\text{Re}\{z\}$. Gaussian noise splits evenly between the real and imaginary parts of z [3]. Therefore under H_0, $\Upsilon \sim N(0, E\beta^2/2)$ and under H_1, $\Upsilon \sim N(E, E\beta^2/2)$. Evaluating the P_{FA} expression,

$$\begin{aligned} P_{FA} &= \int_T^\infty p_\Upsilon(\Upsilon|H_0)\, d\Upsilon \\ &= \int_T^\infty \frac{1}{\sqrt{2\pi E\beta^2/2}} e^{-\frac{\Upsilon^2}{E\beta^2}}\, d\Upsilon \end{aligned} \tag{1.27}$$

Making the substitution $t = \frac{\Upsilon}{\sqrt{E\beta^2}}$, it is clear that

$$P_{FA} = \frac{1}{2}\left[1 - \text{erf}\left(\frac{T}{\sqrt{\beta^2 E}}\right)\right] \tag{1.28}$$

$$T = \sqrt{\beta^2 E}\, \text{erf}^{-1}(1 - 2P_{FA}) \tag{1.29}$$

Repeating the same procedure used before,

$$P_D = \frac{1}{2}\,\text{erfc}\left(\text{erfc}^{-1}(2P_{FA}) - \sqrt{\frac{E}{\beta^2}}\right) \tag{1.30}$$

As before, E is the signal energy and β^2 is the signal power. Therefore the SNR, $\chi = E/\beta^2$. Then,

$$P_D = \frac{1}{2}\,\mathrm{erfc}\left(\mathrm{erfc}^{-1}(2P_{FA}) - \sqrt{\chi}\right) \tag{1.31}$$

We arrive at the same conclusion that SNR must be increased to improve detection performance for a given false alarm probability.

1.3.4 Unknown parameters

Up to now, every component of the pdfs $p_\Upsilon(\Upsilon|H_0)$ and $p_\Upsilon(\Upsilon|H_1)$ were assumed to be known. In the Gaussian example, the signal means $\mathbf{m}$ and the noise variance β^2 were assumed to be known. This is not the case in the real world because these parameters cannot be known perfectly. Now consider the case where the magnitude of the returning signal m is known but the phase is not. Let $m = \tilde{m}\exp(j\theta)$ where the signal phase, θ can be modeled as a uniform random variable distributed in $[0, 2\pi]$.

Using the *Bayes theorem*, one can find the pdf of a random variable by conditioning it on other random variables. For this case,

$$p_\mathbf{y}(\mathbf{y}|H_i) = \int p_\mathbf{y}(\mathbf{y}|H_i,\theta)p_\theta(\theta)\,d\theta \qquad i = 0{,}1 \tag{1.32}$$

Then the following pdfs can be defined:

$$p_\mathbf{y}(\mathbf{y}|H_0,\theta) = \frac{1}{\pi^N\beta^{2N}}\exp\left[-\frac{1}{\beta^2}\mathbf{y}^H\mathbf{y}\right] \tag{1.33}$$

$$p_\mathbf{y}(\mathbf{y}|H_1,\theta) = \frac{1}{\pi^N\beta^{2N}}\exp\left[-\frac{1}{\beta^2}(\mathbf{y}-\tilde{\mathbf{m}}e^{j\theta})^H(\mathbf{y}-\tilde{\mathbf{m}}e^{j\theta})\right] \tag{1.34}$$

Note that $p_\mathbf{y}(\mathbf{y}|H_0)$ does not depend on θ, so there is no need to use Bayes theorem. (ie: $p_\mathbf{y}(\mathbf{y}|H_0) = p_\mathbf{y}(\mathbf{y}|H_0,\theta)$) For $p_\mathbf{y}(\mathbf{y}|H_1)$:

$$\begin{aligned} p_\mathbf{y}(\mathbf{y}|H_1,\theta) &= \frac{1}{\pi^N\beta^{2N}}\exp\left[-\frac{1}{\beta^2}(\mathbf{y}^H\mathbf{y} - 2\,\mathrm{Re}\{\tilde{\mathbf{m}}^H\mathbf{y}e^{-j\theta}\} + \mathbf{m}^H\mathbf{m})\right] \\ &= \frac{1}{\pi^N\beta^{2N}}\exp\left[-\frac{1}{\beta^2}(\mathbf{y}^H\mathbf{y} - 2|\tilde{\mathbf{m}}^H\mathbf{y}|\cos(\phi-\theta)+E)\right] \end{aligned} \tag{1.35}$$

where ϕ is the unknown, but fixed, phase of the inner product $\tilde{\mathbf{m}}^H\mathbf{y}$. Using, $p_{\mathbf{y}}(\mathbf{y}|H_1) = \int p_{\mathbf{y}}(\mathbf{y}|H_1,\theta)p_\theta(\theta)\,d\theta$ we get:

$$p_{\mathbf{y}}(\mathbf{y}|H_1) = \frac{1}{\pi^N\beta^{2N}}e^{-(\mathbf{y}^H\mathbf{y}+\mathbf{E})/\beta^2}\frac{1}{2\pi}\int_0^{2\pi}\exp\left[\frac{2}{\beta^2}|\tilde{\mathbf{m}}^H\mathbf{y}|\cos\theta'\right]d\theta' \tag{1.36}$$

where $\theta' = \phi - \theta$. To evaluate this integral, one needs to use the following definition of the modified Bessel function of the first kind, $I_0(x)$.

$$I_0(x) \triangleq \frac{1}{2\pi}\int_0^{2\pi} e^{x\cos\theta}\,d\theta \tag{1.37}$$

Using this result, (1.36) becomes

$$p_{\mathbf{y}}(\mathbf{y}|H_1) = \frac{1}{\pi^N\beta^{2N}}e^{-(\mathbf{y}^H\mathbf{y}+\mathbf{E})/\beta^2}I_0\left(\frac{2|\tilde{\mathbf{m}}^H\mathbf{y}|}{\beta^2}\right) \tag{1.38}$$

The LRT and the log-LRT now become

$$\begin{aligned}\Lambda = \frac{p_{\mathbf{y}}(\mathbf{y}|H_1)}{p_{\mathbf{y}}(\mathbf{y}|H_0)} &= \frac{\frac{1}{\pi^N\beta^{2N}}e^{-(\mathbf{y}^H\mathbf{y}+\mathbf{E})/\beta^2}I_0\left(\frac{2|\tilde{\mathbf{m}}^H\mathbf{y}|}{\beta^2}\right)}{\frac{1}{\pi^N\beta^{2N}}e^{\left(-\frac{1}{\beta^2}\mathbf{y}^H\mathbf{y}\right)}} \\ &= e^{-\frac{E}{\beta^2}}I_0\left(\frac{2|\tilde{\mathbf{m}}^H\mathbf{y}|}{\beta^2}\right)\end{aligned} \tag{1.39}$$

$$\ln\Lambda = \ln\left[I_0\left(\frac{2|\tilde{\mathbf{m}}^H\mathbf{y}|}{\beta^2}\right)\right] - \frac{E}{\beta^2} \underset{H_0}{\overset{H_1}{\gtrless}} \ln(-\lambda) \tag{1.40}$$

Alternatively,

$$\underbrace{\ln\left[I_0\left(\frac{2|\tilde{\mathbf{m}}^H\mathbf{y}|}{\beta^2}\right)\right]}_{\Upsilon} \underset{H_0}{\overset{H_1}{\gtrless}} T \tag{1.41}$$

Using the LRT and (1.41) the signal processing required for optimal detection of a signal in the presence of unknown phase is determined. One needs to pass the signal through a

matched filter, then take its magnitude and apply the nonlinearity $\ln[I_0()]$. Because the function $\ln[I_0()]$ is a monotonically increasing function, same results can be obtained using the threshold test

$$|\tilde{\mathbf{m}}^H\mathbf{y}| \underset{H_0}{\overset{H_1}{\gtrless}} T' \tag{1.42}$$

To establish the performance of this detector, we need to find P_{FA} and P_D for the threshold test explained in (1.42). Let $z = \tilde{\mathbf{m}}^H\mathbf{y}$. We need to find the distribution of $|z|$ under both H_0 and H_1. Under H_0, we know $\tilde{\mathbf{m}}^H\mathbf{y} \sim N(0, E\beta^2)$ and the real and imaginary parts of $\tilde{\mathbf{m}}^H\mathbf{y}$ are independent with variance $E\beta^2/2$. Using $|z| = \sqrt{{z_R}^2 + {z_I}^2}$ where z_R and z_I denote the real and imaginary parts of z respectively and both of them are $N(0, E\beta^2/2)$. Using the following property of Rayleigh distribution; $R \sim$ Rayleigh(σ) is a Rayleigh distribution if $R = \sqrt{X^2 + Y^2}$ where $X \sim N(0, \sigma^2)$ and $Y \sim N(0, \sigma^2)$ are two independent normal distributions. Since z_R and z_I are two independent zero-mean normal distributions, $|z|$ is Rayleigh distributed with the parameter $\sigma^2 = E\beta^2/2$. Then:

$$p_z(z|H_0) = \frac{z}{E\beta^2/2} e^{(-z^2/E\beta^2)} \tag{1.43}$$

The false alarm probability is

$$\begin{aligned} P_{FA} &= \int_T^\infty p_z(z|H_0)\, dz \\ &= \int_T^\infty \frac{2z}{E\beta^2} e^{(-z^2/E\beta^2)}\, dz \end{aligned}$$

The substitution $t = \frac{z^2}{E\beta^2}$, $dt = \frac{2zdz}{E\beta^2}$ leads the following results,

$$P_{FA} = \int_{\frac{T^2}{E\beta^2}}^\infty e^{-t}\, dt = \exp\left(\frac{-T^2}{E\beta^2}\right) \tag{1.44}$$

$$T = \sqrt{-E\beta^2 \ln P_{FA}} \tag{1.45}$$

The threshold for a given false alarm probability can be computed from (1.44) and (1.45). Let us calculate the P_D for this given threshold. Under H_1 , $\tilde{\mathbf{m}}^H\mathbf{y} \sim N(E, E\beta^2)$. Again let $z =$

$\tilde{\mathbf{m}}^H\mathbf{y}$. Since E is real valued, $z_R \sim N(E, E\beta^2/2)$ and $z_I \sim N(0, E\beta^2/2)$. Using the following property of Rician distribution: $R \sim \text{Rice}(\nu, \sigma)$ has a Rice distribution if $R = \sqrt{X^2 + Y^2}$ where $X \sim N(\nu\cos\theta, \sigma^2)$ and $Y \sim N(\nu\sin\theta, \sigma^2)$ are statistically independent normal random variables and θ is any real number. In our case, $\nu = E$, $\theta = 0$ and $\sigma^2 = E\beta^2/2$.

$$p_z(z|H_1) = \frac{2z}{E\beta^2}\exp\Big[-(\frac{z^2+E^2}{E\beta^2})\Big]I_0(\frac{2z}{\beta^2}) \tag{1.46}$$

The required integral is $P_D = \int_T^\infty p_z(z|H_1)\,dz$. The required integral can be reduced to the following normal form which is called *Marcum's Q function*.

$$Q_M(\alpha,\gamma) = \int_\gamma^\infty t\exp[-\frac{1}{2}(t^2+\alpha^2)]I_0(\alpha t)\,dt \tag{1.47}$$

Letting $t = \frac{z}{\sqrt{E\beta^2/2}}$ and $\alpha = \frac{\sqrt{2E}}{\beta}$ gives the following result:

$$P_D = Q_M\Big(\sqrt{\frac{2E}{\beta^2}}, \sqrt{\frac{2T^2}{E\beta^2}}\Big) \tag{1.48}$$

Using the definition of SNR, $\chi = E/\beta^2$ and $T = \sqrt{-E\beta^2\ln P_{FA}}$ one arrives at:

$$P_D = Q_M(\sqrt{2\chi}, \sqrt{-2\ln P_{FA}}) \tag{1.49}$$

All these calculations lead to the following detector setup.

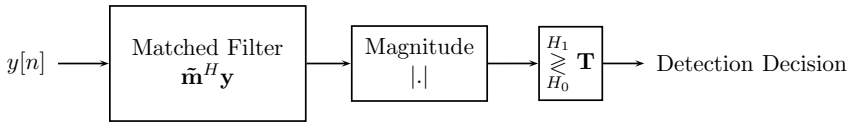

Figure 1.2: Detector setup for a known constant with unknown phase in Gaussian noise

1.3.5 Radar Signal Detection

In a realistic radar scenario, the amplitude and phase of the returning signal, the time of arrival and the Doppler shift are unknown. To account for these unknown parameters, some more signal processing techniques must be used. There are many methods used to improve detection performance under such situations such as coherent integration, non-coherent integration, binary integration etc. Also targets can be non-fluctuating (ie: constant) or fluctuating. Here only the case of non-fluctuating targets will be explained.

Let y_n be the individual data samples where under H_0, $y_n = w_n$ and under H_1, $y_n = m + w_n$. Here $m = \tilde{m} \exp j\theta$ for some real amplitude $\tilde{m}$ and random phase θ. Under H_0, the pdf of $z_n = |y_n|$ is the Rayleigh pdf:

$$p_{z_n}(z_n|H_0) = \frac{2z_n}{\beta^2} e^{(-{z_n}^2/\beta^2)} \tag{1.50}$$

Under H_1, z_n has the Rician pdf:

$$p_{z_n}(z_n|H_1) = \frac{2z_n}{\beta^2} e^{-(\frac{z_n^2+\tilde{m}^2}{\beta^2})} I_0(\frac{2\tilde{m}z_n}{\beta^2}) \tag{1.51}$$

The joint pdf of N-samples of z_n is:

$$p_{\mathbf{z}}(\mathbf{z}|H_0) = \prod_{n=0}^{N-1} \frac{2z_n}{\beta^2} e^{(-{z_n}^2/\beta^2)} \tag{1.52}$$

$$p_{\mathbf{z}}(\mathbf{z}|H_1) = \prod_{n=0}^{N-1} \frac{2z_n}{\beta^2} e^{-(\frac{z_n^2+\tilde{m}^2}{\beta^2})} I_0(\frac{2\tilde{m}z_n}{\beta^2}) \tag{1.53}$$

The LRT and the log-LRT can be written as:

$$\begin{aligned}
\Lambda = \frac{p_{\mathbf{z}}(\mathbf{z}|H_1)}{p_{\mathbf{z}}(\mathbf{z}|H_0)} &= \frac{\prod_{n=0}^{N-1} \frac{2z_n}{\beta^2} e^{-(\frac{z_n^2+\tilde{m}^2}{\beta^2})} I_0(\frac{2\tilde{m}z_n}{\beta^2})}{\prod_{n=0}^{N-1} \frac{2z_n}{\beta^2} e^{(-{z_n}^2/\beta^2)}} \\
&= \prod_{n=0}^{N-1} \exp(-\frac{\tilde{m}^2}{\beta^2}) I_0(\frac{2\tilde{m}z_n}{\beta^2}) \\
&= e^{-\tilde{m}^2/\beta^2} \sum_{n=0}^{N-1} I_0(\frac{2\tilde{m}z_n}{\beta^2}) \qquad (1.54) \\
\ln\Lambda &= -\frac{\tilde{m}^2}{\beta^2} + \sum_{n=0}^{N-1} \ln\left[I_0(\frac{2\tilde{m}z_n}{\beta^2})\right] \qquad (1.55)
\end{aligned}$$

In sufficient statistic form,

$$\ln\left[I_0(\frac{2\tilde{m}z_n}{\beta^2})\right] \underset{H_0}{\overset{H_1}{\gtrless}} \ln(-\lambda) + \frac{\tilde{m}^2}{\beta^2} = T \tag{1.56}$$

Using the approximation:

$$\ln[I_0(x)] \approx \frac{x^2}{4}, \ln[I_0(\frac{2\tilde{m}z_n}{\beta^2})] \approx \frac{4\tilde{m}^2{z_n}^2}{\beta^4}$$

The detection test then becomes:

$$\begin{aligned}
\sum_{n=0}^{N-1} \frac{\tilde{m}^2{z_n}^2}{\beta^4} &\underset{H_0}{\overset{H_1}{\gtrless}} T \\
\underbrace{\sum_{n=0}^{N-1} {z_n}^2}_{\Upsilon} &\underset{H_0}{\overset{H_1}{\gtrless}} T' \qquad (1.57)
\end{aligned}$$

Looking at (1.57) the sufficient statistic Υ turns out to be the sum of the magnitude squares of the individual data samples. The performance of this detector should now be calculated. Figure (1.3) shows the proposed detector setup and the corresponding pdfs under H_0.

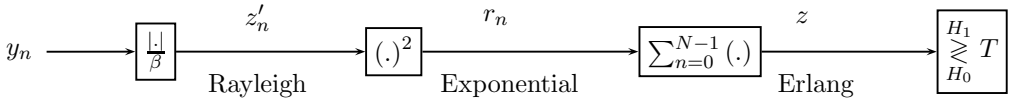

Figure 1.3: Detector setup for nonfluctuating targets and corresponding pdfs under H_0

We know that y_n is normal with zero mean and variance β^2 distributed evenly between its independent real and imaginary parts. Its magnitude, $|y_n|$, is Rayleigh distributed, $|y_n| \sim$ Rayleigh($\sqrt{\beta^2/2}$). Since $z'_n = |y_n|/\beta$, $z'_n \sim$ Rayleigh($1/\sqrt{2}$). Using the fact that, for $Y \sim$ Rayleigh(σ), $X \sim$ Exponential(λ) if $X = \frac{Y^2}{2\lambda\sigma^2}$. Here $\sigma^2 = 1/2$, so $\lambda = 1$ and $r_n \sim$ Exponential(1). If $x_i \sim$ Exponential(λ) and $Y = \sum_{i=0}^{N-1} x_i$ then $Y \sim$ Gamma($N, 1/\lambda$). Here $\lambda = 1$ so $z \sim$ Gamma($N, 1$), Gamma distribution with integer shape parameter is also called the Erlang distribution. So the pdf of z is:

$$f_z(z|H_0) = \frac{z^{N-1}e^{-z}}{(N-1)!} \tag{1.58}$$

To find P_{FA} one needs to consider the integral:

$$P_{FA} = \int_T^\infty \frac{z^{N-1}e^{-z}}{(N-1)!}\, dz \tag{1.59}$$

For the case of $N = 1$ (single sample) (1.59) reduces to:

$$P_{FA} = e^{-T} \tag{1.60}$$

$$T = -\ln P_{FA} \tag{1.61}$$

The detection probability for this case and when $N \neq 1$ is quite complicated and can be found in [4]. When the normalization (division by β in Figure (1.3)) is not done, $T = -\beta^2 \ln P_{FA}$ and $P_{FA} = e^{-T/\beta^2}$. The square law detector for a single sample uses these expressions to set the threshold for a given false alarm probability and known noise variance.

CHAPTER 2

CONSTANT FALSE ALARM RATE DETECTION

Up to this point, we have assumed that the interference level (ie: β) is known and constant, thus allowing one to set the threshold for a desired false alarm probability. However, in real life applications that is rarely the case. In practice, the interference levels are rarely known and they are not constant. To combat these difficulties, a set of adaptive methods jointly known as *Constant False Alarm Rate (CFAR)* methods are used. These methods provide predictable detection performance and false alarm probability under real life scenarios.

2.1 Case of Unknown Interference Power

As noted earlier, the expressions for false alarm probability and the required threshold are as follows $T = -\beta^2 \ln P_{FA}$ and $P_{FA} = e^{-T/\beta^2}$. Assume that the threshold is preset using an estimated interference power β_0 to satisfy a false alarm probability P_{FA0}, but the actual interference power is β^2. Let us calculate the increase in the false alarm probability.

$$\begin{aligned} P_{FA} &= \exp\left(\frac{\beta_0^2 \ln P_{FA0}}{\beta^2}\right) = \exp\left(\ln P_{FA0}^{\beta_0^2/\beta^2}\right) \\ &= P_{FA0}^{\beta_0^2/\beta^2} \end{aligned} \tag{2.1}$$

(2.1) shows that even a small increase of 2dB can cause an increase in P_{FA} of 1.5 to 3 orders of magnitude. This is highly undesirable and shows that the actual interference power β^2 must be estimated from the data in real time. This estimate will then be used to set the threshold. A detector that can maintain a constant false alarm rate regardless of the interference power, is called a constant false alarm rate (CFAR) processor.

2.2 Types of CFAR Processors

General scheme of a CFAR detector is shown in Figure 2.1. The threshold adjustment for a specific resolution cell is based on the signal levels around it. There are many variations and modifications to this general scheme.

