

# A Comprehensive Review on Energy Conserved Multi-object Tracking over Wireless Sensor Networks

S.P.Ponsudha

P.G Student, Department of IT  
PSN college of Engineering and  
Technology, Tirunelveli District,  
Tamilnadu, India

R.Lakshmi Priya

Assistant professor, Department  
of IT,  
PSN college of Engineering and  
Technology, Tirunelveli District,  
Tamilnadu, India

L. Josline Edward

Professor, Department of IT  
PSN college of Engineering and  
Technology, Tirunelveli District,  
Tamilnadu, India

## ABSTRACT

Sensor networks are used in wide variety of real-world applications including detecting, tracking and monitoring. One of the main issues in the deployment of the sensor nodes in the field of target tracking is power optimization. Various methods of target tracking are available such as making improvement in hardware, processor design etc. But still, an optimized solution to this problem is not available. To rectify this problem here we formulate the mechanism that the sensors may be put into a sleep mode with a timer that determines the sleep duration. The assumption made is that a sensor that is asleep cannot be with or waken up, and hence the sleep duration depends on the timely information it receives during the awaken period. The objective is to track the location of the objects to within the accuracy of the range of the sensor. The result is in the form of a trade-off between the energy savings and the tracking errors that result from the sleeping policies of the sensors. Sub-optimal sleeping policies that optimize this trade-off are designed, and their performance in a different tracking architecture is examined.

## General Terms

Sub optimal sleeping policies, partially observable state based formulations.

## Keywords

Target tracking, Detecting, Monitoring, Energy efficiency, Power optimization, Centralized and Decentralized tracking.

## 1. INTRODUCTION

Sensor, a handy and versatile sensing node employed in the field of military, wildlife studies, environmental monitoring, surveillance, vehicle tracking etc [1]. The important feature of the sensor network is to minimize the power consumption. Each node in the network is responsible for sensing the data and to inform its neighbours or base station after processing the identity of the tracked target.

Event driven data model enables the sensor node to send data over the wireless channel only when there is a need to sense the data or when the events of interest occur [2]. Moreover, the sensor node itself consumes power so that the radio power usage in the node is also taken into consideration for control.

With recent advances in hardware, ubiquitous sensing is achieved and the power consumed by the constituents of a Wireless Sensor Node is minimized [3]. Sophisticated Radio standards such as IEEE 802.15.4 (Zig Bee), IEEE 802.15.2

(Ultra Wide Band) requires relatively high power for a short transmission range. Hence a Battery Driven approach and stochastic models are needed for the longer network's lifetime.

Energy efficiency can also be obtained by compressing and reducing the amount of data transmitted. From the results of [4] it was revealed that, the data funneling and compression schemes reduces only half the power. The scheduling approach is considered to be a better alternate for consuming the transceiver power in case of frequent transmissions [5]. Past researches discuss a mechanism for power management by altering the power saving modes of the sensor node is another profitable approach [6].

From [6] & [7] it was assumed that the wake-up channel approach is infeasible for a real-time application like tracking. Hence, this paper considers other alternatives to the wakeup channel approach and schemes that lower the cost afforded. The switching approach between various powers saving mode is extensively used for the applications like tracking.

The information available from the corresponding neighbour node helps the node to make a decision and accordingly switch-over from one mode to another as in [6]. Still, a question arises, which scheme will be a successful for tracking. Single Target Tracking (STT) and Multi Target Tracking (MTT) are the tracking methods of which Multi Target Tracking is not said to be a mere extension of the single target tracking due to the Data association problem. This problem arises whenever the identity of the objects cannot be determined from the observations. Thus even if all the locations where objects are known exactly, it may not be known which location corresponds to which object. This gives rise to several set of possibilities that make the optimal solution difficult.

Here, the focus is on sleeping policies for tracking multiple objects and the design problem where the object movement is not known. For perfect predictions we use POMDP and Q-Learning to a Markov decision process (QMDP).

