MULTI-STEP ADAPTIVE SENSOR SCHEDULING FOR TARGET TRACKING IN WIRELESS SENSOR NETWORKS

Wendong Xiao[1], Lihua Xie[2], Jianfeng Chen[1], and Louis Shue[1]

[1]Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
[2]School of School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798

ABSTRACT

Sensor scheduling is essential to collaborative target tracking in Wireless Sensor Networks (WSNs). In this paper, we present a Multi-step Adaptive Sensor Scheduling algorithm (MASS) by selecting the next tasking sensor and its associated sampling interval based on the prediction of tracking accuracy and energy cost over a finite horizon of steps. MASS adopts alternative tracking mode for each prediction step, i.e., the fast tracking mode (FTM) or the tracking maintenance mode (TMM) dependent on whether the estimated or predicted tracking accuracy is satisfactory. The Best Sensor Schedule Sequence (BSSS) is found by searching and comparing the Candidate Sensor Schedule Sequences (CSSSs) at two levels, i.e., the logical tracking mode level which is simplely defined on multi-step tracking modes and the physical quantity performance level by considering the tradeoff between tracking accuracy and energy cost. MASS employs the extended Kalman filter (EKF) algorithm to predict the tracking accuracy and an energy consumption model to predict the energy cost. Simulation results show that, compared with the traditional non-adaptive sensor scheduling algorithm and the single-step adaptive sensor scheduling algorithm, MASS can achieve fast tracking speed and superior energy efficiency without degrading the tracking accuracy.

1. INTRODUCTION

Sensor scheduling is essential to collaborative target tracking in Wireless Sensor Networks (WSNs). In most existing work, usually the tasking sensors are scheduled based on uniform sampling intervals, and the sensor scheduling is degraded to a sensor selection problem. For example, in the information-driven sensor querying (IDSQ) approach [1], the next tasking sensor is selected to maximize the information gain while minimize the resource cost. Multi-step lookahead technique is introduced to IDSQ [2] by predicting information gain over a finite horizon.

A drawback of a uniform sampling interval and pure objective function based sensor scheduling scheme is its difficulty in specifying a specific tracking accuracy goal, which is often required by a practical tracking system [3]. In addition, they are short of strategies to account for the changing of the environment and the dynamics of the target. Recently we have proposed an adaptive sensor scheduling algorithm [4][5] by jointly determining the next tasking sensor and the corresponding sampling interval according to the predicted tracking accuracy and the energy cost. However the adaptive sensor scheduling algorithm in [4][5] is greedy single step based, and suffers from trapping at local minima. In this paper we present a Multi-step Adaptive Sensor Scheduling algorithm (MASS) to incorporate more global knowledge to improve tracking accuracy and energy efficiency.

This work is supported by A*Star SERC Grant No. 052 101 0037.

2. SYSTEM MODELS AND THE EKF ALGORITHM

In this paper, we will consider the single target tracking problem under the distributed architecture. We assume at each time step, only one sensor can be used as the tasking sensor that is responsible for sensing and estimation update, sensor scheduling, and transmitting the sensor scheduling result to the selected tasking sensor. We assume a linear target motion model and a non-linear measurement model, both with Gaussian noise distributions. EKF is used as the estimation algorithm. The target motion is modeled as the following state equation

$$X(k+1) = F(\Delta t_k)X(k) + w(k, \Delta t_k) \tag{1}$$

where $X(k)$ is the state of the target at the k-th time step which happens at t_k, $\Delta t_k = t_{k+1} - t_k$ is the k-th sampling interval. $F(\Delta t_k)$ is the transition matrix dependent on Δt_k. $w(k, \Delta t_k)$ is the process noise, which is also dependent on Δt_k.

