Mathematical Analysis of Optimal Tracking Interval Management for Power Efficient Target Tracking Wireless Sensor Networks

H. Jamali-Rad*, B. Abolhassani* and M. Abdizadeh**

Abstract: We study the problem of power efficient tracking interval management for distributed target tracking wireless sensor networks (WSNs). We first analyze the performance of a distributed target tracking network with one moving object, using a quantitative mathematical analysis. We show that previously proposed algorithms are efficient only for constant average velocity objects; however, they do not ensure an optimal performance for moving objects with acceleration. Towards an optimal performance, first, we derive a closed-form mathematical expression for the estimation of the minimal achievable power consumption by an optimal adaptive tracking interval management algorithm. This can be used as a benchmark for energy efficiency of these adaptive algorithms. Second, we describe our recently proposed energy efficient blind adaptive time interval management algorithm called Adaptive Hill Climbing (AHC) in more detail and explain how it tries to get closer to the derived optimal performance. Finally, we provide a comprehensive performance evaluation for the recent similar adaptive time interval management algorithms using computer simulations. The simulation results show that using the AHC algorithm, the network has a very good performance with the added advantage of getting 9 % closer to the calculated minimal achievable power consumption compared with that of the best previously proposed energy efficient adaptive time interval management algorithm.

Keywords: Wireless Sensor Networks (WSNs), Distributed Target Tracking, Tracking Interval, Network Lifetime, Power Efficiency.

1 Introduction

Target tracking is one of the most important applications of distributed wireless sensor networks (WSNs). A target tracking wireless sensor network is a sensor network that monitors and tracks moving objects in the area under its radio coverage [1]. Since sensors are powered by small batteries, with no chance of replacing or recharging them, they have limited lifetime, which makes the tracking network to fail. This fact clarifies the need for an appropriate mechanism to decrease the power consumption in the network.

The performance of a tracking network is deeply dependent upon "*Tracking Resolution*" or "*Tracking Interval*", which is the time interval between two consecutive sensing operations. It is obvious that if the tracking interval decreases, the miss probability of the object decreases too; however, this improvement in the miss probability occurs at the expense of more power consumption due to using smaller tracking intervals. To meet both target values for the miss probability and network lifetime, a predictive mechanism can be used to adjust the tracking interval. Prediction-based methods try to wake up only those sensors that are in the neighboring area of the moving object, while all other sensors can stay in sleep mode that leads to very low power consumption by the network. Pioneering idea for the prediction-based monitoring in sensor networks, using past reading history and spatial and temporal relationships of readings from sensors, is proposed by Goel and Imielinski [2]. Comprehensive studies of power consumption on WINS nodes developed by Rockwell and UCLA show that long distance transmissions dominate the energy consumption of wireless sensor networks (WSNs) [3]; hence, in further

Iranian Journal of Electrical & Electronic Engineering, 2012.
Paper first received 14 March 2011 and in revised form 10 July 2012.
* The Authors are with the Department of Electrical Engineering, Iran University of Science and Technology (IUST), Tehran, Iran.
E-mails: abolhassani@iust.ac.ir, h.jamali@ee.iust.ac.ir.
** The Author is with the Department of Electrical Engineering, Sharif University of Technology (SUT), Tehran, Iran.
E-mail: abdizadeh@ee.sharif.ir.

studies a series of local computations are done in sensor nodes to avoid long distance transmissions [4]. However, aforementioned studies do not address the issue of scalability for coordinating a wireless sensor network for the purpose of target tracking.

A Distributed Prediction-based Tracking (DPT) mechanism for managing and coordinating a tracking sensor network is proposed in [5]. The objective function of the proposed mechanism is to minimize the total power consumption of the tracking network. Due to the uncertainty and unpredictability of real-world moving objects' movement patterns, the tracking algorithm is required to adapt to real-time changes of velocities and directions of a moving target. Hence, adaptive mechanisms should also be added to the prediction capability of the target tracking networks. This idea was the origin of further studies in the literature [6-15]. In [6], an adaptive framework to reduce communication in the context of information collection for mobile target tracking is explored. In [7], a protocol for Predictive Accuracy-based Tracking Energy Saving (PATES) is well developed to conserve energy of WSNs. As well, a quantitative analytic model is proposed in [7] for calculating the optimal tracking interval for a moving object with approximately constant average velocity (with no acceleration) movement pattern.

From one point of view, further research studies in adaptive target tracking are divided into two basic directions. In the first, enhancement of the complexity and capability of tracking is of major concern; besides, power efficiency is considered as well. To be more specific, the first direction includes the application of more complex functions for the tracking network such as multiple-object tracking [8-10]. Specifically, the application of extended Kalman filtering (EKF) [8] or $Q(\lambda)$-learning [9] is considered to improve the quality of tracking. In the second research direction, however, the power efficiency is of major concern, while a simple target tracking network (in most cases for practical moving objects) is considered [6-7, 11-15]. Our proposed algorithm lies in the second direction of research studies.

Since the optimal tracking interval in [7] is not accurate if the average velocity of movement varies over time (movements with acceleration), in [11] an energy efficient adaptive tracking algorithm called Predict-and-Mesh (PaM) is proposed to consider movements with acceleration. The proposed algorithm consists of a prediction model and a failure recovery process. In [12], a novel algorithm is proposed that can significantly improve the performance in the PaM. But, the proposed algorithm in [12] is not blind and requires prior knowledge about the network infrastructure to be trained with a lookup table, which is not available in most practical situations. To remove this constraint, in [13] a new blind adaptive prediction-based tracking algorithm (called AEC) is proposed, which significantly outperforms the performance of the PaM in a similar tracking scenario. In [14], the drawbacks of the conventional PaM are studied and two novel modifications in its basic modules are proposed to improve its functionality and achieve a better power efficiency. The proposed modified algorithm is called Modified PaM (MPaM). The simulation results illustrated a considerable improvement in comparison with the conventional PaM and the AEC.