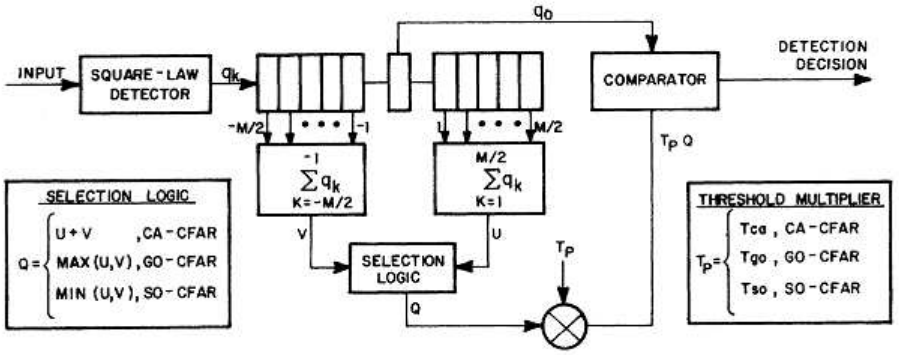

Figure 2.1: General CFAR Detector scheme

2.2.1 Cell Averaging CFAR

Previously, it was shown that for the square law detector, each cell x_i has an exponential pdf with the parameter β^2. It can easily be shown that the maximum likelihood (ML) estimate of β^2 for a vector $\mathbf{x}$ of N such samples is just the average of the available samples. [4].

$$\widehat{\beta^2} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.2}$$

The threshold is then set as a multiple of estimated interference power.

$$\hat{T} = \alpha\widehat{\beta^2} \tag{2.3}$$

Since the interference power and therefore the threshold are estimated from the average power in the cells around the test cell, this approach is called cell-averaging CFAR. To find α, one needs to consider the effect of estimating noise power over a finite number of cells, rather than

knowing it exactly.

The estimated threshold can be written as

$$\hat{T} = \frac{\alpha}{N} \sum_{i=1}^{N} x_i \tag{2.4}$$

Let $y = \sum_{i=1}^{N} x_i$. We know that the sum of N independent identically distributed exponential random variables with rate λ has the distribution Gamma$(N, 1/\lambda)$. Then pdf of y is the Erlang density

$$p_y(y) = \frac{1}{\beta^{2N}} \frac{y^{N-1}}{(N-1)!} e^{-y/\beta^2} \tag{2.5}$$

Since $\hat{T} = (\alpha/N)y$, the pdf of $\hat{T}$ is (for $\hat{T} > 0$)

$$p_{\hat{T}}(\hat{T}) = \left(\frac{N}{\alpha\beta^2}\right)^N \frac{\hat{T}^{N-1}}{(N-1)!} e^{-N\hat{T}/\alpha\beta^2} \tag{2.6}$$

For the threshold $\hat{T}$, $P_{FA} = e^{-\hat{T}/\beta^2}$. The expected value of P_{FA} is now

$$\begin{aligned} \bar{P}_{FA} &= \int_{-\infty}^{\infty} e^{-\hat{T}/\beta^2} p_{\hat{T}}(\hat{T})\, d\hat{T} \\ &= \left(\frac{N}{\alpha\beta^2}\right)^N \frac{1}{(N-1)!} \int_0^{\infty} \hat{T}^{N-1} e^{-\hat{T}/\beta^2} e^{-N\hat{T}/\alpha\beta^2}\, d\hat{T} \\ &= \left(\frac{N}{\alpha\beta^2}\right)^N \frac{1}{(N-1)!} \int_0^{\infty} \hat{T}^{N-1} e^{-[(N/\alpha)+1]\hat{T}/\beta^2}\, d\hat{T} \end{aligned} \tag{2.7}$$

Using the fact that

$$\int_0^{\infty} x^n e^{-ax}\, dx = \frac{n!}{a^{n+1}} \qquad \text{for integer } n \tag{2.8}$$

one can compute (2.7) as

$$\bar{P}_{FA} = \left(1 + \frac{\alpha}{N}\right)^{-N} \tag{2.9}$$

and the threshold multiplier for a given P_{FA} as

$$\alpha = N(P_{FA}^{-1/N} - 1) \tag{2.10}$$

Equations (2.9) and (2.10) can be used to set the threshold for a fixed false alarm probability. It also shows that since the false alarm probability does not depend on the actual interference power, this detector exhibits CFAR behavior. The average detection probability is found as [4].

$$\bar{P}_D = \left(1 + \frac{\alpha}{N(1+\chi)}\right)^{-N} \tag{2.11}$$

where χ is the SNR.

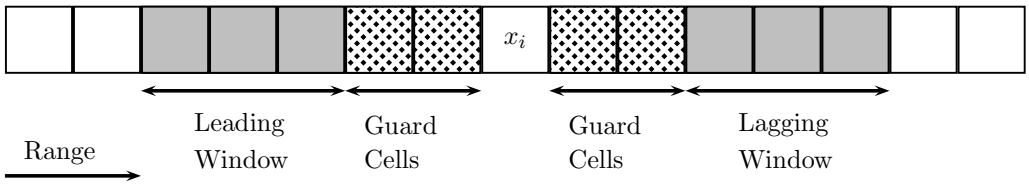

Figure 2.2: CFAR reference window. x_i is the cell under test (CUT)

Figure(2.2) shows the reference window arrangement for a one-dimensional data vector of range (or time) cells. The gray cells are called the *reference cells* and represent the data from ranges nearer and farther from the cell under test (x_i). These cells are averaged to find the estimate of the noise power. The dotted cells are called *guard cells* and are excluded from the averaging operation. This is due to the fact that the cells immediately adjacent to x_i would contain both interference and target energy and do not represent the interference alone. If these cells were included in the averaging operation, the estimate of β^2 would be too high. This, in turn, causes the detection threshold to be too high, lowering the P_{FA} and P_D. Obviously, this is not desired. The combined window of guard cells, reference cells and the cell under test (CUT) is called as *CFAR window*.

Since the threshold is estimated from a finite number of N samples rather than exactly known, the threshold is usually higher than the ideal one. This is necessary to compensate for the unknown interference power and to guarantee the desired P_{FA}. Since the threshold multiplier is higher than ideal, this means that to achieve a specified P_D for a given P_{FA} a higher SNR is necessary than would be were the noise power known exactly. This increase in SNR is called the *CFAR loss*. To quantify this CFAR loss, plug (2.10) into (2.11)

$$P_D = \left(1 + \frac{N(P_{FA}^{-1/N} - 1)}{N(1+\chi)}\right)^{-N} \tag{2.12}$$

If (2.12) is solved for the required SNR for a given P_D and P_{FA} we get

$$\chi_N = \frac{(P_D/P_{FA})^{1/N} - 1}{1 - P_D^{1/N}} \tag{2.13}$$

where χ_N is a function of the number of cells averaged. It can be shown that ([4], [6]) that as $N \to \infty$, the SNR χ_∞ is

$$\chi_\infty = \frac{\ln(P_{FA}/P_D)}{\ln(P_D)} \tag{2.14}$$

Then, the CFAR loss is defined as

$$L_{\text{CFAR}} = \frac{\chi_N}{\chi_\infty} \tag{2.15}$$

It has been reported that for small ($N < 20$) reference windows, the CFAR loss can be several dB [4]. High CFAR losses does not permit the use of values of N less than 10.

CA CFAR algorithms have some drawbacks. First of all, the threshold value increases in the vicinity of the target. When two or more targets are present and one target is in the test cell while another target is in the reference cell, the threshold value will increase. The target in the reference window can 'mask' the the target in the test cell because of this increased threshold, this effect is called target masking. Also, for targets that accompany many range cells, the reference cell and the test cell might both contain targets and this might cause the target to

mask 'itself'. This is called self masking. In addition to masking effects, CA CFAR suffers from false alarms at *clutter edges*. Clutter edges are the boundaries between two clutter regions having different reflectivities. These edges can cause false alarms at the edge and allow masking of targets in the lower-reflectivity region near the edge.

Figure (2.3) illustrates the operation of CA CFAR. The simulated data corresponds to two regions of Gaussian I/Q noise with power $10\log_{10}({\beta_1}^2) = 20$dB and $10\log_{10}({\beta_2}^2) = 30$dB. The transition between these two regions occur at the range bin 750. There is a single target that occupies 20 range cells with power 35dB therefore with an SNR of $10\log_{10}(\chi) = 15$dB. P_{FA} is set to 10^{-3}. The CFAR reference window consists of leading and lagging windows of 10 cells and 20 guard cells in each side. 20 guard cells are used because that is the length (or duration) of the target. It can be seen that the estimated threshold tracks the ideal threshold well, except in the vicinity of the target. In the vicinity of the target, the threshold increases due to the fact that the target will be in the leading or lagging windows therefore effecting the threshold computation.

The clutter edge performance of CA CFAR is not very satisfying. The estimated threshold tracks each region, but the transition region is very large. Because the estimated threshold only rises to the correct levels several cells after the transition, false alarms might be observed. Also, targets near the clutter edge may be missed because of the clutter from the high-reflectivity region.

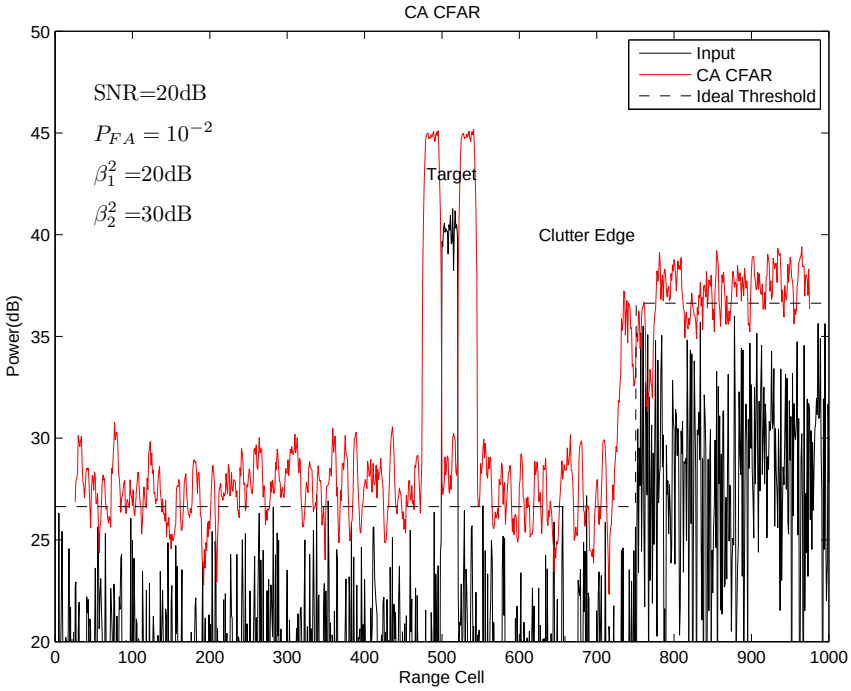

Figure 2.3: CA CFAR threshold behavior and clutter edge performance

The main advantage of CA CFAR is that it is simple to perform and does not require a lot of computing power. Nonetheless, non homogeneous clutter and the presence of interfering targets have led to the development of some extensions to the CA CFAR concept. These extensions are designed to combat one or more of the shortcomings of the CA CFAR algorithm.

2.2.2 Smallest-of-Cell-Averaging CFAR

The smallest-of-cell-averaging CFAR (SOCA CFAR) is intended to combat the masking effects caused by interfering targets among the CFAR reference cells. Let the estimate of the noise power in the leading window be β_1^2 and the lagging window be β_2^2. The SOCA CFAR approach estimates the threshold by taking the minimum of these estimates.

$$\hat{T} = \alpha_{so} \min(\widehat{\beta_1^2}, \widehat{\beta_2^2}) \tag{2.16}$$

If an interfering target is present in one of the reference windows, its threshold estimate will

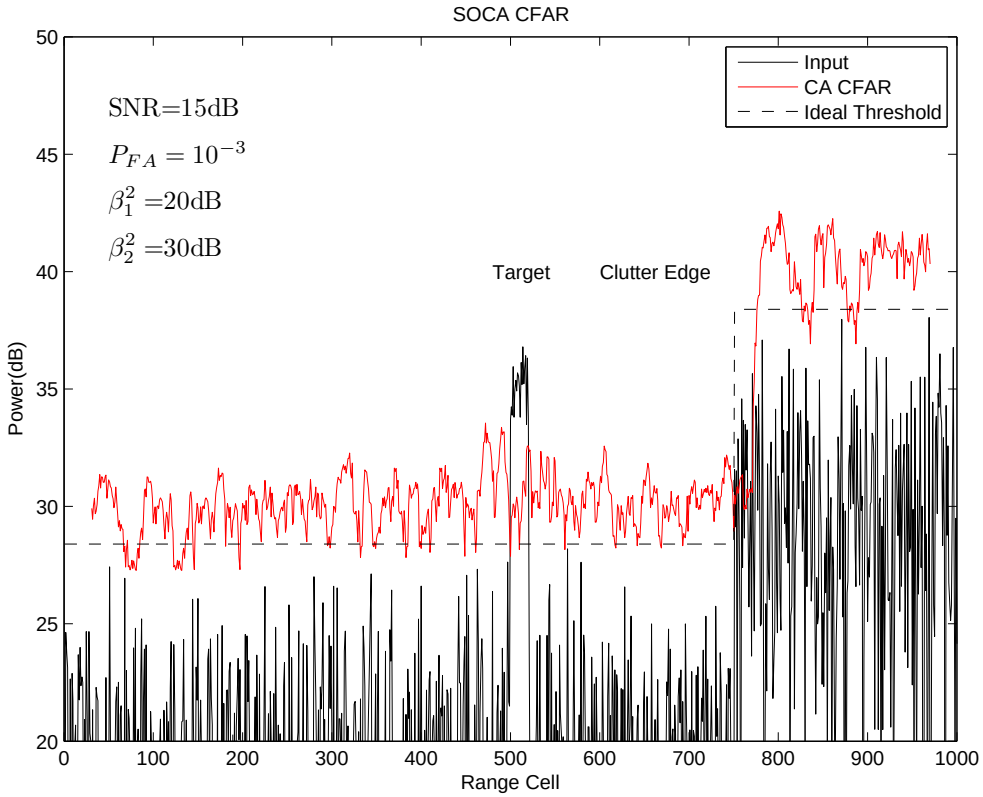

Figure 2.4: SOCA CFAR threshold behavior and clutter edge performance

increase. Because of the minimum operator, the other reference cell will be chosen and detection will successfully be performed. Since the threshold is estimated using $N/2$ cells instead of N cells, the threshold multiplier will have to change. It is shown that the required multiplier can be found from solving (2.17) iteratively [7]

$$P_{FA}/2 = \left(2 + \frac{\alpha_{so}}{N/2}\right)^{-N/2}\left[\sum_{k=0}^{N/2-1}\binom{N/2-1+k}{k}\left(2 + \frac{\alpha_{so}}{N/2}\right)^{-k}\right] \tag{2.17}$$

Figure (2.4) shows the SOCA CFAR behavior and clutter edge performance. Unlike CA CFAR, the threshold does not increase in the vicinity of the target. This is due to the use of minimum operator. When the target is in the lagging window, the leading window has a smaller estimated threshold and therefore the leading window is chosen by the minimum operator. Similarly, when the target is in the leading window, the lagging window has a smaller estimated threshold and therefore the lagging window is chosen. This allows for the correct

setting of threshold. This also allows for combating the effects of multiple targets and self masking. Using SOCA CFAR, detection of closely spaced targets is possible.

The main failing point of SOCA CFAR algorithm is seen in clutter edges. When the CFAR window crosses the clutter edge, there will be a region where the test cell is in the higher interference region and the leading window is filled mostly with the lower interference region. The minimum operator makes sure that the threshold is estimated using the lower interference power and therefore many false alarms are seen during the transition region.

2.2.3 Greater-of-cell-averaging CFAR

For environments where interfering or closely-spaced targets are unlikely but the clutter is highly non homogeneous and clutter edge false alarms are very important, another CFAR approach is developed. In this case, the greater-of-cell-averaging CFAR (GOCA CFAR) is used. Just like the previous method, GOCA CFAR uses the threshold estimate shown in (2.18)

$$\hat{T} = \alpha_{GO} \max(\widehat{\beta_1^2}, \widehat{\beta_2^2}) \tag{2.18}$$

Similar to the SOCA case, the GOCA multiplier is found from [7]

$$P_{FA}/2 = \left(1 + \frac{\alpha_{GO}}{N/2}\right)^{-N/2} - \left(2 + \frac{\alpha_{GO}}{N/2}\right)^{-N/2} \times \left[\sum_{k=0}^{N/2-1} \binom{N/2-1+k}{k} \left(2 + \frac{\alpha_{GO}}{N/2}\right)^{-k}\right]$$

The operation of GOCA CFAR can be seen in Fig.(2.5). Not surprisingly, the maximum operator successfully avoids the false alarms at the clutter edge. However, the threshold increases abruptly in the vicinity of the target and this increase is larger than that of CA CFAR. This means that the GOCA CFAR algorithm is more vulnerable target masking and self masking effects. If there is more than one target present, the stronger targets may mask the weaker targets. The GOCA CFAR may also miss targets near the clutter edge due to the transition region.

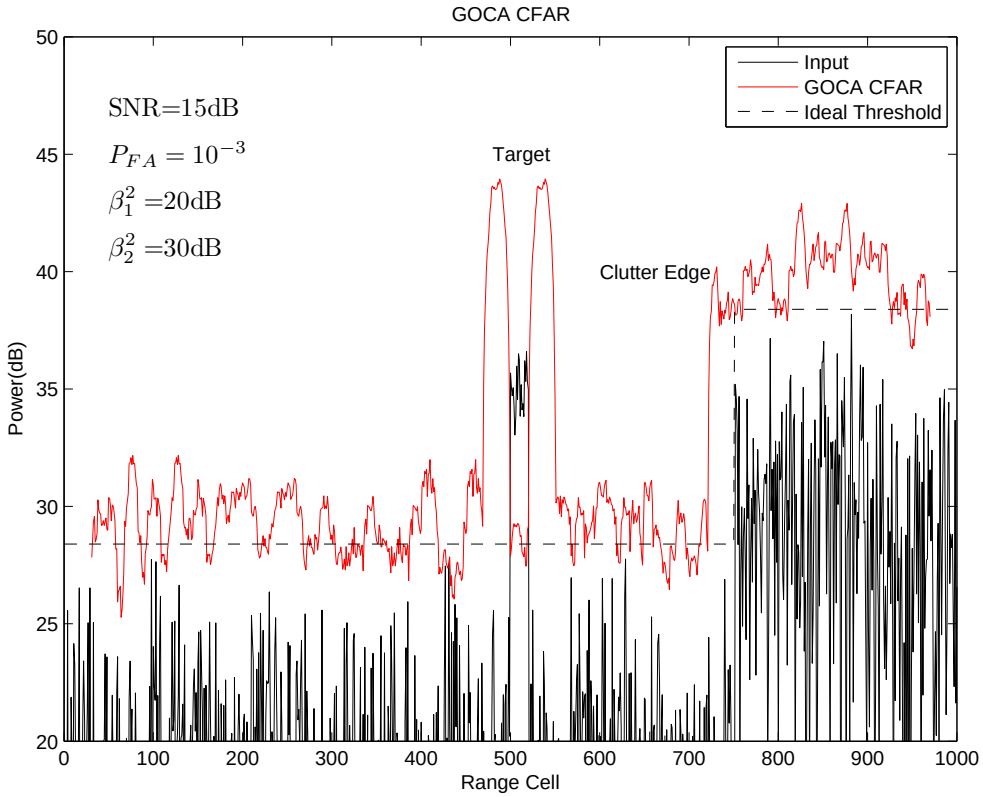

Figure 2.5: GOCA CFAR threshold behavior and clutter edge performance

2.2.4 Censored or Trimmed mean CFAR

Another way to combat with interfering targets and target masking problem is to make use of the *censored* or *trimmed mean* CFAR ([8], [9]). In this approach, the N_C reference cells ($N_C < N$) having the highest power are discarded and the threshold is estimated using the remaining $N - N_C$ cells. Such detectors are called as *Censored Mean Level Detector* (CMLD). Censoring the cells having the lowest power is also possible and this approach is useful in reducing bias problems when the noise has a spectral slope [10]. However, here we are concerned with detection robustness in multiple target environments and will therefore concentrate on CMLDs with censoring applied to cells having higher power since when an interferer is present in the noise reference samples, that sample is expected to have a high power. The CMLD detector uses the threshold estimate shown in (2.19)

$$\hat{T} = \alpha_{\text{CMLD}} \sum_{j=1}^{N-N_C} x_{(j)} \tag{2.19}$$

where α_{CMLD} is the threshold multiplier, N_C is the number of censored cells and the sequence $\{x_{(1)}, x_{(2)}, \ldots, x_{(n)}\}$ is the ordered sequence of reference cells such that $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(k)} \leq \ldots \leq x_{(n)}$. The threshold multiplier is found from [11]

$$P_{FA} = M \times \left(\frac{1}{1 + \alpha_{\text{CMLD}}} \right)^{N - N_C - 1} \tag{2.20}$$

where

$$M = \frac{N!}{N_C!(N - N_C - 1)!(N - N_C)} \times \sum_{j=0}^{N_C} \frac{\binom{N_C}{j}(-1)^{N_C - j}}{\frac{N-j}{N-N_C} + \alpha_{\text{CMLD}}} \tag{2.21}$$

This process of censoring will eliminate the elevating effect of interfering targets on the threshold estimate. The choice of N_C requires the knowledge of how many interfering targets are expected. It is noted that typically half to a quarter of the cells are discarded [12]. The censoring method can be combined with any of the CA, SOCA or GOCA techniques as in [13]. In censored CA CFAR (CMLD), the N_C cells having the highest power are discarded from both the leading and lagging windows, then conventional CA method is used as in (2.19). The censored GOCA and censored SOCA CFAR methods are also reasonable.

