THOMSON REUTERS ENDNOTE

[Bhaskar* *et al.*, 5(7): July, 2016] ISSN: 2277-9655
IC™ Value: 3.00 Impact Factor: 4.116

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

THE TARGET TRACKING IN MOBILE SENSOR NETWORKS

Kesu Bhaskar*, S Babu

* M-Tech Research Scholar Avs College Of Engg & Tech Venkatachalam, Nellore, Andhrapradesh, India
Asst.Prof in CSE Avs College Of Engg & Tech Venkatachalam, Nellore, Andhrapradesh, India

DOI: 10.5281/zenodo.57158

ABSTRACT

Tracking mobile targets using wireless sensor networks is a potential surveillance application of practical importance suitable for military as well as civilian fields. It can be used for guiding robots in hard to reach areas e.g. nuclear power plants etc. The capabilities of a simple target tracking solution can be enhanced by providing guidance information for a friendly object to reach the manoeuvring adversary target. This paper presents a practical architecture for target tracking using wireless sensor network by providing solutions to key components like node localization, time synchronization, target detection and tracking. The methodology presented here gives a solution to compute the state parameters of the adversary target, tracks it and associate the same with the location in the periphery of wireless sensor networks. The simulation is done using OMNeT++ under Castalia framework.

KEYWORDS: Simulation, Adversary Target, Target Tracking, Wireless Sensor Networks.

INTRODUCTION

Wireless communications have revolutionized the way people live and have also resulted in the development of wireless sensor networks. Wireless Sensor Networks (WSNs) are distributed embedded systems consisting of a large number of low-cost, low-power, multi-functional sensor nodes. This project deals with the target tracking applications of WSNs. In target tracking, the presence of target(s) is/are detected and then tracked at regular intervals of time. An appropriate sensor can be used depending on the target‟s signature to detect the presence of the target. At every localization interval, the target tracking algorithm collects information from a set of sensor nodes which have detected the presence of a target. This is used to calculate the target‟s location. Often, Kalman filter and other filters can be used to reduce error and to predict future target locations.

The network architecture consists of a set of sensor nodes (with appropriate sensors for tracking) and a main processing node, called the base station or sink node. Target tracking involves target detection and localization at successive time instants [1].

Examples of target tracking applications can be found in the defence sector for intruder detection, in the tourism sector for tracking animals in wildlife sanctuaries, in nuclear power plants for guiding robots in critical areas. Tracking an enemy soldier or vehicle in a military battlefield, an intruder or trespasser in the perimeter area around sensitive establishment /building and or animals in a forest area are some of the potential applications of target tracking [1]. A predictive mechanism is employed during processing to predict the future position of the target. The prediction of future target position enables the majority of clusters to sleep when not needed and wake up only when the target is in the vicinity.

Characteristics of wireless sensor network

The main characteristics of a WSN include:

1. Power consumption constrains for nodes using batteries or energy harvesting
2. Ability to cope with node failures
3. Mobility of nodes

4. Communication failures
5. Heterogeneity of nodes
6. Scalability to large scale of deployment
7. Ability to withstand harsh environmental conditions
8. Ease of use

Sensor networks may consist of many different types of sensors such as seismic, low sampling rate magnetic, thermal, visual, infrared, acoustic ,radar, which are able to monitor a wide variety of ambient conditions that include the following:

[1] Temperature
[2] Humidity
[5] Vehicular movement
[6] Lightning condition
[7] Pressure
[8] Soil makeup
[9] Noise levels
[10] The presence or absence of certain kinds of objects
[11] Mechanical stress levels on attached objects

RELATED WORKS

Sensor networks are typically used to monitor the environment, one fundamental issue is the target tracking, whose goal is to trace the roaming paths of moving objects/individuals in the area in which sensors are deployed. This problem is challenging in two senses:

i) There are no central control mechanisms and backbone network in such an environment and
ii) The wireless communication is very limited.

At present, location tracking is done using GPS. However, GPS has its limitations. It cannot be used in most indoor environments. It depends on Line of Sight. Also in non-urban outdoor settings, GPS does not yield accurate results because it depends too much on factors such as terrain, foliage and topographical settings of the place where the object is located. Since, GPS receivers may be too large, too expensive or too power intensive, using wireless sensor networks provides us with a better alternate for location tracking since the nodes are relatively small, inexpensive and low power devices. They are much more viable considering economic and convenience constraints.