In [15], the AHC algorithm is proposed and briefly explained. It is shown that AHC also improves the power consumption over the MPaM. However, the performance of this algorithm is compared with a rough (linear) estimation of optimal achievable performance. Now, the question is how much more improvement can be achieved? Or is there a lower bound or minimal power consumption (we call it $P_{min\_overall}$) for the movements with acceleration? In this paper, first, we consider the above question to develop a framework in order to derive a mathematical equation for the minimal achievable power consumption ($P_{min\_overall}$) due to the use of an optimal adaptive algorithm for the real-time management of tracking interval values when compared with non-adaptive algorithms, both for a movement pattern with acceleration. The mathematical derivations, here, are based on a quadratic interpolation of tracking time interval versus the velocity of the object, which significantly improves the accuracy of closed form formulas in comparison with linear interpolation in [15]. Second, to highlight the achievable performance of a dynamic tracking interval management algorithm, we describe the performance of Adaptive Hill Climbing (AHC) algorithm [15] and show how this algorithm results in significantly decreasing the network power consumption and getting closer to its minimal value ($P_{min\_overall}$). Meanwhile, we illustrate that using AHC a smaller miss-probability and lower power consumption is achieved as well (when compared to best existing similar adaptive algorithm MPaM [14]).

The rest of the paper is organized as follows. In Section 2, the network model and its components are described. In Section 3, a closed-form formula for the minimal achievable power consumption ($P_{min\_overall}$) of the movements with acceleration is derived. In Section 4, operational mechanism of AHC algorithm is described and analyzed to show how it gets closer to the derived minimal power consumption. In Section 5, using computer simulations, the performance of AHC is evaluated with respect to the newly derived optimal performance and also compared with those of four existing similar prediction-based tracking algorithms (i.e. the PATES, the PaM, the AEC and the MPaM). Finally, this paper is concluded in Section 6.

2 Network Model

In this section, we describe the network model and discuss about our proposed Adaptive Prediction-based Tracking (APT) scheme.

2.1 Prediction-Based Tracking Sensor Netoworks

In order to track a moving object, the network should provide enough coverage and the ability to localize the object. We assume that the sensors used in the network are able to estimate the distance D and the direction θ of the moving object (DOA sensors [11, 14, 15]). Thus, the density of sensor nodes should be large enough so that at any given time and location in the two-dimensional area under its coverage, at least one sensor is able to sense the moving object with its normal sensing range. Since the number of sensor nodes is large, the distribution of the number of sensor nodes in any given area A is Poisson with rate λA in which λ nodes/m^2 is the sensor nodes density [5]. Therefore, to have at least one sensor with its normal sensing range in any given point with probability 0.99, by substituting a value for r as the sensing range, the required sensor node density λ can be calculated by solving Eq. (1), given below:

$$0.99 = \sum_{i=1}^{\infty} \frac{e^{-\lambda \pi r^2} (\lambda \pi r^2)^i}{i!} = 1 - e^{-\lambda \pi r^2}. \qquad (1)$$

Sensor nodes have different power consumptions in transmit, receive, idle, sleep (also called power down) and sense modes [3, 4]. In the sleep mode the power consumption of the sensor nodes reduces to a minimum level, while the transmit mode requires the maximum value of power consumption. So, more number of long transmissions increases the power consumption and reduces the network lifetime. To increase the network lifetime, predictive algorithms use the past readings at sensor nodes and process them to predict the location of the moving object for the next tracking interval. Then using this prediction, the cluster head awakes only those sensors which are supposed to have the object in their sensing range. Different prediction models are discussed in [4].

2.2 Power Consumption and Tracking Model

We denote the power consumption in different operational modes, namely, transmit, receive, idle, power down and sense by W_T, W_R, W_I, W_D and W_S, respectively. Then, the overall energy consumption of i-th sensor during T seconds (Total tracking duration) can be written as

$$\begin{aligned} &e_i = W_T.t_T + W_R.t_R + W_I.t_I + W_D.t_D + W_S.t_S \\ &i = 1, 2, ..., N \end{aligned} \qquad (2)$$

where t_T, t_R, t_I, t_D and t_S , respectively, represent the time periods required for the mentioned working modes during T seconds. It is notable that $T = t_T + t_R + t_I + t_D + t_S$. Thus, assuming that the tracked object has been present in the sensing area of the network for the interval T, the total network energy consumption can be given by

$$E = \sum_{i=1}^{N} e_i. \qquad (3)$$

Also, object's movement pattern and the prediction model to predict the next location of the moving object can be simply described as follows. The instantaneous velocity of the object is considered to vary between zero and V_{max} with an instantaneous acceleration $|\alpha| \in [0, \alpha_{max}]$. The object changes its direction (θ_v) randomly over [-π, π]. As it is clear from the movement pattern, it is not a fully random movement pattern. However, by selecting different values for V_{max} and α_{max} it can be mapped to different movement patterns appropriate for a wide variety of practical moving objects e.g. a human or a vehicle. Such practical movement patterns are of special interest in the literature and are also considered in [6, 7, 11-15].

As mentioned earlier, in this study, we focus on the importance of an energy efficient tracking interval management algorithm (power efficiency instead of a complex tracking scheme). Therefore, to reduce the complexity of our calculations, we use a simple averaging prediction model [4, 5, 7, 11-15] wherein the next location of the moving object is predicted as follows. Assume that the location of the object at present and next tracking intervals (t_i and t_{i+1}, respectively) are denoted by x_i and x_{i+1}, respectively. Then, by estimating the instantaneous velocity using

$$\begin{cases} |v_i| = \dfrac{\sqrt{(x_{i,1} - x_{i-1,1})^2 + (x_{i,2} - x_{i-1,2})^2}}{\Delta t}, \\ where \quad \Delta t = t_i - t_{i-1}, \end{cases} \qquad (4)$$

and knowing the direction ($\theta_{v,i}$) of the moving object (from sensor readings) during Δt, the next location of the moving object can be calculated using

$$\begin{aligned} &x_{i+1} = x_i + v_i.\Delta t + w_{i+1}, \\ &v_i = |v_i| \angle \theta_{v,i}. \end{aligned} \qquad (5)$$

In practice, the sensor measurements are imprecise. Hence, w_{i+1} denotes the process noise (e.g. position measurement error due to unknown acceleration). Then assuming an instantaneous acceleration α, the velocity vector can be estimated as follows

$$v = v_0 + \alpha.\Delta t, \qquad (6)$$

where v_o stands for the initial value of the velocity [11-15].