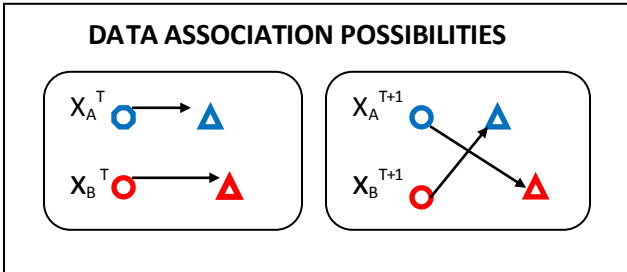
The simulation results focus on two object case. The effectiveness of centralized and decentralized tracking schemes is also one of the issues to be addressable. The following sections discuss them in detail.

In Section 2 the tracking problem and definition of optimization problem are discussed. In Section 3 suboptimal solutions are formulated. The simulation results are discussed in Section 4 and then prove this approach is efficient optimized approach which induces less data error and tracking cost over

the network. Finally the summarization and conclusion in arrived in Section 5.

## 2. PROBLEM FORMULATION

Power consumption can be actively performed in the levels of communication, data processing and sensing module itself. In case of hardware or circuit level, power consumption is achieved in the areas such as change over from one state to another through which tracking an object is done. Tracking multiple objects is not as simple as tracking single object due to data association problem [8]. MTT is not a trivial extension of single target tracking. To elaborate, consider the simple case of tracking two targets, shown in the following Figure.1



**Figure.1 Example of Data association problem**

At time  $t$ , say that target  $X_A$  is believed to be located at point, and that target  $B$  is believed to be located at a in their own confined area, as shown in Figure 1. At time  $t$ , the system observes two measurements with the changing object position as shown in [7]. The further uncertainty in multiple target tracking is the question of which measurement was generated by target  $A$  and which was generated by target  $B$ . The following section discusses the formulation of sub-optimal sleeping policies for solving this problem by DPTA like approaches [8].

### 2.1 POMDP Formulation

POMDP's provide a frame work for agents that learn how to act in their environment or world. The POMDP model incorporates uncertainty about the agent's perceptions, actions and feedback. The formulation is based upon the sensor network with  $n$  sensors. Without overlapping, the sensors covers the whole area is the assumption made as opposed to wakening based approach in[9]. The whole region is in turn divided into  $n$  cells and the sensor in its coverage area is responsible for performing tracking operation within the cell. Each sensor can be in one of two states: awake or asleep. A sensor in the awake state consumes more energy than one in the asleep state. However, only in the awake state object sensing can be performed. An awake sensor can only detect whether one or more objects is within its range and can detect neither the exact number of objects present nor which objects are present.

Multiple sensors receiving simultaneous noisy observations of the objects is the complicated area under analysis. The movement of each object to be tracked is described by a first-order Markov chain whose state is the current location of the object to within the accuracy of a cell. However, we also append an additional state that occurs when the object leaves the network. Thus there are  $n+ 1$  possible state for each object. Here track  $q$  objects are tracked that move independently according to their first order Markov-chain process. The combined state of  $q$

objects is denoted by the vector of length  $q$ . There are  $(n+1)q$  possible states for this vector. The state  $T-[n+1...n+1]$  is the terminal state that occurs when all objects have left the network. After this state no further cost is incurred. A kernel  $P$  is defined such that  $P(x,y)$  is the probability that the next state is  $y$ . Here the current state is termed to be  $x$ . Then 't' time steps towards the future is predicted by the equation by defining  $P1=P$  inductively as in [10]

$$P^t(x, y) = \sum_{z \in \{1, \dots, n+1\}^q} P^t(x, z) P(z, y).$$

In order to provide the means of centralized control, extra node called as centralized controller is assumed[10]. This controller keeps track of the state of the network and it is responsible for assigning sleep times for the sensors that are awake[10]. For a time period of one unit, each sensor is assumed to remain awake. During the mean time the following actions takes place such as

- The sensor sends the binary information to the central controller which indicates that indicates whether one or more objects are within its range.
- The sensor receives a new sleep time (which may equal zero) from the central controller.
- The sleep time input initializes a timer at the sensor that is decremented by one time unit each time step.
- When this timer expires, the sensor wakes up.
- Since we assume that wakeup signals are impractical, this timer expiration is the only mechanism for waking a sensor.