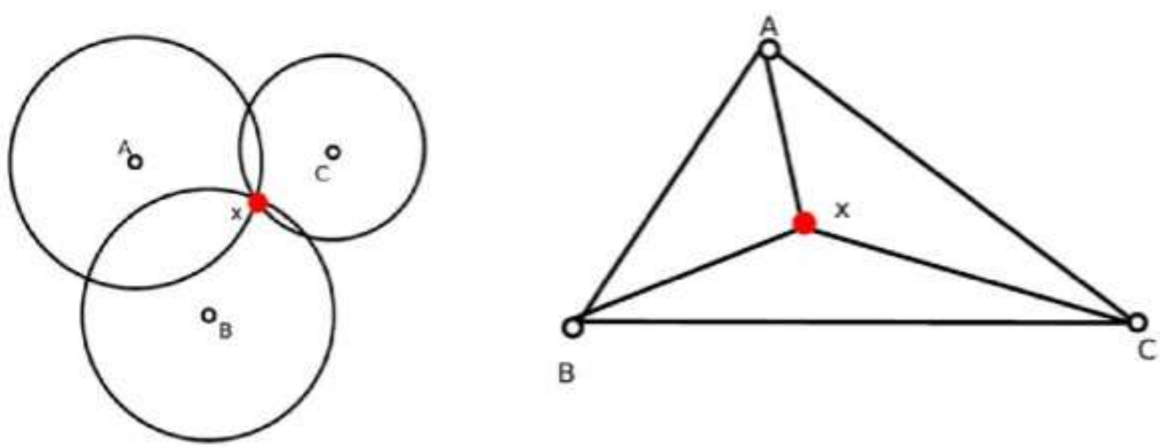


Fig.1 (a) Trilateration

Some other techniques [5] have also been proposed in the past as alternates to the trilateration technique. They are:

Infrared: RFID tags emit infrared radiations carrying a unique ID. This is received by a number of receivers scattered across a facility which resolve the location of the badge based on distance.

Ultrasound: These are also distance based systems but provide a better estimate by measuring time-of-flight of ultrasound with respect to a reference RF signal. MIT"s Cricket system is an example of this.

Radio: The systems which utilize radio waves provide a better approximation for location detection because of the ability of these waves to penetrate various materials. Instead of using differences in arrival times as in Ultrasound, these systems utilize signal strength to measure the location.

Below are listed some of the techniques proposed till now in an attempt to solve the problem of Target Tracking in wireless sensor network:

The following technique has been proposed by Yu-Chee Tseng, Sheng-Po Kuo, Hung Wei Lee and Chi-Fu Huang at the Department of Computer Science and Information Engineering at National Chiao-Tung University, Taiwan [6]. The paper discusses the following technique:

Whenever an object is detected, based on the distances of the sensor nodes from the object, three closest nodes are selected to monitor the movements of the object. At any time, these sensors monitor the movements of the object. These three agents (master and slaves) will perform the trilateration algorithm and calculate the (x, y) coordinates of the object. The sensors tracking the object keep changing as the object moves. The election process is constantly done based on the location of the object at different time instants.

There is a certain signal strength threshold used to determine when to revoke/reassign a slave agent. The master may forward tracking histories to the location server. The paper has discussed the above technique with some constraints on the movements of the object. The object is assumed to be moving at a constant speed of 1-3 m/s and the sensors are not able to detect the object if it moves at a speed of more than 5 m/s.

This technique has been proposed by Asis Nasipuri and Kai Li at the Department of Electrical and Computer Engineering at The University of North Carolina at Charlotte [7]. The technique is as follows:

Consider a network in which sensor nodes are scattered at random. These nodes track the object and relay the information to the Control Unit as and when required. For various operations such as signal processing, data transmission, information gathering and communications the sensor nodes have a memory, a processor and supporting hardware. The sensor nodes have limited transmission range. They rely on store and forward multi-hop packet transmission to communicate. Each beacon signal is an RF signal of a separate frequency on a narrow directional beam with a constant angular speed of ώ degrees/s. Thus, the transmissions are distinguishable. The sensor nodes determine their angular bearings with respect to these signals. The supposition in this case is that transmission range is sufficient for the beacon nodes to be received by all sensor nodes in the network.

In the paper, the authors have considered a rectangular network area at the corners of which are located the four beacon nodes. Consequently, each sensor node receives periodic bursts of the four beacon nodes with the same period of 360/ ώ seconds. The localization principle is based on a sensor node noting the times it receives the different beacon signals and evaluating its angular bearings with respect to the beacon nodes by triangulation.

Data-centric and Location-centric approaches to the Target Tracking problem have been elaborated by R. R. Brooks, Sr. Research Associate, Applied Research Laboratory, Pennsylvania State University, P. Ramanathan and A. M. Sayeed, Professor, Electrical and Computer Engineering Department, University of Wisconsin [8].

In the Data-centric approach, sensor nodes respond to particular requests. Whenever the nodes detect a request corresponding to the data they have, they transmit the data. Other nodes do not respond but take note for future use. Subscribed nodes receive data over the network.