2.3 Adaptive Prediction-Based Tracking (APT)

The key idea in the prediction-based tracking schemes is that the lifetime of a sensing system (e.g. sensor networks) can be extended by the help of a set of prediction-based activation mechanisms. Making use of such mechanisms, most sensors will be kept in sleep mode for conserving energy until they are triggered by moving object to become active. Another solution for minimizing the power consumption besides the mentioned policy is to dynamically modify the tracking interval (S) based on some given parameters. Now, with attention to what was described about prediction-based tracking sensor networks (subsection 2.1) and by taking the idea from [5], [7] and [11-15], we propose a new Adaptive Prediction-based Tracking (APT) scheme for WSNs. The novelty of the proposed APT scheme returns to a new adaptive tracking interval management algorithm, which works in one of APT modules. The new APT scheme comes into the play after detecting the object by sensor nodes and has the following two basic modules: "*Prediction Module*" and "*Recovery Module*".

1) Prediction Module: Once the target is detected, the active sensor (current node) will predict the next location of the object using Eq. (5). Then it will activate another proper sensor node as the next current node (collaborative prediction). To reduce the power consumption, current node estimates the time required for the object to move out of its sensing range (escape period). Using this knowledge, it is able to modify the tracking interval to avoid unnecessary activations while not losing the object. Hence, an important consideration in the design of power efficient distributed target tracking sensor networks is an efficient algorithm capable of dynamic modification of tracking interval. We consider this issue in Section 4.

2) *Recovery Module:* The predictions are always inexact due to the process noise in Eq. (5) and probable blind coverage areas, which can make the tracking system to lose the object. Hence, a recovery module is required to compensate for this loss. If the predicted location is very different from the actual location of the object sensed by the current node, the object may be out of coverage of the next current node. In this case, the object is lost. Therefore, some other surrounding sensor nodes should be awakened to recapture the lost object (Recovery process). This process can be done in two or more stages, but, we consider only two stages. In the first stage (*1st_Recovery*), some of the nearest neighboring sensor nodes are awakened to find the actual location of the object. If the object is found, the task of the Recovery Module completes and no extra sensor node is required to localize the object. If the object is not found, a more assertive approach is applied and in this stage (*2nd_Recovery*) a lot of other neighboring sensor nodes should be awakened and with a high probability, the object will be found.

3 Mathematical Analysis of Optimal Tracking Interval Management

As we described, we consider practical movement patterns for the object wherein the object (e.g. a human or a vehicle intruding a covered area) may not move like fully random movement patterns.

However, in practice, an intruding object also does not necessarily keep its average velocity constant as studied in some of previous research works in the literature. Analytical and simulation results in [7], reveals that for a movement pattern with approximately constant average velocity, the optimal tracking interval to be used for management of the tracking network for minimal power consumption is approximately a fixed value. However, for the movements with acceleration, a fixed optimal value does not exist anymore (the optimal value will be time variant) and the tracking interval should be dynamically modified according to variations in the objects' movement pattern. Hence, a fixed value does not result in the minimal total power consumption for the tracking network. This motivated us to develop a mathematical framework to analyze the achievable minimal power consumption for target tracking networks based on adaptive modification of tracking interval for the objects moving with acceleration. It is worth noting that, since practical situations are of special interest in our study, a blind adaptive tracking interval modification is to be proposed, which (different from the previous research study in [12]) requires no prior knowledge about the network structure.

As described earlier, the power consumption performance of the explained network model deeply depends upon an optimized policy for modification of tracking interval values. As described in [7], for an approximately constant average velocity a minimal value for the power consumption exists. Therefore, the power consumption function can be approximated by a convex-shaped function. For the sake of simplicity, we approximate such convex-shaped functions by a second order polynomial (quadratic polynomial) with a minimum at S_{opt} for a given v. Hence, for a movement pattern with a fixed average velocity the power consumption of the network can be approximated by

$$P = f(S,v) = A(S - S_{opt})^2 + B, \quad if\ E(v) = const, \tag{7}$$

where $E(.)$ represents statistical expectation.

In general, A, B and S_{opt} ($A(v)$, $B(v)$ and $S_{opt}(v)$) are positive values and also functions of velocity of the moving object, which varies over time for movement patterns with acceleration. Hence, to minimize (7), S should be equal to S_{opt}, which is a function of v. In this case there is minimal power consumption, i.e.,

$$P_{\min} = B(v). \tag{8}$$

By defining P_{total} as follows

$$P_{total} \stackrel{\Delta}{=} \frac{1}{T}\int_0^T P\,dt, \tag{9}$$

and by substituting the R.H.S of Eq. (7) for P in Eq. (9),

$$P_{total} = \frac{1}{T}\int_0^T A(v)(S - S_{opt}(v))^2 dt + \frac{1}{T}\int_0^T B(v)dt. \tag{10}$$

Hereafter, we call the second term of Eq. (10) $P_{min\_overall}$, which is achieved when the condition $S = S_{opt}$ is continuously met during the tracking process. However, the proposed algorithms in [11-14] are unable to keep the first term equal to zero, instantaneously. To solve this problem, a proper adaptive algorithm should be proposed to dynamically modify the instantaneous value of tracking interval. The more precise and faster the proposed algorithm in updating the tracking interval value to its instantaneous optimal value, the more P_{total} gets closer to $P_{min\_overall}$ and therefore to the optimal performance. Since the first term of Eq. (10) is always positive, then

$$P_{total} \geq \frac{1}{T}\int_0^T B(v)dt = \frac{1}{T}\int_0^T P_{\min} \, dt = P_{\min\_ \, overall} \, . \quad (11)$$