The operation of CMLD can be seen in Fig.(2.6). There is an increase of the threshold in the vicinity of the target however it is much less than the increase in standard CA CFAR. This means this method can tolerate interfering targets and masking effects. The clutter edge performance of CMLD is worse than CA CFAR and SOCA CFAR [11]. The main disadvantage of censored CFAR methods is that they require implementation of a logic to rank and order the cells. N number of cells must be ordered for every reference window and this means increased algorithm complexity and computational power. Alternate equations for P_D and P_{FA} can be found in [8].

2.2.5 Order Statistic CFAR

The censored CFAR approach has introduced the idea of ordering to the CFAR concept. A new class of CFAR detectors, namely *rank-based* or *order-statistic* CFARs (OS CFAR), then

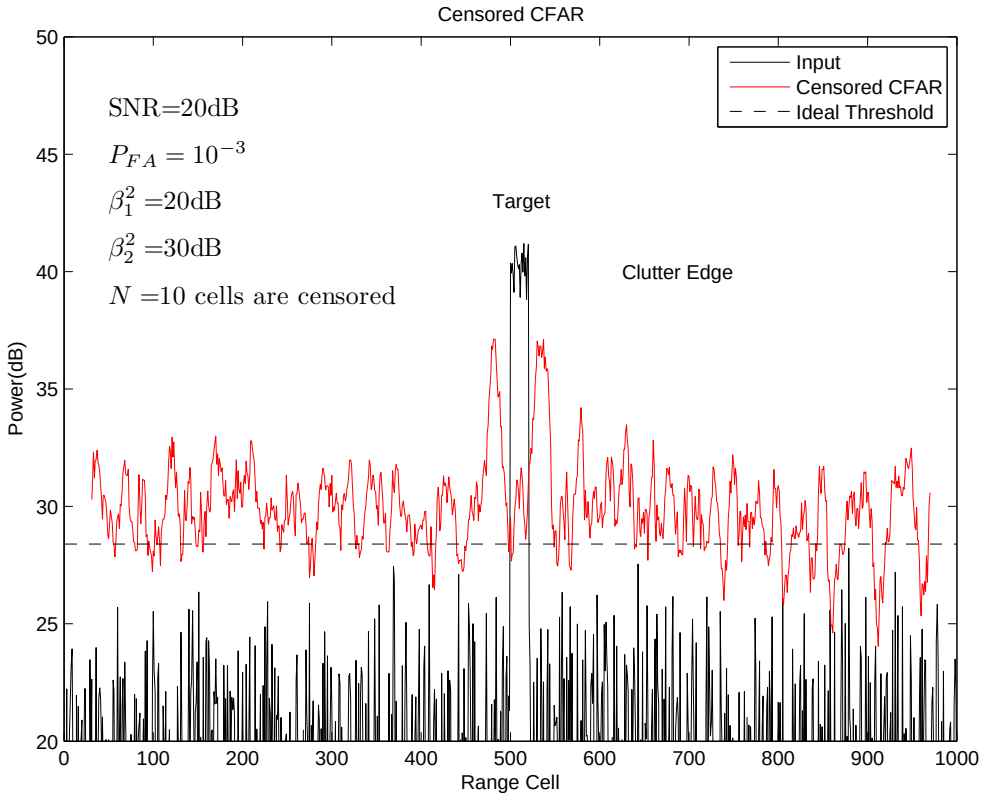

Figure 2.6: CMLD threshold behavior and clutter edge performance

emerged. The primary purpose of this type of detectors is to combat multiple targets and masking effects. The sliding window structure of CA CFAR is still used but guard cells are much less important because the averaging operator is not used to estimate the interference level. Instead, OS CFAR rank orders the reference window data samples and chooses only one element of the ordered list as a representative of the threshold level. The samples are ordered to form a new sequence in ascending numerical order $\{x_{(1)}, x_{(2)}, \ldots, x_{(n)}\}$ where $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(k)} \leq \ldots \leq x_{(n)}$. The kth element of the ordered list ($x_{(k)}$) is called the kth order statistic. In OS CFAR, the kth order statistic is used as a representative of the interference level and the threshold is set as a multiple of this value

$$\hat{T} = \alpha_{OS}\, x_{(k)} \tag{2.22}$$

Note that the threshold is estimated using only one sample, instead of an average of all data samples. However, all the data samples are required to determine which will be the kth largest,

therefore actually the threshold is dependent on all data samples. It can be shown that this algorithm is in fact CFAR (the false alarm probability is not dependent on the interference power β^2). The false alarm probability is ([4], [6])

$$P_{FA} = k\binom{N}{k}\frac{\Gamma(\alpha_{OS} + N - k + 1)\Gamma(k)}{\Gamma(\alpha_{OS} + N + 1)} \tag{2.23}$$

For integer α_{OS}

$$P_{FA} = \frac{N!(\alpha_{OS} + N - k)!}{(N-k)!(\alpha_{OS} + N)!} \qquad (\alpha_{OS} \quad \text{integer}) \tag{2.24}$$

P_D can also be calculated from (2.23) by letting, [6]

$$\alpha_{OS}^{D} = \frac{\alpha_{OS}}{1 + \chi}$$

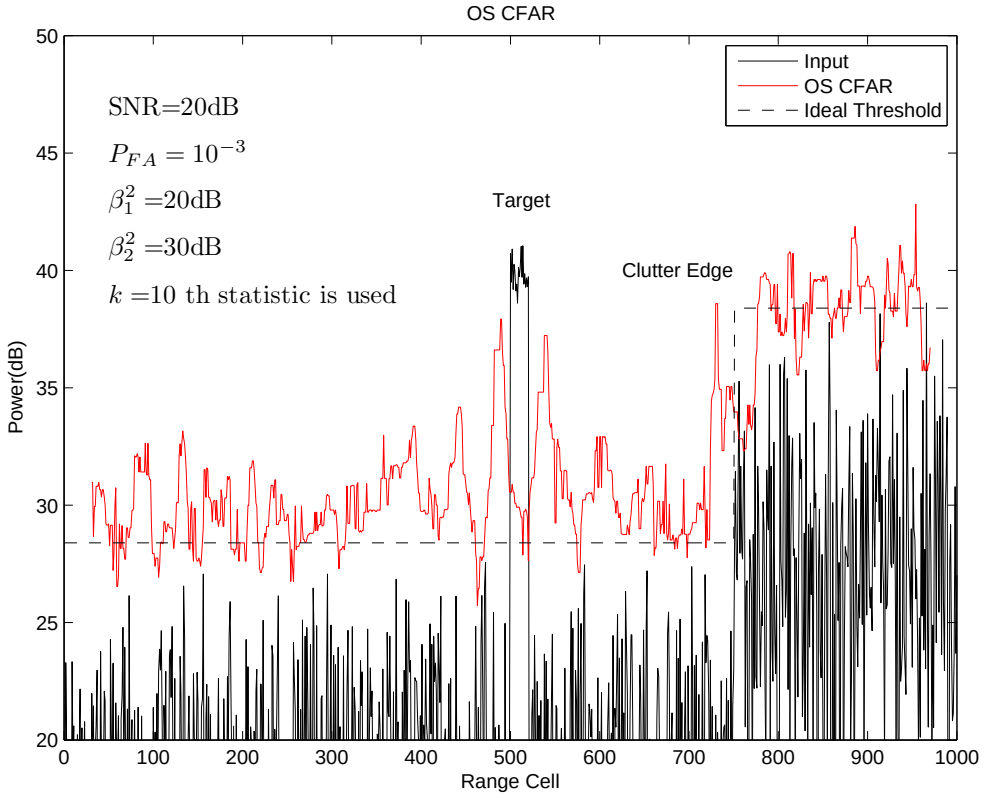

Figure 2.7: OS CFAR threshold behavior and clutter edge performance, k=10th statistic is used

Fig(2.7) shows the operation of OS CFAR and its clutter edge performance when $N = 20$ and $k = 10$. There is small increase in the threshold after the reference cell has passed the target window, the reason for this is the target is long and occupies more than $N - k$ reference cells. However, this increase is not very large and can be tolerated. Guard cells are less important in OS CFAR, since the rank ordering process will not be affected by targets outside the test cell. The clutter edge performance of OS CFAR is given in [14]. Typically k is on the order of $3N/4$ [12]. OS CFAR method is completely insensitive to masking of closely spaced targets, however it requires the implementation of a sorting logic and therefore the computational load for this algorithm is high. OS CFAR losses are lower than CA CFAR losses in the case of interfering targets [15]. Some additional results on the performance of OS CFAR in non homogeneous clutter and Weibull clutter are available in [16].

2.2.6 Variability Index CFAR

The Variability Index CFAR (VI CFAR) utilizes a background estimation algorithm which is a mixture of the CA CFAR, SOCA CFAR and GOCA CFAR. VI CFAR uses a statistical test to dynamically select the particular group of reference cells to use to estimate the threshold. This particular group can be; leading half of reference cells, lagging half of reference cells, or all of the reference cells. This dynamical selection provides the VI CFAR algorithm with low loss CFAR performance in a homogeneous environment and also robust performance in non homogeneous environments filled with multiple targets and clutter edges. ([17], [18])

The block diagram of VI CFAR can be seen in Fig(2.8). The VI is a second-order statistic closely related to an estimate of the shape parameter of a probability distribution. It closely resembles the coefficient of variation. It is defined as

$$\mathrm{VI} = 1 + \frac{\hat{\sigma}^2}{\hat{\mu}^2} \tag{2.25}$$

where $\hat{\mu}$ is the estimated mean and $\hat{\sigma}^2$ is the estimated variance of the split window. Then, the VI is calculated for each split window and compared with a threshold K_{VI} to decide if the cells with which the VI is computed are from a homogeneous or non homogeneous environment. The following logical test is used

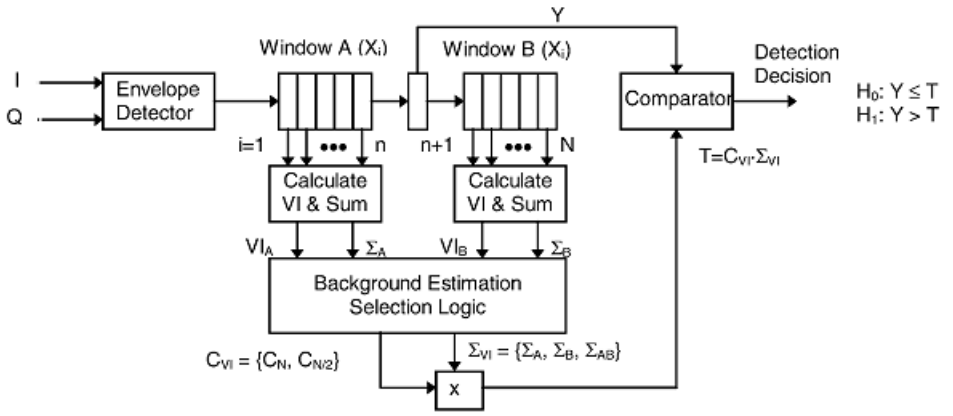

Figure 2.8: VI CFAR Block Diagram

$$\begin{aligned} \text{VI} \leq K_{\text{VI}} &\Longrightarrow \text{Nonvariable} \\ \text{VI} > K_{\text{VI}} &\Longrightarrow \text{variable} \end{aligned} \tag{2.26}$$

The mean ratio (MR) is defined as the ratio of the estimated mean values of the leading and lagging reference window cells.

$$\text{MR} = \frac{\bar{X}_A}{\bar{X}_B} = \sum_{i \in A} X_i / \sum_{i \in B} X_i \tag{2.27}$$

The MR is compared with a threshold K_{MR} to decide if the populations have the same means or different means. The following logical test is used

$$\begin{aligned} \text{MR} < {K_{\text{MR}}}^{-1} \quad \text{or} \quad \text{MR} > K_{\text{MR}} &\Longrightarrow \text{Different means} \\ \text{Otherwise} &\Longrightarrow \text{Same means} \end{aligned} \tag{2.28}$$

After the VI and MR has been calculated for each split reference window, Table (2.1) is used to set the adaptive threshold.

The VI CFAR is less complex than the OS CFAR and offers better P_{FA} performance in a

Leading Window Variable?	Lagging Window Variable?	Different Means?	VI-CFAR Threshold	Equivalent CFAR method
No	No	No	$C_N.\Sigma_{AB}$	CA CFAR
No	No	Yes	$C_{N/2}.\max(\Sigma_A, \Sigma_B)$	GOCA CFAR
Yes	No	-	$C_{N/2}.\Sigma_B$	CA CFAR
No	Yes	-	$C_{N/2}.\Sigma_A$	CA CFAR
Yes	Yes	-	$C_{N/2}.\min(\Sigma_A, \Sigma_B)$	SOCA CFAR

Table 2.1: VI CFAR adaptive threshold selection

clutter edge environment and similar P_D performance in a multiple target environment. The selection of the parameters K_{MR} and K_{VI} and performance results of the VI CFAR algorithm can be found in [18].

2.2.7 Switching CFAR

A threshold is set using the magnitude of the cell under test. Then, the cells in the reference window are put into two groups, those above the said threshold and those below them. If the number of cells that are below the threshold is larger than some N_t, all N cells are used in a cell averaging calculation. If the number of low amplitude cells is less than N_t, the threshold is set using only the low amplitude cells. This method causes reduced CFAR losses compared to other methods and somewhat improved clutter-edge performance. The need for sorting is avoided. The clutter edge performance is not so good, it is worse than VI-CFAR in terms of false alarms in clutter edges. Equations for P_D and P_{FA} can be found in [19].

2.2.8 Log CFAR

To combat target masking effects, one can use a log detector (instead of a linear or square law detector) and then can apply the conventional CA CFAR methods to the logarithm of the samples. There is no simple closed form for determining the relationship between an average of the log-detected data and the interference power β^2. However, motivated by the form of (2.3) one can write

$$\hat{T}_{\log} = \frac{1}{N} \sum_{i=1}^{N} 10 \log_{10}(x_i) + \alpha_{\log} \tag{2.29}$$

where $\alpha_{\log}$ is the threshold factor to ensure the designed false alarm probability. The log CFAR has the ability to operate over a large dynamic range. Also it is less vulnerable to target masking effects. Interfering targets in the reference window cannot raise the threshold too greatly because of the log operator. Unfortunately, log CFAR shows poor performance at clutter edges and has an increased vulnerability to false alarms near clutter edges.

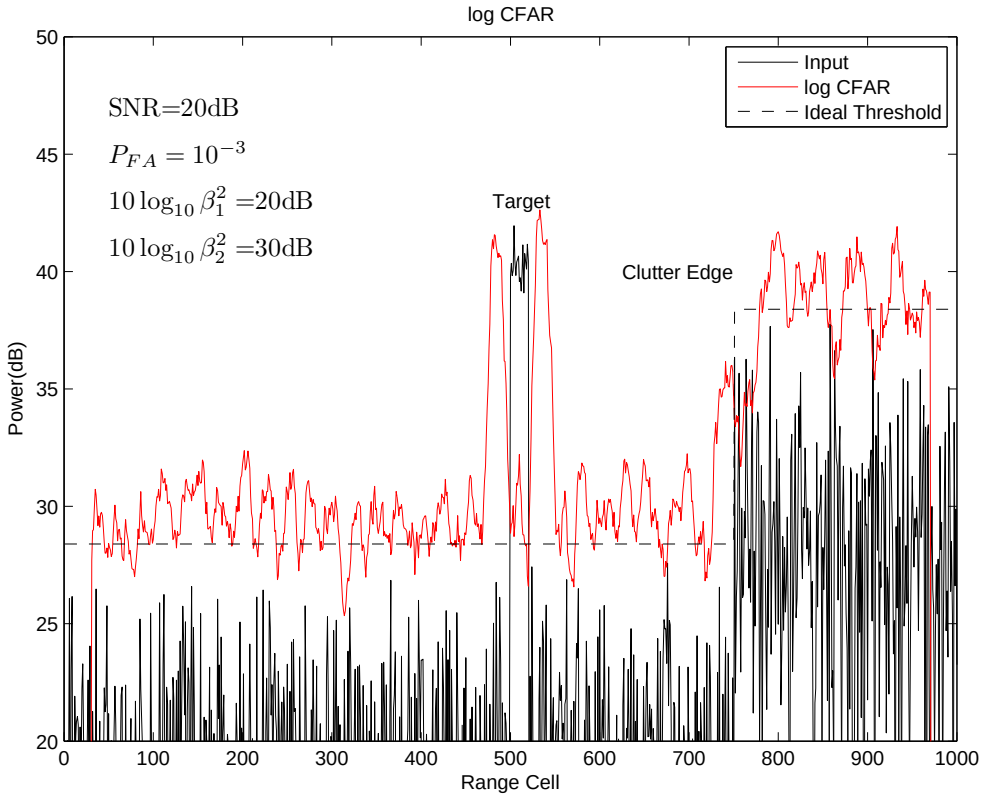

Figure 2.9: log CFAR threshold behavior and clutter edge performance

Fig.(2.9) shows the thresholds estimated by log CFAR method. Here the multiplier $\alpha_{\log}$ is found from [4] and [20]. $\alpha_{\log}$ = 11.85 dB for N = 20 reference cells and P_{FA} = 10^{-3}. The increase in the threshold around the vicinity of the target still exists, but compared to the performance of CA CFAR or GOCA CFAR, this increase is more tolerable. This allows for easy detection of closely spaced targets and prevention of masking effects. The clutter edge performance of log CFAR is seen to be poor, because false alarms can occur at clutter edges. The CFAR loss of log CFAR is larger than that of CA CFAR and the use of log detector increases the required CFAR window size by about 65 percent [21].

2.2.9 Adaptive CFAR

These type of algorithms are constructed to increase CFAR performance in non homogeneous clutter. Typically, a statistical test is made to determine if the reference cells span one or two clutter fields. This information is then used to select if SOCA, GOCA or CA CFAR approaches will be used. A basic approach is described by [22]. This approach first assumes that the reference cells span two clutter fields and tries to find the point at which the statistics change, M, where $M < N$. M is found by maximizing the following equation for all possible M where $M \in (1 : N)$

$$L_M = -[M \ln \widehat{\beta^2_{1(M)}} - (N - M) \ln \widehat{\beta^2_{2(M)}}] \tag{2.30}$$

where $\widehat{\beta^2_{1(M)}}$ is the sample mean in the first clutter region where M is assumed to be the transition point and $\widehat{\beta^2_{2(M)}}$ is the sample mean in the second region spanning the remaining $N - M$ cells. Once this M is identified, the cell under test is also known to be in the first or second clutter region. If the cell under test is in the lower clutter region, the threshold is set using cells in the lower clutter region only. If the cell under test is in the higher clutter region, the threshold is set using cells in the higher clutter region. This prevents excessive false alarms caused by comparing the cell to the threshold estimated using wrong part of the clutter region.

The procedure above does not account for the possibility that the clutter is uniform. To account for this possibility, another statistical test can be conducted. Assume $M = 0$ and calculate the likelihood shown in (2.30), $L_0 = -N \ln \hat{\beta}^2$, and compare this to the likelihood computed before. If $L_0 > L_M$ the clutter is assumed to be uniform and standard CA CFAR procedure is applied. The main advantage of this method is that it has great performance at clutter edges (optimal in Gaussian clutter, [22]). However, it requires additional operations to determine the changing point M, therefore increasing the required computational power. Also if M is estimated incorrectly, P_{FA} increases. Fig.(2.10) shows the operation of this method. The exceptional clutter edge performance should be noted. The detection performance is similar to that of CA CFAR.

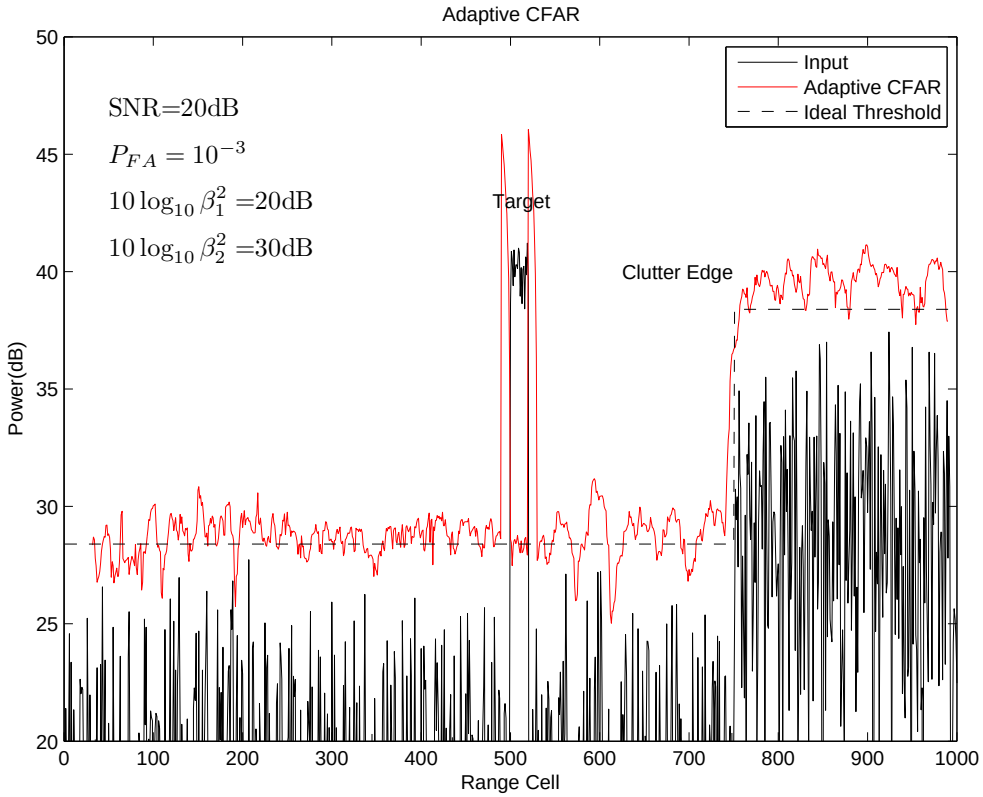

Figure 2.10: Adaptive CFAR threshold behaviour and clutter edge performance

2.2.10 Clutter Map CFAR

Clutter mapping is a technique used for detection of slowly moving or stationary targets when the Doppler shift of the target is very close to zero. The threshold for each range cell is computed as a multiple of the measured clutter in that same cell. The clutter measurement is updated as follows:

$$y[n] = \gamma y[n-1] + (1-\gamma)x[n] \tag{2.31}$$

where $y[n]$ is the estimated clutter reflectivity and $x[n]$ is the currently measured clutter sample, both at time n. The factor γ controls the weight of the current measured sample compared to the previous measured samples. This factor can be changed on real time if necessary. The threshold is then set as

$$T[n] = \alpha y[n-1] \tag{2.32}$$

It can be seen from (2.32) that the threshold $T[n]$ depends on the samples from the previous scan. The reason the current data are not included is, if the current data contains a target, the clutter measurement would be distorted and the threshold would be raised too high, therefore creating a self masking effect. The first order difference equation (2.31) corresponds to an IIR filter and the output $y[n]$ can be written as

$$y[n] = (1-\gamma)\sum_{m=0}^{\infty} \gamma^m x[n-m] \tag{2.33}$$

Average P_{FA} and P_D equations are derived in [23] for a clutter map CFAR approach using multiple scans. The basic CA and OS approaches can be combined with clutter mapping techniques to fit the operational needs ([24],[25]).

