COMPARISON OF BAYESIAN NETWORKS AND DEMPSTER-SHAFER THEORY IN ATTRIBUTE TRACKING SYSTEMS

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

BY

ÇAĞLAR SAKA

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

JUNE 2013

Approval of the thesis:

COMPARISON OF BAYESIAN NETWORKS AND DEMPSTER-SHAFER THEORY IN ATTRIBUTE TRACKING SYSTEMS

submitted by **ÇAĞLAR SAKA** 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. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** ____________

Prof. Dr. Mustafa Kuzuoğlu
Supervisor, **Electrical and Electronics Engineering Dept., METU** ____________

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Dept., METU ____________________

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

Assoc. Prof. Dr. Çağatay Candan
Electrical and Electronics Engineering Dept., METU ____________________

Assist. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Dept., METU ____________________

Assoc. Prof. Dr. Asım Egemen Yılmaz
Electrical and Electronics Engineering Dept., Ankara Univ. ____________________

Date: **18.06.2013**

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	: ÇAĞLAR SAKA
Signature	:

ABSTRACT

COMPARISON OF BAYESIAN NETWORKS AND DEMPSTER-SHAFER THEORY IN ATTRIBUTE TRACKING SYSTEMS

Saka, Çağlar
M. Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Mustafa Kuzuoğlu

June 2013, 177 pages

In modern civil and military avionic systems, traffic control constitutes one of the most critical parts that requires high-speed, reliable and robust decisions to be done even under noisy conditions. In conjunction with the rapidly developing computer technology, systems became available to perform very long processes within milliseconds.

In this thesis, probabilistic models will be used in order to classify a target and detect if the target is making use of electronic counter-measures (ECM). Thereafter, the performances of different systems under the same conditions will be compared.

Bayesian Networks Theory and Dempster-Shafer Evidence Theory are two most well-known and applicable approaches to classification and attribute tracking problems. Therefore, aforementioned two approaches are chosen in order to simulate desired attribute tracking and detection scenarios.

Subsequent to presenting results obtained by applying abovementioned theories to the selected scenarios, improvements are made in order to increase system performance. The effects of quality of the information source and improvements are presented within this thesis as well as a general comparison of implemented theories.

Keywords: Bayesian Networks, Dempster-Shafer, Evidence Theory, Attribute Tracking, ECM Detection, Probability Update, Belief Update.

ÖZ

NİTELİK TAKİP SİSTEMLERİNDE BAYES AĞLARI VE DEMPSTER-SHAFER TEORİLERİNİN KIYASLAMASI

Saka, Çağlar
Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Mustafa Kuzuoğlu

Haziran 2013, 177 sayfa

Günümüz askeri ve sivil havacılık sistemlerinde trafik kontrolü; gürültülü durumlarda bile hızlı, güvenilir ve sağlam kararlar alınması gereken en kritik kısımlardan biridir. Bilgisayar teknolojisinin gelişmesi ile birlikte, sistemler çok uzun işlemleri bile milisaniyeler seviyesinde gerçekleştirecek duruma gelmiştir.

Bu tezde, hedef sınıflandırmasına ve hedefin herhangi bir elektronik karşı tedbir (ECM) kullanıp kullanmadığının belirlenmesine yönelik olarak olasılıksal modeller kullanılacaktır. Sonrasında, kullanılan farklı sistemlerin aynı şartlar altındaki performansları karşılaştırılacaktır.

Bayes Ağları Kuramı ve Dempster-Shafer Kuramı, sınıflandırma ve nitelik takibi problemlerine uygulanabilir ve en çok bilinen iki yaklaşımdır. Bu sebeple, hedeflenen nitelik takibi ve belirleme senaryolarına uygulanmak üzere bahsi geçen yaklaşımlar seçilmiştir.

Seçilen senaryolara bahsi geçen kuramların uygulanmasının ve sonuçların sunulmasının ardından, sistem performansını arttırmaya yönelik iyileştirmeler yapılmıştır. Bilgi sağlanan kaynağın kalitesi ve yapılan iyileştirmenin etkisi ile birlikte her iki yaklaşımın da genel bir kıyaslaması tez içinde sunulmaktadır.

Anahtar Kelimeler: Bayes Ağları, Dempster-Shafer, Kanıt Kuramı, Nitelik Takibi, ECM Belirlenmesi, Olasılık Güncelleme, Kanaat Güncelleme.

To My Family

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Mustafa KUZUOĞLU for his encouragement, guidance, advice and support throughout this study.

I would like to thank to my family and my friends for their continuous support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

BN	:	Bayesian Networks
BNT	:	Bayesian Network Theory
DST	:	Dempster-Shafer Evidence Theory
ECM	:	Electronic Counter-Measures
ECCM	:	Electronic Counter-Countermeasures
MC	:	Monte Carlo
DAG	:	Directed Acyclic Graphs
RADAR	:	RAdio Detecting and Ranging
BPA	:	Basic Probability Assignment
RGPO	:	Range Gate Pull-Off
ROC	:	Receiver Operating Characteristics
TPR	:	True Positive Rate
FPR	:	False Positive Rate

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND AND SCOPE OF THESIS

Since the beginning of the usage of first radar system, electromagnetic waves have been widely used to detect/track targets together with their attributes and identify their hostility status. The studies regarding detection and tracking of the targets involve many different techniques including various types of probabilistic theories.

Probabilistic theories are being used among many application fields covering military areas such as target classification and electronic counter-measure (ECM) detection. Time and accuracy are two most important parameters that designate performance of the overall system and specify which system is superior to one another.

After detection of a target, target's attributes such as altitude, speed, identification (friend or foe), acceleration etc… will most probably be required to be determined and tracked depending on the purpose of the overall system. Tracking can be preferably defined as: "processing measurements obtained from a target in order to maintain an estimate of its current state, which typically consists of kinematic components (such as position, speed, acceleration etc.) and other components (radiated signal strength, spectral characteristic, feature information etc.)" [1]

Tracking the abovementioned attributes increases chance of classifying the target and hence taking necessary precautions. Using the radar signals, the system can also keep track of the signal power level in order to detect if the target is using any kind of electronic counter-measures (ECM) to fool the radar and lose its track. Upon deciding that the tracked target is using ECM, the radar may decide whether to use any electronic counter-countermeasures (ECCM) technique not to lose track of the target.

In today's world, it is of prime importance to correctly identify a target and correctly evaluate if it poses a treat. Since modern aircrafts are high-speed and capable of high destruction power, reliable systems that gives information beforehand are needed not to lose sight of the target or shoot own aircraft by mistake.

Bayesian Networks Theory (BNT) and Dempster-Shafer Evidence Theory (DST) are two different and most well-known approaches for classification problems. While BNT is dealing with probabilities, DST is concerned with beliefs. Beliefs provide a wider flexibility including "unknown" state which does not exist in BNT. These approaches are implementable to the abovementioned cases and their overall performances can be compared in order to identify each system's advantages and disadvantages to the other.

Throughout this thesis, BNT and DST approaches are studied together with their comparison. Both of these methods are applied to target classification and ECM detection problems in various cases. The main goal of both problems is to reach a convincing probability rate within a considerably short time interval.

While these two approaches have a different belief update algorithm, they contain many important similarities as well.

The studies in the literature on target classification via using BNT, DST and both of them together can be briefly summarized below:

Firstly and most importantly, Bayes' theorem, which is the underlying rule of BNT, is founded by Thomas Bayes in 18th century. Bayes' rule is used to update beliefs in the light of new or existing information. Bayesian statistics, having a wide range of application areas covering engineering, science, economics and medicine is based on this theorem. Development of microchips and thus computers has allowed implementing many successful applications which are based on Bayes' rule. Thomas Bayes' work was documented in 1763 [2].

In 1997, Dennis M. Buede and Paul Girardi published a paper to demonstrate how BNT and DST methods can address the same identification problem. Throughout the paper, demonstration of target identification by using multisensors via these two methods is made and their convergence time to a long run scenario is compared. In the conclusion part, it is shown that DST method has a higher complexity compared to BNT method. Furthermore, according to the simulations made, BNT method has a better convergence rate which indicates that BNT is a better method for combining statistical information [3].

In 2000, Henry Leung and Jiangfeng Wu studied a combined target identification and multitarget tracking approach. Four important attributes were chosen to be tracked in their study: identity of the target, height, speed and acceleration. In their publication, both BNT and DST methods are employed to develop a target track identification system for radar surveillance. It is stated that even though human operators are still used, the necessity of an automated decision system which takes into account many inputs from multiple sensors is inevitable. In conclusion; their work shows that both methods produce accurate information while BNT is computationally simpler and DST is more robust to clutter [4].

In 2003, Barry R. Cobb and Prakash P. Shenoy's publication compares similarities and differences between BNT and DST approaches. Their claim is that given a model in one method, it is possible to transform the model to the other and achieve the same qualitative results. The example given in the document which was claimed to show superiority of DST is discussed and it is argued that DST and BNT approaches could give identical results in case DST model is transformed to BNT model correctly. Computational complexity of DST approach is mentioned as well as the effect of causality on performance of both approaches. When causality is supplied, BNT approach is stated to be easier to implement [5].

In 2005, Don Koks and Subhash Challa published a public report in which two main approaches including BNT and DST methods are discussed. Both of these methods are considered within the scope of identification purposes and simulations obtained from single and multiple sensors are inspected. Results of the simulations clearly show that data fusion from multiple sensors increase reliability of the system. The necessity for initialization of both methods is stated implying the need to consider unknown states in DST case. A thorough comparison of these methods is included in the last part of the report concluding that DST method calculations are usually more complex and simulations of DST give slightly better results [6].

Taking all of these publications into account, it can be concluded that both BNT and DST methods are used for identification, tracking and classification purposes. There are cases and simulations which claim that BNT method is superior whilst there are also proponents of DST method despite its complexity. A general comparison for both of the methods will be presented in the last chapter.

1.2 THESIS OUTLINE

Target classification and detection of ECM using both BNT and DST methods are studied comparatively within the context of the thesis. Both of these methods are tested in numerous cases with various quality sensors and their performance is obtained using their convergence times. Convergence time is calculated by measuring the time required for probability or belief of the system to reach a pre-determined threshold value that indicates a certain amount of belief. When the system exceeds this value, it is considered that miscall ratio of the system is almost zero.

Two different scenarios take place in the scope of this thesis:

In the first scenario, speed and attribute information of an airborne target which is detected by the radar is tracked and used in order to classify the target. In the basic model, target is considered to be either an aircraft or a helicopter. This scenario can be considered as a data fusion problem which gets information from multiple sensors, namely speed and altitude sensors. Performances of BNT and DST methods are observed using the plots that indicate probabilities and beliefs.

In the second scenario, the system keeps track of the signal level of the target right after detection. Modern aircrafts are usually capable of vanishing without a trace by using different ECM techniques. One of the well known ECM techniques is called Range Gate Pull-Off (RGPO) technique. Main idea behind this technique is to amplify signals that are coming from the radar before sending them back. After amplifying reflected signals to a definite level, the aircraft reduces its signal level to normal level which causes the radar lose track of the target since radar's minimum detection level is also increased. Keeping track of the incoming signal levels and comparing them for pre-defined time intervals allow the system to spot whether any ECM technique is used.

The objective to be achieved in abovementioned problems is to construct a model that receives information in short time intervals at a level of seconds or milliseconds depending on the necessity level of promptness and reliability. Information is assumed to be received from various types of sensors such as speed, altitude and signal power level sensors. The quality of these sensors is defined to be different as well in order to inspect overall effects of sensor quality. After starting to collect information, the system updates its initial probability/belief regarding the target's attributes. Depending on the problem, convergence time required to classify target's type or deciding whether the target is using RGPO technique is calculated using both techniques.

In Chapter 2, Bayesian Networks are explained in detail and some important properties are inspected. Naïve Bayes Networks, a subtype of Bayesian Networks are emphasized.

In Chapter 3, Dempster-Shafer Theory is explained with basic assumptions and important properties.

In Chapter 4, two different scenarios that will be used in simulations are explained in detail and algorithms of BNT and DST approaches are presented.

In Chapter 5, simulation results are presented together with performance measurements.

In Chapter 6, simulation results are evaluated and comparison tables are provided. Performances of BNT and DST approaches are compared one-to-one for each scenario and concluding remarks are presented.

CHAPTER 2

BAYESIAN NETWORKS THEORY

In this chapter, basic understanding of Bayesian Networks is to be given together with several examples to its application areas. Firstly, the fundamental theorem called "Bayes' Theorem" is going to be explained with the main lines and an essential example will be given in order to emphasize power of Bayes' Theorem. Afterwards, the Naive Bayesian Network model, which is going to be used throughout the thesis is presented and explained in detail.

2.1 BAYES' THEOREM

18th century mathematician Thomas Bayes stated a theorem used for calculating conditional probabilities. Bayes' theorem, which can be considered as the underlying principle of probabilistic theory, is also known as Bayes' Rule and Bayes' Law. It can be stated as:

$$P(H \mid E) = \frac{P(E \mid H).P(H)}{P(E)} \tag{3.1}$$

In the above formula, H represents "hypothesis" and E represents "evidence". Therefore, the theorem shows how the belief on hypothesis H is updated given evidence E.

P(H|E) is known as the "posterior probability" as it points to the probability of H after taking into account the effect of E.

The term P(H) is known as "prior probability" of H as it does not take into account of any effect coming from E.

P(E|H) is known as "likelihood" as it gives the probability of E assuming H is true.

Finally, P(E) is known as "expectedness" and is independent of the hypothesis. It can be thought of a scaling factor as it scales the probabilities to the range 0 to 1. P(E) can further be mapped out as:

$$P(E) = \sum_i P(E \mid H_i).P(H_i) \tag{3.2}$$

The subscript "i" denotes one of the states of hypothesis H. Summation is taken for all states of H and all states of H.

2.2 INTRODUCTORY EXAMPLE: MONTY HALL PROBLEM

In 1975, a question based on the popular TV show "Let's Make a Deal" had been asked raising a question mark in minds. The solution of the question was given using the most basic probabilistic rule: Bayes' rule. However, there were many objections to the solution since the solution is against intuition and thus called Monty Hall Paradox for a long time. After making computer simulations in the order of millions, the results showed that the solution was correct.

Problem**:** In that TV show, the contestant was given a choice to select one of three doors. Behind one of the doors there was a car and behind others there were goats. As soon as the contestant picks a door, the host opens one of the other doors and reveals a goat. The host then asks the contestant if he would like to stick with his original choice or whether he'd like to switch doors.

Even though first intuition is to assume there is 50%-50% chance of winning the car after the host opens one of the doors, the probability of winning the car increases to 2/3 from 1/3 by switching the doors.

Solution using Bayes' Rule:

The three doors are called X, Y, Z.

C: the number of the door hiding the Car,

S: the number of the door Selected by the player, and

H: the number of the door opened by the Host.

One of the randomly scenarios compatible with the problem definition can be stated as:

The contestant chooses Door X initially and the host opens Door Z and reveals a goat. Then, the probability of winning the car by switching to Door Y can be found by using Bayes' Rule as explained below:

$$P(C=Y \mid H=Z, S=X) = \frac{P(H=Z, C=Y \mid S=X)}{P(H=Z \mid S=X)}$$

$$= \frac{P(H=Z \mid C=Y, S=X)P(C=Y \mid S=X)}{\sum_{i \in \{X,Y,Z\}} P(H=Z \mid C=i, S=X)P(C=i \mid S=X)}$$

$$= \frac{1 \times \frac{1}{3}}{\frac{1}{2} \times \frac{1}{3} + 1 \times \frac{1}{3} + 0 \times \frac{1}{3}} = \frac{2}{3}$$

Because of the random placement of the car and the goats, the contestant initially has a chance of 1/3 winning the car. Thus, it can be concluded that switching the doors double the chance of winning the car.

2.3 BASICS OF BAYESIAN NETWORKS

Probabilistic graphical models are concerned with dealing cases in a graph based model which shows probabilistic interaction between events such as conditional dependency and independency. Their popularity is increasing from day to day because of the facts that graphs are easy to understand and very complex cases can be modeled using these models [7].

Bayesian Networks (BN) can be described as a sub-branch of probabilistic graphical models that obey several fundamental rules and they are also known as Bayesian Belief Networks. They are basically used for representing information flow within a knowledge domain using Bayesian probabilities [8]. The main purpose is to infer posterior probabilities of a hypothesis using the information available. Decision within the light of posterior probability tables is left to the system or user.

Bayesian Networks are directed acyclic graphs (DAG) consisting of nodes and edges between the nodes. DAG can be defined as a directed graph that contains no cycles. An example of a graphical network well-suited to DAG is given below:

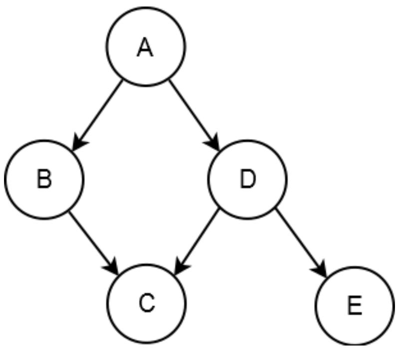

Figure 2-1: Example of a DAG

In the figure above, numbered circles are called as “nodes” and the links between the nodes are called “edges”. Necessary conditions for a graph to be considered as a Bayesian Network are being formed of nodes and directed acyclic edges. Nodes represent random variables with a finite number of states and directed edges represent dependency between the nodes.

The fundamental rules for a graphical model to be considered as BN can be listed as below [8]:

Each node has a finite set of mutually exclusive states.

The nodes form a DAG together with the directed edges.

A conditional probability table is attached to each node X with parents $Y_1, Y_2, \ldots, Y_n$: $P(X| Y_1, Y_2, \ldots, Y_n)$.

Direction of the edges between the nodes indicates influence. In Figure 2.1, Node-A has a direct effect on Node B. The types of nodes can be summarized as:

Child Node: The node that is influenced by another one is called “Child Node” with respect to the node that influences. For example; Node-B is a child node of Node-A.

Parent Node: The node that influences another node is called “Parent Node” of the node that is influenced. For example; Node-A is a parent node of Node-B.

Root Node: A node is called “Root Node” if it does not have any parents which influence it.

Leaf Node: A node is called “Leaf Node” if it does not have any child nodes connected to it.

The convenience provided by using BN is going to be discussed in the following subsection.

2.3.1 STRUCTURE OF BAYESIAN NETWORKS

A Bayesian Network is composed of random variables that are represented by nodes and directed links represented by edges. In this thesis, all random variables are assumed to be discrete and have finite number of states. Nodes can be further classified according to the type of their states such as “hypothesis node” and “evidence node”.

A hypothesis node consists of states which include one of the alternative hypotheses and these kinds of nodes are not directly observable.

An evidence node, which is also known as observed node, consists of states which are directly observed and has assigned a probability value between (0,1).

Evidence concept can be subdivided into “hard evidence” and “soft evidence”. When any state of a node has probability 1, that node is said to be instantiated with hard evidence and it can be said that node is “known”. On the other hand, if any state of that node has a value (0,1), it is said to be instantiated with soft evidence.

In the following subsections, types of Bayesian Networks with respect to their structural design will be explained.

2.3.1.1 CONNECTION TYPES

2.3.1.1.1 SERIAL CONNECTIONS

In the following connection type, nodes are serially connected as shown in the below figure.

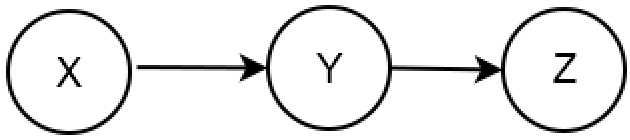

Figure 2-2: Structure of Serial Connected Bayesian Network

When the structure is inspected, it can be observed that X has a direct influence on Y and Y has a direct influence on Z. Information of X can affect Z and information of Z can inversely affect X over the node Y unless Y is fully known. When Y is known, the path between X and Z is blocked and they become independent. This situation is called as “X and Z are d-separated”.

2.3.1.1.2 DIVERGING CONNECTIONS

In these kind of connections, nodes are connected such that all nodes have a common parent as shown in the below figure.

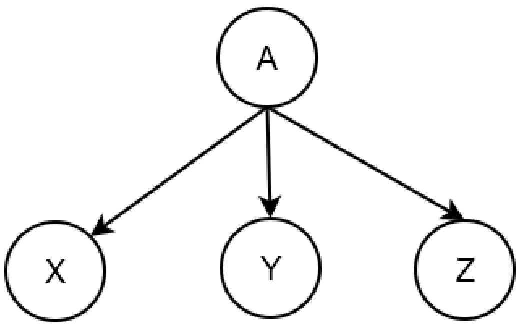

Figure 2-3: Structure of Diverging Connection

It can be observed that information can flow between child nodes (X,Y,Z) through parent node A when A is not known. If the state of A is known, knowledge of any child node does not affect other child nodes since the channel (node A) is blocked. Hence, X and Y and Z are said to be d-separated.

Naïve Bayesian Networks are a subclass of Bayesian Networks that only include one parent node and several child nodes.

2.3.1.1.3 CONVERGING CONNECTIONS

In these kind of connections, nodes are connected such that all nodes have a common child node as shown in the below figure.

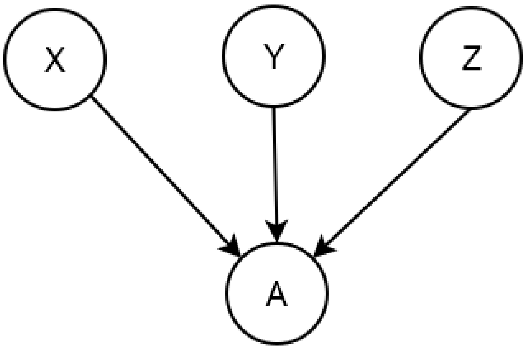

Figure 2-4: Structure of Converging Connection

When the state of child node A is not known, parent nodes have no effect on each other and therefore they are independent. X and Y and Z are said to be d-separated when A is not known. However, if state of A is known, any information about a parent node can increase or decrease probability of other parent node. This effect is called “explaining-away” effect.

2.3.1.2 D-SEPARATION

The concept of d-separation can be defined as a criterion of independence of distinct nodes. D-separation can be examined with respect to connection types:

When figures in the previous part are considered, two separate nodes X and Z are said to be d-separated under the below-mentioned cases:

1) For serial connected structures, if the intermediate variable (node Y) connecting X and Z is known,

2) For diverging connections, if the parent variable (node A) is known,

3) For converging connections, if the child node (node A) is not known.

If two nodes are d-separated, any change in the belief of one node will have no effect on probability of the other node.

2.3.2 CHAIN RULE FOR BAYESIAN NETWORKS

2.3.2.1 GENERAL CHAIN RULE:

The chain rule allows the calculation of joint distributions using conditional probabilities.

Firstly, general rule for a set of random variables will be inspected:

For a set of variables $U=\{X_1, X_2,..., X_n\}$, the joint probability distribution is found by using the general chain rule:

$$P(U) = P(X_n \mid X_1,..., X_{n-1})P(X_{n-1} \mid X_1,..., X_{n-2})...P)(X_2 \mid X_1)P(X_1) \quad (3.3)$$

$$\begin{aligned}
&P(U) = P(X_n \mid X_1,..., X_{n-1})P(X_1,..., X_{n-1})\\
&P(X_1,..., X_{n-1}) = P(X_{n-1} \mid X_1,..., X_{n-2})P(X_1,..., X_{n-2})\\
&:\\
&P(X_1, X_2) = P(X_2, X_1)P(X_1)
\end{aligned} \quad (3.4)$$

2.3.2.2 CHAIN RULE FOR BAYESIAN NETWORKS:

Secondly, chain rule for the set of variables $U=\{X_1, X_2,..., X_n\}$ which constitute a Bayesian Network will be inspected.

BN provide opportunity to simplify the general chain rule and construct the joint probability distribution as:

$$P(U) = \prod_{i=1}^{n} P(X_i \mid pa(X_i)) \quad (3.5)$$

where $pa(X_i)$ represents parent nodes of X_i in the Bayesian Network.

2.3.3 NAIVE BAYESIAN NETWORKS

Naive Bayes Network is a simple model that describes particular class of Bayesian Networks. The type of connection is diverging where the parent node is hypothesis node and child nodes are evident nodes. This type of connection is often called “Naive Bayesian Classifier” which is abbreviated as NBC. NBC have found many application areas as a result of its simplicity and high accuracy. Application areas include text mining, spam filtering, medical diagnosis and other classification problems.

A general structure of a Naive Bayesian Network can be modeled as below:

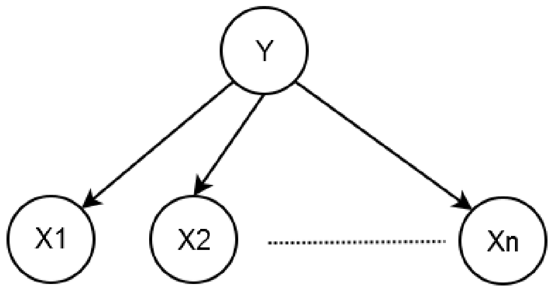

Figure 2-5: Structure of Naive Bayesian Network

In spite of the fact that child nodes are not independent given parent node Y, in NBC structures, it is assumed that all child nodes are mutually conditionally independent given node Y. This assumption is called “Naive Bayesian Assumption”.

Joint probability distribution of the system S={X_1, X_2,…,X_n, Y} is:

$$P(S) = P(Y)\prod_{i=1}^{n} P(X_i \mid Y) \tag{3.6}$$

In this thesis, Naive Bayesian Classifier is used for classification purposes, namely target classification and ECM detection. The parent node is called class variable and the child nodes are called attribute variables in the cases inspected.

CHAPTER 3

DEMPSTER-SHAFER THEORY

In this chapter, a basic understanding of Dempster-Shafer Theory (DST) is to be given together with several examples to its application areas. Firstly, introduction to DST is given with differences to BNT and an example is presented. Afterwards, basics of DST are explained together with the necessary formulas and their explanations. In addition to these, the equations presented are explained via representative examples.

3.1 BACKGROUND OF DEMPSTER-SHAFER THEORY

Dempster-Shafer Theory was firstly introduced to the literature by Arthur Dempster and revised by Glen Shafer in the 1970s [9]. It is also known as “Evidence Theory” and sometimes used together with it as “Dempster-Shafer Evidence Theory”. In this thesis, short form of DST is used.

DST is often considered as a generalization of BNT and provides a mathematically strong way to handle cases when incomplete or conflicting information about the problem is present. There are several key differences between DST and BNT. Two of the most important points can be listed as:

Instead of assigning beliefs to prior probabilities, belief can be assigned to “unknown” cases when the information is limited and therefore preserve ignorance. Thus, this method does not require prior probability initiation.

Beliefs are not restricted to single states and can be assigned to combination of several states.

These two properties provide DST to handle situations where information is missing.

DST has many application areas including target recognition and detection which are two scenarios to be examined in this thesis.

3.1.1 EXAMPLE OF ASSIGNING BELIEFS

In this example, a hypothesis set is considered including two aircraft types: Aircraft and Helicopter. A speed sensor is providing information in order to classify target. If this sensor provides information that the aircraft is 90% aircraft, BNT assigns P(Aircraft)=0.9 and assigns the remaining part of the probability to P(Helicopter)=0.1. Unlike BNT approach, DST approach assigns a belief of 0.9 to aircraft and assigns 0.1 to the unknown state as there is no evidence to support that the target is a Helicopter. Unknown state can be considered as "Aircraft or Helicopter" in this example.

While BNT approach takes information as something is either true or wrong, DST approach allows more ambiguous states since it does not assign any belief to the state that is missing information.

3.2 BASICS OF DEMSPTER-SHAFER THEORY

The basic probability assignment function (abbreviated as "bpa" or "m"), the Belief function (abbreviated as "Bel"), and the Plausibility function (abbreviated as "Pl") are three important functions that are in the scope of DST. The basic probability assignment (bpa) is the basis of DST. Even though bpa is measured against probability of BNT, it does not have to be exactly equivalent. In the following subsection, bpa will be examined in detail.

3.2.1 BASIC PROBABILITY ASSIGNMENTS

Assumption: H is the universal set consisting of all possible states such as H={X_1, X_2,…,X_n}. This set is often called "frame of discernment". Bpa, often represented by "m", provides a mapping from the power set 2^H to the interval 0 to 1. The power set 2^H includes all subsets of H including null set. Bpa of null set is 0 and summation of all other subsets of H is 1.

$$\text{Discernment Frame}: H = \{X_1, X_2, ..., X_n\} \tag{4.1}$$

Power set of H can be shown as follows:

$$\begin{aligned} P(H) = \{&\varnothing, \{X_1\}, \{X_2\}, \{X_1 \cup X_2\}, \{X_3\}, \{X_1 \cup X_3\}, \\ &\{X_2 \cup X_3\}, \{X_1 \cup X_2 \cup X_3\}, \{X_4\}, ..., H\} \end{aligned} \tag{4.2}$$

Considering a subset A={X_1, X_4}, bpa assigned to subset A indicates the ratio of available information that show a particular state belongs to the set A. However, it does not give any further information regarding any subsets of A such as bpa of {X_4} in this case. If any subset of A is of interest, it could be represented by another subset, namely B such that $B \subset A$.

Unless any evidence is provided, bpa of null subset is assigned 0 and bpa of H (the set including all possible states) is assigned 1.

The abovementioned explanations can be formally formalized as:

$$m : P(H) \to [0,1] \tag{4.3}$$

$$m(\varnothing) = 0 \tag{4.4}$$

$$\sum_{A \in P(H)} m(A) = 1 \tag{4.5}$$

In the above formulas, $\varnothing$ denotes null set and P(H) denotes the power set of H.

In order to make a distinction between "basic probability assignment" and "probability" in the classical sense, three further properties can be stated. In bpa, unlike a classical probability, it is not required that:

$$m(\Omega) = 1 \tag{4.6}$$

$$m(A) \leq m(B) \; if \;\; A \subset B \tag{4.7}$$

and it does not require a relationship between m(A) and m(A'), where A' denotes the complement of subset A. For that reason, "belief" and "probability" are going to have two distinct meanings in this thesis.

3.2.2 BELIEF FUNCTIONS

Using the bpa defined in previous section, upper and lower limits of a hypothesis can be calculated. These upper and lower limits are often called "plausibility" and "belief" functions.

Assuming B represents all proper subsets of A, in order to calculate the belief function, masses of all m(B) are required to be summed up to obtain total evidence of support. Belief function is shown as Bel(A) and pre-requisites for Bel(A) to be classified as a belief function are listed as follows:

$$Bel(\varnothing) = 0 \tag{4.8}$$

$$Bel(H) = 1 \tag{4.9}$$

$$Bel(A_1, ..., A_n) \geq \sum_{\substack{I \subset [1,...,n] \\ [I \neq \varnothing]}} (-1)^{|I|+1} Bel\left(\bigcap_{i \in I} A_i \right) \tag{4.10}$$

Belief function is a mapping from the power set 2^H to the interval 0 to 1.

$$Bel : P(H) \rightarrow [0,1] \tag{4.11}$$

$$Bel(A) = \sum_{B \subset A} m(B) \tag{4.12}$$

Equation (3.12) gives the fundamental definition of a belief function and this equation can be explained through an example.

3.2.2.1 EXAMPLE OF BELIEF FUNCTION

Assuming that there are four types of aircrafts that can be detected by the radar system, a belief function can be calculated as below.

Firstly, we define the hypotheses set such that

$$\mathrm{H} = \{\mathrm{F-16},\ \mathrm{F-22},\ \mathrm{MIG-29},\ \mathrm{A-10}\}$$

$$\mathrm{A} = \{\mathrm{F-22},\ \mathrm{MIG-29},\ \mathrm{A-10}\}$$

$$\mathrm{Bel}(\mathrm{A}) = \mathrm{Bel}(\{\mathrm{F-22},\ \mathrm{MIG-29},\ \mathrm{A-10}\}) =$$

$$= \mathrm{m}(\{\mathrm{F-22},\ \mathrm{MIG-29},\ \mathrm{A-10}\}) + \mathrm{m}(\{\mathrm{F-22},\ \mathrm{MIG-29}\}) + \mathrm{m}(\{\mathrm{F-22},\ \mathrm{A-10}\}) +$$

$$\mathrm{m}(\{\mathrm{MIG-29}, \mathrm{A-10}\}) + \mathrm{m}(\{\mathrm{F-22}\}) + \mathrm{m}(\{\mathrm{MIG-29}\}) + \mathrm{m}(\{\mathrm{A-10}\})$$

This example shows that Bel(A) is not the belief assigned only to A but it is the total amount of belief in A. When A is a subset of single element, it is called singleton. (i.e. A=({F-22})). In that case, belief function and mass become equal:

$$Bel(A) = m(A) \tag{4.13}$$

When only information available is composed of belief functions, an inverse function of equation (3.12) can be used:

$$m(A) = \sum_{B \subset A} (-1)^{|A-B|} Bel(B) \tag{4.14}$$

A counterpart of belief function is plausibility function and it can be defined as the sum of all the basic probability assignments of subsets B which intersect the set A. A similar equation to (3.12) can be given as:

$$Pl(A) = \sum_{B \cap A \neq \varnothing} m(B) \tag{4.15}$$

Plausibility shows the degree of hypotheses' possibility and is not the complement of belief function. The connection between belief and plausibility functions can be shown via the following equation:

$$Pl(A) = 1 - Bel(\overline{A}) \tag{4.16}$$

The common property of Belief and Plausibility functions are their nonadditiveness. Summation of all belief or plausibility functions does not require to be equal to 1. Equation (3.16) can be restated as follows using the mass functions:

$$\sum_{B \cap A \neq \varnothing} m(B) = 1 - \sum_{B \cap A = \varnothing} m(B) \tag{4.17}$$

To conclude, since belief and plausibility functions are lower and upper limits, probability of an event is situated between these functions:

$$Bel(A) \leq \mathrm{Prob}(A) \leq Pl(A) \tag{4.18}$$

3.2.3 COMBINATION OF EVIDENCE

Information can be gathered via multiple sources and they need to be aggregated in order to calculate total belief. By using Dempster's rule of combination, basic probability assignments can be combined given that all sources are independent. The rule for combining two mass functions is given by the following equation:

$$m_{12}(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \varnothing} m_1(B) m_2(C)} \quad (4.19)$$

In equation (3.19), denominator is obtained by subtracting summation of all null sets and is used as a normalizing constant. In that equation, m_{12} represents the combined basic probability assignment.

On the other hand, numerator sums the basic probability assignment products of B and C whose intersection is the set A. This rule can be explained through an example.

3.2.3.1 EXAMPLE OF COMBINATION OF EVIDENCE

Example: Assuming that there are two sensors observing an aircraft which can be classified as F-16, F-22, MIG-29 or A-10. Therefore, H={F-16, F-22, MIG-29, A-10}. The sensors give the evidence as in the table below:

Table 3-1: Evidence from sensors and their combination

		Sensor 2		
		F-16,F-22=0.3	F-16,A-10=0.55	□=0.15
Sensor 1	F-16,MIG-29=0.8	**{F-16}=0.24**	**{F-16}=0.44**	{F-16,MIG-29}=0.12
	F-22=0.1	{F-22}=0.03	null=0.055	{F-22}=0.015
	□=0.1	{F-16,F-22}=0.03	{F-16,A-10}=0.055	□=0.015

The table shows individual masses assigned by Sensor-1 and Sensor-2 and their intersection. In order to calculate m_{12}(F-16), the numerator of equation (3.19) will be used. However, the belief assigned to the null set (0.055) violates the rule in equation (3.4) and therefore denominator has to normalize total belief of F-16. Therefore, the combination of evidence can be applied as can be seen below:

$$m_{12}(F-16) = \frac{0.24 + 0.44}{1 - 0.055} \cong 0.72$$

In case there are more than two sources of information, the generalized rule is as follows:

$$m(Y) = \frac{\sum_{X_1 \cap X_2 ... \cap X_n = Y} m_1(X_1) m_2(X_2) ... m_n(X_n)}{1 - \sum_{X_1 \cap X_2 ... \cap X_n = \varnothing} m_1(X_1) m_2(X_2) ... m_n(X_n)} \qquad (4.20)$$

CHAPTER 4

PROBLEM SCENARIOS

AND

PERFORMANCE CRITERIA

In this chapter, the scenarios that are going to be inspected in Chapter 5 are defined. Each scenario is explained for both BNT and DST approaches and introductory examples are given in order to explain probability and belief updates. After each scenario is explained, corresponding algorithms are explained in detail and additional notes are added to the end of each algorithm.

4.1 SCENARIO 1: ATTRIBUTE TRACKING

In modern civil and military systems, it is of utmost importance to detect and identify a target well in advance. Therefore the air traffic can be organized and necessary precautions can be taken in both civil and military aviation.

Problem Statement:

A military base is monitoring air traffic through air defense radar that has approximately 100 km range. As soon as a target is detected, two sensors start to record information regarding the detected target. Types of these two sensors are speed and altitude sensors which measure horizontal speed and vertical height respectively.

Control system is receiving position information from the radar monitor and speed/altitude information from the sensors. The information coming from the sensors are aggregated in order to come to a decision for the target's type. Type of the target is considered to be either an aircraft or a helicopter.

Sensors do not give perfect information regarding the target and are assumed to have two different quality levels, namely "poor" and "good". Furthermore, it is important to note that sensors provide discrete data such as "slow" or "fast" for speed sensor and "low" or "high" for altitude sensor.

The main purpose of this scenario is to combine information gathered by multiple sources Therefore, probability of final decision is increased and the robustness of the system is improved.

4.1.1 BAYESIAN NETWORK THEORY APPROACH

In BNT approach for the problem, several assumptions are made in order to start solution part.

Firstly, the variables are defined as follows:

TargetType={Aircraft, Helicopter}. "TargetType" is the hypothesis variable and denoted as "T" in the equations. Aircraft and Helicopter are denoted with lower-case letters "a" and "h" correspondingly.

Speed={Slow, Fast} "Speed" is an evidence variable and denoted as "S" in the equations. Slow and Fast states are shown with "sl" and "fa" correspondingly.

Altitude={Low, High} "Altitude" is an evidence variable and denoted as "A" in the equations. Low and High states are shown with "lo" and "hi" correspondingly.