If sensor i is used for the k-th measurement $Z_i(k)$ of the target at t_k, the measurement model is given by

$$Z_i(k) = h_i(X(k)) + v_i(k) \tag{2}$$

where h_i is a (generally non-linear) measurement function. $v_i(k)$ is the measurement noise in sensor ***i***. Both w and v_i are independent and assumed to be with zero-mean, white, Gaussian probability distributions. The covariance matrices of $w(k, \Delta t_k)$ and $v_i(k)$ are $Q(\Delta t_k)$ and $R_i(k)$ respectively.

EKF operates in the following way: Given the estimate $\hat{X}(k|k)$ of $X(k)$ at t_k with covariance $P(k|k)$ and assuming sensor j is used for measurement at t_{k+1}, the predicted state $\hat{x}(k+1|k)$ of sensor j at t_{k+1} can be calculated as

$$\hat{X}(k+1|k) = F(\Delta t_k)\hat{X}(k|k), \tag{3}$$

with the predicted state covariance

$$P(k+1\mid k)=F(\Delta t_k)P(k\mid k)F(\Delta t_k)^T+Q(\Delta t_k)\,, \quad (4)$$

and the predicted measurement of sensor j is

$$\hat{Z}_j(k+1\mid k)=h_j(\hat{X}(k+1\mid k))\,. \quad (5)$$

Then the innovation is given by

$$\gamma_j(k+1)=Z_j(k+1)-\hat{Z}_j(k+1\mid k)\,, \quad (6)$$

with the covariance

$$S_j(k+1)=H_j(k+1)P(k+1\mid k)H_j^T(k+1)+R_j(k+1) \quad (7)$$

where $H_j(k+1)$ is the Jacobian matrix of the measurement function h_j at t_{k+1} with respect to the predicted state $\hat{x}(k+1\mid k)$. The EKF gain is given by

$$K(k+1)=P(k+1\mid k)H_j(k+1)^T S_j^{-1}(k+1)\,, \quad (8)$$

and the state estimation will be updated as

$$\hat{x}(k+1\mid k+1)=\hat{x}(k+1\mid k)+K(k+1)\gamma_j(k+1) \quad (9)$$

with the covariance matrix

$$P(k+1\mid k+1)=P(k+1\mid k)-K(k+1)S_j(k+1)K^T(k+1)\,. \quad (10)$$

Particularly, for a 2-dimensional constant velocity model with $X(k)=(x(k),x_v(k),y(k),y_v(k))^T$ where $x(k)$ and $y(k)$ are the x- and y-coordinates of the target at time step k, $x_v(k)$ and $y_v(k)$ are the velocities of the target along x- and y-directions at t_k, the matrix $F(\Delta t_k)$ is given by

$$F(\Delta t_k)=\begin{pmatrix}1 & \Delta t_k & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & \Delta t_k\\ 0 & 0 & 0 & 1\end{pmatrix}. \quad (11)$$

In this paper, the matrix $Q(\Delta t_k)$ is assumed to be

$$Q(\Delta t_k)=q\begin{pmatrix}\frac{1}{3}\Delta t_k^3 & \frac{1}{2}\Delta t_k^2 & 0 & 0\\ \frac{1}{2}\Delta t_k^2 & \Delta t & 0 & 0\\ 0 & 0 & \frac{1}{3}\Delta t_k^3 & \frac{1}{2}\Delta t_k^2\\ 0 & 0 & \frac{1}{2}\Delta t_k^2 & \Delta t\end{pmatrix}. \quad (12)$$

where q is a known scalar that determines the intensity of the process noise.

The above EKF operations are used for the estimation update when a tasking sensor gets an actual measurement and for prediction of the tracking accuracy in sensor scheduling without an actual measurement being taken.

3. MASS FOR TARGET TRACKING

3.1. Overview of MASS

MASS extends the single-step sensor scheduling scheme [4] [5] to the multiple-step case, where for each step the tasking sensor and its associated sampling interval are determined jointly based on the predicted tracking accuracy and the energy cost over the prediction horizon. Hereafter we assume the horizon for prediction in MASS is n steps. MASS is based on the determination of the sampling interval for each step for a given n-step Candidate Sensor Schedule Sequence (CSSS), and implemented by searching such CSSSs to find the Best Sensor Schedule Sequence (BSSS). Only the sensor schedule of the first step of the BSSS will be used as the sensor scheduling result. Similarly to [5], for each step, MASS schedules sensors in the fast tracking mode (FTM) or the Tracking Maintenance Mode (TMM) (Section 3.3). These two tracking modes will be incorporated in determination of the sampling interval for each step of a given CSSS and in comparison of CSSSs.

