Tracking of Moving Objects in a Wireless Sensor Network with use of Kalman Filtering and Machine Learning

Mehdi Darbandi

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical and Electronic Engineering

Eastern Mediterranean University
May 2019
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Acting Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Electrical and Electronic Engineering.

Prof. Dr. Hasan Demirel
Chair, Department of Electrical and Electronic Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Electrical and Electronic Engineering.

Prof. Dr. Şener Uysal
Supervisor

Examining Committee

1. Prof. Dr. Osman Kukrer
2. Prof. Dr. Sener Uysal
3. Asst. Prof. Dr. Kamil Yurtkan

ABSTRACT

WSN encompasses a significant number of sensor nodes that were generally fortified by sensing, processing as well as communication competences. Such equipment's, gather conservational data using their sensors, besides, send such information to the base station. Considering that in these types of networks there is restricted electricity capacity and it is usually impossible to revitalize or substitute dead nodes; therefore, energy management is of considerable significance at such types of systems. Different methods are suggested and proved for tracking the targets in such a network using various techniques as well as evolutionary algorithms. Many types of research have tried to improve the performance of each of these methods. Among the presented methods, methods that can estimate the subsequent position of the goal at the next period and in terms of specific techniques are more efficient. By predicting a target move, only a portion of the network nodes should be involved for tracking in the next period, resulting in a significant reduction in energy consumption. In this thesis, a new method is presented for target tracking with the use of Kalman filtering and machine learning. The proposed method results in a network with better performance and less energy consumption.

Keywords: Kalman filtering, WSN, Estimation and prediction, Target Tracking.

ÖZ

Kablosuz sensör ağları, genellikle algılama, işleme ve iletişim yetenekleriyle donatılmış çok sayıda sensör düğümünden oluşur. Sensör düğümleri, sensörlerini kullanarak çevresel bilgileri toplar ve baz istasyonuna gönderir. Bu tip ağlarda sınırlı enerji depolanması olduğunu ve ölü düğümlerin yeniden şarj edilmesi ya da değiştirilmesinin genellikle imkansız olduğunu göz önünde bulundurarak [1]; bu nedenle, enerji yönetimi bu tür şebekelerde önemli noktalardan biridir. Evrimsel algoritmaların yanı sıra, çeşitli teknikler kullanarak bu tür bir ağdaki hedefleri izlemek için farklı yöntemler önerilmiş ve kanıtlanmıştır [1, 2]. Birçok araştırma bu yöntemlerin her birinin performansını iyileştirmeye çalışmıştır. Sunulan yöntemler arasında, bir sonraki zaman diliminde ve belirli teknikler açısından hedefin bir sonraki yerini tahmin eden yöntemler daha verimlidir. Bir hedef hareketi öngörerek, ağ düğümlerinin sadece bir kısmının gelecek dönemde izlemeye dahil olması gerekir; bu da enerji tüketiminde önemli bir düşüşe neden olur. Bu tezde Kalman filtreleme ve makine öğrenmesi ile hedef izleme için yeni bir yöntem sunmaya çalışıyorum. Önerilen yöntemin, şebekenin daha optimum çalışmasına ve daha az enerji tüketimine yol açacağını göstereceğim.

Anahtar Sözcükler: Kalman filtreleme, WSN, Tahmin ve tahmin, Hedef izleme.

ACKNOWLEDGMENT

Special appreciations to my generous and kind supervisor, Prof. Dr. Şener UYSAL, for his inspiration as well as provision throughout my master degree's period. Now, in the final stages of compilation of the documentation, I would like to express my gratitude to the efforts of all those who helped me in any way during my work.

Special thanks also to all my graduate friends, especially my dear friend Dr. Pouya Bolourchi and Mr. Masoud Moradi for sharing the literature and invaluable assistance. That is my pleasure to catch the generous friends, and I am truly honored for having such contacts.

I would like to express my sincere thankfulness to my lovely family; they give me a chance for completing my higher education in Cyprus. Without their support both in financial and emotional matter achievement of this level was impossible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

BS	Base Station
CH	Cluster Head
DD	Directed Diffusion
EKF	Extended Kalman Filter
ELM	Extreme Machine Learning
GPS	Global Positioning System
MAC	Message Authentication Code
ML	Machine Learning
LEACH	Low Energy Adaptive Clustering Protocol
P2P	Peer-to-peer
RMS	Root Mean Square
SAR	Sequential Assignment Routing
SLFN	Single Hidden-Layer Feed Forward Neural Network
SPIN	Sensor Protocol for Information via Navigation
UKF	Unscented Kalman Filter
WSN	Wireless Sensor Network

Chapter 1

INTRODUCTION

1.1 General Introduction

With the growing expansion of small smart devices in addition to the convergence of wireless communications with little processing, there is the possibility of emerging very low-cost networks in a variety of sizes. These networks can collect environmental information, have no specific infrastructure, and are deployed. They can organize themselves, can have a very long lifetime, and do not need energy from external sources. The most important of such networks are sensory nets. Sensor networks are considered as one of a variety of Ad-Hoc (type of temporary computer-to-computer connection) networks. The significant and distinguishable characteristic of the sensor networks is equipping of nodes with a kind of electronic sensor. Such nodes gather the information and submit those data toward the user or supervisor [2].

Sensor networks introduce a kind of communication that uses very low-cost nodes. There are wide varieties of sensors, which can measure various quantities. These include accelerometers, temperature sensors, smoke detectors, humidity sensors, sound sensors, and image sensors. Another essential feature of sensor networks is that they are highly dependent on the application. This means that different forms have different properties on the network. For example, in applications such as surveying agricultural lands, we need to scatter hundreds of high radio-range nodes on each hectare of agricultural land, but in applications such as health monitoring in hospitals, nodes with a lower radio range of ten meters are placed in an environment, that is the amount of sensors is similar to the number of patients. The observer will make decisions, based on the received information.

In many cases, knowing the physical information of the sensor node can be very helpful in deciding what information it receives. As an instance, if we are collecting environmental information regarding heat at an industrial environment, we will not benefit from receiving a temperature-specific message unless we know the location of the sending node of this message [3]. In addition, given that sensor networks need to be very smart in consuming energy, all the algorithms proposed in this regard should be designed in such a way that they have a low energy consumption. In these cases, having spatial information of nodes can help such algorithms. On the other hand, based on such fact that the nodes were arbitrarily distributed from the outside (usually by the aircraft for ground applications and boats for marine applications), the supervisors are not precisely aware of the position of the nodes. Therefore, a mechanism is needed to find the physical location of the nodes [3].

Perhaps, the first solution, which comes to the mind, to receive the physical information, is to use the public location finding system, but for different reasons, the use of such policy does not seem to be logical in this regard. First, GPS receivers need a lot of energy, and they quickly deplete sensor node energies, which decreases the life of the whole nodes. Secondly, equipping each sensor by the GPS receiver, involves high hardware costs on the network, remember that a price of each node is about \$ 10, and expenditure of a GPS receiver is about \$ 150 to \$ 200. Another reason is that the GPS has many problems in the indoor environments and its accuracy decreases sharply. Additionally, sensor networks should be independent of any infrastructure, and the GPS, is considered as an infrastructure [1].

1.2 Definition of Problem

The WSN is a group of tiny, energy-restricted sensors with limited giving out as well as communication competences. The wireless sensor network is designed to collect environmental information (such as seismic, audio, video, magnetic, and infrared data). The restriction of electricity sources is of the vital subject at every usage of wireless sensor networks. At wireless equipment (especially in wireless sensor networks for tracking targets), there must be an intercession between electricity in addition to pursuing excellence. Furthermore, constant use of radio equipment may take energy from sensor nodes, and result in the sensor being degraded, thereby reducing tracking quality [2].

Tracking and tracing moving targets are the most significant usages of WSN, which aims to recognize one or more moving objects and indicate their route direction during tracking. In other words, the problem of tracking in wireless sensor networks is defined as being a network of *n* wireless sensor nodes for tracking targets in an area. In this area, *m* targets are moving. Sensor nodes are sampling the detected signals, released from the targets to whether identify specific targets or not. In a particular wireless sensor network used for objective chasing purpose, all of the sensors can participate in detecting targets. Collected information by the sensor nodes is collected in a base node (sink). Then, reported data are used to estimate the route of movement of one or more moving targets. The target can be an animal, a car, a robot or a human being that moves in the network coverage. The objective chasing procedure may chase a particular target whereas hiding further targets at chasing area. Therefore, the essential challenges of tracking targets in wireless sensor networks are efficiency and tracking quality as well as the power consumption of the system. Tracking the targets at the maximum high speed and quality, as well as do not having an undesirable consequence over network presentation (for example, at energy, etc.), is the most crucial issue in this thesis [1].

In this thesis, a new method will be presented for tracking moving targets using the Kalman filtering and Machine Learning algorithm. In the proposed way, acceleration of moving targets is combined with information acquired from the network, to estimate and forecast position of versatile targets. Actually, in such a method, the database obtained by the machine learning algorithm estimates the starting points of the targets using the strength of the received signals, and estimates (or predicts) the route using the Kalman filtering. It is expected that the proposed method, improve the efficiency and total electricity usages of nodes [3].

1.3 Review of State-of-the-Art

WSN's are a kind of systems comprising a small number of small sensors in that there were several sensors in each node, and this sensor network continually interacts with the physical environment. Through the sensors, it takes environment information and sends it to a central station. The connection among the sensors are wireless, and every sensor works self-sufficiently deprived of person's involvement. These nodes are bodily slight in addition to have limitations in dispensation authority, reminiscence size, as well as energy supply. Such constraints generate difficulties that were the source of numerous of the investigation topics regarding this issue [1].

Today, life without communications is unthinkable. The advancement of expertise besides formation of slighter microcontroller boards makes it possible to use wireless circuits in most of today's smart equipment's. Such development makes the expansion of microsensors. These microsensors can perform various measurements in applications such as voice recognition for earthquake sensation. It also provides data collection in remote areas and places that are not suitable for human exploration. Cars can use micro-wireless sensors to control engine status, tire pressure, and oil balance. The assembly lines can use these sensors to control the production process. In tricky situations, micro-sensors can be distributed by an aircraft and then used to track targets (such as cars or humans) [1, 2]. The basic characteristic of such systems is its connection to physical milieu plus occurrences. Conventional systems deliver communication among persons as well as datasets, whereas WSN is straightforwardly related to the real world. These networks use nodes to detect the actual surroundings, decide about own seeing is in addition to perform proper operations.

Below are some of the wireless sensor network applications listed:

1. Armed (as an instance, tracing targets)
2. Well-being (as an instance, control of spirited indicators)
3. Milieu (as an instance, regular habitat analysis)
4. Entertainment (for example, virtual game)
5. Digital life (for example, tracking the car park location)

1.3.1 The general structure of the wireless sensor network

First, several critical definitions are mentioned:

- *Sensor:* An equipment, which converts a physical phenomenon into an electrical pointer. The sensor has various types as an example of heat, heaviness, moisture, light, accelerometers, as well as magnetometers sensor.
- *Agent:* Devices that actuate a particular action by actuating a key, for example, inaugurating and shutting down one tap or changing one key.
- *Sensor node:* Called the node that contains only one or more sensors.
- *Agent node:* Called the node that contains only one or more factors.
- *Sensor/Agent node:* Called the node that is equipped with a sensor and agent.
- *Sensor network:* One net that merely contains sensor nodes. Such nets are one specific kind of sensor/agent network. In applications that aim to collect information and research on a phenomenon, like studying on tornadoes.
- *Field:* The area where network nodes are distributed.
- *Sink:* The node that collects the data, and sets up the connection between sensor nodes/agent nodes and the task manager node. The sinks are of two types: the mobile sink and the static sink.
- *Task manager node:* One node in that a user as admin of the net interacts with the rest of the nodes. The control commands, as well as queries, were submitted as of such node toward the net, and the information collected is delivered it.
- *Sensor/Agent network*: A network consists of sensor and agent nodes, or a sensor/agent, which is the general state of the networks under discussion. In other words, the sensor network/agent network is a node with a large number of sensors; every sensor could generally have a number of sensors as well as several factors. In a particular case, a node may only be a sensor or agent. The nodes are dispersed in a region where the sensor/agent field is called high

density. Sinks monitor the whole network. The management of tasks performs data collection, and the commands are distributed through the sink. The sensor/agent network can be centralized or distributed. Depending on what level of decision-making is to be made on the reaction, there are two different automated and semi-automated structures that can be used [2].

1.4 Importance and Necessity of This Research

The efficiency and quality of tracking required for high-speed targets is a significant challenge in tracking targets in WSN. The chasing of versatile objectives should be done in such a way as to guarantee the quality of tracking. Objective chasing with use of WSN requires the variety of challenges to be considered, including energy limitation of sensor nodes and high probability of loss of target. While comprehensive research has been done in this area, but it remains an exciting issue for research. Therefore, this issue has always been of interest to researchers. Consequently, it is essential to provide new methods aimed at enlightening the excellence of tracking targets in WSN's.

1.5 Hypothesis

- We can use machine learning for estimating the location of objectives in WSN.
- With the use of Kalman filtering, we can predict the route of moving targets.
- Combining machine learning and Kalman filtering would be applying for versatile chase targets with high-performance WSN.

1.6 Thesis Purposes

- **Scientific goals:** In this research, the main goal is to track moving targets. To achieve this goal, in this thesis, a new approach will be proposed, that uses

Kalman filtering and machine learning to track targets in wireless sensor networks.

- **Practical purposes:** Such purposes are consisting of:
  - Permanent Coverage of Moving targets
  - Improving the quality of target detection
  - Uniform and balanced energy consumption in all nodes
  - Improved tracking of targets without affecting the net routine (such as energy consumption).

1.7 Novelty and Innovation

Fuzzing the acceleration parameters of the vehicles, the availability of the car, the location of the car, and the use of these parameters in routing, with the improvement of the delay in multimedia applications, is the main innovation of this thesis. Additionally, the data to be sent also has time limits (they are real-time), because the environment is a multimedia and real-time situation, and the provision of routing method for such a situation is also an innovation in this thesis. According to the search in the scientific databases, this has not been done before in multimedia applications, taking into account such parameters and fuzzing of them.

1.8 Organization

In the second chapter, the literature of research is reviewed, in a way that makes it easier to understand the thesis. In the third chapter, the proposed method is expressed, and detailed explanations are given to understand the proposed method fully. In the fourth chapter, the evaluation is done on the proposed method and is associated with existing techniques. In the last section, conclusions and suggestions will make to conclude overall work, as well as future approaches that can be made on the proposed method to improve the proposed method.

1.9 Simulation Plan

In the solution, which will be presented in this thesis, the environment is considered as two-dimensional. Of course, this method can be applied to multidimensional situations with increasing variables and incremental measurements. Each of the targets in the background is articulated by a circle with radius R and center *(x, y)*. This means that the target here is seen as a circle, which is considered the center of the target by moving only the new *(x, y)*. Therefore, in the first step of the proposed method, all targets should be specified as (x, y). The proposed approach will be capable of predicting $N + 1$ location using Kalman Filter by having N items from different target locations in a two-dimensional environment.

In this proposed method, we will use the ELM (Extreme) Machine Learning algorithm, which is a kind of neural network. ELM is one of SLFN's (Single Hidden Layered Feedforward Neural Networks) learning algorithm. Tracking is a real-time and should be performed with algorithms, which have high speeds. Using of standard learning machines, such as Support Vector Machine (SVM), has a low speed; so the speed of the targets should be very low, while in the actual applications, we do not have such, in addition to the speed of the objects that we intend to track may be very high. In this thesis, we will introduce new real-time and very high-speed tracking system, which will use ELM and Kalman Filtering. In recent neural network structures, SLFN is very common and used. In the real world, it is demonstrated that whether the submission function would have calculated correctly, then SLFN could estimate each task with the least possible error.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the basic concepts and fundamentals were expressed, until the understanding of proposed method become easier. Also in this chapter, techniques, which are close to the proposed method, are evaluated and discussed about each of these methods.

2.2 Wireless Sensor Networks

Wireless sensor network is a network consisting of a number of small nodes, in which there are several sensors in each node, and this sensor network continually interacts with the physical environment. Through the sensors, it takes environment information and sends it to a central station. The connection amongst sensors are wireless, and every sensor acts self-sufficiently deprived of persons' involvement. These nodes are bodily tiny besides having limitations at giving out supremacy, memory volume, as well as energy supply [1]. Such constraints make many difficulties, that were a source of numerous of the investigation topics regarding this issue.

Today, life without communications (wireless communication) is unthinkable. The advancement of knowledge plus formation of small microcontroller boards has created it possible to use wireless circuits in most of today's automated equipment's. Such development paves the way for the advancement of microsensors. These microsensors can perform various measurements in applications such as voice recognition for earthquake sensation. It also provides data collection in remote areas and places that are not suitable for human exploration. Cars can use micro wireless sensors to control engine status, tire pressure, and oil balance. The assembly lines can use these sensors to control the production process. In tricky situations, micro-sensors can be fired by aircraft on enemy regions, and then used to track targets (such as cars or humans).

The primary characteristic of these networks is their connection to the physical environment and phenomena. Traditional systems provide communication between humans and databases, while the sensor network is directly related to the physical world. These networks use the sensors to sense the physical environment, decide on their observations and perform appropriate operations.

Below is the list of some of the wireless sensor network applications:

- Military applications (for example, tracking objects)
- Health applications (for example, control of vital signs)
- Environmental applications (for example, natural habitat analysis)
- Fun (for example, virtual game)
- Digital life (for example, tracking the car park location)

2.2.1 The general Structure of Wireless Sensor Network

Before introducing the general structure of the wireless sensor network, several critical definitions are mentioned:

- *Sensor:* An instrument, which converts a physical phenomenon into an electrical signal. The sensor has not the same categories, such as temperature, pressure, humidity, light, accelerometers, and magnetism sensors.