The scenario is presented via using a Naive Bayes Network model:

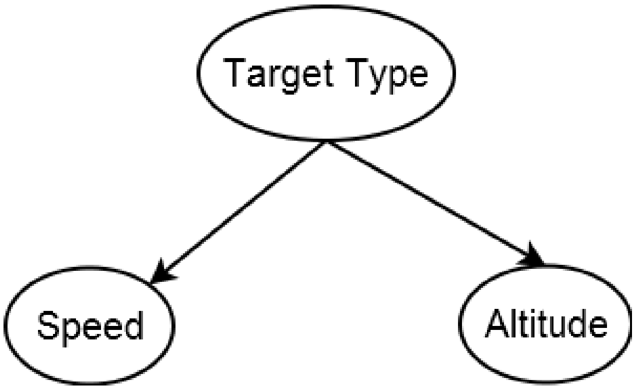

Figure 4-1: Naive Bayes Model of the System

Using the chain rule for Bayesian Networks which was stated in Chapter-2;

$$P(T,S,A) = P(T)P(S\,|\,T)P(A\,|\,T) \qquad (5.1)$$

As can be seen from the above equation, prior information for "TargetType" has to be initialized. Since there is no information regarding the target's type, both aircraft and helicopter states are assigned 0.5 probabilities such that:

P(TargetType=Aircraft)=0.5

P(TargetType=Helicopter)=0.5

Besides prior probability of P(TargetType), conditional probability tables of P(S|T) and P(A|T) are also required. The format of conditional probability tables are assigned as:

P(TargetType=Aircraft)=0.5

P(TargetType=Helicopter)=0.5

$$P(\mathrm{A|T}) = \begin{pmatrix} P(\mathrm{A=lo|T=a}) & P(\mathrm{A=lo|T=h}) \\ P(\mathrm{A=hi|T=a}) & P(\mathrm{A=hi|T=h}) \end{pmatrix}$$

Furthermore, as mentioned previously, sensors have two levels of quality and conditional probability tables for both quality levels are assumed as follows:

If the speed sensor quality is poor:

$$P(\mathrm{S|T}) = \begin{pmatrix} 0.25 & 0.70 \\ 0.75 & 0.30 \end{pmatrix}$$

If the speed sensor quality is good:

$$P(\mathrm{S|T}) = \begin{pmatrix} 0.05 & 0.90 \\ 0.95 & 0.10 \end{pmatrix}$$

If the altitude sensor quality is poor:

$$P(\mathrm{A|T}) = \begin{pmatrix} 0.25 & 0.75 \\ 0.75 & 0.25 \end{pmatrix}$$

If the altitude sensor quality is good:

$$P(\mathrm{A|T}) = \begin{pmatrix} 0.08 & 0.85 \\ 0.92 & 0.15 \end{pmatrix}$$

The states of the evidence nodes may or may not be observed. Calculation of the posterior probability of TargetType in light of information coming from the sensors can be explained via the following example.

4.1.1.1 EXAMPLE FOR POSTERIOR CALCULATION

Considering a case where the quality of speed sensor is good and height sensor is poor, the following information is given: Target is moving fast and there is no information regarding its altitude.

Before starting calculation of posterior probability of “TargetType”, it is of importance to note that prior probability of TargetType is assumed as

[P(Aircraft=0.5)P(Helicopter=0.5)]

Therefore, the following equation is used:

$$P(T=a \mid S=fa)=\frac{P(T=a,S=fa)}{P(S=fa)}=\frac{\sum_{A\in\{lo,hi)} P(T=a,S=fa,A)}{\sum_{\substack{T\in\{a,h\}\\ A\in\{lo,hi\}}} P(T,S=fa,A)}$$

$$=\frac{0.5\times0.95\times0.25+0.5\times0.95\times0.75}{0.5\times0.95\times0.25+0.5\times0.95\times0.75+0.5\times0.1\times0.75+0.5\times0.1\times0.25}$$

$$=\frac{0.475}{0.525}=0.905$$

Therefore, posterior probability for TargetType's being an aircraft is 0.905. Within only one observation, probability of P(TargetType=Aircraft) increased from 0.5 to 0.905.

From the calculations above, it can be observed that quality of Altitude sensor has no effect on the result unless it provides information.

4.1.1.2 ALGORITHMS FOR BNT APPROACH

Two different cases will be examined under the scope of Scenario-1:

First Case: The target is observed for k distinct time intervals and decisions are made regarding TargetType. In this case, previous decisions do not have any effect on the next stage. At each stage, prior probability of TargetType is set to 0.5 for both aircraft and helicopter. Hereinafter, this case will be called "Memoryless Case".

Second Case: Target is observed at each time interval and prior probability of TargetType is updated after each stage. Posterior probability of TargetType at time k is used as prior probability of TargetType at time (k+1). Therefore, after a certain time interval, decision of the system converges to either 0 or 1 depending on the observations and number/quality of the sensors. Hereinafter, this case will be called "Case with Memory".

Therefore, two different algorithms are going to be presented in the following subsection:

4.1.1.2.1 MEMORYLESS CASE

Algorithm for BNT approach for "Memoryless Case" is given below:

1) Bayesian Network topology is created by assigning conditional dependencies between the nodes.
2) TargetType is assigned a status (either aircraft or helicopter) for the intended number of time steps in which observations will be performed.
3) The length of the observation period is called k_{max}. TargetType is fixed during k_{max} time steps.
4) In order to measure performance of the system, confusion matrix variables (TP, TN, FP, FN), Accuracy and TPR variables are created. Furthermore, different threshold values are set in the variable Threshold.
5) Prior probability of TargetType is assigned as:

   P(TargetType)=[P(Aircraft)=0.5 P(Helicopter) =0.5]
6) Conditional probabilities for P(S|T) and P(A|T) are set taking into account the quality of the sensors (evidence nodes).
7) FOR k=1: k_{max} (for each time step)

   a) Information from the sensors is received. For each sensor:

   - If the sensors are providing evidence, it is taken into account and the probability of TargetType is updated using the following equation:

$$P(T=t_i \mid S=s_i, A=a_i) = \frac{P(T,S,A)}{P(S,A)} = \frac{P(T=t_i, S=s_i, A=a_i)}{\sum_{T\in\{a,h\}} P(T, S=s_i, A=a_i)}$$

   where t_i, s_i and a_i represent given information of "TargetType", "Speed" and "Altitude" nodes at time instant "i" respectively.

   - If only one sensor is providing information, i.e. Speed sensor, the following equation is used:

$$P(T=t_i \mid S=s_i) = \frac{P(T,S)}{P(S)} = \frac{\sum_{A\in\{lo,hi\}} P(T=t_i, S=s_i, A)}{\sum_{\substack{T\in\{a,h\}\\ A\in\{lo,hi\}}} P(T, S=s_i, A)}$$

   - If there is no information from both of the sensors, the prior probability of "TargetType" does not change.

   b) After updating probability of TargetType, the probability for the current prediction is compared to threshold level and confusion matrix is updated accordingly.

   END for

8) The posterior probability of "TargetType" at each time step is plotted with respect to time step k.

Accuracy and TPR values are calculated.

4.1.1.2.2 CASE WITH MEMORY

Algorithm for BNT approach for "case with memory" is given below:

1) Bayesian Network topology is created by assigning conditional dependencies between the nodes.
2) TargetType is assigned a status (either aircraft or helicopter) for the intended number of time steps in which observations will be performed.
3) The length of the observation period is called k_{max}. TargetType is fixed during k_{max} time steps.
4) In order to detect when the classifier reaches to a decision, convergence limit is set to 0.99 so that time steps required to reach that point can be calculated as convergence time.
5) Prior probability of TargetType is assigned as:

   P(TargetType)=[P(Aircraft)=0.5 P(Helicopter)=0.5]
6) Conditional probabilities for P(S|T) and P(A|T) are set taking into account the quality of the sensors (evidence nodes).
7) FOR k=1: k_{max} (for each time step)

   a) Information from the sensors is received. For each sensor:

   - If the sensors are providing evidence, it is taken into account and the probability of TargetType is updated using the following equation:

$$P(T=t_i \mid S=s_i, A=a_i) = \frac{P(T,S,A)}{P(S,A)} = \frac{P(T=t_i, S=s_i, A=a_i)}{\sum_{T\in\{a,h\}} P(T, S=s_i, A=a_i)}$$

   where t_1, s_1 and a_1 represent given information of "TargetType", "Speed" and "Altitude" nodes respectively.

   - If only one sensor is providing information, i.e. Speed sensor, the following equation is used:

$$P(T=t_i \mid S=s_i) = \frac{P(T,S)}{P(S)} = \frac{\sum_{A\in\{lo,hi\}} P(T=t_i, S=s_i, A)}{\sum_{\substack{T\in\{a,h\} \\ A\in\{lo,hi\}}} P(T, S=s_i, A)}$$

   - If there is no information from both of the sensors, the prior probability of "TargetType" does not change.

b) Posterior probability of TargetType is assigned to prior probability of TargetType for the next time step.

c) The time step where the probability exceeds 0.99 point is saved as convergence time.

END for

8) The posterior probability of "TargetType" at each time step is plotted with respect to time step k.

The only difference of this case in comparison to "Memoryless Case" is updating prior probability of TargetType at each time step. By this way, probability of posterior decision for TargetType is increased and it is seen that the probability converges to either 0 or 1. While increasing reliability of the system, this method provides robustness to the instant information changes from the sensors.

This case can be simulated via Monte Carlo Method by averaging the values of each time step.

4.1.2 DEMPSTER-SHAFER THEORY APPROACH

In DST approach for the same problem, the same topology is used and similar assumptions are made with only insignificant notational differences.

Firstly, the variables are defined as follows:

TargetType= {Aircraft, Helicopter}. "TargetType" is the universal set and shown as "T" in the equations. Aircraft and Helicopter are denoted with lower-case letters "a" and "h" correspondingly. BPA for TargetType is shown via $m_T(.)$.

Speed= {Slow, Fast} "Speed" is an evidence variable and denoted as "S" in the equations. Slow and Fast states are shown with "sl" and "fa" correspondingly. BPA for Speed is shown via $m_S(.)$.

Altitude= {Low, High} "Altitude" is an evidence variable and denoted as "A" in the equations. Low and High states are shown with "lo" and "hi" correspondingly. BPA for Altitude is shown via $m_A(.)$.

Differently from BNT approach, there is no requirement for prior information in DST approach.

The behavior of the sensors is characterized by the following classifier matrices:

For speed sensor:

$$C_S = \begin{pmatrix} P(S = fa \mid T = a) & P(S = sl \mid T = a) \\ P(S = fa \mid T = h) & P(S = sl \mid T = h) \end{pmatrix} \quad (5.2)$$

For altitude sensor:

$$C_A = \begin{pmatrix} P(A = hi \mid T = a) & P(A = lo \mid T = a) \\ P(A = hi \mid T = h) & P(A = lo \mid T = h) \end{pmatrix} \quad (5.3)$$

Since the sensors have two different levels of quality, classification matrices for both quality levels are presented as follows:

If the speed sensor quality is poor:

$$C_{SP} = [0.75, 0.25; 0.30, 0.70]$$

If the speed sensor quality is good:

$$C_{SG} = [0.95, 0.05; 0.10, 0.90]$$

If the altitude sensor quality is poor:

$$C_{AP} = [0.75, 0.25; 0.25, 0.75]$$

If the altitude sensor quality is good:

$$C_{AG} = [0.92, 0.08, 0.15, 0.85]$$

Classifier matrices of DST approach are analogous to conditional probability tables of BNT approach.

Basic probability assignment for TargetType is assumed as:

$m_T(a = 0, h = 0, \Theta = 1)$ where Θ denotes unknown case which includes {aircraft, helicopter}.

In the light of evidences, belief of "TargetType" is updated. Calculation of the posterior belief is performed using Dempster's Combination of Evidence Rule. The following example illustrates a basic situation.

4.1.2.1 EXAMPLE OF COMBINATION OF EVIDENCE

The problem is the same as explained in previous subsection for BNT approach. The system is observing a target and trying to decide its type given that Target is moving fast and there is no information regarding its altitude. Quality of speed sensor is good and height sensor is poor.

BPA for TargetType is $m_T(a = 0, h = 0, \Theta = 1)$.

BPA for Speed sensor is $m_S(a = 0.95, h = 0, \Theta = 0.05)$ using C_{AG}. It is of importance to note that 0.05 belief is assigned to unknown state but not to the helicopter state.

BPA for Altitude sensor is $m_A(a = 0, h = 0, \Theta = 1)$ since there is no information coming from that sensor.

In order to find posterior belief for TargetType, the following equation is used:

$$m_{T_2}(T) = \frac{\sum_{T_1 \cap S_1 \cap A_1 = a} m_{T_1}(T) m_{S_1}(S) m_{A_1}(A)}{1 - \sum_{T_1 \cap S_1 \cap A_1 = \varnothing} m_{T_1}(T) m_{S_1}(S) m_{A_1}(A)}$$

where $m_{T_2}(T)$ denotes bpa of TargetType at time step 2.

Combining evidences that are of the form m(0, 0, 1) has no effect on final bpa. Henceforth, posterior belief of TargetType is found as (0.95, 0, 0.05) which is the only information coming from Speed sensor. Therefore, $m_T(a) = 0.95, m_T(\Theta) = 0.05$ after single observation is made.

4.1.2.2 ALGORITHMS FOR DST APPROACH

Two different cases will be examined under the scope of Scenario-1:

The target is observed for k distinct time intervals and decisions are made regarding TargetType. In this case, previous decisions do not have any effect on the next stage. At each stage, prior bpa of TargetType is set to $m_T(a = 0, h = 0, \Theta = 1)$. Hereinafter, this case will be called "Memoryless Case".

Target is observed at each time interval and bpa of TargetType is updated after each stage. Posterior bpa of TargetType at time k is used as prior bpa of TargetType at time (k+1). Therefore, after a certain time interval, decision of the system converges to either 0 or 1 depending on the observations and quality of the sensors. Hereinafter, this case will be called "Case With Memory".

Therefore, two different algorithms are going to be presented in the following subsection.

4.1.2.2.1 MEMORYLESS CASE

Algorithm for DST approach for "Memoryless Case" is given below:

1) TargetType is initialized with a fixed state for the intended number of time steps in which observations will be performed.
2) The length of the duration of observation period is determined and this value is set to k_{max} variable. Once TargetType is set, it does not change during k_{max} time steps.
3) Prior bpa for sensors is set as $m(a = 0, h = 0, \Theta = 1)$ since no information is given before starting observations.
4) In order to measure performance of the system, confusion matrix variables (TP, TN, FP, FN), Accuracy and TPR variables are created. Furthermore, different threshold values are set in the variable Threshold.
5) Classification matrices are set depending on quality levels of the sensors.
6) FOR k=1: k_{max} (for each time step)
   a) Information received from the sensors is evaluated:
      - If the sensors are providing evidence, corresponding bpa's are updated in the below format:

$m_S(a=..., h=..., \Theta=...), m_A(a=..., h=..., \Theta=...)$,

- If one or both of the sensors do not provide evidence, the belief is assigned to unknown state and the following bpa is used:

  $m_S(a=0, h=0, \Theta=1)$ or $m_A(a=0, h=0, \Theta=1)$

b) Prior beliefs are updated using Dempster's rule of combination:

$$m_{12}(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C)}$$

c) The combined evidence is preserved in another bpa to be plotted in the end of the simulation.

d) After updating probability of TargetType, the probability for the current prediction is compared to threshold level and confusion matrix is updated accordingly.

END for

7) Posterior belief of "TargetType" at each time step is plotted with respect to time step k.
8) Accuracy and TPR values are calculated.

4.1.2.2.2 CASE WITH MEMORY

Algorithm for DST approach for "case with memory" is given below:

1) TargetType is initialized with a fixed state for the intended number of time steps in which observations will be performed.
2) The length of the duration of observation period is determined and this value is set to k_{max} variable. Once TargetType is set, it does not change during k_{max} time steps.
3) Prior bpa for sensors is set as $m(a=0, h=0, \Theta=1)$ since no information is given before starting observations.
4) In order to detect when the classifier reached to a decision, convergence limit is set to 0.99 so that time steps required to reach that point can be calculated as convergence time.
5) Classification matrices are set depending on quality levels of the sensors.
6) FOR k=1: k_{max} (for each time step)

a) Information received from the sensors is evaluated:

- If the sensors are providing evidence, corresponding bpa's are updated in the below format:

  $m_S(a=..., h=..., \Theta=...), m_A(a=..., h=..., \Theta=...)$

- If one or both of the sensors do not provide evidence, the belief is assigned to unknown state and the following bpa is used:

  $m_S(a=0, h=0, \Theta=1)$ or $m_A(a=0, h=0, \Theta=1)$

b) Prior beliefs are updated using Dempster's rule of combination:

$$m_{12}(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \varnothing} m_1(B) m_2(C)}$$

c) The combined evidence is preserved in another bpa to be plotted in the end of the simulation.

d) Initial belief of the sensor which had provided evidence is updated so that unknown state is not "1" in the next time step.

END for

7) Posterior belief of "TargetType" at each time step is plotted with respect to time step k.

The only difference of this case in comparison to "Memoryless Case" is updating bpa's of the sensor that are providing evidence. Via this step, belief assigned to TargetType is increased and it is seen that the belief of the states "aircraft" and "helicopter" converge to either 0 or 1. Adding this bullet to the algorithm provides robustness to the instant information changes from the sensors while increasing reliability of the system.

This case can be simulated via Monte Carlo Method as explained BNT approach.

4.2 SCENARIO 2: ECM DETECTION

In today's world, military applications have evolved so that a target is capable of understanding if it is being detected and applying preventive techniques in order to remove its trace. These preventive techniques are often called electronic counter-measures (ECM). Since military vehicles have high destructive capabilities, it is a matter of life and death for the tracking system not to lose track of the detected hostile target.

Range Gate Pull-Off (RGPO) is one of the most commonly used ECM techniques that is being used by military air vehicles. When an air vehicle detects electromagnetic waves received from the radar, it can amplify the electromagnetic waves to be returned and therefore may cause a misconception in the radar side. The radar starts to receive amplified waves from the target and increases the threshold signal power level for detection. It is after that the hostile target stops amplifying the waves and vanishes without a trace.

Problem Statement:

A military base is monitoring air traffic through air defense radar that has approximately 100 km range. As soon as a target is detected, the radar starts to receive information of the signal power level coming from the target. The purpose of keeping record of the signal power level information is to give a clear understanding if the target is applying an ECM technique and take necessary precautions.

According to the Scenario, the target will start with low signal power level and after some time, the target will increase its signal power level to "high". Performance measurements and detection speed of the system will be observed. In the cases with memory, the system will be stabilized as soon as the probability and belief level of 0.99 is reached. The duration required to reach convergence level of 0.99 will be called "convergence time".

Via this problem, performance measurements and robustness of BNT and DST approaches will be examined under a "change of state" situation and their recovery times will be compared.

In Scenario-2, differently from Scenario-1, ROC curves will be drawn by means of TPR and FPR values and these curves will help investigating performance of the classifier. In Scenario-1, since the target type is always same, TPR values are equal to Accuracy values. The reason why ROC curves cannot be used in Scenario-1 is since the FPR values cannot be calculated. Furthermore, an alternative approach that is called "Finite Memory Case" will be inspected that only uses latest n time steps in the memory and predicts accordingly.

The problem will be studied in two main cases, namely "Memoryless Case" and "Case with Memory". "Case with Memory" will further be divided to three different categories and studied thoroughly, which are named "Ordinary Case", "Limited Case" and "Finite Memory Case".

In "Memoryless Case", performances of single and multiple sensors will be studied together with performance measurements. Accuracies, costs, TPR and FPR values are going to be calculated and ROC curves will be drawn by setting different threshold values.

"Case with Memory" will be studied under three sub-sections:

In the first case, the target is observed for k distinct time intervals and decisions are made regarding ECM usage. Since usage of ECM is rare as explained previously, prior ECM probability is assigned as 0.1. The target will increase its signal power level at different time steps (i.e. 10,40 etc.).

In the second case, simulations are performed with an improvement factor. In order to increase system performance, a lower limit is set for the probabilities and beliefs to prevent their approaching very close to zero. Getting very small values prevents system from quick recoveries when target is applying ECM.

In the last case, a hybrid approach is implemented by using latest time steps together with the current observation in order to predict ECM usage.

4.2.1 BAYESIAN NETWORK THEORY APPROACH

In BNT approach of the Scenario-2, several assumptions are made providing the algorithms:

ECM = {Yes, No}. “ECM” is the hypothesis variable and also shown as ECM in the equations. The states “Yes” and “No” are denoted with lower-case letters “y” and “n” correspondingly.

Signal Power = {Low, High}. “Signal Power” is evidence variable and denoted as “SP” in the equations. Slow and Fast states are shown with “sl” and “fa” correspondingly.

The scenario is presented via using a Naïve Bayes Network model:

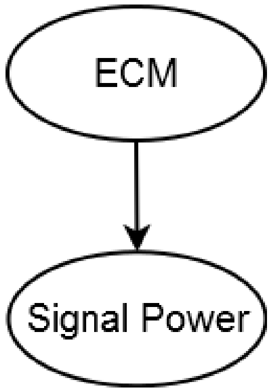

Figure 4-2: Bayesian Model of the System

Using the chain rule for Bayesian Networks which was stated in Chapter-2;

$$P(ECM, SP) = P(ECM)P(SP \mid ECM) \quad (5.4)$$

In order to use the above equation, prior information for “ECM” has to be initialized. ECM technique is often used in warfare or under critical situations such as an aircraft violating airfield of another country. Since the usage of ECM is rare when compared to “not used” cases, prior probabilities are assigned as:

$P(ECM = yes) = 0.1$

$P(ECM = no) = 0.9$

Besides prior probability of P(ECM), conditional probability table of P(SP|ECM) is also required. The format of conditional probability table is assigned as:

$$P(\text{SP}|\text{ECM}) = \begin{pmatrix} P(\text{SP=hi}|\text{ECM=n}) & P(\text{SP=hi}|\text{ECM=y}) \\ P(\text{SP=lo}|\text{ECM=n}) & P(\text{SP=lo}|\text{ECM=y}) \end{pmatrix} \quad (5.5)$$

Similarly to Scenario-1, the source of information has two different levels of quality and conditional probability tables for “poor” and “good” levels are presented as:

If the signal power sensor quality is poor:

$$P(\text{SP|ECM}) = \begin{pmatrix} 0.25 & 0.75 \\ 0.75 & 0.25 \end{pmatrix}$$

If the signal power sensor quality is good:

$$P(SP \mid ECM) = \begin{pmatrix} 0.05 & 0.90 \\ 0.95 & 0.10 \end{pmatrix}$$

4.2.1.1 ALGORITHMS FOR BNT APPROACH

4.2.1.1.1 MEMORYLESS CASE

Algorithm for BNT approach for "Memoryless Case" is given below:

1) Bayesian Network topology is created by assigning conditional dependencies between the nodes.
2) ECM is initialized "no" for a period called "low signal period" which is shown with "t" and initialized "yes" for the rest of the period in which observations will be performed.
3) The length of the duration of observation period is determined and this value is set to k_{max} variable. ECM only changes its state once during k_{max} time steps.
4) In order to measure performance of the system, confusion matrix variables (TP, TN, FP, FN) and Accuracy, TPR, FPR variables are created. Furthermore, different threshold values are set in the variable Threshold.
5) Prior probability of ECM is assigned as:

   P(ECM)=[P(Yes=0.1) P(No=0.9)]
6) Conditional probability table for P(SP|ECM) is set taking into account the quality of the sensor which provide signal power level information.
7) FOR k=1: k_{max} (for each time step)
   a) Information from the sensor is received. If there are more than one sensor providing information, the evidences are combined.
      - If there is no information from both of the sensors, the prior probability of "ECM" does not change.
   b) Posterior probability of ECM is compared to threshold value resulting in increase in one of TP,TN,FP or FN variables (confusion matrix variables).

   END for

8) Accuracy, TPR, FPR and Cost are calculated by using confusion matrix and cost matrix variables.
9) The posterior probability of “ECM” at each time step is plotted with respect to time step k.
10) ROC curve is plotted by using TPR and FPR values.

4.2.1.1.2 CASE WITH MEMORY

4.2.1.1.2.1 ORDINARY CASE

Algorithm for BNT approach for “ordinary case” is given below:

1) Bayesian Network topology is created by assigning conditional dependencies between the nodes.
2) ECM is assigned a status (either yes or no) for the intended number of time steps in which observations will be performed. According to the scenario, the signal power level will make an increase at time step “t”. This time step will be called “low signal period” henceforth.
3) The length of the observation period is called k_{max}. ECM rises from “low” to “high” only once during k_{max} time steps.
4) Prior probability of ECM is assigned as:

   P(ECM)=[P(Yes=0.1) P(No=0.9)]
5) “Convergence Time” variable is defined as a matrix of 1x k_{max}.
6) Conditional probability table for P(SP|ECM) is set taking into account the quality of the sensor which provide signal power level information.
7) FOR k=1: k_{max} (for each time step)

   a) Information from the sensor is received.

   - If the sensor is providing evidence, it is taken into account and the probability of ECM is updated using the following equation:

     $$P(ECM = e_i \mid SP = sp_i) = \frac{P(ECM, SP)}{P(SP)} = \frac{P(ECM = e_i, SP = sp_i)}{\sum_{ECM \in \{y,n\}} P(ECM, SP = sp_i)}$$

     where e_i and sp_i represent given information of “ECM” and “Signal Power” nodes at time instant “i” respectively.
   - If there are several sensors providing information, evidences are combined.

b) Posterior probability of ECM is assigned to prior probability of ECM for the next time step.

END for

8) Convergence Time variable is assigned with the time step at which probability exceeds 0.99.

9) The posterior probability of "ECM" at each time step is plotted with respect to time step k.

The same procedure can further be expanded so that Monte Carlo Simulation results are obtained. If the same procedure is applied for a predetermined number of times (i.e. 1000 times) and the results are averaged in the end, performance of the systems can be compared.

4.2.1.1.2.2 LIMITED CASE

Algorithm for BNT approach for "limited case" is given below:

1) Bayesian Network topology is created by assigning conditional dependencies between the nodes.

2) ECM is assigned a status (either yes or no) for the intended number of time steps in which observations will be performed. According to the scenario, the signal power level will make an increase at time step "t". This time step will be called "low signal period" henceforth.

3) The length of the observation period is called k_{max}. ECM rises from "low" to "high" only once during k_{max} time steps.

4) Prior probability of ECM is assigned as:

   P(ECM)=[P(Yes=0.1) P(No=0.9)]

5) "Convergence Time" variable is defined as a matrice of 1x k_{max}.

6) "Lower Limit" is set so that P(ECM) cannot take values lower than that limit.

7) Conditional probability table for P(SP|ECM) is set taking into account the quality of the sensor which provide signal power level information.

8) FOR k=1: k_{max} (for each time step)

   a) Information from the sensor is received.

   - If the sensor is providing evidence, it is taken into account and the probability of ECM is updated using the following equation:

$$P(ECM = e_i \mid SP = sp_i) = \frac{P(ECM, SP)}{P(SP)} = \frac{P(ECM = e_i, SP = sp_i)}{\sum_{ECM \in \{y,n\}} P(ECM, SP = sp_i)}$$

     where e_i and sp_i represent given information of "ECM" and "Signal Power" nodes at time instant "i" respectively.

- If there are several sensors providing information, evidences are combined.

b) Updated ECM probability is compared to limit value at each time step and if posterior probability of ECM starts to get infinitesimal values, the lower limit is assigned to posterior probability of ECM and that value is used as prior probability of the next step.

c) Posterior probability of ECM is assigned to prior probability of ECM for the next time step.

END for

9) Convergence Time variable is assigned with the time step at which probability exceeds 0.99.

10) The posterior probability of "ECM" at each time step is plotted with respect to time step k.

The same procedure can further be expanded so that Monte Carlo Simulation results are obtained.

4.2.1.1.2.3 FINITE MEMORY CASE

4.2.1.1.2.3.1 APPROACH-1

1) Bayesian Network topology is created by assigning conditional dependencies between the nodes.
2) Bayesian Network topology is created by assigning conditional dependencies between the nodes.
3) ECM is initialized "no" for a period called "low signal period" which is shown with "t" and initialized "yes" for the rest of the period in which observations will be performed.
4) The length of the duration of observation period is determined and this value is set to k_{max} variable. ECM only changes its state once during k_{max} time steps.
5) Prior probability of ECM is assigned as:

   P(ECM)=[P(Yes=0.1) P(No=0.9)]
6) Conditional probability table for P(SP|ECM) is set taking the quality of the sensor into account which provide signal power level information.
7) FOR k=1: k_{max} (for each time step)

   a) Information from the sensor is received and following equation is used to update:

$$P(ECM = e_i \mid SP = sp_i) = \frac{P(ECM, SP)}{P(SP)} = \frac{P(ECM = e_i, SP = sp_i)}{\sum_{ECM \in \{y,n\}} P(ECM, SP = sp_i)}$$

b) Posterior probabilities are stored in ECM variable which are independently calculated as in “Memoryless Case”. For the final decision, latest three independent results (k-2,k-1,k) are combined.

END for

8) Posterior probability of “ECM” at each time step is plotted with respect to time step k.

4.2.1.1.2.3.2 APPROACH 2

1) Bayesian Network topology is created by assigning conditional dependencies between the nodes.
2) ECM is initialized “no” for a period called “low signal period” which is shown with “t” and initialized “yes” for the rest of the period in which observations will be performed.
3) The length of the duration of observation period is determined and this value is set to k_{max} variable. ECM only changes its state once during k_{max} time steps.
4) Prior probability of ECM is assigned as:

   P(ECM)=[P(Yes=0.1) P(No=0.9)]
5) Conditional probability table for P(SP|ECM) is set taking the quality of the sensor into account which provides signal power level information.
6) FOR k=1: kmax (for each time step)
   a) Weighted update algorithm is used where most of the belief is assigned to the latest observation.
   b) Update algorithm for Approach-2:

      m(k+1)=0.6×m(k)+0.25×m(k-1)+0.15×m(k-2)

   END for
7) Convergence Time variable is assigned with the time step at which probability exceeds 0.99.
8) Posterior probability of “ECM” at each time step is plotted with respect to time step k.

4.2.2 DEMPSTER-SHAFER THEORY APPROACH

In DST approach, the same topology is used and similar assumptions are made with only insignificant notational differences.

First, the variables are defined as follows:

ECM= {Yes, No}. "ECM" is the universal set and shown as "ECM" in the equations. Yes and No states are denoted with lower-case letters "y" and "n" correspondingly. BPA for ECM is shown via $m_{ECM}(.)$.

Signal Power= {Low, High} "Signal Power" is an evidence variable and denoted as "SP" in the equations. Low and High states are shown with "lo" and "hi" correspondingly. BPA for Altitude is shown via $m_{SP}(.)$.

Differently from BNT approach, there is no requirement for prior information in DST approach.

The behavior of the sensor is characterized by the following classifier matrix:

$$C_{SP} = \begin{pmatrix} P(\text{SP=hi}|\text{ECM=y}) & P(\text{SP=lo}|\text{ECM=y}) \\ P(\text{SP=hi}|\text{ECM=n}) & P(\text{SP=lo}|\text{ECM=n}) \end{pmatrix}$$

Since the sensor has two different levels of quality, classification matrices for both quality levels are presented as follows:

If the speed sensor quality is poor:

$$C_{SP} = \begin{pmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix}$$

If the speed sensor quality is good:

$$C_{SG} = \begin{pmatrix} 0.95 & 0.05 \\ 0.10 & 0.90 \end{pmatrix}$$

Basic probability assignment for ECM is assumed as:

$m_{ECM}(yes = 0, no = 0, \Theta = 1)$ where Θ denotes the unknown case corresponding to {yes,no}.

In light of evidences, belief of "ECM" is updated. Calculation of the posterior belief is performed using Dempster's Combination of Evidence Rule.

4.2.2.1 ALGORITHMS FOR DST APPROACH

4.2.2.1.1 MEMORYLESS CASE

Algorithm for DST approach for “Memoryless Case” is given below:

1) ECM is initialized “no” for a period called “low signal period” which is shown with “t” and initialized “yes” for the rest of the period in which observations will be performed.
2) The length of the duration of observation period is determined and this value is set to k_{max} variable. ECM only changes its state once during k_{max} time steps.
3) In order to measure performance of the system, confusion matrix variables (TP, TN, FP, FN) and Accuracy, TPR, FPR variables are created. Furthermore, different threshold values are set in the variable Threshold.
4) Prior bpa for the sensor is set as $m(y=0, n=0, \Theta=1)$ since no information is given before starting observations.
5) Classification matrix is set depending on quality levels of the sensor.
6) FOR k=1: kmax (for each time step)
   a) Information from the sensor is received. If there are more than one sensor providing information, the evidences are combined by using Dempster’s Combination of Evidence.
      - If there is no information from both of the sensors, the prior belief of “ECM” does not change.
   b) Posterior belief of ECM is compared to threshold value resulting in increase in one of TP,TN,FP or FN variables (confusion matrix variables).

   END for
7) Accuracy, TPR, FPR and Cost are calculated by using confusion matrix and cost matrix variables.
8) The posterior belief of “ECM” at each time step is plotted with respect to time step k.
9) ROC curve is drawn by using TPR and FPR values.

4.2.2.1.2 CASE WITH MEMORY

4.2.2.1.2.1 ORDINARY CASE

Algorithm for DST approach for “ordinary case” is given below:

1) ECM is initialized “no” for a period called “low signal period” which is shown with “t” and initialized “yes” for the rest of the period in which observations will be performed.
2) The length of the duration of observation period is determined and this value is set to k_{max} variable. ECM only changes its state once during k_{max} time steps.
3) Prior bpa for the sensor is set as $m(y=0, n=0, \Theta=1)$ since no information is given before starting observations.
4) “Convergence Time” variable is defined as a matrice of 1x k_{max}.
5) Classification matrix is set depending on quality levels of the sensor.
6) FOR k=1: k_{max} (for each time step)
   a) Information received from the sensors is evaluated:
      - If the sensor is providing evidence, the bpa is updated in the below format:

        $$m_{SP}(y=..., n=..., \Theta=...)$$

      - If the sensor is providing does not provide evidence, the belief is assigned to unknown state and the following bpa is used:

        $$m_{SP}(y=0, n=0, \Theta=1)$$

   b) Prior beliefs are updated using Dempster’s rule of combination:

   $$m_{12}(A)=\frac{\sum_{B\cap C=A} m_1(B)m_2(C)}{1-\sum_{B\cap C=\varnothing} m_1(B)m_2(C)}$$

   c) The combined evidence is preserved in another bpa to be plotted in the end of the simulation.
   d) Initial belief of the sensor is updated so that unknown state is not “1” in the next time step.

   END for
7) Convergence Time variable is assigned with the time step at which belief exceeds 0.99.
8) Posterior belief of “ECM” at each time step is plotted with respect to time step k.

The same procedure can further be expanded so that Monte Carlo Simulation results are obtained.

4.2.2.1.2.2 LİMİTED CASE

1) ECM is initialized "no" for a period called "low signal period" which is shown with "t" and initialized "yes" for the rest of the period in which observations will be performed.
2) The length of the duration of observation period is determined and this value is set to k_{max} variable. ECM only changes its state once during k_{max} time steps.
3) Prior bpa for the sensor is set as $m(y=0, n=0, \Theta=1)$ since no information is given before starting observations.
4) "Convergence Time" variable is defined as a matrix of 1x k_{max}.
5) Classification matrix is set depending on quality levels of the sensor.
6) FOR k=1: kmax (for each time step)
   a) Information received from the sensors is evaluated:
      - If the sensor is providing evidence, the bpa is updated in the below format:

        $$m_{SP}(y=..., n=..., \Theta=...)$$

      - If the sensor does not provide evidence, the belief is assigned to unknown state and the following bpa is used:

        $$m_{SP}(y=0, n=0, \Theta=1)$$

   b) Prior beliefs are updated using Dempster's rule of combination:

      $$m_{12}(A) = \frac{\sum_{B\cap C=A} m_1(B)m_2(C)}{1-\sum_{B\cap C=\varnothing} m_1(B)m_2(C)}$$

   c) Updated ECM belief is compared to limit value at each time step and if posterior belief of ECM gets a value lower than the limit value, the lower limit is assigned to posterior belief of ECM and that value is used as prior belief of the next step.
   d) The combined evidence is preserved in another bpa to be plotted in the end of the simulation.
   e) Initial belief of the sensor is updated so that unknown state is not "1" in the next time step.

   END for
7) Convergence Time variable is assigned with the time step at which belief exceeds 0.99.
8) Posterior belief of "ECM" at each time step is plotted with respect to time step k.

The same procedure can further be expanded so that Monte Carlo Simulation results are obtained.

4.2.2.1.2.3 FINITE MEMORY CASE

Two different approaches are given in this case:

4.2.2.1.2.3.1 APPROACH-1:

1) ECM is initialized “no” for a period called “low signal period” which is shown with “t” and initialized “yes” for the rest of the period in which observations will be performed.
2) The length of the duration of observation period is determined and this value is set to k_{max} variable. ECM only changes its state once during k_{max} time steps.
3) Prior bpa for the sensor is set as $m(y=0, n=0, \Theta=1)$ since no information is given before starting observations.
4) Classification matrix is set depending on quality levels of the sensor.
5) FOR k=1: k_{max} (for each time step)
   a) Information received from the sensors is evaluated:
      - If the sensor is providing evidence, the bpa is updated in the below format:

        $$m_{SP}(y=..., n=..., \Theta=...)$$

      - If the sensor does not provide evidence, the belief is assigned to unknown state and the following bpa is used:

        $$m_{SP}(y=0, n=0, \Theta=1)$$

   b) Predictions are calculated by using Dempster’s rule of combination. Decisions are made at each time step independent of previous time steps and results are stored in a variable. For the final decision, latest three independent results (k-2,k-1,k) are combined.

   END for
6) Posterior belief of “ECM” at each time step is plotted with respect to time step k.

4.2.2.1.2.3.2 APPROACH 2

1) ECM is initialized “no” for a period called “low signal period” which is shown with “t” and initialized “yes” for the rest of the period in which observations will be performed.
2) The length of the duration of observation period is determined and this value is set to k_{max} variable. ECM only changes its state once during k_{max} time steps.

3) Prior bpa for the sensor is set as $m(y=0, n=0, \Theta=1)$ since no information is given before starting observations.

4) "Convergence Time" variable is defined as a matrix of 1x k_{max}.

5) Classification matrix is set depending on quality level of the sensor.

6) FOR k=1: k_{max} (for each time step)

   a) Information received from the sensors is evaluated:

      - If the sensor is providing evidence, the bpa is updated in the below format:

        $$m_{SP}(y=..., n=..., \Theta=...)$$

      - If the sensor does not provide evidence, the belief is assigned to unknown state and the following bpa is used:

        $$m_{SP}(y=0, n=0, \Theta=1)$$

   b) Weighted update algorithm is used where most of the belief is assigned to the latest observation.

   c) Update algorithm for Approach-2:

      $$m(k+1)=0.6\times m(k)+0.25\times m(k-1)+0.15\times m(k-2)$$

   d) Initial belief of the sensor is updated so that unknown state is less than "1" in the next time step.