2.2.11 Other CFAR types

There are other CFAR types such as the two-parameter CFAR which works for log-normal or Weibull pdf ([26], [27] and [28]). The distribution free CFAR [29] is a method in which no specific form of the pdf of the interference is assumed. For a system operating in a clutter limited environment, a threshold setting algorithm based on a particular pdf may produce very large errors when actually another interference pdf is present. This is why distribution free CFAR methods are important. Also, the combination of various types of CFAR methods is possible and are presented in literature frequently. ([30], [31], [32], [33] and [34]).

CHAPTER 3

CFAR DETECTION USING IIR FILTERS

Motivated by the fact that all CFAR processors use some kind of averaging and windowing, one can propose a structure similar to that of Clutter Map CFAR. Instead of using a simple averaging FIR filter, we can use a weighted IIR filter to calculate a weighted moving average which is still a good estimate of the noise background. The ”exponential smoothing filter” will be used here which is shown in (2.31) and (2.32). This is different from Clutter Map CFAR shown in 2.2.10. In clutter map CFAR, the threshold estimate is formed from different scans of the same range cell. In the IIR filter approach, $x[n]$ represents the cells in time (or range) but from the same scan. The detector structure using the IIR filter is shown in Fig.(3.1). Here the threshold is estimated by filtering the range cells using an IIR filter and then threshold comparison is made.

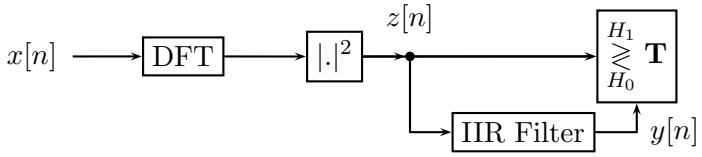

Figure 3.1: The IIR Filter Detector

Here the structure of the IIR filter is:

$$y[n] = \gamma y[n-1] + (1-\gamma)z[n]$$
$$y[n] = (1-\gamma)\sum_{m=0}^{\infty} \gamma^m z[n-m]$$
$$T[n] = \alpha y[n-1]$$

where $T[n]$ is the estimated threshold at time n and $z[n]$ is the magnitude squared DFT bin at time n. Figure 3.2 shows a block diagram implementation of such a system. This system is named as IIR CFAR method.

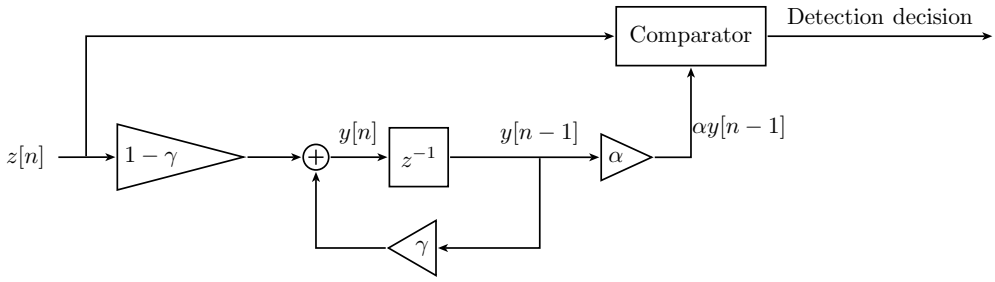

Figure 3.2: Detection using an IIR Filter

To find the parameter γ, we make the following analysis. We are trying to estimate the parameter β^2 and we know that the optimum estimator [35] (for the homogeneous case, Gaussian interference and signal-free reference cells) is the average, $\frac{1}{N}\sum_{i=1}^{N} z_i$. Let us call the optimum estimator as estimator 1, or E_1 and let the IIR filter estimator be E_2. We will try to select appropriate γ so that these two estimators achieve the same (or close) results. Let mean(z) = μ_z and var(z) = σ_z^2 (both can be found from properties of DFT operator but has no significance here).

$$\text{mean}(E_1) = \text{mean}(\frac{1}{N}\sum_{i=1}^{N} z_i) = \frac{1}{N}\sum_{i=1}^{N}\text{mean}(z_i) = \mu_z$$

$$\text{mean}(E_2) = \text{mean}\Big\{(1-\gamma)\sum_{m=0}^{\infty}\gamma^m z[n-m]\Big\}$$

$$= (1-\gamma)\sum_{m=0}^{\infty}\gamma^m\mu_z = \mu_z$$

Both E_1 and E_2 are unbiased estimators, independent of γ. Let us look at the variances of E_1 and E_2.

$$\text{var}(E_1) = \text{var}(\frac{1}{N}\sum_{i=1}^{N} z_i) = \frac{1}{N^2}\text{var}(\sum_{i=1}^{N} z_i) = \frac{1}{N^2}\sum_{i=1}^{N}\sigma_z^2 = \frac{1}{N}\sigma_z^2 \tag{3.1}$$

$$\text{var}(E_2) = \text{var}\Big\{(1-\gamma)\sum_{m=0}^{\infty}\gamma^m z[n-m]\Big\}$$

$$= (1-\gamma)^2\text{var}\Big\{\sum_{m=0}^{\infty}\gamma^m z[n-m]\Big\}$$

$$= (1-\gamma)^2\sum_{m=0}^{\infty}\gamma^{2m}\sigma_z^2$$

$$= \sigma_z^2(1-\gamma)^2\sum_{m=0}^{\infty}\gamma^{2m}$$

$$= \sigma_z^2\frac{(1-\gamma)^2}{1-\gamma^2}$$

$$\text{var}(E_2) = \frac{1-\gamma}{1+\gamma}\sigma_z^2 \tag{3.2}$$

Equate $\text{var}(E_1)$ and $\text{var}(E_2)$ to obtain

$$\frac{1}{N} = \frac{1-\gamma}{1+\gamma}$$

leave γ alone to get

$$\gamma = \frac{N-1}{N+1} \tag{3.3}$$

(3.3) can be used to select appropriate γ to 'mimic' cell averaging behavior of N cells. Since this filter is meant to average out the samples, the threshold multiplier α can be set the same

way as in CA CFAR. However, from the literature it can be seen that $\alpha_{iir} \neq \alpha_{ca}$. The threshold multiplier can be found from the following formulas from [23].

$$P_{FA} = \frac{1}{\prod_{m=0}^{M} [1 + \alpha(1-\gamma)\gamma^m]} \qquad M \to \infty \tag{3.4}$$

$$P_D = \frac{1}{\prod_{m=0}^{M} [1 + \alpha_D(1-\gamma)\gamma^m]} \qquad M \to \infty \tag{3.5}$$

where

$$\alpha_D = \frac{\alpha}{1+\chi} \tag{3.6}$$

From [36] a faster converging equation can be found.

$$P_{FA} = \frac{1}{1 + \sum_{m=0}^{M} \prod_{k=0}^{m} \frac{\alpha(1-\gamma)\gamma^k}{1-\gamma^{k+1}}} \qquad M \to \infty \tag{3.7}$$

P_D can be found by replacing α with α_D as in (3.6). If one chooses γ as in (3.3) and compares the CA CFAR performance with the IIR Filter performance, we get interesting results. Let us plot P_D vs SNR curves for CA CFAR and IIR Filter CFAR from the equations (3.7) and (2.9). The plot is shown in Fig.(3.3).

As seen from Fig.(3.3) the performances of CA CFAR and IIR CFAR are nearly identical for different values of N. This motivates one to use the IIR filter because it is much simpler than CA CFAR (a one tap filter) and performs the same in homogeneous environment. Let us see how as $N \to \infty$ the performance of IIR CFAR changes. Since $\gamma = \frac{N-1}{N+1}$, $\gamma \to 1$ as $N \to \infty$. Let us evaluate this limit and plug it in (3.7).

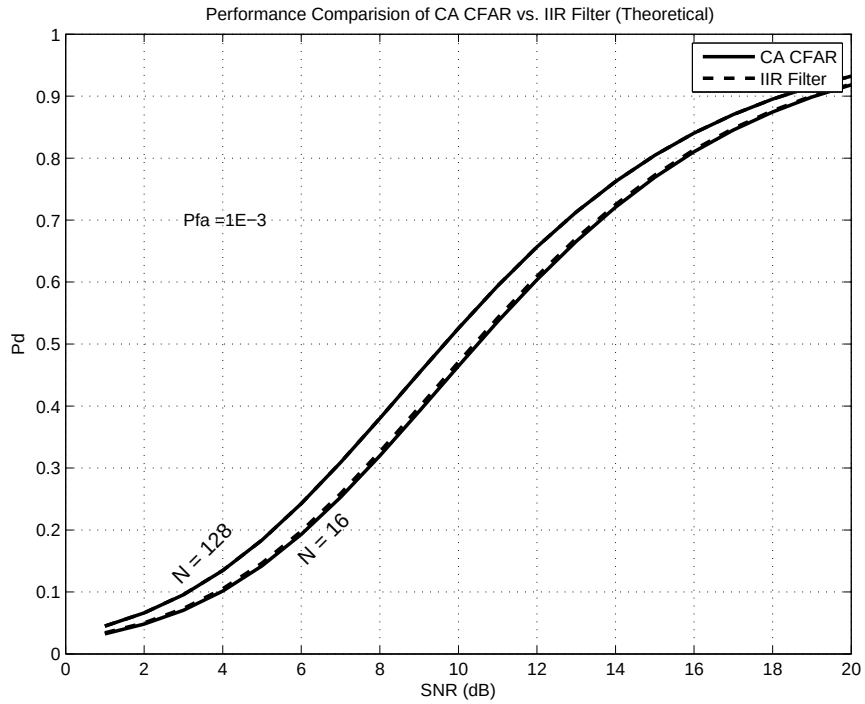

Figure 3.3: P_D vs SNR curves for CA CFAR and IIR CFAR for different N values and equivalent γ parameters

$$\begin{aligned}
\lim_{\gamma\to 1}\frac{\alpha(1-\gamma)\gamma^k}{1-\gamma^{k+1}} &= \alpha\lim_{\gamma\to 1}\frac{\gamma^k-\gamma^{k+1}}{1-\gamma^{k+1}} = \frac{0}{0}, \qquad \text{use L'Hôpital rule}\\
&= \alpha\lim_{\gamma\to 1}\frac{k\gamma^{k-1}-(k+1)\gamma^k}{-(k+1)\gamma^k}\\
&= \alpha\lim_{\gamma\to 1}\frac{\gamma^k(k\gamma^{-1}-k-1)}{-\gamma^k(k+1)}\\
&= \frac{\alpha}{k+1} \qquad (3.8)
\end{aligned}$$

Inserting (3.8) into (3.7).

$$P_{FA} = \lim_{M\to\infty} \frac{1}{1 + \sum_{m=0}^{M} \prod_{k=0}^{m} \frac{\alpha}{k+1}}$$
$$P_{FA} = \lim_{M\to\infty} \frac{1}{1 + \sum_{m=0}^{M} \frac{\alpha^{m+1}}{(m+1)!}}$$
$$P_{FA} = \lim_{M\to\infty} \frac{1}{\sum_{m=1}^{M} \frac{\alpha^{m}}{m!}}$$
$$P_{FA} = \frac{1}{\sum_{m=1}^{\infty} \frac{\alpha^{m}}{m!}} = e^{-\alpha} \tag{3.9}$$

Similarly $P_D = e^{-\alpha/1+\chi}$ and $P_D = {P_{FA}}^{1/1+\chi}$ can then be reached, which is the well known result for the non-CFAR case. This explains us the theoretical reasoning behind Fig.(3.3) since we see that both CA CFAR and IIR CFAR converge to the same optimal detection characteristic as N is selected arbitrarily large. From Fig.(3.3) we see that N does not need to be so large, even setting N=16 provides visually the same performance with CA CFAR. Even though the performance of IIR CFAR looks promising under homogeneous conditions, examining its threshold behavior will reveal that its performance degrades seriously under non homogeneous operating conditions.

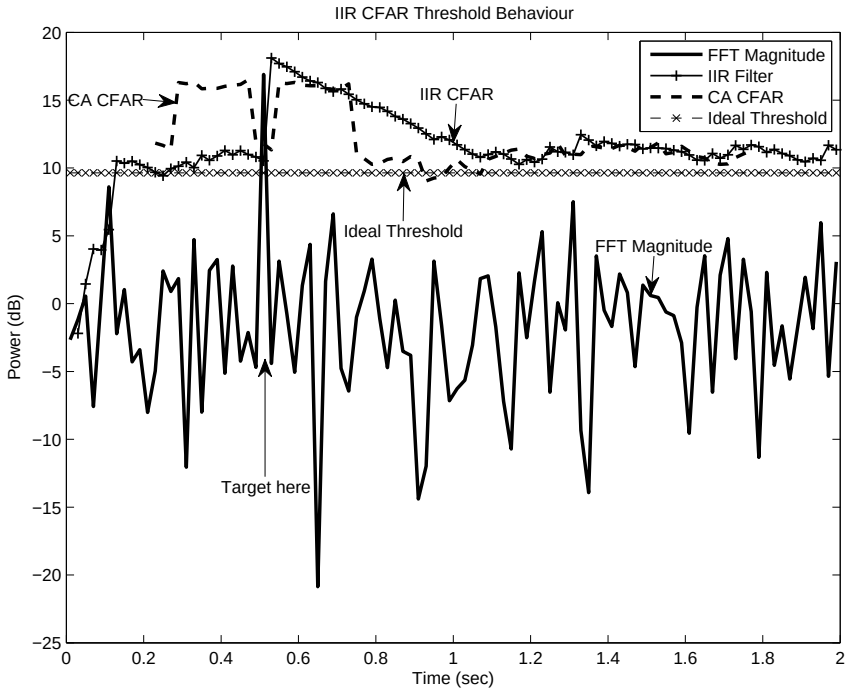

Figure 3.4: Threshold behaviour of IIR CFAR

It is seen from Fig.(3.4) that the target at 0.5 second causes the IIR CFAR threshold to go unnecessarily high and it takes a very long time for this threshold to settle to its normal value. This problem also exists in the CA CFAR method, for the next N samples after a target, the threshold remains unnecessarily high. A very simple solution to this problem is proposed in [23]. Here the threshold is only updated if a target absent decision is made. If a target present decision is made the threshold is not updated and the same threshold is used. Even though this can solve detection problems in a multiple target environment, in a clutter edge filled environment this algorithm collapses as seen in Fig(3.5).

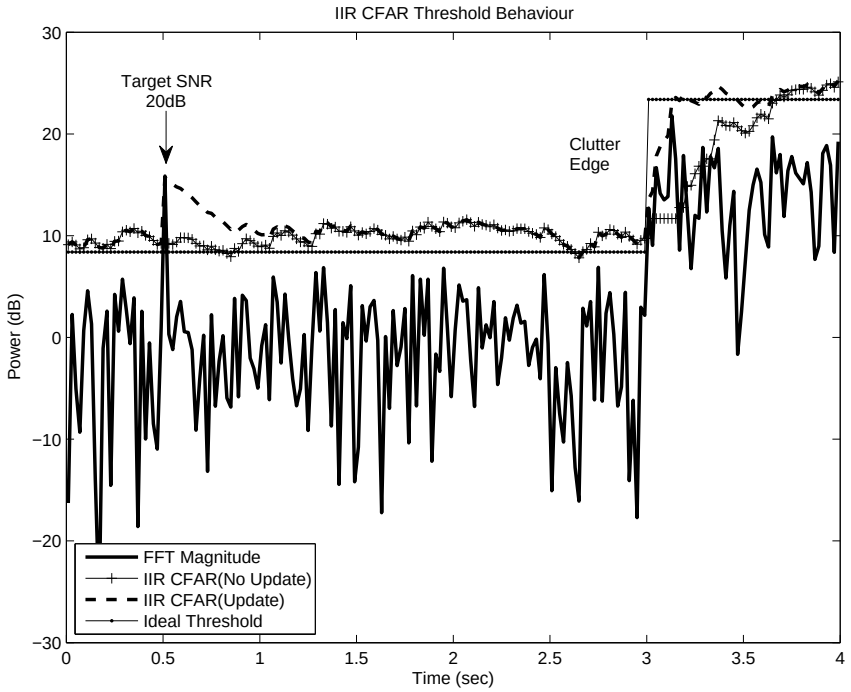

Figure 3.5: Threshold behavior of IIR CFAR under a clutter edge

In Fig.(3.5) we see the advantage and failing point of this partial solution. ”The IIR CFAR (Update)” curve shows the IIR CFAR with the threshold updated at every sample not regarding whether there is a target or not. The ”IIR CFAR (Not Update)” curve corresponds to the solution mentioned in [23] where the threshold is not updated in the presence of a target. The target at 0.5 seconds is successfully detected and declared as a target because the threshold is not updated and the target is not included in the background estimate. At the 3rd second, however, there is a clutter edge and the threshold actually needs to be updated to avoid exces-

sive false alarms. The excessive false alarms of the ”IIR CFAR (Not Update)” algorithm can easily be seen after the 3rd second.

3.1 Proposed Algorithm

Problem: Modify the IIR CFAR algorithm so that it performs robustly under clutter edges.

Observation: Two filters with different rate parameters perform differently under clutter edges as seen in Figure 3.6.

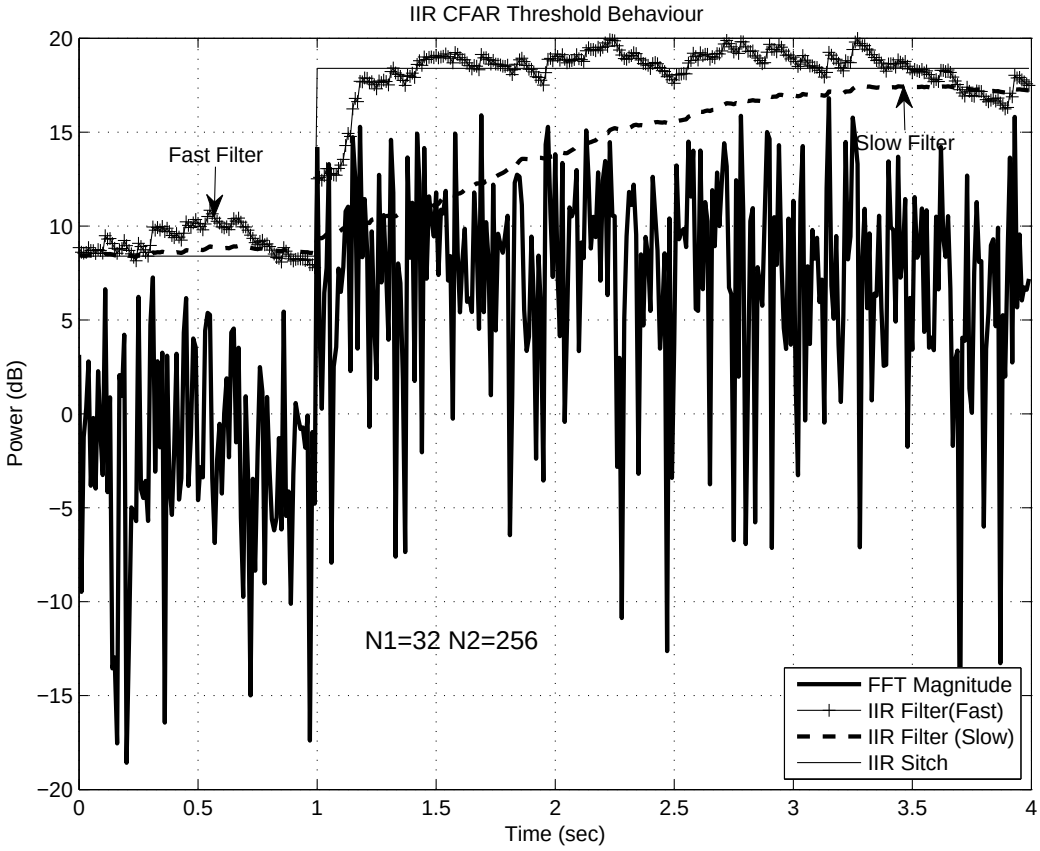

Figure 3.6: Different behaviors of two filters under a clutter edge

Solution: Use two IIR filters with different rate parameters γ_1 and γ_2 which represent different number of cells averaged N_1 and N_2. As shown before, $\gamma_1 = \frac{N_1-1}{N_1+1}$ and $\gamma_2 = \frac{N_2-1}{N_2+1}$. Select

N_1 and N_2 ($N_2 > N_1$) such that there is at least one order of magnitude between them for example, $N_1 = 32$ and $N_2 = 256$ is a good choice. From now on the filter with the parameter γ_1 (and hence N_1) will be referred as the "fast filter" because it responds to changes faster than the other filter. Similarly, the filter with the parameter γ_2 (and hence N_2) will be referred as the "slow filter". When the environment is homogeneous, it is desired to use the slow filter because it has better detection performance as seen in Figure 3.3. However, in a non homogeneous environment (such as one with clutter edges) the fast filter can adapt better to the non homogeneity and therefore performs better, so it is desired to use the fast filter.