- *Agent:* Devices that perform a special action, by actuating a key, comprising of turning on and shutting down one tap or changing one key.
- *Sensor Node:* Denoted to one node that contains merely one or additional sensors.
- *Agent Node:* Denoted to one node that contains merely one or additional agents.
- *Sensor/Agent Node:* Denoted to a node that is equipped with a sensor and agent.
- *Sensor Network:* One network that merely contains sensor nodes. Such system is a specific kind of sensor/agent networ in applications that aim to collect information and research about one phenomenon, like studying on tornadoes.
- *Field*: The region in which the network nodes are distributed.
- *Sink:* The node that collects the data and sets up the relationship between sensor nodes/agent nodes and the Task Manager Node. The sinks are of two types: the mobile sink and the static sink.
- *Task Manager Node:* One node in that an individual in place of operator or system manager interconnects through the net. The control commands, as well as queries, were submitted through such node toward the net, and the collected data is given back toward such node.
- *Sensor/Agent Network:* A network consists of sensor and agent nodes, or a sensor/agent, which is the general state of the systems under discussion. In other words, the sensor/agent network is a network with a largenumber of nodes; every node could generally have a number of sensors as well as several agents. In a particular case, a node may only be a sensor or agent. The

nodes are dispersed in a region, which named as the sensor/agent field, with high density. A sink monitors the whole network. Data collection is done by the sink of the management of tasks, and the commands are distributed through the sink (Fig. 2.1). The sensor/agent network can be centralized or distributed. Depending on what level of decision-making is to be made on the reaction, two different automated and semi-automated structures can be used.

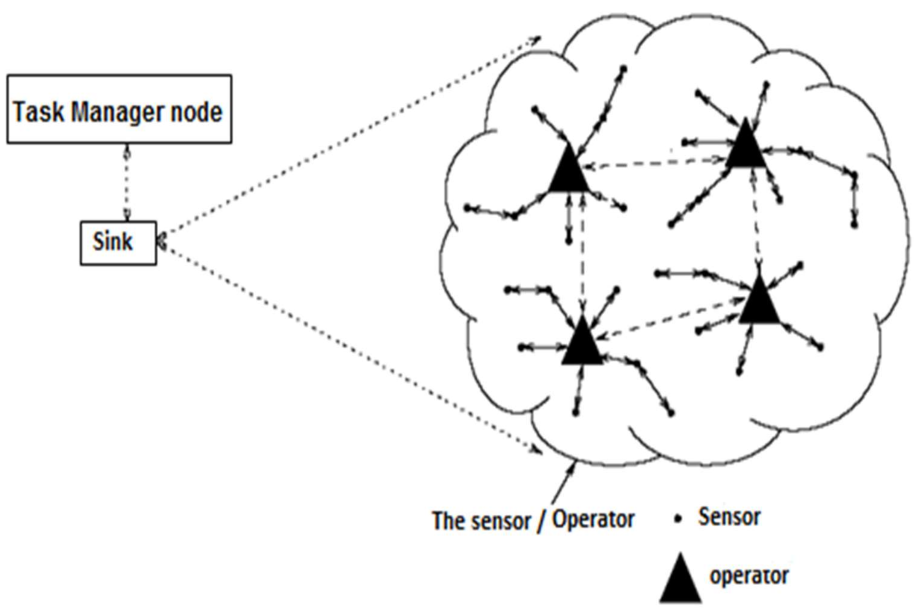


Figure 2.1: General structure of sensor/agent network [2].

- *Automatic Structure:* Nodes that distinguish one occurrence or an event, submit acknowledged information toward the agent, for processing plus responding correctly. The neighboring agent nodes, work organized to coordinate each other. No centralized regulator existed, besides resolutions were prepared nearby (Fig. 2.2).

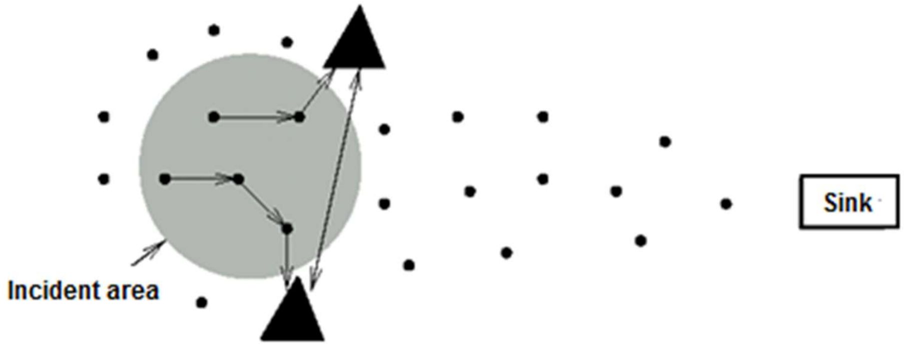


Figure 2.2: Structure of automatic [2].

- *Semi-automatic structure:* At such assembly, the information is directed through the sensors toward the sink, besides the command is passed through the sink to the agent nodes (Figure 2.3).

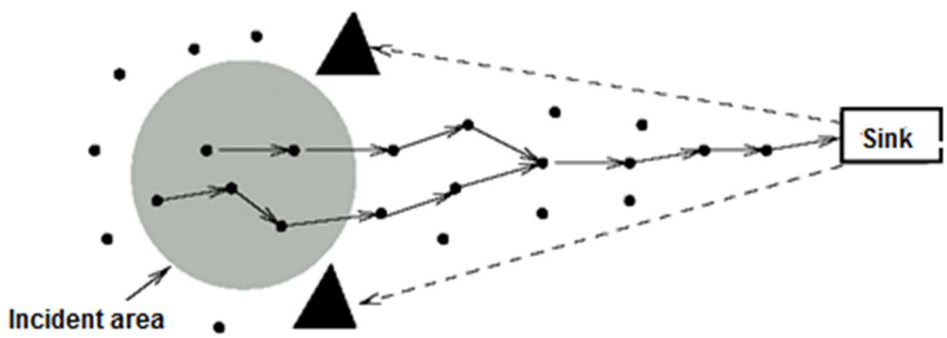


Figure 2.3: Structure of Semi-automatic [2].

On the other hand, in specific applications, a partitioned or cellular structure may be used, in which, in each section, there is a cluster that sends the data of its cluster nodes to the sink. Each group acts like an entry [1].

2.2.2 Characteristics of Sensor Network

The presence of some features in the sensor/actuator network distinguishes it from other traditional and wireless networks, which includes:

1. Hardware bottlenecks; including physical size limitations, power source, processing power, memory capacity.
2. Too many nodes.
3. High-density distribution of nodes in the operating area.
4. The existence of the potential failure in the nodes.
5. Dynamical changes of topology, and possibly alternating.
6. Use of the public distribution method for communication between nodes, versus point-to-point connection.
7. The ability of data-driven means that the nodes do not have the identification code.

Sinks, in sensor networks, are known as raw data collectors of sensor nodes and the processing's on them, and delivered to the end user [11]. In other words, the sink is the gateway between the user and the sensor node. In sensor networks, sinks can be mobile or fixed. Recent researches have shown that mobile sinks have many advantages over fixed sinks, which are [1 – 3]:

1. The mobile sink, can move across the sensor network, but the fixed sinks do not have this capability and are usually located in a predetermined location. Therefore, the fixed sinks can consume a lot of energy, because the data is moving in specific directions, while the mobile sinks can move in such a way that they include goals such as energy reduction [3].

2. Mobile sinks increase the sensor network lifetime and reduce energy consumption of sensor nodes [1], while the fixed sinks, depending on where it is located in the network, causes low or high-energy dissipation of the sensor nodes, as well as in the terminal nodes that are the connected to the fixed sink, creating a throat.
3. The mobile sink, in the sensor network, is not accessible and identifiable for people who should not have access to the sink or network, because they are always on the move. This feature is more important in combat areas.
4. Failure tolerance, in sensor networks that use a fixed sink, is low. If the sensor network uses one or more sinks, part or the entire system will fail in the event of problems in either of these sinks. Depending on where you are using the sink, fixing the problem, it will be time consuming or difficult.

2.3 Clustering

Clustering is a way to reduce the energy needed to exchange information, which increases the lifetime of the sensor nodes. In a cluster-based model, the network is divided into clusters, each of which contains several nodes. The header of each cluster inside that cluster is responsible for routing information, to send to the head of the other cluster groups or sinks. The head of the cluster is selected to perform various tasks such as collecting data, composing and sending them [8]. Because of data transfer from one cluster to another cluster or sink, this model covers more distances.

2.3.1 LEACH Clustering Protocol

The Low-Energy Adaptive Clustering protocol (LEACH), is the first and most prominent clustering protocol in wireless sensor networks. Such an algorithm makes clusters in a distributed architecture. The primary purpose of LEACH is to have ground-based stations (head clusters) to reduce energy consumption due to the transfer of data to a remote base station. LEACH selects a few sensor nodes randomly as cluster headers, and organizes local nodes, as regional clusters. The assignment of the nodes to the appropriate cluster head, is based on the closeness (distance). Non-cluster head nodes (called normal nodes) transmit their data to the cluster headers. Therefore, the only overhead that is existed here is intra-cluster communication. The cluster head nodes requires further electricity than ordinary ones. As a result, the selection of immovable head cluster nodes makes into the early evacuation of energy plus their quick expiry. Energy equilibrium of the cluster heads is rotated through the rotation of the cluster head between the various nodes. Also, the use of data collection/aggregation in cluster heads reduces the quantity of message submitted toward the base station and saves energy. The operation of the LEACH protocol is alienated into several slices. Every slice begins by the setting up segment (the creation of clusters), in that the clusters are organized (Figure 2.4).

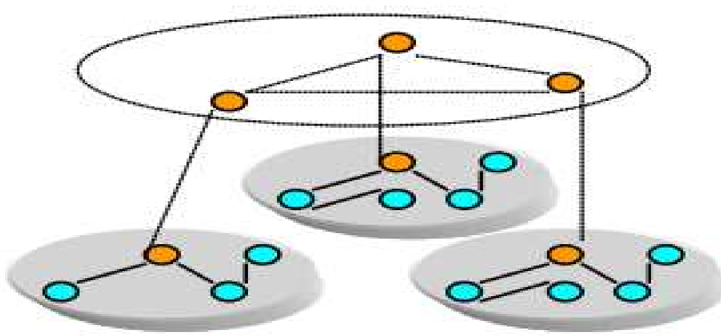


Figure 2.4: Structure of clustered sensor network [4].

Following the installation phase, the data transfer step is located, in which ordinary sensors submit their data toward the cluster headers, and the cluster headers, later the assembly /integration of information, transport the integrated packet to the base station, to reduce the amount of posting information that must be provided to the base station. In LEACH, the timing of sending sensor data is performed by code division multiple access (CDMA) or by time division multiple access (TDMA). Selection of the cluster head is accomplished by the probability function. Every node selects an accidental digit amid zero and one, and if the nominated number is not as much of T(n), that node is designated as the existing cluster head (2-1) [5].

$$T(n) = \begin{cases} \dfrac{P}{1 - P(r \bmod \frac{1}{P})} & if\ n \in G \\ 0 & otherwise \end{cases} \tag{2-1}$$

Where:

P: Probability of clustering

r: Current round number

G: Set of nodes that have not been the cluster head in the last *1/p* round

In the (2-1), *P* is the probability of clustering, *r* is the current round number, and *G* is the set of nodes that have not been the cluster head in the last *1/p* round. Derived from the simulation model, it has been proven that merely *5%* of the nodes required being cluster heads. LEACH's strength point is in the rotation mechanism of the role of head clusters and assembly of data and can increase network lifetime, but it also has some disadvantages:

Primary, it assumes that the entire network nodes have enough energy for submitting data toward the base node as well as having sufficient computing power for provisioning different MAC protocols. Therefore, it cannot be applied to large-scale networks. That similarly undertakes which of the sensors permanently has information for submitting, and the adjacent sensors have dependent data in comparison with similar ones. This protocol assumes that all nodes in each round are selected to begin, with an equal amount of energy; with this assumption that the cluster head is assuming energy just like other nodes. The greatest significant fault in LEACH is that it is not known that in what way the scheduled number of clusters (i.e., *p*), wants to be circulated uniformly across the network (Figure 2.5).

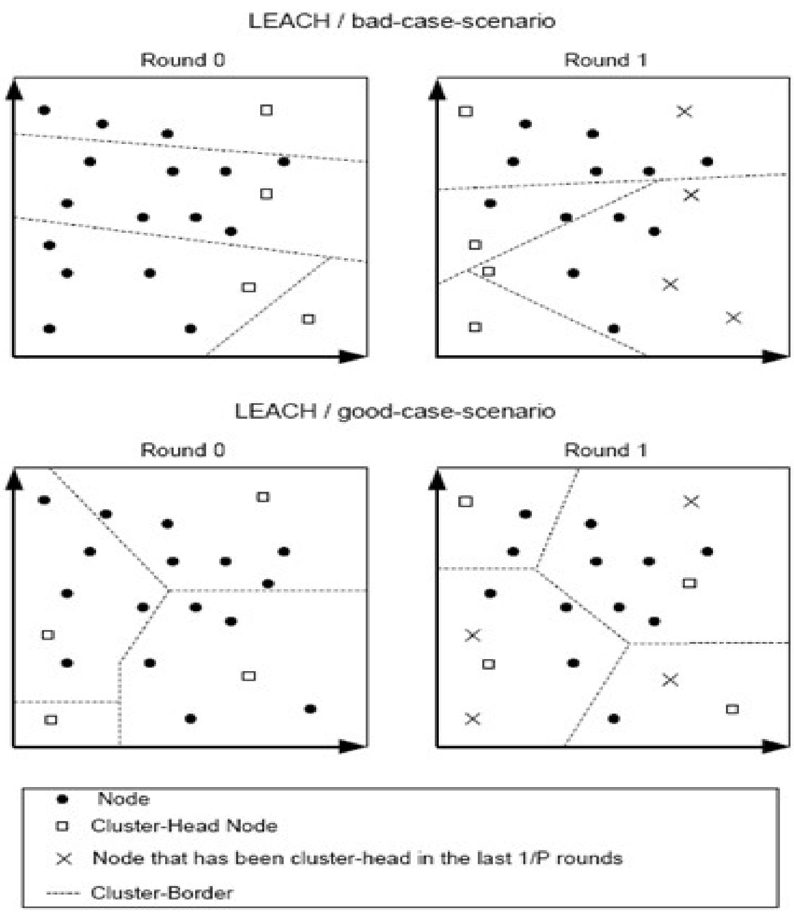


Figure 2.5: An example of LEACH network [7].

There is no agreement for the position or quantity of clusters in every round. Therefore, it is conceivable that the designated cluster heads are focused on a fragment of the network. An answer to such difficulty can be the use of a centralized clustering procedure.

2.3.2 The Centralized LEACH Clustering Protocol

Centralized LEACH is a clustering procedure in which clusters are formed centrally and by the base station. This algorithm has the same phase of data transfer (static) mode as the LEACH algorithm, during which every node submits data regarding its status as well as its present level of energy toward the base station. Typically, it is assumed that every node has a GPS. The base station must ensure an even supply of energy between all of the clusters. Therefore, the base station set thresholds for the energy level, and select nodes as probable cluster headers, that have an energy level, greater than the threshold. The problem of determining the optimal number of clusters is an NP-hard problem. The LEACH-C uses the simulated annealing algorithm [34] to solve this problem. The base station submits a message comprising the cluster ID to every node, after setting the current cluster heads. If the ID of a cluster node is compatible with its ID, that node is a cluster header. Otherwise, it is a normal node, and it can be put into sleep mode, until the data transfer phase, that itself is in place. LEACH-C is more competent than LEACH, as well for every part of energy; it would be handover further 40 percent information. Because the base station has a global knowledge about the position and battery level of the network nodes. Also, LEACH-C, unlike LEACH, assurances the optimal amount of clusters (k) per round.

2.4 Routing Features in the Wireless Sensor Network

Routing in sensor networks is very challenging, because of the intrinsic characteristics of these types of networks, which are distinguished as of additional wireless nets, such as movable contingency nets, or mobile nets. The primary feature is that, because of the great number of such nodes, it would not likely in creating one routing structure to accommodate an extra amount of such nodes, since ID loads are very high. Therefore, customary IP-oriented conventions are not applicable for WSN [10]. Additionally, such nodes that were placed at the ad-hoc scheme must be self-organized. Because of the placement of ad-hoc, such sensors necessitate systems that set up communications and manage the distribution of nodes, primarily because unauthorized sensor network operations must be performed. In wireless sensor networks, occasionally receiving information are extra significant than perceiving the

ID of the information to be sent. Secondly, unlike conventional communication networks, nearly the whole usages of sensor networks necessitate the transmission of detected information from multiple sources to a specific base station. Although such a feature does not stop the stream of information into other systems (e.g., multi-distribution of data or point-to-point data transfer). Thirdly, such nodes were severely restricted regarding electricity, dispensation, as well as saving capabilities. Therefore, nodes need accurate supply administration. Fourthly, every applicable situation, sensor network nodes, except some of the moving sensor nodes, are usually fixed after placement [11]. Nevertheless, in other traditional wireless networks, nodes are free to move, so the topology of these networks is always and unpredictably changing. While at more or fewer usages, many of such nodes probably able in shifting plus altering its position (albeit by less flexibility). Fifth, such nets were usage-oriented (means that, the necessities of sensor net design, vary according to the application). For example, the challenging war observation problem that needs to be done accurately and with little delay is different from the climate monitoring survey. Sixth, the state awareness of the sensor nodes is essential, because data collection is usually based on location. This status of consciousness may be met by GPS, but it is also possible to have GPS independent methods, for the problem of locating in sensor networks [12]. Finally, the data collected by many sensor nodes in wireless sensor networks usually relates to a single phenomenon; so the possibility of redundancy in these data is very high. This redundancy should be extracted by direction-finding rules, in the direction of increase electricity plus bandwidth usages. Typically, wireless sensor networks were information-driven, because information was demanded according to specific features (means that, feature-oriented locating). Feature-oriented locations are a combination of onequotient query-amount couples. As an instance, whether the request was a bit similar to {over 65° temperatures}, only such nodes which feeling the heat above 65° should respond and report their reading [13].