   END for

7) Convergence Time variable is assigned with the time step at which belief exceeds 0.99.

8) Posterior belief of "ECM" at each time step is plotted with respect to time step k.

The same procedure can further be expanded so that Monte Carlo Simulation results are obtained. If the same procedure is applied for a predetermined number of times (i.e. 1000 times) and the beliefs of posterior "ECM" are averaged in the end, performance of the system is better understood.

4.3 PERFORMANCE CRITERIA

In order to organize classification methods and compare their performances in various cases, different techniques are defined. Broadly accepted performance measurement techniques that are applicable to the scenarios defined in parts 4.1 and 4.2 include:

Accuracy Calculation (via Confusion Matrix)

Receiver Operating Characteristics (via Confusion Matrix)

Cost Calculation (via Cost Matrix)

Before going into detailed explanation for these three methods, it is important to present definitions of “Confusion Matrix” and “Cost Matrix”.

Confusion Matrix: A matrix which shows the actual and predicted classifications for N number of different values is called a confusion matrix. A confusion matrix for N=2 is presented below:

		Predicted Class	
		Class=Yes	Class=No
Actual Class	Class=Yes	a	b
	Class=No	c	d

a	TP	:	True Positive
b	FN	:	False Negative
c	FP	:	False Positive
d	TN	:	True Negative

“a” is the number of correct predictions that an instance is positive,

“b” is the number of incorrect of predictions that an instance negative,

“c” is the number of incorrect predictions that an instance is positive, and

“d” is the number of correct predictions that an instance is negative.

FP is also known as “Type-I error” and FN is known as “Type-II” error in the literature.

Cost Matrix: A cost matrix can be defined as a modification of confusion matrix in which pre-defined costs are assigned to the cells. Positive numbers in the cells can be used as negative outcomes and negative numbers as positive outcomes. The systems that has lower cost is said to be better performing compared to the high-cost system.

	Predicted Class		
Actual Class	C(i|j)	Class=Yes	Class=No
	Class=Yes	C(yes|yes)	C(no|yes)
	Class=No	C(yes|no)	C(no|no)

C(i|j):Cost of misclassifying class j example as class i

4.3.1 ACCURACY CALCULATION

Accuracy is a widely used performance measure in classification systems. Even though it is simple and assumes both types of errors (FP and FN) have the same effect, it provides a clear comparison of systems unless the observed class is imbalanced (i.e. one of the classes is dominant).

Accuracy is calculated by dividing total correct predictions to the total number of observations:

$$\text{Accuracy}=\frac{a+d}{a+b+c+d}=\frac{TP+TN}{TP+TN+FP+FN} \tag{5.6}$$

Accuracy ranges from 0 to 1 (0% to 100%) depending on the characteristics of the classifier. The following bullets sum up main properties of accuracy measurement.

- 90% accuracy means 10% error rate,
- If the accuracy is increased from 90% to 95%, error rate is reduced by 50%.

4.3.2 RECEIVER OPERATING CHARACTERISTICS (ROC)

ROC technique involves plotting a graph by using the information provided by confusion matrix. It is useful for comparing classifiers and visualizing the information acquired.

In order to plot ROC graphs, two additional definitions are needed to be given which are namely “True Positive Rate (TPR)” and “False Positive Rate (FPR)”.

$$TPR=\frac{a}{a+b}=\frac{TP}{TP+FN} \quad FPR=\frac{c}{c+d}=\frac{FP}{FP+TN} \tag{5.7}$$

Higher TPR and lower FPR indicate better performance in classification systems.

ROC graphs are plotted by using TPR in y-axis and FPR in x-axis.

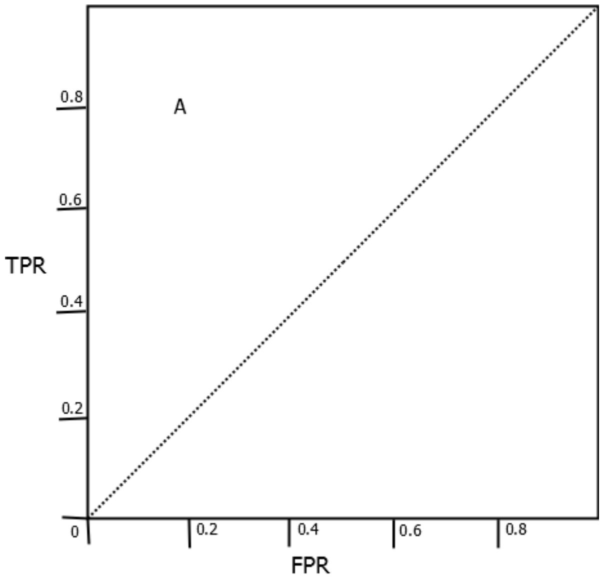

Figure 4-3: ROC Graph Template

There are important points in the ROC graph that needs further explanation:

Point (0,0) indicates the situation where the classifier never issues a positive classification. This situation results in never receiving positive errors but also never gaining true positives.

Point (1,1) indicates the situation where the classifier is always issuing positive classifications. This situation can be noted as opposite of point (0,0).

Point (0,1) indicates a perfect classification as TPR=1 and FPR=0 in this situation. TPR=1 and FPR=0 is achieved when FP=0 and FN=0 as given in formula (4.6).

The dashed line that is drawn from (0,0) to (1,1) represents randomly predicting situation. Thus, if a classifier is plotted in the lower right triangle, that classifier is performing worse than random prediction.

Curves in ROC space are obtained by applying different threshold levels to the classifier. The applied threshold levels decrease from +inf to 0.

A threshold of +inf produces the point (0, 0). As the threshold is reduced to a level between (0,1) the first positive instance is classified positive, resulting in (0, x). When the threshold is further decreased, the curve climbs to the upper right corner ending up at point (1,1).

If there is only one threshold level, the correspondent ROC graph can be drawn as below:

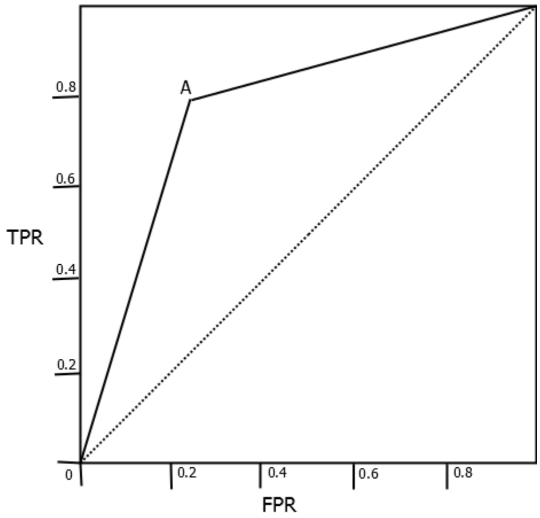

Figure 4-4: ROC Graph of a classifier with single threshold level

By using the ROC curves, highest accuracy values can be found for different threshold levels.

4.3.3 COST CALCULATION

Using cost matrices by assigning different costs to the correct and incorrect predictions provides a significant convenience while handling imbalanced data sets. This method also gives a second idea regarding classification systems where accuracy is not sufficient or controversial (such as in cases where two different threshold levels result in almost same accuracy levels).

The following example shows a case where accuracy and cost calculation methods lead to different conclusions regarding the systems.

Example:

Provided that there are two similar classifier systems, the performances can be compared according to their accuracies and costs. The cost matrix of the classification is given as:

<table>
<tr><td></td><td colspan="3">Predicted Class</td></tr>
<tr><td rowspan="3">Actual Class</td><td>C(i|j)</td><td>+</td><td>-</td></tr>
<tr><td>+</td><td>-1</td><td>100</td></tr>
<tr><td>-</td><td>1</td><td>0</td></tr>
</table>

The cost of classifying a positive class as a negative class (FN) is much higher than the cost of classifying a negative class as a positive class (FP).

The following systems provide the numbers of detection values of both systems. The comparison will be made according to these matrices.

Classifier 1

	Predicted Class		
Actual Class	C(i|j)	+	-
	+	180	48
	-	72	300

Accuracy-1=80%

Cost-1=4692

Classifier 2

	Predicted Class		
Actual Class	C(i|j)	+	-
	+	300	54
	-	6	240

Accuracy-2=90%

Cost-2=5106

Even though Accuracy-2 is higher than Accuracy-1 (error rate of the second classifier is 50% less than the first classifier), misclassifications of the second classifier result in a higher cost according to the pre-determined error costs.

CHAPTER 5

SIMULATIONS AND RESULTS

In this chapter, the scenarios that are explained in Chapter 4 are implemented and their outputs are inspected correspondingly. The performances of the scenarios are measured using performance methods such as accuracy, ROC curves and cost calculation methods. Important points such as convergence time step of the plot and average convergence time of the algorithm are added below the plots.

5.1 SCENARIO 1: ATTRIBUTE TRACKING

Firstly, attribute tracking scenario will be inspected using BNT and DST approaches respectively under the conditions given in Chapter 4.

5.1.1 BAYESIAN NETWORK THEORY APPROACH

The cases, namely "Memoryless Case" and "Case with Memory" will be examined. After presenting the plots, the accuracies will be presented in order to compare performances.

5.1.1.1 MEMORYLESS CASE

In this case, since the observed class is single type (aircraft at all times), Accuracy values will be the same with TPR values since FP and TN are zero. The cases will be examined with respect to quality of the sensor levels. Total number of observations, which is known as k_{max}, is selected as 200 and the threshold level for predicting the target as an aircraft is selected as 0.8 for the next 4 subsections:

Poor speed sensor and poor altitude sensor case

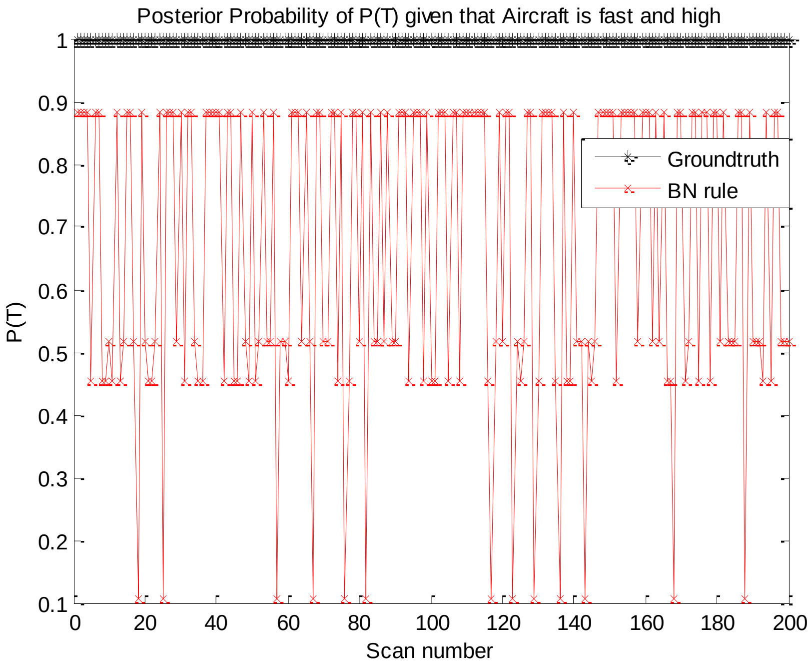

Figure 5-1: Poor speed sensor and poor altitude sensor case

P(T=aircraft|S=fast,A=high)= 0.8824
P(T=aircraft|S=fast,A=low)= 0.4545
P(T=aircraft|S=slow,A=high)= 0.5172
P(T=aircraft|S=slow,A=low)= 0.1064

Target is aircraft for k=1:200.
Accuracy of the classifier is found as 0.5626.

Poor speed sensor and good altitude sensor case

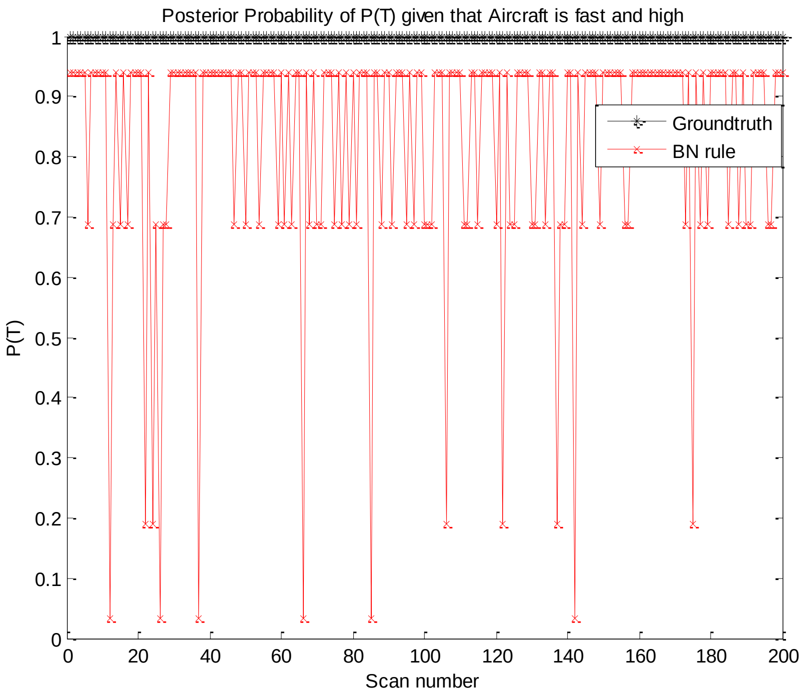

Figure 5-2: Poor speed sensor and good altitude sensor case

P(T=aircraft|S=fast,A=high)= 0.9388
P(T=aircraft|S=slow,A=high)= 0.6866
P(T=aircraft|S=fast,A=low)= 0.1905
P(T=aircraft|S=slow,A=low)= 0.0325

Accuracy of the classifier is found as 0.6959.
The accuracy is increased from 56.26% to 69.59% when only altitude sensor is improved.

Good speed sensor and poor altitude sensor case

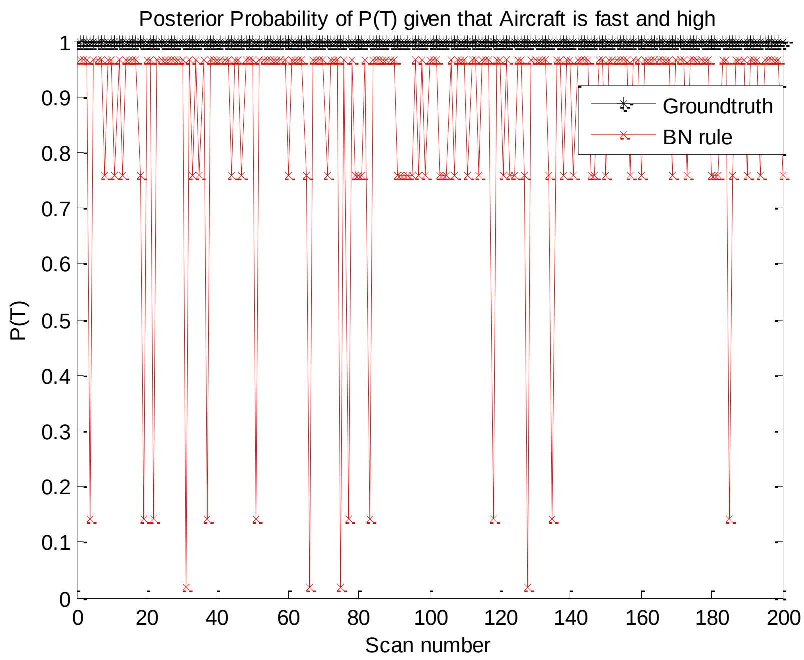

Figure 5-3: Good speed sensor and poor altitude sensor case

P(T=aircraft|S=fast,A=high)= 0.9661   P(T=aircraft|S=fast,A=low)= 0.7600
P(T=aircraft|S=slow,A=high)= 0.1429   P(T=aircraft|S=slow,A=low)= 0.0182

Accuracy of the classifier is found as 0.7108.
The accuracy is increased from 56.25% to 71% when only speed sensor is improved.

Good speed sensor and good altitude sensor case

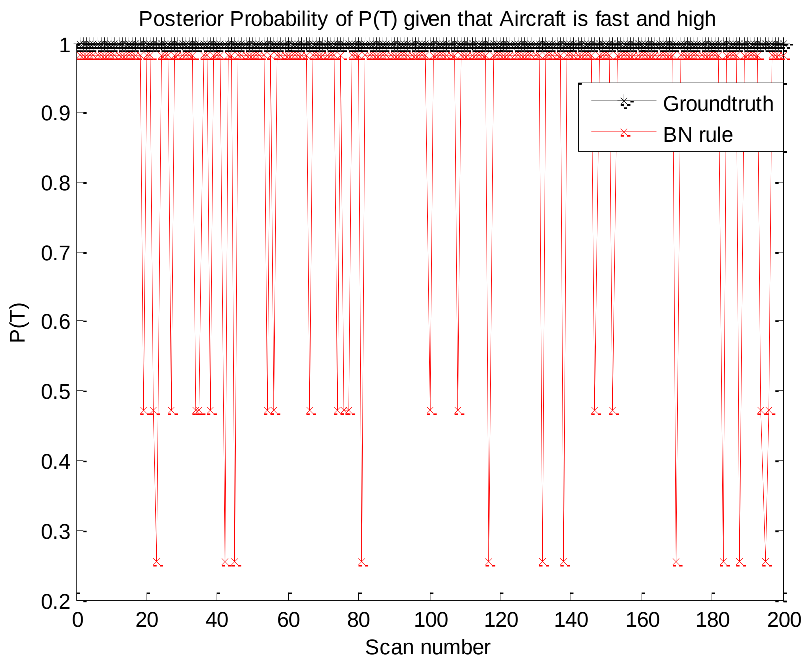

Figure 5-4: Good speed sensor and good altitude sensor case

P(T=aircraft|S=fast,A=high)= 0.9831 P(T=aircraft|S=fast,A=low)= 0.4720
P(T=aircraft|S=slow,A=high)= 0.2541 P(T=aircraft|S=slow,A=low)= 0.0052

Accuracy of the classifier=171/200= 0.8704.
The accuracy is increased to 87% when both speed and altitude sensors are improved.

5.1.1.2 CASE WITH MEMORY

In this case, TargetType probabilities are updated to be used for the following round. The observation period is chosen as k_{max}=50 since convergence is achieved much earlier than k=50.

P(TargetType=aircraft)>0.99 is set as the limit for convergence.

Poor speed sensor and poor altitude sensor case

Two random sequences are shown in order to examine convergence times:

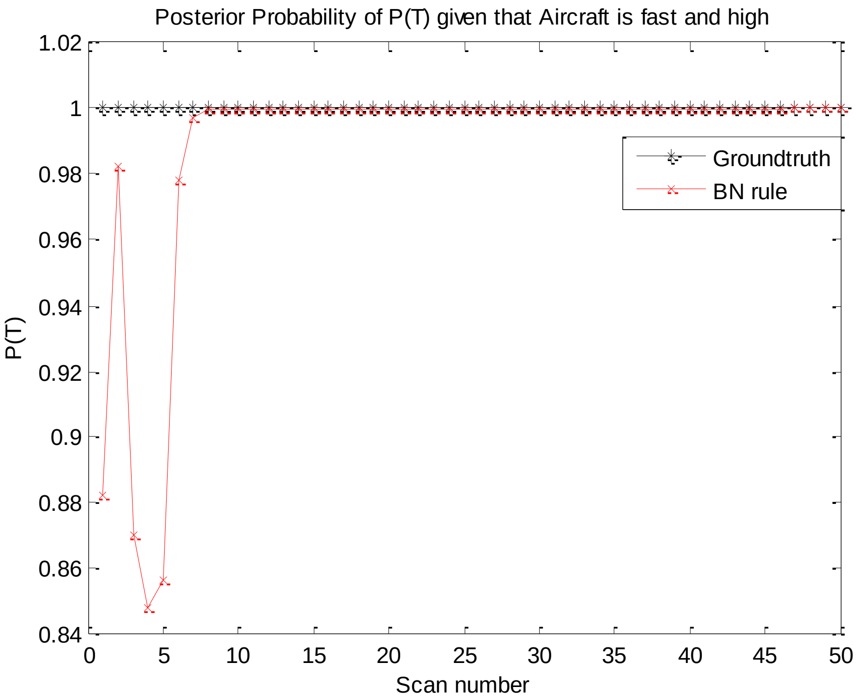

Figure 5-5: Poor speed sensor and poor altitude sensor case with memory-1

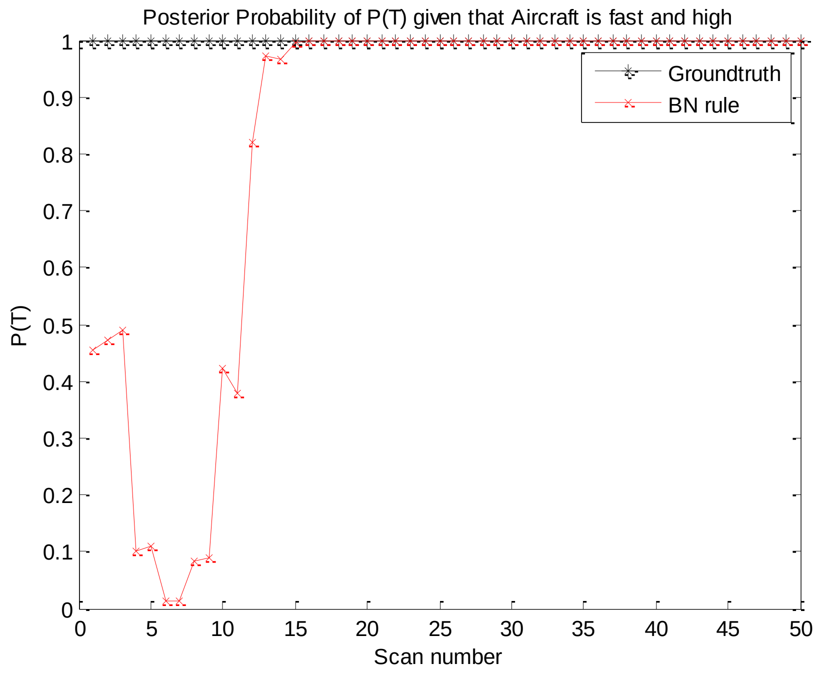

Figure 5-6: Poor speed sensor and poor altitude sensor case with memory-2

Using Monte Carlo Number=1000, average convergence value 5.66 is found for poor speed sensor and poor altitude sensor case.

Poor speed sensor and good altitude sensor case

Two random sequences are shown in order to examine convergence times:

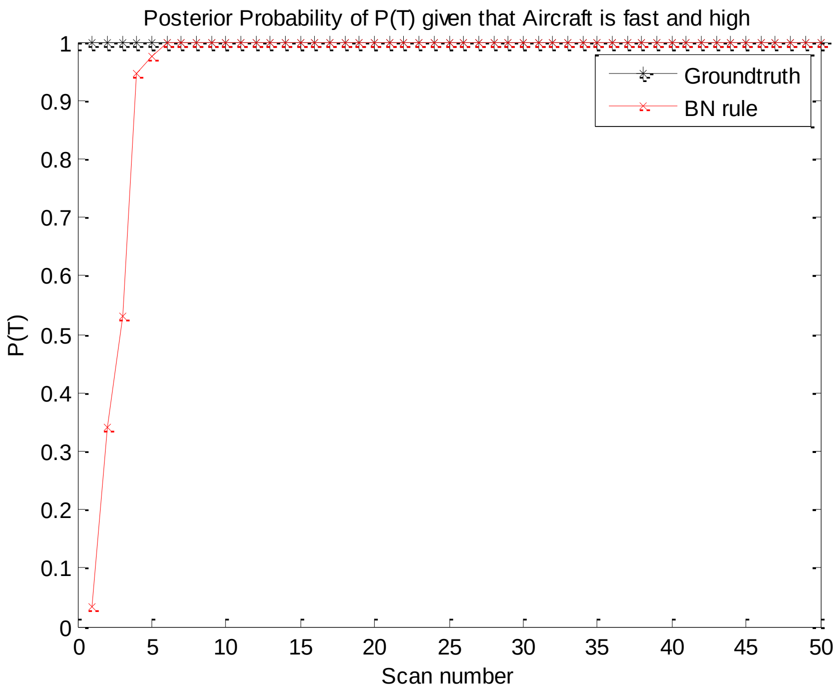

Figure 5-7: Poor speed sensor and good altitude sensor case with memory-1

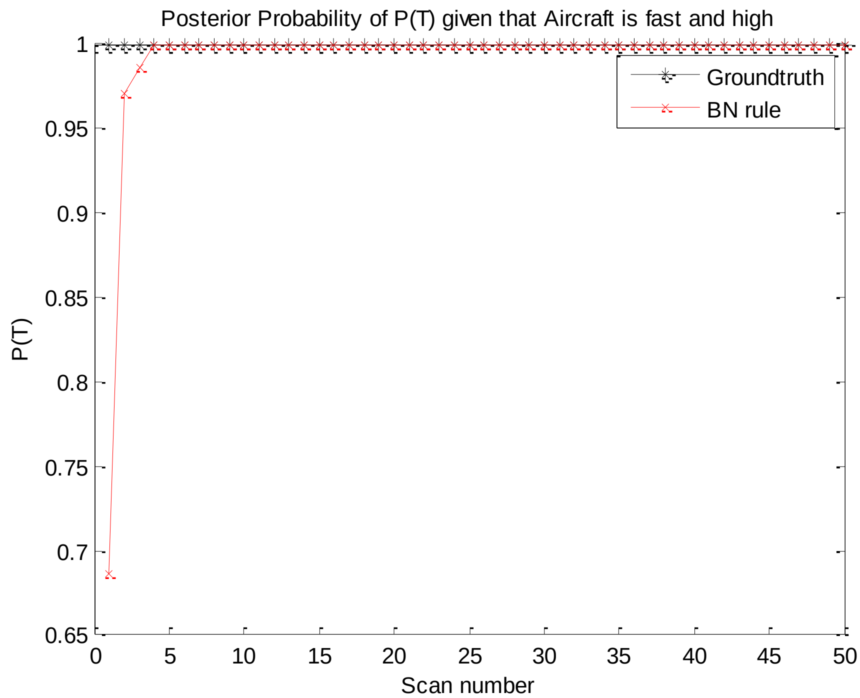

Figure 5-8: Poor speed sensor and good altitude sensor case with memory-2

Using Monte Carlo Number=1000, average convergence value 3.11 is found for poor speed sensor and good altitude sensor case. Convergence time is improved compared to the previous case.

Good speed sensor and poor altitude sensor case

Two random sequences are shown in order to examine convergence times:

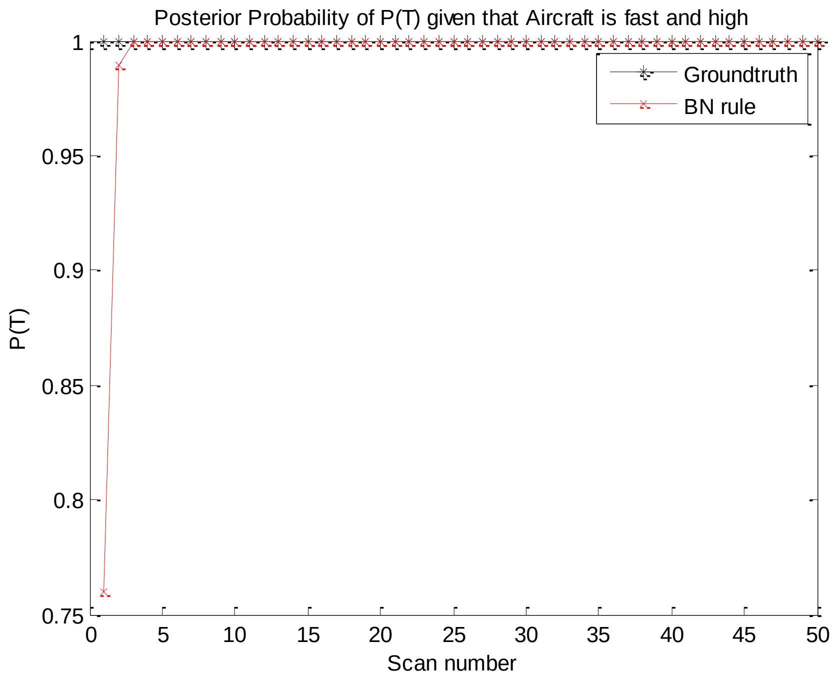

Figure 5-9: Good speed sensor and poor altitude sensor case with memory-1

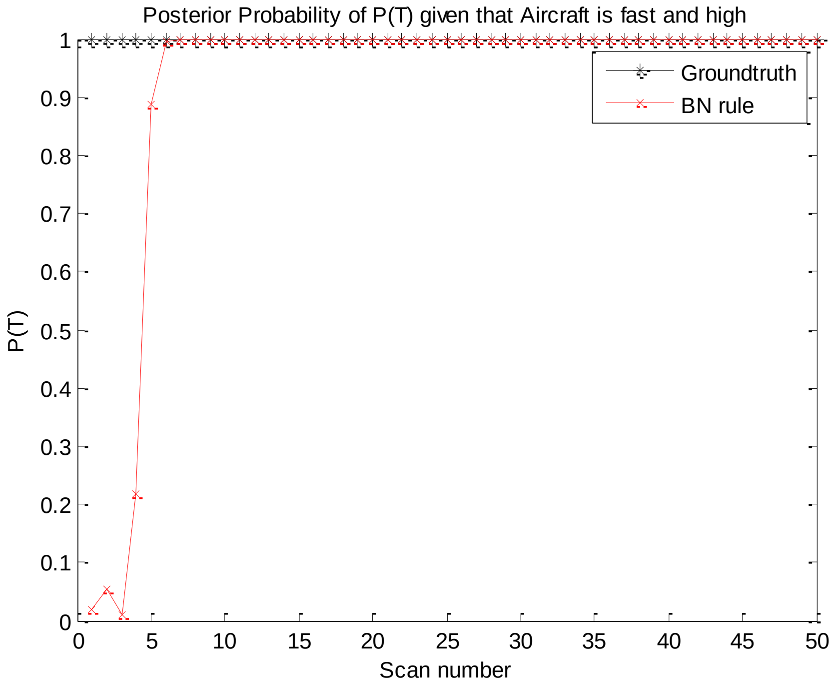

Figure 5-10: Good speed sensor and poor altitude sensor case with memory-2

Using Monte Carlo Number=1000, average convergence value 2.67 is found for good speed sensor and poor altitude sensor case.

Good speed sensor and good altitude sensor case

Two random sequences are shown in order to examine convergence times:

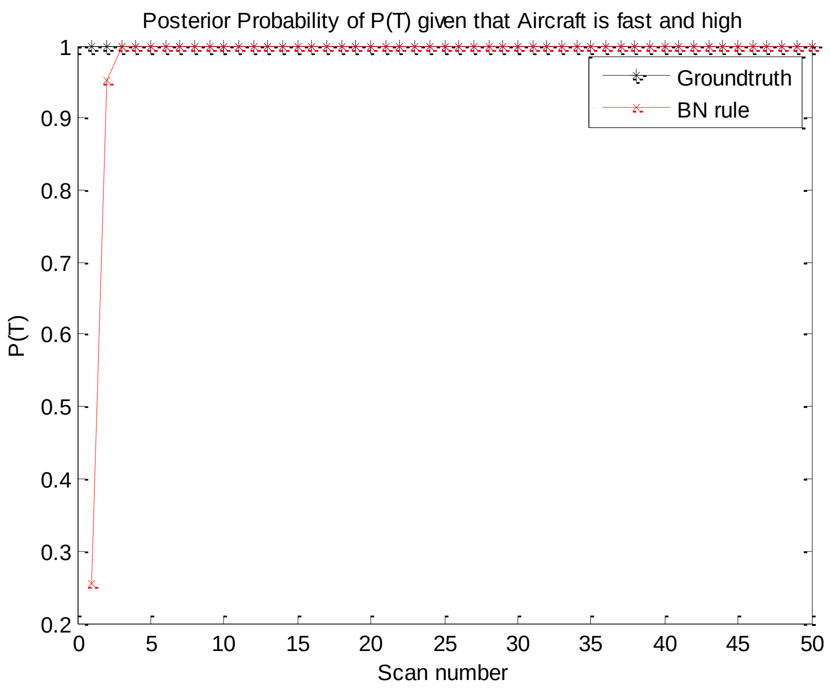

Figure 5-11: Good speed sensor and good altitude sensor case with memory-1

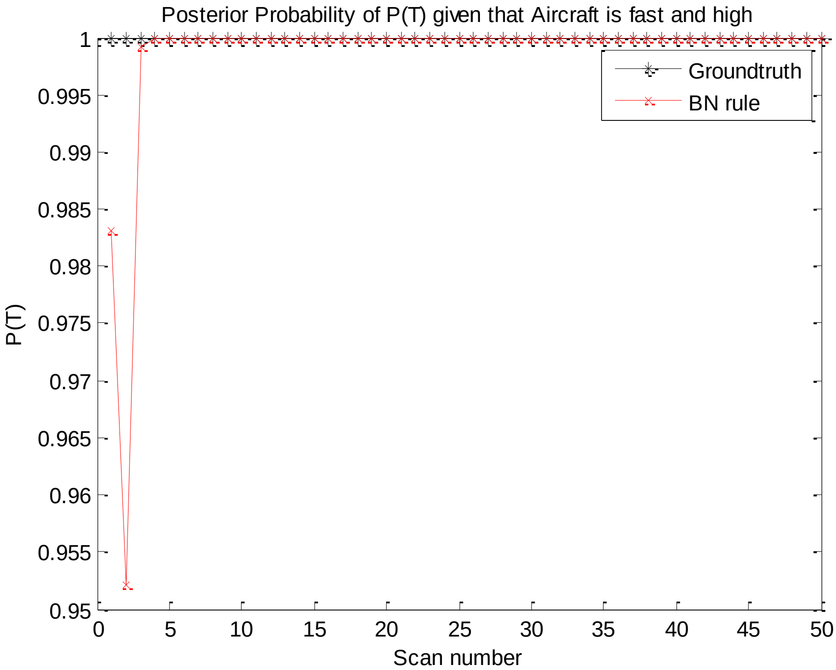

Figure 5-12: Good speed sensor and good altitude sensor case with memory-2

Using Monte Carlo Number=1000, average convergence value 2.23 is found for good speed sensor and good altitude sensor case.

5.1.2 DEMPSTER-SHAFER THEORY APPROACH

The cases, namely “memoryless case” and “case with memory” will be examined.

5.1.2.1 MEMORYLESS CASE

As explained in BNT case, Accuracy and TPR values are same in this case since the observed class is single type. The cases will be examined with respect to quality of the sensor levels. Total number of observations, which is known as k_{max}, is selected as 100 and the threshold level for predicting the target as an aircraft is selected as 0.85 for the next 4 subsections.

Poor speed sensor and poor altitude sensor case:

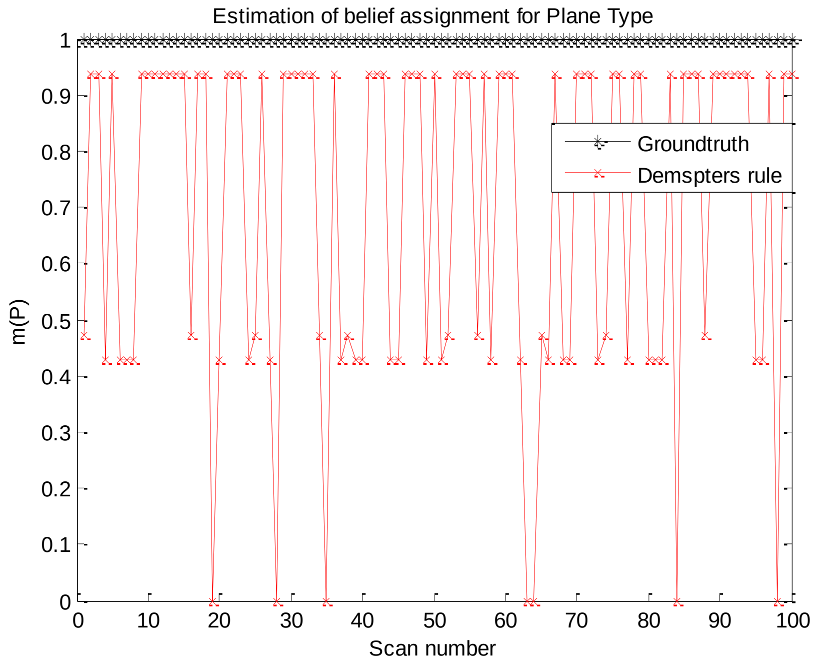

Figure 5-13: Poor speed sensor & poor altitude sensor case

Accuracy of the classifier is found as 0.5631.

Poor speed sensor and good altitude sensor case:

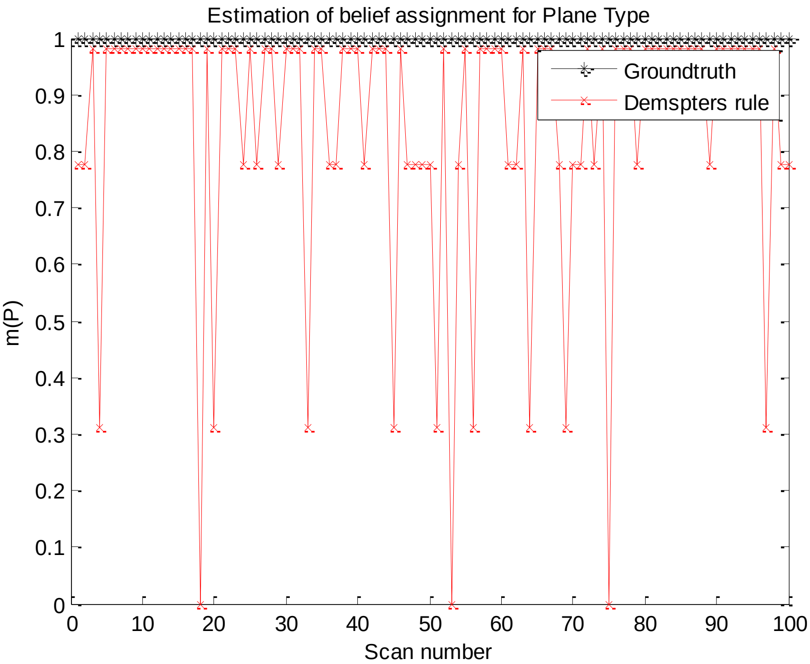

Figure 5-14: Poor speed sensor & good altitude sensor case

Accuracy of the classifier is found as 0.6868.