Diffusion routing is one of the solutions proposed to route data in the data-centric approach. The authors illustrate a location-centric approach developed at the University of Washington. In this case, a Route Request (RREQ) is needed to forward data from cell to cell unlike the creation of paths in diffusion routing. The cells are addressed by their geographic locations. As the RREQ propagates, state information is temporarily deposited in the network to identify an efficient route from the source to the destination.

On receiving the RREQ, the node in the addressed cell responds with Request Reply (RREP) which is routed to the destination cell resulting in a single path from source to destination cell along which data is sent to all nodes in the latter by the manager node.

The other technique for location tracking was proposed by Saikat Ray, Rachanee Ungrangsi, Francesco De Pallegrini, Ari Trachtenberg and David Starobinski at IEEE Infocom 2003 [5].

No.	Simulator	Program-ming language/	Key features	Limitations
1	Ns-2	C++	-Easy to add new protocols. -A large number of protocols available publicly. -Availability of a visualization tool.	-Supports only two wireless MAC protocols, 802.11, and a single-hop TDMA protocol.
2	TOSSIM	nesC	-High degree of accuracy or running the application source code unchanged. -Availability of a visualization tool.	-Compilation steps lose the fine- grained timing and Interrupt -properties of the code,
3	GloMoSim	Parsec	-Parallel simulation capability. -It is tailored specifically for wireless networks. -Availability of a visualization tool.	-Effectively limited to IP networks because of low level design assumptions. -Unavailability of new protocols.
4	Avrora	Java	-Can handle networks having up to 10,000 nodes. -Enables validation of time-dependent properties of large-scale networks	-Fails to model clock drift. -50% slower than TOSSIM. -Cannot model mobility.
5	SENS	C++	-Platform-independent Users can assembl e application-specific environments - Defines an environment as a grid of interchangeable tiles.	-Not accurately simulate a MAC protocol. -Provides support for sensors, actuators, and physical phenomena only for sound.
6	Castalia	C++	-Physical process modeling, sensing device bias and noise, node clock drift, and several MAC and routing protocols implemented. -Highly tunable MAC protocol and a flexible parametric physical process model.	-Not a sensor specific platform. -Not useful if one would like to test code compiled for a specific sensor node platform.
7	OMNeT++	C++	-Powerful graphical User Interface	-Number of protocol is not

			(making tracing and bugging easier) -Simulate power Consumption problem	large enough. -Compatibility problem (not portable).

Table I Comparison Of Wireless Network Simulators

Different flavours of distributed tracking methodology are used in [9],[10], [11], [12], [13]. Both [10] and [12] use a distributive predictive approach. The basic localization algorithm used in both methods is a simple centroid algorithm, but [10] uses a message passing version of Kalman Filtering method. In [14] the problem of tracking multiple targets is dealt. It discusses multiple hypothesis tracking and joint probabilistic data association filter based tracking.

One of the factors affecting the performance of the system is the network architecture used in the system. In cluster-based tracking algorithms, such as those reported in [15], [16], the cluster member nodes detect the target"s presence and report the sensed values to the cluster head.

The cluster head processes the information to determine the target location. The advantage of the cluster based approach is reduced energy consumption, which maximizes the network lifetime. Another group of tracking algorithms are based on tree topology. In [17], [18], nodes that detect the target communicate with each other and select a root node that collects data from the nodes via a distributed tree. Although the tree based approaches track the moving object more accurately, tree configurations cause high energy consumptions. In order to improve tracking accuracy due to sensing and localization errors, often filters such as Kalman filter, particle filter, etc. are used.

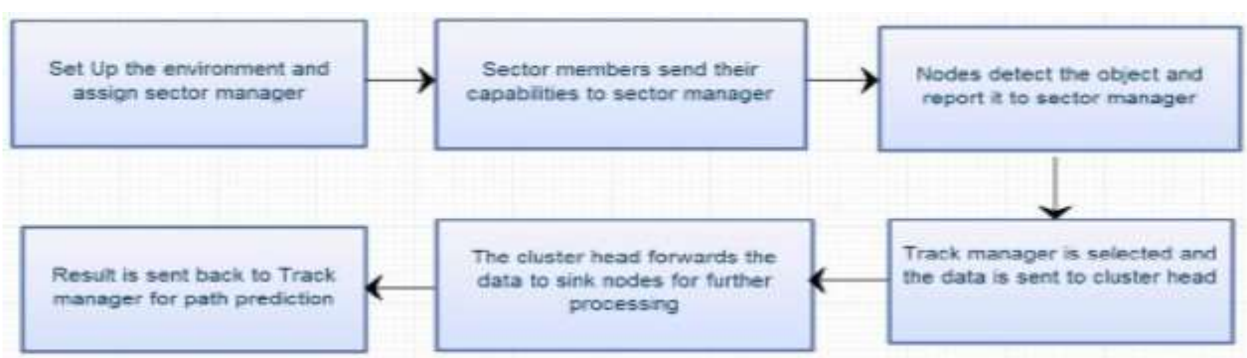
In [19], [20], prediction based tracking algorithms are presented; here, the subsequent position of the target is predicted based on the current moving speed and direction of the target.