Also, in the routing protocols of wireless sensor networks, consider the following points:

1. *Coverage:* In wireless sensor networks, each sensor node acquires a particular perception of the environment. The sensor's knowledge over the situation is limited to its range and precision, and it may only cover a limited physical area of the situation. Therefore, area coverage can be an essential design parameter in wireless sensor networks.
2. *Data Collection:* Because sensor nodes may generate significant, redundant data, the same data packets generated from multiple nodes can be integrated to reduce the number of transfers [6 -14]. A data collection is a combination of data from various sources that are based on a particular community function (such as maxima, average, duplicate, suppression, minima). This method is used in some routing protocols for energy efficiency and data transmission optimization. Signal dispensation approaches could similarly be applied aimed at information collection. At such situation, it was named by way of an information combination (data combination), that the node will be able to produce a more accurate output signal, by using of methods such as beamforming, with the combination of input signals and noise cancellation of such messages [15].
3. *Quality of Service:* In some applications, the data must be sent within a given period of sensation. Otherwise it will be useless. Therefore, the limited delay in data transmission is another condition of modern applications. However, in

many applications, energy storage that relates precisely to the lifetime of the network is relatively more important than the quality of the service [16]. It may be necessary to reduce the quality of the results to reduce the energy loss in the nodes and increase the network lifetime. Therefore, energy-awareness routing protocols often need to sacrifice this feature (quality of service).

2.5 Kalman Filter

The Kalman filter also referred to as the second-order linear filter, is a procedure that predicts the state of an active arrangement, using a set of measurements including error over time. This filter usually provides a more accurate estimate, than the forecast according to one particular amount based on Bayesian extrapolation as well as the approximation of likelihood distribution of one random variable in a time interval. This filter was named by the name of one of the founders of this algorithm, Rudolf Kalman.

However, Thorvald Nicolai Thiele, as well as Peter Swerling, had already offered the same algorithm, this filter was named in honor of Rudolf Kalman, as the Kalman filter; and Stanley F. Smith was generally known for developing the first implementation of the Kalman filter. This incident occurred, when he met Kalman at the NASA Research Center, and he saw the utility of Kalman idea in estimating the route to launch the Apollo project, which resulted in its annexation to the Apollo navigation computer. This filter was created on paper, in 1958 by Swerling, and developed in 1960 by Kalman and in 1961 by Kalman and Busey.

Kalman filter has many applications in science and technology, such as tracing and tracking of vehicles, especially airplanes and spacecraft. Kalman filter introduces many concepts in the field of time series, signal processing, and econometrics. This filter is a fundamental concept in the planning and monitoring of robots as well as modeling the nervous system. Depending on the time delay between sending commands and receiving their answers, the use of the Kalman filter makes it possible to estimate the different states of the system.

This algorithm runs in two stages. At the prediction stage, the Kalman filter provides an estimate of the present state of the parameters in indeterminate circumstances. Once the next measurement result is attained, the earlier estimate is updated by the weighted averag in the way that, the weight of data that is more certain, will be greater. The recursive algorithm is implemented in real time, using new inputs and prior calculated states.

Regarding the Kalman filter inputs, it cannot be said that all errors are Gaussian. Nevertheless, in practice, the filter performs probabilistic estimates with the assumption of normal distribution [15].

The basis of the dynamic system model of Kalman filters is derived from discrete dynamic linear systems, over time intervals. They are modeled based on the Markov chain, constructed by using linear operators, and triggered by the Gaussian noise, and the vector of real numbers expresses the state of the system. In each time increment that happens in a discrete interval, a linear operator is applied to the current state, to produce the next state with some noise, and if the controllers of the system are known, extracts certain related information; then another linear operator with some amount of output noise is generated from this unspecified state [16, 17].

The Kalman filter is capable of acting like an uncertain Markov exemplary. By this main alteration, which, the unspecified state parameters were located in one incessant environment (the opposite point of the discrete mode space at the Markov model); besides, the Markov model could provide a random circulation, aimed at the following values of the state of the parameters, which contradicts the Gaussian noise model, which was applied in the Kalman filter. Here is a big contrast among Kalman filter formulas and Markov model. One of Kalman filter applications is to predict and alert, for example, Kalman filter can be used to predict and report a flood.

In control theory, the Kalman filter refers to the second-order approximation (LQE). Today, an extensive diversity of Kalman filters has created. As of the original Kalman formula, such filters have already been developed: simple Kalman, Schmidt developed information, and various Filters of Birman, Thornton and many others. It is believed that the most common type of Kalman filter is the phase locked loop, which today is used in radios, computers and almost all kinds of video and communication tools.

Providing up-to-date and accurate information on the location and speed of a given object, is only possible by sequence observations, about the position of that object, where each is containing some errors. Such an algorithm was applying at extensive variety of manufacturing usages, since sonar toward machine apparition. The Kalman filter method was one of the major topics in the control concept as well as the engineering of regulatory schemes.

As an instance, for sonar use, data regarding the position, velocity as well as hastening of objective would be dignified by one tremendous amount of deviation because of noise, in every moment. The Kalman filter uses target's changing aspects, in such a way as to control its evolution, so that it eliminates the properties of noise as well as provides a decent approximation of the objective's position in current (refining) and in the future (prediction) or in the past (insertion or flattening). A secure form of the Kalman filter was an alpha-beta filter, which is used usually; it uses still increment factors in place of covariance matrices [17].

Headed for approximation the internal state of a procedure, presented by the set of mixed observations with noise, it must be by the Kalman filter framework. So, the below matrices were used:

- F_k: State transition model
- H_k: The observed model
- Q_k: Process noise covariance
- R_k: The observed noise covariance
- B_k: Control-input model

The Kalman filter states that state of *k* can be calculated, using the state *(k-1)* and with the help of equation (2-2) [14].

$$X_k = F_k X_{k-1} + B_k U_k + W_k \qquad (2\text{-}2)$$

Where:

F_k: Transition state applied to X_{k-1}.

B_k: Input-control model applied to control vector U_k.

W_k: Noise process with normal distribution, mean zero and covariance Q_k.

At time *k*, the Z_k observation is obtained by (2-3) according to the state of X_k [14].

$$Z_k = H_k X_k + V_k \tag{2-3}$$

H_k: Is the observed model.

V_k: Is the observed noise.

X_k: Is the actual state vector.

So that H_k is the observed model, mapped to the observed region, and V_k is the observed noise with Gaussian distribution mean zero and $R_k$covariance.

It should be noted tha the initial state and noise vector are independent of each other in each location.

Many real dynamic systems do not follow this model. Some active systems can reduce the effect of this filter, even when it examines an unknown source of input. Because the impact of these systems affects the input, signal and thus can cause instability of the filter estimate. In addition, independent white noise does not spill out the filter. The issue of white noise separation and dynamic systems, in the branch of control theory and the framework of the robust control is discussed [15 -19].

The Kalman filter is a recursive estimator, which is the estimate of the previous state and the current observation is needed to calculate the current state estimate. On the other hand, many estimators do not need to keep, forecasts and observations of all the previous scenarios. Here $\hat{x}_{x|m}$ represents an estimate of x at time n, if observations are before this time.

Two variables describe the current filter state:

- $\hat{x}_{k|k}$: Estimating the posterior state at time k under the condition of observations before k.
- $P_{k|k}$: Late covariance matrix.

A formula expresses the Kalman filter; nonetheless, it is regularly divided into two parts: prediction as well as update. In the prediction phase, using estimates of states in previous periods, an evaluation is obtained for the present state. Such predicted estimate, is the same as last information, for the reason that it is only depending on the previous estimates, and does not comprise any observations in the present state of the system. In the update phase, the previous estimate combines with the current views, to provide a view of the current state of the system.

Typically, such two phases are recurrent regularly, which means that the prediction is made up to the following observation is performed, and then updated using the present observations. If no observation is done in the interval, the projections are made until the subsequent observation, and the update is done rooted from several prediction stages. Likewise, if quite a few independent observations are performed over some time, based on each of them several updates with different H_k matrices are obtained [20, 21].

Prediction

- Estimated prediction (previous) [14]: $\hat{X}_{k|k-1} = F_k \hat{X}_{k-1|k-1} + B_k u_k$ (1)
- Estimated predicted covariance (previous) [14]: $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$ (2)

In the above formulas:

$\hat{x}_{k|k}$: Estimating the posterior state at time k under the condition of observations before k.

$P_{k|k}$: Late covariance matrix.

F_k: Transition state applied to X_{k-1}.

B_k: Input-control model applied to control vector U_k.

Update

- New dependent observation [14]: $\hat{y}_k = z_k - H_k\hat{X}_{k|k-1}$ (3)
- New dependent covariance [14]: $S_k = H_k P_{k|k-1} H_k^T + R_k$ (4)
- Optimum Kalman result [14]: $K_k = P_{k|k-1} H_k^T S_k^{-1}$ (5)
- An estimate of updated state (previous) [14]: $\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k \hat{y}_k$ (6)
- An estimate of updated covariance (previous) [14]: $P_{k|k} = (I - K_k H_k) P_{k|k-1}$ (7)

In the above formulas:

H_k: Is the observed model.

R_k: The observed noise covariance

I: Identity matrix.

$\hat{X}_{k|k-1}$: Is the predicted state vector.

$P_{k|k-1}$: Is the predicted state error covariance matrix.

z_k: Measurement of the output.

2.6 Extended Kalman Filter

For Extended Kalman Filter (EKF), states changeover plus observations require linear or nonlinear functions. These are differentiable functions [14].

$$x_k = f(X_{k-1}, u_k) + w_k \quad (8)$$

$$z_k = h(x_k) + v_k \quad (9)$$

Where in the above formulas:

w_k: Is the process noise.

v_k: Is the measurement noise.

$function\, f$: Associates the state at the former time step $k-1$ to the state at the present time step k. It comprises as variables any driving function u_k and the zero-mean process noise w_k.

The nonlinear function h in the measurement formula associates the state X_k to the measurement z_k.

The *f* function can be used to calculate the predicted state of the previous estimate. In addition, *h* is used to find observation of the prior state. The functions *f*, as well as h, would not be used directly for covariance, but the matrix of fractional products (their Jacobi matrix) must be calculated.

At any given time interval, the Jacobi matrix is calculated using the previously predicted states. These matrices are used in Kalman filter equations. This process involves the linearization of nonlinear functions about existing approximation.

2.7 Survey of References

In this section, the previous methods are described in terms of routing and location finding. We intend to provide a new way of locate, also requires a routing method to send information from the nodes that performed the detection to the supervisor node or workstation. Since the energy used to transmit data in wireless sensor networks is large, it varies depending on the volume of data transferred and the transmission distance. In another word we can say, the higher the distance and data volumes, the higher the amount of energy consumed, so the proper routing methods should be used to reduce the amount of energy consumed needed to transmit data. In addition, methods for predicting can be used, to reduce the number of transmitted data and reduce energy consumption. For more reduction in consumed energy, the life of the sensor network will also increase, as noted earlier, the sensor nodes have limited energy and should try to reduce the energy consumption of the system, the later nodes will end their life, and the network life will increase. When node life is over, the area covered by nodes is reduced.

Target tracking methods in wireless sensor network are divided into two main groups, cluster-based and tree-based groups. Each of these methods can be implemented using prediction or without prediction. Cluster-based methods are divided into three categories: static, dynamic and hybrid (static-dynamic). From the perspective of processing and shaping, the structure of the wireless sensor network algorithms can also be divided into distributed and centralized groups. In a centralized method, a central node obtains information from the entire network (assuming the data is transmitted from all nodes to this central node) and then, based on this global information; the optimal structure (tree or cluster) is formed. In distributed methods, nodes form the desired structure for tracking by exchanging information with their neighbors.

Dengley et al. (2010) [20] have investigated the problem of tracking a group of moving objects (which simultaneously exist in the environment) in wireless sensor networks based on adjacent binary nodes. In this paper, a group of tracking algorithms is presented with each other cooperation, and with the help of adjacent binary sensors for tracking moving targets. The algorithms used in that, including a target detection algorithm, an error-tolerate algorithm, a dynamic reporter node selection algorithm, and control algorithms. Also, in the methods proposed by Kim et al. (2005) [21], Shir Yustova et al. (2006), Singh et al. (2007) [23], Smith et al. (2006), Zhang et al. (2004) [25] and Zhao et al. (2002) [26], were only possible to track a single moving target or only a limited number of moving targets in a wireless sensor network with binary adjacent sensors. Previously mentioned methods, do not have adequate performance in tracking multiple movable targets, and in many cases, regardless of the energy consumption of the network, tracking of targets are performed.

Nardaran et al. (2014) [27], presented a new solution for tracking multiple moving objects. In this method, we have tried to collect efficient data to achieve low energy costs and low storage space occupancy.

In Mostafaei et al. (2014) [28], an energy-awareness scheduling method, for border coverage and tracking of multiple mobile targets in wireless sensor networks is presented, by using of learning automata. In that paper, they applied a learning automaton to extend network lifespan, in which each node in the network is equipped with learning automata. The learning automata in each node has the task of selecting the node for borderline operations and tracking moving targets.

Benyuan et al. (2008) [29], examined the conditions required to create strong coverage in a wireless sensor network with adjacent binary sensors. At that paper, sensors belonging to the same and separate coverage, are specified and scheduled, to not simultaneously awake. That is, at any time, the sensors belonging to the two covers are not sleeping, so that the path does not go through. For this purpose, an algorithm called division and conquest is used in this paper.

John et al. (2013) [30], presented a distributed algorithm for tracking multiple targets in wireless sensor networks. The method presented in this paper is scalable and is used in any size of wireless sensor networks. In addition, the suggested approach is not limited to definite shapes and spaces. Yang et al. (2009) [31] presented an efficient energy-based sleep, based on energy scheduling method for tracking moving targets, based on the collaboration between neighboring sensors; that tries to reduce the number of active (awake) nodes.

Silvestri et al. (2011) [32], have introduced a distributed and asynchronous method for tracking moving targets. In this paper, network conditions are assumed to be considered, as mobile and movable nodes.

Zhou et al. (2012) [33], presented a method for tracking targets in wireless sensor networks based on how the sensor nodes move. In this paper, the sensor nodes are positioned in such a way as to, create a reliable coverage in the sensor network, and then detect moving targets.

In Yanjun et al. (2013) [34], two coverage algorithms are presented, based on a customizable model for tracking purposes in wireless sensor networks, in which sensors can measure their range of measurements (such as maximum, Small and tiny). The purpose of this article was to reduce the overlapping coverage area between sensor nodes, to help in saving energy, and thereby intensification the lifetime of the network.

2.7.1 Routing Methods in Wireless Sensor Networks

Based on the differences between the wireless sensor networks and other networks, many new algorithms have been proposed for the routing problem in wireless sensor networks, and several review studies have been conducted, such as [17] [18]. Such direction-finding procedures has brought into account, intrinsic properties of sensor networks, accompanied by specific usages as well as construction necessities of these networks. It is not simple to find, storing paths in wireless sensor networks, because energy constraints and sudden changes in node conditions (such as failures) cause frequent and unpredictable changes in the network topology. To minimize energy consumption, the proposed routing methods, use more or less of the popular routing techniques, such as data collection as well as internetworking processes, clustering, assigning unlike characters to nodes and data-driven approaches. Nearly all routing protocols are classified into three general categories, according to the structure of the network: flat, hierarchical, or location-based [19].

2.7.1.1 Flat Routing

In the extended network routing protocols, usually, all nodes play the same role, as well as work together for executing the sensory operation. Because of significant amount of these sensors, the global identifier cannot be assigned to each node. This feature leads to data-driven routing, in which the base station sends its request to certain areas, and waits for the responses of the nodes in the selected regions. Since the data is requested through queries, a feature-based naming is necessary to specify the data properties. Initial works on data-driven routing, SPIN (Sensor Protocol for Information via Negotiation) [18] and direct distribution [17], have proven that energy storage can be achieved by negotiating data and deleting jobless information. Such procedures have been a motive behind the plan of many further rules, which pursue the same idea.

2.7.1.2 Location-based Routing

In this type of routing, the sensor nodes are managed along with their position. The distance among neighboring nodes is projected based on the strength of the input signals. The approximate coordinates of neighboring nodes are obtained by exchanging this information between neighboring nodes. First, we divide the network into immovable regions, plus making one cybernetic network. Within every area, sensors work together to play different roles. The nodes referring to the single point in the system, were measured comparable (as the price) for closed packet direction finding. This was equivalent to holding more or less of the sensors at sleep mode in the grid area, to save energy; for example, nodes will select a sensor node, that remains awake for a certain period, and the rest of the nodes will sleep. This node is responsible for monitoring and reporting data to the base station, from the nodes of that area. Among the most prominent protocols in this category, are GAF (Geographic Adaptive Fidelity) [37] and GEAR (Geographical Energy Aware Routing) [38].

2.7.1.3 Hierarchical Routing (Based on the Clustering)

In hierarchical routing methods (cluster-based), more energy sensors could be applied to dealing with as well as submit data, even though fewer energy sensors, could be used in running sensory near the target. Actually, in hierarchical process, by creating clusters and assigning specific tasks to cluster heads, can have a significant contribution to the scalability, lifetime, and overall energy efficiency of the system, and avoid single-strike architecture. Hierarchical routing, by performing a collection and data combination, to reduce the number of messages, sent to the base station is an efficient way to reduce energy consumption in a cluster. Hierarchical routing is principally a two-layer routing, where a layer is used to select cluster headers, also another layer used for routing. However, most of the technologies in this category are not about routing, but more about who and when they should submit or process the data, the channel division, and so on.

Clustering routing approaches were potentially the effective approaches for reducing energy consumption in sensor networks, and have been used in many applications in these days. The most prominent protocol of this category, are LEACH [39]. This protocol inspired the development of many other clustering protocols, each of which attempted to resolve some of its problems. Due to the importance of the LEACH protocol, and its direct relationship with the work done in this thesis, we will continue to describe it precisely.