The costs present in the tracking problem are described below:

- Energy cost: This cost  $C > 0$  is calculated for the period when the sensor is assumed to be awoken. As the Battery life of the sensor decreases corresponding variations also reflect in the energy cost. Mathematical representation of the energy cost is represented as [10]

$$\sum_{i=1}^n C$$

- Tracking error: The tracking error is also addressed as the cost which can further be subdivided into two components. They are observation error and data association error. The first error occurs when we fail to observe an object. In case of repeated observations, the cost function  $J$  can be defined as follows[10]

$$J^*(I_0) = \min_{\mu_0, \mu_1, \dots} J(I_0, \mu_0, \mu_1, \dots)$$

Thus the goal is to find the solution to this optimization problem by prediction tracking algorithm as in [11],[12]. Depending upon different values of  $c$  optimal sleeping policies can be obtained. This kind of optimization problem is known to be partially observable Markovian Decision process, where  $I_0$  is the initial known state of system.

In case of Decentralized implementation the parameter  $Z_k$  which is the observation available to the central controller at time  $k$  is the same as the information communicated across the neighbouring nodes where  $Z_k = (S_k, r_k)$  and the value of  $S_k$  depends on the object sensed or not and  $r_k$  depends on the sleep timer and the total information available to the next node depends on  $Z_k$  alone.

## 2.2 Dealing with partial observability

Partial observability presents a problem since the information for decision-making at time given in is bounded in memory. To overcome this problem a sufficient statistic is present for optimization in memory. The statistic can be presented as the probability distribution of the state  $x_k$ , given  $I_k$  which is the control input information of the sensor. The residual sleep time is found to be observable and the corresponding statistic is written as

$$\vartheta_k = (P_k, \gamma_k)$$

Where  $P_k$  is the probability mass function. The total cost in [10] is defined as

$$J(P_0, \gamma_0, \mu_0, \mu_1, \dots) = \epsilon \sum_{k=1}^{\infty} g(P_k, \gamma_k) \vartheta_0$$

## 3. SUBOPTIMAL SOLUTION

The optimal policy can also be found by solving through the Bellman's Equation [13]

$$J(p, \gamma) = \min_{\mu} \{ g(P_1, \gamma_1) + J(P_1, \gamma_1) P_0 = p, \gamma_0 = \gamma, \mu_0 \\ = \mu(p, \gamma_0) \}$$

As in [13] deriving suboptimal solutions to this problem, We will make assumptions about the observations that will be available in the future. These assumptions will allow simplifying the evolution which are opposed to MCMC approach in [14].

### 3.1 Q<sub>MDP</sub> Policy

Q<sub>MDP</sub> solution is one in which it is assumed that the partially observed state becomes fully known after a control input has been chosen [13]. In this assumption that there will be no future data association errors and thus the only tracking costs present in designing this policy are observation errors. As the result the Bellman-equation can be written as in [9]

$$J^l(P) = \min_{\mu} (\sum_{b \neq \tau} (\sum_{j=1}^k \sum_p P^j(b) \sum_{i=1}^q 1(b_{i=i})) + \\ c p P^{u+1}(b) + p P^{u+1}(b) J(l) \partial_b)$$

The terms inside the minimization represents the tracking cost, energy cost, future cost at given sleep time of  $u$ . Thus the equation represents the assumption of future observations. In this solution, it is assumed more information than is actually available.

The right hand side of the equation is calculated continually until the convergence state is reached. This policy does not perform well in cases when the number of sensors or number of objects becomes large. Thus, the cost function obtained under this is the lower bound on optimal performance.

### 3.2 FirstCostReduction Policy

To evaluate the tracking cost that involves only the observation error further modifications are made and the tracking cost is observed by formatting the Q<sub>MDP</sub>. This is because data association errors do not allow the tracking cost to be written as a sum of tracking costs for each sensor. To do this the

per-sensor Bellman equation is written as the summation of tracking cost, energy cost and future cost if the sleep time is given. The minimization values of all the terms represent the optimal cost.