A survey of various network simulation environments simulators can be found in [21]. These include NS-2, Glo-MoSim, SENS, OMNeT++, TOSSIM, Avrora etc. Castalia under OMNeT++ gives a generic and realistic framework for evaluating algorithms developed for WSN. More details on OMNeT++ and Castalia can be found in [22], [23].

SYSTEM DESIGN

Proposed System

The block diagram below shows the proposed design methodology for target tracking in wireless sensor network.



The steps involved in target tracking are as follows:

a. The nodes are arranged or scattered, and have varied orientations. One agent is assigned to each node.
b. The environment is first partitioned into sectors and sector managers are then assigned.
c. Sector members send their capabilities to their managers each manager then generate and disseminate a scan schedule.

d. Nodes in the scan schedule perform scanning actions and detections and are reported to manager and track manager is selected.
e. Track manager discovers and coordinates with tracking nodes.
f. New tracking tasks may conflict with existing tasks at the node
g. Tracking data sent to an agent which performs the fusion and target is tracked.

Target Tracking Architecture

A distributed hierarchical target tracking architecture is proposed. The area in which the target needs to be tracked is a 2-D sensor network deployment (R2) [1].

Fig. 2 Organisation overview for Target Tracking in Wireless Sensor Network

The system consists of a large number of sensor nodes distributed over the region of interest. The distribution of the nodes is uniform to obtain maximum coverage of the entire region. The entire region of interest is divided into a number of clusters. Each cluster has a Cluster Head (CH) and several normal nodes (NN). The cluster heads perform computationally intensive tasks in the cluster.

The normal nodes are capable of sensing seismic, acoustic and magnetic activity. It is assumed that there is a high probability of targets originating outside the region of interest and entering through the periphery, therefore, nodes with higher capabilities are deployed at the periphery. Cluster heads then find out the neighbouring nodes by broadcasting information to all the nodes in the close proximity of the cluster heads. The target originates from outside the region of interest and enters the coverage area, through the periphery. It is assumed that there is a high probability of targets originating outside the region of interest and entering through the periphery, therefore, nodes with higher capabilities are deployed at the periphery.

Sensors transmit the sensed information using a single- hop or a multi-hop path to the base station. The base station processes the sensed data to determine the targets path, once a target is detected in a particular cluster or, a track manager is assigned and follows the target in that cluster. The track is generated as shown in the figure above. The base station processes the sensor data, determines the target's trajectory and may provide the navigation guidance to the friendly object. The selection of sensors is dependent on the target type and the signature emitted by the target. For this paper, we assume that the sensors have both the detection as well as ranging capacity and the same is available to sensor node. A combination of passive infrared sensors and ultra-sonic ranging sensors can be used to for object detection.

The flow of information and generation of tree when a target is encountered in the proximity of the sensor network is shown in the fig 3. It is a dynamic process and changes as the target progresses.

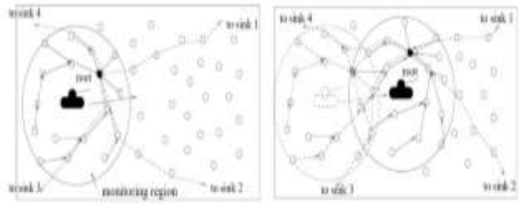


Fig.3 Tree Generation in Target Tracking

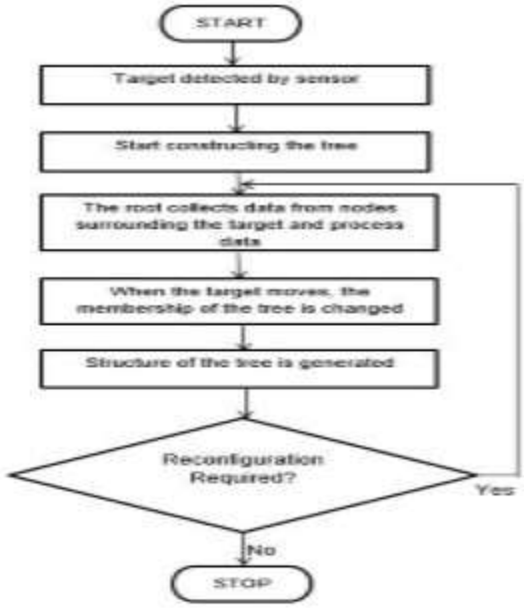
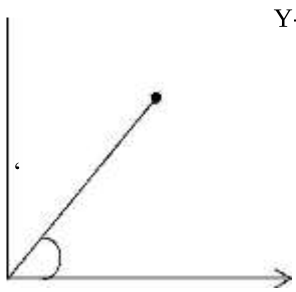


Fig. 4 Flowchart showing Tree Generation Algorithm

The flowchart above shows the process of target tracking and formation of convoy tress. The root collects data from nodes surrounding the target and processes the data. When the target moves and membership of the tree changes the structure of the tree is reconfigured when required.