3.2. Tracking Accuracy and Energy Model

In this paper, the tracking accuracy $\Phi(k)$ at time step k is defined as the trace of the covariance matrix $P(k\mid k)$, i.e.

$$\Phi(k)=\mathrm{Trace}(P(k\mid k))\,. \quad (13)$$

An accuracy threshold Φ_0 is predefined. $\Phi(k)$ is considered to be satisfactory if it is not greater than Φ_0, otherwise it is considered to be unsatisfactory.

Energy consumption is used as the tracking cost. If the current tasking sensor i considers another sensor j ($i\neq j$) as the candidate tasking sensor for the next step, the energy is mainly consumed by data communication (consisting of data transmission from sensor i to sensor j, and data receiving by sensor j) as well as the sensing/processing within sensor j. We adopt the energy models in [6]. The energy consumed by sensor i in transmissions is $E_t(i,j)=(e_t+e_d r_{ij}^{\alpha})b$ where e_t and e_d are decided by the specifications of the transceiver of sensor i, r_{ij} is the distance between sensor i and sensor j, b is the number of bits sent, and α depends on the channel characteristics and is assumed to be known. The energy consumed in data receiving by sensor j is $E_r(j)=e_r b$ where e_r is decided by the specification of the receiver of sensor j. The energy spent in sensing/processing data of b bits by sensor j is $E_s(j)=e_s b$. Thus the total energy consumption by selecting sensor j is

$$E(i,j)=E_t(i,j)+E_r(j)+E_s(j)=e_0+e_1 r_{ij}^{\alpha} \quad (14)$$

where $e_0=(e_t+e_r+e_s)b$, and $e_1=e_d b$.

If $i=j$, i.e., sensor i selects itself as the next tasking sensor, no communication is performed, the associated energy consumption is used for sensing/processing and is expressed as

$$E(i,j)=E_s(j)=e_s b\,. \quad (15)$$

3.3. Scheduling for a Given CSSS

Assume current tasking sensor is sensor i and the current time step is k, and $j_0, j_1, \ldots, j_n$ is a given CSSS, with j_0 being sensor i who performs the sensor scheduling and j_s is the sensor selected for the s-th step. For this CSSS and s=1, 2, …, n, we need to know the predicted sampling interval $\Delta\overline{t_s}$ between j_{s-1} and j_s, the achievable tracking accuracy Φ_{j_s} after s prediction steps, and the predicted energy cost $E(j_{s-1}, j_s)$ for the s-th prediction step. In order to get these parameters, we also need to calculate the state estimation $\hat{x}_{j_s}$ and its covariance matrix P_{j_s} for each prediction step s. For the prediction step s, FTM is used when the estimated tracking accuracy ($\Phi_{j_{s-1}}$) is not satisfactory, or after another prediction step the predicted tracking accuracy (Φ_{j_s}) can not be satisfactory using any sampling interval in $[T_{\min}, T_{\max}]$ where $T_{\min}$ and $T_{\max}$ are the given minimal and maximal sampling intervals respectively. Otherwise, TMM is used where $\Phi_{j_{s-1}}$ is satisfactory and there exists a sampling interval $\Delta\overline{t_s}$ such that Φ_{j_s} can remain to be satisfactory. We also define an unsatisfactory flag $Flag_{j_s}$ to stand for the tracking mode for the *s*-th prediction step. If TMM is used for the *s*-th prediction step then $Flag_{j_s}$ =0 otherwise $Flag_{j_s}$ =1.