2.7.2 Location Finding Methods in Wireless Sensor Networks

2.7.2.1 Moving Guide Method

One of the methods that uses reference nodes, and assumes that the reference node can move, is called the Moving Guide. In this method, sensor nodes are considered constant and distributed randomly in the environment. There are a few reference nodes in the background that, by moving in the background, in different forms and sending location messages at specific times, locate the entire other nodes. This method uses a distributed mechanism for location finding, and each node is informed of its location. Also, the directional moving guide does not need to be directed to the antenna, since the reference node locates each node from three different places, and calculates the exact location of the node using three corners. The signal strength method is also used to estimate the reference node distance to the sensor. This method is a one-way method, and each node should be exposed directly to the reference node, to locate which of nodes, that can be achieved by moving the reference node. Therefore, we must consider this method as a distance-based method, in which the length is measured [6].

Since this method usually uses a high-density reference node and this node has unlimited energy, this method is also referred to as a single-reference Mobile Beacon method in some resources [7].

2.7.2.2 Central Node Method

In the central node method, the constant number of the reference nodes is arranged in such a way that each sensor node is placed in the communication coverage of a number of them. The reference nodes create overlapping regions relative to the position of the sensor node. These reference nodes, periodically release their location messages. Other sensor nodes use a connectivity criterion, which determines the percentage of connection of the sensor nodes to each reference node, to decide whether to use that reference node, in determining their position. Each sensor node interacts with several reference nodes with the minimum amount of connections assigned to that node, in its location finding of itself. This is a non-distance-based method. However, the sensor node makes dependence the connection or disconnection of itself to the strength of the received signal from the reference node but only use this signal strength in its location finding. Therefore, in other features, it is familiar by the Moving Guide method, and its advantages and disadvantages are similar to the advantages and disadvantages of the Moving Guide method. It can only be said that the accuracy of this method is slightly lower than the previous way, and this is due to the independence of distances of this method. However, although this method consumes a high cost and uses a moving reference node, it does not have much consistent performance and is less useful [8] [9].

2.7.2.3 The Average Load Distance Method

The average load method uses only two reference nodes, one in the horizontal direction and the other in the vertical direction. For example, one-tenth or eventually one-fifth of the length of the environment or the width of the situation. If we could have radio-range nodes at an extent to the length of the situation, there would be no need for location finding. The present method in their assumptions assumes that reference nodes with radio range equal to half of the length and width of the positioning environment of available; that assumption does not appear reasonable, unless the network of small sensors used, in this case, the accuracy should be very high, and in such systems, due to the short distance, a robot can be used to pick up the sensor nodes. In this way, the reference nodes move along the width and height of the network and distribute location-finding messages. In this case, both reference nodes pass through the center [10] [11].

2.7.2.4 Semi-Definite Programming Method

Semi-Definite planning is one of the computational methods in geometry science. In this method, the constraints and geometric properties of a set of points are given by a matrix as input to the system, and the coordinates of each point are received as outputs. Of course, the level of constraints must be such that it can yield a single coordinate for each location. In semi-definitive programming, first, the properties of node connections in the environment are derived from non-distance-based methods, and certain constraints such as triangular relations are obtained. This information is collected in a matrix, and after calculations, the location of each node is obtained. Therefore, the method is a centralized method that uses a constant reference node. In this method, there is no need for the directional antennas, but the directional antenna can create more constraints. This method is a multi-step approach, and the number of steps, itself is a constraint [2, 9, 12].

2.7.2.5 Multi-Dimensional Scaling

The multidimensional scaling finds location method also uses geometry science and transforms the matrix of distances into point matrices. The primary difference between this method and the previous method is that this method is based on distance, and the last way is non-distant. Therefore, the precision of this method is higher than the last one [10]. One of the challenges of the MDS method is to convert local maps into the final plans. Given that the node range is limited, you need to divide the environment into sectors and locate each section separately, then combine the information of the different parts and get the final map. Usually, there is a great deal of error in computing, at this stage, and removes the error from the optimal one. Another problem with this method is its extensive calculation [5 - 13].

2.8 Conclusion

In this chapter, the basic concepts were expressed, and the information about the research resources was expressed; also in the next section, the proposed method is described, which is based on the Kalman filter. In this chapter, basic concepts of Kalman filter is introduced, to make it easier to understand the next section, which is to use it in the field of location finding.

Chapter 3

PROPOSED METHOD

3.1 Introduction

Recent advances in radio and embedded systems, have led to the emergence of Wireless Sensor Networks (WSNs); which is a significant field of research in last years. These networks are fast-paced for many applications such as home monitoring, monitoring Industrial and application-based software's.

Target tracking is one of the fields of research and applied disciplines in wireless sensor networks, which includes a quick estimate of the position of a moving target. Target tracking is seen as a successive local problem. Therefore, usually, a real-time estimation algorithm is needed.

The method, which is explained here, attempts to track and trace the target in the shortest possible time. Tracking of objects with using the Kalman filter and the learning machine is not a new concept, but the techniques, which were mentioned, previously are not high-speed tracking algorithms. On the other hand, real-world tracking applications and tasks should be done in real time. Here, the proposed method is capable of tracking through the Kalman filter in real-time.

3.2 Proposed Method

To elude the filter variance issue in target tracking produced through the unidentified or altering arithmetical representative of the noise at the Kalman filtering, an innovative Extreme Learning Machine created adaptive Kalman filter-chasing method are suggested at such thesis. Through understanding variance among hypothetical covariance as well as applied covariance of the origination that was, explain by way of dimension deposit over Extreme Learning Machine, the adaptive parameter of the covariance matrices of the seen smash is attained. After that, the covariance matrices of the perceived noise could do attuned accessible in line with the ELM learning information.

The steps of the proposed method are as follows:

1. The real location of the target object comes to the system.
2. The estimated location of the target with the use of the Kalman filter is available.
3. The difference between the above locations is calculated.
4. The accuracy of the system will be validated.
5. The output data are given to the ELM suggested system as the training set.
6. Factor justification of the n will be done.

In the proposed way, which will be explained here, the environment is considered as two dimensions. Of course, this method can be applied to multidimensional situations with increasing variables and incrementing the calculations of variables. Each of the targets in the environment is expressed by a circle with radius R and center x, y. This means that the goal here is seen in a circle that is considered the center of the target, and by moving only the new x, y which is concerned about center of the circle is considered. Therefore, in the first step of the proposed method, all targets should be specified as x, y. The proposed method can predict the $N + 1$

location using the Kalman filter by having *N* items from different target locations in a two-dimensional environment.

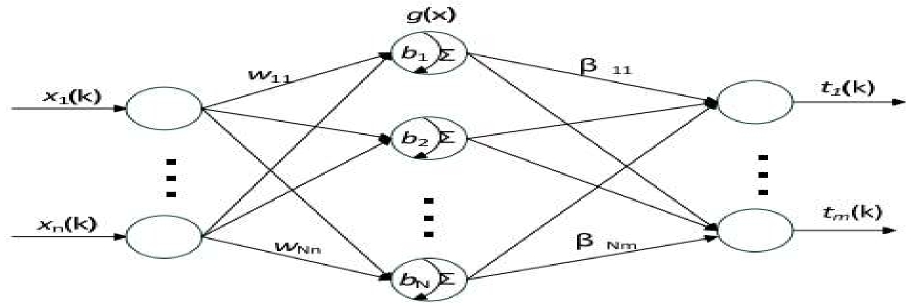


Figure 3.1: The general structure of SLFN algorithm with inputs and outputs [9]. In this structure $x_n(k)$ are inputs, W_{Nm} are input weights, g(x) is trigger function, b_N is threshold, β_{Nm} are output weights and $t_m(k)$ are outputs.

In this proposed method, the ELM machine learning, which is a kind of neural network, is used. ELM is one of the learning algorithms of SLFNs. Tracking is a real-time application and should be performed with algorithms that have high speeds. The use of common learning machines, such as an SVM machine, has a low speed. As a result, the speed of the targets should be very low; while in the real world, this is not the case, and the speed of the objects that we intend to track may be very high. In this thesis, using the ELM algorithm and Kalman filter, a new method is described that has a very high speed, and almost can be said to track moving objects in real time. In the structure of the current neural networks, SLFN is one of the most commonly used neural networks that is used frequently. In the world of science, it has been proven that if the trigger function is designated correctly, the SLFN can estimate any function with the least error [20].

The SLFN configuration is displayed in Fig. 3.1. This network has *N* hidden nodes, *n* input nodes, and *m* output nodes. For the *M* independent sample (x_i, t_i), $x_i \in R^n$ and $t_i \in R^m$. Standard SLFN with *N* hidden node as well as trigger subordinate *g(x)* could stand as shown by way of (3-1) [15]:

$$\sum_{i=1}^{N} \beta_i g\left(\mathbf{x}_j\right) = \sum_{i=1}^{N} \beta_i g\left(\mathbf{w_i} . \mathbf{x}_j + b_i\right) = \mathbf{0}_j, \quad j = 1, \cdots, M \tag{3-1}$$

W_i: Is the weighted vector, connects the *i^th^* hidden node to the input nodes.

β_i: Is the weighted vector, that connects the *i^th^* hidden node to the output nodes

b_i: Is the threshold of the *i^th^* hidden node.

$w_i . x_j$: The inner product of w_i and x_j.

$g(x)$: The trigger function.

In (3-1), $\mathbf{w}_i = \left[w_{i_1}, w_{i_2}, \cdots, w_{i_n}\right]^T$ is the weighted vector, that connects the i^{th} hidden node to the input nodes, $\boldsymbol{\beta}_i = \left[\beta_{i_1}, \beta_{i_2}, \cdots, \beta_{i_m}\right]^T$ is the weighted vector, that connects the i^{th} hidden node to the output nodes, and the b_i is the threshold of the i^{th} hidden node.

The standard SLFN can approximate *N* hidden nodes in addition to the trigger function *g(x),* for *M* sample with zero faults, that is $\sum_{i=1}^{N} \left\| \mathbf{o_j} - \mathbf{t}_j \right\| = \mathbf{0}$, in this case, the formula (3-1) is converted to equation (3-2) [15].

$$\sum_{i=1}^{N} \beta_i g\left(\mathbf{w_i} . \mathbf{x}_j + b_i\right) = \mathbf{t}_j, \quad j = 1, \cdots, M \tag{3-2}$$

The M equation in (3-2), can be stated as $H \cdot \beta = T$ (3-3) [15]:

$$H\left(\mathbf{w_1},\cdots,\mathbf{w_N},b_1,\cdots,b_N,\mathbf{x_1},\cdots,\mathbf{x}_M\right)=\begin{bmatrix} g\left(\mathbf{w_1}.\mathbf{x}_1+b_1\right) & \cdots & g\left(\mathbf{w_N}.\mathbf{x}_1+b_N\right) \\ \vdots & \ddots & \vdots \\ g\left(\mathbf{w_1}.\mathbf{x}_M+b_1\right) & \cdots & g\left(\mathbf{w_N}.\mathbf{x}_M+b_N\right)\end{bmatrix}_{M\times N} \quad (3\text{-}3)$$

Where [15]:

$$\beta = \begin{bmatrix}\beta_1^T \\ \vdots \\ \beta_N^T\end{bmatrix}_{N\times m} \quad (4)$$

$$\mathrm{T} = \begin{bmatrix}t_1^T \\ \vdots \\ t_M^T\end{bmatrix}_{M\times m} \quad (5)$$

H: Is the output matrix of the hidden layer of neural network [20].

T: $H \cdot \beta = T$

H Here is the output matrix of the hidden layer of the neural network. The i^{th} column of the *H* is the same as the i^{th} output hidden node, each of its rows corresponding to the output with the vertices $x_1, x_2, \ldots, x_M$.

For a group of sample inputs (x_i, t_i), $x_i \in R^n$, $t_i \in R^m$, the trigger function *g(x)*, and with a network structure containing *N* hidden nodes, the ELM algorithm is expressed as follows [20]:

1. The random production of the input weights W_i and b_i thresholds, $i = 1, 2, \ldots, N$.
2. Compute the output-hidden matrix.
3. Calculation of output weights β, $\beta = H^t T$ (here *t* means inverse matrix of *H*).

Once the number of examples are significant, there occurs high quantity of measurement at the employment of matrices increases of input load matrices w_i that arbitrarily produces in addition to samples matrix x_i. Now to shorten the measurements of the network, in such operation, consistently merely one row of w_i

was produced in addition multiple by example matrices. Later i times measurement, the $w_i.x_j$ could similarly be obtained [20].

The state formula as well as the observed formula, for a discrete linear random system, discussed here can be expressed as (3-4) and (3-5) formulas [15].

$$X_k = F_k X_{k-1} + B_k U_k + W_k \tag{3-4}$$

$$\mathbf{Z}_k = \mathbf{H}_k \mathbf{X}_k + V_k \tag{3-5}$$

X_k: Is the *n*-dimensional case conveyor of the scheme in time *k.*

Z_k: Is the *m*-dimensional observed conveyor in time *k.*

W_k: Is independent white noise with zero quantity, which means the covariance matrices Q_k and R_k, respectively [20].

V_k: Is independent white noise with zero quantity, which means the covariance matrices Q_k and R_k, respectively.

$A_{k,k-1}$: Is the transition matrix between *k-1* and *k* times, that has $n \times n$ dimensions.

H_k: Is the measurement matrix of the $m \times n$ dimension at time *k.*

Aimed at these collections of model formulas, the most significant approximation of the state vector X_k could be specified through the Kalman filter [20]:

$$X_k = F_k X_{k-1} + B_k U_k + W_k \tag{6}$$

$$X'_{k,k} = X'_{k,k-1} + K_k y_k \tag{7}$$

$$y_k = Z_k - H_k X'_{k,k-1} \tag{8}$$

$$K_k = P_{k,k-1} H_k^T (H_k P_{k,k-1} H_k^T + R_k)^{-1} \tag{9}$$

$$S_k = H_k P_{k,k-1} H_k^T + R_k \tag{10}$$

$$P_{k,k-1} = A_{k,k-1}P_{k,k-1}A_{k,k-1}^{T} + Q_{k-1} \tag{11}$$

$$P_{k,k} = (I - K_k H_k)P_{k,k-1} \tag{12}$$

Describe the origination as the calculating remainder (8) that the applied mean in addition to covariance are correspondingly explained through equation (13) as well as (14). Equation (10) was the model covariance matric of computing remainder. Through taking a suitable amount l, constant arithmetical outcomes can be attained. l is normally explained through practice and obtains the amount in the range of 30 to 50 [20].

$$P_y = \frac{1}{l}\sum_{j=k-l}^{k} y_j \tag{13}$$

$$S_y = \frac{1}{l}\sum_{j=k-l+1}^{K} y_j\, y_j^T \tag{14}$$

In (3-4) and (3-5) formulas, X_k is the *n*-dimensional case conveyor of the scheme in time *k*. $Z_k$was the *m*-dimensional observed conveyor in time *k*. W_k plus V_k are self-determining white noise by zero quantity, which means the covariance matrices Q_k and R_k, respectively. $A_{k,k-1}$ is the transition matrix between *k-1* and *k* times, that has $n \times n$ dimensions. H_k was dimension matric of the $m \times n$ dimension at time *k*. For the equations stated here, the best estimate of the X_k vector, is obtained by the Kalman filter [15].

In this proposed method, VAR is calculated as follows:

$$VAR = S_k - S_y \tag{3-15}$$

Once the statistical features of the noise are precise, the accuracy of the calculation must be including white noise that means zero quantity, in which case the VAR value should be close to zero. Big VAR means poor tracking and low performance. In the tracking system with the use of the Kalman filter, the Noise Covariance (Q_k), and the observed Noise Covariance (R_k), are required to be known and usually determined by the designer themselves [20].

In this proposed method, $R_k = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$ is considered. After the theoretical experiments and calculations, it was found that when a is very large, the tracking effect is very weak, and the target tracking object has many differences in location. When $a$very small, stable tracking is not very stable, and the result is traceable in an interval around the moving target [20].

Therefore, in this proposed method, a multiplying $r_k$vector in R_k is used that is adaptable according to the conditions. When the Kalman filter prediction result is less than the actual location of target, r_k will take a less value to reduce the R_k value, and in the opposite case, r_k is increased to increase $R_k$value, and the predicted value with use of Kalman filter is much closer to the target location. Therefore, R_k can be set by r_k, which causes the VAR to constantly have an amount close to zero, resulting in increased traceability. If VAR is as input and r_k is the ELM output then the neural network is trained regularly, in which case a fairly accurate network model can be found that can establish a relation between input and output, while there is no need to know the object movement model. In the proposed method at each step, the VAR is computed and placed in the ELM to obtain r_k and through which R_k is controlled.

Therefore, in this proposed method, a multiplying r_k vector in R_k is used that is adaptable according to the conditions. When the Kalman filter prediction result is less than the actual location of target, r_k will take a less value to reduce the R_k value, and in the opposite case, r_k is increased to increase R_k value, and the predicted value with use of Kalman filter is much closer to the target location. Therefore, R_k can be set by r_k, which causes the VAR to constantly have an amount close to zero, resulting in increased traceability. If VAR is as input and r_k is the ELM output then the neural network is trained regularly, in which case a fairly accurate network model can be found that can establish a relation between input and output, while there is no need to know the object movement model. In the proposed method at each step, the VAR is computed and placed in the ELM to obtain r_k and through which R_k is controlled.

3.2.1 Machine Learning Algorithm

ELM neural network is a three-layer feed-forward network consisting of the input layer, the hidden layer, and the output layer, as shown in Fig. 3.2. Mapping of the input layer to the hidden layer is a non-linear transformation, while mapping of the hidden layer to the output layer is linear [45].