Target Tracking Methods

The target tracking problem is to determine the target state parameter X(t), defined below, at different instances of time.



Y-Axis /\

X(t)=[Px,Py,Vx,Vy,θ]

Θ

X-Axis

X(t)=[Px,Py,Vx,Vy,θ] 1

where [Px,Py] "belongs to" R2 are the X and Y co-ordinates of the target, [Vx, Vy] are the velocity components along X and Y directions and θ is the direction of motion of the target.

Basic Centroid Method

Let „n" be the number of sensors sensing the target"s physical attribute above the set threshold and that all the „n" sensors report the sensed value to the cluster head within a time period T. (Time Period T between states t-1 and t should be empirically chosen as per the requirement of the application). In basic centroid method X co-ordinate and Y co-ordinate of the target is computed as follows:

$$P_x = \frac{\sum_{i=1}^{n} S_{ix}}{n} \quad 2$$

$$P_y = \frac{\sum_{i=1}^{n} S_{iy}}{n} \quad 3$$

Where *Six* and *Siy* denotes the X Co-ordinate and Y co-ordinate of the ith sensor.

Weighted Centroid Method

Basic centroid method can be improved by assigning weights to the location value depending upon the sensed values. Assignment of weights depends upon the way to which a physical phenomenon diffuses over space. We set $Wi = Vi\alpha$, where Wi is the weight assigned to i[th] node and α is an empirically chosen constant.

In weighted centroid method the position of the moving target can now be computed as

$$P_x = \frac{\sum_{i=1}^{n} S_{ix} W_i}{\sum W_i}$$

$$P_y = \frac{\sum_{i=1}^{n} S_{iy} W_i}{\sum W_i} \quad 5$$

The velocity components and direction of motion of the target are calculated as follows:

$$Vx(t) = \frac{|Px(t) - Px(t-1)|}{T} \quad 6$$

$$Vy(t) = |Py(t) - Py(t-1)| \quad 7$$

$$\theta = \tan^{-1} \frac{[Py(t) - Py(t-1)]}{[Px(t) - Px(t-1)]} \quad 8$$

Predictive Regenerative Algorithm:

The centroid and weighted centroid algorithm discussed above were found error-prone, since failure to report presence of target by a few sensors in the vicinity would result in large variations in computed position of the target. Hence, a prediction based algorithm was developed. In this, the next state of the target is predicted based on its current parameters. The position of the target at a particular time t is first computed using the weighted centroid algorithm. It is then compared with the predicted position. Depending upon the comparison, a correction is provided and the position is recomputed. The motivation behind this approach is that the motion parameters of a moving object cannot change drastically and is dependent on its previous state. A constant velocity model (over a short time-scale) is used, where it is assumed that the velocity of the target is constant to predict the next position of the target.

The error function, defined as the Euclidean Distance between the actual position of the target and the position computed, is given by:

$$Er(t) = \sqrt{(Px - P_x^A)^2 + (Py - P_y^A)^2}$$

where $[P_x^A, P_y^A]$ are the actual co-ordinates of the target.

TARGET TRACKING IMPLEMENTATION

Assume that the n sensors report their sensed reading to the cluster head at the time instant *t*. The sensors with top 3 readings are considered for the target"s localization. Let these sensor nodes be *Si*, *Sj* and *Sk* with their respective sensor readings being r_i, r_j and r_k. Let us assume the locations of the three nodes being (xi, yi), (xj, yj) and (xk, yk). A variation of the weighted centroid algorithm is used to calculate the target location for the time instant t. The following formulation

is used to calculate the target location (X_t^T ,Y_t^T).

$$X_t^T = \frac{r_i x^i + r_j x^j + r_k x^k}{r_i + r_j + r_k}$$

$$Y_t^T = \frac{r_i y^i + r_j y^j + r_k y^k}{r_i + r_j + r_k}$$

13
4

It has been assumed that the calculated target state is available at every cluster head. Using the immediate previous state and the current state of the target, its direction and velocity (θ_t^T and v_t^T) are calculated.