To evaluate if the environment is homogeneous or not, we will use the difference of the magnitudes between the two filter. Under a homogeneous environment, we expect both y_1 and y_2 to perform similarly and their values to be close to each other. However in a non homogeneous environment the outputs of these two filters can be different by a margin. This difference will be used in detecting multiple targets and/or clutter edges. If a target or clutter edge is detected, it is reasonable to use the fast filter (with smaller time constant or smaller N) because that way the threshold will quickly adapt to the non homogeneity. In a homogeneous environment, on the other hand, it is advisable to use the slow filter (with larger time constant or larger N) because that way the CFAR losses are low and detection performance is high. The aim is to 'switch' amongst two filters in a non homogeneous environment. This algorithm will be called Switching IIR CFAR (SIIR CFAR). We propose a measure of switching called *False Switching Probability* (P_{FS}) just like false alarm probability or clean sample rejection rate as in [37] and [38]. P_{FS} is defined as

$$P_{FS} = P\{|y_1 - y_2| > T; |\text{Homogeneous Environment}\} \tag{3.10}$$

Obviously, when P_{FS} is increased, more switching decisions will happen. This means for larger P_{FS}, more switching decisions are made and therefore the fast filter is used more frequently and the performance is expected to be closer to that of fast filter. On the other hand, for smaller P_{FS}, the performance is expected to be closer to the slow filter. To find T, the analytical pdfs of y_1 and y_2 are required but to the best of our knowledge, closed form expressions are not available [39].

3.2 Calculation of T_S

For large N_1 and N_2 the random variates y_1 and y_2 can be modeled as $\mathcal{N}(0, {\sigma_{y_1}}^2)$ and $\mathcal{N}(0, {\sigma_{y_2}}^2)$ as mentioned in [39]. However the Gaussian approximation does not work so well for $y_1 - y_2$ as seen by the simulation results, so some error correction term is needed. The error correction is performed as described in [40] pg. 217, using *Hermite polynomials*. We will use first order correction as shown in 3.11.

$$f_x(x) \approx \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2}\left[1 + \frac{m_3}{6\sigma^3}\left(\frac{x^3}{\sigma^3} - \frac{3x}{\sigma}\right)\right] \tag{3.11}$$

where $x = y_1 - y_2$ and $\sigma = \text{var}\{x\}$. To make this error correction we need moments of $y_1 - y_2$,namely we need $m_n = E\{(y_1 - y_2)^n\}$. Here the calculation of such moments will be explained.

First, let us find the cross-correlation coefficient ρ which is defined as

$$\rho = \frac{E\{(y_1 - \mu_1)(y_2 - \mu_2)\}}{\sigma_1\sigma_2} \tag{3.12}$$

where $\mu_1 = E\{y_1\}$ and $\mu_2 = E\{y_2\}$. Let us evaluate (3.12).

$$\begin{aligned}
y_1[n] &= (1 - \gamma_1)\sum_{m=0}^{\infty} {\gamma_1}^m z[n - m] \\
y_2[n] &= (1 - \gamma_2)\sum_{m=0}^{\infty} {\gamma_2}^m z[n - m] \\
y_1[n] - \mu_1 &= (1 - \gamma_1)\sum_{m=0}^{\infty} {\gamma_1}^m (z[n - m] - \mu_z) \\
y_2[n] - \mu_2 &= (1 - \gamma_2)\sum_{m=0}^{\infty} {\gamma_2}^m (z[n - m] - \mu_z)
\end{aligned}$$

Now let $u[m] = z[n - m] - \mu_z$. So now,

$$y_1[n] - \mu_1 = (1-\gamma_1)\sum_{m=0}^{\infty} {\gamma_1}^m u[m]$$
$$y_2[n] - \mu_2 = (1-\gamma_2)\sum_{m=0}^{\infty} {\gamma_2}^m u[m]$$

Using $u[n]$ we have,

$$\begin{aligned} E\{(y_1[n]-\mu_1)(y_2[n]-\mu_2)\} &= E\Big\{(1-\gamma_1)(1-\gamma_2)\sum_{m=0}^{\infty}{\gamma_1}^m u[m]\sum_{m=0}^{\infty}{\gamma_2}^m u[m]\Big\} \\ &= (1-\gamma_1)(1-\gamma_2)E\Big\{\sum_{m=0}^{\infty}{\gamma_1}^m u[m]\sum_{m=0}^{\infty}{\gamma_2}^m u[m]\Big\} \end{aligned} \tag{3.13}$$

$u[m]$ is an independent sequence since $z[m]$ is. Therefore,

$$\begin{aligned} E\{(y_1[n]-\mu_1)(y_2[n]-\mu_2)\} &= (1-\gamma_1)(1-\gamma_2)E\Big\{\sum_{m=0}^{\infty}\sum_{k=0}^{m}{\gamma_1}^k u[k]{\gamma_2}^{m-k}u[m-k]\Big\} \\ &= (1-\gamma_1)(1-\gamma_2)\sum_{m=0}^{\infty}{\gamma_1}^m{\gamma_2}^m E\{u[m]^2\} \end{aligned} \tag{3.14}$$

since $E\{u[m]^2\} = E\{(z[n-m]-\mu_z)^2\} = {\mu_z}^2$. Evaluating the infinite sum in (3.14) we get

$$E\{(y_1[n]-\mu_1)(y_2[n]-\mu_2)\} = \frac{(1-\gamma_1)(1-\gamma_2)}{1-\gamma_1\gamma_2}{\mu_z}^2 \tag{3.15}$$

We also know that

$$\begin{aligned} \sigma_1 &= \sqrt{\text{var}\{y_1\}} = \sqrt{\sigma_z^2/N_1} \\ \sigma_2 &= \sqrt{\text{var}\{y_2\}} = \sqrt{\sigma_z^2/N_2} \\ \sigma_1\sigma_2 &= \frac{{\sigma_z}^2}{\sqrt{N_1N_2}} \end{aligned} \tag{3.16}$$

Using (3.15) and (3.16) in (3.12) we have

$$\rho = \sqrt{N_1 N_2}\frac{(1-\gamma_1)(1-\gamma_2)}{1-\gamma_1\gamma_2} \tag{3.17}$$

Note that the term $\frac{{\mu_z}^2}{{\sigma_z}^2}$ term cancels since z is an exponential variate, ${\mu_z}^2 = {\sigma_z}^2$. Also, if we substitute γ_1 and γ_2 in terms of N_1 and N_2 we get a compact expression.

$$\rho = 2\frac{\sqrt{N_1 N_2}}{N_1 + N_2} \tag{3.18}$$

Now, let us start with an exponential random variable z and evaluate $E\{z^n\}$ using the definition of the moment generating function, $m_Z(t) = E\{e^{tZ}\}$.

$$\begin{aligned} E\{z^n\} &= \frac{d^n m_z}{dt^n} \\ E\{z^n\} &= \frac{d^n}{dt^n}(1-\mu t)^{-1} \quad \text{differentiating } n \text{ times gives} \\ E\{z^n\} &= n!\mu^n \end{aligned} \tag{3.19}$$

where in (3.19), $\mu = E\{z\}$ and is the parameter of the exponential distribution. Now, let us calculate the n^{th} expectation of the same random variable with mean equated to zero, $E\{(z-\mu)^n\}$. Using the binomial theorem from [41].

$$\begin{aligned} E\{(z-\mu)^n\} &= E\left\{\sum_{k=0}^{n}(-1)^k\binom{n}{k}z^{n-k}\mu^k\right\} \\ &= \sum_{k=0}^{n}(-1)^k\binom{n}{k}E\{z^{n-k}\}\mu^k \end{aligned} \tag{3.20}$$

Use (3.19) in (3.20) to get

$$E\{(z-\mu)^n\} = \sum_{k=0}^{n}(-1)^k\binom{n}{k}(n-k)!\mu^{n-k}\mu^k$$
$$E\{(z-\mu)^n\} = n!\mu^n\sum_{k=0}^{n}\frac{(-1)^k}{k!} \tag{3.21}$$

Now we shall calculate $m_n = E\{(y_1 - y_2)^n\}$ using these equations. However, one more definition simplifies these calculations. Let $w_1 = y_1 - \mu$ and $w_2 = y_2 - \mu$, so that w_i's are zero mean random variables. Also note that,

$$w_1 = (1-\gamma_1)\sum_{m=0}^{\infty}{\gamma_1}^m u[m]$$
$$w_2 = (1-\gamma_2)\sum_{m=0}^{\infty}{\gamma_2}^m u[m]$$

where $u[m] = z - \mu$ as defined before. Let us evaluate $E\{w_1^p w_2^q\}$ for $p, q > 0$.

$$\begin{aligned}E\{w_1^p w_2^q\} &= E\left\{\left((1-\gamma_1)\sum_{m=0}^{\infty}{\gamma_1}^m u[m]\right)^p\left((1-\gamma_2)\sum_{m=0}^{\infty}{\gamma_2}^m u[m]\right)^q\right\}\\ &= (1-\gamma_1)^p(1-\gamma_2)^q E\left\{\left(\sum_{m=0}^{\infty}{\gamma_1}^m u[m]\right)^p\left(\sum_{m=0}^{\infty}{\gamma_2}^m u[m]\right)^q\right\}\end{aligned} \tag{3.22}$$

Since $u[m]$'s are zero mean, the cross terms will be zero and (3.22) will be reduced to

$$\begin{aligned}E\{w_1^p w_2^q\} &= (1-\gamma_1)^p(1-\gamma_2)^q E\left\{\sum_{m=0}^{\infty}\gamma_1^{mp}\gamma_2^{mq}u[m]^{p+q}\right\}\\ &= (1-\gamma_1)^p(1-\gamma_2)^q\sum_{m=0}^{\infty}(\gamma_1^p\gamma_2^q)^m E\{u[m]^{p+q}\}\end{aligned} \tag{3.23}$$

Evaluating the infinite sum and using (3.21), (3.23) becomes

$$E\{w_1^p w_2^q\} = \frac{(1-\gamma_1)^p(1-\gamma_2)^q}{1-\gamma_1^p\gamma_2^q}\times(p+q)!\mu^{p+q}\sum_{j=0}^{p+q}\frac{(-1)^j}{j!} \tag{3.24}$$

Now we can easily calculate $m_n = E\{(y_1 - y_2)^n\}$ noting that,

$$E\{(y_1 - y_2)^n\} = E\{(w_1 - w_2)^n\}$$

Use the binomial expansion,

$$\begin{aligned} E\{(w_1 - w_2)^n\} &= E\left\{\sum_{k=0}^{n}(-1)^k\binom{n}{k}{w_1}^{n-k}{w_2}^k\right\} \\ &= \sum_{k=0}^{n}(-1)^k\binom{n}{k}E\{w_1^{n-k}w_2^k\} \end{aligned} \tag{3.25}$$

Use (3.24) in (3.25) to get

$$E\{(y_1 - y_2)^n\} = \sum_{k=0}^{n}(-1)^k\binom{n}{k}\frac{(1-\gamma_1)^{n-k}(1-\gamma_2)^k}{1-\gamma_1^{n-k}\gamma_2^k} \times n!\mu^n \sum_{j=0}^{n}\frac{(-1)^j}{j!} \tag{3.26}$$

Note that since some terms are omitted, the expression 3.26 is only valid for $n \leq 3$. For $n > 3$, the expression becomes difficult to evaluate. 3.26 will be used in approximating the output pdf of the random variable $y_1 - y_2$.

Using the approximate pdf in (3.11) the threshold T can be calculated by numerically solving the following equation

$$Q(T/\sigma) + e^{-AT^2}(k_1T^2 + \frac{k_1}{A} - k_2) = \frac{P_{FS}}{2} \tag{3.27}$$

where Q is the Gaussian Q function defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty}\exp\left(\frac{-u^2}{2}\right)du \tag{3.28}$$

and the constants k_1, k_2 and A are defined as

$$k_1 = \frac{m_3}{6\sigma^5\sqrt{2\pi}} \tag{3.29}$$

$$k_2 = \frac{m_3}{2\sigma^3\sqrt{2\pi}} \tag{3.30}$$

$$A = \frac{1}{2\sigma^2} \tag{3.31}$$

and m_3 is the 3rd moment of the random variable $y_1 - y_2$, ie: $m_3 = E\{(y_1 - y_2)^3\}$. To see if this approximation holds, we make the following test. For a desired P_{FS}, the threshold T is found from (3.27). The only parameter required for calculation of T is μ, the parameter of the exponential distribution. The achieved vs. desired P_{FS} values are obtained by Monte Carlo simulations and shown in the Figures 3.7 through 3.10 for different N_1 and N_2. Here, the ideal case refers to the case where Achieved P_{FS} = Desired P_{FS}. The curve 'Gaussian approximation' refers to the case where $T = \sigma Q^{-1}(P_{FS})$, the threshold is put assuming a Gaussian distribution for y_3. The 'corrected IIR' curve means the threshold is found by numerically solving (3.27) and μ is known. The 'corrected IIR spot' curve means that μ is not known but estimated using the slow filter (y_2) and $\sigma^2 = y_2$.

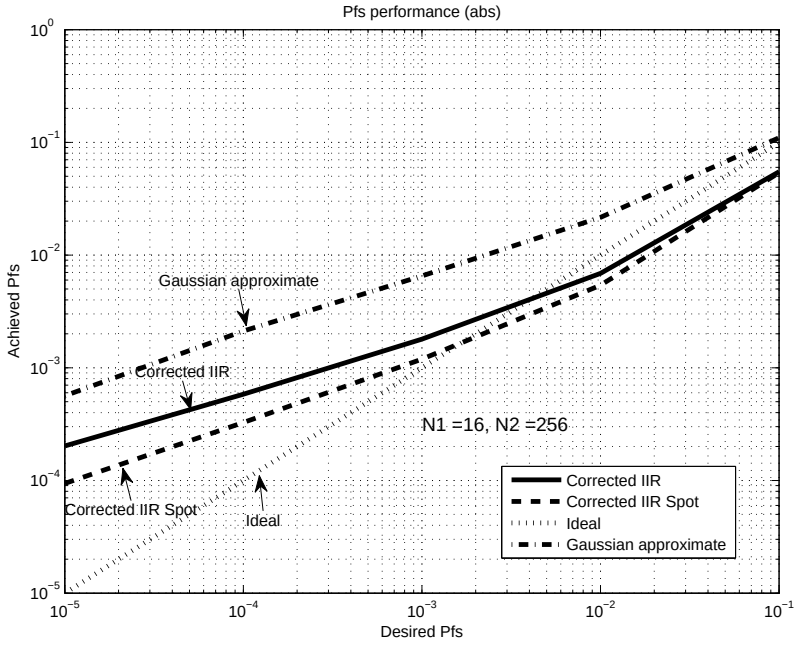

Figure 3.7: P_{FS} simulations for $N_1 = 16$ and $N_2 = 256$

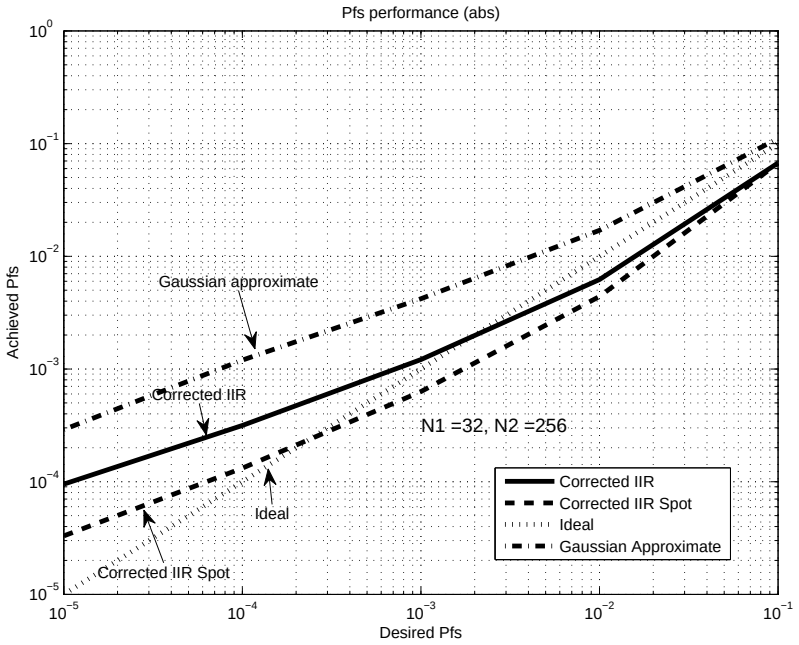

Figure 3.8: P_{FS} simulations for $N_1 = 32$ and $N_2 = 256$

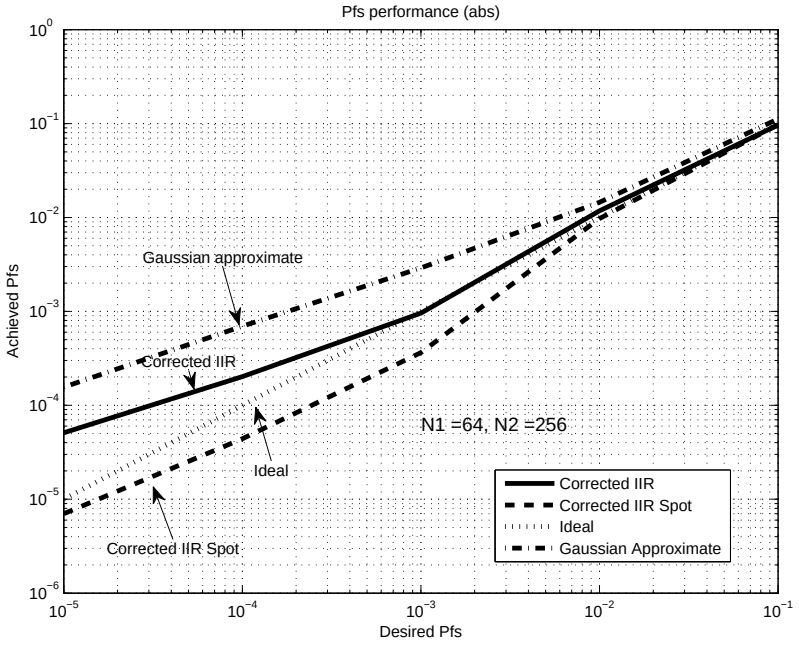

Figure 3.9: P_{FS} simulations for $N_1 = 64$ and $N_2 = 256$

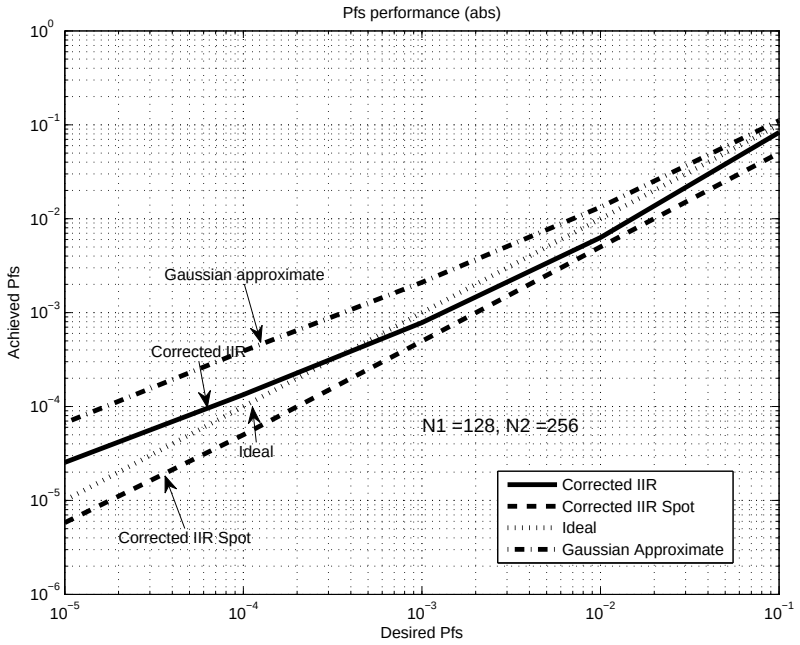

Figure 3.10: P_{FS} simulations for $N_1 = 128$ and $N_2 = 256$

Figures 3.7 through 3.10 show us that the additional terms to the pdf shown in (3.11) have improved our approximation of the original pdf. This also means that the threshold T for (3.10) can be set using (3.27) for a wide range of P_{FS} values and for different N_1 and N_2. What is interesting is that, for some situations the 'Corrected IIR Spot' curve is closer to the ideal curve than the 'Corrected IIR' curve. This is due to the fact that since y_2 is averaging a large number of samples, its estimate of the noise variance can be better than the theoretical variance itself, for that particular time instant. Also, numerically solving (3.27), one can see that there is a highly linear relationship between σ^2 and T for a given P_{FS}. Since we use $y_2 = \sigma^2$, this means a linear relationship can be used to determine the threshold from output of y_2 for a given P_{FS}. In other words $T[n] \simeq cy_2[n]$ where c is found from Table 3.1.

