Mobile Netw Appl
DOI 10.1007/s11036-011-0311-9

Distributed Active Sensor Scheduling for Target Tracking in Ultrasonic Sensor Networks

Fan Zhang · Jiming Chen · Hongbin Li · Youxian Sun · Xuemin (Sherman) Shen

Abstract Active ultrasonic sensors for target tracking application may suffer from inter-sensor-interference if these highly dense deployed sensors are not scheduled, which can degrade the tracking performance. In this paper, we propose a dynamic distributed sensor scheduling (DSS) scheme, where the tasking sensor is elected spontaneously from the sensors with pending sensing tasks via random competition based on Carrier Sense Multiple Access (CSMA). The channel will be released immediately when sensing task is completed. Both simulation results and testbed experiment demonstrate the effectiveness of DSS scheme for large scale sensor networks in terms of system scalability and high tracking performance.

Keywords wireless sensor networks · target tracking · sensor scheduling

Part of this paper was presented at ChinaCom'10, Beijing, China, Aug. 2010 [1].

F. Zhang · J. Chen (✉) · H. Li · Y. Sun
State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, 310027, China
e-mail: jmchen@iipc.zju.edu.cn, jmchen@ieee.org

F. Zhang
e-mail: zhangfan@iipc.zju.edu.cn

H. Li
e-mail: hbli@iipc.zju.edu.cn

Y. Sun
e-mail: yxsun@iipc.zju.edu.cn

X. Shen
Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada
e-mail: xshen@bbcr.uwaterloo.ca

1 Introduction

Wireless sensor networks (WSNs) have been considered as a promising technique for area surveillance applications [2, 3], and target localization/tracking is essential for these applications [4, 5]. Many approaches have been proposed for target tracking within WSNs [6–8, 15]. According to target behavior, the previous works can be categorized into two classes: cooperative [6–8] and non-cooperative [9, 15]. A cooperative target is part of the network and emits certain forms of physical signals that reveal its presence or report its own identification. In the non-cooperative scenario, however, there exists no information exchange between the target and the network infrastructure. Therefore, sensor nodes need to detect and identify the target *actively* by emitting energy and measuring the feedback. Our previous work [9], a tracking system aiming at non-cooperative targets, utilized passive infrared sensors for target detection and the active ultrasonic sensors for ranging.

In non-cooperative tracking systems using active ultrasonic sensors, Inter-Sensor Interference (ISI) problem is frequently presented if ultrasonic signals of different sensors are emitted within a short time interval. When nearby active sensors work simultaneously at same frequency, the ultrasonic receiver may receive the direct or reflected chirp from other transmitter

Published online: 12 May 2011

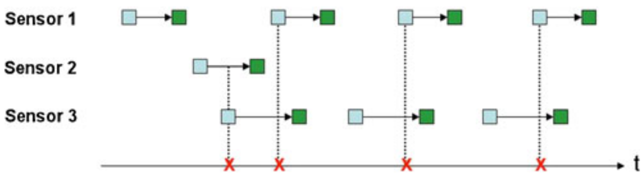


Fig. 1 Inter-sensor interference in ultrasonic ranging

and results in erroneous sensor readings, consequently leads to unacceptable estimation results. Figure 1 shows an example of ISI, where Sensor 3's ranging process periodically collides with that of Sensor 1. Sensor 3 is always interfered by sensor 1 and can not get accurate measurement signal. An occasional ranging task initiated by sensor 2 is interfered by the signal from Sensor 3. Erroneous ranging measurements are even more harmful than missing the reflected signal, since the target estimator accepts the measurement with high confidence. Therefore, sensor scheduling is needed to ensure that, during any time, only one sensor in an ISI region can work to detect the target.