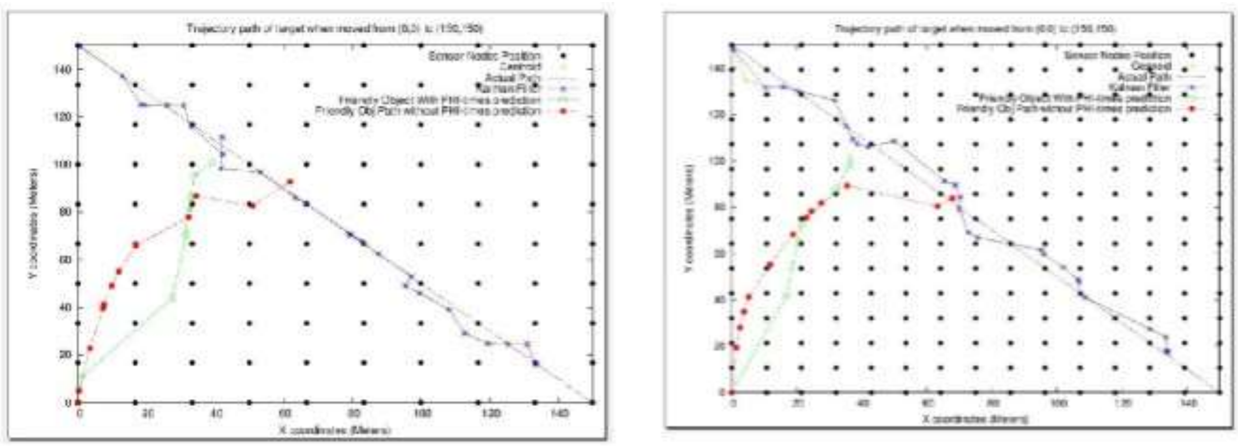


Fig. 5 Straight line target trajectory path with 100 nodes and 255 nodes

Measured target state (X_t^T , Y_t^T) may contain noise (random variations) and other inaccuracies (e.g. processing inaccuracies). The Kalman filter is used to obtain the target state that tends to be closer to the true target state because of its lesser computational overhead compared to the Particle filter.

Kalman Filter

A Kalman filter is an optimal estimator – i.e. infers parameters of interest from indirect, inaccurate and uncertain observations. It is recursive so that new measurements can be processed as they arrive.

The Kalman filter uses a system's dynamics model (e.g., physical laws of motion), known control inputs to that system, and multiple sequential measurements (such as from sensors) to form an estimate of the system's varying quantities (its state) that is better than the estimate obtained by using any one measurement alone. As such, it is a common sensor fusion and data fusion algorithm.

All measurements and calculations based on models are estimates to some degree. Noisy sensor data, approximations in the equations that describe how a system changes, and external factors that are not accounted for introduce some uncertainty about the inferred values for a system's state. The Kalman filter averages a prediction of a system's state with a new measurement using a weighted average. The purpose of the weights is that values with better (i.e., smaller) estimated uncertainties are "trusted" more. The weights are calculated from the covariance, a measure of the estimated uncertainty of the prediction of the system's state. The result of the weighted average is a new state estimate that lies between the predicted and measured state, and has a better estimated uncertainty than either alone. This process is repeated every time step, with the new estimate and its covariance informing the prediction used in the following iteration. This means that the Kalman filter works recursively and requires only the last "best guess", rather than the entire history, of a system's state to calculate a new state. More details can be found in [24].

RESULT AND DISCUSSION

The algorithm described earlier have been implemented in the Castalia Simulator framework based on OMNeT++ 3.3p1.The network studied consist of various network deployment scenarios like uniform or random with varying field sizes and number of nodes.

The average inter node distance is taken to be 15 meters. The node density is the same for all the network sizes to maintain the same inter-node distance and to avoid increasing the cluster head transmission power. It has been assumed that the target is an acoustic source. All the sensor nodes are having acoustic sensing modality with uniform sensing range. All the sensor nodes use the CC2420 radio stack with single channel. Transmission power of the cluster head is kept at 0 dBm and that of its cluster members at -5 dBm.

The graph below shows different deployment scenarios i.e. 100 nodes, 255 nodes and 900 nodes with field size 150X150m.