Now, the SIIR CFAR algorithm can be summarized as shown in Figure 3.11. Here $y_1[n]$ is the fast filter and $y_2[n]$ is the slow filter. (ie: $N_1 < N_2$). $z[n]$ is the magnitude squared input and $T[n]$ is the threshold at time n.

Table 3.1: The linear multiplier for $N_2 = 256$ and N_1 for {16,32,64,128} to achieve a desired P_{FS}

	P_{FS}				
N_1	1.00E-001	1.00E-002	1.00E-003	1.00E-004	1.00E-005
16	1.1289	1.0073	0.8652	0.6873	0.4253
32	0.7142	0.6351	0.5426	0.4270	0.2624
64	0.4053	0.3594	0.3057	0.2390	0.1470
128	0.1725	0.1526	0.1294	0.1007	0.0622

1. Initialize $y_1[0], y_2[0], T[0]$
2. For desired N_1 and N_2 compute γ_1 and γ_2 from (3.3)
3. For desired P_{FA} compute α_1 and α_2 from (3.7)
4. For desired P_{FS} compute c from Table 3.1
5. Update y_1 and y_2 as

   $y_1[n] = \gamma_1 y_1[n-1] + (1-\gamma_1)z[n]$

   $y_2[n] = \gamma_2 y_2[n-1] + (1-\gamma_2)z[n]$
6. **if** $|y_1[n-1] - y_2[n-1]| > c y_2[n-1]$
   - **then** $T[n] = \alpha_2 y_2[n]$
   - **else** $T[n] = \alpha_1 y_1[n]$
7. Detection decision is based on: $z[n] \underset{H_0}{\overset{H_1}{\gtrless}} T[n-1]$

Figure 3.11: The SIIR CFAR algorithm explained

A block diagram implementation of the switching algorithm can be seen in Figure 3.12. Both of the filters are running and the output is switched between two of them. This switching decision is based on $|y_1 - y_2|$ as explained before. This concludes the construction of the Switching IIR CFAR algorithm. Figure 3.13 shows the behavior of SIIR CFAR under a clutter edge. Simulation results are presented in the next chapter.

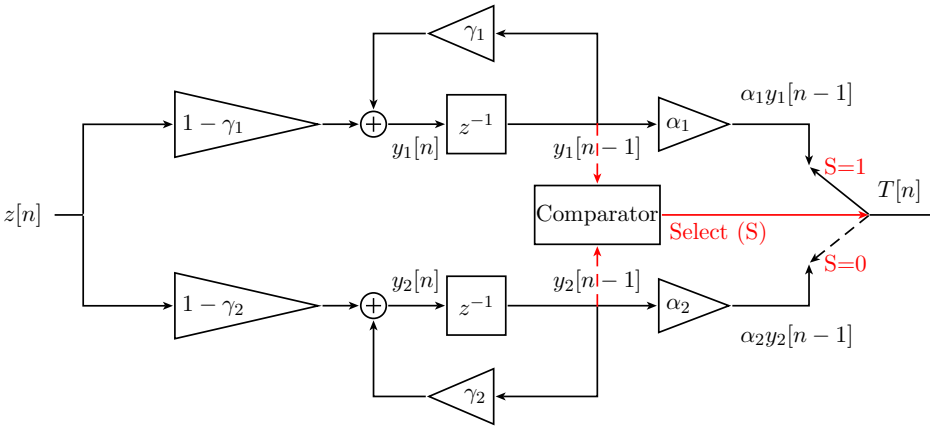

Figure 3.12: Block diagram implementation of the switching algorithm

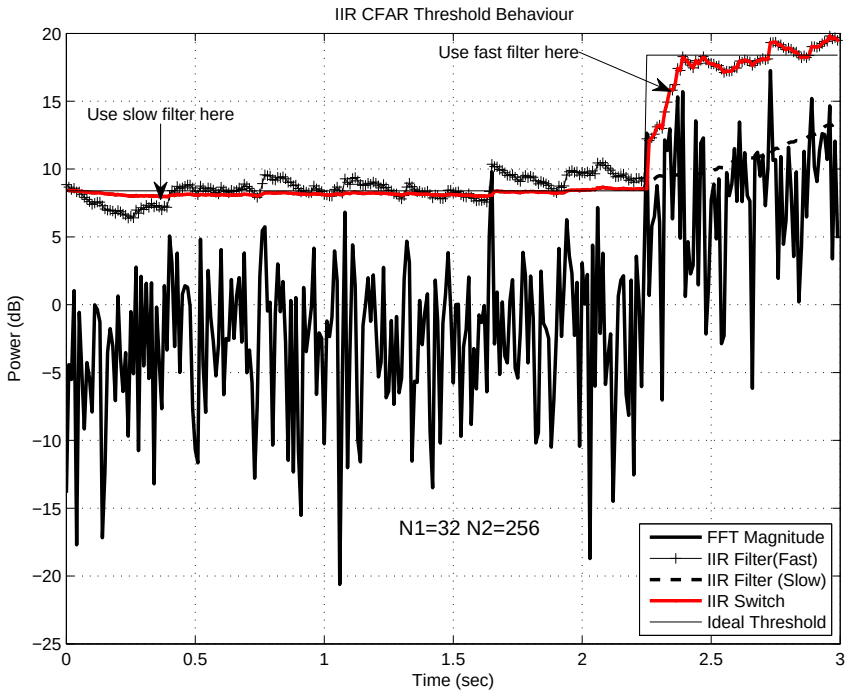

Figure 3.13: SIIR CFAR at work

CHAPTER 4

SIMULATION RESULTS

All simulations are performed in MATLAB 7.9.0 (R2009b). The simulation results of the original SIIR CFAR algorithm are mentioned in Sections 4.2 through 4.4.

4.1 False alarm probability under homogeneous conditions

The false alarm probability under homogeneous environment of the SIIR CFAR algorithm is the same as that of Slow IIR CFAR and Fast IIR CFAR because both of the filters y_1 and y_2 are tuned to operate in the same false alarm probability. Figure 4.1 verifies this line of thought. The false alarm probability does not change because of the switching operation. This is intuitively correct since with some probability p, the first filter is selected and with some probability $1 - p$ the second filter is selected. The false alarm probabilities of both of these filters are the same, therefore the combined switching detector also has the same false alarm probability.

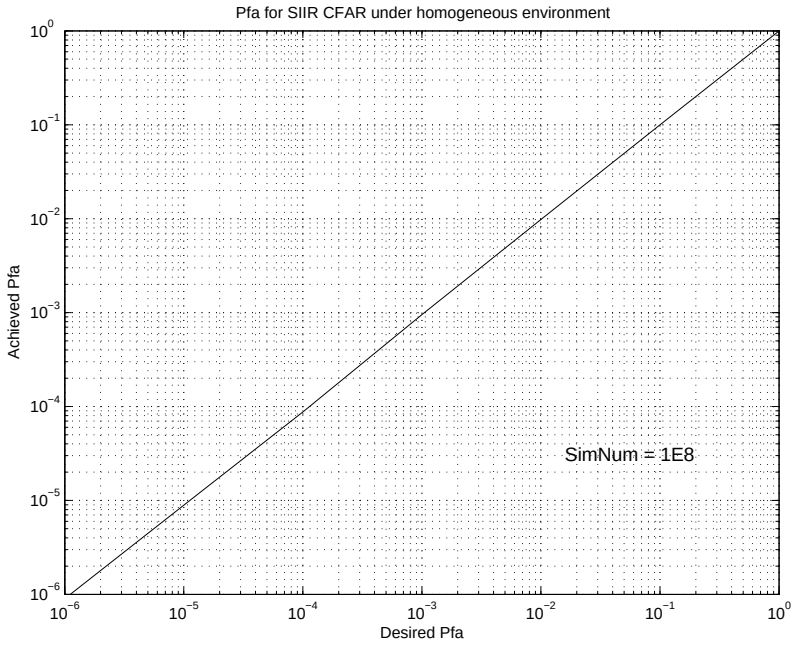

Figure 4.1: Pfa for SIIR CFAR under homogeneous environment

4.2 False alarm probability under clutter edge

4.2.1 Simulation Setup

The downfall of using a single IIR filter with large N, is the excessive false alarms under a clutter edge. The SIIR CFAR algorithm should solve that problem. For algorithms that use windows which span both the leading and lagging samples, the deviation in false alarm probability can both be a decrease or an increase. The false alarm probability can increase, because the test cell can be in the high clutter region but the threshold may be estimated using the low clutter cells therefore causing a too small threshold estimate a false alarm. The false alarm probability can also decrease because the test cell may be in the low clutter region and the threshold may be estimated using the high clutter cells, therefore causing a too high threshold and too small false alarm probability.

Since we are using causal IIR filters with no delay, we are not considering the 'leading window', we only consider the lagging window or 'only the past'. Therefore, the false alarm

simulations done in [18] or [19] are not valid in our case. A new definition of P_{FA} in clutter edge is required and is made in the following sense. First of all, remember that we are using two filters, with approximate lengths N_1 and N_2 and $N_2 > N_1$. During a clutter transition, it is expected that after N_2 cells has passed, both of the filters will be settled and the false alarm probability returns back to its design value. To assess this, the false alarm probability of numerous test cells will be simulated. Figure 4.2 will be helpful. This shows the simulation setup for particular values of the parameters. The false alarm probabilities of the cells for different time instances will be found by simulation. This will show the change in the obtained false alarm probability versus time. Obviously, just after the clutter edge has passed, the false alarm probability will increase due to the fact that the threshold is still in the process of adapting and it is too low. After some time, it is expected that the threshold will settle down and false alarm probability will approach to its steady state level.

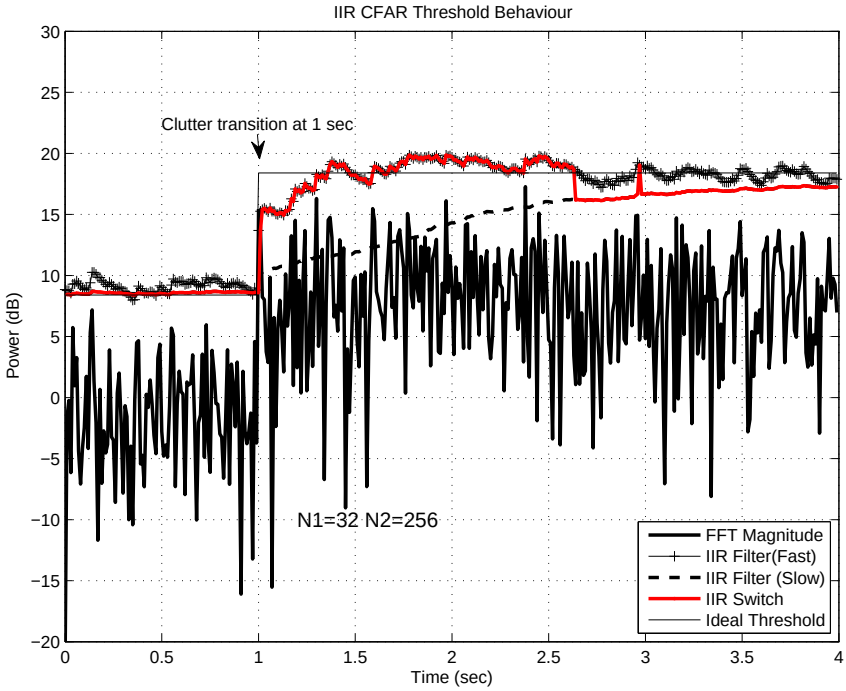

Figure 4.2: Pfa under clutter edge simulation setup

The clutter cells are modeled as exponential random variables with the following pdf.

$$f_z(z) = \frac{1}{\lambda} \exp\left(\frac{-z}{\lambda}\right)$$

where

$$\lambda = \begin{cases} \beta^2, & \text{for lower clutter region} \\ \beta^2(1 + C), & \text{for higher clutter region} \end{cases}$$

Here $z[n]$ is the magnitude squared DFT outputs of a single frequency bin, β^2 is the total noise power and C is the clutter-to-thermal noise ratio (CNR). For all of the simulations, $\beta^2 = 1$ is used. For clutter edge simulations, it is assumed that there are no targets. The simulation results will be in the form of achieved P_{FA} versus time, for a fixed desired P_{FA}. There are four variables that affect the outcome of the simulation results. These are

- Desired P_{FA}
- P_{FS}
- N_1 and N_2
- CNR

Each item is varied while keeping the others fixed and simulation results are obtained.

4.2.2 Simulation Results

Using the simulation setup explained in 4.2.1, the following results are obtained.

4.2.2.1 Effect of varying P_{FA}

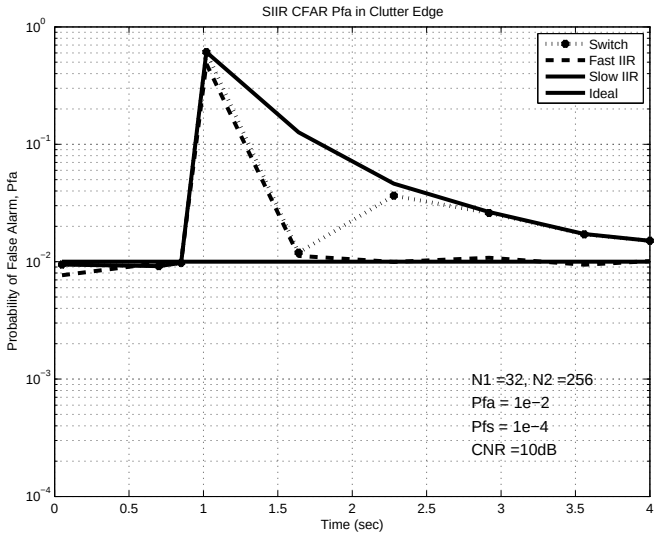

Figure 4.3: P_{FA} simulations under clutter edge for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-2}$,CNR = 10dB, $P_{FS} = 10^{-4}$

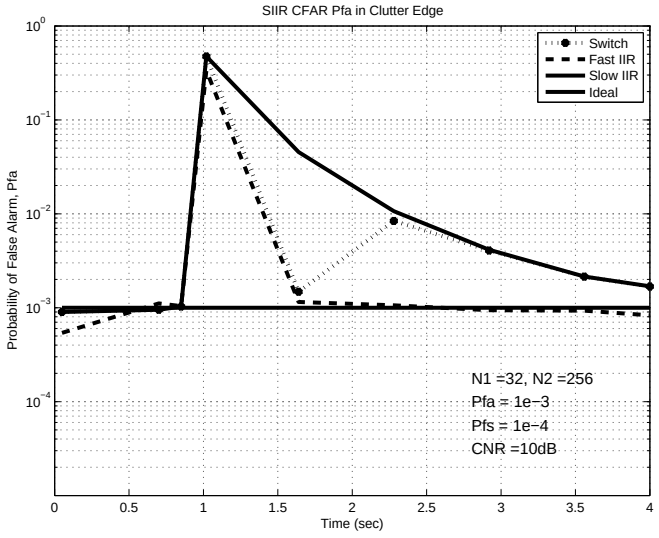

Figure 4.4: P_{FA} simulations under clutter edge for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-3}$,CNR = 10dB, $P_{FS} = 10^{-4}$

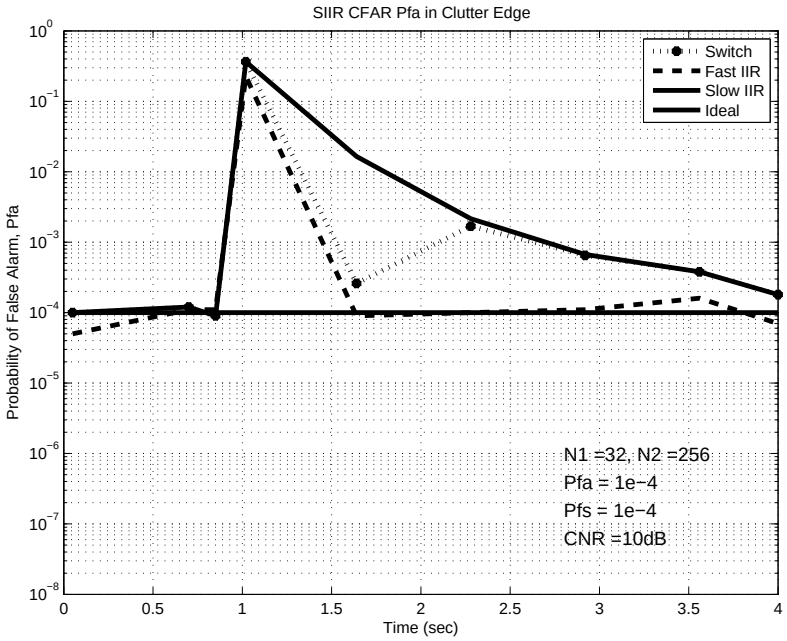

Figure 4.5: P_{FA} simulations under clutter edge for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-4}$,CNR = 10dB, $P_{FS} = 10^{-4}$

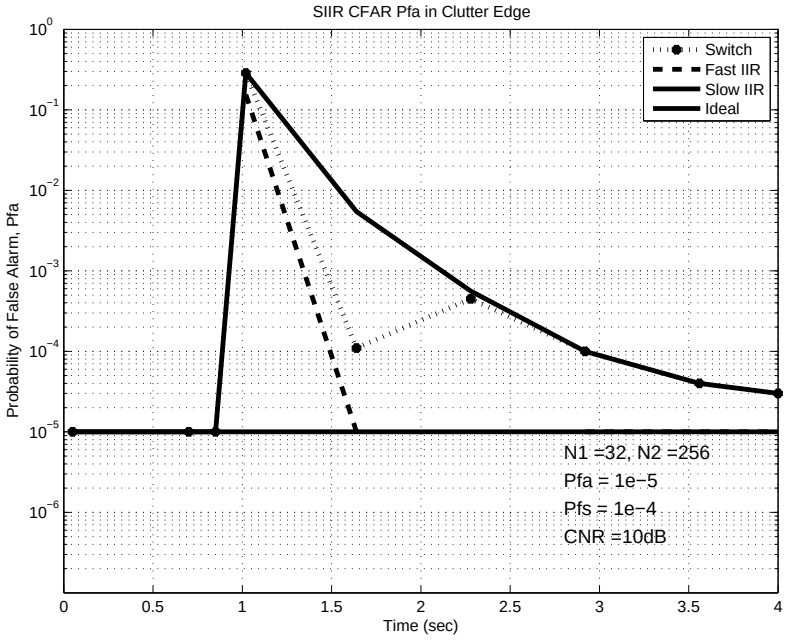

Figure 4.6: P_{FA} simulations under clutter edge for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-5}$,CNR = 10dB, $P_{FS} = 10^{-4}$

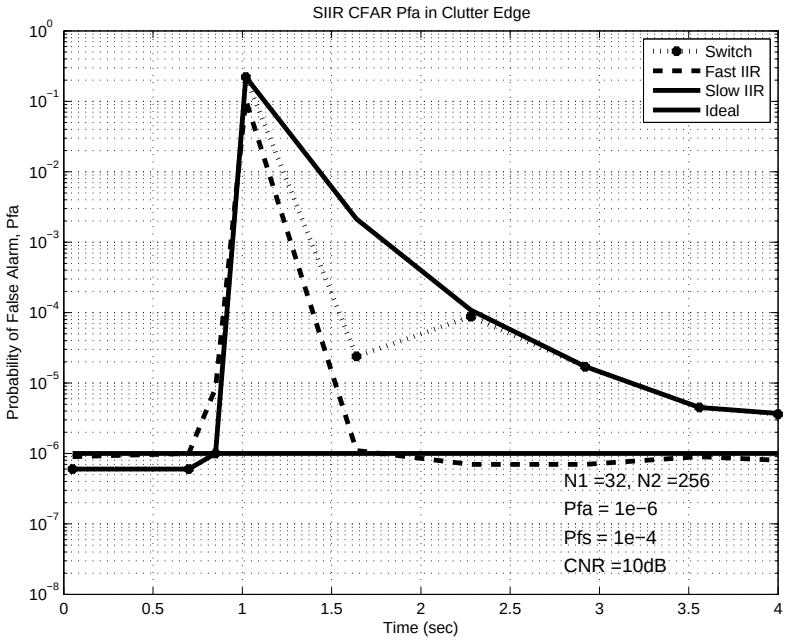

Figure 4.7: P_{FA} simulations under clutter edge for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-6}$,CNR = 10dB, $P_{FS} = 10^{-4}$

From Figure 4.3 through 4.7 we see the change in the performance of the SIIR CFAR algorithm for different values of desired P_{FA}. In each of these figures, P_{FA} increases abruptly during the clutter edge transition and after some time returns to its desired value. It is interesting to note that, no matter what the desired P_{FA} is, the achieved P_{FA} is on the order of 10^{-1} just after the clutter edge has passed. Also at around 1.5th second, the false alarm probability returns to normal for the fast filter but it takes more time for the slow filter to return to the desired false alarm probability. The selection of P_{FA} is not relevant to the performance of SIIR CFAR algorithm but is more of a design choice and is affected by factors such as overall system performance and detection performance.