From both Eq. (8) and Eq. (11), it can be concluded that $P_{min\_overall}$ is constant for the case of movement patterns with constant average velocity and leads to a fixed optimal tracking interval during the tracking process. This is also consistent with our previous explanations. However, in practice, there are cases wherein objects move with acceleration. For the movements with acceleration, P_{total} during a given time duration can be considered as a combination of successive quadratic functions with different minima. Each of the mentioned successive quadratic functions is considered to be the corresponding power consumption of a very small time interval. Intuitively, we can consider the average of these quadratic functions as the power consumption performance of the algorithms like the PATES in [7], which assume a constant average velocity during the given time span for a movement pattern with acceleration. We illustrate this concept using the following explanations.

We consider a total tracking duration of 6 seconds and divide this time duration into six shorter time intervals of 1 second. We can consider the average velocity of moving object being constant in each of these six intervals of 1 second. Therefore, if we illustrate the power consumption of the network during each of these shorter intervals, there will be a convex function with a given minimum corresponding to S_{opt} in that tracking duration. Fig. 1 shows the simulation results for the six successive short time intervals with six convex functions. The bold dashed line represents the aforementioned average power consumption during the overall time duration of 6 seconds. As described earlier, this average is the minimal achievable power consumption of the tracking algorithms (like PATES), which ignore the instantaneous variations of the object's average velocity during the tracking process, when tracking a moving object with acceleration.

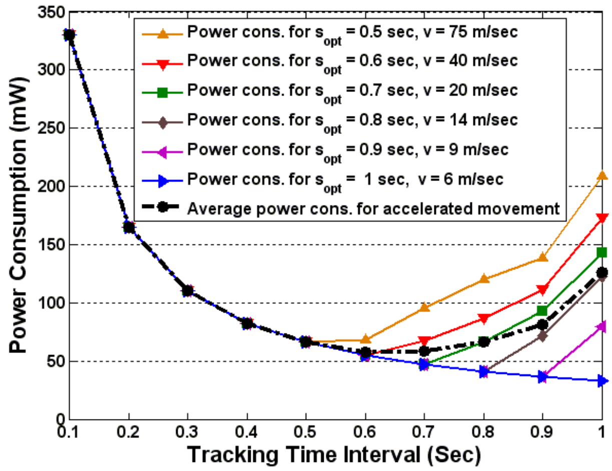

Fig. 1 Power consumption functions against S for diff. values of v.

From Fig. 1, in some of the short time intervals we can achieve smaller total power consumption for the mentioned movement patterns, which is impossible to achieve by algorithms like PATES (the minimum of the bold dashed line). In the following, we mathematically prove the above explanations and also derive an equation for minimal achievable power consumption. To this aim, we assume that the successive minimum values for the power consumption functions (S_{opt} values), in the concerned span, can be fitted to a quadratic function of v (with three constants β, γ and ζ)

$$S_{opt} \approx \beta . v^2 + \gamma . v + \zeta, \qquad 0.2 < S_{opt} < 1. \quad (12)$$

To show the feasibility of this assumption, we check our assumption with the simulation results in Fig. 1. The S_{opt} values versus velocity according to data of Fig. 1 is shown in Fig. 2.

As it is clear from the figure, the quadratic interpolation for the S_{opt} (as a function of v) is acceptable and its mean squared error (MSE) is about 0.0014 and can be calculated as $MSE = \frac{1}{N}\sum_{i=1}^{N}(S_{opt_i} - S_{opt\_fitted_i})^2 \approx 0.0014.$

It is notable that a linear interpolation of S_{opt} values versus velocity could also be used in our mathematical analysis [15]. However, the quadratic interpolation leads to more accurate results (due to providing much lower MSE) in the following derivations. The result of the following mathematical derivations for linear interpolation of S_{opt} versus velocity can be found in [15]. In Appendix A, we show that the rough estimations in [15] can also be validated by doing simplifications in the proposed and more accurate derivations in this paper. As described earlier about Eq. (10), the optimal

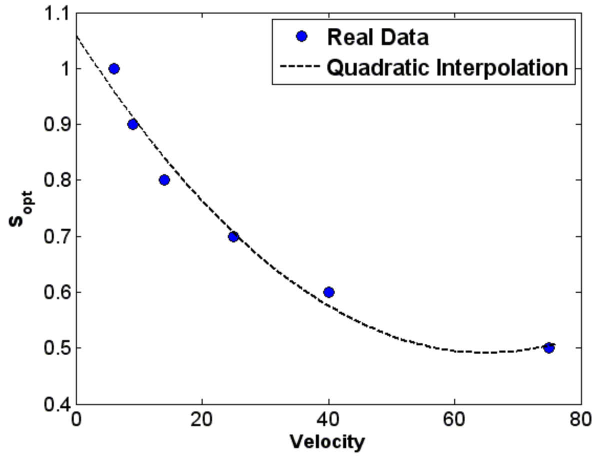

Fig. 2 Quadratic interpolation of Sopt values.

tracking interval management continuously updates the tracking interval to its instantaneous optimum value,which results in the minimal power consumption. Since $P_{min\_overall}$ indicates this optimum performance, it is constant and independent of tracking interval values.