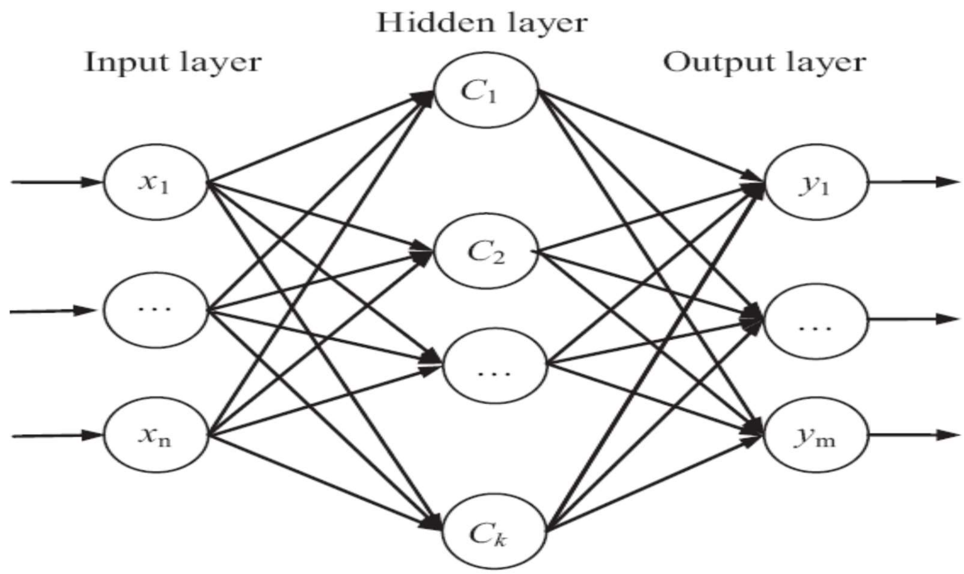


Figure 3.2: Network structure of ELM neural network [45].

The hidden layer is composed of a series of radial basis functions. The Gaussian function is a standard radial basis function. In this thesis, a target tracking prediction algorithm based on ELM neural network is composed of the following steps:

1. input samples
2. determine the network structure
3. normalize the data of the input layer
4. divide the input data into the training sample and the testing sample, and train the network with training samples, test the accuracy of the system with testing samples
5. build the prediction model of a target node by the trained ELM neural network, which can predict the coordination of a target node at next time
6. Select tracking nodes to track a target node according to the coordination of a target node, for example, select the tracking node closest to the target node as the tracking node at next time [45].

3.2.2 Executive Procedure of the Proposed Algorithm

Initially, slice target as of contextual through image processing besides obtaining the midpoint position of the target in the pictures that were named by way of the real place. After that reset Kalman filter, besides, predict the target's location position along with primed norms or the target's position predicted second earlier, that was name as the expected coordinate. Deduct the estimated coordinate as of the real coordinate and obtain their variance that was named the innovation. Each *VAR* that originates as of changes relates to an adaptive parameter. The *VAR* and r_k that were drawn from the trials were recognized as examples that were devote to Extreme Machine Learning aimed at teaching. Later teaching definite network exemplary would be created. Once target shifts, compute *VAR* as stated by *(15)* every second as well as put it into the taught ELM model to attain the regulating parameter.

3.2.3 Target Tracking System in WSN with the use of ELM based Kalman Filter Algorithm

The algorithm of regulating the Kalman filter gain using ELM could progress the tracking worth in WSN tracking system. In my thesis, I apply one camera to attain target images, as well as extract the target criteria's through image processing in addition to investigating technique. Lastly, we can acquire the target location on the screen over the plotting model from target criterions to the target location. The target tracking is to apply prediction norms as well as observation norms of Kalman filter for approximating target midpoint. In contiguous two picture plates of cinematic order, Kalman filter chasing procedure controls objective mammon of the following frame along with expectation formula besides all at once pursuit objective through applying characteristic indicator. Exploration region of characteristic indicator was resolute through the objective plotting location. The extent of the pursuit region was resolute through procedure smash covariance matric:

$$P_{k.k-1} = A_{k.k-1}P_{k-1.k-1}A_{k,k-1}^{T} + Q_{k-1}$$

Eventually, fuse the detection norms as well as prediction norms to create a final approximation state of Kalman filter. However, occasionally characteristic detector could not catch the goal at exploration region of the following case since the expectation formula of Kalman filter was permanently explained by way of the linear model. Once the target course alterations significantly, it cannot predict the target position very accurate. The exploration region of the target is resolute through the process noise covariance P and prediction formulas, thus it could miss the target detection, finally causing to the tracking miscarriage. So as to unravel such issue, I propose the technique of applying extreme learning machine to regulate scheme smash covariance Q as well as seen smash covariance matric R, then at that time regulate the Kalman gain factor. Since we could realize Kalman filter gain factor as observation as well as expectation norms weighting parameter, thus it could applied to control following second's part of discovery norm Z_k as well as expectation norm $HX'_{k,k-1}$.

One more clarification is that as the measurement noise covariance R is near zero, computed parameter Z_k is of larger weight, in addition to $HX'_{k,k-1}$ is of lesser weight. Alternatively, as covariance Q is near zero, measured parameter Z_k is of lesser weight, and $HX'_{k,k-1}$ is of larger weight. Consequently, regulating the Kalman filter gain factor over ELM could have an improved approximation for the target location [20]. The process error covariance matrix P regulates the target exploration region of characteristic indicator, which was objective finding is perceived at exploration region rather than the full picture with the purpose of real-time chasing was eventually developed. Seen smash covariance R was largely to stimulus Kalman gain factor. Either to regulate P or R, ultimate aim was regulating the Kalman achievement factor for precisely tracking target.

3.2.4 Fast Learning

Fast or online education, is a powerful learning machine that was unlike as of offline education as well as batch education. In offline training, typically the customary neural networks are used, so in these types of networks, the parameters are set when the system trains with many examples. In this situation, there are problems that, as the system trains, there may still be cases where the system has not seen any training regarding them. Therefore, learning in offline mode is uncertainly. In online learning mode, the system is trained as soon as the samples are entering and the network parameters are set up in real-time consistent with the network result. Online learning can update the network as soon as new data are arriving, and retrieves feedback from the system and updates the network parameters in real time. Online learning that is used in the proposed method does not require much memory to store sample data, which makes the proposed method more flexible and more comfortable to execute.

In the proposed method, the samples are not high and the system execution time is not high, so in the proposed way, the new samples imported into the system will be added to the primary samples as teaching examples. In this system, once environmental alterations occur, meant here the target displacement, the accuracy of the calculation of the new samples diverges from the accuracy of the prediction of the primary samples, and thus the network is couldn't to create the proper adaptability criteria. Putting the calculation accuracy by way of original example added to the primary examples in the proposed method that uses ELM requires a solution. So far, the proposed method was able to use offline learning, but this is a solution that makes the proposed method use online learning or quick learning.

To put online learning in the proposed method, four steps are needed:

Network training using very few examples

Setting the accuracy of the present calculation in the training network model and obtaining an adaptability factor, and then setting the R_k relating to the Kalman filter by the corresponding factor.

Embed the accuracy of the present calculation and factor as an example in the network and re-training the system.

Repeat steps 2 to 4 as long as online learning is set.

The benefits of online learning include the less quantitative knowledge that is needed to link the target shifts as well as the Kalman filter-chasing scheme, and in this case, the exact model of the system would not require. Regulating the observed smash covariance matric of the Kalman filter through extreme learning machine online education will increase the ability of tracking of the Kalman filter effectively.

3.2.5 Routing of the Proposed Method

In the proposed method, we use LEACH routing algorithm to collect information from different nodes in the tracking field, because it has a very low energy consumption algorithm and relatively high reliability, which is very important in the proposed method. If the reliability is small and the coverage is not properly executed, tracking cannot be done correctly and the target will be lost, so in tracking topics, there should be routing trustworthy algorithms.

Chapter 4

RESULTS AND EVALUATION

4.1 Introduction

In this section, we intend to compare our proposed method with similar methods and find out its advantages and disadvantages. Here are some parameters, which are used for comparison that are described below.

4.2 Root Mean Square Deviation

Root Mean Square Error or Root Mean Square Deviation (RMSD) is the difference between the predicted value of the model or the statistical estimator and the actual value. RMSD is a useful tool for comparing prediction errors with a data set and is not applicable for analyzing multiple datasets.

In the root mean square error of the of a statistical estimator, θ is defined as the squared square root of the mean square error, according to the predicted parameter θ [15]:

$$\theta_1 = \begin{bmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{1,n} \end{bmatrix} \quad and \quad \theta_2 = \begin{bmatrix} x_{2,1} \\ x_{2,2} \\ \vdots \\ x_{2,n} \end{bmatrix} \quad (1)$$

The equation is as follows [16]:

$$\mathbf{RMSD}(\theta_1,\theta_2) = \sqrt{\mathbf{MSE}(\theta_1,\theta_2)} = \sqrt{\mathbf{E}\left((\theta_1,\theta_2)^2\right)} = \sqrt{\frac{\sum_{i=1}^{n}\left(x_{1,i} - x_{2,i}\right)^2}{n}} \quad (2)$$

Where:

$(x_{1,i} - x_{2,i})^2$: Squared differences

n: Sample size

4.3 Simulation Results of Using Image processing in Estimating Initial Position of the Target

In this thesis, a WSN target-tracking scheme is applied for implementing extreme learning machine centered Kalman filter chasing. The dimension of the contextual picture was 320× 24, and video arrangement have 60 picture frames. Trials were done in MATLAB Software. Objective chasing scheme formula can be shown as [20]:

$$X_k = A_{k,k-1}X_{k-1} + W_k \tag{16}$$

Where:

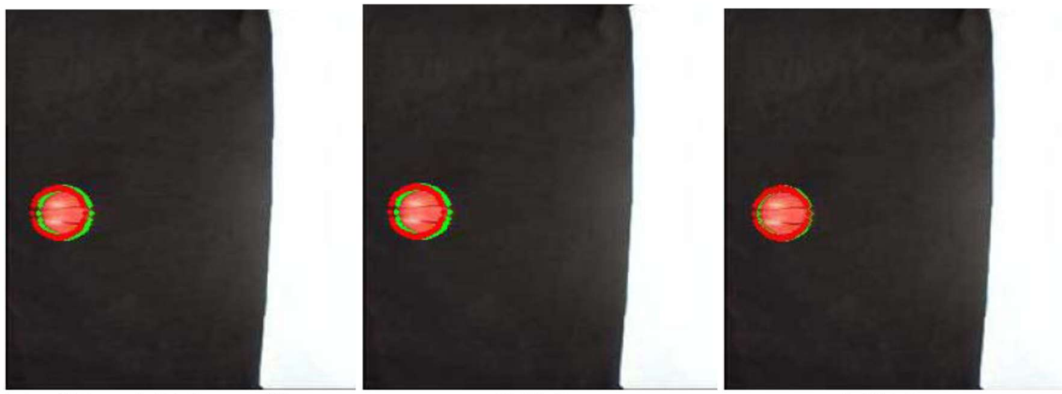
W_k: Is self-determining white noise by zero quantity.

$$\mathbf{A}_{k,k-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{X}_k = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}$$

$x_1(t), x_2(t)$ are correspondingly abscissa as well as ordinate of the target midpoint in picture in time t. Aim of the trial was acquiring total best Kalman filter approximation X_k derived from the observation value Z_k. The initial value of X_k is X_0.

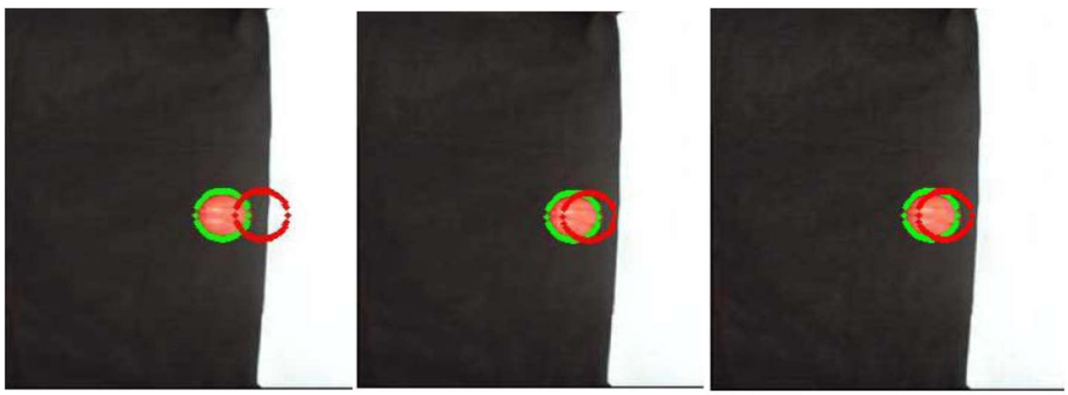
In this thesis, we use transformation techniques for attaining the exact location of the target, once a camera take a picture from an environment. We have the exact location of each sensor nodes (we give the exact location of each node as input to the system), and when a camera take a picture from the environment by use of the geometry laws we can calculate the exact location of the target, and also we can find out that the target is adjacent to which sensors and start tracking from those points. Because each target in the environment will be surrounded by at least three sensors, we can calculate the exact location of each target very accurately.

The tracking enactment of customary Kalman filter, SVM centered adaptive Kalman filter, as well as ELM, centered adaptive Kalman filter can be shown as [20]:



Traditional Kalman filter SVM ELM

B



Traditional Kalman filter SVM ELM

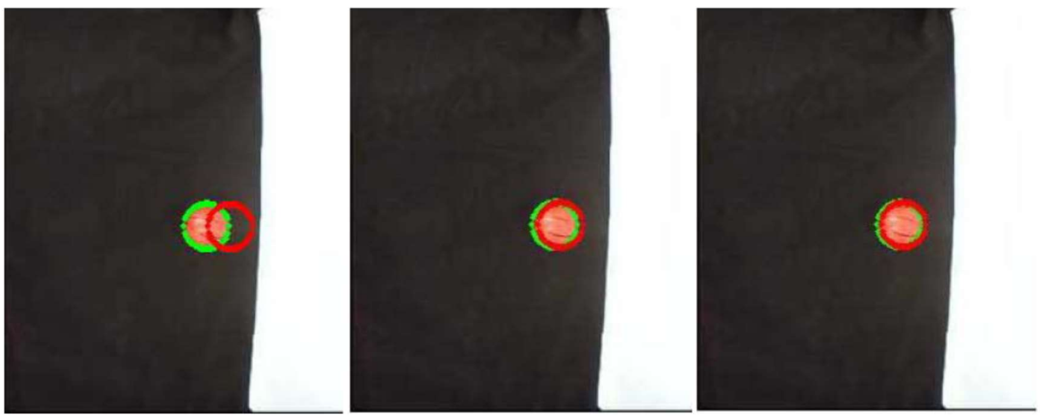
C

Traditional Kalman filter SVM ELM

D

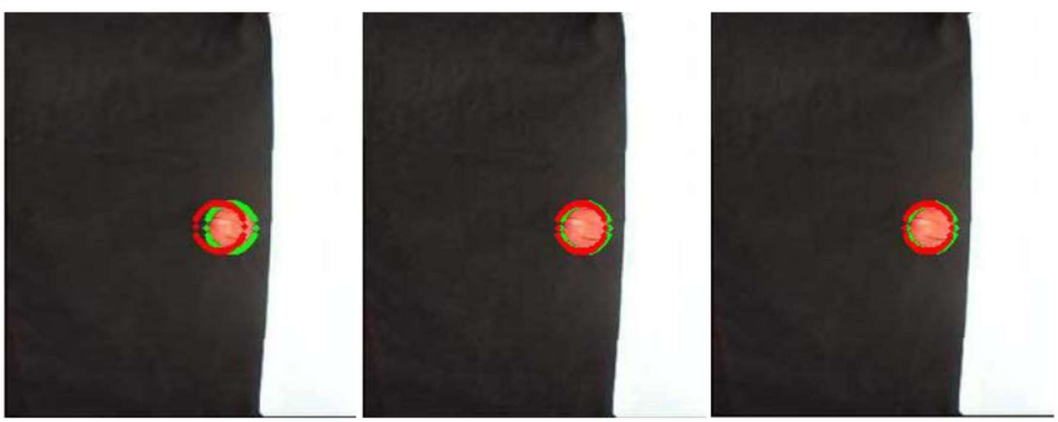
Traditional Kalman filter SVM ELM

E



Traditional Kalman filter SVM ELM

F



Traditional Kalman filter SVM ELM

Figure 4.1: The tracking consequences of customary Kalman filter, SVM based adaptive Kalman filter and ELM based adaptive Kalman filter. (A) is by using of 10 frames, (B) is by using of 16 frame, (C) is by using of 20 frame, (D) is by using of 25 frame, (E) is by using of 30 frame, (F) is by using of 35 frame [20].

The simulation outcomes demonstrate that the tracking enactment of adaptive Kalman filter was clearly enhanced than that of customary Kalman filter. Also, the adaptive Kalman filter tracking is further swift and stable [20]. Furthermore, using unrestored ELM as well as customary SVM to teach the same set of models that comprises 40 groups of information takes 0.0156s and 0.2429s correspondingly where ELM's hidden layer has 20 neurons. Matching the time duration between ELM and SVM, ELM training is quicker than that of SVM. Once the quantity of samples is very large, the difficulty of SVM procedure is quickly augmented [20], even though ELM is much appropriate for training a significant amount of samples.

4.4 Validation of Test Results

In this thesis, an SLFN is put forward to improve forecast accuracy with the help of its powerful nonlinear mapping capability. Different from traditional learning algorithms that all the parameters of the feedforward neural network (NN) need to be tuned to minimize the cost function, ELM theories claim that the hidden node learning parameters can be randomly assigned independently and the network output weights can be analytically determined by solving a linear system using the least squares method [46]. The training phase can be efficiently completed without time-consuming learning iterations, and ELM can achieve an excellent generalization performance [46].