Good speed sensor and poor altitude sensor case:

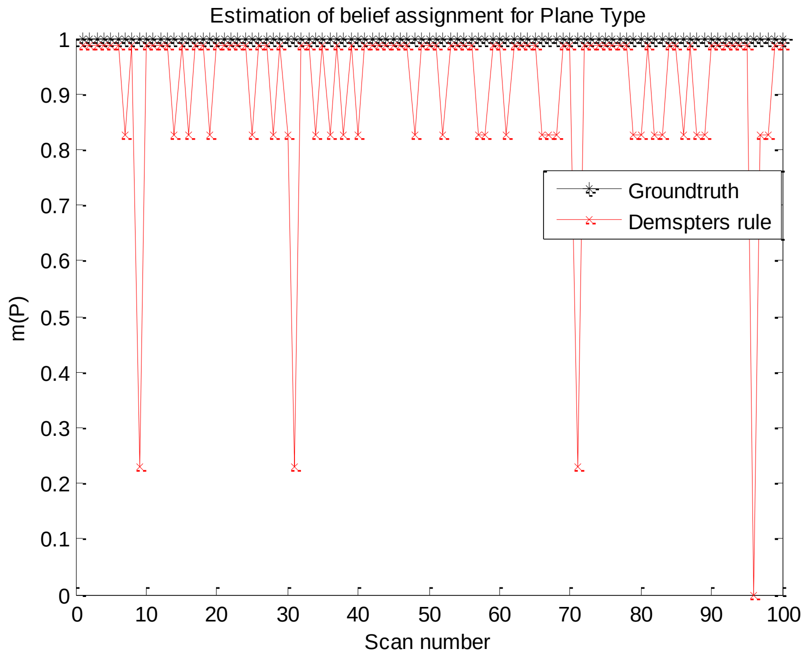

Figure 5-15: Good speed sensor & poor altitude sensor case

Accuracy of the classifier is found as 0.7179.

Good speed sensor and good altitude sensor case:

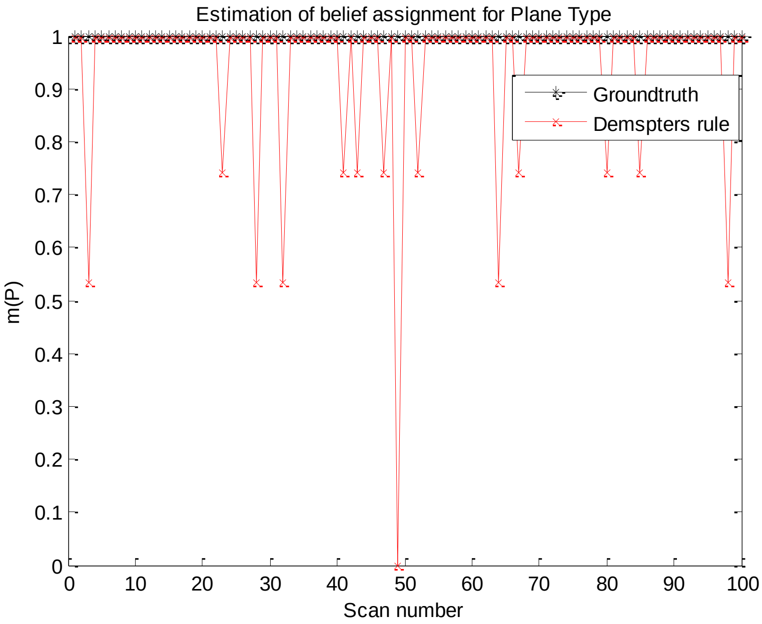

Figure 5-16: Good speed sensor & good altitude sensor case

Accuracy of the classifier is found as 0.8715.

5.1.2.2 CASE WITH MEMORY

In this case, TargetType beliefs are updated to be used for the following round. P(TargetType=aircraft)>0.99 is set as the limit for convergence.

Poor speed classifier & Poor altitude sensor case:

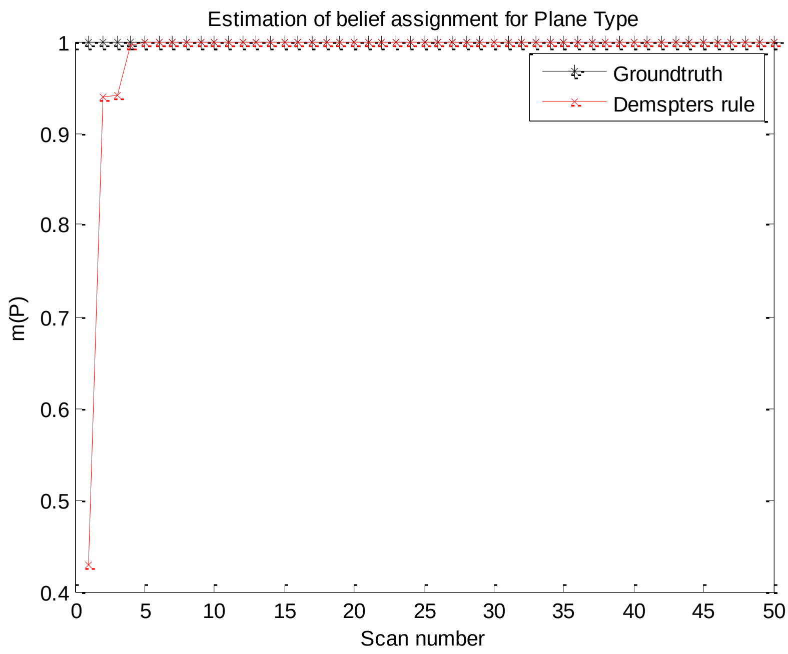

Figure 5-17: Poor speed sensor and poor altitude sensor case with memory (DST)

Average convergence time for poor speed classifier case = 4.15.

Poor speed classifier & Good altitude sensor case:

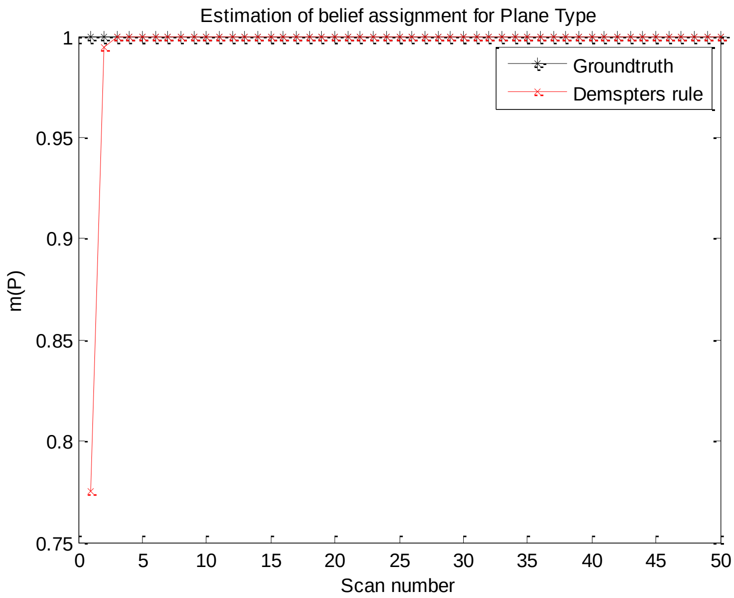

Figure 5-18: Poor speed sensor and Good altitude sensor case with memory (DST)

Average convergence time for poor speed classifier case = 2.3.

Good speed classifier & Poor altitude sensor case:

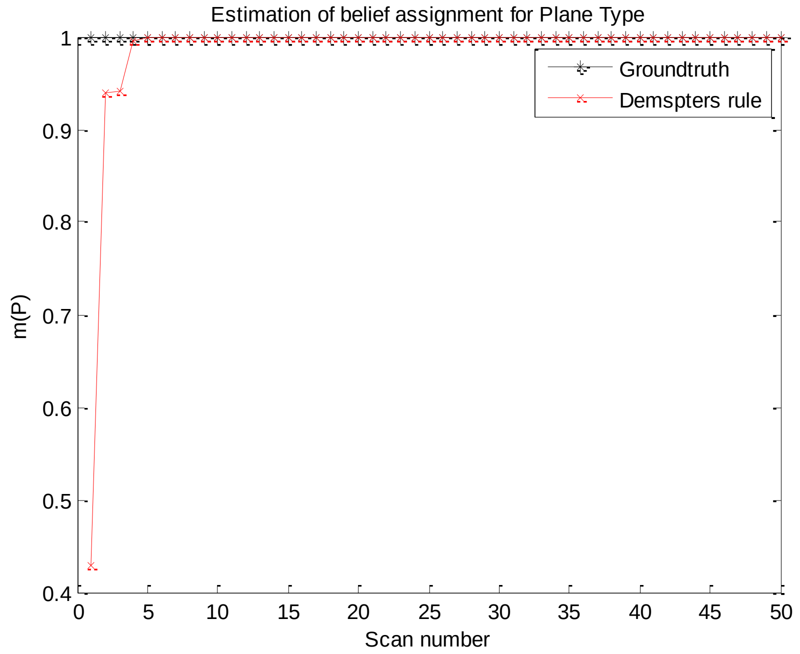

Figure 5-19: Good speed sensor and poor altitude sensor case with memory (DST)

Average convergence time for poor speed classifier case = 2.2.

Good speed classifier & Good altitude sensor case:

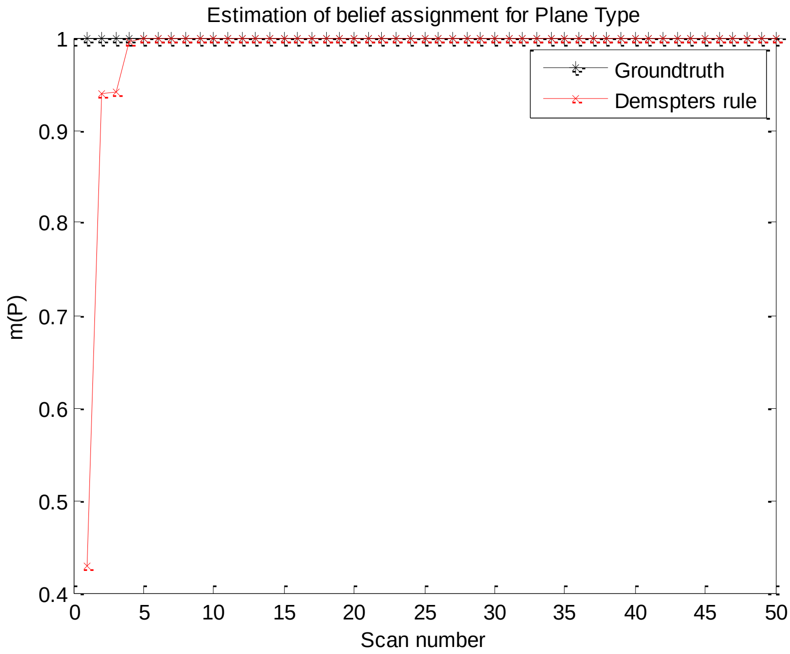

Figure 5-20: Good speed sensor and good altitude sensor case with memory (DST)

Average convergence time for poor speed classifier case = 1.15.

5.2 SCENARIO 2: ECM DETECTION

Secondly, ECM detection scenario will be inspected using BNT and DST approaches respectively under the conditions given in Chapter 4. The inspections will be done for two cases, namely "memoryless case" and "case with memory". The effect of using multiple sensors, especially for poor sensor case will be inspected. The case with memory will further be inspected with a lower limit in order to achieve convergence time improvement.

5.2.1 BAYESIAN NETWORK THEORY APPROACH

5.2.1.1 MEMORYLESS CASE

In the following sub-sections, single and multiple sensors are used respectively in both quality levels where the system does not keep prediction results of previous cases. Low signal period is set an average value (k_{low}=40) for this case. The results are plotted together with Accuracy, TPR and FPR values.

Single Poor Sensor Case: Threshold value of 0.2 is used.

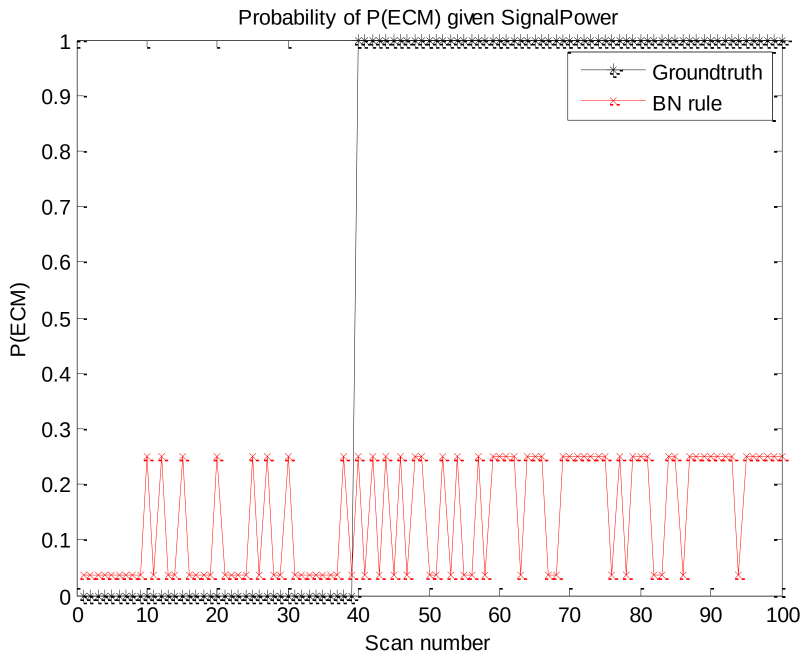

Figure 5-21: Single Poor Sensor in Memoryless ECM Case

Accuracy =0.7511, TPR =0.7498, FPR =0.2469.

When the simulation is repeated 500 times, following values are achieved:, Accuracy=0.7532, TPR=0.7510, FPR=0.2433

Single Good Sensor Case: Threshold value of 0.8 is used.

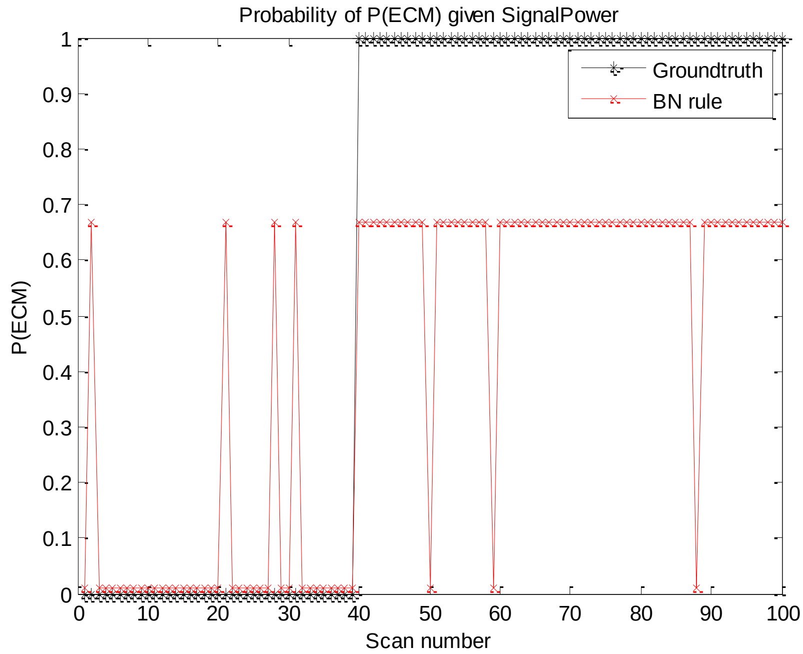

Figure 5-22: Single Good Sensor in Memoryless ECM Case

Accuracy =0.93, TPR =0.9508, FPR =0.1026.

When the simulation is repeated 500 times, following values are achieved: Accuracy=0.9330, TPR=0.9539, FPR=0.0998.

Multiple Sensors Case**:** 4 poor sensors are used.

This case will be examined in terms of accuracy, TPR, FPR and cost. ROC plot will be shown in order to observe effect of various threshold values. In order to obtain a ROC curve, following threshold values are chosen according to the values observed in the plot: Threshold=[+inf, 0.8, 0.4, 0.05, 0].

Results of first single run:

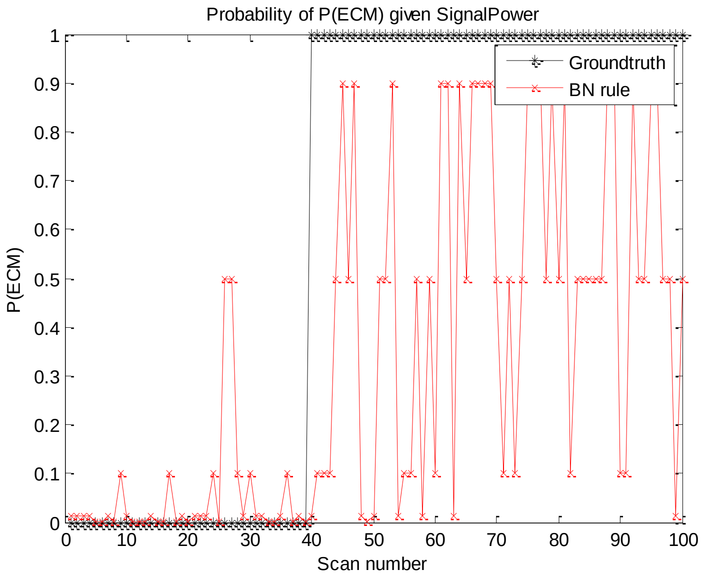

Figure 5-23: 4 Poor Sensors in Memoryless ECM Case-1

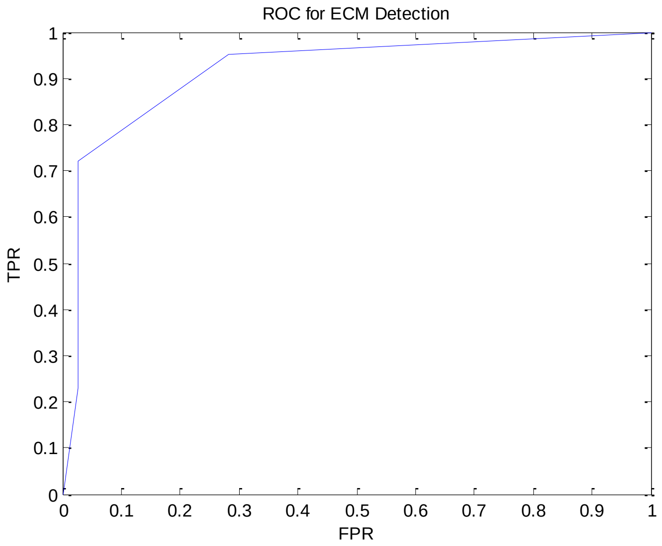

Figure 5-24: ROC Curve of 4 Poor Sensors in Memoryless ECM Case-1

Overall results can be summed up in the table below:

Table 5-1: Performance Measures-1 (BNT)

Threshold	1	0.8	0.4	0.05	0
TPR	0	0.230	0.721	0.951	1.00
FPR	0	0.026	0.026	0.282	1.00
Accuracy	0.39	0.52	0.82	0.86	0.61

Accuracy is highest when the threshold is 0.05.

Results of second single run:

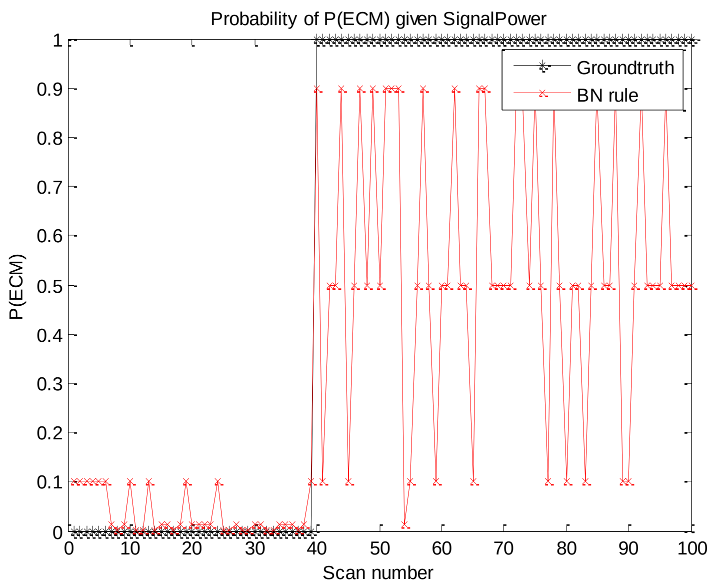

Figure 5-25: 4 Poor Sensors in Memoryless ECM Case-2

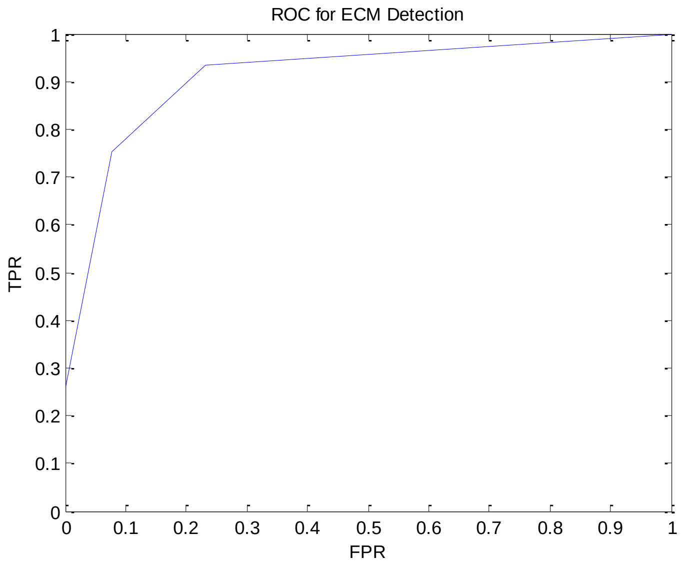

Figure 5-26: ROC Curve of 4 Poor Sensors in Memoryless ECM Case-2

Overall results can be summed up in the table below:

Table 5-2: Performance Measures-2 (BNT)

Threshold	1	0.8	0.4	0.05	0
TPR	0	0.262	0.754	0.934	1.00
FPR	0	0.000	0.077	0.231	1.00
Accuracy	0.39	0.55	0.82	0.87	0.61

Accuracy is highest when the threshold is 0.05.

Average Results when the simulation is run 100 times:

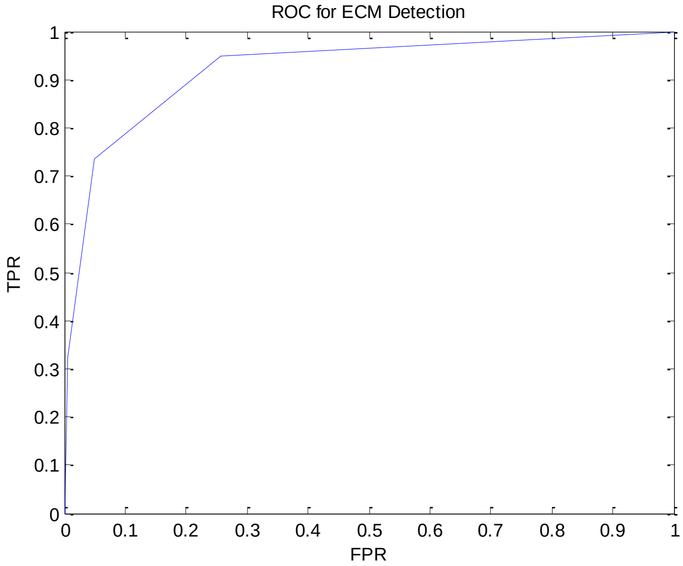

Figure 5-27: Average ROC Curve of 4 Poor Sensors in Memoryless ECM Case

Average results can be summed up in the table below:

Table 5-3: Performance Measures-Average (BNT)

Threshold	1	0.8	0.4	0.05	0
TPR	0	0.324	0.736	0.949	1.0
FPR	0	0.006	0.049	0.258	1.0
Accuracy	0.3999	0.5921	0.8223	0.8665	0.60
Cost	300050	203677	76753	18819	33989

Accuracy is maximum and Cost is minimum when the threshold is 0.05. However, setting the threshold value so low can lead to misclassifications when no target is present. As a matter of fact, this type of misclassifications can be observed from the FPR values in the corresponding table:

$$\text{FPR}(0.05) = 5 \times \text{FPR}(0.4)$$

The threshold level must be set depending upon the requirements of the system and the importance of misclassifications, namely type-I error and type-II error. If FPR is needed to be lower than a certain level, such as 0.1, then threshold must be selected as 0.4 where

Accuracy=82.23%. On the other hand, if TPR is desired to be higher than a certain limit, such as 0.9, then threshold must be set as 0.05 where Accuracy=94.9%.

Above study shows that observing only Accuracy does not provide most accurate information and several parameters should be observed together taking into account the requirements of the system.

5.2.1.2 CASE WITH MEMORY

Performances of Single and Multiple Sensors will be presented for both ordinary and limited cases. Different low limit values will be conducted in single sensor simulations in order to observe best-performing lower limit.

5.2.1.2.1 SINGLE SENSOR

During the sub-sections, classifications with only one sensor will be simulated.

5.2.1.2.1.1 ORDINARY CASE

Poor Signal Power Sensor Case

This case will be examined for different lengths of low signal periods: t_{low}=10,20,40,60,80.

ECM is not used for the first 10 time steps:

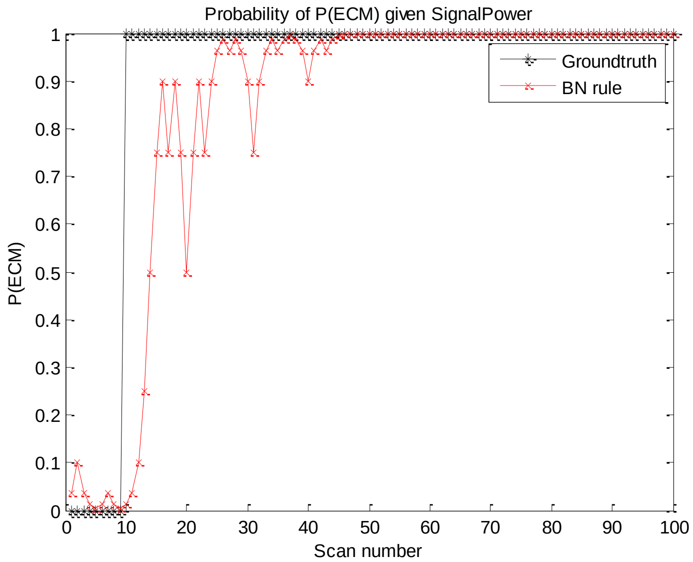

Figure 5-28: Single poor signal power sensor, ordinary case, t=10 (BNT)

Convergence time is 27 for single run. However, the predictions get lower values even after crossing over 0.99 threshold level.

When the simulation is performed 1000 times, convergence is achieved at 32.73. Therefore, average convergence time is: 22.73.

Around k=10; probability is around: 4.1e-03.

ECM is not used for the first 20 time steps:

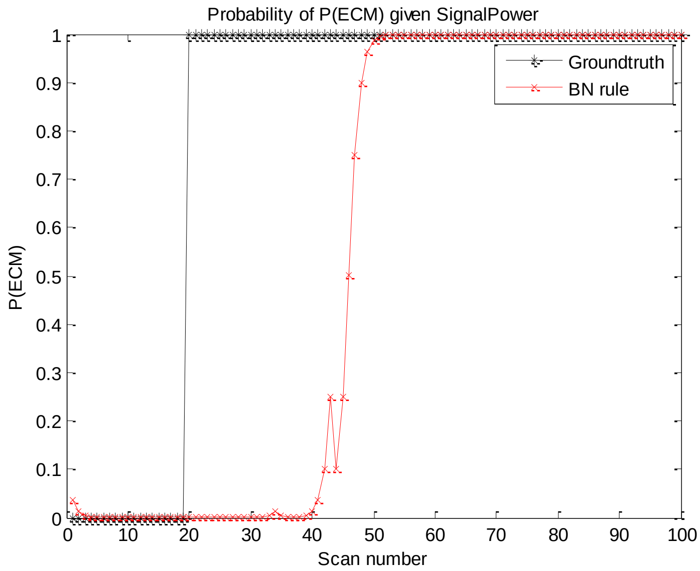

Figure 5-29: Single poor signal power sensor, ordinary case, t=20 (BNT)

Convergence time is 51-20=31 for single run.

When the simulation is performed 1000 times, convergence is achieved at 50.12. Therefore, average convergence time is: 30.12.

Around k=20; probability is around: 6.27e-07.

ECM is not used for the first 40 time steps

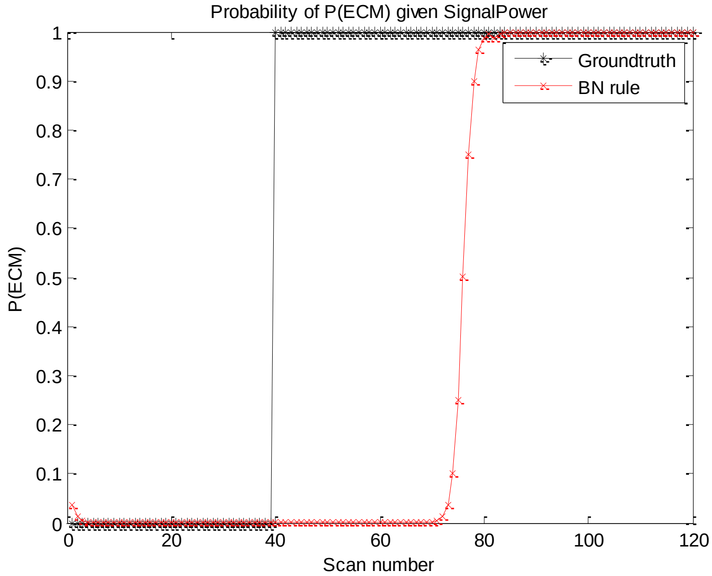

Figure 5-30: Single poor signal power sensor, ordinary case, t=40 (BNT)

Convergence time is 81-40=41 for single run.

When the simulation is performed 1000 times, convergence is achieved at 92.86. Therefore, average convergence time is: 52.86.

Around k=40; probability is around: 9.56e-11.

ECM is not used for the first 60 time steps:

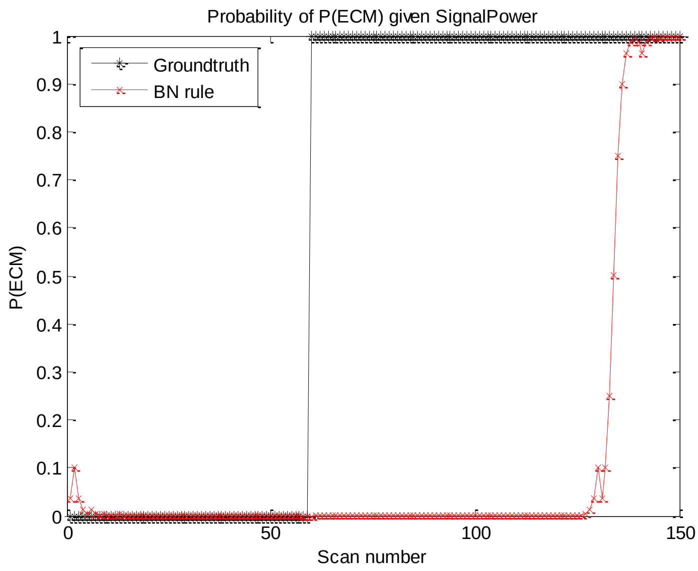

Figure 5-31: Single poor signal power sensor, ordinary case, t=60 (BNT)

Convergence time is 139-60=79 for single run.

When the simulation is performed 1000 times, convergence is achieved at 133. Therefore, average convergence time is: 73.

Around k=60; probability is around: 1.45e-14.

Good Signal Power Sensor Case

This case will be examined for t_{low}=10,20,40,60,80.

ECM is not used for the first 10 time steps

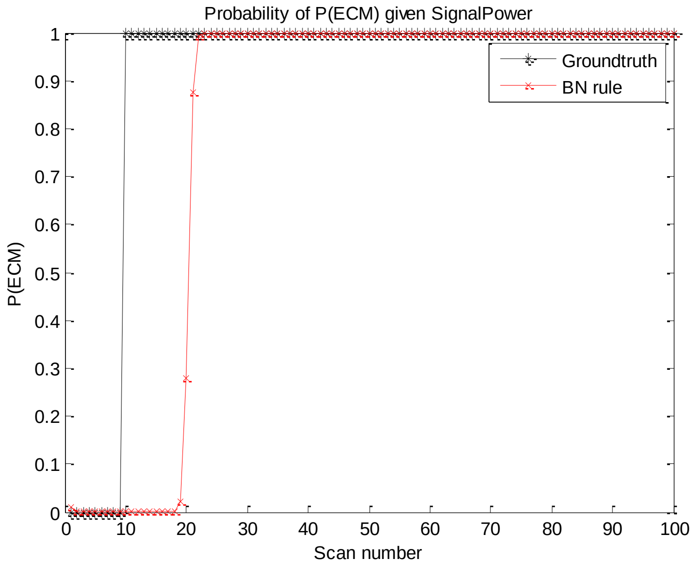

Figure 5-32: Single good signal power sensor, ordinary case, t=10 (BNT)

Convergence time is 22-10=12 for single run.

When the simulation is performed 1000 times, convergence is achieved at 20.1. Therefore, average convergence time is: 10.1.

Around k=10; probability is around: 1.76e-10.

ECM is not used for the first 20 time steps

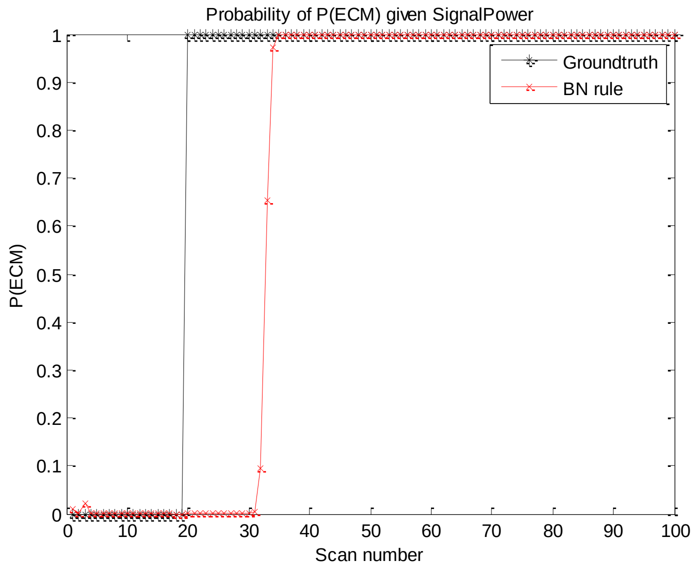

Figure 5-33: Single good signal power sensor, ordinary case, t=20 (BNT)

Convergence time is 37-20=17 for single run.

When the simulation is performed 1000 times, convergence is achieved at 38.4. Therefore, average convergence time is: 18.4.

Around k=20; probability is around: 2.94e-20.

ECM is not used for the first 40 time steps

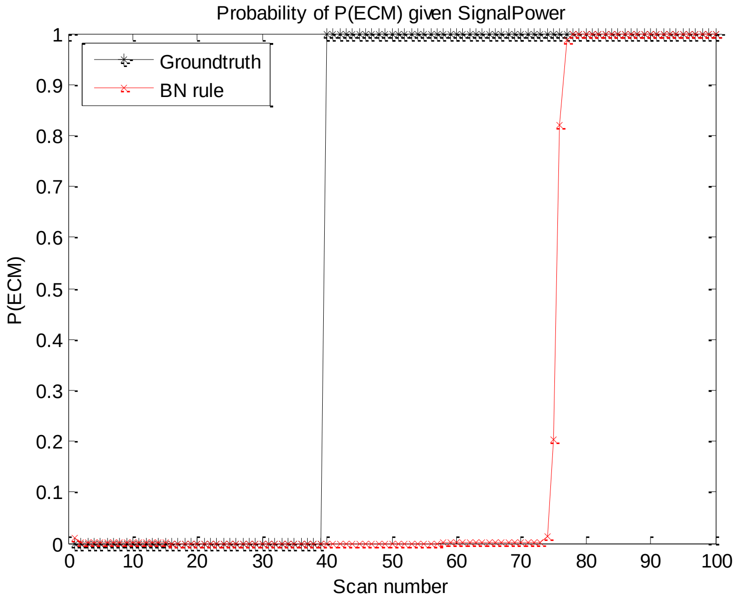

Figure 5-34: Single good signal power sensor, ordinary case, t=40 (BNT)

Convergence time is 78-40=38 for single run.

When the simulation is performed 1000 times, convergence is achieved at 75.22. Therefore, average convergence time is: 35.22.

Around k=40; probability is around: 8.21e-40.

ECM is not used for the first 60 time steps

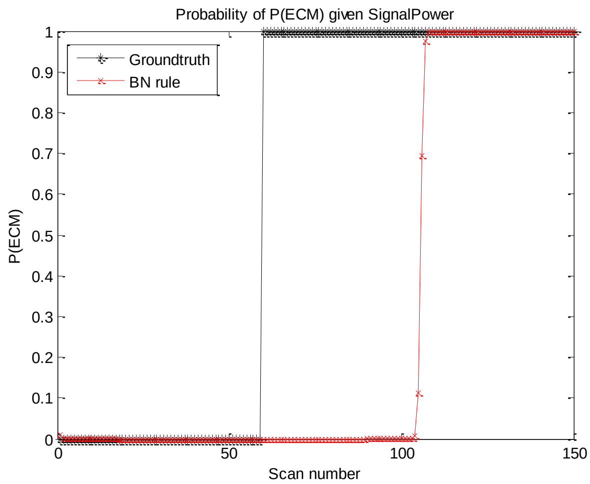

Figure 5-35: Single good signal power sensor, ordinary case, t=60 (BNT)

Convergence time is 113-60=53 for single run.

When the simulation is performed 1000 times, convergence is achieved at 111.73. Therefore, average convergence time is: 51.73.

Around k=60; probability is around: 1.1e-52.

ECM is not used for the first 80 time steps

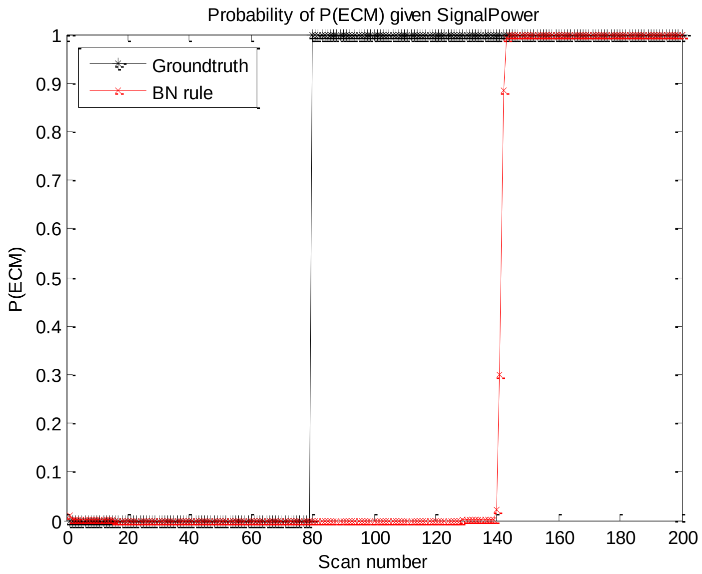

Figure 5-36: Single good signal power sensor, ordinary case, t=80 (BNT)

Convergence time is 143-80=63 for single run.