Hence, only the first term of Eq. (10) is dependent of S and indicates the power consumption difference which can be achieved by those algorithms which ignore modifying the tracking interval according to variations of the average velocity (i.e. non-adaptive algorithms like PATES [7]) and the optimal tracking interval management algorithm. Therefore, using Eq. (10) and Eq. (12) we can write

$$P_{total} - P_{\min\_overall} = \frac{1}{T}\int_0^T A(v)(S - S_{opt}(v))^2 dt$$
$$= \frac{1}{T}\int_0^T A(v)(S - [\beta \times v^2 + \gamma \times v + \zeta])^2 dt. \quad (13)$$

In Fig. 1, it can be shown that for the values of S over [0.2, 1], variations of $A(v)$ for the successive time intervals is negligible. Hence, for the sake of simplicity we write

$$P_{total} - P_{\min\_overall} = \frac{A}{T}\int_0^T (S - [\beta \times v^2 + \gamma \times v + \zeta])^2 dt. \quad (14)$$

To simplify the analysis, α can also be considered to be a constant value. Hence,

$$P_{total} - P_{\min\_overall} =$$
$$\frac{A}{T}\int_0^T (S - [\beta \times (v_0 + \alpha \times t)^2 + \gamma \times (v_0 + \alpha \times t) + \zeta])^2 dt, \quad (15)$$

where, A denotes a constant related to the quadratic interpolation of convex-shaped power consumption functions (as in Eq. (7)), β, γ and ζ are the coefficients of quadratic interpolation of S_{opt} values (in Eq. (12)), α is the average value of acceleration and T represents the total time duration of tracking process. After calculating the above integral, we have,

$$P_{total} - P_{\min\_overall} =$$
$$A(x^2 + xyT + \frac{y^2}{3}T^2 + \frac{2xz}{3}T^2 + \frac{yz}{2}T^3 + \frac{z^2}{5}T^4) \quad (16)$$

where

$$x = S - [\beta \times {v_0}^2 + \gamma \times v_0 + \zeta],$$
$$y = -2\alpha \times \beta \times v_0 + \gamma \times \alpha, \quad (17)$$
$$z = -\alpha^2 \times \beta.$$

Knowing that $P_{min\_overall}$ is independent of S, the minimal achievable power consumption value of algorithms, which ignore the effect of variable average velocity in calculating the optimum value of tracking interval (non-adaptive algorithms) can be found by setting the derivative of Eq. (16) with respect to S equal to zero.

$$\frac{\partial}{\partial S}\{P_{total} - P_{\min\_overall}\} = \frac{\partial}{\partial x}\{P_{total} - P_{\min\_overall}\} \times \frac{\partial x}{\partial S} = 0\,. \quad (18)$$

Also, we know that

$$\frac{\partial x}{\partial S} = \frac{\partial}{\partial S}\left(S - [\beta \times {v_0}^2 + \gamma \times v_0 + \zeta]\right) = 1. \quad (19)$$

Hence,

$$\frac{\partial}{\partial x}\{P_{total} - P_{\min\_overall}\} = 0 \Rightarrow \; x = -\frac{1}{2}(yT + \frac{2}{3}zT^2). \quad (20)$$

By substituting x from Eq. (20) into Eq. (16) and after simplifications, the minimal power consumption can be calculated as follows

$$\min(P_{total}) = P_{\min\_overall} + \frac{AT^2}{60}\left[5y^2 + 10yzT + \frac{16}{3}z^2T^2\right], \quad (21)$$

where, A , y, z and T are described earlier. The $\min(P_{total})$ is the minimum power consumption that a non-adaptive algorithm can achieve, which is $AT^2/60[5y^2 + 10yzT + 16/3z^2T^2]$ Watts larger that the minimal achievable power consumption ($P_{min\_overall}$). The faster and the more precise the modification of the tracking interval is, the closer the tracking algorithm gets to $P_{min\_overall}$.

4 Performance Analysis of Adaptive Hill Climbing (AHC)

In this section, we analyze the performance of our recently proposed adaptive time interval management algorithm in [15] with respect to the derived optimal performance in Section 3.

To modify the tracking interval, the AHC algorithm considers the distance error (*Error*) between the measured and predicted locations of the object. So, this algorithm has an indirect cognition of the network power consumption. A sequence of actions: sense-comparison-modification-sense ensures that the network keeps tracking the moving object with the near minimal power consumption value. The idea behind the AHC algorithm is to update the tracking interval, from the *Old_Step* to *New_Step* considering the distance *Error* values, where *Error* is the difference between the measured and predicted locations of the moving object. The AHC starts its search from initial direction (*Direc._Flag*) and tracking interval (*Old_Step*), and modifies them after each tracking interval. To describe this idea, we propose the following equation for updating the tracking interval

$$New\_Step - Old\_Step = Dirc.\_Flag \times Step\_Modif., \tag{22}$$

where *Step_Modif.* is the adaptation factor for the AHC algorithm. The value of step modification (*Step_Modif.*) can be kept constant or can be updated during the tracking process. The latter provides the algorithm with a faster performance and therefore, we use the following equation to update it.

$$Step\_Modif. \leftarrow Step\_Modif. / Smoothing\_Factor, \tag{23}$$

where *Smoothing_Factor*, as it is self-explanatory, is used to smoothly update the *Step_Modification*. To avoid from the divergence of the algorithm, the *Step_Modification* is kept greater than a minimum value (*Minimum_Step_Modification*), which is the minimum accuracy required for the modification of tracking interval values. Small values for *Minimum_Step_Modification* makes the algorithm slow in modifying the tracking interval, and consequently degrading its performance for real-time applications. However, large values for *Minimum_Step_Modification* leads to inaccurate modifications of tracking interval. Therefore, there is a tradeoff between accuracy and speed of the algorithm due to the value of *Minimum_Step_Modification*. *Threshold_Error* is the threshold of distance error between the measured and predicted locations of the object, which does not lead to miss the tracked object before awakening and assigning a new *Current Node*, as well as to a high measure of cost (power consumption). In other words, the value of *Threshold_Error* is set such that it results in meeting both a target miss probability and corresponding minimum power consumption. So, *Threshold_Error* should be set to values less than the sensing range of sensors. When the *Error* is less than the *Threshold_Error*, it meets a miss probability better than the target miss probability but it increases the power consumption. So, the tracking interval is less than the value required and it should be increased. Therefore, with a positive value for *Direc._Flag* and *Step_Modif.*, Eq. (22) leads to an increase in the tracking interval. On the contrary, if the *Error* is greater than *Threshold_Error*, it doesn't meet the target miss probability although it decreases the power consumption. So, the tracking interval is greater than a proper value. In this case, the algorithm negatives the *Direc._Flag* and hence the total negative value of the R.H.S of Eq. (22) leads to a decrease in the tracking interval. It is notable that if we set the value of *Step_Modification* to zero, it means we are not adapting the value of tracking interval. However, a large value for *Step_Modification* increases the miss probability of the object due to improper adaptation of the tracking interval values. Since *Threshold_Error* value is set properly, this process leads to a very good functionality of the AHC algorithm.