The name First Cost Reduction Policy comes from the fact that since the sensor comes awake only when the expected tracking cost exceeds the expected energy cost when awakened. FCR policy does not have the tracking problems that are associated with the Q<sub>MDP</sub> policy. Because the objects move independently, the marginal distributions in the absence of observations also evolve independently. FCR policy in case of sub optimal MTT algorithms are used to determine the marginal for each object.

### 3.3 All Awake policy

All Awake policy is a Q<sub>MDP</sub> like policy that makes the following assumptions instead of assuming the state is known since the lower bounds that arise from the Q<sub>MDP</sub> is likely to be lose when the data association errors dominate in case of tracking. Accordingly in [9]

- 1) At a current time step (i.e.) after selecting sleep times all sleeping sensors will be allowed to make observations accruing no energy cost.
- 2) At future time steps, the distribution for the object location will evolve as if all sensors are awake.

This gives rise to the term All Awake (AA policy). By making use of this AA Assumption we will have perfect observations. However, this does not imply perfect knowledge of the state due to the presence of data association errors. Note that since it is assumed more information than is actually available, the AA assumption does yield a lower bound on optimal performance.

Here the probability that a particular value of  $x_k$  will be the result of allowing all sensors to wake up in the current time step. Thus using the concept of dynamic programming the expected tracking cost at time  $k$  is found out.

## 4. SIMULATION RESULTS

The performance of the policies is discussed after simulating. In each simulation run, the objects were initially placed at the center of the network and after the locations of the objects made known to each sensor. This initial condition would not occur in practice, but was used for simplicity as multi target tracking in [15]. Alternatively, a partially unknown initial state was observed but the trends described in the simulation results below are expected to remain unchanged. A simulation run concluded when all objects have left the network. Results of many simulation runs have been averaged to compute an average tracking cost and an average energy cost. To allow for easier interpretation of the results, then the normalized costs are obtained by dividing by the expected time for a simulation run. The normalized costs are referred as the cost/unit time. From the runs, it was concluded that the Q<sub>MDP</sub> solution is awake only when few sensors are awake. Here only observation errors dominate.

- 1) The lower bound due to the AA assumption is tight only when many sensors are awake.
- 2) The Q<sub>MDP</sub> policy performs best when only a few sensors are awake.
- 3) AA policy performs best when many sensors are awake.

The FCR policy is the worst-performing policy. The difference between the FCR policy and the other policies, while never especially large in terms of the trade-off curves, shrinks as data association errors become small.

#### 4.1 Performance of suboptimal Sleeping policies in Centralized architecture

Tracking results indicate that using laterally inhibited distributed tracking is currently about as efficient as centralized tracking in network resource consumption. Lateral inhibition is simpler computationally and scales better.

In large-scale networks it is likely to be the better alternative. Work still needs to be done on optimizing the  $Q_{MDP}$  as well as the AA policies. From the results it was found that the solutions reached are much closer to the optimal solutions which are found in literature.

We illustrate the performance of these policies by the object movement in a Network which is parameterized by the scalar  $\alpha$ , which may range from 0 to 1 and for various values of  $\alpha$  such as 0.55, 0.75, 0.95 the policy curves are plotted and the results according to their bounds are examined. The Maximum value of U i.e.)  $U_{max}$  taken to be 50.

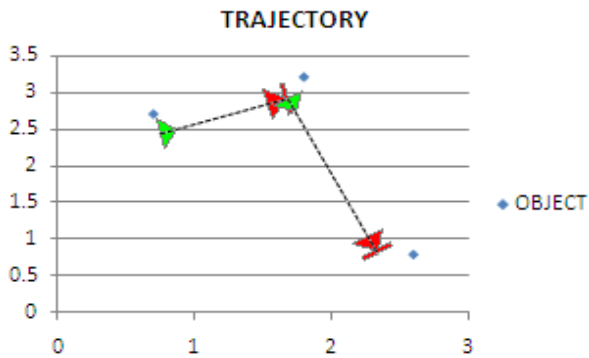


Figure: 2 Distribution trajectories of an object moving across a network

- In figure.2 it is known that one object is located at position 3 and one is located at position 6.
- For each object we select the estimated object location from among the locations where a sensor is awake.
- From these locations, we select the one with the largest value of the marginal distribution for that object.