Sensor scheduling, also referred as sensor selection or sensor management, concerns with turning on the right sensors at the right time to achieve desirable performance with minimal energy consumption. Some previous works have addressed the sensor scheduling problem for different tracking systems in WSNs [10–14]. However, most of them mainly study the tradeoff between tracking performance and energy consumption. In [10], this problem is formulated as a partially observable Markov decision process, and Monte Carlo method is developed using a combination of particle filters for belief-state estimation and sampling based Q-value approximation for lookahead. Adaptive sensor activation [11] selects the next tasking sensor and its associated sampling interval based on the prediction of tracking accuracy and energy cost. Priority list sensor scheduling [12] facilitates efficient distributed estimation in sensor networks, even in the presence of unreliable communication, by prioritizing the sensor nodes according to local sensor schedules based on the predicted estimation error. In [13], a scheduling scheme is proposed to improve scalability and energy-efficiency of localization by limiting the number of active nodes participating in localization, while maintaining sufficient coverage to ensure acceptable location error. However, all the methods mentioned above can not be applied directly to the ISI problem for active sensors.

In [15], two sensor scheduling schemes for active sensors are proposed. In the periodic sensor scheduling (PSS) scheme, each ultrasonic sensor detects the target in turn, within the predefined time slots assigned to it. A critical drawback of this sensor scheduling scheme is the existence of empty measurement, where a scheduled sensor may not be in the vicinity of the target. As a result, the system expects lower tracking accuracy and wastes of energy. Whereas in the adaptive sensor scheduling scheme, the next tasking ultrasonic sensor is selected adaptively according to the state prediction of the target. In this scheme, each node needs to know the positions of its neighbors, and the sensor selection process is very computational intensive, since the current tasking node takes the complex calculation for node selection. The considerable computation time may cause delay to the next step sensing, deteriorate tracking accuracy and even lead to target loss. In addition, due to the distributed nature of WSNs, the previously mentioned scheduling schemes are less applicable since scheduling has to be performed in a distributed way to ensure scalability.

In this paper, we propose a distributed sensor scheduling (DSS) scheme for target tracking in ultrasonic sensor network. The tasking sensor node is elected spontaneously from the sensor candidates in a distributed way via random competition based on Carrier Sense Multiple Access (CSMA). As soon as the sensing task is completed, the channel is released immediately for other sensor nodes with pending sensing tasks. It is demonstrated the DSS scheme is effective for large scale sensor networks and robust to dynamic topology changes.

The remainder of the paper is organized as follows. The system model and problem setup are introduced in Section 2. Section 3 presents the DSS scheme. Sections 4 and 5 evaluate the scheme with extensive simulation and testbed experiment, respectively. Finally, conclusions and future work are given in Section 6.

2 Modeling and problem formulation

Typically, each sensor node in WSNs has short sensing range and the on-board processor is limited both in memory and processing speed. In our tracking system, each sensor node is built with one Passive InfraRed (PIR) sensor, multiple ultrasonic ranging sensors and one processing/communication board. The sensor nodes only take charge of detecting/sensing the target and transmitting the sensing results to the central computer. Noise corrupted measurements may cause solving the tracking problem with insufficient accuracy. Therefore, the Kalman Filter (KF) is used to estimate the state of our dynamic system with noisy measurements. The current state $\mathbf{X}(k)$ of the system is calculated recursively from the previous state $\mathbf{X}(k-1)$ and a new measurement $z_i(k)$. An error covariance matrix $\mathbf{P}$ is calculated along with $\mathbf{X}$ for demonstrating the accuracy of the estimation.

In this paper, we focus on single target tracking of a 2D space and the system is set as $\mathbf{X} = [x, y]^T$. The state-space model of the system can be written as

$$\mathbf{X}(k) = \mathbf{F}\mathbf{X}(k-1) + \mathbf{w}(k)$$

where $\mathbf{X}(k)$ is the state of the target at k-th measurement time, $\mathbf{F}$ models the system dynamics, and it is assigned as identity matrix due to the unknown target motion pattern, $\mathbf{w}(k)$ is the variable representing unmodeled process noise or higher order motion.