Scheduling for a given CSSS is used to determine the step-wise tracking mode, sampling interval, tracking accuracy and energy cost, it is implemented sequentially from the first prediction step to the *n*-th prediction step as follows:

1. For the first prediction step, $\hat{x}(k|k)$, $P(k|k)$ can be updated using the true measurement of sensor j_0 (i.e., sensor *i*) according to (9) and (10), Φ_k can be calculated by (13), and $E(j_0, j_1)$ can be calculated by (14) or (15). $\Delta\overline{t_1}$ can be calculated in two ways:
   1) If Φ_k is unsatisfactory or Φ_k is satisfactory but Φ_{j_1} can not be satisfactory even using $T_{\min}$ as sampling interval, then FTM will be used, $Flag_{j_1}$ =1, and the $\Delta\overline{t_1}$ is set as $T_{\min}$, accordingly $\hat{x}_{j_1}$ and P_{j_1} will be calculated by (3) and (10), and Φ_{j_1} can be calculated from P_{j_1} by (13).
   2) Otherwise TMM is adopted, $Flag_{j_1}$ =0, and $\Delta\overline{t_1}$ is searched in $[T_{\min}, T_{\max}]$ approximately using the discrete search algorithm in [5]. Based on this $\Delta\overline{t_1}$, $\hat{x}_{j_1}$, P_{j_1} and Φ_{j_1} can be calculated in the same way as in 1).
2. For s=2, 3, …, *n* sequentially, by using the (*s-1*)-th step $\hat{x}_{j_s-1}$, P_{j_s-1} as the initial state estimation, the calculation of parameters $E(j_0, j_1)$, $Flag_{j_s}$, $\Delta\overline{t_s}$, $\hat{x}_{j_s}$, P_{j_s}, and Φ_{j_1} can be performed in the same way as in 1.

For each prediction step, an objective function is defined as

$$J_{j_s} = \begin{cases} w\Phi_{j_s}(k) + (1-w)\dfrac{E(j_{s-1}, j_s)}{\Delta\overline{t_s}}, & \text{if } \mathrm{Flag}_{j_s} = 1 \\ \dfrac{E(j_{s-1}, j_s)}{\Delta\overline{t_s}}, & \text{if } \mathrm{Flag}_{j_s} = 0. \end{cases} \quad (16)$$

In (16), $E(j_{s-1}, j_s)/\Delta\overline{t_s}$ is the normalized energy consumption over $\Delta\overline{t_s}$. For FTM, the step-wise objective function is a linear combination of the tracking accuracy and the normalized energy consumption with a weighting parameter $w \in [0,1]$ to balance the tracking accuracy and the energy consumption. For TMM, only the energy consumption is used as the objective criterion.

3.3. Searching BSSS from CSSSs

The step-aggregated objective function of a given CSSS is defined as

$$J(CSSS) = \sum_{s=1}^{n} J_{j_s}. \quad (16)$$

In addition, a decision number is defined using the tracking modes for a CSSS:

$$\mathrm{DN(CSSS)} = \sum_{s=1}^{n} 2^{n-s} * Flag_{j_s} \quad (17)$$

which is the binary number concatenated by the unsatisfactory flags of the CSSS.

The comparison of two different CSSSs is done in 2 different levels, i.e., the logical tracking mode level purely based on the decision numbers and physical quantity performance level based on the step-aggregated objective function. In the logical tracking mode level, a CSSS (denoted as CSSS1) is considered to be better than another CSSS (denoted as CSSS2) if DN(CSSS1)< DN(CSSS2). If CSSS1 and CSSS2 are with the same decision number, the physical quantity performance level is used, and CSSS1 is considered to be better than CSSS2 if *J*(CSSS1)< *J*(CSSS2). For example, in Fig. 1 (a), we can easily to sequence the four CSSSs from the best to the worse as CSSS4, CSSS2, CSSS3, and CSSS1 because their decision numbers are 00, 01, 10, and 11 respectively. However in Fig. 1 (b), CSSS1 and CSSS2 are with the same decision number, so they should be compared by their physical quantity performance.