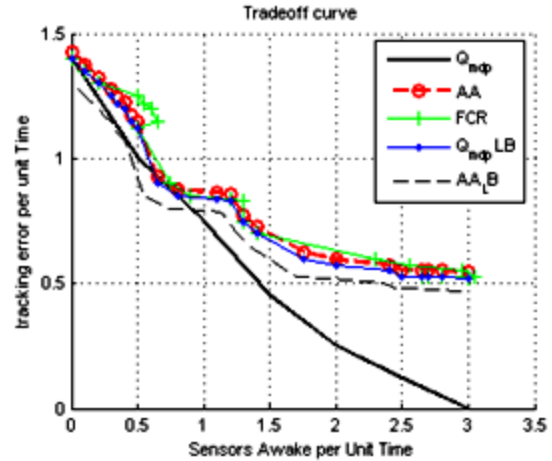


Figure 3: Trade off curve for QMDP, FCR, AA policies when  $\alpha=0.55$

From the observation in centralized environment it was inferred that the lower bounds still holds good and the performance of AA is still the same as centralized one. The FCR is same as before where the implementation can be accompanied by further duty cycling policies in order to improve the performance. The analysis is done for various values of  $\alpha$ . The following figure 4 depicts the trade-off curve when the value of  $\alpha$  is 0.75.

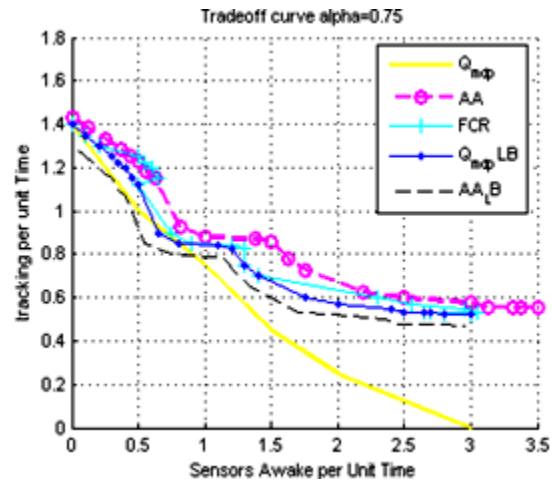
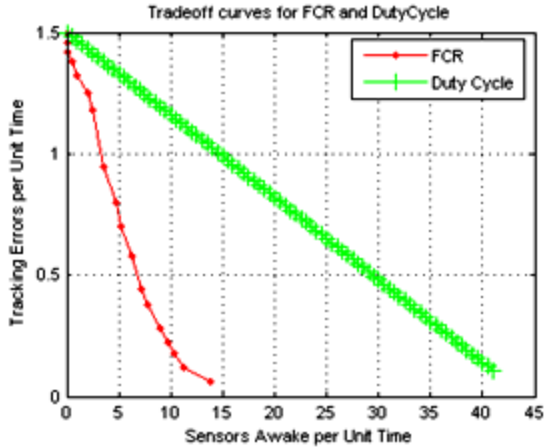


Figure 4: Trade off curve for QMDP, FCR, AA policies when  $\alpha=0.75$

The FCR policy comparatively performs well than other policies in case of a larger network.

The following Figure.5 shows the performance of FCR policy with the duty cycle policy since the lower bound of FCR is fair to be compared with duty cycling where each sensor comes awake with some fixed probability at each time step. The duty cycle policy value is chosen by varying the movements of sensor between 0 and 1. Since the bound of FCR is lower than duty cycle it considerably outperforms the duty cycle policy.