Under the data fusion scheme, the proposed method is used to predict the data for the next period. When the sensor node senses data of the next period, the sensor node compares the sensed data with the predicted data. If the expected error is under the threshold ε, the sensor node does not send the sensed data to the sink node, and the data transmission is canceled to save the energy, which is the goal of data fusion. At the same time, the sink node uses the same prediction mechanism as the sensor node to predict the data of the next period and then considers the expected data as the sensed data in the current period. On the contrary, the sensor node must transmit the sensed data to the sink node when the prediction error is above the threshold ε. Here ε is a turning parameter defined by those end-users. It is used to balance energy efficiency and data accuracy [46].

4.5 Evaluation of Proposed Method

4.5.1 First Test

In this assessment, the nodes are distributed in an environment of 100 by 100 Centimeter, and here the number of sensors would be considered 100. Locations of the sensors are randomly distributed in the background, and we did not use a specific data file for the nodes' position, so that each time the code is executed, the result can be determined, and it is not necessary to change the location of nodes every time so that different effects can be obtained, in various experiments. Of course, there is also an option in the code that enables to reads the file by setting the value of that parameter in the program. In the following simulations, we show the name of our algorithm by MKF, which stands for MyKalmanFilter, and we compared our proposed algorithm with two EKF and UKF algorithms, which we discussed these two algorithms in the previous chapter. These two algorithms are one of the most prominent algorithms for the Kalman filter that we intend to demonstrate here that our proposed method is more optimum than these two algorithms, and has higher and better performance rates or similar to these algorithms.

In figure (4.2), you can see the location of nodes in a 100*100 environment:

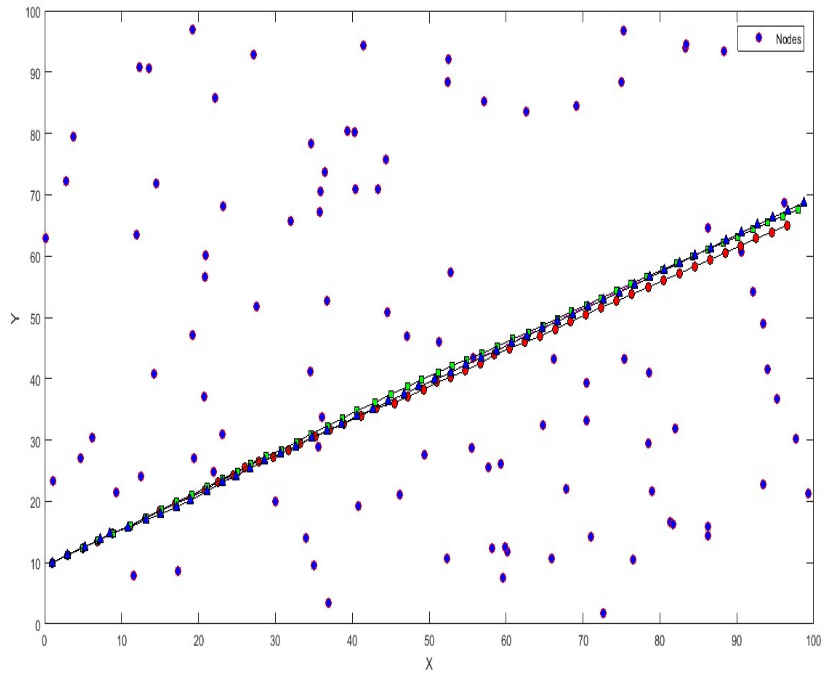


Figure 4.2: The location of nodes and tracking of target for three algorithms with 100 nodes.

As can be seen in Fig. 4.2, outputs of three algorithms are shown for objective chasing. Our procedure was blue, UKF procedure was green also the EKF algorithm is shown in red, in the image.

It can be seen from this picture that the outputs of the three algorithms are nearly identical aimed at objective chasing, as well as these three objective chasing procedures, track correctly. To better compare these three algorithms, it is better to compare them in terms of speed and the tracking time, as well as the RMS or Root Mean Square error; we did this, later in this chapter.

In Figure 4.3, we can see the output of the average square error of the three algorithms at different times.

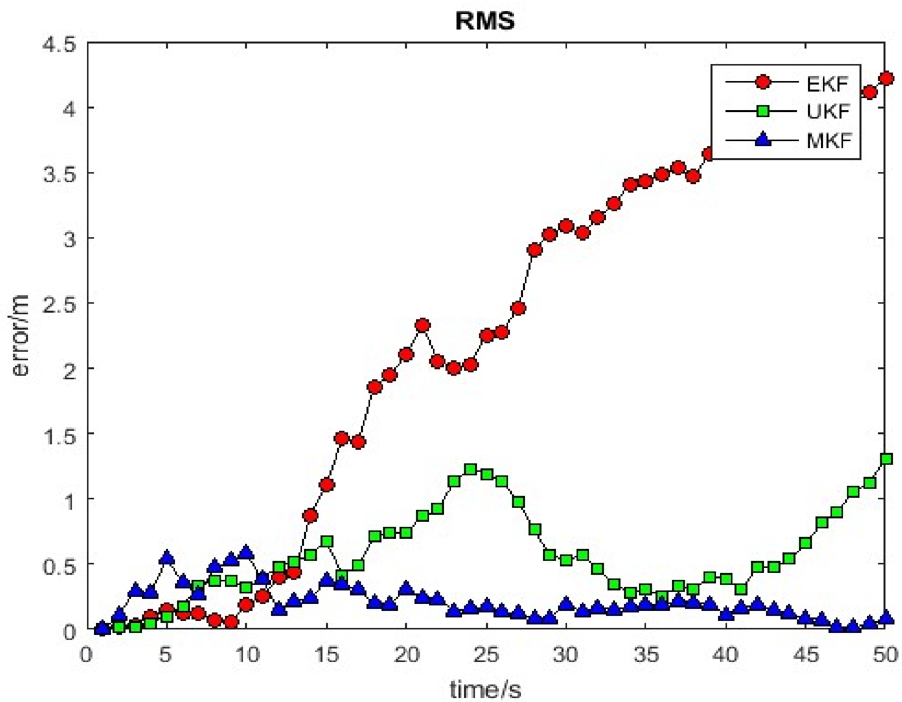


Figure 4.3: The output of average square error for the three algorithms and for 100 nodes.

According to Fig. 4.3, we can find that our algorithm has an average number of fewer errors at approximate times between 12 and 50. At first, our algorithm has a higher error than the other three algorithms, which is because of the learning machine that we used in the algorithm. The learning machine should initially gather information to help with the gained knowledge in the procedure. Here we can also see that our algorithm has a smaller RMS than the other two algorithms after approximation time of 12, and this amount will continue to be less. That means, our algorithm will become stronger over time, and whatever happens, it will get more information, and it is able to show better performance; while two other algorithms are worse with the continuation of the process, notably the EKF algorithm that can be seen at the beginning of the work between the approximate times of 5 to 12, it works best in comparison with all algorithms, but it has gone worse over time due to the fact that this algorithm approximates wrong from the wrong point, and because the next approximation results are obtained from the previous approximations, it is worsened by the continuation of its performance. The UKF has shown better performance than EKF, but its performance has been worse than our algorithm.

We continued to examine the algorithms step-by-step and to find and compare the time needed for each step. You can see this output in Figure 4.4.

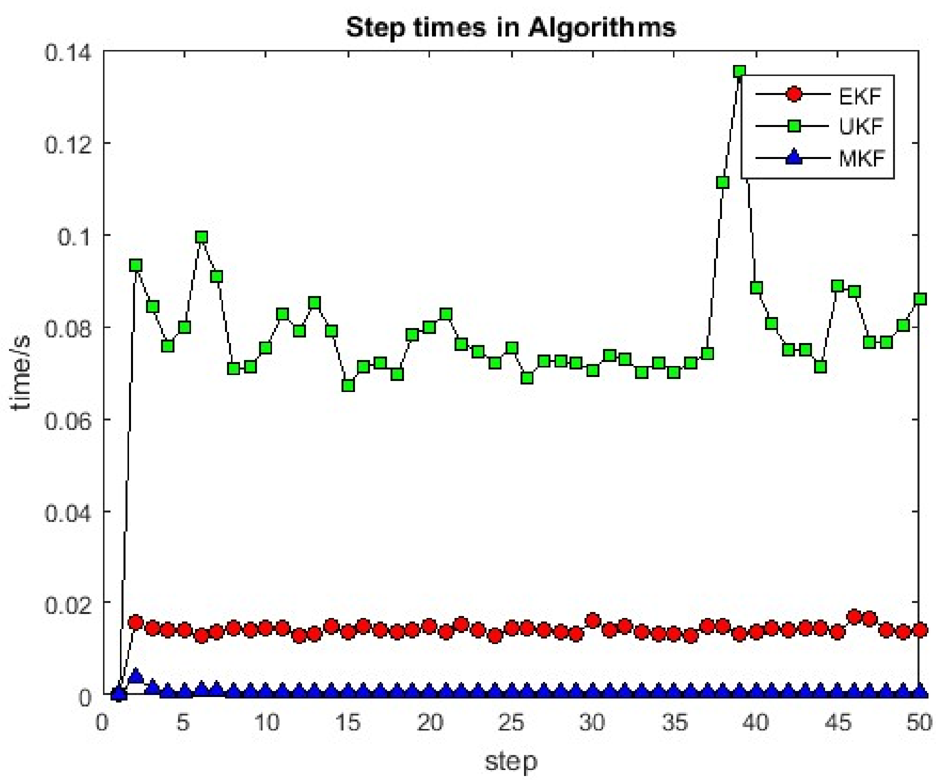


Figure 4.4: The needed time for the steps of different algorithms for 100 nodes.

It is permanent in figure (4.4) that this procedure performs tracking at minimal times, and it can be said that it is faster than the other two algorithms, that is, it can follow in the fastest possible time. It can be seen that considering that the computational overhead of the UKF is more significant than EKF, it takes more time to track in each step. One of these factors can be attributed to the fact that the UKF acts nonlinearly and EKF operates linearly to UKF, but it has more computational overhead in comparison with our algorithm. Therefore, our algorithm has the least computational overhead compared to these three algorithms.

In Figure 4.5, we can see the mean of RMS for the three comparison algorithms. Regarding this output, one can find that the algorithm has an average RMS of less than the EKF and UKF algorithms. Therefore, it can be concluded that this procedure does improve in comparison to the additional two methods in addition to having improved recital as well as swiftness.

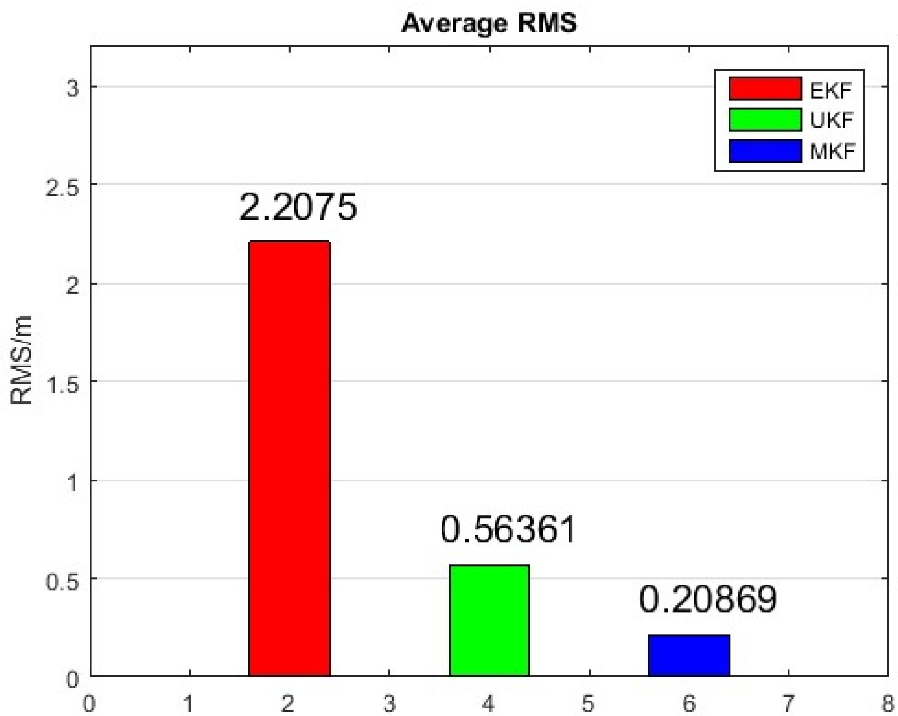


Figure 4.5: The average RMS for the proposed algorithm and the two UKF and EKF algorithms for the 100 nodes.

4.5.2 Second Test

We change the test conditions slightly. In this assessment, nodes are distributed in an environment of 100 Centimeter heights by 100 Centimeter widths, and here we consider the number of nodes is four. Different outputs are shown from Figure 4.6 to Figure 4.9.

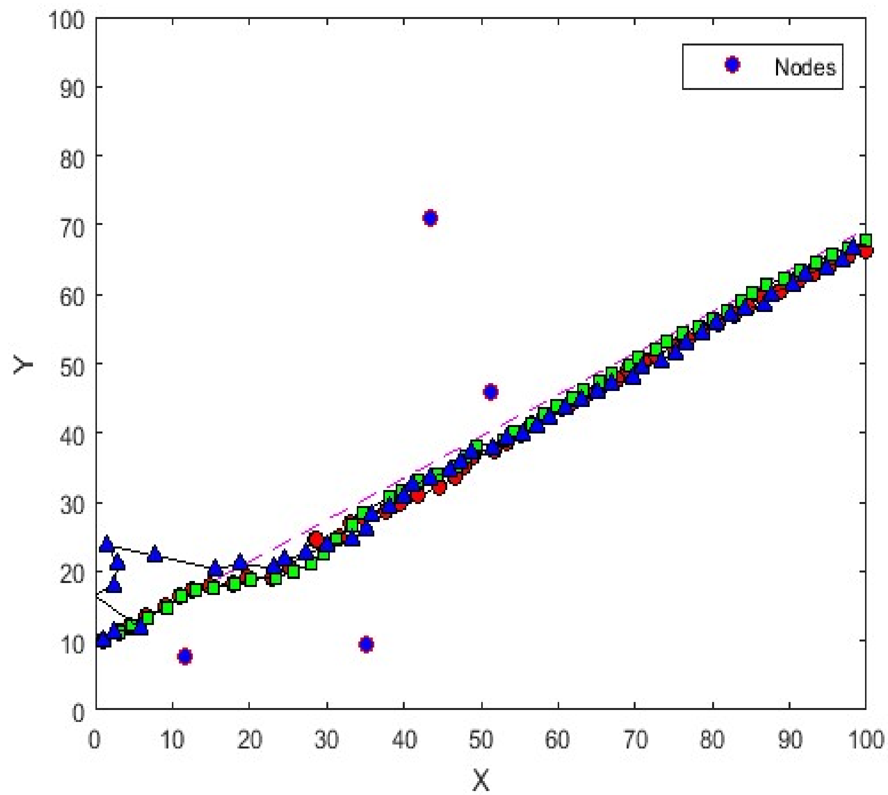


Figure 4.6: The nodes locations and tracking of targets for three algorithms with 4 nodes.

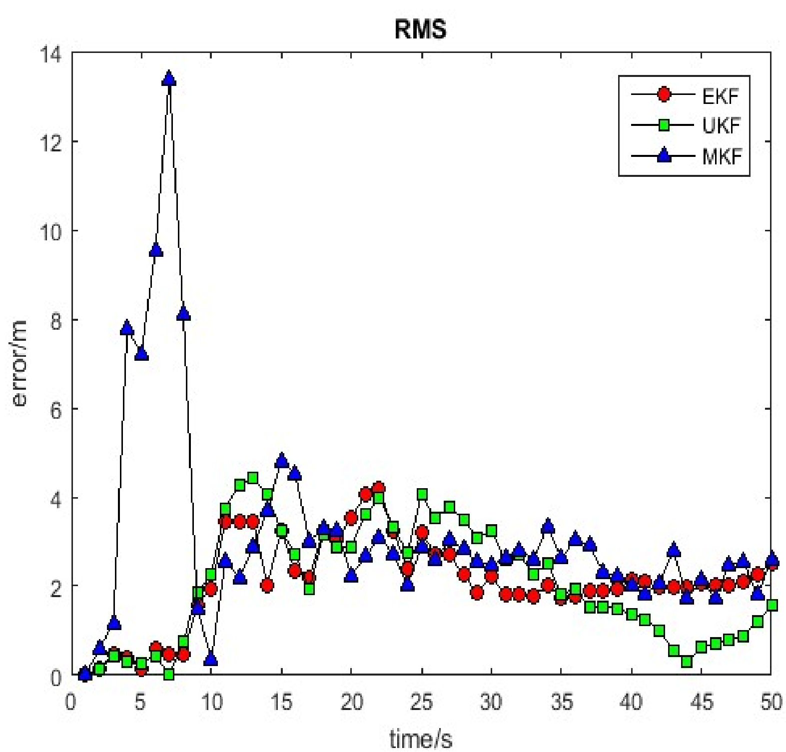


Figure 4.7: The output of mean square error of three algorithms for 4 nodes.