The BSSS can be found by search CSSSs. Suppose the number of sensors that can be selected for each prediction step is upbounded by *m*, then the computation complexity for this search procedure is O(m^n) and should be affordable because the prediction horizon *n* is usualy a small number such as 2 or 3.

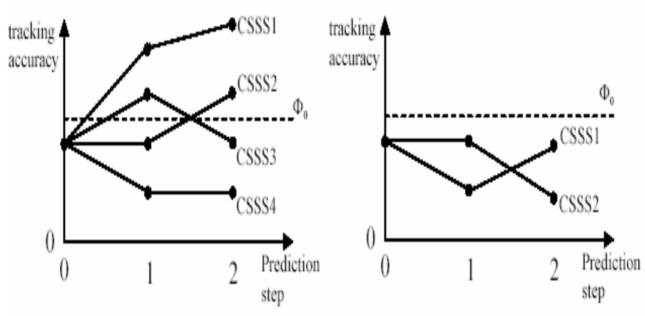
4. EXPERIMENTAL RESULTS

We apply MASS to tracking of a moving sound source (the target) using a network of acoustic amplitude sensors. The measurement model for sensor *j* is

$$z_j(k) = \frac{a}{\|(x(k), y(k)) - (x_s(j), y_s(j))\|} + v_j(k) \quad (22)$$

where $a \in R$ is the assumed known amplitude of the sound source, $(x(k), y(k))$ is the location of the sound source at time step *k* required to be estimated, $(x_s(j), y_s(j))$ is the known position of sensor *j*, and $v_j(k)$ is the zero-mean Gaussian measurement noise of sensor *j* with variance σ_j^2. We use the constant velocity model explained in Section 2 as the target motion model.

The monitored field is 100m$\times$100m with the coordinate from (0, 0) to (100, 100), it is covered by 50 randomly placed sensors. The sound source produces sound with a constant amplitude a=40 and travels at a constant speed v=10m/s along the straight diagonal line starting from (20, 20) and ending at (80, 80). We assume σ_j^2=0.001 for any sensor *j* and q=9 in the process noise. For initialization, we assume the nearest sensor detects the sound source and initiates the tracking procedure. The initial estimation of the sound source is random generated with the mean (20, 7.07, 20, 7.07)^T and the covariance matrix 10*I* where *I* is the identity matrix. The following parameters taken from [6] are used in the energy model: α =2, e_t = 45x10^{-6}, e_r = 135x10^{-6}, e_s =50 x10^{-6}, all in J/bit, and e_d =10x10^{-9} in mJ/bit-m^2. *b* is assumed to be 1024.



(a) (b)

Fig. 1. Comparison of CSSSs (a) at the logical tracking mode level, (b) at the physical quantity performance level

For the sampling interval, we suppose $T_{\min}$ =0.1, $T_{\max}$ =0.5, and the sampling interval is selected from 0.1, 0.2, …, 0.5. We also assume *w*=0.16 for the objective function (16) and the threshold of the tracking accuracy is set as Φ_0 =10.

We compare the performance of MASS (with *n*=2), the non-adaptive sensor scheduling scheme, and the single-step adaptive sensor scheduling. Fig. 2 demonstrates the tracking accuracy achieved by such 3 sensor scheduling schemes, where the non-adaptive sensor scheduling scheme shows big fluctuations in tracking accuracy due to the lack of accuracy control strategy. Both adaptive sensor scheduling schemes can remain to be satisfactory once the tracking accuracy becomes satisfactory. MASS is the fastest (Fig. 2) to achieve the satisfactory tracking accuracy. Fig. 3 demonstrates their accumulated energy consumptions. Although initially MASS uses more energy than single-step adaptive sensor scheduling algorithm, it consumes less total energy. The changes of the sampling intervals of the 2 adaptive sensor scheduling schemes are shown in Fig. 4. As for MASS, to adjust the tracking accuracy, during the initial fast tracking mode, MASS adopts the minimal sampling interval $T_{\min}$ for 5 time steps, then increase the sampling intervals to 0.2, 0.3 and 0.5 seconds. During the tracking maintenance mode, the sampling intervals switch between 0.2 s and o.5 s. The averaged sampling interval for MASS and single-step adaptive sensor scheduling algorithm are 0.332 s and 0.311 seconds respectively.