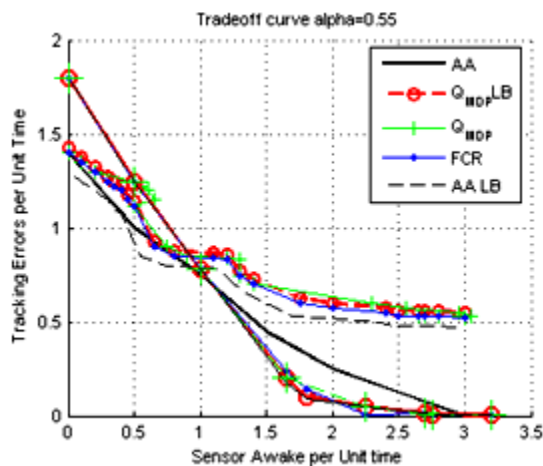


This analysis is repeated for the same values of  $\alpha=0.95, 0.55$  and  $0.75$  as proceeded in the centralized architecture. This is achieved by making changes over the value of  $U_{k1}$  which is the sleep time supplied from the central controller. Also the observation available to the central controller  $Z_k$  is not available

**Figure 5: Trade off curve for FCR and duty cycle policies for changing objects movement from -2 to +2 in a larger network**

#### 4.2 Performance of suboptimal Sleeping policies in Decentralized architecture

The primary motto of this study is to analyze the performance in case of distributed or a decentralized architecture. The reason for choosing the decentralized architecture is that it potentially introduces single point of failure. Also sending data to all nodes irrespective of the trajectory introduces unusual latency as well as synchronization issues [16].



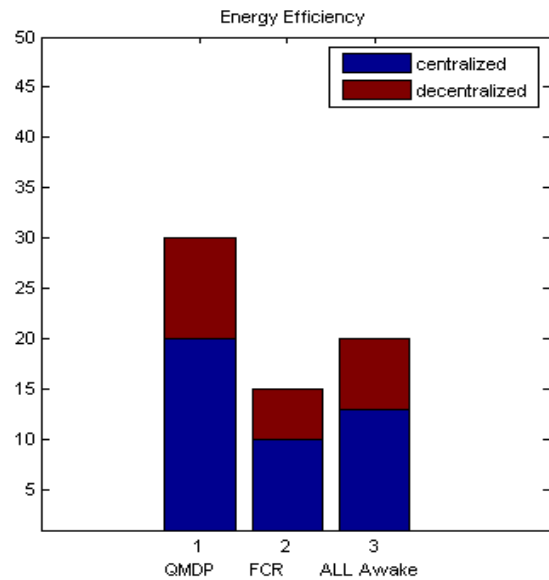
**Figure 6: Trade off curve for QMDP, FCR, AA policies when  $\alpha=0.55$  in Centralized Vs Decentralized architectures**

and the total input available to the control unit at time  $k$  depends only on the value of  $\mu_k$ . This  $\mu_k$  value acts as the function of the information value  $I_k$  in order to yield the sleeping policy at time  $k$ . Since the results of  $\alpha=0.95, 0.75$  is not clear the simulation results of all the policies at  $\alpha=0.55$  is shown in Figure 6.

From the results of the Fig.6 the bounds of Decentralized architecture is found lower than the centralized which infers that the Distributed architecture performs better with suboptimal policies than centralized ones.

#### 4.3 Comparison of suboptimal Sleeping policies in Centralized and Decentralized architecture

The Figure.7 is the Comparison graph showing the energy efficiency in both the architectures



**Figure 7: Energy efficiency in centralized Vs Decentralized architectures**

The decentralized architectures are conserving 10-15% of total energy conservation and said to be more efficient compared to the centralized one

### 5. CONCLUSION

In this work, methodologies are framed to formulate methods other than the wake-up channel approach. Since the cost of establishing a channel between the nodes are very much higher than the tracking cost in previous approach, an improved policy to optimize power consumption is formulated. Also, the relevant Data association problem in case of MultiobjectTracking is solved by some dynamic programming methodologies and a generic solution is found out. More efficient policies such as POMDP that predict the dynamic state changes are improved. Experiments are carried even on decentralized schemes of multiple object tracking and its energy efficiency, error rate are

compared with the existing centralized schemes and proved to be more efficient. Our future work includes experimentation of decentralized schemes on Power Aware Multiple Object Tracking (PAMOT).