When the simulation is performed 1000 times, convergence is achieved at 148.37. Therefore, average convergence time is: 68.37.

Around k=80; probability is around: 1.1e-76.

5.2.1.2.1.2 LIMITED CASE

Poor Signal Power Sensor Case

A. Lower Limit is set: 0.01.

Two different lengths of low signal periods are used: t_{low}=10,40.

i. ECM is not used for the first 10 time steps

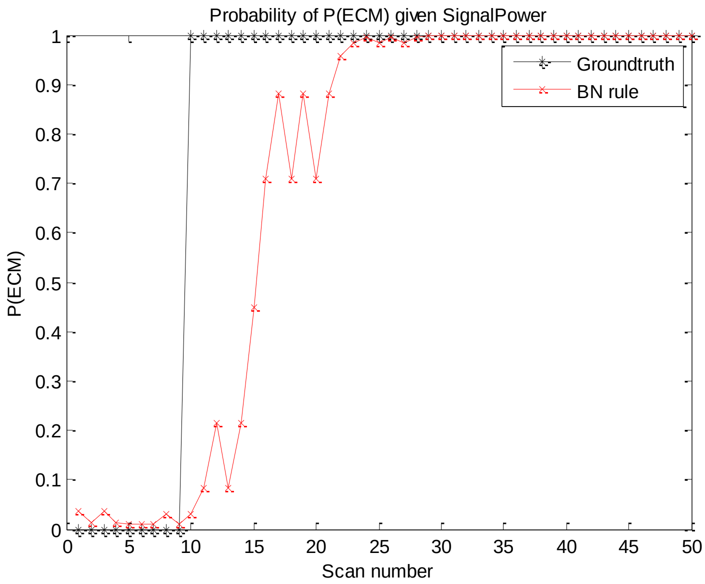

Figure 5-37: Poor signal power sensor, 0.01 limited case, t=10 (BNT)

Convergence time is 24-10=14 for single run.

Average convergence is achieved at 24.84. Then, t_{conv}=14.84.

ii. ECM is not used for the first 40 time steps

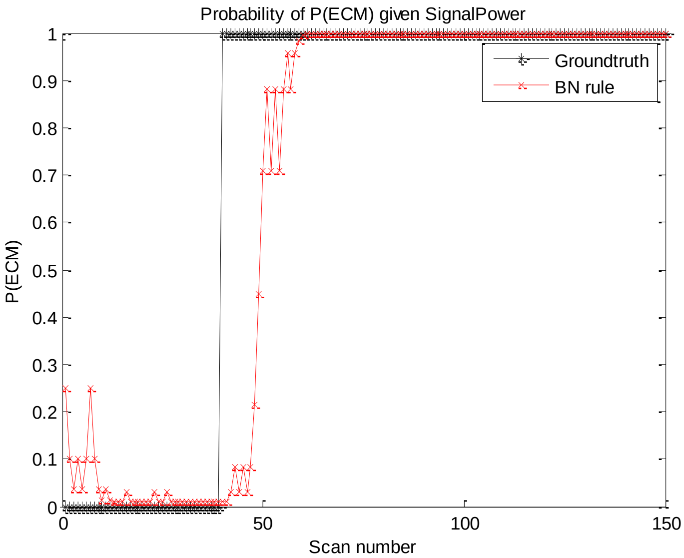

Figure 5-38: Poor signal power sensor, 0.01 limited case, t=40 (BNT)

Convergence time is 60-40=20 for a single run.

Average convergence time is achieved at 55. Then, t_{conv}=15.

The curve indicates system is not reliable for k<20 and k=[40,60].

B. Lower Limit is set: 0.001.

As in part A, two different lengths of low signal periods are used: t_{low}=10,40.

i. ECM is not used for the first 10 time steps

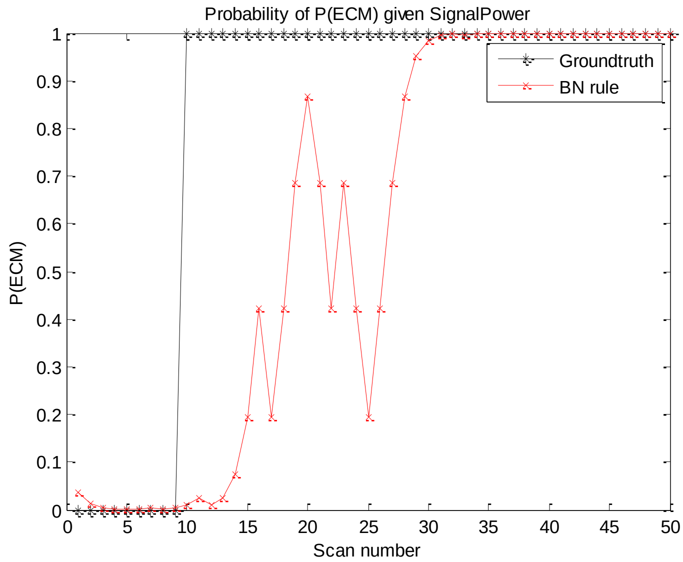

Figure 5-39: Poor signal power sensor, 0.001 limited case, t=10 (BNT)

Convergence time is 31-10=21 for single run.

Average convergence is achieved at 28.65. Then, t_{conv}=18.65.

ii. ECM is not used for the first 40 time steps

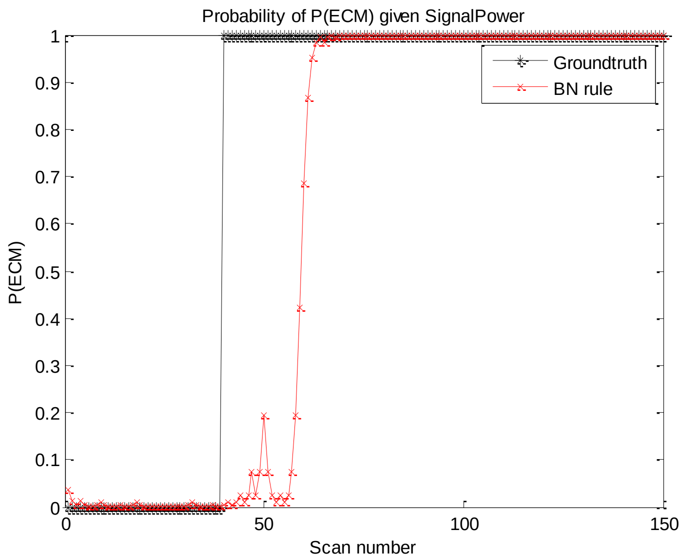

Figure 5-40: Poor signal power sensor, 0.001 limited case, t=40 (BNT)

Convergence time is 56-40=16 for single run.

Average convergence is achieved at 58.9. Then, t_{conv}=18.9.

Good Signal Power Sensor Case

A. Lower Limit is set: 0.01.

Two different lengths of low signal periods are used: t_{low}=10,40.

i. ECM is not used for the first 10 time steps

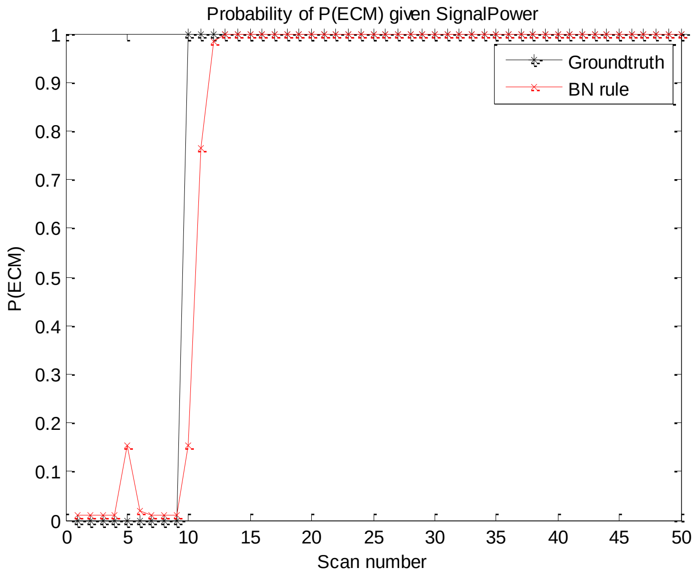

Figure 5-41: Good signal power sensor, 0.01 limited case, t=10 (BNT)

Convergence time is 13-10=3 for single run.

Average convergence is achieved at 13.39. Then, t_{conv}=3.39.

ii. ECM is not used for the first 40 time steps

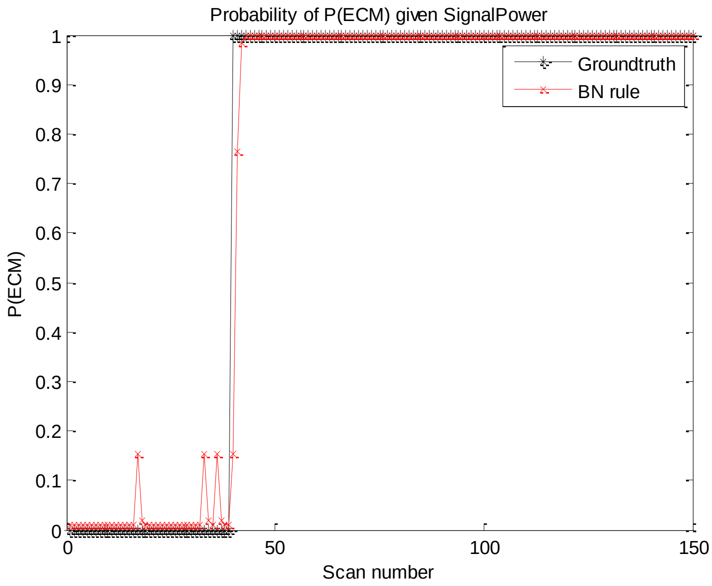

Figure 5-42: Good signal power sensor, 0.01 limited case, t=40 (BNT)

Convergence time is 43-40=3 for single run.

Average convergence is achieved at 43.49. Then, t_{conv}=3.49.

B. Lower Limit is set: 0.001.

Two different lengths of low signal periods are used: tlow=10,40.

i. ECM is not used for the first 10 time steps

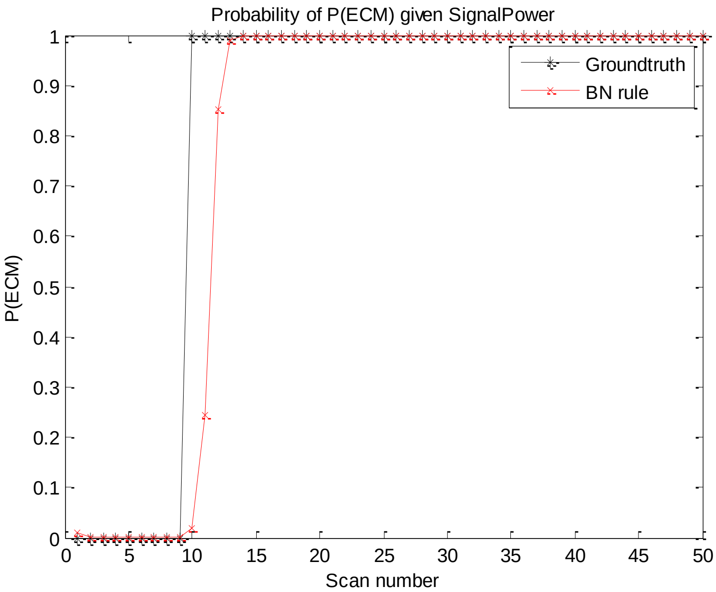

Figure 5-43: Good signal power sensor, 0.001 limited case, t=10 (BNT)

Convergence time is 13-10=3 for single run.

Average convergence is achieved at 13.8. Then, t_{conv}=3.8.

ii. ECM is not used for the first 40 time steps

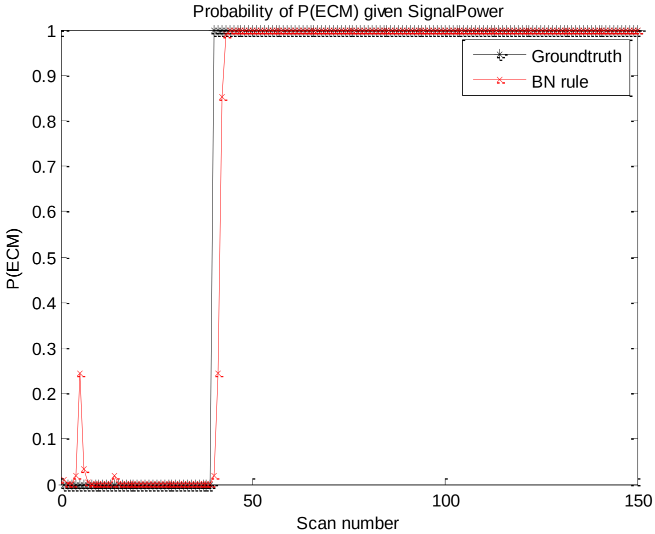

Figure 5-44: Good signal power sensor, 0.001 limited case, t=40 (BNT)

Convergence time is 43-40=3 for single run.

Average convergence is achieved at 43.74. Then, t_{conv}=3.74.

5.2.1.2.1.3 FINITE MEMORY CASE

Approach-1:

In this case, decision is made by combining three independent predictions belonging to time steps k, k-1 and k-2.

Single Poor Sensor

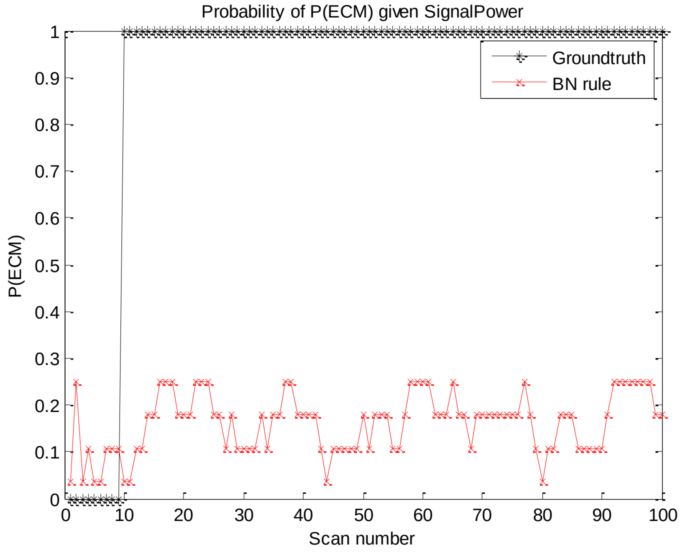

Figure 5-45: Single poor signal power sensor, finite memory case-1, t=10 (BNT)

Single Good Sensor

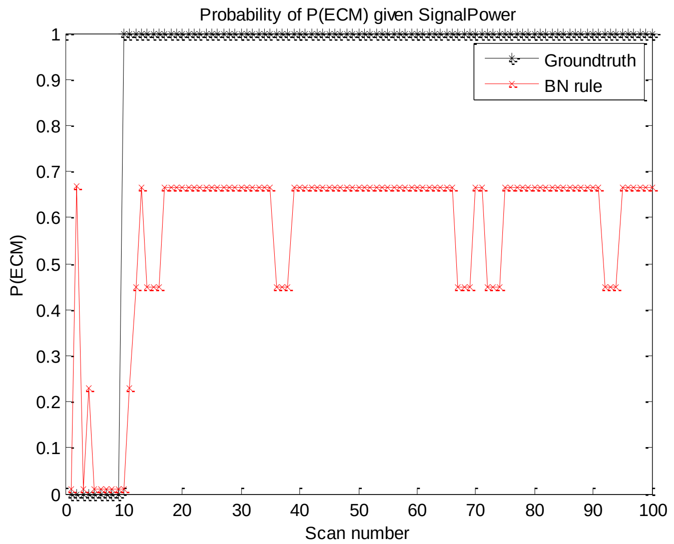

Figure 5-46: Single good signal power sensor, finite memory case-1, t=10 (BNT)

Approach-2:

Belief is updated using previous n time steps that have weighted coefficients inversely proportional to their distance to the present time step.

Update algorithm: $P(k+1) = 0.6 * P(k) + 0.25 * P(k-1) + 0.15 * P(k-2)$

Single Poor Sensor: Convergence duration is 45.

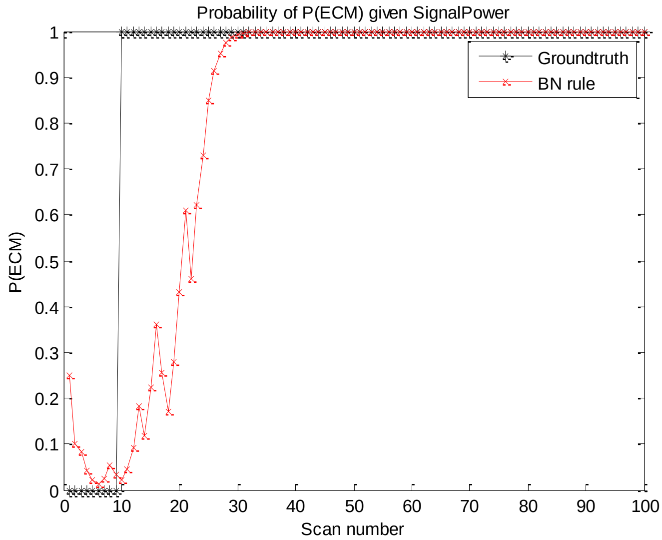

Figure 5-47: Single poor signal power sensor, finite memory case-2, single run, t=10 (BNT)

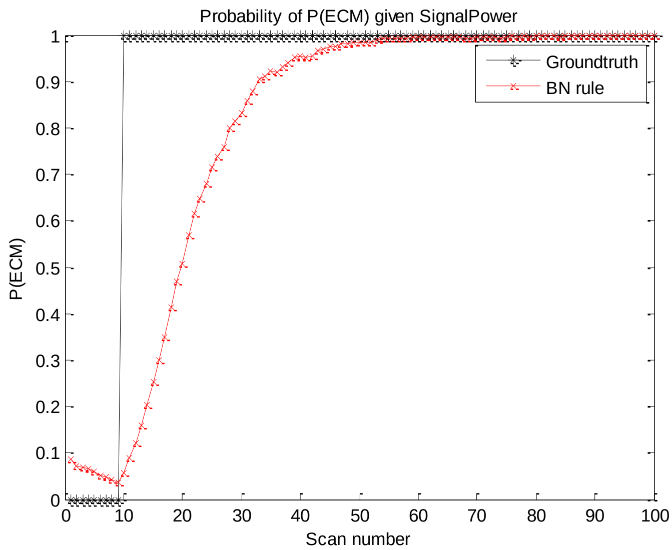

Figure 5-48: Single poor signal power sensor, finite memory case-2, Monte Carlo Average, t=10 (BNT)

Single Good Sensor:

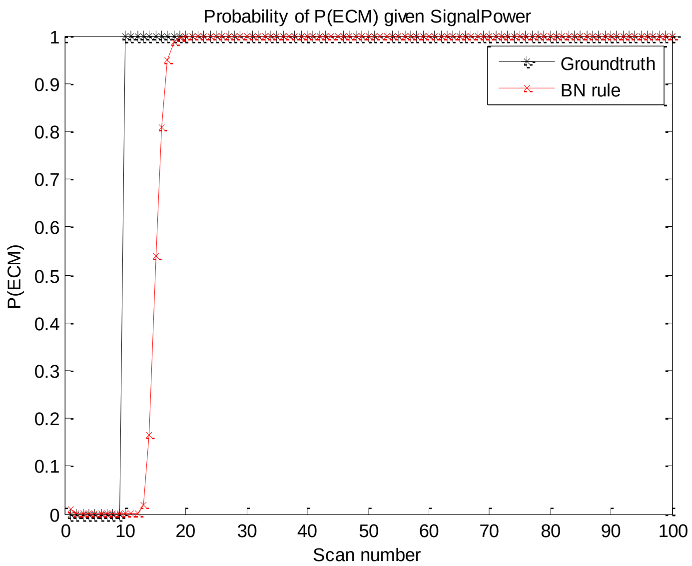

Figure 5-49: Single good signal power sensor, finite memory case-2, single run, t=10 (BNT)

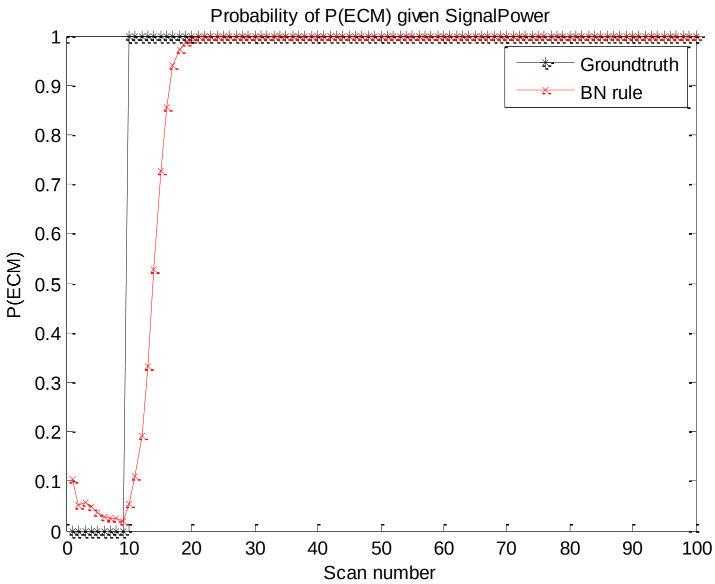

Figure 5-50: Single good signal power sensor, finite memory case-2, Monte Carlo Average, t=10 (BNT)

Convergence duration is 11.

Approach 2 makes approximately 40% improvement in good sensor case whereas the performance in poor sensor case remains almost same. However, despite the performance increase in good sensor case, the results obtained in this case are far away from the results obtained in "limited" case.

5.2.1.2.2 MULTIPLE SENSORS

During the sub-sections, classifications with different number of sensors will be simulated in order to observe the relationship between increasing sensor number and the performance of the classifier.

5.2.1.2.2.1 ORDINARY CASE

The following plots show averaged values for each time step that are obtained by Monte Carlo simulations. t_{low}=10,40 is selected in order to observe effect of Low signal periods on the classifier performance.

Poor Signal Power Sensor Case

ECM is not used for the first 10 time steps and 2 sensors are used:

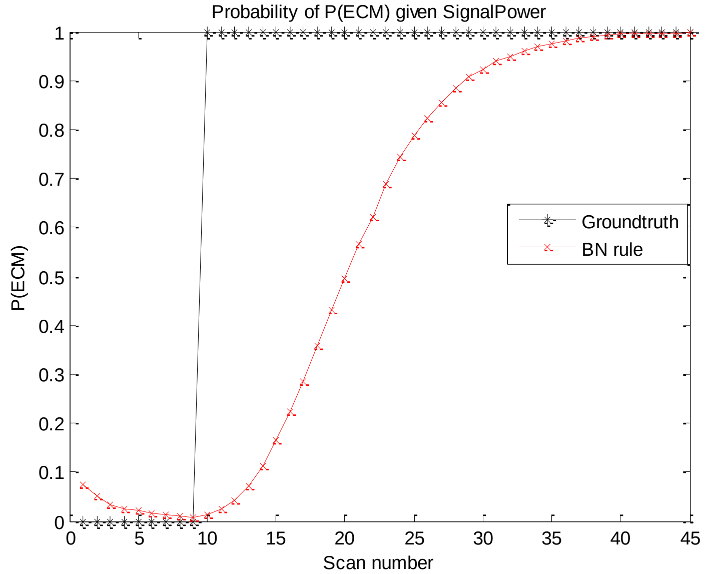

Figure 5-51: Multiple (2) poor signal power sensors, ordinary case, t=10 (BNT)

Monte Carlo simulation convergence time=30.

When the simulation is performed 1000 times, average convergence time is: 17.12.

Doubling the sensor number reduces Monte Carlo convergence time nearly 33%.

ECM is not used for the first 40 time steps and 2 sensors are used:

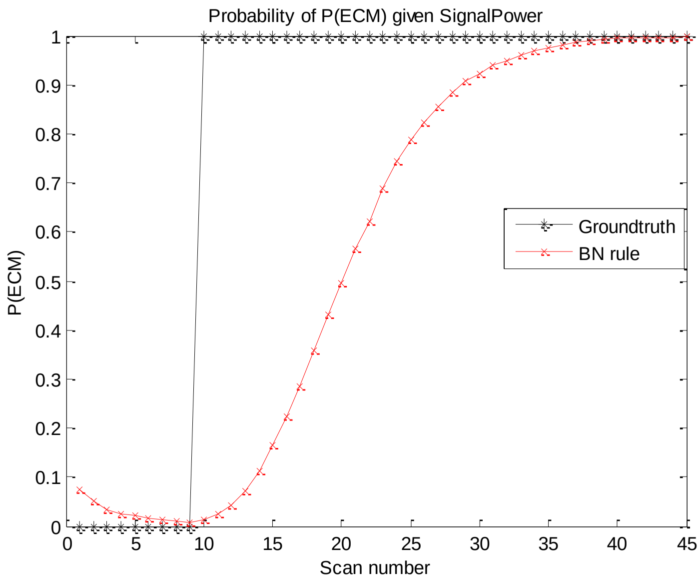

Figure 5-52: Multiple (2) poor signal power sensors, ordinary case, t=40 (BNT)

Monte Carlo simulation convergence time=71.

When the simulation is performed 1000 times, average convergence time is: 47.05.

ECM is not used for the first 10 time steps and 4 sensors are used:

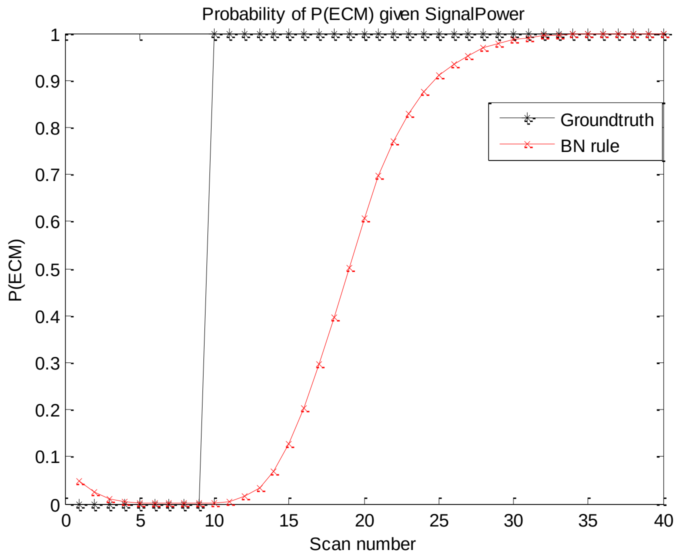

Figure 5-53: Multiple (4) poor signal power sensors, ordinary case, t=10 (BNT)

Monte Carlo simulation convergence time=22.

When the simulation is performed 1000 times, average convergence time is: 13.17.

ECM is not used for the first 40 time steps and 4 sensors are used:

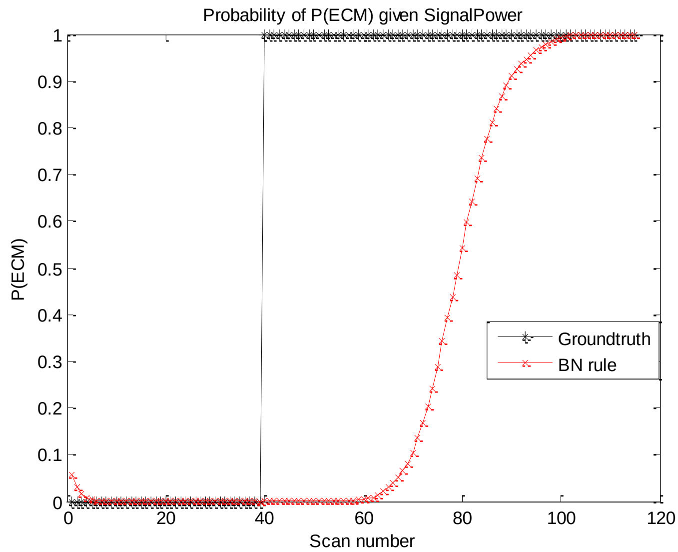

Figure 5-54: Multiple (4) poor signal power sensors, ordinary case, t=40 (BNT)

Monte Carlo simulation convergence time=60.

When the simulation is performed 1000 times, average convergence time is: 43.32.

ECM is not used for the first 10 time steps and 8 sensors are used:

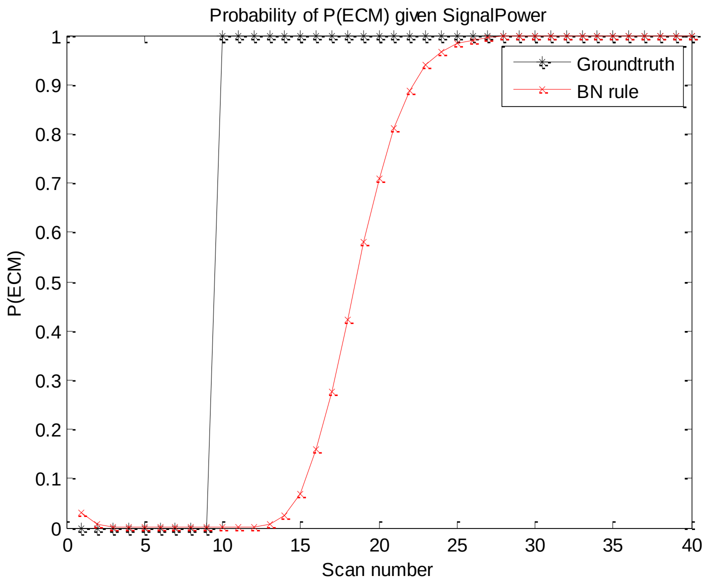

Figure 5-55: Multiple (8) poor signal power sensors, ordinary case, t=10 (BNT)

Monte Carlo simulation convergence time=17.

When the simulation is performed 1000 times, average convergence time is: 11.40.

ECM is not used for the first 40 time steps and 8 sensors are used:

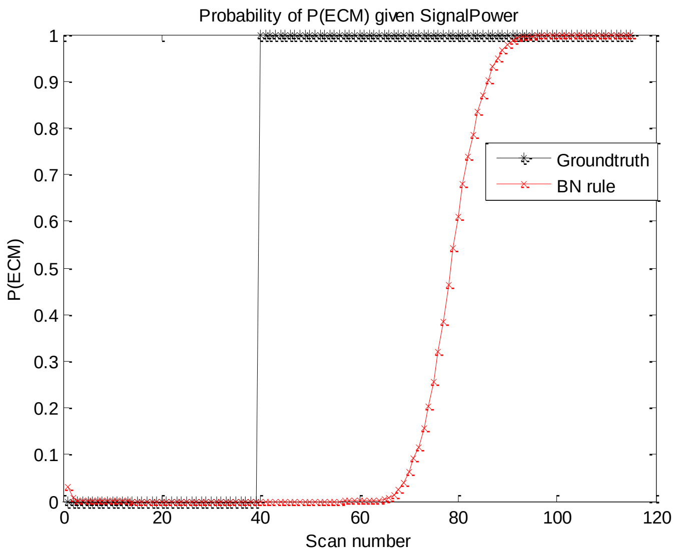

Figure 5-56: Multiple (8) poor signal power sensors, ordinary case, t=40 (BNT)

Monte Carlo simulation convergence time=53.

When the simulation is performed 1000 times, average convergence time is: 41.07.

Good Signal Power Sensor Case

ECM is not used for the first 10 time steps and 2 sensors are used:

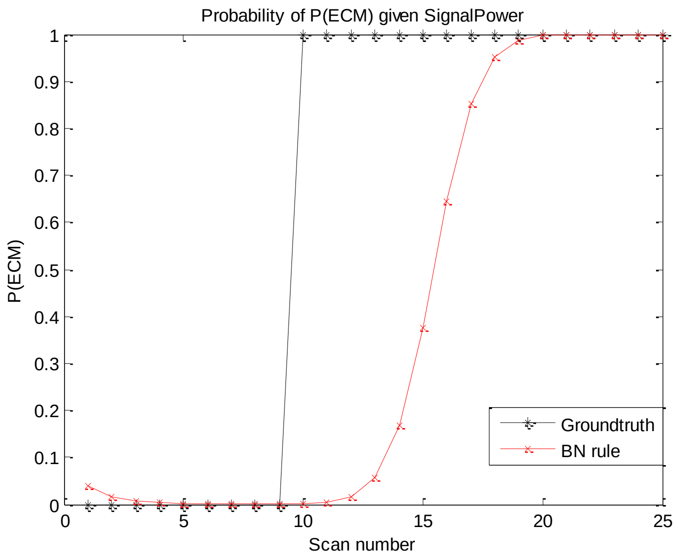

Figure 5-57: Multiple (2) good signal power sensors, ordinary case, t=10 (BNT)

Monte Carlo simulation convergence time=11.

When the simulation is performed 1000 times, average convergence time is: 7.97.

ECM is not used for the first 40 time steps and 2 sensors are used:

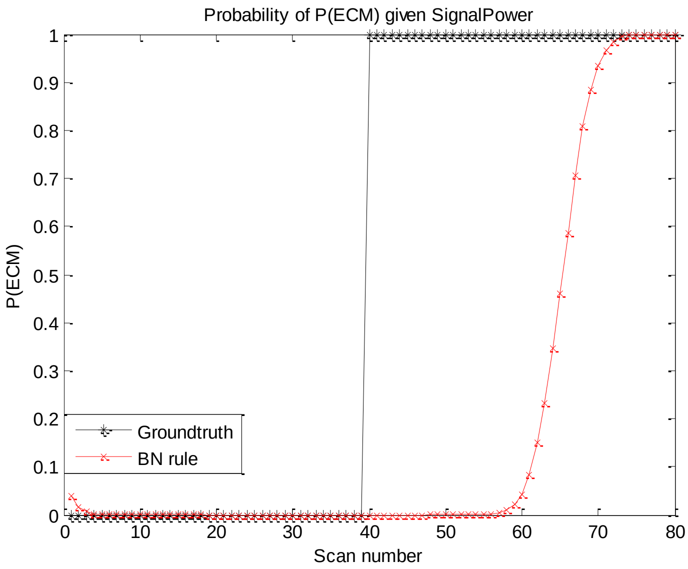

Figure 5-58: Multiple (2) good signal power sensors, ordinary case, t=40 (BNT)

Monte Carlo simulation convergence time=34.

When the simulation is performed 1000 times, average convergence time is: 27.34.

ECM is not used for the first 10 time steps and 4 sensors are used:

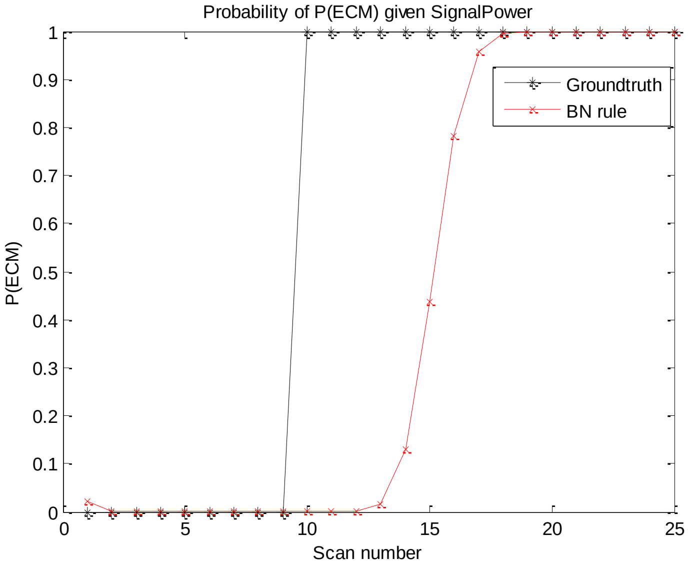

Figure 5-59: Multiple (4) good signal power sensors, ordinary case, t=10 (BNT)

Monte Carlo simulation convergence time=9.

When the simulation is performed 1000 times, average convergence time is: 7.09.

ECM is not used for the first 40 time steps and 4 sensors are used:

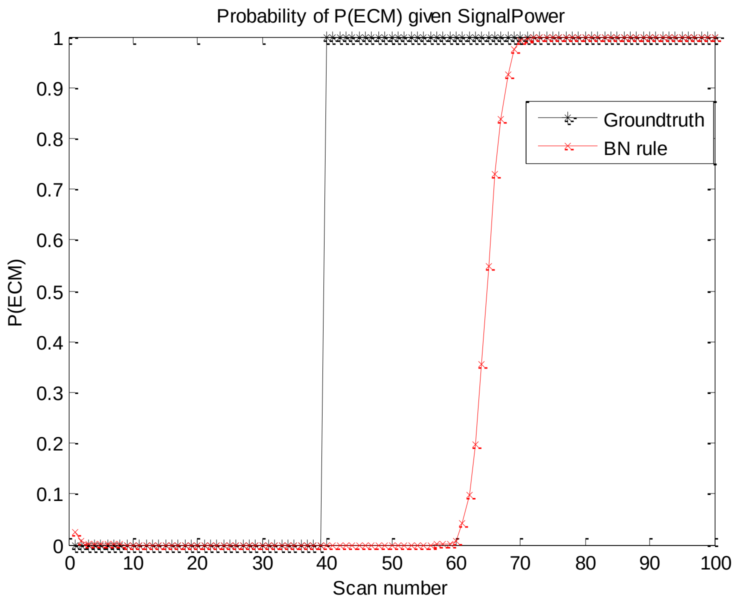

Figure 5-60: Multiple (4) good signal power sensors, ordinary case, t=40 (BNT)

Monte Carlo simulation convergence time=31.

When the simulation is performed 1000 times, average convergence time is: 26.84.