5. CONCLUSIONS

In this paper, we present the Multi-step Adaptive Sensor Scheduling algorithm (MASS) by introducing the multi-step lookahead prediction strategy. Based on the fast tracking mode (FTM) and tracking maintenance mode (TMM), the Best Sensor Schedule Sequence (BSSS) can be found from the Candidate Sensor Schedule Sequences (CSSSs) at the logical tracking mode level or the physical quantity performance level. Simulation results show that MASS outperforms the traditional non-adaptive sensor scheduling algorithm and the single-step adaptive sensor scheduling algorithm in terms of tracking speed and energy efficiency. As the future work, more advanced techniques (such as particle filter) are required for adaptive sensor scheduling with more general non-linear non-Gaussian tracking scenarios, adaptive motion model based sensor scheduling, and sensor scheduling for multi-target tracking are the other challenging problems for further investigations.

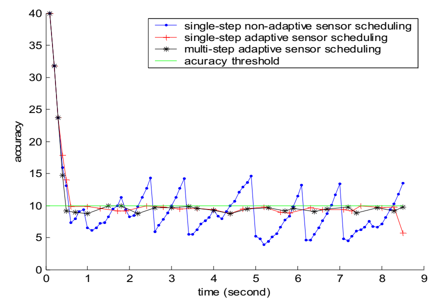


Fig. 2. Tracking accuracy achieved by sensor scheduling schemes.

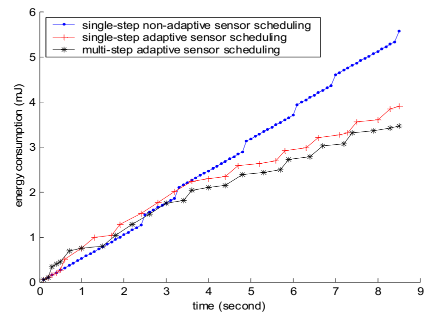


Fig. 3. Accumulated energy consumptions of sensor scheduling schemes

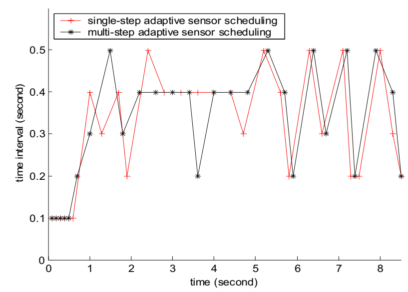


Fig. 4. Change of time interval of the adaptive sensor scheduling schemes

REFERENCES

[1] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich, "Collaborative Signal and Information Processing: an Information Directed Approach," *Proc. IEEE*, vol. 91, pp. 1199-1209, Aug. 2003

[2] J. Liu, D. Petrovic, and F. Zhao, "Multi-Step Information-Directed Sensor Querying in Distributed Sensor Networks," in *Proc. ICASSP'03*, Hong Kong, pp. V-145-8, 2003.

[3] M. Kalandros, and L. Y. Pao, "Covariance control for multisensor systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, pp. 1138-1157, 2002.

[4] W. Xiao, J. Wu, and L. Xie, Adaptive Sensor Scheduling for Target Tracking in Wireless Sensor Network, *Proc. of SPIE,* vol. 5910, pp. 59100B-1~9, 2005.

[5] W. Xiao, L. Xie, and S. Zhang, "Power-efficient Adaptive Sensor Scheduling for Target Tracking in Wireless Sensor Networks," *IEEE Transactions on Systems, Man And Cybernetics-Part C,* submitted.

[6] M. Bhardwaj, and A. P. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in *Proc. IEEE INFOCOM*, New York, pp. 1587-1596. 2002.