As it is clear from the given explanations, the AHC algorithm tries to modify the tracking interval in different portions of the movement pattern based on the variations of the average velocity. This can be explained in simple terms. For example, when the average velocity increases, it results in an incorrect estimation of the next location (using Eq. (5)) and hence increasing *Error*. This in turn leads to the modification of the tracking interval using Eq. (22) and Eq. (23). The AHC Algorithm is also described in pseudo code in the following:

Algorithm: Adaptive Hill Climbing (AHC)

Local Variables: Error, Modified_Error.
Parameters Definition:
Threshold_Error: Threshold value of distance error.
Old_Step: Previous tracking interval.
New_Step: Modified tracking interval.
Direction_Flag: Initial search direction for algorithm.
Step_Modifiction: Adaptation factor.
Smoothing_Factor: A factor that smoothes the adaptation process of Step_Modfication.
System Functions: [New_Step, Step_Modification] = AHC (.)
Procedure:

```
Error ←Abs (Measured_Position – Predicted_Position)
if Direction _Flag =1
    if Error > Threshold_Error
        New_Step ← Old_Step – Step_Modification
    elseif Error < Threshold_Error
        Direction _Flag ← ~ Direction _Flag
        if Step_Modification >
                        Minimum_Step_Modification
            Step_Modification ← Step_Modification
                                  /Smoothing_Factor
        end
        New_Step ← Old_Step + Step_Modification
    end
else
    if Error < Threshold_Error
        New_Step ← Old_Step + Step_Modification
    elseif Error > Threshold_Error
        Direction _Flag ← ~ Direction _Flag
        if Step_Modification >
                        Minimum_Step_Modification
            Step_Modification ← Step_Modification
                                  /Smoothing_Factor
        end
        New_Step ← Old_Step – Step_Modification
    end
end
Old_Step ← New_Step
```

5 Simulation Results

In this section, we evaluate the performance of the existing similar adaptive algorithms (including AHC) with respect to the derived optimal performance to find out how much they can get close to the optimal power conservation. The following simulations are performed using MATLAB.

We simulated a scenario for tracking a moving object with the explained movement pattern in subsection 2.2. The covered area by the network is a two-dimensional sensing area of the size 1000×1000 m^2. Fig. 3 illustrates a sample trajectory of the moving object with the explained movement pattern as well as the layout of the sensing field. The dot points represent the sensors and the dashed line represents the moving track of the object. The power consumptions for each operating mode of the sensor nodes are based on the Berkely MICA Mote node [11]. Hence, the following energy consumption values are used in the simulations: *E_Sense* = 12, *E_Transmit* = 27, *E_Idle* = 1.8, *E_Receive* = 21, *E_1st_Rec.* = 144 and *E_2nd_Rec.* = 442 all in mJ/sec, where *E_1st_Rec.* and *E_2nd_Rec.* denote the total energy consumption for the recovery process of lost object in its stages. It is worth noting that we consider only two stages for the recovery process in our simulations; however, for more assertive approaches, the number of the stages can be increased.

The sensing range of the sensors is assumed to be r = 30 m. As well, following the explanations given in subsection 4, the value of *Smoothing_Factor* is set to be 4, which results in an acceptable accuracy. *Threshold_Error*, as described in subsection 4, stands for the threshold of the distance error (*Error*) between measured and predicted locations of the object, which is set to a value about sensing range. However, it can be set to values less than sensing range, which leads to a better tracking accuracy.

To demonstrate the capability of the adaptive time interval management algorithms in terms of their power efficiency and tracking accuracy, we consider two different movement patterns which both are modeled based on the explanations given for practical movements in subsection 2.2. The first movement pattern can be considered as the movement of an intruding soldier (a human) and has the following parameters: V_{max} = 2 m/sec and α_{max} = 10 m/s^2. The second movement pattern is comparable with an intruding vehicle in the covered area and has the following parameters: V_{max} = 40 m/sec and α_{max} = 20 m/s^2.

To compare the simulation results with the mathematical derivations, we consider one of the movement patterns (the intruding vehicle) and calculate the relevant parameters for a tracking process during T = 10 seconds. The average value of α for the intruding vehicle is $\alpha_{max}/2$ = 10 m/s^2. The value of A can be extracted from the data illustrated in Fig. 1, which also depicts the power consumption performance of the explained scenario in some successive short time intervals.

The values of β, γ and ζ can be extracted from Fig. 2. These values, as described earlier in Section 3, should be extracted using a quadratic interpolation for the power consumption values (for A) and quadratic interpolation of S_{opt} values as a function of velocity (for β, γ and ζ). As is clear from Figs. 1 and 2, and considering (7) and (12), for the given data, we have A = 1091, β = 0.0001, γ = -0.0186 and ζ = 1.0598. The values of the parameters used in our simulations are summarized in Table 1. Now, having the above values, we are able to calculate the expected optimal improvement in power consumption in comparison with non-adaptive algorithms due to using the optimal tracking interval management algorithm. According to (21), the expected improvement is $AT^2/60[5y^2 + 10yzT + 16/3z^2T^2]$ = 1091 × (10)^2/60 × (0.103)^2 = 18.68 mW. That is, if the optimal tracking interval management algorithm is used for tracking of the described movement pattern of intruding vehicle with the explained network structure, in the best condition, 18.68 mW enhancement can be achieved in comparison with the case in which a non-adaptive algorithm is used.