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

$$z_i(k) = h_i(\mathbf{X}(k)) + v_i(k)$$

where $v_i(k)$ is measurement noise in sensor i. Both $\mathbf{w}$ and v_i are assumed to have white noise properties with zero mean and their covariances are $\mathbf{Q}$ and $\mathbf{R}_j$ respectively. h_i is the measurement function and following the ultrasonic ranging model, it can be calculated as

$$h_i(\mathbf{X}(k)) = ||\mathbf{X}(k) - \mathbf{X}_i||$$

where $\mathbf{X}_i$ denotes the coordinates of sensor i. Thus, the measurement process is non-linear with respect to $\mathbf{X}$ in our case, and traditional KF is no longer applicable. An Extended Kalman Filter (EKF) can be applied with the following linearization:

$$\mathbf{H}_i(k) = \frac{\partial h_i(\mathbf{X}(k))}{\partial \mathbf{X}} = \left[\frac{x - x_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2}}, \frac{y - y_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2}}\right]$$

$\mathbf{X}(k)$ and $\mathbf{P}(k)$ constitute the state of the EKF and the recursion is comprised of two phases: (a) a prediction phase during which the current state is estimated based on the previous state according to the state-space model, and (b) a correction phase which is used to refine the predicted state based on the new measurements.

KF prediction phase Given the previous state $[\mathbf{X}(k-1), \mathbf{P}(k-1)]$, the predicted state $[\bar{\mathbf{X}}(k), \bar{\mathbf{P}}(k)]$ is

$$\bar{\mathbf{X}}(k) = \mathbf{F}\mathbf{X}(k-1)$$

$$\bar{\mathbf{P}}(k) = \mathbf{F}\mathbf{P}(k-1)\mathbf{F}^T + \mathbf{Q}$$

EKF correction phase With the new measurement $z_i(k)$ and the predicted state $[\bar{\mathbf{X}}(k), \bar{\mathbf{P}}(k)]$, the new state $[\mathbf{X}(k), \mathbf{P}(k)]$ is refined by

$$\mathbf{K}(k) = \bar{\mathbf{P}}(k)\mathbf{H}_i(k)^T[\mathbf{H}_i(k)\bar{\mathbf{P}}(k)\mathbf{H}_i(k)^T + \mathbf{R}_i]^{-1}$$

$$\mathbf{X}(k) = \bar{\mathbf{X}}(k) + \mathbf{K}(k)[z_i(k) - h_i(\bar{\mathbf{X}}(k))]$$

$$\mathbf{P}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}_i(k)]\bar{\mathbf{P}}(k)$$

where $\mathbf{K}(k)$ is Kalman Filter gain.

For the sensor scheduling problem, we only concern how to sense the target efficiently via the sensor nodes negotiation. We first define some notations in Table 1. The definition of function f is:

$$f(t) = \begin{cases} i, & \text{select node } i \text{ as the tasking node at time } t \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

In order to avoid ISI between ultrasonic sensors, only one sensor node is tasked to actuate its ultrasonic sensors for range measurement each time. And the time difference between two successive measurement epoches should be larger than T_d. Simulation results in [9] reveal that the tracking performance can benefit from higher sampling frequency. With view to the existence of empty detection when the scheduled sensor is not in vicinity of the target, we define another function $g(t)$ to represent the effective scheduling:

$$g(t) = \begin{cases} 1, & f(t) \neq 0 \text{ and } ||x_{f(t)} - x(t)|| < R \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Table 1 Notation definitions

Symbol	Definition
$\lvert\cdot\rvert$	$\lvert\cdot\rvert$ represents the cardinality of a set.
V	Set of the sensor nodes, i.e., V={1,2,...,$\lvert V\rvert$}.
$[0, T]$	Duration that the target in the monitored area.
$x(t)$	Position of the target at time t, and $t \in [0, T]$.
x_i	Position of sensor node i, and $i \in V$.
T_d	Die-out time of the ultrasonic wave in a ranging operation.
R	Sensing range of the ultrasonic sensors.
$f(t)$	Function that record the process of the scheduling during $[0, T]$, the detailed definition is given in Eq. 1 and $f : [0, T] \to V \cup \{0\}$.
N	Total number of nonzero element in $f(t)$, means the number of the selection.
n_i	The ith nonzero element in $f(t)$, means the ith tasking node. So $i = 1, 2, ..., N$, $n_i \in V$.
t_i	Time of the ith tasking node selection, i.e, $f(t_i) = n_i$.