## 6. ACKNOWLEDGMENT

The authors are thankful to Dr. P. Suyambu, Chairman, PSN Group of Institutions for his constant support and encouragement.

## 7. REFERENCES

- [1] B. Gholamzadeh, and H. Nabovati, "Concepts for Designing Low Power Wireless Sensor Network", *World Academy of Science, Engineering and Technology*, issue: 45, pp. 559–565, 2008.
- [2] Zheng Zeng-wei, Wu Zhao-hui, Lin Huai-zhon, Hangzhou, "An event-driven clustering routing algorithm for wireless sensor networks", *proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1802–1806, 2004.
- [3] G. Chen, S. Hanson, D. Blaauw, Sylvester, "Circuit Design Advances for Wireless Sensing Applications", *proceedings of the IEEE*, vol. 98, issue: 11, pp. 1808–1827, 2010.
- [4] D.Petrovic, R.C.Shah, K.Ramchandran, Rabaey, "Data funneling: routing with aggregation and compression for wireless sensor networks", *proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 156–162, 2003.
- [5] Xiaoyan Yin, Xingshe Zhou, Yanjing Lei and Shining Li, "An Energy Efficient Scheduling Scheme for Target Tracking Wireless Sensor Networks", *proceedings of International Conference on Intelligent Control and Information Processing*, pp. 506–511, 2010.
- [6] Venugopal V. Veeravalli and Jason A. Fuemmeler, "Efficient Tracking in a Network of Sleepy Sensors", *International Conference on Acoustics, Speech and Signal Processing*, vol. 13, pp. 6003–6006, 2006.
- [7] J. Liu, M. Chu, and J. E. Reich, "Multitarget tracking in distributed sensor networks," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 36–46, May 2007.
- [8] H. Yang and B. Sikdar, "A protocol for tracking mobile targets using sensor networks," *proceedings of 1<sup>st</sup> IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 71–81, May 2003.
- [9] Aysegül Alaybeyoğlu, Orhan Dagdeviren, "A Distributed Wakening Based Target Tracking Protocol for Wireless Sensor Networks," *proceedings of the 4<sup>th</sup> International Symposium Information Processing in Sensor Networks*, pp. 246–253, April 2005.
- [10] Jason A. Fuemmeler and Venugopal V. Veeravalli "Energy Efficient Multi-object tracking in Sensor Networks", *IEEE Transactions on Signal Processing*, vol. 58, Issue. 7, July 2010
- [11] Y. Xu, J. Winter, and W.-C. Lee, "Prediction-based strategies for energy saving in object tracking sensor networks," *proceedings of IEEE International Conference on Mobile Data Management*, pp. 346–357, 2004.
- [12] L. Yang, C. Feng, J. W. Rozenblit, and J. Peng, "A multi-modality framework for energy efficient tracking in large scale wireless sensor networks," *proceedings of IEEE International Conference on Networks, Sensor and Control*, pp. 916–921, April 2006.
- [13] J. A. Fuemmeler and V. V. Veeravalli, "Smart sleeping policies for energy efficient tracking in sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 2091–2101, May 2008.
- [14] J. Liu, M. Chu, and J. E. Reich, "Multitarget tracking in distributed sensor networks," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 36–46, May 2007.
- [15] Yeow, Chen-Knong, Wai-choong Wong, "Energy Efficient Multiple Target Tracking in Wireless Sensor Networks", *IEEE Transactions on Vehicular Technology*, Vol: 56, Issue: 2, pp: 918 – 92, 2007.
- [16] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed target classification and tracking in sensor networks," *Proc. IEEE*, vol. 91, no. 8, pp. 1163–1171, Aug 2004.