4.2.2.2 Effect of varying P_{FS}

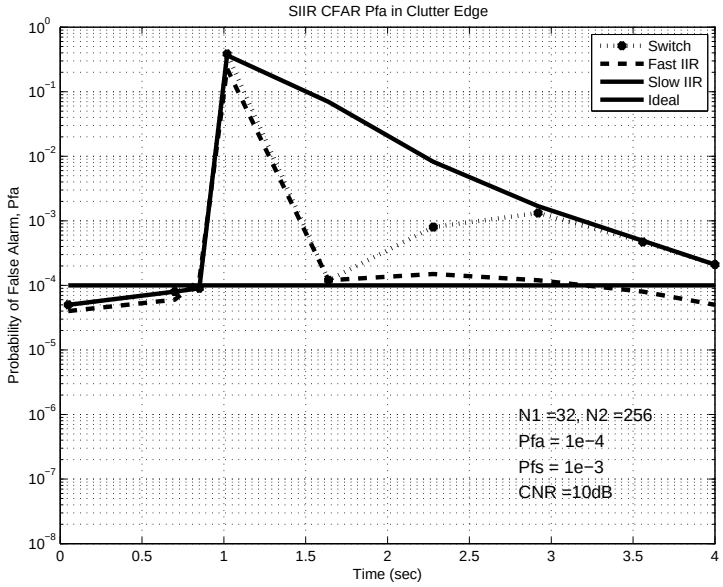

Figure 4.8: P_{FA} simulations under clutter edge for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-4}$,CNR = 10dB, $P_{FS} = 10^{-3}$

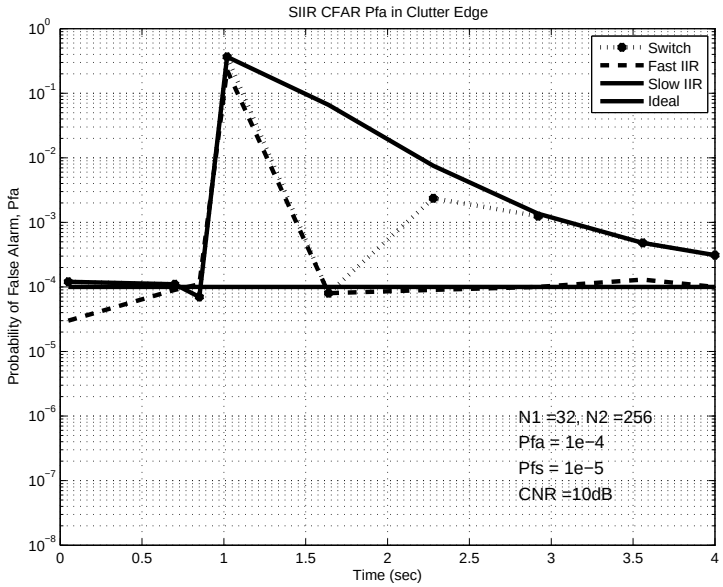

Figure 4.9: P_{FA} simulations under clutter edge for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-4}$,CNR = 10dB, $P_{FS} = 10^{-5}$

It can be seen from Figures 4.8, 4.5 and 4.9, that changing P_{FS} does change the performance of the SIIR CFAR slightly during a clutter transition. All of the figures above show that until up to 1.5th second, the fast filter is selected and after around 3rd second, the slow filter is selected. What happens in between $t = 1.5$ and $t = 3$ is decided by the set P_{FS}. As expected, an increase in P_{FS} means, more switching decisions are performed and the fast filter is selected more frequently. Therefore, the false alarm probability will be closer to that of the fast filter as P_{FS} is increased. This is validated by the performed simulations.

4.2.2.3 Effect of varying N_1

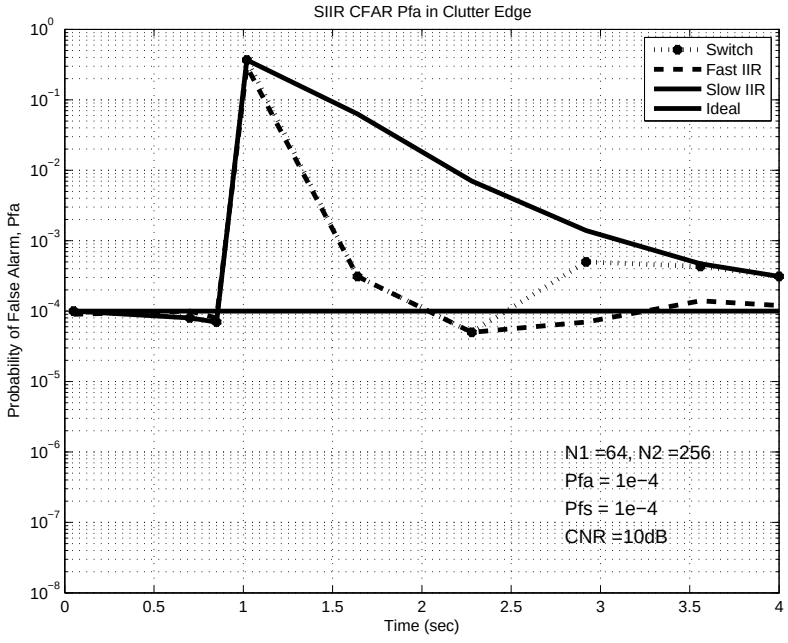

Figure 4.10: P_{FA} simulations under clutter edge for $N_1 = 64$, $N_2 = 256$, $P_{FA} = 10^{-4}$,CNR = 10dB, $P_{FS} = 10^{-4}$

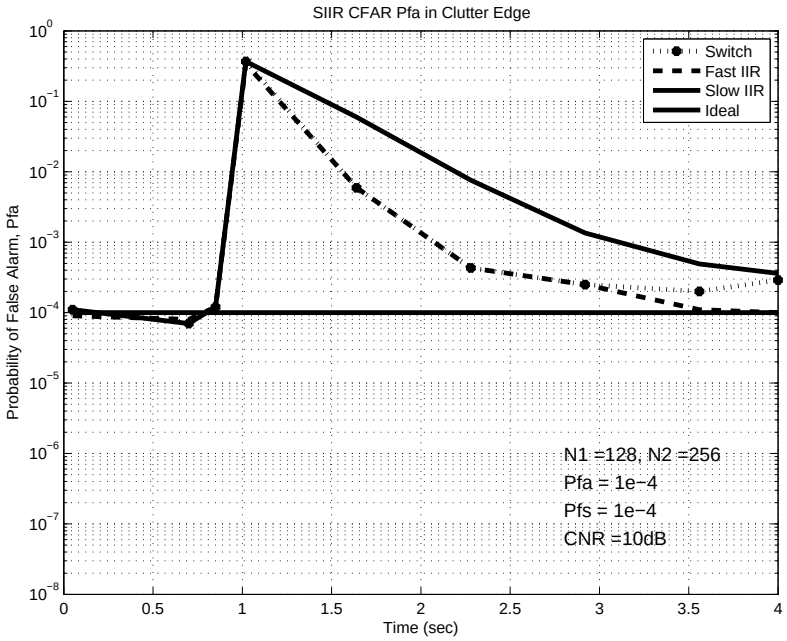

Figure 4.11: P_{FA} simulations under clutter edge for $N_1 = 128$, $N_2 = 256$, $P_{FA} = 10^{-4}$,CNR = 10dB, $P_{FS} = 10^{-4}$

It can be seen from Figures 4.5, 4.10 and 4.11 that changing N_1 has the effect of 'delaying' the transition point from the fast filter to the slow filter. This is also expected because as N_1 increases, the update rate of the fast filter decreases. Initially, in the beginning of the clutter edge the difference $|y_1 - y_2|$ is large, therefore the fast filter is selected. For small N_1, this difference decreases quickly because the update rate of the fast filter is large. For large N_1, however, it takes more time for this difference to become small and therefore the switching point from the fast filter to the slow filter is delayed. It can be seen that in Figure 4.5, this switching point is at 1.5th second, in Figure 4.10 the switching point is at 2.25th second and in Figure 4.11 the switching point is at 3.5th second. These findings support our initial predictions.

4.2.2.4 Effect of varying CNR

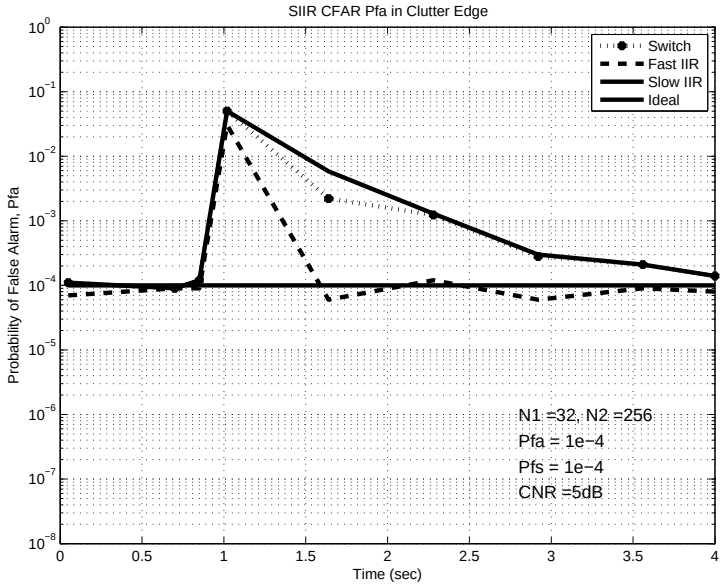

Figure 4.12: P_{FA} simulations under clutter edge for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-4}$,CNR = 5dB, $P_{FS} = 10^{-4}$

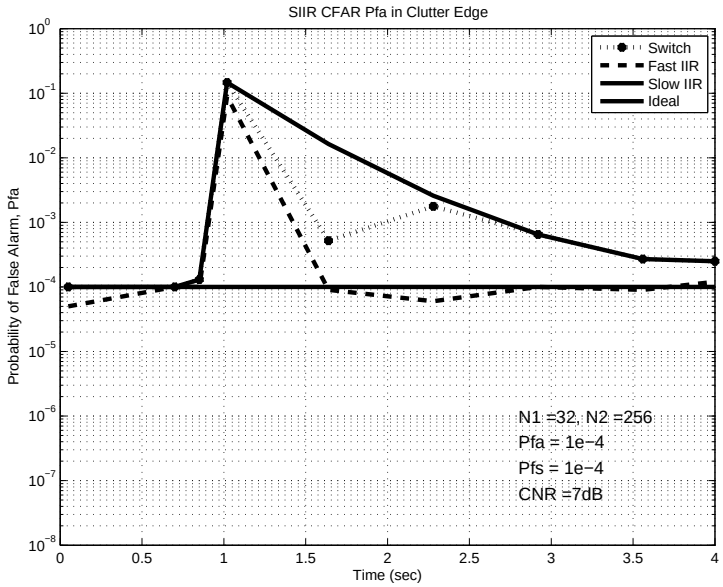

Figure 4.13: P_{FA} simulations under clutter edge for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-4}$,CNR = 7dB, $P_{FS} = 10^{-4}$

Figures 4.12, 4.13 and 4.5 show the simulation results when CNR values of 10dB, 7dB and 5dB are used respectively. As expected, the false alarm increase in the clutter edge is directly proportional with CNR value. (ie: as CNR increases, so does P_{FA} under clutter edge). The switching from fast filter to the slow filter happens at the same point for different CNR values.

4.3 Probability of detection under homogeneous conditions

4.3.1 Simulation Setup

The detection probability P_D under homogeneous conditions is simulated as follows. The pdf of $z[n]$ in this case is:

$$f_z(z) = \frac{1}{\lambda} \exp\left(\frac{-z}{\lambda}\right)$$

where

$$\lambda = \begin{cases} \beta^2, & \text{under } H_0 \\ \beta^2(1+S), & \text{under } H_1 \end{cases}$$

where S is the average signal to noise ratio (SNR) of the target as in [42]. In other words, Swerling 1 model [43] is assumed for the radar returns. $\beta^2 = 1$ is used for all the simulations. The clutter is assumed to be uniform and there are no interfering targets. The probability of detection under homogeneous conditions is expected to be very close to the ideal case. This is shown in the following figures.

4.3.2 Simulation Results

4.3.2.1 Effect of varying P_{FA}

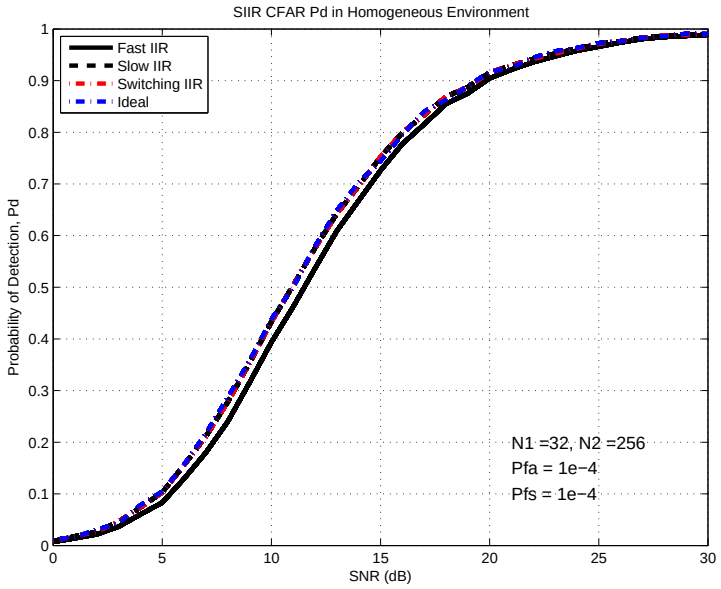

Figure 4.14: P_D simulations for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-4}$, $P_{FS} = 10^{-4}$ under homogeneous conditions

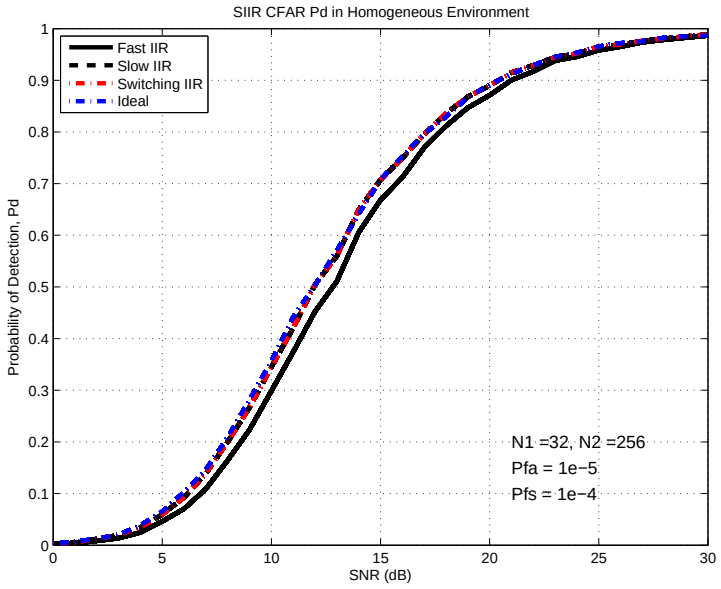

Figure 4.15: P_D simulations for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-5}$, $P_{FS} = 10^{-4}$ under homogeneous conditions

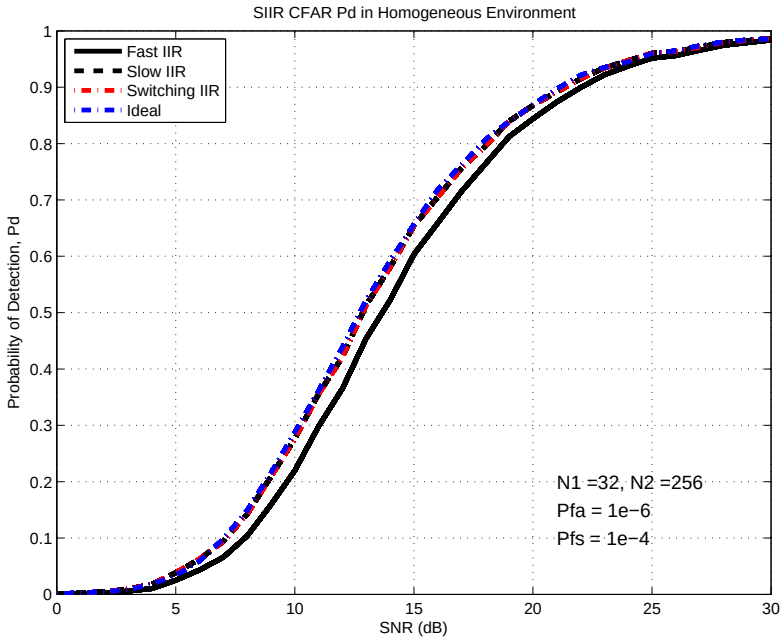

Figure 4.16: P_D simulations for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-6}$, $P_{FS} = 10^{-4}$ under homogeneous conditions

4.3.2.2 Effect of varying N_1

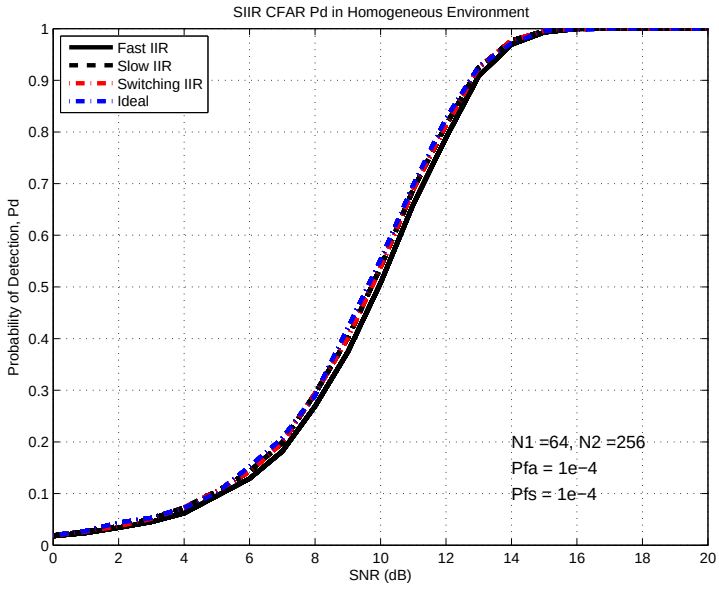

Figure 4.17: P_D simulations for $N_1 = 64$, $N_2 = 256$, $P_{FA} = 10^{-4}$, $P_{FS} = 10^{-4}$ under homogeneous conditions

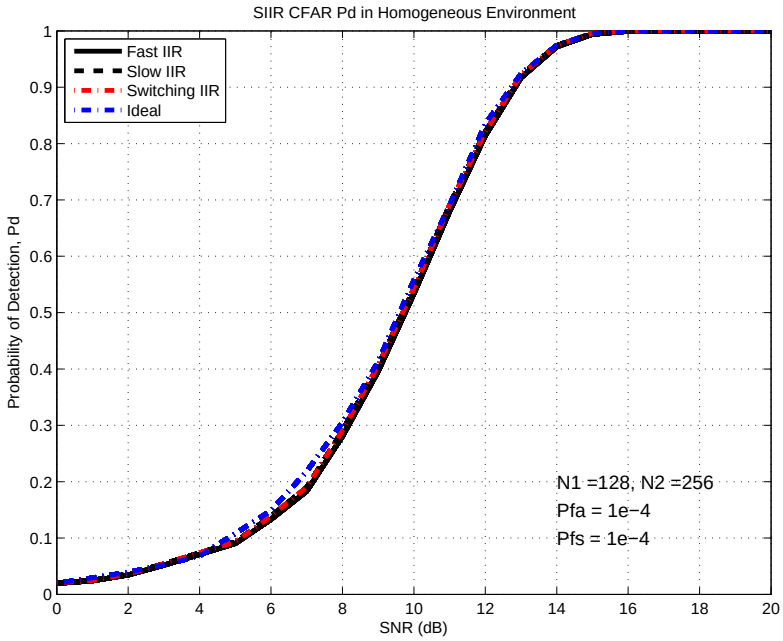

Figure 4.18: P_D simulations for $N_1 = 128$, $N_2 = 256$, $P_{FA} = 10^{-4}$, $P_{FS} = 10^{-4}$ under homogeneous conditions