Therefore, the ISI problem can be represented by an optimization form:

$$\begin{aligned} \text{maximum} \quad & \frac{1}{T}\int_0^T g(t)dt \\ \text{subject to} \quad & t_i - t_{i-1} \geq T_d \\ & i = 2, 3, ..., N \end{aligned} \tag{3}$$

3 Distributed sensor scheduling scheme

By considering the ranging operation of a sensor node as the occupation of a shared channel, the ISI problem among active ultrasonic sensors in WSNs can be converted to the problem of multiple access in a shared channel. Hence, the MAC protocols can be used to solve the ISI problem.

A common MAC paradigm is TDMA (time-division multiple access) which schedules all nodes to transmit messages at different times. The PSS scheme mentioned above is kind of this fashion. However, TDMA often requires a central unit to find a collision-free schedule and developing an efficient schedule with high channel utilization is very difficult. Meanwhile, it uses topology information as a basis for access scheduling and needs clock synchronization among neighbors, thus lacks efficiency and scalability when the networks subject to frequent topology changes.

Another classic MAC protocol is CSMA, which is a probabilistic protocol. In a nutshell, CSMA verifies the absence of other traffic before transmitting in a shared channel. It is popular because of its simplicity, scalability and robustness. It does not require clock synchronization and global topology information, thus is able to handle dynamic topology changes which generally occur in WSNs without extra operations. Because of these features, sensor nodes negotiate with each other using CSMA is our proposed scheme.

The main idea of DSS scheme is that when a node has a pending ultrasonic sensing request, it checks if there is already an occupation announced by other node in recent σ millisecond. If no occupation is recorded, the node broadcasts its own message to announce occupation in the upcoming σ millisecond and then conducts target range sensing. Otherwise, the node waits a bounded random time for next round occupation. Obviously, the minimal σ should be equal to T_d . The detailed procedure is summarized below:

- a) In initial state, all the nodes are in sleep mode; when a mobile target enters the monitored area, sensor nodes close to the target will be activated by the PIR sensor. As the target moves along the trace in the area, some activated sensor nodes may go back to sleep mode because the target moves out of its sensing region. Also, there are some newly activated sensor nodes;
- b) Once a sensor node is activated, it will calculate a random T_{backoff}, and then start its *delay timer* with interval T_{backoff}. Let

  $T_{\text{backoff}} \in [T_{\min}, T_{\max}]$ and $T_{\min} = T_d$
- c) As soon as the node with the smallest T_{backoff} triggers its *delay timer*, it will broadcast a DETECT message to the other activated nodes and become the tasking node to sense the target. Any node that overhears the DETECT message will immediately give up its declaration as the tasking sensor node by restarting the *delay timer* with a new random T_{backoff}. The $timer_{\text{delay}}$ in the tasking sensor node is also restarted with new random T_{backoff}.

Without loss of generality, it is assumed there is no transmission delay between sensor nodes. Therefore, the DETECT message broadcasted by the selected node can be used as a locally synchronization signal to make all the activated nodes start their *delay timer* at the same time. It can be seen that the computation burden is distributed among the activated sensor nodes and the sensor selection is totally distributed. Another advantage of the DSS scheme is that each node does not need to know the positions of its neighboring nodes and thus conserving the limited memory for other processing. Therefore, DSS is well suited for implementation in WSNs.