ECM is not used for the first 10 time steps and 8 sensors are used:

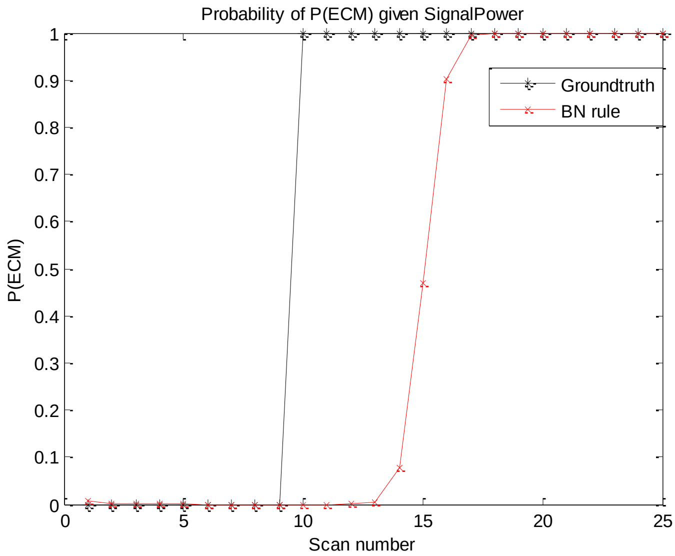

Figure 5-61: Multiple (8) good signal power sensors, ordinary case, t=10 (BNT)

Monte Carlo simulation convergence time=8.

When the simulation is performed 1000 times, average convergence time is: 6.77.

ECM is not used for the first 40 time steps and 8 sensors are used:

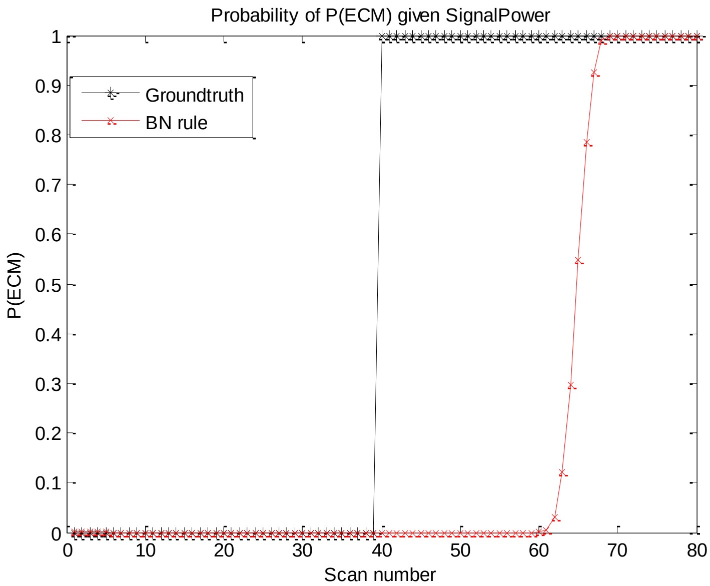

Figure 5-62: Multiple (8) good signal power sensors, ordinary case, t=40 (BNT)

Monte Carlo simulation convergence time=30.

When the simulation is performed 1000 times, average convergence time is: 26.7.

Increasing the sensor number used in classifiers increases the performance of the system by reducing time steps needed to converge. However, convergence times do not decrease evenly when sensor number is doubled at each case. When Monte Carlo results are taken into account, as sensor number is increased to 2, convergence time is decreased around 25-38% whilst this ratio is around 7-10% when 16 sensors are used.

5.2.1.2.2.2 LIMITED CASE

In limited case, lower limit is set to 0.001 in order to reduce convergence time of the classifier and make convergence time independent of the low signal period. Thus, after necessary simulations are observed, it will be seen that overall performance of the classifier is dramatically increased via implying lower limit together with multiple sensors.

Low signal periods are chosen as t_{low}=10,40 while studying performances of 4 and 8 sensors.

Poor Signal Power Sensor Case

ECM is not used for the first 10 time steps and 4 sensors are used:

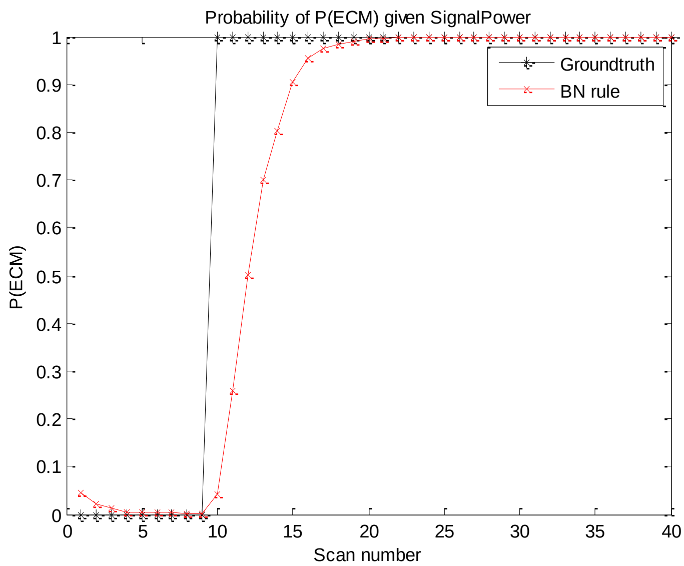

Figure 5-63: Multiple (4) poor signal power sensors, limited case, t=10 (BNT)

Monte Carlo simulation convergence time=10.

ECM is not used for the first 40 time steps and 4 sensors are used:

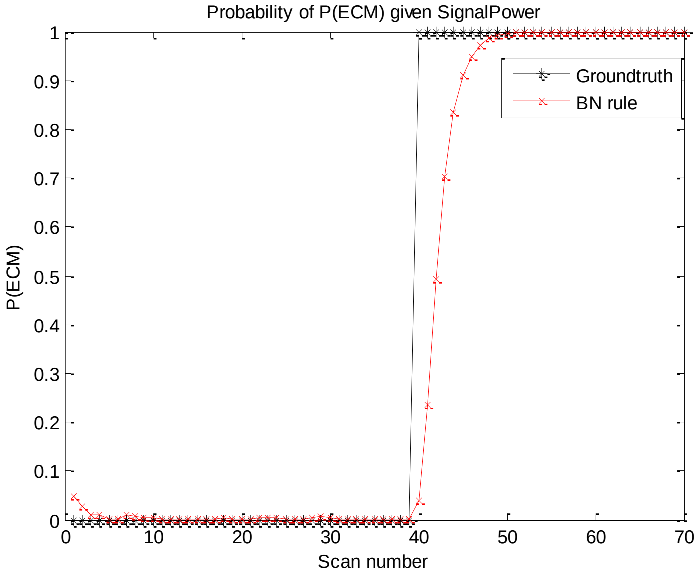

Figure 5-64: Multiple (4) poor signal power sensors, limited case, t=40 (BNT)

Monte Carlo simulation convergence time=10.

ECM is not used for the first 10 time steps and 8 sensors are used:

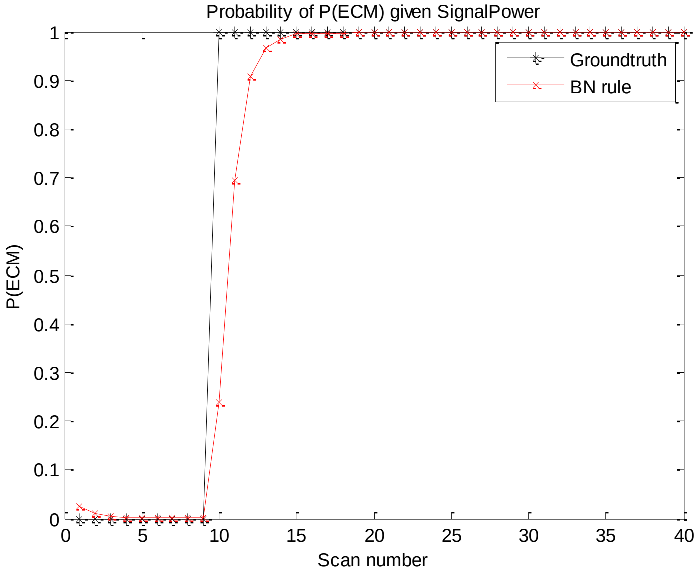

Figure 5-65: Multiple (8) poor signal power sensors, limited case, t=10 (BNT)

Monte Carlo simulation convergence time=6.

ECM is not used for the first 40 time steps and 8 sensors are used:

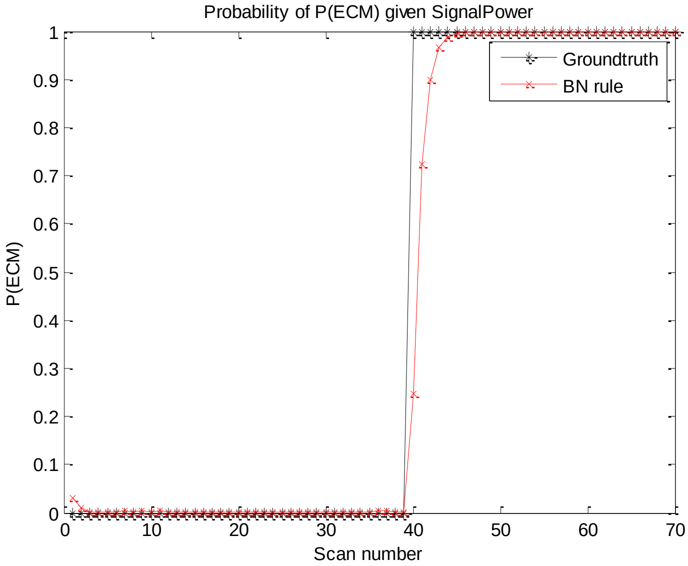

Figure 5-66: Multiple (8) poor signal power sensors, limited case, t=40 (BNT)

Monte Carlo simulation convergence time=6.

Applying a lower limit to the belief function has a dramatic effect on the overall performance as the convergence times become independent of low signal periods (t_{low}) and reduce significantly.

When 4 poor sensors are used instead of one, convergence time reduces from 34 to 10. This is equivalent to saying that system converges 70.6% faster than single poor sensor case.

Furthermore, when sensor level is increased to 8, system converges 40% faster than 4-sensor case and 82.35% faster than single sensor case.

Good Signal Power Sensor Case

ECM is not used for the first 10 time steps and 4 sensors are used:

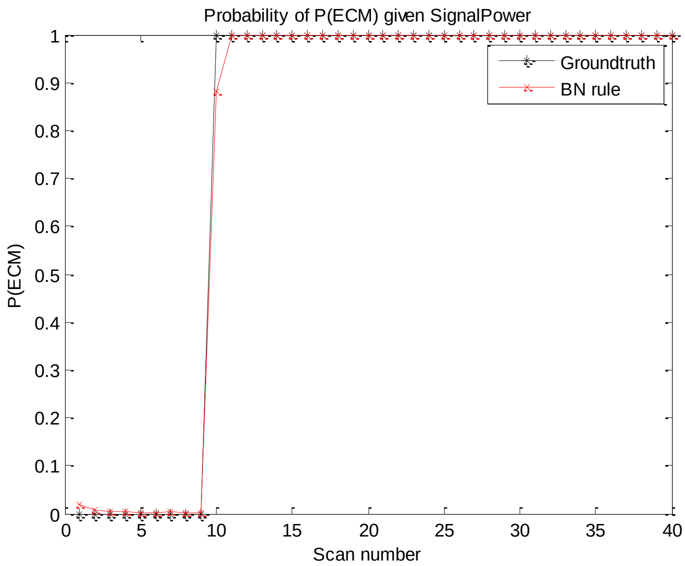

Figure 5-67: Multiple (4) good signal power sensors, limited case, t=10 (BNT)

Monte Carlo simulation convergence time=2.

ECM is not used for the first 40 time steps and 4 sensors are used:

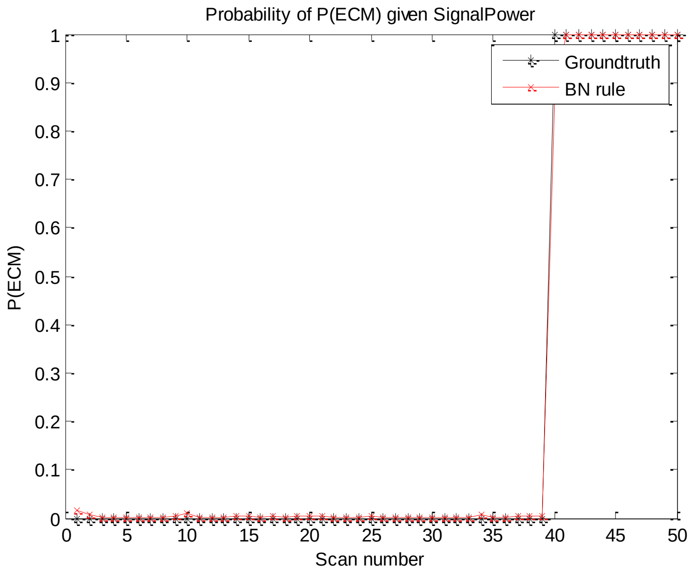

Figure 5-68: Multiple (4) good signal power sensors, limited case, t=40 (BNT)

Monte Carlo simulation convergence time=2.

ECM is not used for the first 10 time steps and 8 sensors are used:

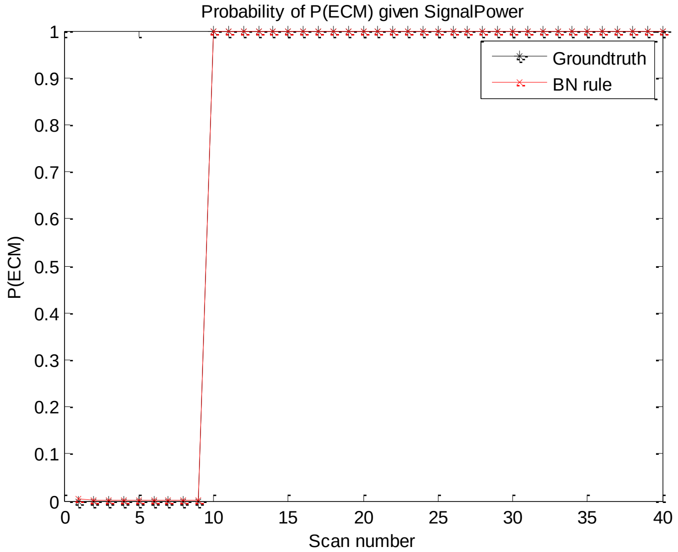

Figure 5-69: Multiple (8) good signal power sensors, limited case, t=10 (BNT)

Monte Carlo simulation convergence time=1.

ECM is not used for the first 40 time steps and 8 sensors are used:

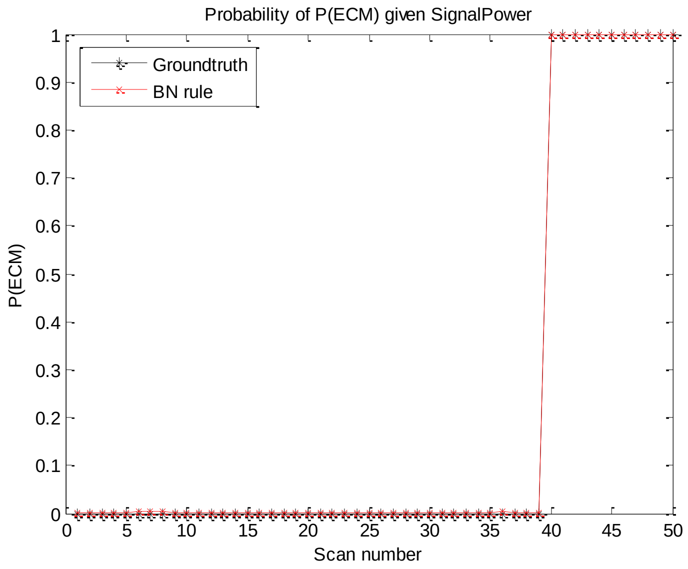

Figure 5-70: Multiple (8) good signal power sensors, limited case, t=40 (BNT)

Monte Carlo simulation convergence time=1.

It is important to note that using multiple poor sensors provide a performance which is equivalent to using a good sensor. Taking into account the correct prediction probability of the poor sensor, which is around 75%, multiplying the poor sensors can be more cost efficient.

5.2.2 DEMPSTER-SHAFER THEORY APPROACH

5.2.2.1 MEMORYLESS CASE

In the following sub-sections, single and multiple sensors are used respectively in both quality levels where the system does not keep prediction results of previous cases. Low signal period is set an average value (k_{low}=40) for this case. The results are plotted together with Accuracy, TPR and FPR values.

Single Poor Sensor Case: Threshold value of 0.7 is used.

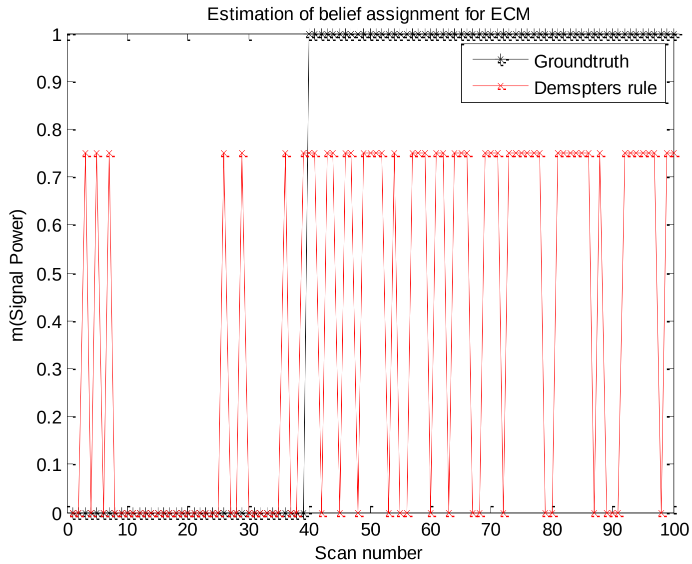

Figure 5-71: Single Poor Sensor in Memoryless ECM Case

Accuracy =0.73, TPR =0.7049, FPR =0.2308.

When the simulation is repeated 500 times, following values are achieved:, Accuracy=0.75, TPR=0.7489, FPR=0.2483.

Single Good Sensor Case: Threshold value of 0.7 is used:

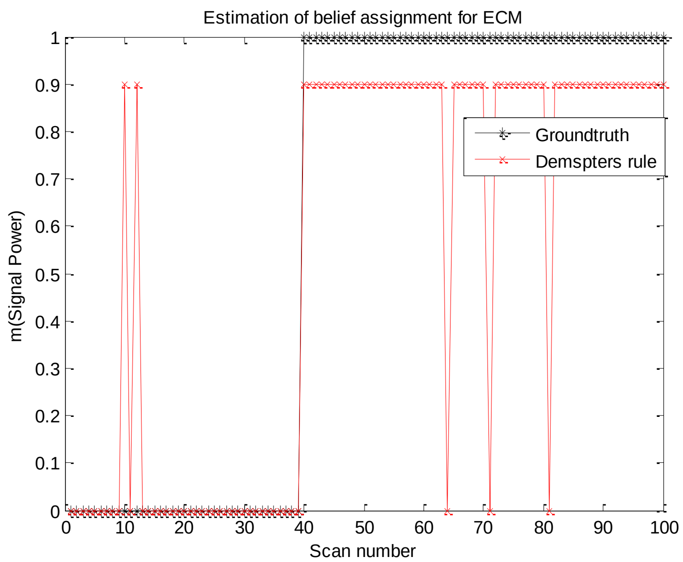

Figure 5-72: Single Good Sensor in Memoryless ECM Case (DST)

Accuracy =0.95, TPR =0.9508, FPR =0.0513.

When the simulation is repeated 500 times, following values are achieved:, Accuracy=0.9197, TPR=0.90, FPR=0.0498.

Multiple Sensors Case: 4 poor sensors are used.

This case will be examined in terms of accuracy, TPR, FPR and cost. ROC plot will be shown in order to observe effect of various threshold values. In order to obtain a ROC curve, following threshold values are chosen according to the values observed in the plots: Threshold [+inf, 0.95, 0.9, 0.4, 0.01, 0].

Results of first single run:

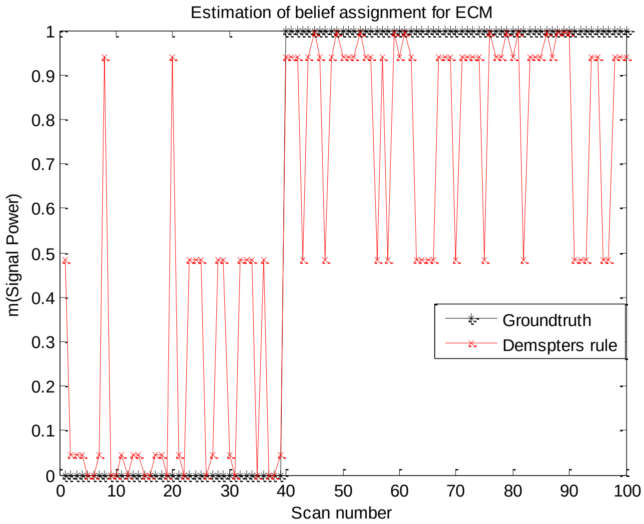

Figure 5-73: 4 Poor Sensors in Memoryless ECM Case-1 (DST)

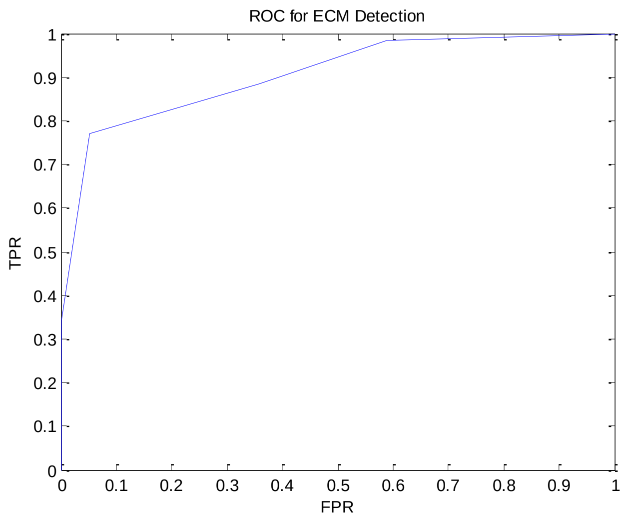

Figure 5-74: ROC Curve of 4 Poor Sensors in Memoryless ECM Case-1 (DST)

Overall results can be summed up in the table below:

Table 5-4: Performance Measures-1 (DST)

Threshold	1	0.95	0.9	0.4	0.01	0
TPR	0	0.344	0.770	0.885	0.984	1.00
FPR	0	0.000	0.051	0.359	0.590	1.00
Accuracy	0.39	0.60	0.84	0.79	0.76	0.61

Accuracy is highest when the threshold is 0.9.

Results of second single run:

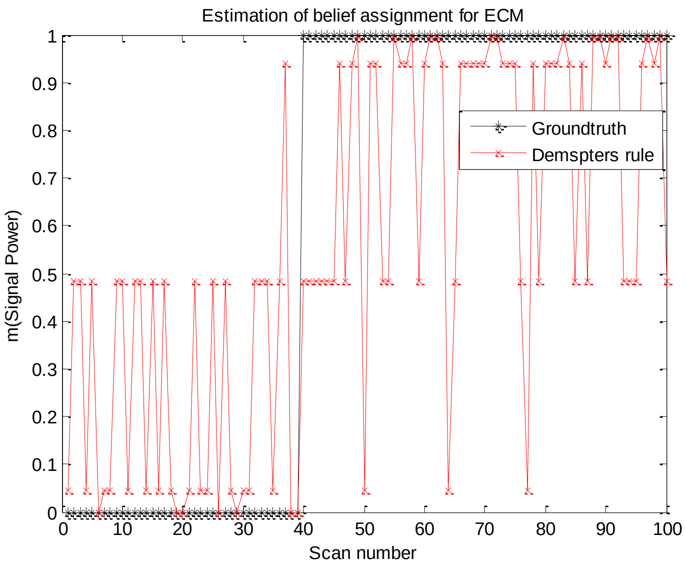

Figure 5-75: 4 Poor Sensors in Memoryless ECM Case-2 (DST)

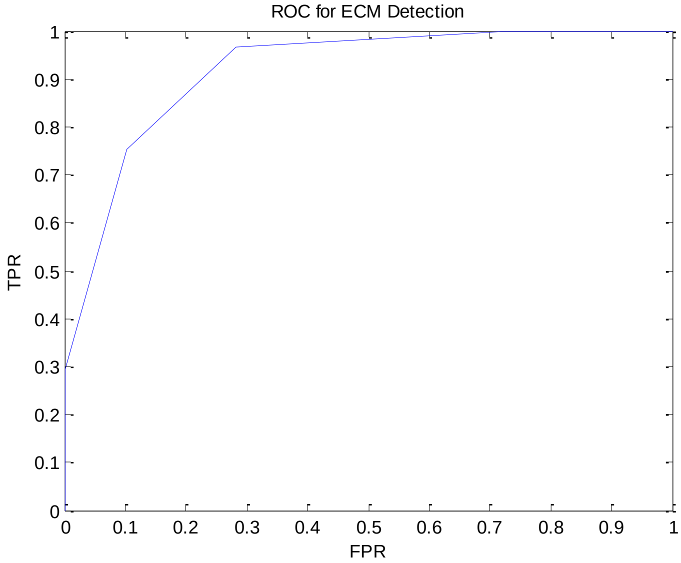

Figure 5-76: ROC Curve of 4 Poor Sensors in Memoryless ECM Case-2 (DST)

Overall results can be summed up in the table below:

Table 5-5: Performance Measures-2 (DST)

Threshold	1	0.95	0.9	0.4	0.01	0
TPR	0	0.295	0.754	0.967	1.00	1.00
FPR	0	0.000	0.103	0.282	0.718	1.00
Accuracy	0.39	0.57	0.81	0.87	0.72	0.61

Accuracy is highest when the threshold is 0.4.

Average Results when the simulation is run 100 times:

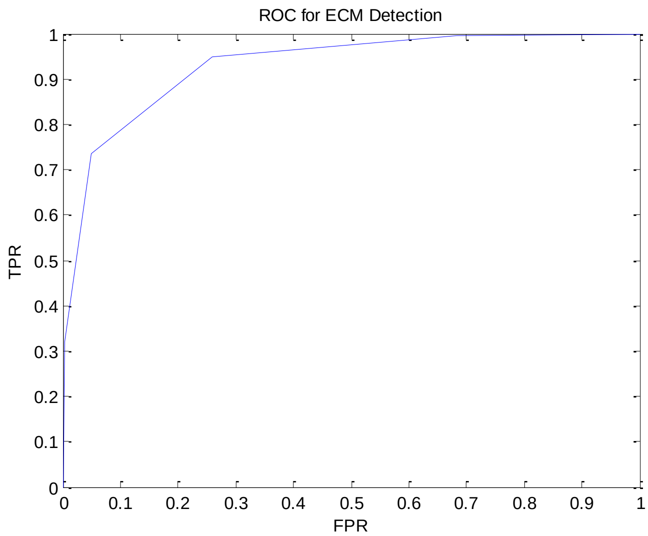

Figure 5-77: Average ROC Curve of 4 Poor Sensors in Memoryless ECM Case (DST)

Average results can be summed up in the table below:

Table 5-6: Performance Measures-Average (DST)

Threshold	1	0.95	0.9	0.4	0.01	0
TPR	0	0.315	0.736	0.945	0.996	1
FPR	0	0.003	0.048	0.269	0.693	1
Accuracy	0.3999	0.5877	0.8222	0.86	0.72	0.6
Cost	300050	205218	78023	18755	22685	33989

Accuracy is maximum and Cost is minimum when the threshold is 0.4. However, as explained in BNT method of the same study, setting the threshold 0.4 can lead to misclassifications when no target is present. As a matter of fact, this type of misclassifications can be observed from the FPR values in the corresponding table:

FPR(0.9)=0.048 is much lower than FPR(0.4)=0.269.

Having difficulty in deciding which threshold level is suitable to the classification system can be overcome by evaluating the obligatory requirements. Setting the threshold to 0.4 highly increases both TPR and FPR values which means that every one out of four predictions is incorrect when the target is not applying ECM. On the other hand, when 0.9

is selected as the threshold, one out of every four predictions is incorrect when the target is applying ECM.

The contradictory results show that there are cases in which no exact algorithm can be applied to find the best solution. In these situations, several methods can be applied in order to gain a wider perspective to the problem and find an optimized solution.

5.2.2.2 CASE WITH MEMORY

Performances of Single and Multiple Sensors will be presented for both ordinary and limited cases. Different low limit values will be conducted in single sensor simulations in order to observe best-performing lower limit.

5.2.2.2.1 SINGLE SENSOR

During the sub-sections, classifications with only one sensor will be simulated.

5.2.2.2.1.1 ORDINARY CASE

Poor Signal Power Sensor Case

This case will be examined for different lengths of low signal periods: t=10,20,40,60,80.

ECM is not used for the first 10 time steps

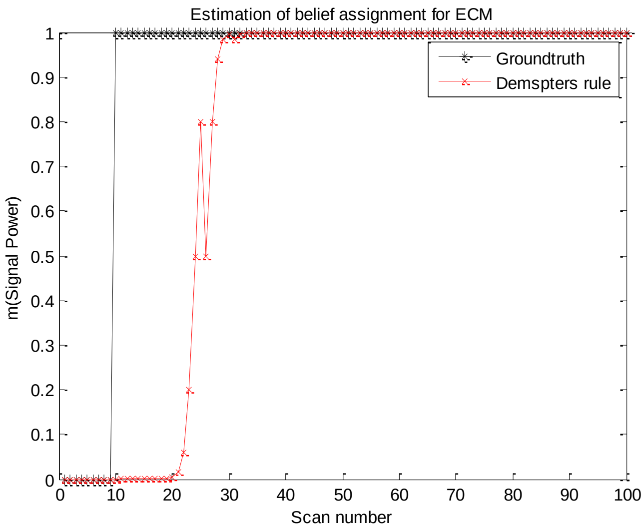

Figure 5-78: Single poor signal power sensor, ordinary case, t=10 (DST)

Convergence time is 44-10=34 for single run.

Convergence is achieved at 32.73 in average. Therefore, average t_{conv}=22.73.

ECM is not used for the first 20 time steps

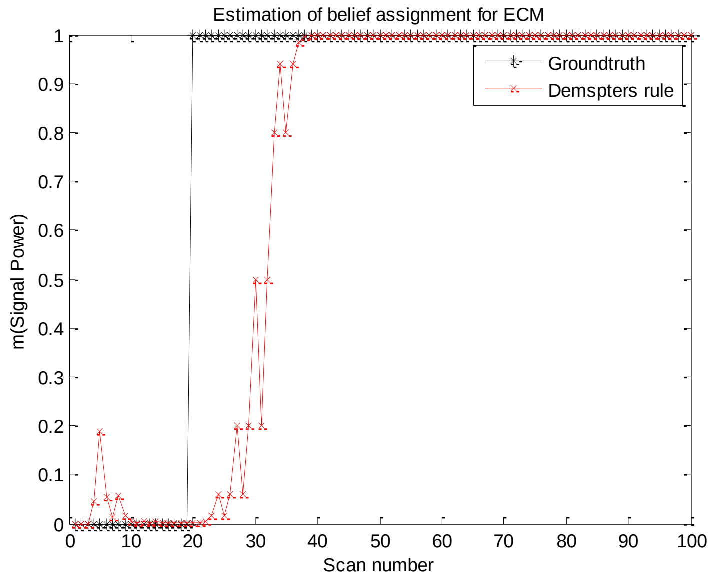

Figure 5-79: Single poor signal power sensor, ordinary case, t=20 (DST)

Convergence time is 38-20=18 for single run.

Convergence is achieved at 45.85 in average. Therefore, average t_{conv}=25.85.

ECM is not used for the first 40 time steps

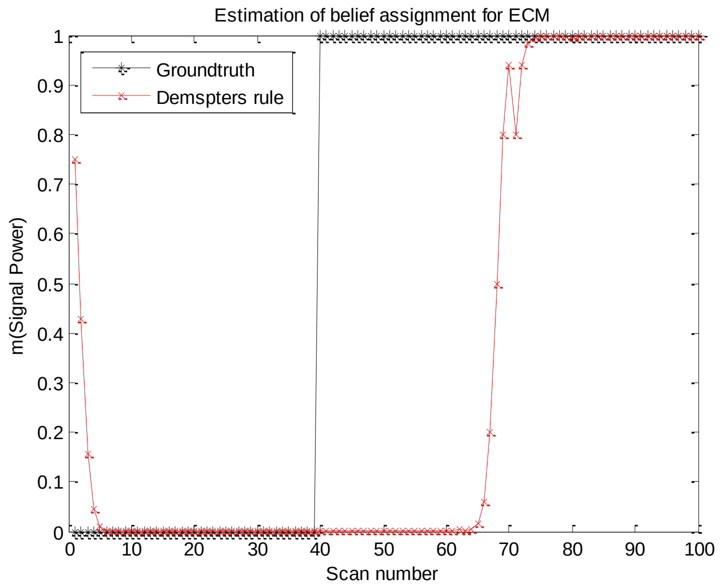

Figure 5-80: Single poor signal power sensor, ordinary case, t=40 (DST)

Convergence time is 74-40=34 for single run.

Convergence is achieved at 84.58 in average. Therefore, average t_{conv}=44.58.

ECM is not used for the first 60 time steps

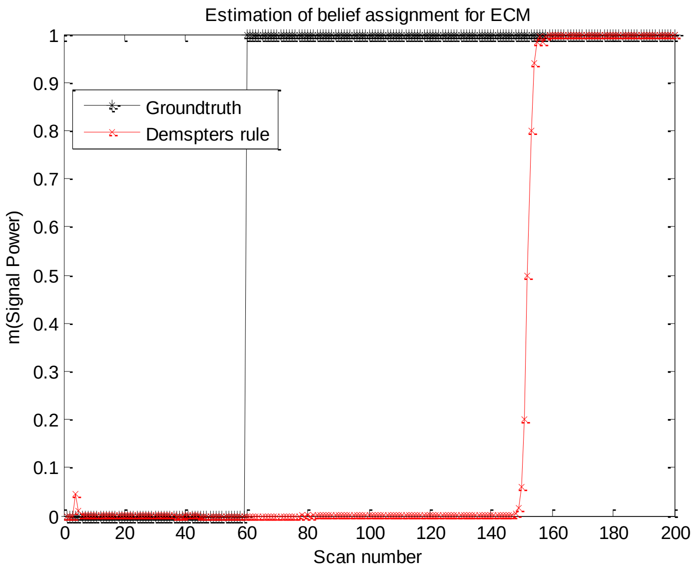

Figure 5-81: Single poor signal power sensor, ordinary case, t=60 (DST)

Convergence time is 156-60=96 for single run.

Convergence is achieved at 124.56 in average. Therefore, average t_{conv}=64.56.

ECM is not used for the first 80 time steps

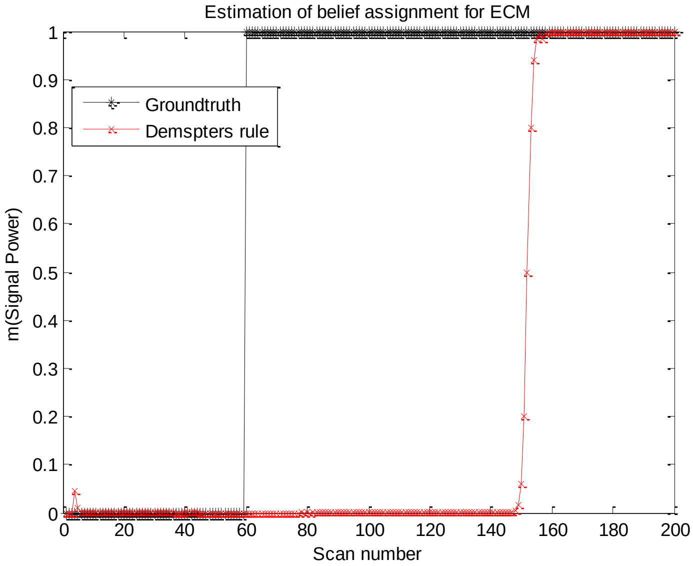

Figure 5-82: Single poor signal power sensor, ordinary case, t=80 (DST)

Convergence time is 176-80=96 for single run.

Convergence is achieved at 164.4 in average. Therefore, average t_{conv}=84.4.

Good Signal Power Sensor Case

This case will be examined for different lengths of low signal periods: t=10,20,40,60,80.

ECM is not used for the first 10 time steps

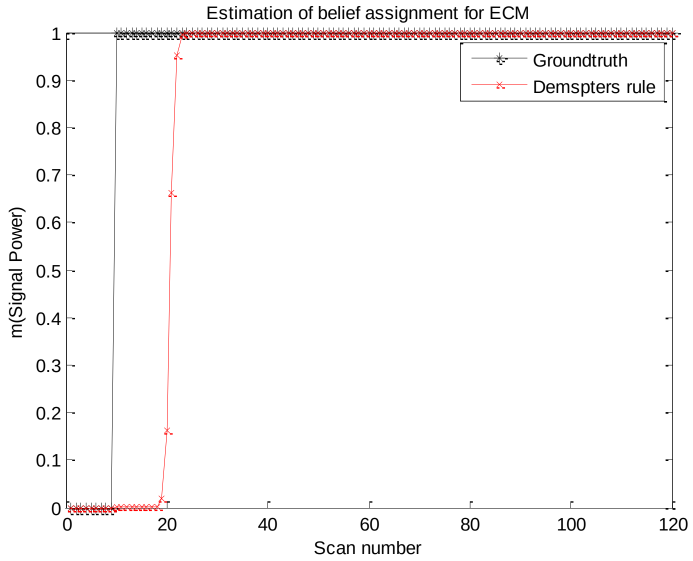

Figure 5-83: Single good signal power sensor, ordinary case, t=10 (DST)

Convergence time is 23-10=13 for single run.

Convergence is achieved at 26.22 in average. Therefore, average t_{conv}=16.22.

ECM is not used for the first 20 time steps

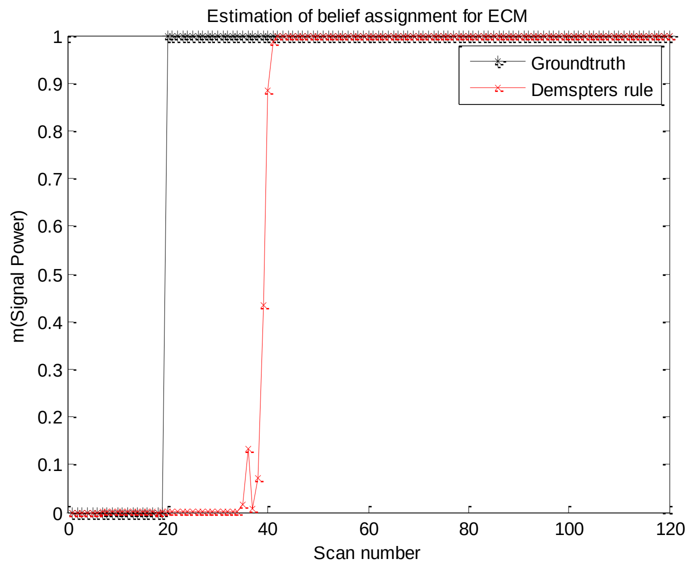

Figure 5-84: Single good signal power sensor, ordinary case, t=20 (DST)

Convergence time is 42-20=22 for single run.