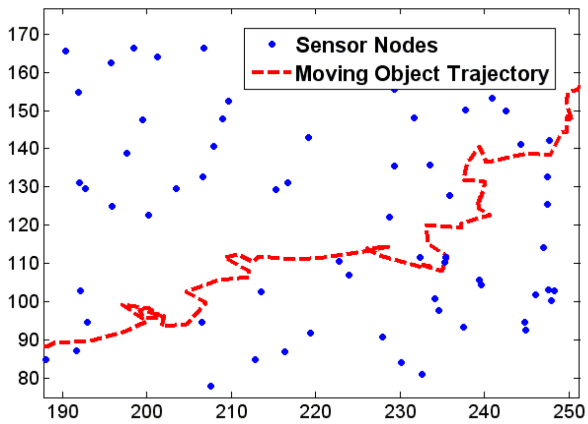

Fig. 3 Moving track of the object with acceleration.

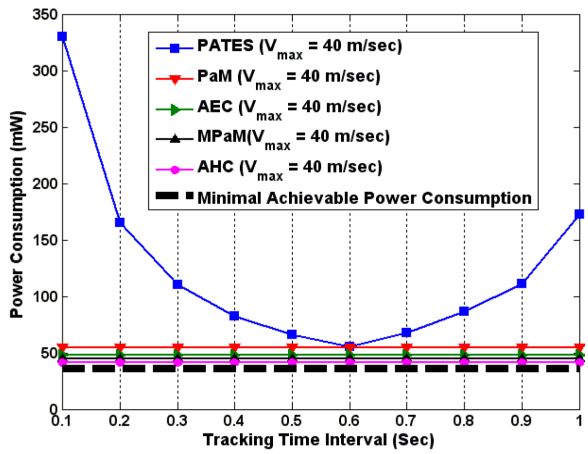

Fig. 4 Power consumption performance vs. tracking interval.

Table 1 Values of the Parameters.

Parameter	Simulation Values	
Run Time (T)	10 Seconds	
A	1091	
B	0.0001	
Γ	-0.0186	
Z	1.0598	
V_{max}	2 m/sec	40 m/sec
α_{max}	10 m/s^2	20 m/s^2

Table 2 Performance Evaluation.

Algorithm	Accuracy		Power Cons. (mW)	
	V_{max} = 2 m/s	V_{max} = 40 m/s	V_{max} = 2 m/s	V_{max} = 40 m/s
PATES (S = 0.2)	99.9 %	99.9 %	144.50	144.50
PATES (S = 0.5)	99.9 %	85 %	57.05	61.14
PATES (S = 1)	99 %	40 %	27.83	158.90
PaM	99 %	94 %	16.31	54.50
AEC	99.9 %	99 %	15.10	48.70
MPaM	99.9 %	99%	13.42	44.61
AHC	99.9 %	99.9%	12.03	41.30
Minimal Achievable Power Consumption			11.18	35.82

The results of simulations for the intruding vehicle are illustrated in Fig. 4.

As it is clear from the figure, simulation results for different fixed values of tracking interval (the PATES [7]) for the described movement pattern show that the power consumption reduces to a minimum value for S = 0.6 second, which leads to the minimal power consumption of non-adaptive algorithms. It is also notable that the AHC is better than the best previously proposed similar algorithm (the MPaM).

Further results of the simulations are summarized in Table 2. In the table, the total power consumption values (in mW) are calculated during T (total duration of tracking process). Also, we bring the power consumption values of tracking network for successive tracking intervals (non-adaptive). It is notable that we consider the calculated minimal power consumption as a benchmark and calculate the capability of all of the mentioned adaptive and non-adaptive algorithms in getting closer to this optimal performance. From first three rows of Table 2, we can notice that smaller tracking intervals (smaller values for S) result in a better tracking accuracy at the expense of larger power consumption in the non-adaptive algorithms such as PATES. But, adaptive time interval management algorithms (PaM, AEC, MPaM and AHC), by dynamic modification of tracking interval, reduce the power consumption while providing even a better tracking accuracy. Now, by considering the minimal power consumption for adaptive algorithms in the last row of the table, for example for the intruding vehicle (V_{max} = 40m/sec and α_{max} = 20 m/s^2), i.e. 35.82mW, the AHC and the MPaM respectively get close to the minimal power consumption by {1- [41.3 – 35.82] /35.82} × 100 = 84.70 % and {1- [44.61 – 35.82] /35.82} × 100 = 75.46 %. This means that the AHC acts about [84.70 % - 75.46 %] = 9.24 % better than the best existing similar adaptive time interval management algorithm (the MPaM).

6 Conclusions

In this paper, we studied the problem of power optimization for distributed target tracking sensor networks using modification of tracking interval. To prove the inefficiency of previously proposed adaptive and non-adaptive algorithms for the case of a moving object with acceleration, we developed a quantitative mathematical analysis and derived the minimal achievable power consumption due to using the optimal adaptive time interval management algorithm. As well, to get closer to the derived minimal achievable power consumption, we describe a blind adaptive tracking interval management (the AHC) algorithm, to be used in our adaptive prediction-based tracking (APT) scheme, and we explained its functional procedure using a detailed conceptual reasoning. Simulation results show that the AHC algorithm gets 9 % closer to the derived minimal achievable power consumption than that of the best similar existing adaptive time interval management algorithm (MPaM). It means that the AHC is capable of significantly increasing the life time of the tracking network while keeping the tracking accuracy in an acceptable level (better than all of the previous algorithms). Finally, from a practical point of view, the AHC algorithm is blind and therefore achieves a certain level of self cognition for modifying the tracking interval, and requires no prior knowledge about the network infrastructure or movement characteristics of the object to become trained with a look up table. It means the AHC can easily be applied to practical environment tracking and monitoring systems.

Appendix

Herein, we illustrate that the derived minimal achievable power consumption formula (in Section 3) can be simplified to its rough estimation in [15], which was calculated using linear interpolation of S_{opt} values versus velocity.