In theory, it is almost impossible that more than one sensor nodes broadcast DETECT message at the same time because the T_{backoff} is randomly selected. In practice, however, the T_{backoff} can only be selected from a finite discrete set due to quantization constraint. For one sensor node, we have

$$T_{\text{backoff}} \in \{T_0, T_1, \ldots, T_{M-1}, T_M\}$$

$$T_0 = T_{\min}, T_M = T_{\max}$$

$$T_i - T_{i-1} = \Delta t, i = 1, 2, \ldots, M$$

where Δt is a constant value denoting the resolution of the *delay timer* which depends on the sensor nodes and it equals to $1ms$ in our system. In order to avoid the situation that more than one nodes sense the target at the same time, the node with the smallest ID will be chosen as the tasking node among the nodes whose *delay timer* are triggered simultaneously, and Algorithm 1 is the pseudo-code for the DSS scheme. The parameter M

Algorithm 1 Distributed sensor scheduling algorithm

```
Input: the trace of the target x(t)
Output: the sensor scheduling results f(t)
Initialization: f(t) = 0, t ∈ [0, T];
//det_send means broadcast the DETECT message.
//det_recv(j) means receive the DETECT message from node j.
for target moves in the monitored area do
  if target moves into the sensing range of node i then
    sensor i is activated from the sleep mode;
  end if
  for each activated node i do
    generate a random T_backoff;
    start the delay timer with T_backoff;
    wait for det_send or det_recv(j);
    if not det_send and det_recv(j) at time t then
      stop the delay timer;
      generate a new random T_backoff;
      start the delay timer with T_backoff;
      set f(t) = j;
    end if
    if det_send and not det_recv(j) at time t then
      sensing the target;
      generate a new random T_backoff;
      start the delay timer with T_backoff;
      set f(t) = i;
    end if
    if det_send and det_recv(j) at time t then
      if i > j then
        stop the delay timer;
        generate a new random T_backoff;
        start the delay timer with T_backoff;
        set f(t) = j;
      else
        sensing the target;
        generate a new random T_backoff;
        start the delay timer with T_backoff;
        set f(t) = i;
      end if
    end if
  end for
  if target moves out of the sensing range of node i then
    sensor i goes back to sleep mode;
  end if
end for
```

denotes the size of candidate set for T_{backoff}. Larger M indicates that sensor nodes are less possible to collide with each other, but the expected backoff time may be larger and result in sensing latency.

As described above, the DSS scheme employs a quite simple strategy to avoid collision where node ID is used to break ties when multiple nodes trigger their *delay timer* simultaneously. However, the strategy is not so efficient, especially when M is small. This is because the node with smaller ID is preferred to be selected with this strategy and small M always intensifies this inclination. Consider node i among L activated nodes, which i is the k-th smallest number in the L node IDs, $1 \leq k \leq L$. Since node i is selected to be the tasking node if and only if when its T_{backoff} is the smallest and the IDs of other nodes which have the same T_{backoff} are all larger than i, the probability $Pr(k)$ that node i is selected can be expressed as follows:

$$
\begin{aligned}
Pr(k) &= \sum_{i=0}^{M} \frac{1}{M+1} \left(\frac{M-i}{M+1}\right)^{k-1} \left(\frac{M+1-i}{M+1}\right)^{L-k} \\
&= \frac{1}{(M+1)^L} \sum_{i=0}^{M} (M-i)^{k-1} (M-i+1)^{L-k} \\
&= \frac{1}{(M+1)^L} \sum_{i=0}^{M} \left[(M-i+1)^{L-1} \left(1 - \frac{1}{M-i+1}\right)^{k-1} \right]
\end{aligned}
\tag{4}
$$

We can then obtain the gap between the smallest ID node and the largest one as follows:

$$
\begin{aligned}
&Pr(k=1) - Pr(k=L) \\
&= \frac{1}{(M+1)^L} \sum_{i=0}^{M} \left[(M-i+1)^{L-1} - (M-i)^{L-1} \right] \\
&= \frac{1}{M+1}
\end{aligned}
\tag{5}
$$

Clearly, according to Eq. 4, $Pr(k)$ decreases monotonically with increasing k, i.e., node with smaller ID has higher probability to become the tasking node. Meanwhile, Eq. 5 indicates that decreasing M will enlarge the inclination. Figure 2 shows a visualized simulation example.