Convergence is achieved at 51.57 in average. Therefore, average t_{conv}=31.57.

ECM is not used for the first 40 time steps

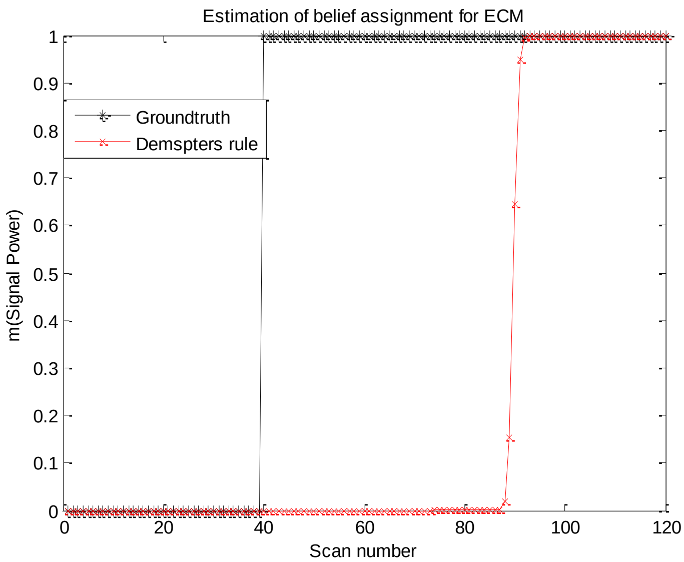

Figure 5-85: Single good signal power sensor, ordinary case, t=40 (DST)

Convergence time is 92-40=52 for single run.

Convergence is achieved at 102.26 in average. Therefore, average t_{conv}=62.26.

ECM is not used for the first 60 time steps

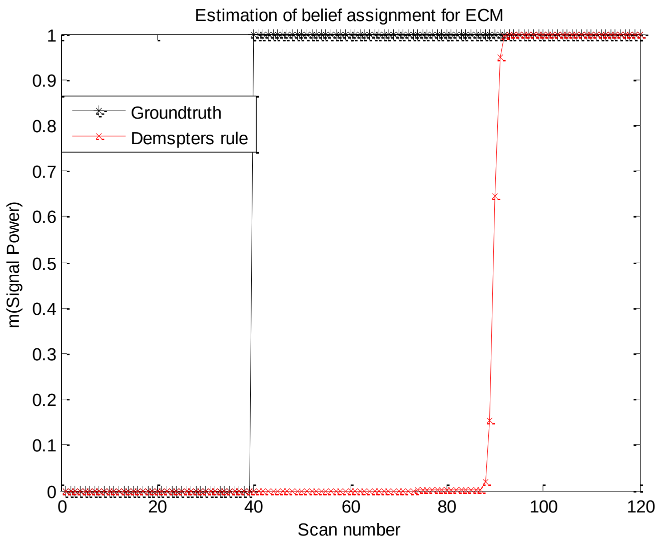

Figure 5-86: Single good signal power sensor, ordinary case, t=60 (DST)

Convergence time is 143-60=83 for single run.

Convergence is achieved at 152.84 in average. Therefore, average t_{conv}=92.84.

ECM is not used for the first 80 time steps

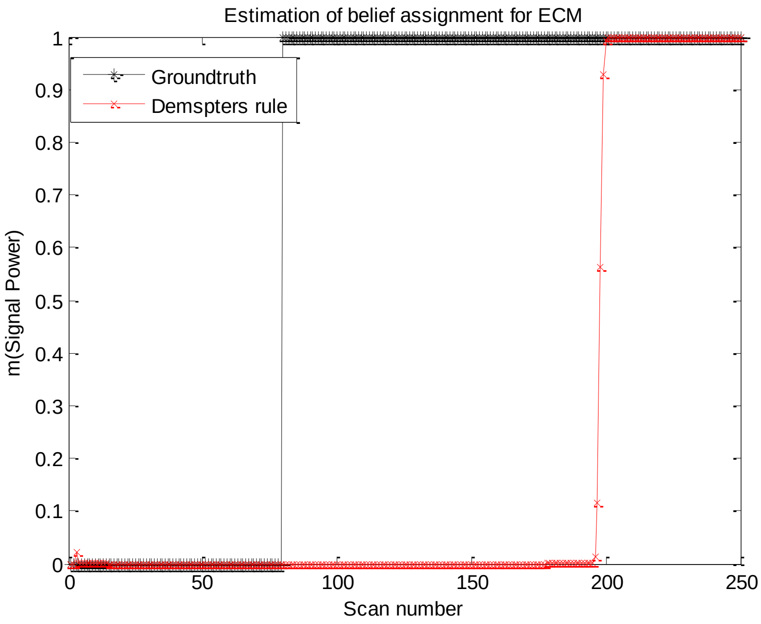

Figure 5-87: Single good signal power sensor, ordinary case, t=80 (DST)

Convergence time is 157-60=97 for single run.

Convergence is achieved at 203.55 in average. Therefore, average t_{conv}=123.55.

5.2.2.2.1.2 LIMITED CASE

Poor Signal Power Sensor Case

A. Lower Limit is set: 0.01.

Two different lengths of low signal periods are used: t_{low}=10,40.

i. ECM is not used for the first 10 time steps

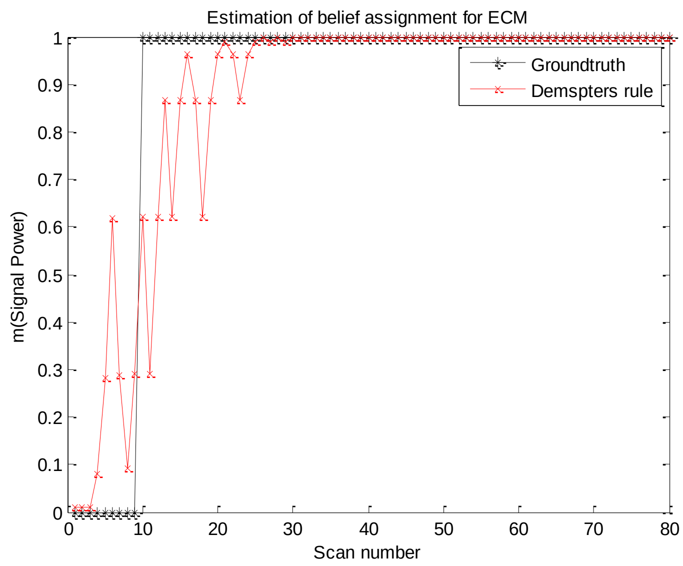

Figure 5-88: Single poor signal power sensor, 0.01 limited case, t=10 (DST)

Convergence time is 21-10=11 for single run.

Average convergence is achieved at 20.78. Then, t_{conv}=10.78.

ii.   ECM is not used for the first 40 time steps

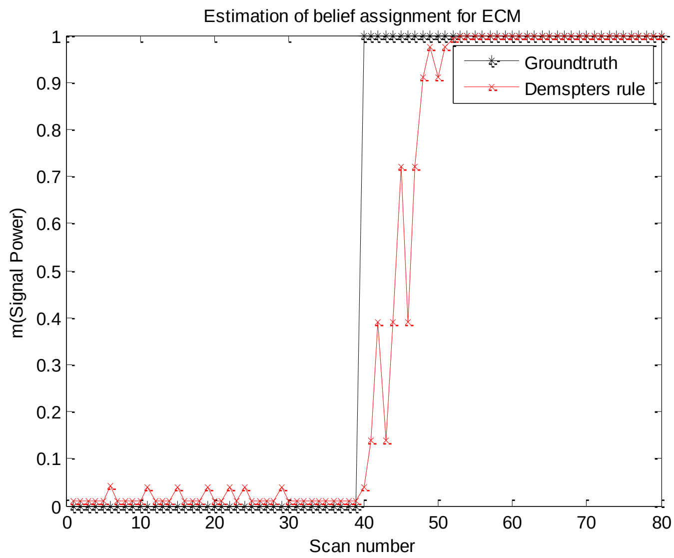

Figure 5-89: Single poor signal power sensor, 0.01 limited case, t=40 (DST)

Convergence time is 52-40=12 for single run.

Average convergence is achieved at 50.6. Then, t_{conv}=10.6.

B. Lower Limit is set: 0.001.

Two different lengths of low signal periods are used: t_{low}=10,40.

i. ECM is not used for the first 10 time steps

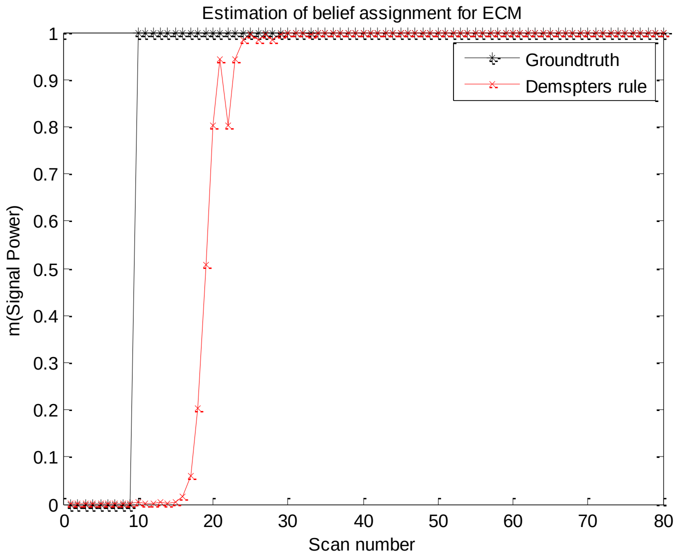

Figure 5-90: Single poor signal power sensor, 0.001 limited case, t=10 (DST)

Convergence time is 25-10=15 for single run.

Average convergence is achieved at 23.54. Then, t_{conv}=13.54.

ii. ECM is not used for the first 40 time steps

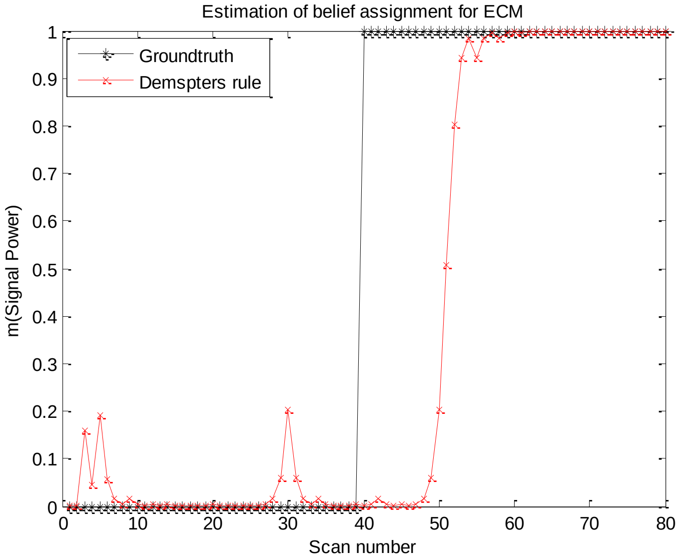

Figure 5-91: Single poor signal power sensor, 0.001 limited case, t=40 (DST)

Convergence time is 57-40=17 for single run.

Average convergence is achieved at 54.16. Then, t_{conv}=14.16.

Good Signal Power Sensor Case

A. Lower Limit is set: 0.01.

Two different lengths of low signal periods are used: t_{low}=10,40.

i. ECM is not used for the first 10 time steps

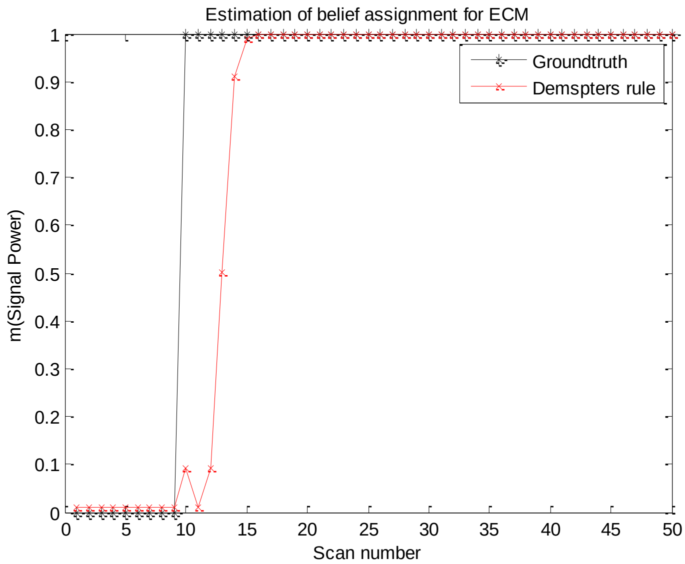

Figure 5-92: Single good signal power sensor, 0.01 limited case, t=10 (DST)

Convergence time is 15-10=5 for single run.

Average convergence is achieved at 14. Then, t_{conv}=4.

ii. ECM is not used for the first 40 time steps

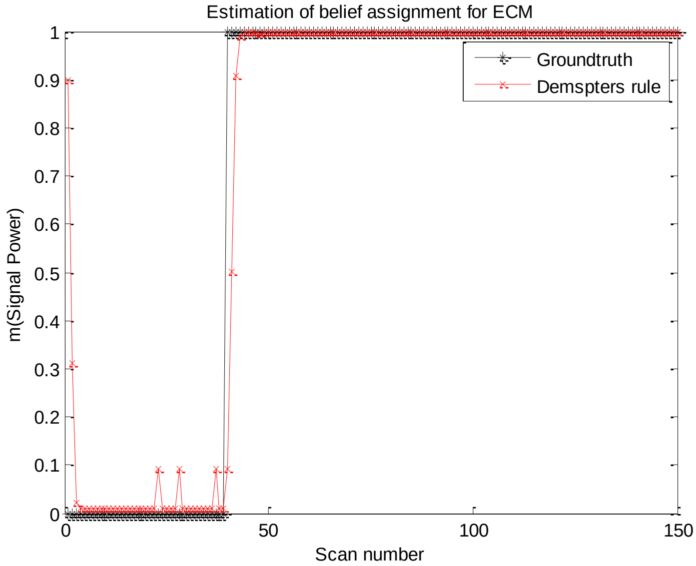

Figure 5-93: Single good signal power sensor, 0.01 limited case, t=40 (DST)

Convergence time is 43-40=3 for single run.

Average convergence is achieved at 44.03. Then, t_{conv}=4.03.

B. Lower Limit is set: 0.001.

Two different lengths of low signal periods are used: t_{low}=10,40.

i. ECM is not used for the first 10 time steps

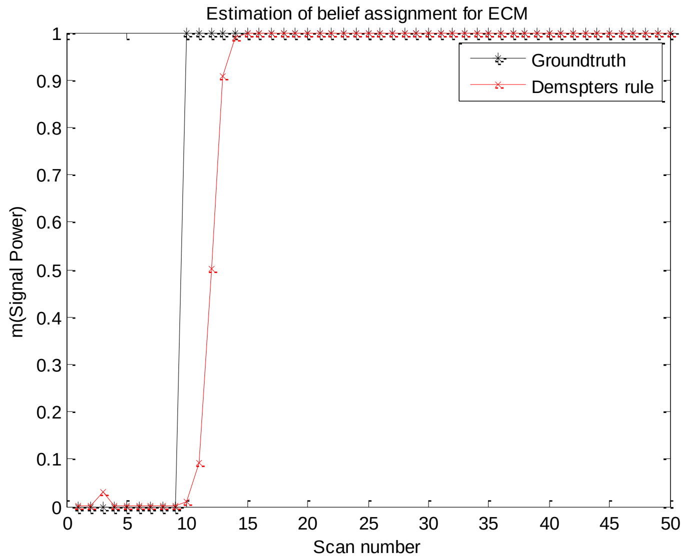

Figure 5-94: Single good signal power sensor, 0.001 limited case, t=10 (DST)

Convergence time is 14-10=4 for single run.

Average convergence is achieved at 15.37. Then, t_{conv}=5.37.

ii. ECM is not used for the first 40 time steps

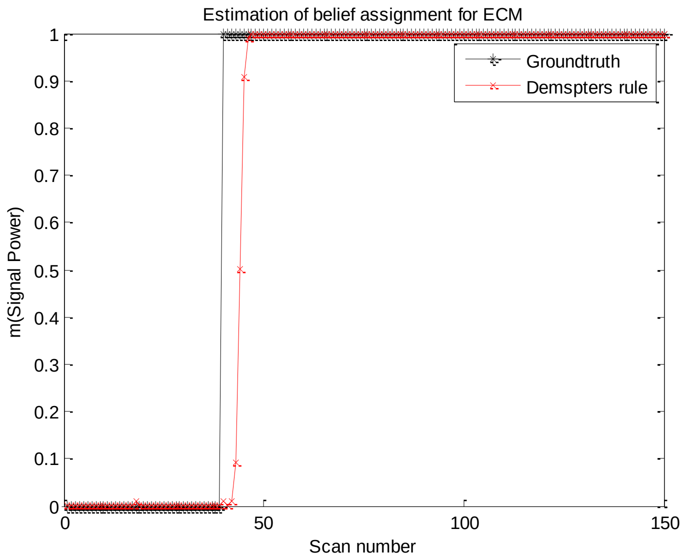

Figure 5-95: Good signal power sensor, 0.001 limited case, t=40 (DST)

Convergence time is 46-40=6 for single run.

Average convergence is achieved at 45.35. Then, t_{conv}=5.35.

5.2.2.2.1.3 FINITE MEMORY CASE

Approach-1:

In this case, decision is made by combining three independent predictions belonging to time steps k, k-1 and k-2.

Single Poor Sensor

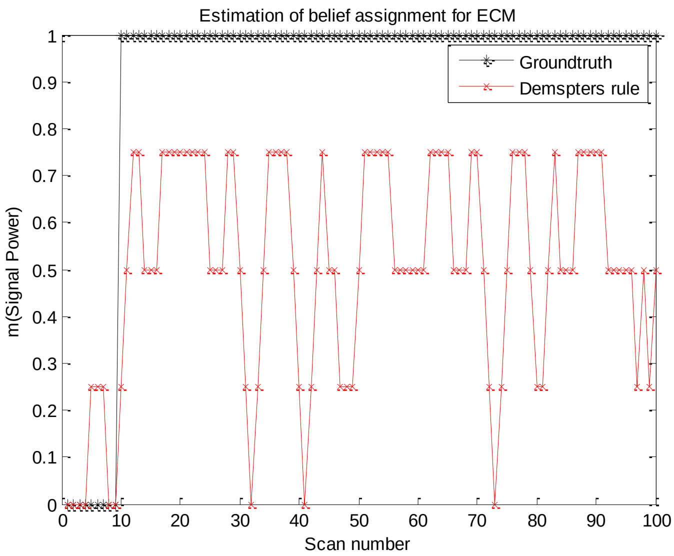

Figure 5-96: Single poor signal power sensor, finite memory case-1, t=10 (DST)

Single Good Sensor

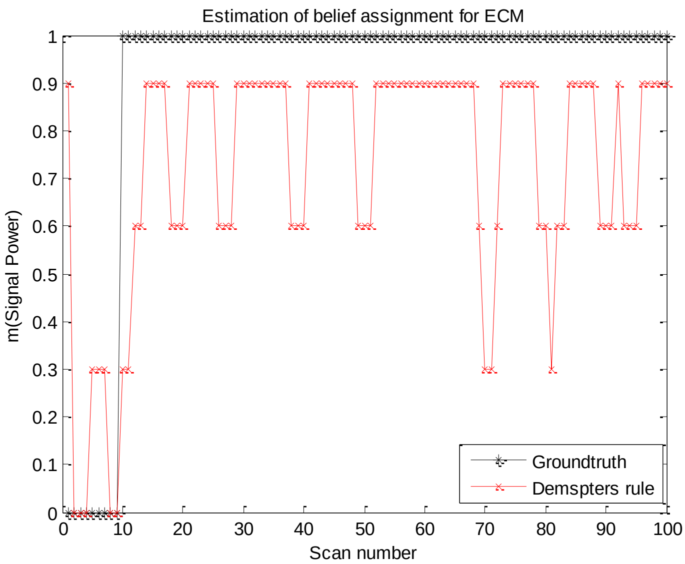

Figure 5-97: Single good signal power sensor, finite memory case-1, t=10 (DST)

Approach-2:

Belief is updated using previous n time steps that have weighted coefficients inversely proportional to their distance to the present time step.

Update algorithm: $m(k+1) = 0.6 \times m(k) + 0.25 \times m(k-1) + 0.15 \times m(k-2)$

Single Poor Sensor: Convergence duration is 50.

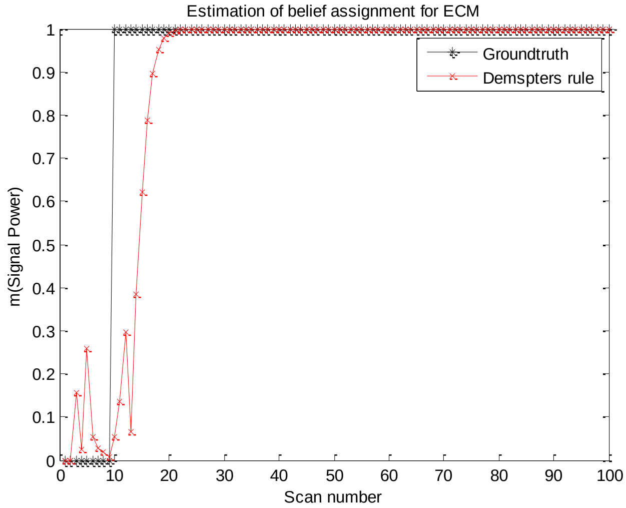

Figure 5-98: Single poor signal power sensor, finite memory case-2, single run, t=10 (DST)

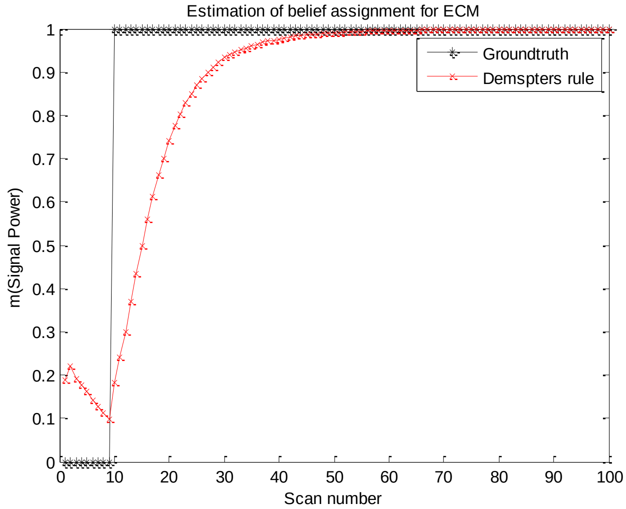

Figure 5-99: Single poor signal power sensor, finite memory case-2, Monte Carlo Average, t=10 (DST)

Single Good Sensor: Convergence duration is 30.

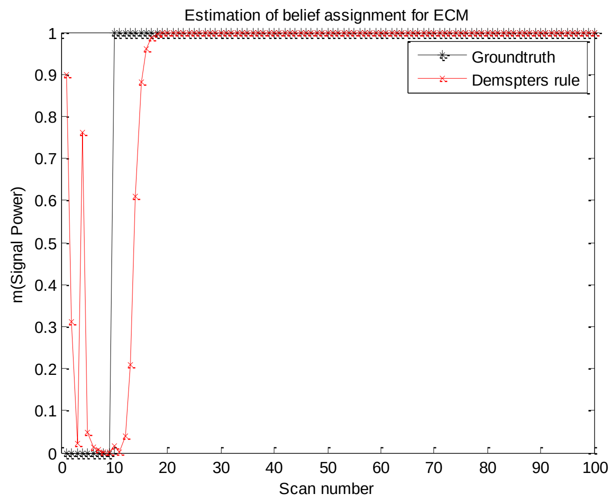

Figure 5-100: Single good signal power sensor, finite memory case-2, single run, t=10 (DST)

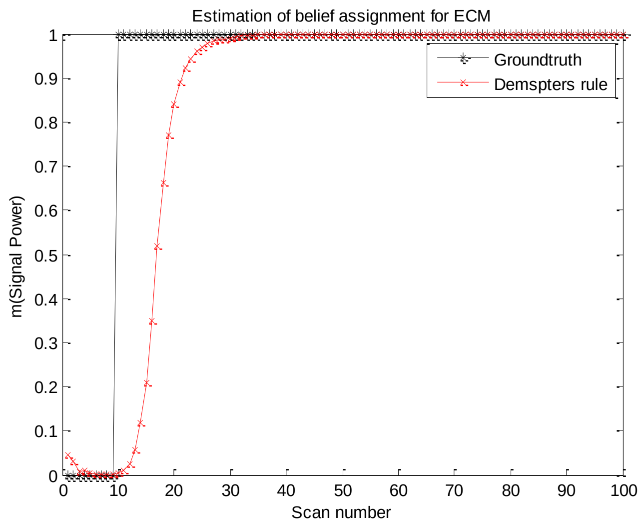

Figure 5-101: Single good signal power sensor, finite memory case-2, Monte Carlo Average, t=10 (DST)

Approach 2 makes an improvement in good sensor case whereas the performance in poor sensor case gets slightly worse. However, despite the performance increase in good sensor case, the results obtained are far away from the results obtained in “limited” case.

5.2.2.2.2 MULTIPLE SENSORS

During the sub-sections, classifications with different number of sensors will be simulated in order to observe the relationship between increasing sensor number and the performance of the classifier.

5.2.2.2.2.1 ORDINARY CASE

The following plots show averaged values for each time step that are obtained by Monte Carlo simulations. t_{low}=10,40 is selected in order to observe effect of Low signal periods on the classifier performance.

Poor Signal Power Sensor Case

ECM is not used for the first 10 time steps and 2 sensors are used:

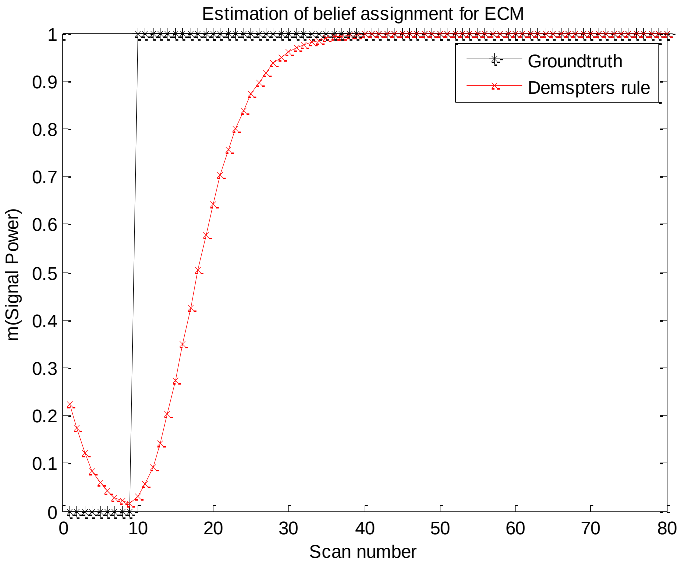

Figure 5-102: Multiple (2) poor signal power sensors, ordinary case, t=10 (DST)

Monte Carlo simulation convergence time=27.

When the simulation is performed 1000 times, average convergence time is: 13.0.

ECM is not used for the first 40 time steps and 2 sensors are used:

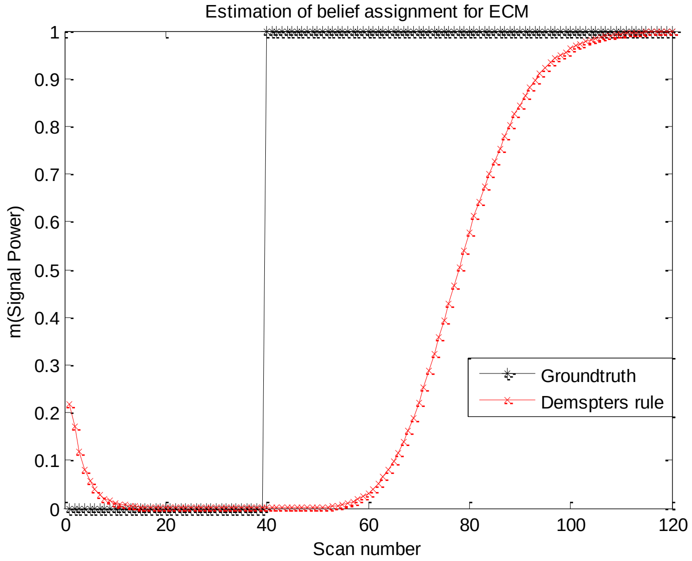

Figure 5-103: Multiple (2) poor signal power sensors, ordinary case, t=40 (DST)

Monte Carlo simulation convergence time=69.

When the simulation is performed 1000 times, average convergence time is: 42.16.

ECM is not used for the first 10 time steps and 4 sensors are used:

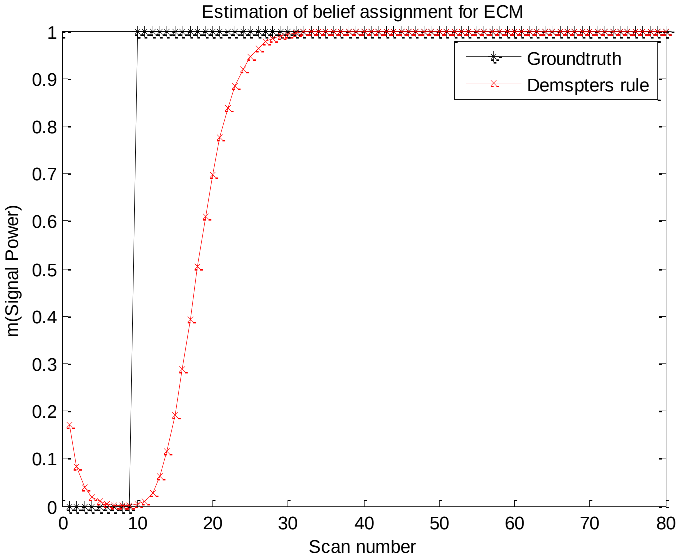

Figure 5-104: Multiple (4) poor signal power sensors, ordinary case, t=10 (DST)

Monte Carlo simulation convergence time=21.

When the simulation is performed 1000 times, average convergence time is: 11.15.

ECM is not used for the first 40 time steps and 4 sensors are used:

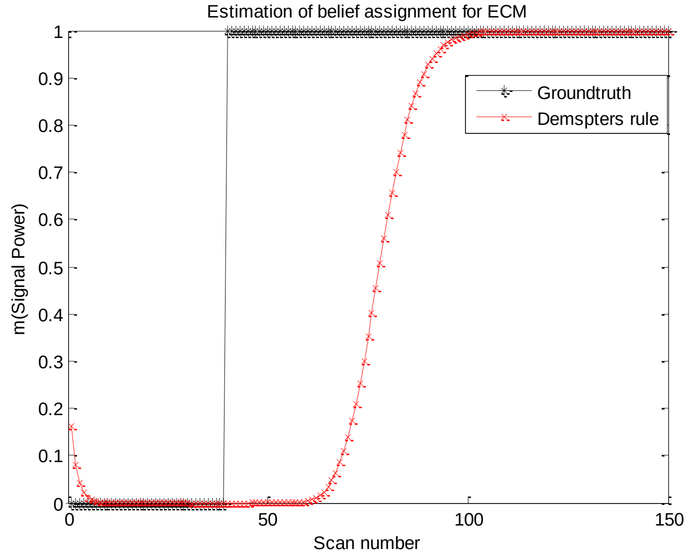

Figure 5-105: Multiple (4) poor signal power sensors, ordinary case, t=40 (DST)

Monte Carlo simulation convergence time=59.

When the simulation is performed 1000 times, average convergence time is: 40.59.

ECM is not used for the first 10 time steps and 8 sensors are used:

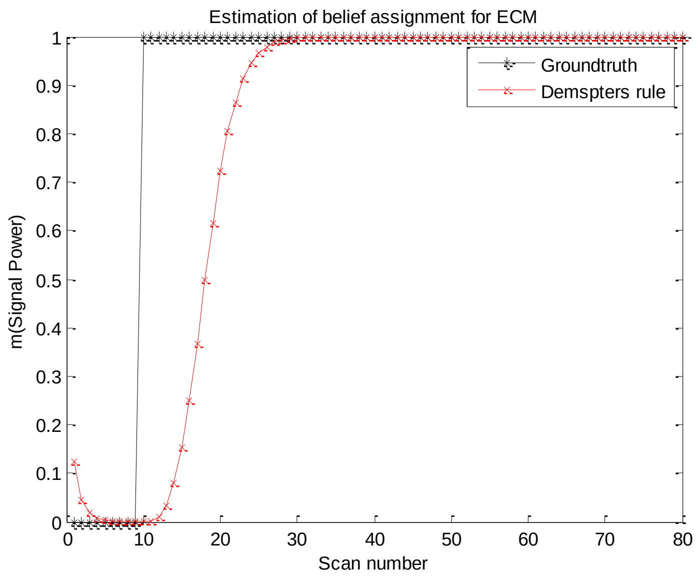

Figure 5-106: Multiple (8) poor signal power sensors, ordinary case, t=10 (DST)

Monte Carlo simulation convergence time=19.

When the simulation is performed 1000 times, average convergence time is: 10.0.

ECM is not used for the first 40 time steps and 8 sensors are used:

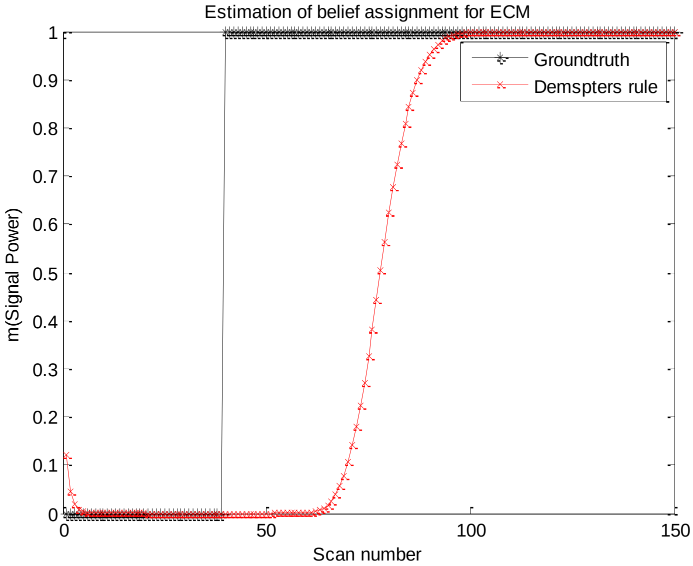

Figure 5-107: Multiple (8) poor signal power sensors, ordinary case, t=40 (DST)

Monte Carlo simulation convergence time=56.

When the simulation is performed 1000 times, average convergence time is: 38.7.

Good Signal Power Sensor Case

ECM is not used for the first 10 time steps and 2 sensors are used:

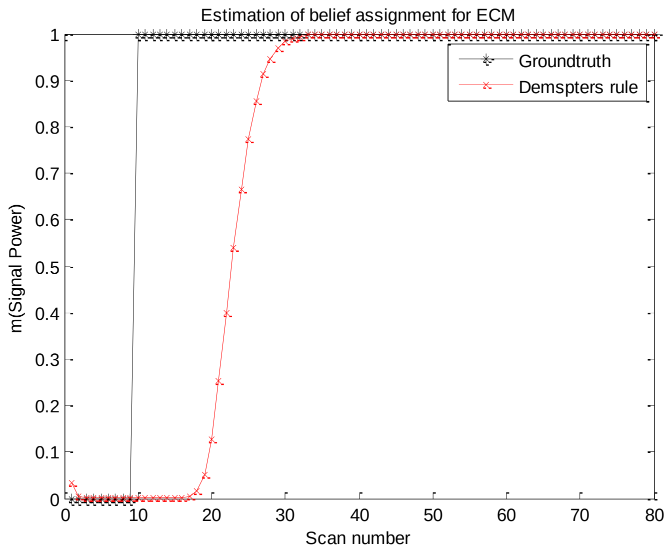

Figure 5-108: Multiple (2) good signal power sensors, ordinary case, t=10 (DST)

Monte Carlo simulation convergence time=23.

When the simulation is performed 1000 times, average convergence time is: 15.86.

ECM is not used for the first 40 time steps and 2 sensors are used:

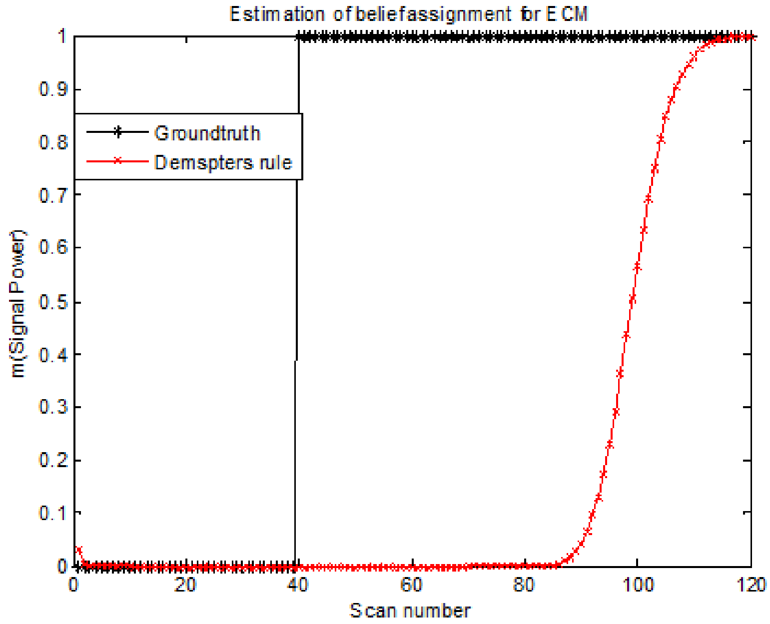

Figure 5-109: Multiple (2) good signal power sensors, ordinary case, t=40 (DST)

Monte Carlo simulation convergence time=75.

When the simulation is performed 1000 times, average convergence time is: 62.0.