As described in [15], the minimal power consumption for the described adaptive time interval management algorithms was estimated as

$$\min(P_{total}) = P_{\min\_overall} + \frac{A}{12}\left[\alpha \times \gamma \times T\right]^2, \qquad \text{(A.1)}$$

where, A, γ, α and T are the same as described in this paper. We can validate this result by showing that the result of the quadratic interpolation in Eq. (21) is convertible to Eq. (A.1). To do this, it is enough to set $\beta = 0$ in Eq. (12), which results to a linear interpolation. This in turn leads to $z = 0$ and $y = \gamma \times \alpha$ in Eq. (18). Now, substituting $z = 0$ and $y = \gamma \times \alpha$ in Eq. (21) returns the same Eq. (A.1).

References

[1] Akyildiz I. F., Su W., Sankarasubramaniam Y. and Cayirci E., "Wireless sensor networks: a survey," *Computer Networks*, Vol. 38, pp. 393-422, March 2002.

[2] Goel S. and Imielinski T., "Prediction-based monitoring in sensor networks taking lessons from MPEG," *ACM computer Communication Review*, Vol. 31, No. 5, pp. 529-535, Oct. 2001.

[3] Raghunathan V., Schurgers C., Park S. and Srivastava M. B., "Energy aware wireless microsensor networks," *IEEE Signal Processing Magazine*, Vol. 19, pp.40–50, March 2002.

[4] Xu Y. and Lee W. C., "On localized prediction for power efficient object tracking in sensor networks," *First International Workshop on Mobile Distributed Computing (MDC)* , pp. 434-439, May 2003.

[5] Yang H. and Sikdar B., "A protocol for tracking mobile targets using sensor networks," in *Proc. of First IEEE Workshop on Sensor Network Protocols and Applications*, pp. 71-81, May 2003.

[6] Yu X., Niyogi K., Mehrotra S. and Venkatasubramnian N., "Adaptive Target Tracking in Sensor Networks," *Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'04)*, Jan. 2004.

[7] Guo Z., Zhou M. and Zakrevski L., "Optimal Tracking Interval for Predictive Tracking in wireless Sensor Networks," *IEEE Communication Letters*, Vol. 9, No. 9, pp. 805-807, Sept 2005.

[8] Xiao W., Xie L., Chen J. and Shue L., "Multi-Step Adaptive Sensor Scheduling for Target Tracking in Wireless Sensor Networks," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06*, pp. 14-19*)*, May 2006.

[9] Yeow W. L., Tham C. K. and Wong W. C., "Energy Efficient Multiple Target Tracking in Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, Vol. 56, pp. 918-928, March 2007.

[10] Vasanthi N. and Annadurai S., "An Adaptive Energy Efficient Low Latency Sleep Schedule for Target Tracking Sensor Networks," *International Journal of Computer Science and Network Security (IJCSNS)*, Vol. 8, No. 4, pp. 291-298, April 2008.

[11] Yang L., Feng C., Rozenblit J. W. and Qiao H., "Adaptive Tracking in Distributed Wireless Sensor Networks," *13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems*, March 2006.

[12] Jamali-Rad H., Azarafrooz M., Shahhosseini H. Sh. And Abolhassani B., "A New Adaptive Power Optimization Scheme for Target Tracking Wireless Sensor Networks," *IEEE Symposium on Industrial Electronics and Applications (ISIEA'09)*, pp. 307-312, Oct. 2009.

[13] Jamali-Rad H., Abolhassani B. and Abdizadeh M., "A New Adaptive Prediction-based Tracking Scheme for Wireless Sensor Networks," *7th Annual Conference on Communication Networks and Services Research (CNSR'09*, pp. 335-341*)*, May 2009.

[14] Jamali-Rad H., Abolhassani B. and Abdizadeh M., "An Energy Efficient Target Tracking Scheme for Distributed Wireless Sensor Networks," *IEEE International Symposium on Wireless Communication Systems (IEEE ISWCS'09)*, pp. 136-140, Sep. 2009.

[15] Jamali-Rad H., Abolhassani B. and Abdizadeh M., "Toward the Optimal Tracking Interval for Target Tracking Wireless Sensor Networks," *International Conference on Advance Technologies for Communications (ATC'09)*, pp. 161-166, Oct. 2009.

Hadi Jamali-Rad received a B.S. degree in Electrical Engineering from Iran University of Science and Technology (IUST), Tehran, Iran, in 2007 and a M.S. degree in Communications Engineering from the IUST, Tehran, Iran, in 2010. He is currently a Ph.D. candidate at the Electrical Engineering Department of Delft University of Technology (TU Delft). His general interests are in the area of signal processing for communications, wireless and cooperative communications. He has served as the reviewer for several ISI journals and IEEE major conferences.

Bahman Abolhassani received the B.S. degree from Iran University of Science and Technology, Tehran, Iran in 1980, the M.S. and Ph.D. degree from University of Saskachewan, Canada in 1995 and 2001, respectively, all in Electrical Engineering. In 2002, he joined as an assistant professor to the Department of Electrical Engineering, Iran University of Science and Technology, Tehran. He has served as the head of the Electrical Engineering Department for two years. His research of interest are in the area of wireless communications, interference cancellation for CDMA systems, spread spectrum, wireless sensor networks and cognitive radios.

Mohammad Abdizadeh received the B.S. degree in electronics and communications engineering from Amirkabir University of Technology, Tehran, Iran, in 2005 and the M.S. degree in microelectronics engineering from the Sharif University of Technology (SUT), Tehran, Iran, in 2008. Since 2010, he is pursuing his Ph.D. degree at the Department of Geomatics Engineering, the University of Calgary, Canada. His research activities are in the area of wireless communications and digital and statistical signal processing. His current research focuses on the interference problems in GPS systems, interference mitigation schemes based on adaptive algorithms, and GPS signal acquisition and tracking in interference channels.