4.3.2.3 Effect of varying P_{FS}

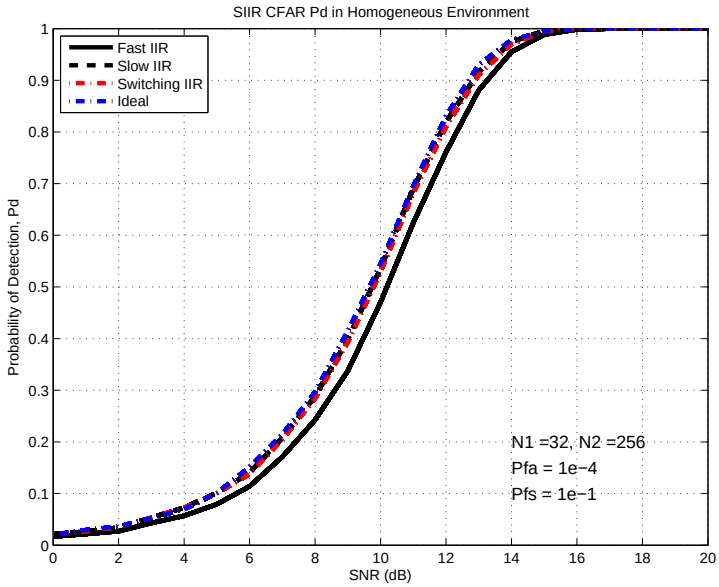

Figure 4.19: P_D simulations for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-4}$, $P_{FS} = 10^{-1}$ under homogeneous conditions

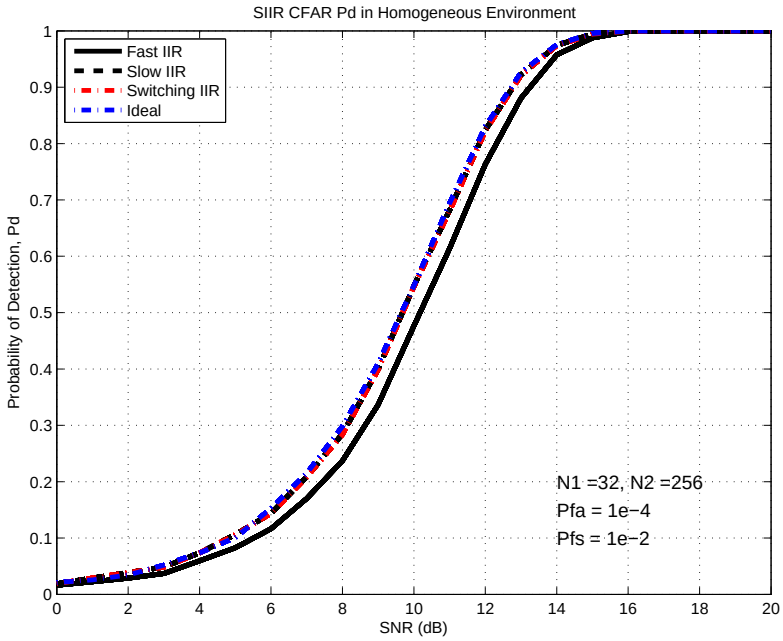

Figure 4.20: P_D simulations for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-4}$, $P_{FS} = 10^{-2}$ under homogeneous conditions

As it is seen from Figures 4.14 through 4.18, the performance of SIIR CFAR is very close to the ideal detector under homogeneous conditions. Changing N_1 does not affect this performance greatly. The effect of changing P_{FS} on detection performance is very minimal. Obviously as P_{FS} decreases, the slower filter will tend to get selected more often and therefore the detection performance will increase. Increasing P_{FS} will lead to a decrease in detection performance however this effect is very subtle and can be omitted. As explained before, the selection of P_{FA} is not relevant to the performance of SIIR CFAR algorithm but is more of a design choice and is affected by factors such as overall system performance and detection performance.

4.4 Probability of detection under interfering targets

In the presence of interfering targets, P_D is not expected to change because the threshold is not updated when a target present decision is made.

The simulation setup is seen in Figure 4.21.

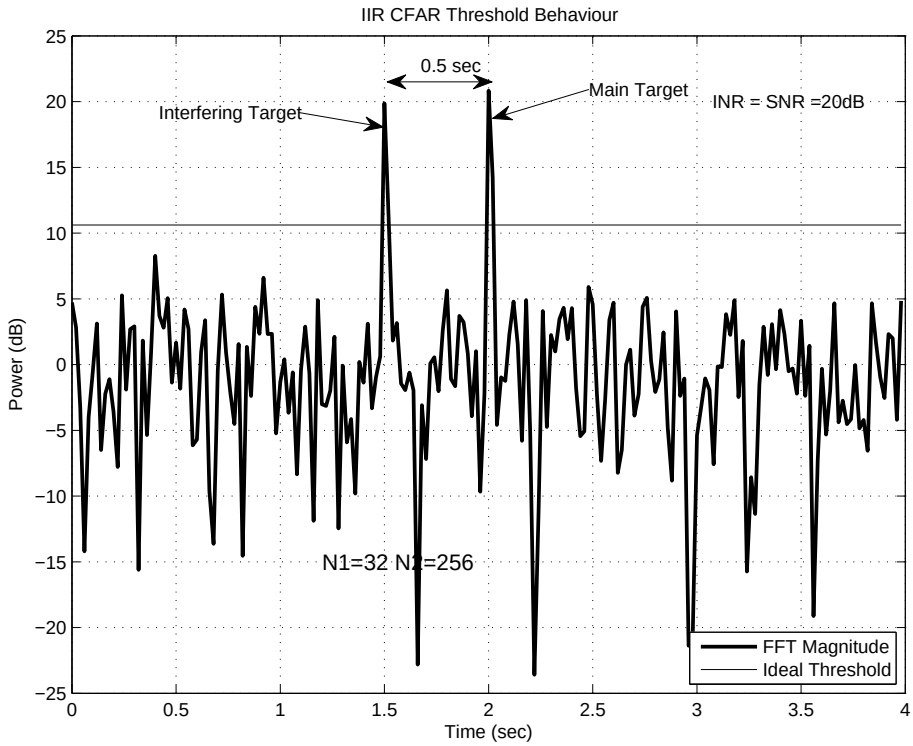

Figure 4.21: Simulation setup for multiple targets

Here one interfering target is shown with a separation of 0.5 seconds.

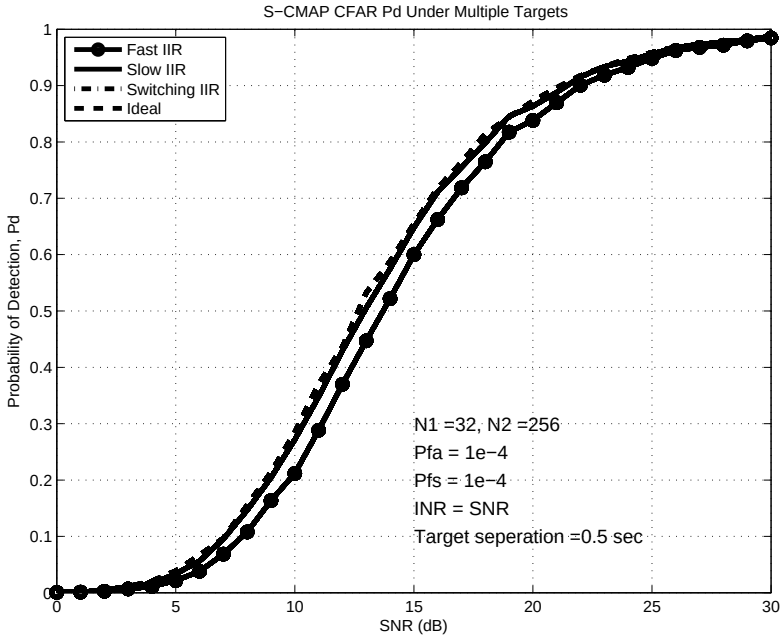

Figure 4.22: P_D simulations for $N_1 = 32$, $N_2 = 256$, $P_{FA} = 10^{-4}$, $P_{FS} = 10^{-4}$ under one interfering target 0.5 second away

Since the threshold is not updated when there is a detection, there is no change in P_D as seen from Figure 4.22.

4.5 Comparison to other CFAR Algorithms

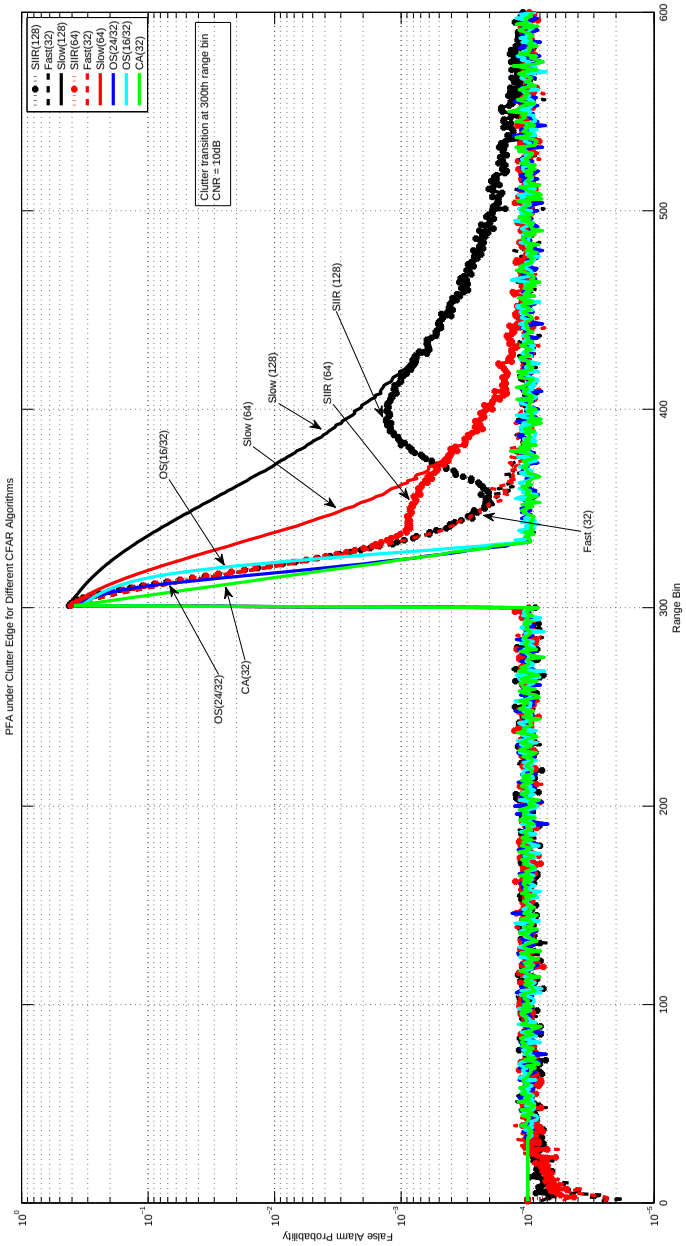

Figure 4.23: P_{FA} simulations under clutter edge for SIIR CFAR vs. other known CFAR algorithms

CHAPTER 5

CONCLUSION

In this thesis, a new algorithm based on IIR filtering and a switching operation for CFAR detection of radar targets is presented, developed and analyzed. The relevant literature is reviewed in Chapter 2. The basics of CFAR detection using IIR filters is presented in Chapter 3. Most CFAR methods use some number of cells (or windows) to set a threshold and then compare the cell under test against this threshold. This means storing at least the window size number of cells and deciding accordingly. Memory requirement and computational demand therefore increases. More advanced CFAR methods (such as ordered statistics or trimmed mean detectors) use ordering of the samples in the window which means even more computation. The simplicity of using an IIR filter to estimate the noise background comes from the fact that only a single value is kept in the memory and this value is updated. The advantages and disadvantages of using IIR filters for detection are summarized in Chapter 3.

In Chapter 4, a new algorithm is proposed which uses two IIR filters with different time constants to address some problems mentioned before. The main reasoning is as follows, two IIR filters with different time constants (preferably one will have a time constant significantly larger than the other one) react differently in a clutter edge environment. It is shown that the slow filter has better detection performance than the fast filter under homogeneous conditions. Also it is shown that the fast filter has better clutter edge performance (the excessive increase of P_{FA} under clutter edge is less) than slow filter. Therefore, it is advised to use the slow filter for the homogeneous environment and the fast filter for the non homogeneous environment.

Then comes the problem of deciding whether the environment is homogeneous or not. This

can be done in terms of comparing the two filter outputs. The absolute value of the difference between the two filter outputs can be used as a measure of homogeneity. In a homogeneous environment, this difference is expected to be small whereas in a non homogeneous environment this difference is expected to be large (due to the differences in the time constants). What we aim to do is, use the slow filter when the difference is small and switch to the fast filter when the difference is large. Then, a measure of 'switching' can be developed as follows:

$$P_{FS} = \{|y_1 - y_2| > T; \text{Homogeneous Environment}\}$$

where P_{FS} is the probability of false switching, T is a predetermined switching threshold and y_1, y_2 are the filter outputs. This probability is denoted as false switching probability because when the environment is homogeneous, no switching operations are expected. Obviously, when P_{FS} is increased, more switching decisions will commence. To find T, the analytical pdfs of y_1 and y_2 are required but to the best of my knowledge, analytical forms are not available.

An approximate pdf for $y_1 - y_2$ is developed in Chapter 4 and it is seen that a linear relationship in the following form performs reasonably well for a large range of design P_{FS} values.

$$P_{FS} = \{|y_1 - y_2| > cy_2; \text{Homogeneous Environment}\}$$

where c can be determined both analytically and through approximations. Using this definition of P_{FS} and calculated c values, the Switching IIR CFAR (SIIR CFAR) algorithm design is complete.

Chapter 5 contains the simulation results. Three main benchmarks were considered. These are, detection under homogeneous conditions, detection under multiple targets and false alarm probability under clutter edge. The detection probability under homogeneous conditions is very close to the ideal case. The detection probability under multiple targets has no performance degradation. The excessive false alarms under clutter edge is close to that of the Fast

As a summary, a new algorithm for CFAR detection of radar targets is developed in this thesis. This algorithm uses two IIR filters and a switching operation to select which filter to use. The main advantage of this algorithm is its computational simplicity, multiple target performance and detection performance. There are algorithms that may perform better in the presence of clutter edges (because of the windowing structure), however efforts are made to increase the clutter edge performance of the SIIR CFAR algorithm. In the future, theoretical approaches to calculating false alarm probability could be investigated. It might also be worthwhile to look into ”double exponential smoothers” (where two past samples are used instead of one) [44] because they might perform better in clutter edges.

REFERENCES

[1] T Bayes. An essay towards solving a problem in the doctrine of chances. 1763. *M.D. computing : computers in medical practice*, 8(3):157–71.

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

[3] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. Prentice Hall, 1998.

[4] Mark A. Richards. *Fundamentals of Radar Signal Processing*. McGraw-Hill, 2005.

[5] Harry L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I*. Wiley-Interscience, 2001.

[6] Nadav Levanon. *Radar Principles*. Wiley-Interscience, 1988.

[7] M. Weiss. Analysis of some modified cell-averaging CFAR processors in multiple-target situations. *IEEE Transactions on Aerospace and Electronic Systems*, AES-18(1):102–114, January 1982.

[8] James Ritcey. Performance Analysis of the Censored Mean-Level Detector. *IEEE Transactions on Aerospace and Electronic Systems*, AES-22(4):443–454, July 1986.

[9] John Rickard and George Dillard. Adaptive Detection Algorithms for Multiple-Target Situations. *IEEE Transactions on Aerospace and Electronic Systems*, AES-13(4):338–343, July 1977.

[10] J Presley Jr. Automated detection in multiple-target environments using the censored mean-level detector. *Acoustics, Speech, and Signal Processing, IEEE*, pages 1–4, 2003.

[11] PP Gandhi and SA Kassam. Analysis of CFAR processors in nonhomogeneous background. *IEEE Transactions on Aerospace and Electronic Systems*, 14(4), 1988.

[12] Fred E. Nathanson, J. Patrick Reilly, Cohen, and Marvin N. *Radar Design Principles: Signal Processing and the Environment, Second Edition*. SciTech Publishing, 1999.

[13] S.L. Wilson. Two CFAR algorithms for interfering targets and nonhomogeneous clutter. *IEEE Transactions on Aerospace Electronic Systems*, 29:57–72, 1993.

[14] Hermann Rohling. Radar CFAR thresholding in clutter and multiple target situations. *IEEE Transactions on Aerospace and Electronic Systems*, (4), 1983.

[15] S Blake. OS-CFAR theory for multiple targets and nonuniform clutter. *IEEE Transactions on Aerospace Electronic Systems*, 24(6):785–790, 1988.

[16] M. Shor and N. Levanon. Performances of order statistics CFAR. *IEEE Transactions on Aerospace and Electronic Systems*, 27(2):214–224, 1991.

[17] ME Smith and PK Varshney. VI-CFAR- A novel CFAR algorithm based on data variability. *IEEE National Radar Conference*, 1997.

[18] M.E. Smith and P.K. Varshney. Intelligent CFAR processor based on data variability. *IEEE Transactions on Aerospace and Electronic Systems*, 36(3):837–847, July 2000.

[19] TTV Cao. Constant false-alarm rate algorithm based on test cell information. *Engineering and Technology*, 2(3):200– 213, 2008.

[20] L.M. Novak. Radar target detection and map-matching algorithm studies. *IEEE Transactions on Aerospace and Electronic Systems*, AES-16(5):620–625, September 1980.

[21] V. Hansen and Harold Ward. Detection Performance of the Cell Averaging LOG/CFAR Receiver. *IEEE Transactions on Aerospace and Electronic Systems*, AES-8(5):648–652, September 1972.

[22] HM Finn. A CFAR design for a window spanning two clutter fields. *IEEE Transactions on Aerospace and Electronic Systems*, (2):155–169, 1986.

[23] Ramon Nitzberg. Clutter Map CFAR Analysis. *IEEE Transactions on Aerospace and Electronic Systems*, AES-22(4):419–421, July 1986.

[24] E. Conte and M. Lops. Clutter-map CFAR detection for range-spread targets in non-Gaussian clutter. Part I: System design. *IEEE Transactions on Aerospace and Electronic Systems*, 33(2):432–443, April 1997.

[25] M Lops. Hybrid clutter-map/L-CFAR procedure for clutter rejection in nonhomogeneous environment. *IEE Proceedings-Radar, Sonar and Navigation*, 1996.

[26] D. Schleher. Harbor surveillance radar detection performance. *IEEE Journal of Oceanic Engineering*, 2(4):318–325, October 1977.

[27] P.P. Gandhi, E. Cardona, and L. Baker. CFAR signal detection in nonhomogeneous Weibull clutter and interference. *Proceedings International Radar Conference*, 0:583–588.

[28] R. Rifkin. Analysis of CFAR performance in Weibull clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 30(2):315–329, April 1994.

[29] A. Sarma and D.W. Tufts. Robust adaptive threshold for control of false alarms. *IEEE Signal Processing Letters*, 8(9):261–263, 2001.

[30] Anas Tom and R. Viswanathan. Switched order statistics CFAR test for target detection. *IEEE Radar Conference, 2008.*, 2(1):1–5, May 2008.

[31] Yu-Zhen Song, Xiang-wei Meng, and Fu-yong Qu. A new CFAR method based on test cell statistics. *IET Conference Publications*, 1(2):426–426, 2009.

[32] A.R. Elias-Fuste, M.G.G. de Mercado, and E. de los Reyes Davo. Analysis of some modified ordered statistic cfar: Osgo and osso cfar. *Aerospace and Electronic Systems, IEEE Transactions on*, 26(1):197 –202, January 1990.

[33] S. Erfanian and V.T. Vakili. Analysis of some CFAR detectors in nonhomogenous environments based on switching algorithm. *2009 International Multimedia, Signal Processing and Communication Technologies*, (1):1–4, March 2009.

[34] A Farrouki and M Barkat. Automatic censored mean level detector using a variability-based censoring with non-coherent integration. *Signal Processing*, 87(6):1462–1473, June 2007.

[35] PP Gandhi and SA Kassam. Optimality of the cell averaging CFAR detector. *IEEE Transactions on Information Theory*, 40(4):1226–1228, 2002.

[36] N. Levanon. Numerically efficient calculations of clutter map CFAR performance. *Aerospace and Electronic Systems, IEEE Transactions on*, (6):813–814, 1987.

[37] J Lehtomäki. Analysis of energy based signal detection. *Department of Electrical and Information*, 2005.

[38] J.J. Lehtomaki, M. Juntti, H. Saarnisaari, and S. Koivu. Threshold setting strategies for a quantized total power radiometer. *IEEE Signal Processing Letters*, 12(11):796–799, November 2005.

[39] M Lops and M Orsini. Scan-by-scan averaging CFAR. *IEE Proceedings For Radar and Signal Processing*, (6), 1989.

[40] Athanasios Papoulis and S.Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw Hill Higher Education, 2002.

[41] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: Foundation for Computer Science*. Addison Wesley, 1994.

[42] H. Goldman. Performance of the excision CFAR detector in the presence of interferers. In *Radar and Signal Processing, IEE Proceedings F*, volume 137, pages 163–171. IET, 2002.

[43] P Swerling. Probability of detection for fluctuating targets. *Information Theory, IRE Transactions on*, 2002.

[44] Joseph J. LaViola. Double exponential smoothing: an alternative to kalman filter-based predictive tracking. In *Proceedings of the workshop on Virtual environments 2003*, EGVE '03, pages 199–206, New York, NY, USA, 2003. ACM.