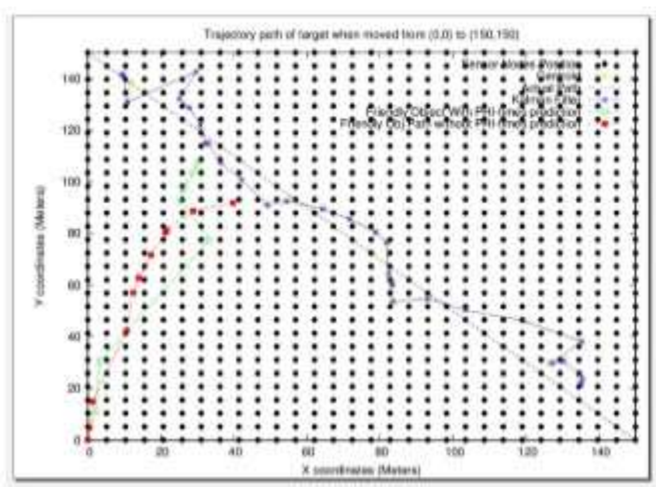


Fig.6 Straight line target trajectory path with 900 nodes

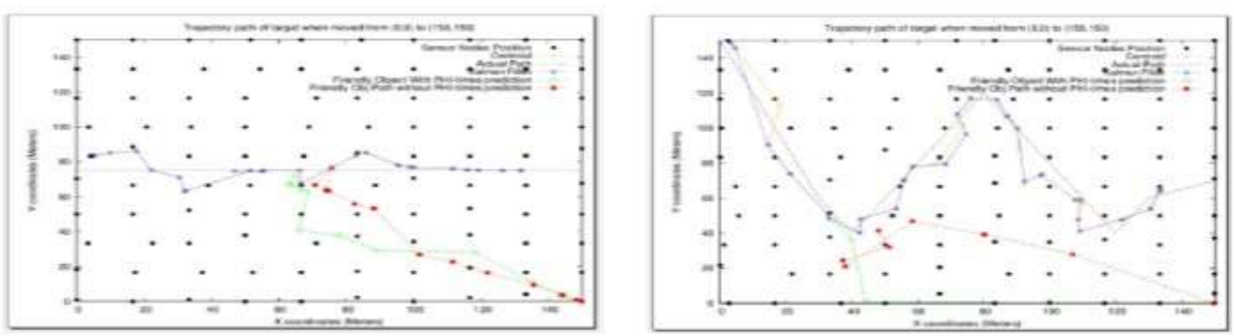


Fig. 7 Straight line trajectory path and near sine wave trajectory of the target

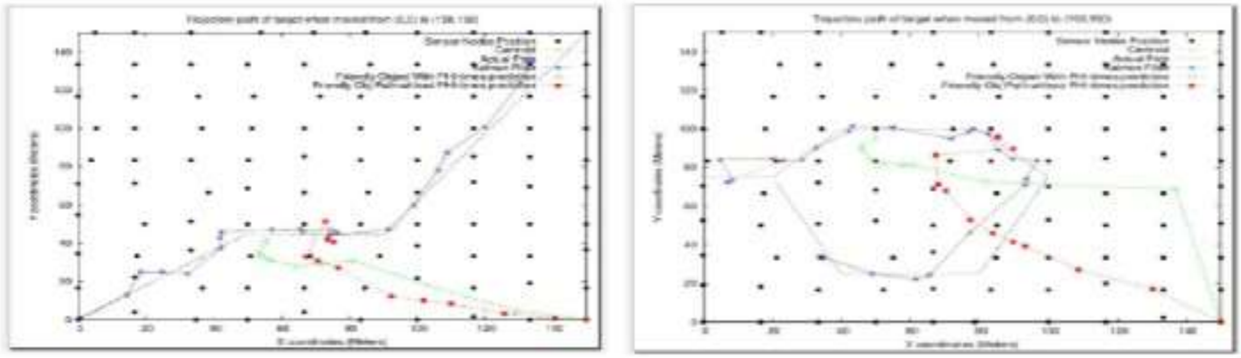


Fig. 8 Step trajectory and circular trajectory of the target

The graphs above show the effect of increasing the number of nodes in determining the targets path in a particular field size. From these graphs we can conclude that if the numbers of nodes are increased from 100 nodes to 900 nodes there is an increase in inconsistency of the result as the computation become more complex.

Now we will consider different types of representative target motion models as described below:

- Straight line trajectory: In this type of trajectory the target moves in a straight line as shown in fig 7.
- Near Sine wave trajectory: Target traverses along a near sine curve as shown in fig 7.
- Step trajectory: Target traverses the area in three phases as shown in fig 8.
- Circular trajectory: Target traverses the area in circular path as shown in fig 8.
- Random trajectory: This represents a target in random motion as shown in fig 9.

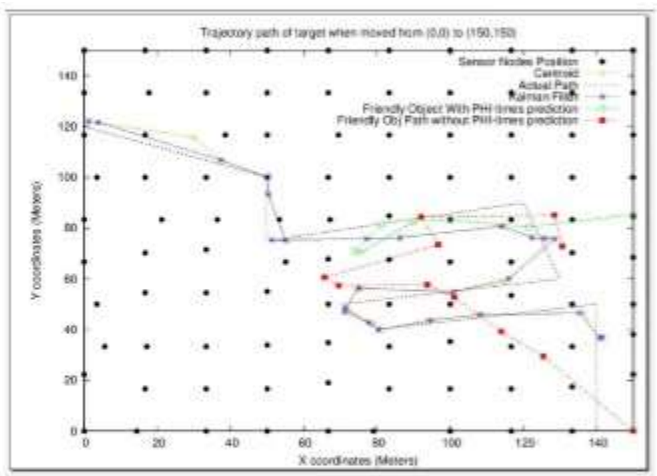


Fig. 9 Random trajectory of the target

From the above graphs we can conclude that target tracking mechanism discussed above works well with linear motion targets and inconsistencies increases with targets having circular and random motion.