High selected probability of specific node leads to repeated selection of that node which always results in large accumulate error. Therefore, we can improve the DSS scheme by modifying the collision avoidance strategy to diminish the accumulate error. The key idea is to use random value to break ties. When a node triggers its *delay timer*, it will generate a random value and broadcast this value to other activated nodes with the DETECT message. If it happens that multiple nodes trigger their *delay timer* simultaneously, the node with the smallest random value will be chosen as the tasking node. With this strategy, each node has the same chance to be the tasking node, which further reduces the accumulate error to a bare minimum.

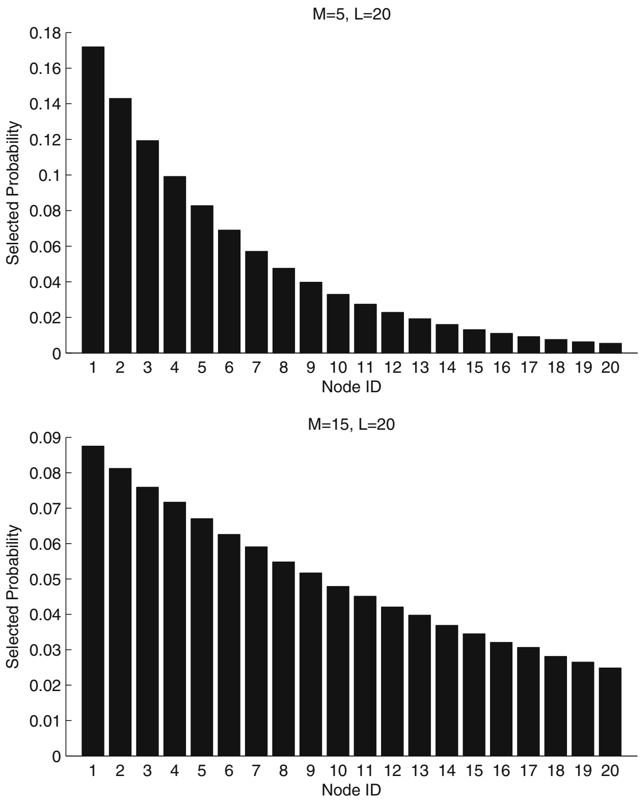


Fig. 2 Selected probability for all nodes with different M in DSS scheme

4 Simulation results

We evaluate the proposed DSS scheme with both simulation and testbed implementation. We compare the tracking performance of the DSS scheme with the PSS scheme and also validate the efficiency of the improved DSS scheme.

In the simulation, the monitored area is modeled as a square field. The Random WayPoint mobility model (RWP) [16] is used to generate movement traces of the mobile target. An estimation error at one point in the trace is defined as the Euclidean distance between the estimated position and corresponding true position

$$e(t) = \|\mathbf{x}(t) - \widetilde{\mathbf{x}}(t)\| \tag{6}$$

where $\mathbf{x}(t)$ is the ground truth position vector and $\widetilde{\mathbf{x}}(t)$ is the estimated position.

The mean tracking error is defined as the averaged error of all the estimated points in the trace

$$\overline{e} = \frac{1}{(t_K - t_0)} \Sigma_{i=1}^{K} (t_i - t_{i-1}) e(t_i) \tag{7}$$

Table 2 Default simulation setup

Parameter	Description
Field area	6×6 m^2
Sensing noise model	$\mathbb{N}(0, 0.0025)$
Number of sensor nodes	12, uniformly deployed
Target velocity	$\mathbb{U}(0.3, 0.7)$ m/s
PIR range and angle	$0 \sim 3$ m, $\pm\pi$
Ultrasonic range and angle	$0 \sim 3$ m, $\pm\pi$
T_d	30 ms
M	15

where t_0 and t_K are the starting time and ending time, respectively, and K denotes the total numbers of estimations.

The presented results are averaged over 50 independent simulation runs for high confidence. Table 2 illustrates the default simulation setup. For the sake of clarity, we assume that the sensing angle of PIR sensor and ultrasonic sensor are both $\pm\pi$, i.e., that the sensing model is a circle.