ECM is not used for the first 10 time steps and 4 sensors are used:

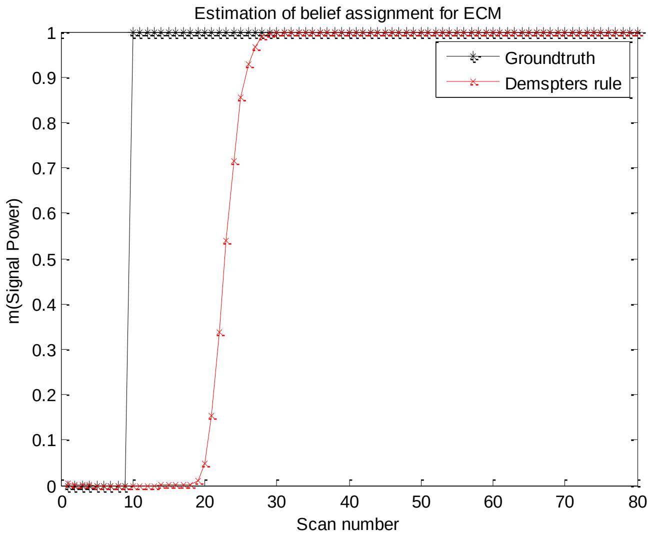

Figure 5-110: Multiple (4) good signal power sensors, ordinary case, t=10 (DST)

Monte Carlo simulation convergence time=20.

When the simulation is performed 1000 times, average convergence time is: 15.1.

ECM is not used for the first 40 time steps and 4 sensors are used:

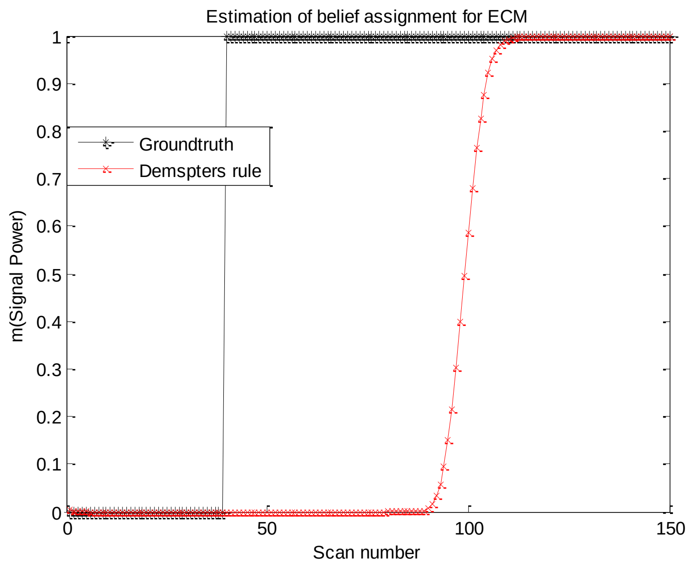

Figure 5-111: Multiple (4) good signal power sensors, ordinary case, t=40 (DST)

Monte Carlo simulation convergence time=71.

When the simulation is performed 1000 times, average convergence time is: 61.3.

ECM is not used for the first 10 time steps and 4 sensors are used:

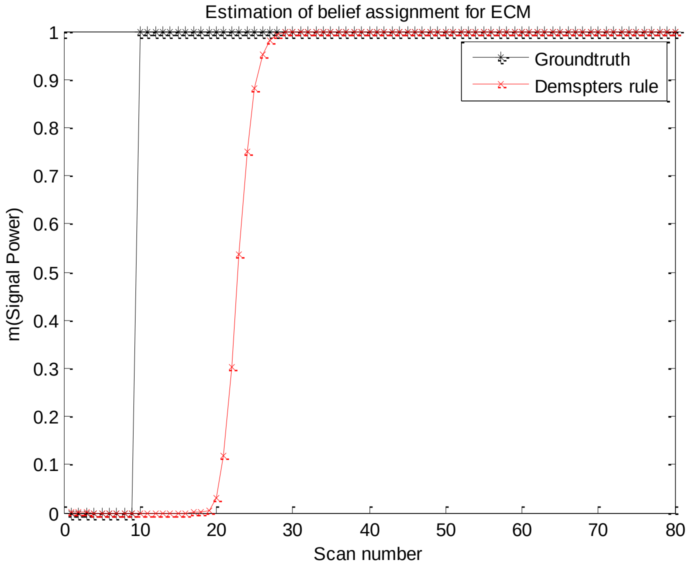

Figure 5-112: Multiple (8) good signal power sensors, ordinary case, t=10 (DST)

Monte Carlo simulation convergence time=19.

When the simulation is performed 1000 times, average convergence time is: 14.73.

ECM is not used for the first 40 time steps and 4 sensors are used:

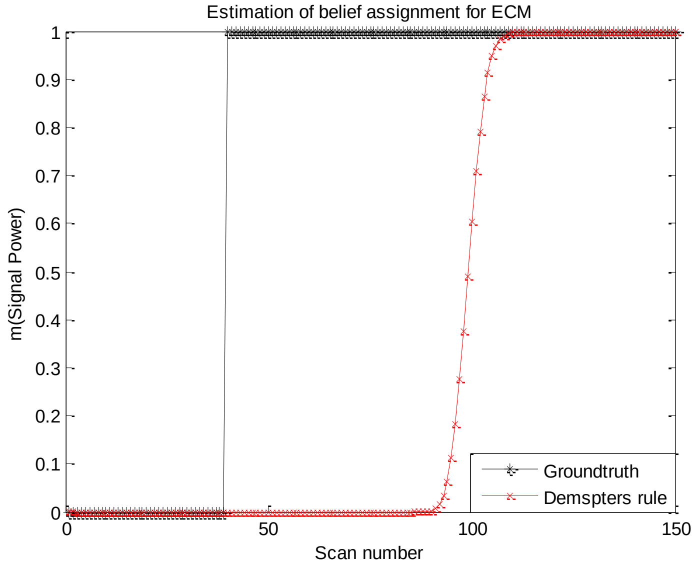

Figure 5-113: Multiple (8) good signal power sensors, ordinary case, t=40 (DST)

Monte Carlo simulation convergence time=69.

When the simulation is performed 1000 times, average convergence time is: 60.73.

5.2.2.2.2.2 LIMITED CASE

In limited case, lower limit is set to 0.001 in order to reduce convergence time of the classifier and make convergence time independent of the low signal period. Thus, after necessary simulations are observed, it will be seen that overall performance of the classifier is dramatically increased via implying lower limit together with multiple sensors.

Low signal periods are chosen as t_{low}=10,40 while studying performances of 4 and 8 sensors.

Poor Signal Power Sensor Case

ECM is not used for the first 10 time steps and 4 sensors are used:

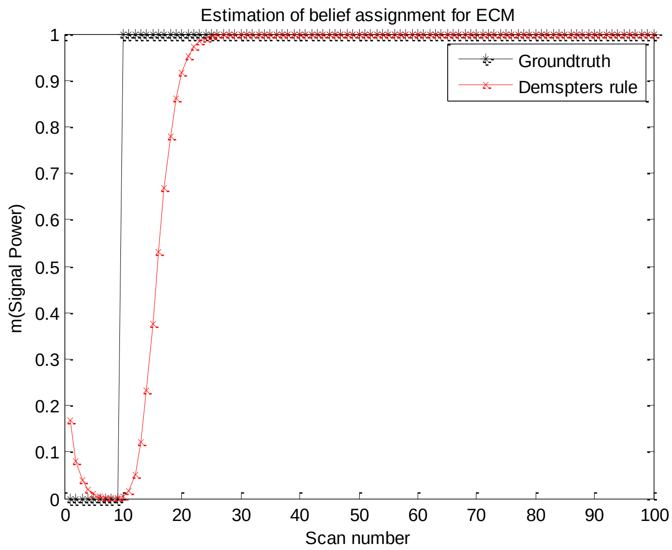

Figure 5-114: Multiple (4) poor signal power sensors, limited case, t=10 (DST)

Monte Carlo simulation convergence time=14.

ECM is not used for the first 40 time steps and 4 sensors are used:

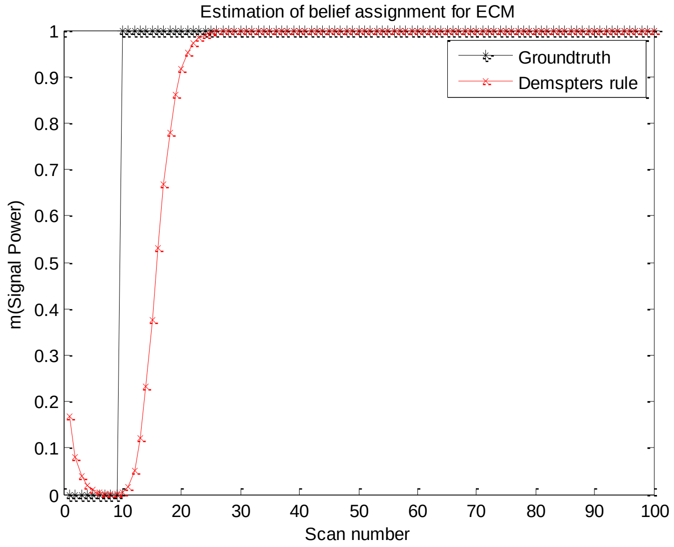

Figure 5-115: Multiple (4) poor signal power sensors, limited case, t=40 (DST)

Monte Carlo simulation convergence time=15.

Applying a lower limit to the belief function greatly reduces the effect of low signal period. As can be seen from the above plots and convergence times, performances of the cases became almost same, which is the desired result.

ECM is not used for the first 10 time steps and 8 sensors are used:

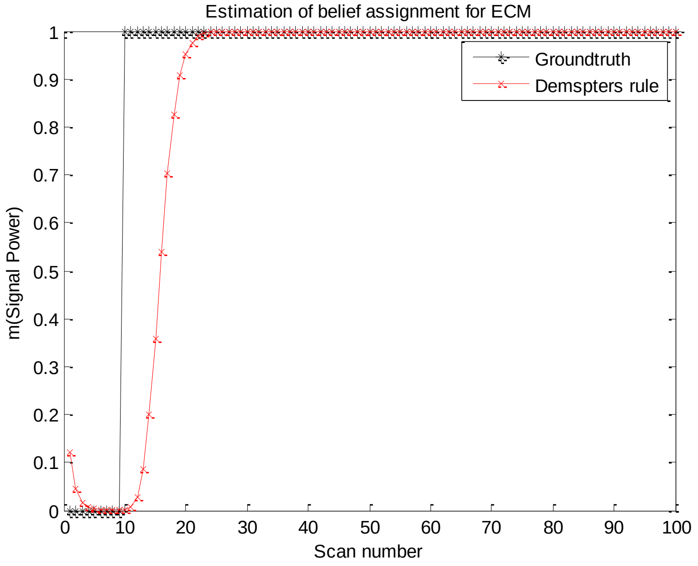

Figure 5-116: Multiple (8) poor signal power sensors, limited case, t=10 (DST)

Monte Carlo simulation convergence time=13.

ECM is not used for the first 40 time steps and 8 sensors are used:

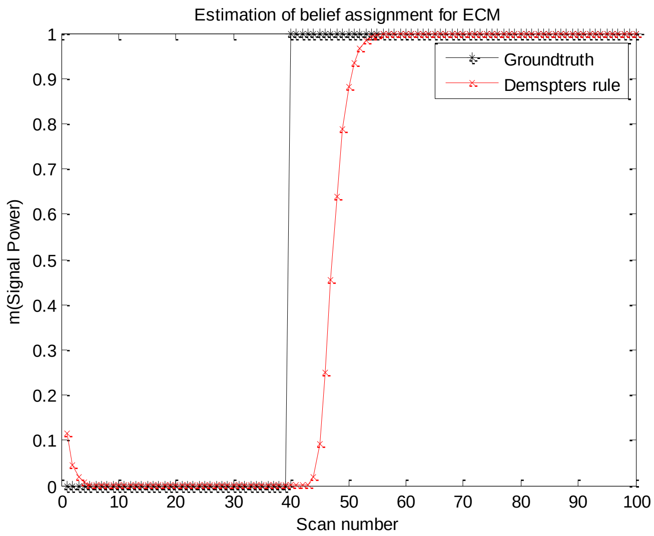

Figure 5-117: Multiple (8) poor signal power sensors, limited case, t=40 (DST)

Monte Carlo simulation convergence time=14.

When the sensor number is increased from one to four, convergence time of the system is halved. However, increasing sensor number to eight does not have a significant effect on the system. The most important point to note is that using multiple poor sensors result in a performance almost same with a system using single good sensor.

Good Signal Power Sensor Case

ECM is not used for the first 10 time steps and 4 sensors are used:

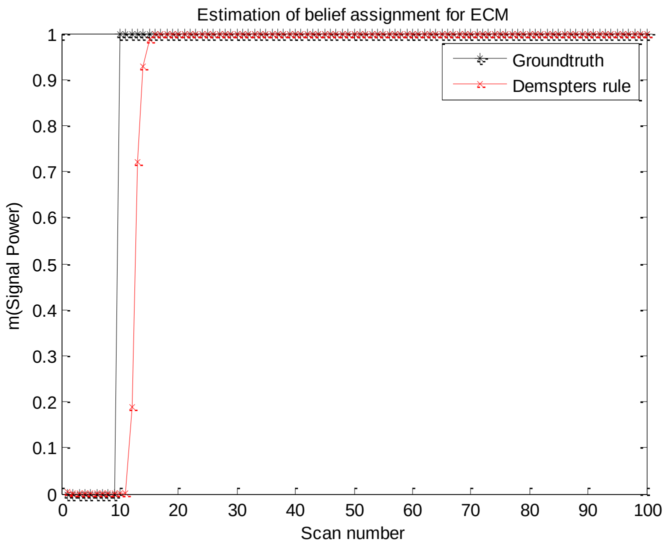

Figure 5-118: Multiple (4) good signal power sensors, limited case, t=10 (DST)

Monte Carlo simulation convergence time=6.

ECM is not used for the first 40 time steps and 4 sensors are used:

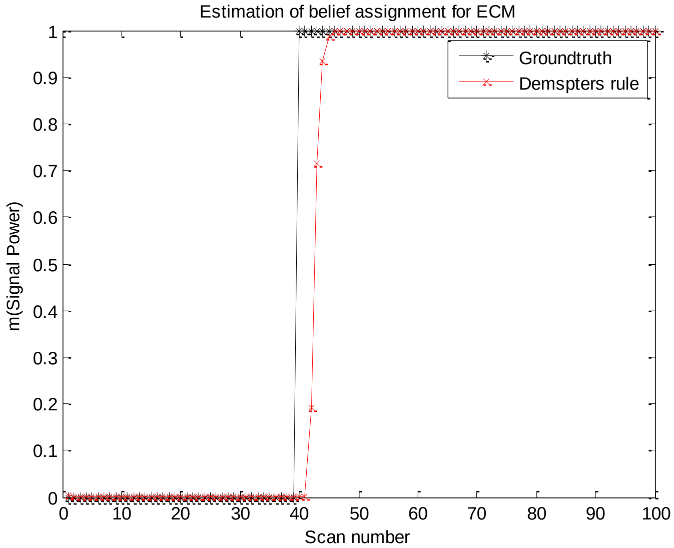

Figure 5-119: Multiple (4) good signal power sensors, limited case, t=40 (DST)

Monte Carlo simulation convergence time=6.

The performances of both cases are independent of the time step ECM is observed and the same.

ECM is not used for the first 10 time steps and 8 sensors are used:

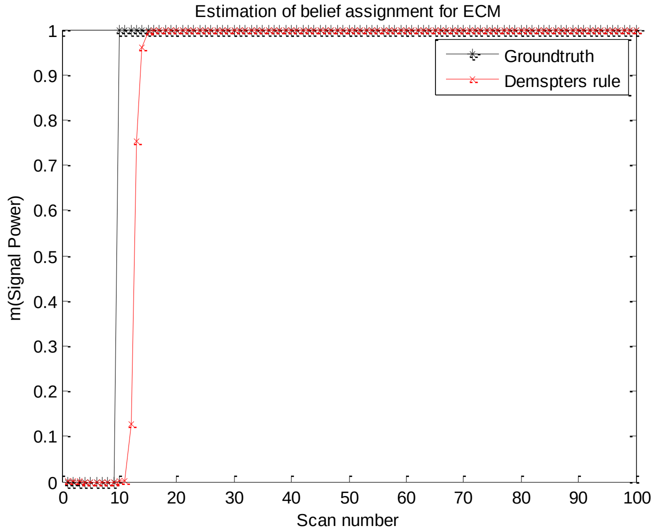

Figure 5-120: Multiple (8) good signal power sensors, limited case, t=10 (DST)

Monte Carlo simulation convergence time=5.

ECM is not used for the first 40 time steps and 8 sensors are used:

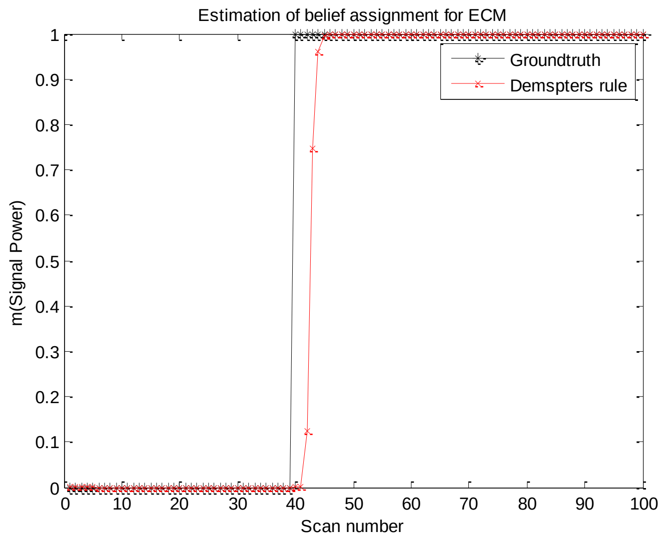

Figure 5-121: Multiple (8) good signal power sensors, limited case, t=40 (DST)

Monte Carlo simulation convergence time=5.

Performance of the systems with four good sensors is increased as the convergence times are halved with respect to a classifier using single good sensor. Increasing the sensor number more than four does not have a significant effect on the classifier performance.

CHAPTER 6

CONCLUSION

Throughout the thesis, two very commonly used statistical methods are studied in various aspects and their performances are measured by using different approaches.

In the first scenario, a fixed target is tracked by the radar system and the classifier used information coming from attribute sensor in order to predict target's type. The scenario is inspected in terms of the system's memory.

In the second scenario, a target is inspected by means of the electromagnetic waves reflected. When the returned electromagnetic waves are predicted to be amplified, ECM decision is given in order to take necessary precautions.

The results obtained from simulations and performance measurements applied in Chapter-5 are presented as a whole in this chapter and concluding remarks are provided. Moreover, BNT and DST methods are compared with each other in order to comprehend which method is superior to another in certain situations.

6.1 COMPARISON OF SCENARIO-1

6.1.1.1 MEMORYLESS CASE

BNT Method:

When the target of interest is fixed and the system does not have memory, performance of BNT classifier is as shown in the below table:

Table 6-1: Accuracy Table of Speed and Altitude Sensors (BNT)

		Speed Sensor	
		Poor	Good
Altitude Sensor	Poor	56.3%	71.0%
	Good	69.6%	87.0%

DST Method:

Accuracy values of DST classifier are as shown in the below table:

Table 6-2: Accuracy Table of Speed and Altitude Sensors (DST)

		Speed Sensor	
		Poor	Good
Altitude Sensor	Poor	56.3%	71.8%
	Good	68.7%	87.2%

Comparison:

As it is expected, when sensor quality gets better, performance of the classifier is increased. While using both poor quality sensors results in accuracy slightly better than random guessing, increasing only one sensor's quality level results in approximately 25% better accuracy and increasing both sensors' quality levels results in 54% better accuracy.

The results obtained in BNT and DST methods are almost the same as can be seen in the tables 6-1 and 6-2.

In this case, it is important to note that Accuracy and TPR values are equal.

6.1.1.2 CASE WITH MEMORY

BNT Method:

When the target of interest is fixed and the system updates the prior belief according to the latest prediction, convergence times are achieved within time steps shown in the below table:

Table 6-3: Convergence Time Table for Speed and Altitude Sensors (BNT)

		Speed Sensor	
		Poor	Good
Altitude Sensor	Poor	5.66	2.67
	Good	3.11	2.23

Updating the probability of TargetType based on previous knowledge makes the system more reliable than the system that decides based on only one time step.

Comparing the "good speed sensor-poor altitude sensor" and "poor speed sensor-good altitude sensor" cases, first one is slightly better because of the quality of the good speed sensor is better than quality of the good altitude sensor.

DST Method:

When DST method is applied and the system updates beliefs according to the preceding time step, convergence times obtained are shown in the below table:

Table 6-4: Convergence Time Table for Speed and Altitude Sensors (DST)

		Speed Sensor	
		Poor	Good
Altitude Sensor	Poor	4.15	2.2
	Good	2.3	1.15

Comparison:

Even though the results obtained in "Memoryless Case" were almost equal for BNT and DST approaches, convergence times differ from each other as DST holds upper hand against BNT method in this case.

When convergence times in tables 6-3 and 6-4 are inspected, it is seen that increasing only one sensor's quality level hastens the system around 45-50% and increasing both sensors' quality level hastens the system 60% for BNT, 72% for DST methods. Further examining the convergence time tables leads to the below differences between BNT and DST methods.

DST method results in:

- %26 better convergence time when poor speed and altitude sensors are used,
- %17 better convergence time when good speed and poor altitude sensors are used,
- 48% better convergence time when good speed and altitude sensors are used.

The reason for getting better convergence times in DST method is keeping certain amount of belief in unknown case while BNT method assigns probabilities only to two cases, namely “aircraft” and “helicopter”.

6.2 COMPARISON OF SCENARIO-2

6.2.1.1 MEMORYLESS CASE

Firstly, results of using only single sensor are provided for each method.

In order to inspect effect of using multiple poor sensors as using a good quality sensor might not always be cost effective, information coming from four different poor sensors is combined for both of the methods and overall averaged results are presented.

BNT Method:

Table 6-5: Performance Measurements in Memoryless Single Sensor Case (BNT)

		Accuracy	TPR	FPR
1xSensor	Poor	0.7532	0.7510	0.2433
	Good	0.9330	0.9539	0.0998

Table 6-6: Performance Measurements of Memoryless 4 Poor Sensors (BNT)

Threshold	1	0.8	0.4	0.05	0
TPR	0	0.324	0.736	0.949	1
FPR	0	0.006	0.049	0.258	1
Accuracy	0.3999	0.5921	0.8223	0.8665	0.6001
Cost	300050	203677	76753	18819	33989

DST Method:

Table 6-7: Performance Measurements in Memoryless Single Sensor Case (DST)

		Accuracy	TPR	FPR
1xSensor	Poor	0.7500	0.7489	0.2483
	Good	0.9197	0.9000	0.0498

Table 6-8: Performance Measurements of Memoryless 4 Poor Sensors (DST)

Threshold	1	0.95	0.9	0.4	0.01	0
TPR	0	0.315	0.736	0.945	0.996	1
FPR	0	0.003	0.048	0.269	0.693	1
Accuracy	0.3999	0.5877	0.8222	0.86	0.72	0.6
Cost	300050	205218	78023	18755	22685	33989

While working on multi poor sensor case, it is observed that maximum accuracy level is reached when either threshold is 0.4 or 0.9.

Comparison:

The results of poor single sensor cases are almost the same for BNT and DST approaches while good single sensor cases has several differences to note.

Accuracy and TPR are higher in BNT approach as well as FPR is almost double of the FPR value of DST approach. Even though achieving greater accuracy is desired, having greater FPR points to classification problems when the target is not applying ECM.

When tables 6-6 and 6-8 are inspected, it is observed that using multiple poor sensors highly increases performance of the classifier and TPR values of single good sensor case is obtained. However, since FPR values of multiple poor sensors are low, accuracies achieved are slightly lower than single good sensor case.

Within the light of the information provided by these tables, it can be said that designing the classifier for highest accuracy may not always provide the best solution at all situations. This is due to the fact that highest accuracy comes with five times higher FPR value.

In both cases, threshold levels can be set according to the system requirements, namely minimum and maximum TPR/FPR/Accuracy values. If the system is intolerant to false negatives, high TPR value is required while threshold level leading to lowest FPR value should be selected when the system is intolerant to false positives.

6.2.1.2 CASE WITH MEMORY

6.2.1.2.1 SINGLE SENSOR

6.2.1.2.1.1 ORDINARY CASE

BNT Method:

Firstly, in the below table, effect of prior probabilities are examined. In general, prior probability of 0.5 is assigned to both classes. However, as explained in previous chapters, prior probability P(ECM=yes) is assigned to 0.1 since coming up against a target that is using ECM is rare.

The table is prepared according to the average time steps required to reach 0.99 probability level after signal power level is increased.

Table 6-9: Average Convergence Time Table for Ordinary Case, Different Prior Probabilities (BNT)

	Poor Signal Power Sensor		Good Signal Power Sensor	
Low Signal Period	P(ECM=Yes)=0.1	P(ECM=Yes)=0.5	P(ECM=Yes)=0.1	P(ECM=Yes)=0.5
10	22.73	18.76	10.1	9
20	30.12	27.2	18.4	17.56
40	52.86	48.72	35.22	34
60	73	65.75	51.73	50.78
80	73.2	68.14	68.37	68.47

As expected, average convergence time is smaller when prior probability is assigned 0.5. However, starting the system with 0.5 probabilities lowers the system performance when the target is not applying ECM, which is the more common case. Furthermore, when prior probabilities are 0.5, the performance gets 1-10% better except poor signal power sensor case where low signal period is 10.

It is also important to note that convergence time steps get larger values as low signal period is increased and advantage of good signal power sensor is lost.

DST Method:

In DST method, no prior beliefs are assigned to the classes and the calculated average convergence times constitute below table:

Table 6-10: Average Convergence Time Table for Ordinary Case, Different Prior Probabilities (DST)

	Poor Signal Power Sensor	Good Signal Power Sensor
Low Signal Period	Average Convergence Time	Average Convergence Time
10	15.93	16.22
20	25.85	31.57
40	44.58	62.26
60	64.56	92.84
80	84.40	123.55

Differently from BNT approach, poor signal power sensor results in better convergence times in this case which is an unexpected result. This is due to the fact that DST approach's getting infinitesimally small values when the sensor is good. Therefore, as the low signal period is increased, the difference grows.

Comparison:

When the system of interest obtains information from a poor sensor, DST approach provides better results (about 15-30%). However, when the sensor quality is good, in DST approach leads to values very close to zero as low signal period is increased. This fact causes lengthy convergence time (40% longer than BNT approach) periods and puts BNT approach to better position.

In general, convergence time increases as the low signal period is increased. Due to this undesired result, an alternative solution will be presented in the next subsection where the effect of low signal period is avoided.

6.2.1.2.1.2 LIMITED CASE

As low signal period is increased, the system performance reduces dramatically due to the probabilities' getting smaller values close to zero. In order to solve this problem, lower probabilities and beliefs are assigned a low limit in order to prevent their approaching to zero.

For this purpose, simulations are performed for two different lower limits: 0.01 and 0.001. Two different convergence times will be present in this case: First type is obtained by summing Monte Carlo simulations and averaging at each time step. Second type is obtained by averaging each single run's convergence time as previous sub-section.

Monte Carlo simulations will result in higher convergence times since each point is averaged in that method.

BNT Method:

Monte Carlo Convergence Times:

Table 6-11: Monte Carlo Convergence Time Table for Limited Case (BNT)

	Poor Signal Power Sensor			Good Signal Power Sensor		
	Convergence Time Using Monte Carlo			Convergence Time Using Monte Carlo		
Low Signal Period	Limit=0.01	Limit=0.001	No limit	Limit=0.01	Limit=0.001	No limit
10	28	34	53	6	7	26
40	30	34	136	6	7	86

Average Convergence Times:

Table 6-12: Average Convergence Time Table for Limited Case (BNT)

	Poor Signal Power Sensor			Good Signal Power Sensor		
	Average Convergence Time			Average Convergence Time		
Low Signal Period	Limit=0.01	Limit=0.001	No limit	Limit=0.01	Limit=0.001	No limit
10	14.84	18.65	22.73	3.39	3.8	10.1
40	15	18.9	52.86	3.49	3.74	35.22

DST Method:

Monte Carlo Convergence Times:

Table 6-13: Monte Carlo Convergence Time Table for Limited Case (DST)

	Poor Signal Power Sensor			Good Signal Power Sensor		
	Convergence Time Using Monte Carlo			Convergence Time Using Monte Carlo		
Low Signal Period	Limit=0.01	Limit=0.001	No limit	Limit=0.01	Limit=0.001	No limit
10	26	30	37	9	11	26
40	25	30	86	9	11	82

Average Convergence Times:

Table 6-14: Average Convergence Time Table for Limited Case (DST)

	Poor Signal Power Sensor			Good Signal Power Sensor		
	Average Convergence Time			Average Convergence Time		
Low Signal Period	Limit=0.01	Limit=0.001	No limit	Limit=0.01	Limit=0.001	No limit
10	10.78	13.54	15.93	4	5.37	16.22
40	10.6	14.16	44.58	4.03	5.35	62.26

Comparison:

When the target is not applying ECM, limiting the lower values that BNT and DST approaches highly increases performance of the system by removing effect of low signal period. The studies are conducted for two different low signal periods, namely 10 and 40. When above tables are examined, it is seen that convergence times are almost the same even though the low signal period is quadrupled.

Besides that important result, results of two different lower limits can further be inspected. When limit=0.01, the systems seems to converge slightly faster than limit=0.001. However, when the plots of chapter-5 are examined, it is seen that limit=0.001 is more robust to sudden changes and provides a more stable system while limit=0.01 is prone to impulsive fluctuations. Therefore, 0.001 will be used in further studies throughout the thesis.

Comparing average convergence periods of BNT and DST approaches lead to similar conclusion to the previous sub-section. When lower limit is set to 0.001, DST approach has approximately 28% better performance in poor sensor case while BNT approach's performance is 30% better in good sensor case. These results are consistent with the "ordinary case" where no lower limit was set.

6.2.1.2.2 MULTIPLE SENSORS

In modern tracking systems, it is very common to combine information coming from multiple sensors. By multiplying two or more sensors, redundancy is increased and the system becomes able to compensate in case one of the sensor stops working. Moreover, combining low quality and inexpensive sensors may provide the same performance where expensive and high quality sensors are used.

"Ordinary Case" and "Limited Case" will be inspected respectively in the following sub-sections.

6.2.1.2.2.1 ORDINARY CASE

When multiple sensors are used in ordinary case, following results are obtained for BNT and DST case.

BNT Method:

Monte Carlo Convergence Times:

Table 6-15: Monte Carlo Conv. Time Table - Multiple Sensor Ordinary Case (BNT)

	Good		Poor	
	t=10	t=40	t=10	t=40
1xSensor	18	47	45	94
2xSensor	11	34	30	71
4xSensor	9	31	22	60
8xSensor	8	30	17	53
16xSensor	8	-	15	49

Average Convergence Times:

Table 6-16: Average Conv. Time Table - Multiple Sensor Ordinary Case (BNT)

	Good		Poor	
	t=10	t=40	t=10	t=40
1xSensor	10.10	35.22	22.73	52.86
2xSensor	7.97	27.34	17.12	47.05
4xSensor	7.09	26.84	13.17	43.32
8xSensor	6.77	26.70	11.40	41.07
16xSensor	6.60	-	10.58	40.41

Exception: No result could be calculated when 16 good sensors are combined as the probability reached to zero in MATLAB simulations.

DST Method:

Monte Carlo Convergence Times:

Table 6-17: Monte Carlo Conv. Time Table - Multiple Sensor Ordinary Case (DST)

	Good		Poor	
	t=10	t=40	t=10	t=40
1xSensor	28	83	38	85
2xSensor	23	75	27	69
4xSensor	20	71	21	59
8xSensor	19	69	19	56
16xSensor	18	68	18	54

Average Convergence Times:

Table 6-18: Average Conv. Time Table - Multiple Sensor Ordinary Case (DST)

	Good		Poor	
	t=10	t=40	t=10	t=40
1xSensor	16.22	62.26	15.93	44.58
2xSensor	15.86	62.00	13.00	42.16
4xSensor	15.10	61.30	11.15	40.59
8xSensor	14.73	60.73	10.00	38.70
16xSensor	14.55	60.80	9.42	36.68

Comparison:

In “ordinary case”, convergence time is directly proportional to the low signal period as explained previously due to probabilities’ and beliefs’ approaching very close to zero. In multiple sensor studies, effect of combining information from multiple same quality sensors is investigated and required number of poor sensors that achieves performance of a high quality and expensive sensor is obtained.

In BNT approach, it requires approximately eight low quality sensors to reach convergence time of a good sensor when low signal period is short. However, when low signal period gets longer, the required low quality sensor level increases to sixteen according to the tables 6-15 and 6-16.

In DST approach, due to the fact explained in previous sub-sections, poor sensors tend to lead shorter convergence times as can be seen in table 6-18. Moreover, it can be calculated that increasing sensor number does not provide significant performance improvement except when sensor is poor and low signal period is 10.

When sensor number is increased to eight, average convergence times are reduced by the ratios shown in below table:

Table 6-19: Average Conv. Time Improvement - Multiple Sensor Ordinary Case

	Good		Poor	
	t=10	t=40	t=10	t=40
BNT	32.97%	24.19%	49.85%	22.30%
DST	9.19%	2.46%	37.23%	13.19%

Even though initial convergence times were lower in BNT approach for good sensor case, a significant performance increase is achieved for good sensor when t=10 and t=40. DST approach results in better convergence times for poor sensor case for single sensors. However, as sensor number is increased, BNT method approaches to the performance of DST method.

6.2.1.2.2.2 LIMITED CASE

In order to overcome the difficulty of reducing convergence time when low signal period is high, the same method used in single sensor case will be used: Setting a lower limit to the probabilities and beliefs. Lower limit value of 0.001 is going to be used in this section as it is providing robustness to the system while avoiding lengthy low signal periods.

Monte Carlo simulation results are going to be provided in this section since systems are convergence at very short notices and average convergence times studied are very close to eachother.

The improvement achieved in this case may be compared to the results obtained in previous section where tables 6-15 and 6-17 are provided.

BNT Method:

Table 6-20: Monte Carlo Conv. Time Table - Multiple Sensor Limited Case (BNT)

		Good		Poor	
		t=10	t=40	t=10	t=40
BNT	1xSensor	7	7	34	34
	4xSensor	2	2	10	10
	8xSensor	1	1	6	6

DST Method:

Table 6-21: Monte Carlo Conv. Time Table - Multiple Sensor Limited Case (DST)

		Good		Poor	
		t=10	t=40	t=10	t=40
DST	1xSensor	11	11	30	30
	4xSensor	6	6	14	15
	8xSensor	5	5	13	14

Comparison:

When lower limit is used, it is seen that even by using single sensor, obtained results are much better than the results obtained in previous “ordinary case”. This result comes from the fact that probabilities and beliefs does not take values lower than 0.001 which enables them to recover quickly when received signal is increased.

In both approaches, it is observed that multiplying sensors dramatically increases the performances. The improvement factors which shows reduction in convergence times when sensor number is multiplied to eight can be seen in the below table:

Table 6-22: Monte Carlo Conv. Time Improvement - Multiple Sensor Limited Case

	Good		Poor	
	t=10	t=40	t=10	t=40
BNT	85.71%	85.71%	82.35%	82.35%
DST	54.55%	54.55%	56.67%	53.33%

Above table shows that improvement factor is around 85% in BNT approach, whereas it remains around 55% in DST approach. While using single poor sensor results in better convergence times in DST method as previous studies, using multiple sensors leads to better performance in BNT method due to the significant improvement factor achieved.

Furthermore, one of the most important points worth noting is achieving performance of a single high quality and expensive sensor by combining information from multiple low quality and inexpensive sensors.

6.3 CONCLUDING REMARKS

Two different scenarios are studied throughout the thesis by applying BNT and DST approaches to both of the scenarios under the same conditions. The purpose of the first scenario was to detect type of the tracked target and the second scenario was to detect possible ECM usage of the target being observed.

After obtaining results of the first simulations by two of the methods, the algorithms are improved in order to increase system performance and reliability.

When overall results are analyzed comparatively, it can be concluded that BNT and DST methods perform almost same in some of the cases. However, there are many cases where one of the methods gets ahead of the other as well.

The main differences of BNT and DST methods stem from the fact that BNT method is dealing with probabilities whereas DST method is working with belief functions where unknown knowledge can be assigned. Thus, BNT method always requires initial probabilities before starting simulations which has a direct affect on the results obtained.

Even though there is an impression that DST method is more complex and cumbersome to apply, simulations performed by BNT method took up much more time while simulating the scenarios in this thesis.

While BNT and DST methods can be alternatively used in single sensor cases depending on the sensor quality, BNT method has shown remarkable improvement when multiple sensors are used and lower limit is applied.

Even though there are many studies involving BNT and DST methods in the literature, there is no strict evidence or a final decision indicating one of the methods is superior to the other. There are scenarios or cases where one of these methods can be preferred, however it is not advisable to make generalizations. If one of the methods is needed to be chosen for the system, the problem should be analyzed comprehensively by applying both methods in all possible scenarios and a profit/loss analysis should be conducted.

REFERENCES

[1] Hautaniemi, S.K., P.T. Korpisaari, and J.P.P. Saarinen, Target identification with Bayesian networks. 2000: p. 55-66.

[2] Bayes, M. and M. Price, An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S. Philosophical Transactions (1683-1775), 1763. 53: p. 370-418.

[3] Buede, D.M. and P. Girardi, *A target identification comparison of Bayesian and Dempster-Shafer multisensor fusion.* Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 1997. 27(5): p. 569-577.

[4] Leung, H. and W. Jiangfeng, *Bayesian and Dempster-Shafer target identification for radar surveillance.* Aerospace and Electronic Systems, IEEE Transactions on, 2000. 36(2): p. 432-447.

[5] Cobb, B. and P. Shenoy, *A Comparison of Bayesian and Belief Function Reasoning.* Information Systems Frontiers, 2003. 5(4): p. 345-358.

[6] Challa, S. and D. Koks, *Bayesian and Dempster-Shafer fusion.* Sadhana, 2004. 29(2): p. 145-174.

[7] Wijayatunga, W.P., Statistical Analysis and Application of Naive Bayesian Network Classifier, 2007, Tokyo Institute of Technology.

[8] Jensen, F.V., *Bayesian networks and decision graphs*. Statistics for Engineering and Information Science 2001, New York: Springer. xv, 268 p.

[9] Shafer, G., *A Mathematical Theory of Evidence*1976: Princeton University Press.