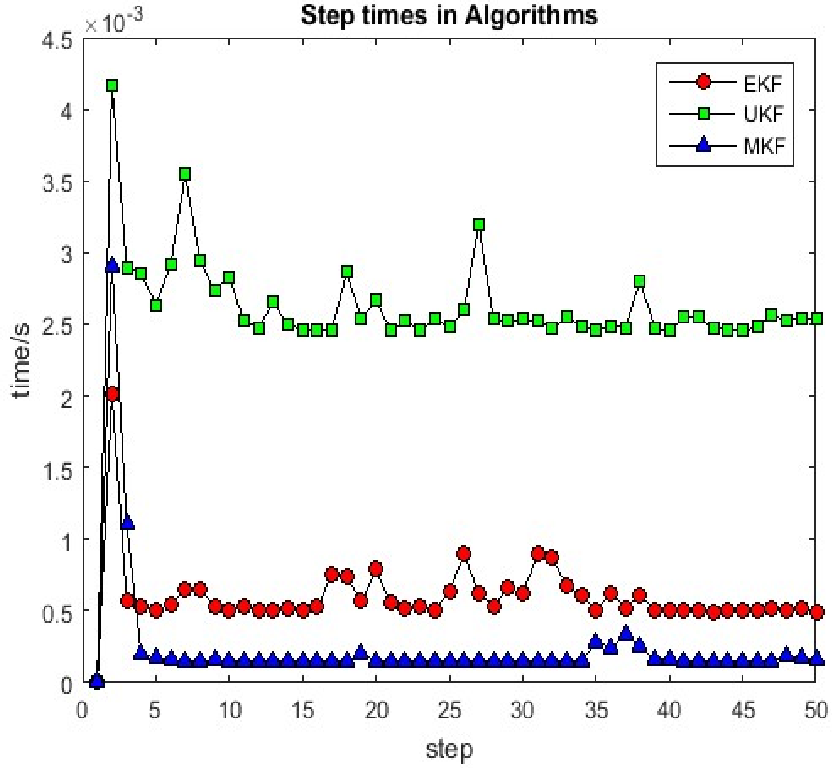


Figure 4.8: The needed time for each step of different algorithms for 4 nodes.

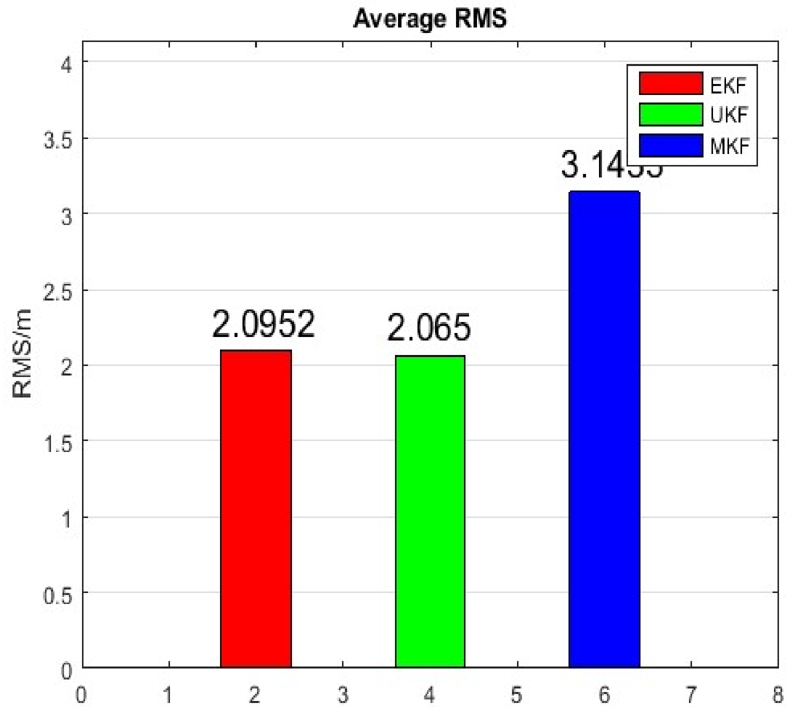


Figure 4.9: The RMS mean for the proposed algorithm and UKF and EKF algorithms, with 4 nodes.

4.5.3 Third Test

We change the test conditions slightly. In this assessment, nodes are distributed in an environment of 100 Centimeter heights by 100 Centimeter widths, and here we consider the number of nodes is 500. Different outputs are shown from Figure 4.10 to Figure 4.13.

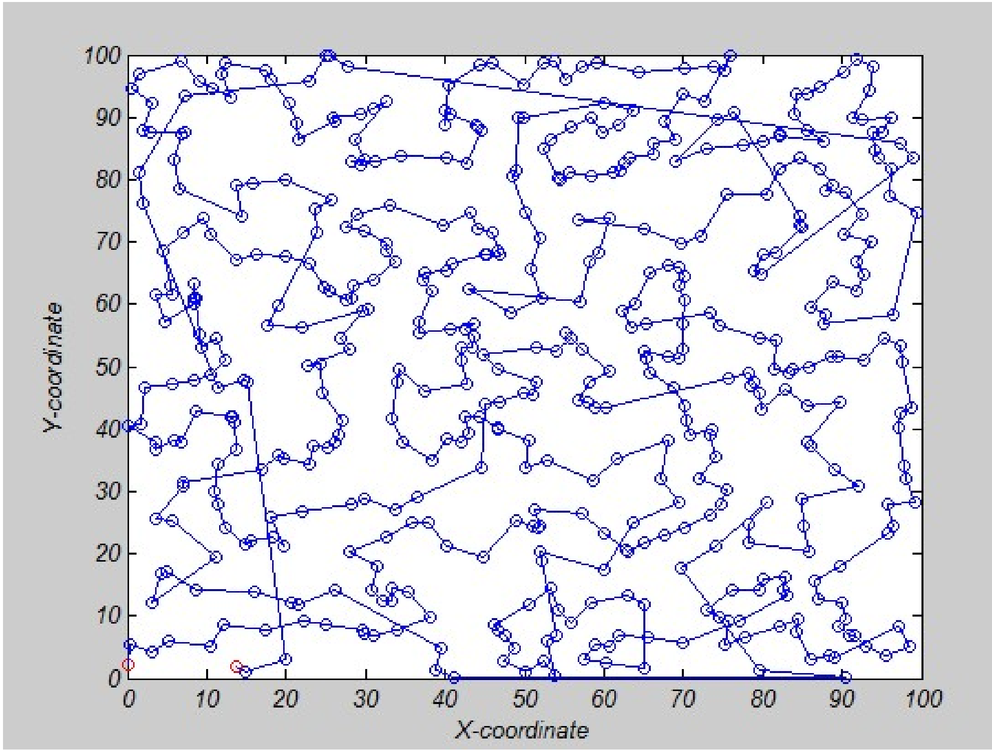


Figure 4.10: The nodes locations and tracking of targets for three algorithms with 500 nodes.

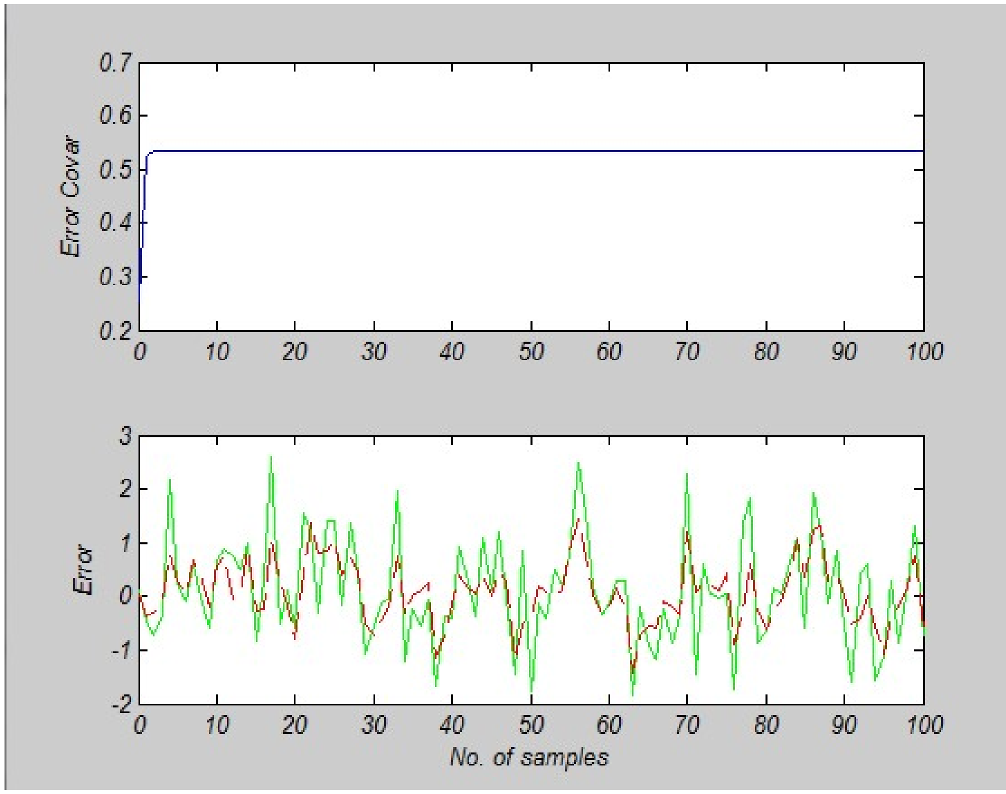


Figure 4.11: The output of mean square error of three algorithms for 500 nodes. The mean square error of MKF were shown in blue, UKF in green and EKF is in red.

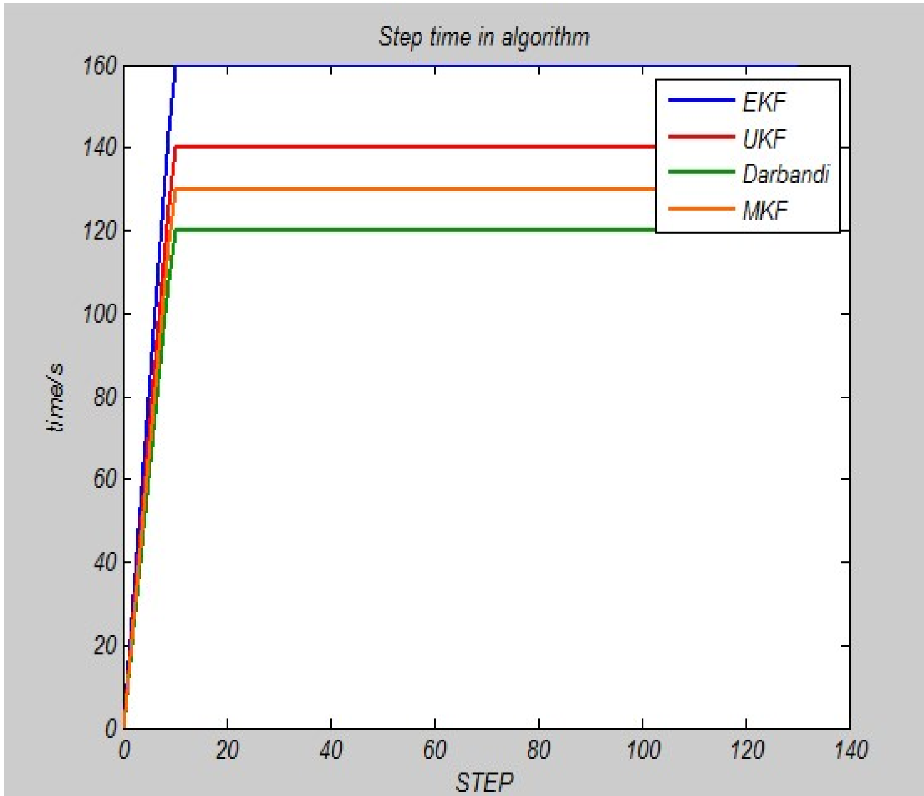


Figure 4.12: The needed time for each step of different algorithms for 500 nodes.

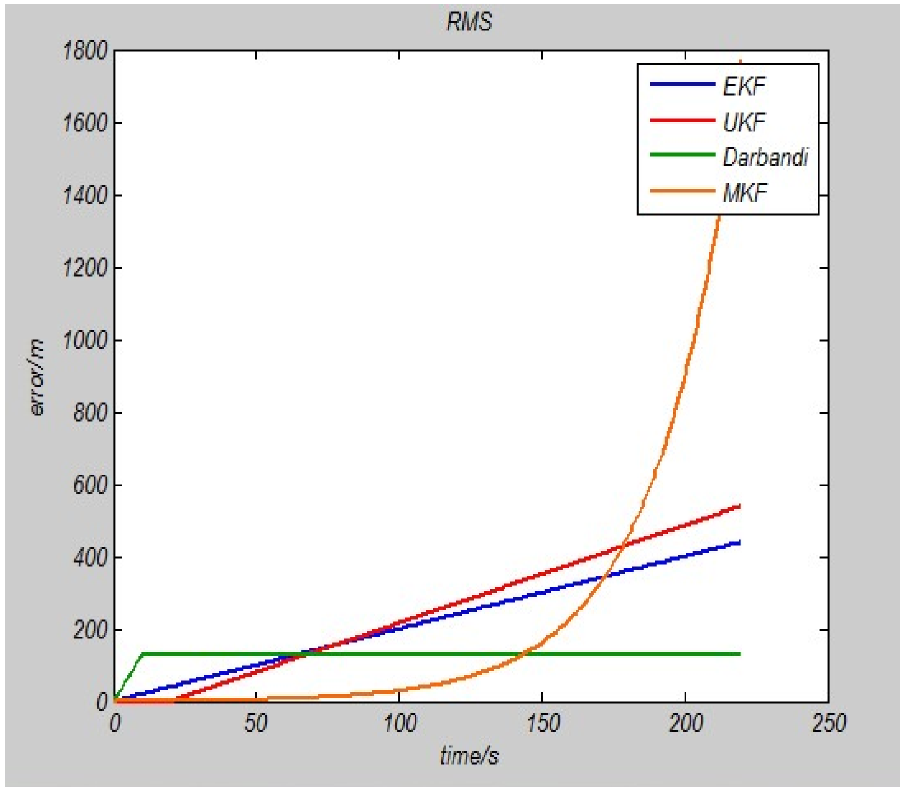


Figure 4.13: The RMS mean for the proposed algorithm and UKF and EKF algorithms, with 500 nodes.

4.5.4 Fourth Test

We change the test conditions slightly. In this assessment, nodes are distributed in an environment of 100 Centimeter heights by 100 Centimeter widths, and here we consider the number of nodes is 1000. Different outputs are shown from Figure 4.14 to Figure 4.17.

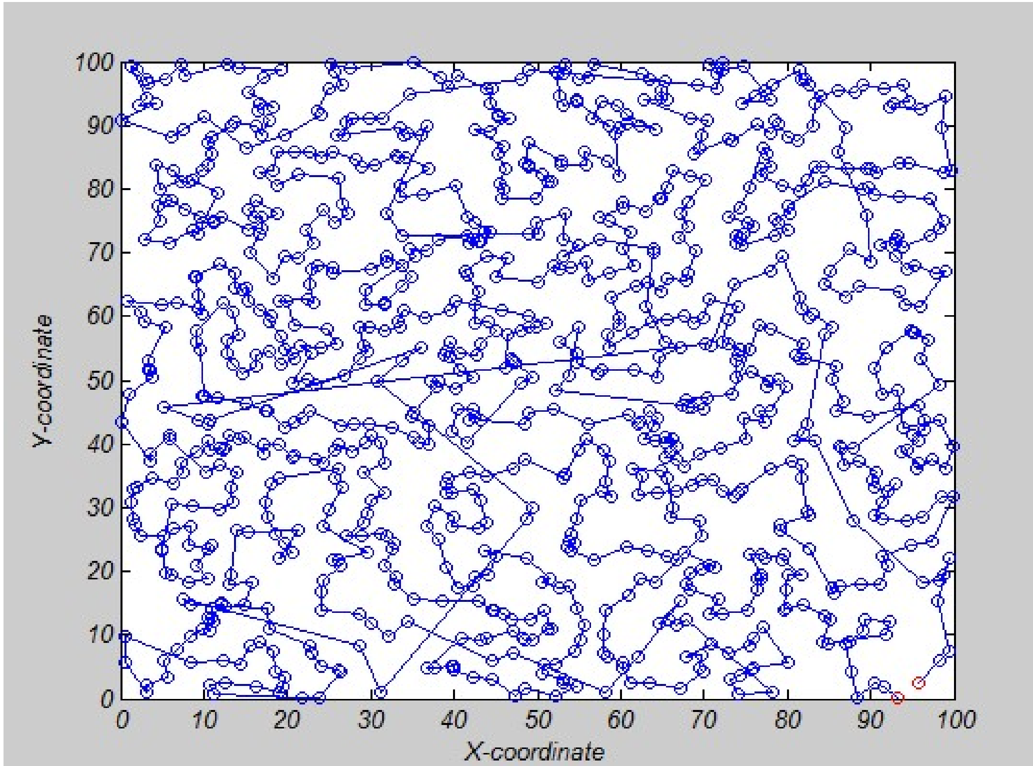


Figure 4.14: The nodes locations and tracking of targets for three algorithms with 1000 nodes.

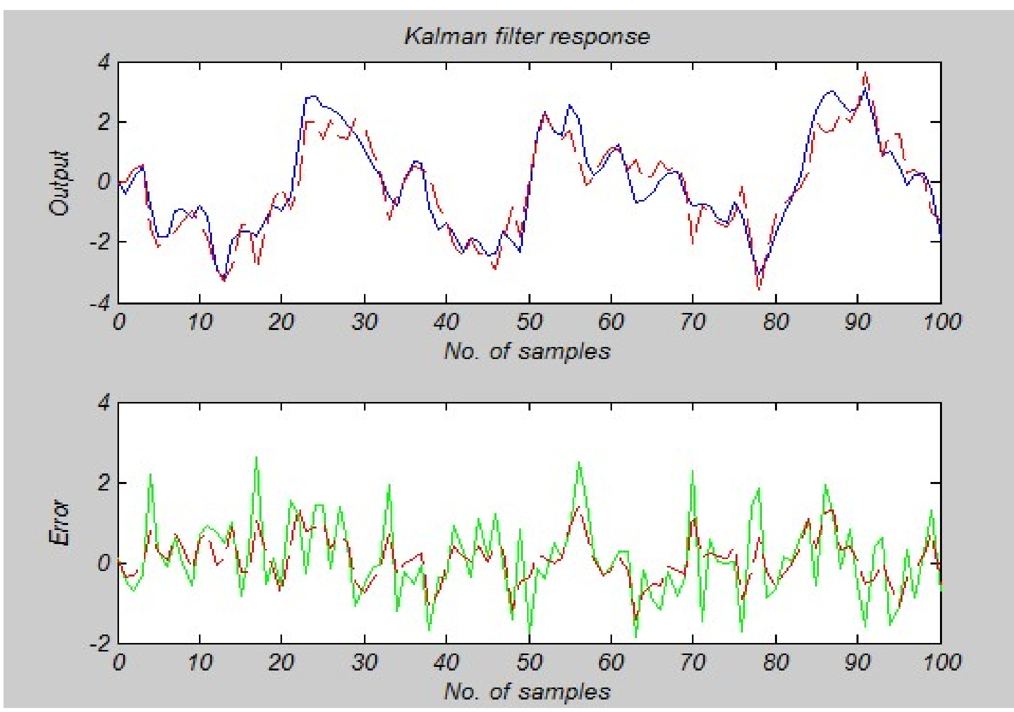


Figure 4.15: The output of mean square error of three algorithms for 1000 nodes. The mean square error of MKF were shown in blue, UKF in green and EKF is in red.