CONCLUSIONS

In this research paper we considered the problem of tracking the target and the mechanism to determine the target‟s position. Target tracking framework was discussed in detail along with cluster tree generation and a mechanism was devised to implement them. OMNeT++ and Castalia framework are used for implementation of target tracking in wireless sensor network. The simulation of different deployment scenarios is shown. From the figures studied it can be concluded that the scheme is suited for smooth linear target motion.

This research has demonstrated a methodology which can be used for tracking the target and guiding a friendly moving object along the periphery of the target. This can be extended to track and guide multiple friendly moving objects in the periphery. This work can also be extended to intercept unknown target in the periphery of wireless sensor network.

REFERENCES

[1] Sachin Deshpande, Umesh Kulkarni, Mritunjay Ojha, "Target Tracking in Wirelesss Sensor Network," in http://www.ijetae.com/files/Volume3Issue9/IJETAE_0913_27.pdf.
[2] A. Savvides, M. Srivastava, L. Girod, and D. Estrin, "Localization in sensor networks," in Wireless Sensor
[3] Networks, C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Eds. Boston, MA: Kluwer Academic Publishers, 2004, ch. 15.
[4] A. Arora et al., "A line in the sand: A wireless sensor network for target detection, classification and tracking," Computer Networks, vol. 46, no. 5, pp. 605–634, 2004.
[5] S. Bhatti and J. Xu, "Survey of target tracking protocols using wireless sensor network," in Proceedings of the 2009 Fifth International Conference on Wireless and Mobile Communications, 2009, pp. 110–115.
[6] Saikat Ray, Rachanee Ungrangsi, Francesco De Pellegrini, Ari Trachtenberg and David Starobinski. Robust Location Detection in Emergency Sensor Networks. IEEE INFOCOM 2003.
[7] Yu-Chee Tseng, Sheng-Po Kuo, Hung-Wei Lee and Chi-Fu Huang. Location Tracking in a Wireless Sensor Network by Mobile Agents and its Data Fusion Strategies. International Workshop on Information Processing in Sensor Networks (IPSN), 2003.
[8] Asis Nasipuri and Kai Li. A Directionality based Location Discovery Scheme for Wireless Sensor Networks. WSNA "02 Atlanta, Georgia, September 28, 2002.
[9] R.R. Brooks, P. Ramanathan and A.M. Sayeed. Distributed Target Classification and Tracking in Sensor Networks. Special Issue in Sensor Networks, Revision, January 2003.
[10] Nirupama Bulusu, Deborah Estrin, Lewis Girod and John Heidemann. Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems. Proceedings of ISCTA 2001, Lake Ambleside, UK, July 2001. Invited Paper.
[11] R. Olfati-Saber and N. F. Sandell, "Distributed tracking in sensor networks with limited sensing range," in Proc. of American Control Conference, 2008, pp. 3157–3161.
[12] Z. Wang, H. Li, X. Shen, X. Sun, and Z. Wang, "Tracking and predicting moving targets in hierarchical sensor networks," Proc. Of IEEE Conference on Networking, Sensing and Control, pp. 1169–1173, 2008.
[13] J. Liu, M. Chu, and J. E. Reich, "Multi-target tracking in distributed sensor networks," IEEE Signal Processing Magazine, pp. 36–46, 2007.
[14] J. Shin, L. Guibas, and F. Zhao, "A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks". In Proc. of 2nd workshop on Information Processing in Sensor Networks (IPSN), April 2003..
[15] A. Oka and L. Lampe, "Distributed target tracking using signal strength measurements by a wireless sensor network," IEEE Journal on Selected Areas in Communications, vol. 28, no. 7, pp. 1006–1015, 2010.
[16] Chong, C.Y., Zhao, F., Mori, S., Kumar, S.: Distributed tracking in wireless ad hoc sensor networks. In: Proceedings of the Sixth International Conference of Information Fusion, vol. 1, pp. 431–438 (July 2003)
[17] Goshorn, R., Goshorn, J., Goshorn, D., Aghajan, H.: Architecture for cluster-based automated surveillance network for detecting and tracking multiple persons. In: ACM/IEEE ICDSC, pp. 219–226 (September 2007).
[18] Zhang, W., Cao, G.: DCTC: Dynamic Convoy Tree-based Collaboration for Target Tracking in Sensor Networks. IEEE Transactions on Wireless Communications 3(5), 1689–1701 (2004)
[19] Zhang, W., Cao, G.: Optimizing tree reconfiguration for mobile target tracking in sensor networks. In: IEEE INFOCOM, pp. 2434–2445 (March 2004).
[20] Xu, Y., Winter, J., Lee, W.C.: Prediction-based strategies for energy saving in object tracking sensor networks. In: IEEE International Conference on Mobile Data Management, pp. 346–357 (2004).