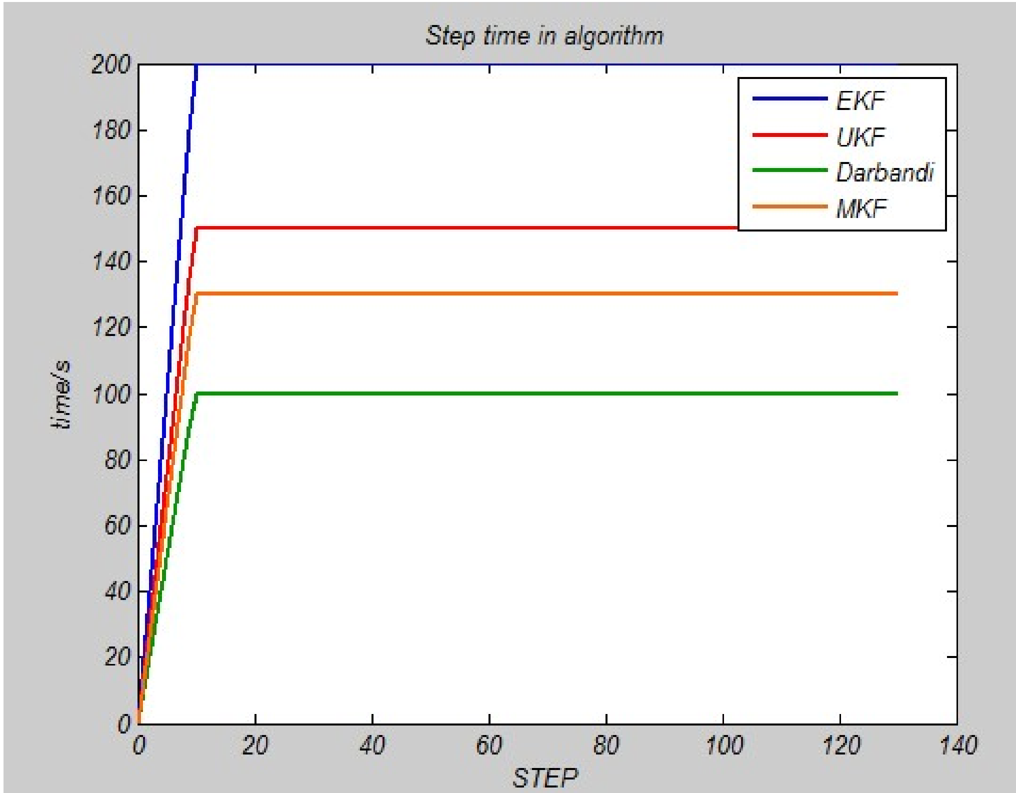


Figure 4.16: The needed time for each step of different algorithms for 1000 nodes.

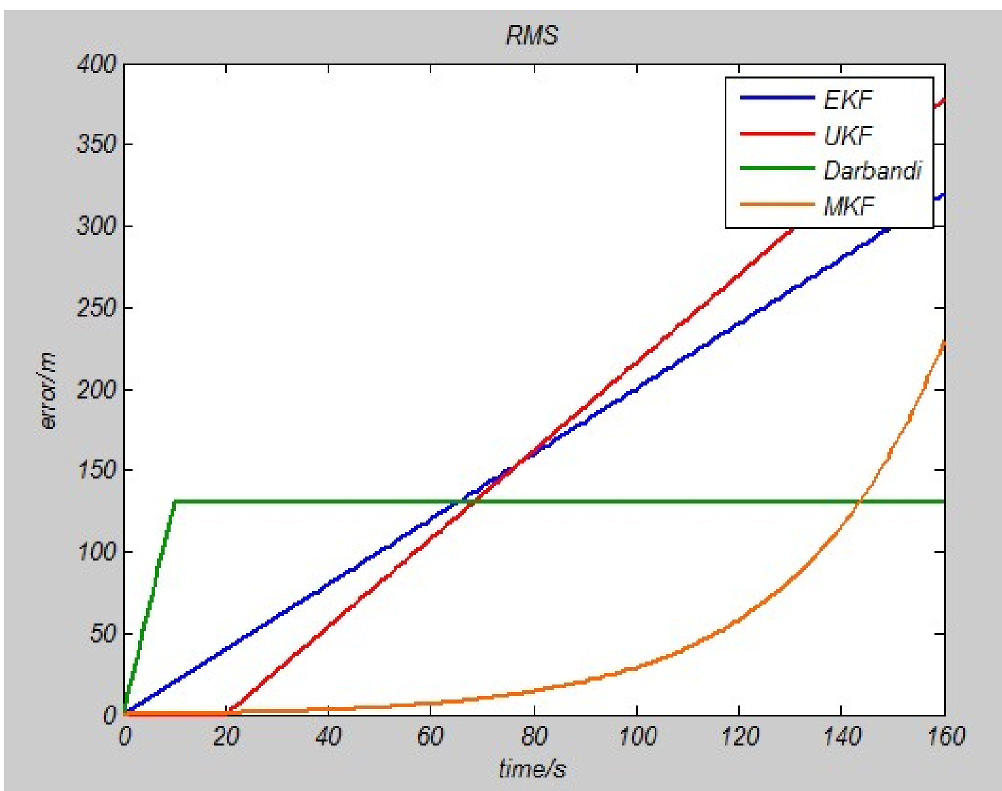


Figure 4.17: The RMS mean for the proposed algorithm and UKF and EKF algorithms, with 1000 nodes.

It is evident from the simulations that, our algorithm for the low number of nodes has a weaker performance than the other algorithms, and only the time overhead in our algorithm, is better than different algorithms. In general, we can say that our algorithm is not optimum once the number of sensors would be minimal; and while the more sensors existed, the more optimum our algorithm works and the benefits of it are perfectly demonstrated. It can be seen that this result is undeniable, because the algorithm is based on the learning machine, and when the more nodes exist, the more data there is to learn, and hence the algorithm's performance will improve

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

During the experiments on the proposed algorithm, we found that the proposed algorithm has an excellent performance if there are ten nodes. In the second experiment, we considered a network, which has only four nodes, while in the real world, the number of wireless sensor nodes is much higher than that, and in different applications, there are different amounts of such sensors. However, the planned procedure would be capable of displaying relatively good performance at a low number of nodes; and very high speed and efficiency, as shown, in the high number of nodes. It was also observed that as the time used in the proposed algorithm is increased, its speed is still constant at the same level and gradually decreases its RMS, while popular algorithms such as EKF and UKF have been observed that their RMS goes up. Therefore, the proposed algorithm can be used in large-scale applications and is undoubtedly capable of performing well, as demonstrated during the experiments. In general, in the tracking of moving objects, the speed is the most critical factor for monitoring because it can also track moving targets at high speeds, which is what our algorithm has shown to be very powerful in this option, and can track targets as quickly as possible.

5.2 Suggestions for Future Work

The proposed algorithm outlined here has many benefits, but this algorithm can still be more extended. The first suggestion that can be made here for future work is the combination of the proposed algorithm with the EKF algorithm because our proposed algorithm is capable of showing very high performance for a large number of sensors, but for the lower amount of sensors, its performance was relatively weak. We can combine our proposed algorithm with the EKF to use the EKF procedure once the amount of sensors is reduced below a definite value, for example, N. Because some sensor nodes are able to charge themselves with a solar battery, they may have left the network during the work thereby sensors nodes are below N. Once charged they may suddenly return to the system; in this case, if the combination of our proposed algorithm with EKF is used the algorithm with the highest efficiency can be used.

Among other suggestions that can be made, is the combination of our algorithm with the UKF algorithm. We can find that our algorithm has a high RMS at the beginning, and after a little bit of the algorithm's passing, and gaining knowledge, it continues with a lower RMS. Given that it in tracking moving targets, the RMS plays an vital role, we can use the UKF algorithm together with the proposed algorithm as a hybrid algorithm that has a very low RMS and has a high speed and very high performance.

The third suggestion can be a combination of these three algorithms, in which the benefits of each one were also expressed in the same section, with the low number of nodes we can use the EKF, and once amount of sensors are relatively great but at start of the operation, we can use the UKF, and a little after the start, where the number of nodes is relatively high, use our proposed algorithm.

REFERENCES

[1] Xu, E., Ding, Z., & Dasgupta, S. (2013). Target tracking and mobile sensor navigation in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 12(1), 177–186.

[2] Bhuiyan, M., Wang, G., & Vasilakos, A. (2015) Local area prediction-based mobile target tracking in wireless sensor networks. *IEEE Transactions on Computers,* 64(7), 1968–1982.

[3] Li, M., Li, Z., & Vasilakos, A. V. (2013). A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues. *Proceedings of the IEEE*.

[4] Wang, X., Vasilakos, A. V., Chen, M., Liu, Y., & Kwon, T. (2012). A survey of green mobile networks: Opportunities and challenges. *Mobile Networks and Applications*, 17(1), 4–20.

[5] Chen, M., Gonzalez, S., Vasilakos, A., Cao, H., & Leung, V. (2011). Body area networks: A survey. *Mobile Networks and Applications*, 16(2), 171–193.

[6] Zeng, Y., Xiang, K., Li, D., & Vasilakos, A. V. (2013). Directional routing and scheduling for green vehicular delay tolerant networks. *Wireless Networks*, 19(2), 161–173.

[7] Boukerche, A., Oliveira, H., Nakamura, E., & Loureiro, A. F. (2008). Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer Communications*,31(12), 2838–2849.

[8] Gribben, J., & Boukerche, A. (2014). Location error estimation in wireless ad hoc networks. *Ad Hoc Networks*, 13, 504–515.

[9] M. Rhudy, R. Salguero, K. Holappa, "A Kalman Filtering Tutorial for Undergraduate Students", *International Journal of Computer Science & Engineering Survey*, Vol. 8, No. 1, Feb. 2017.

[10] Stoleru, R., Stankovic, J. A., & Son, S. H. (2007). Robust node localization for wireless sensor networks. *In Proceedings of the 4th workshop on embedded networked sensors*, Cork, Ireland.

[11] Saric, Z. M., Kukolj, D. D., & Teslic, N. D. (2010). Acoustic source localization in wireless sensor network. *Circuits Systems and Signal Processing*, 29(5), 837–856.

[12] Zhang, J., Yang, R., & Li, J. (2013). An enhanced DV-hop localization algorithm using RSSI. *International Journal of Future Generation Communication and Networking,* 6(6).

[13] Chen, H., Deng, K. P., & So, H. C. (2008). An improved DV-Hop localization algorithm with reduced node location error for wireless sensor networks. *IECE Transactions on Fundamentals*, E91-A (8), 2232–2236.

[14] G. Welch, G. Bishop, "An Introduction to the Kalman Filter", *University of North California, Chapel Hill*, 2001.

[15] Tony Lacey, "Tutorial: The Kalman Filter", *Chapter 11 of Online book for The Kalman Filter*.

[16] https://www.statisticshowto.datasciencecentral.com/rmse/

[17] Hamilton, J. (1994), *Time Series Analysis, Princeton University Press*. Chapter 13, 'The Kalman Filter'.

[18] Ishihara J.Y., M.H. Terra. (2006). Robust Kalman Filter for Descriptor Systems. *IEEE Transactions on Automatic Control*.

[19] Terra .Marco H., Joao P. Cerri. (2014). Optimal Robust Linear Quadratic Regulator for Systems Subject to Uncertainties. *IEEE Transactions on Automatic Control*. 2586–2591.

[20] Jian Nan Chi, C. Qian, P. Zhang; "A Novel ELM based Adaptive Kalman Filter Tracking Algorithm", *Neurocomputing Letters*, 2013.

[21] Kim W., Mechitov K., Choi J.-Y., Ham S. (2005). On target tracking with binary proximity sensors, *in: ACM/IEEE IPSN'05*, Los Angeles, CA, USA, pp. 301-308.

[22] Shrivastava N., Mudumbai R., Madhow U., Suri S. (2006). Target tracking with binary proximity sensors: Fundamental limits, minimal descriptions, and algorithms, *in: ACM Sensys'06, Boulder*, Colorado, USA, pp. 251-264.

[23] J. Singh, U. Madhow, R. Kumar, S. Suri, R. Cagley, Tracking multiple targets using binary proximity sensors, *in: ACM/IEEE IPSN'07*, Cambridge, MA, USA, 2007, pp. 529-538.

[24] D. Smith, S. Singh, Approaches to multisensor data fusion in target tracking: A survey, *IEEE Transactions on Knowledge and Data Engineering* 18 (12) (2006) 1696-1710.

[25] W. Zhang, G. Cao, DCTC: Dynamic convoy tree-based collaboration for target tracking in sensor networks, *IEEE Transactions on Wireless Communications* 3 (5) (2004) 1689-1701.

[26] F. Zhao, J. Shin, J. Reich, Information-driven dynamic sensor collaboration for tracking applications, *IEEE Signal Processing Magazine* 19 (2) (2002) 61-72.

[27] Habib Mostafaei, Mohammad Reza Meybodi. An Energy Efficient Barrier Coverage Algorithm for Wireless Sensor Networks. *Wireless Personal Communications*. 2014. 77 (3):2099-2115.

[28] Benyuan Liu, Olivier Dousse, Jie Wang, Anwar Saipulla. Strong Barrier Coverage of Wireless Sensor Networks.MobiHoc '08 *Proceedings of the 9th*

ACM international symposium on Mobile ad hoc networking and computing. 2008.

[29] Jun He, Hongchi Shi. Constructing sensor barriers with minimum cost in wireless sensor networks. *Journal of Parallel and Distributed Computing*. 2012. 71:1654–1663.

[30] Yang, G., & Qiao, D. Barrier information coverage with wireless sensors. In 28th *IEEE International Conference on Computer Communications, INFOCOM.* Rio de Janeiro, Brazil. 2009. 918–926.

[31] Silvestri, S. MobiBar: Barrier coverage with mobile sensors. In Proceedings of the the *IEEE Global Communications Conference (GLOBECOM)*. 2011. 1–6.

[32] Zhu, C., Zheng, C., Shu, L., & Han, G. A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications*. 2012. 35:619–632.

[33] Natallia Katenka, Elizaveta Levina, and George Michailidis. Tracking multiple targets using binary decisions from wireless sensor networks. *Journal of the American Statistical Association*, 108(502):398-410, 2013.

[34] Goutham Mallapragada, Yicheng Wen, Shashi Phoha, Doina Bein, and Asok Ray. Tracking mobile targets using wireless sensor networks. In 2010 Seventh *International Conference on Information Technology: New Generations (ITNG).,* pages 873-878. IEEE, 2010.

[35] Yanjun Li. Real-time surveillance performance under di_erent sensing models in duty-cycled sensor networks. *Sensors & Transducers, IFSA*, 2013..

[36] M. Naderan, M. Dehghan, H. Pedram, and V. Hakami, "Survey of Mobile Object Tracking Protocols in Wireless Sensor Networks: A Network-Centric Perspective," *Int. J. Ad Hoc and Ubiquitous Computing (IJAHUC),* 2014.

[37] Ash T. (1989) 'Dynamic Node Creation in Back propagation Networks', *technical report, Inst. for Cognitive Science, Univ. of California, San Diego*

[38] Aslam N, Philips W, Robertson W, Siva Kumar SH. (2010) 'A multi-criterion optimization technique for energy efficient cluster formation in Wireless Sensor networks', *In: Information Fusion, Elsevier*

[39] Barbancho J, Leon C, Molina F.J, Barbancho A. (2007) 'Using artificial intelligence in routing scheme for wireless networks', *In: Computer Communications 30, Elsevier*, pp. 2802-2811.

[40] Buchberger M, Jorg K.W, and von Puttkamer E. (1993) 'Laser Radarand Sonar Based World Modelling and Motion Control for Fast Obstacle Avoidance of the Autonomous Mobile Robot MOBOTIV' , *In: Proc. IEEE Int'l Conf. Robotics and Automation*, pp. 534-539.

[41] Chang C.C and Song K.T. (1996) 'Ultrasonic Sensor Data Integration and Its Application to Environment Perception'*, In: Journal of Robotic Systems*, vol. 13, no. 10, pp. 663-677.

[42] Chaudhuri S.P. and Das S. (1990) Neural Networks for Data Fusion, *In: Proc. IEEE Int'l Conf. Systems Eng*.

[43] Chengfa L, Mao Y, Guigai C. (2005) 'An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks', *In: Proceeding. Of IEEE MASS*.

[44] Chung, Y.N., Chong, C.Y., Bar-Shalom, Y. (1986) 'Joint Probabilistic data and association Distributed Sensor Networks', *In IEEE Trans. Automa. Contr*. AC-31, pp.889–897.

[45] Chen Guangzhu, Z. Lijuan, Z. Zhencai, Z. Gongbo (2010) "RBF Neural Network Based Prediction for Target Tracking in Chain-type Wireless Sensor Networks", In *IEEE Conference*, 2010.

[46] Xiong Luo, X. Chang (2015) "A Novel Data Fusion Scheme using Grey Model and Extreme Learning Machine in Wireless Sensor Networks", *International Journal of Control, Automation, and Systems*, PP. 1-8.