We compare the PSS scheme with the proposed DSS scheme. Figure 3 shows the target moving traces estimated by PSS and DSS. It can be seen that the estimated trace generated by DSS is closer to the true one, and the corresponding tracking error of these two estimated traces are 0.1131 and 0.0575 m, respectively. For a larger network, the tracking results are shown in Fig. 4. The monitored area is 12×12 m^2 with 48 sensor nodes deployed. It can be seen that the DSS scheme still performs well under such conditions. On the other hand, PSS scheme almost lost the target during the first corner and at the end of the trace because of the empty

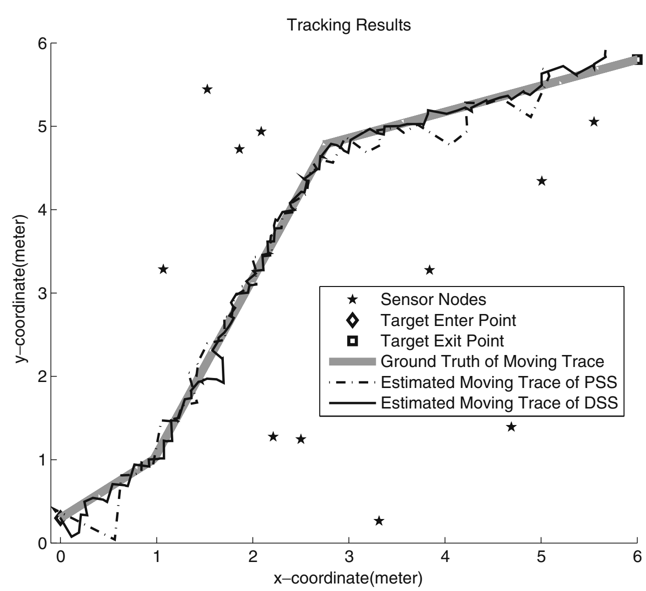


Fig. 3 Simulated trajectories of PSS and DSS schemes

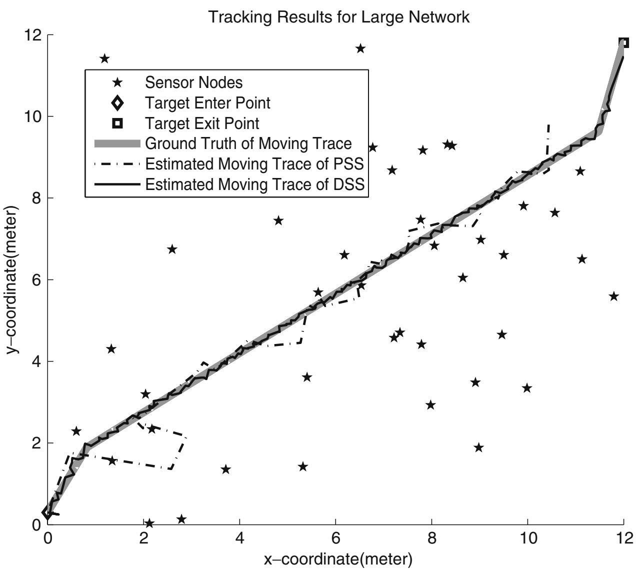


Fig. 4 Simulated trajectories for larger network

detections. The corresponding tracking error of PSS and DSS are 0.3651 and 0.0534 m, respectively.

4.1 Impact of the number of sensor nodes

We compare the DSS scheme with PSS scheme under a different setting of sensor quantity, ranging from 6 to 30. Figure 5 shows that (i) the DSS scheme outperforms the PSS scheme; and (ii) with the increase number of deployed sensor nodes, the tracking error for both schemes are reduced, and the error converges to a constant when the node number is very large. This is because the uncovered area is large when less number of nodes are deployed, which leads to high probability of losing the target. On the other hand, the area coverage is saturated when the node number is large enough, so that any more sensor deployment does not help in terms of tracking performance.

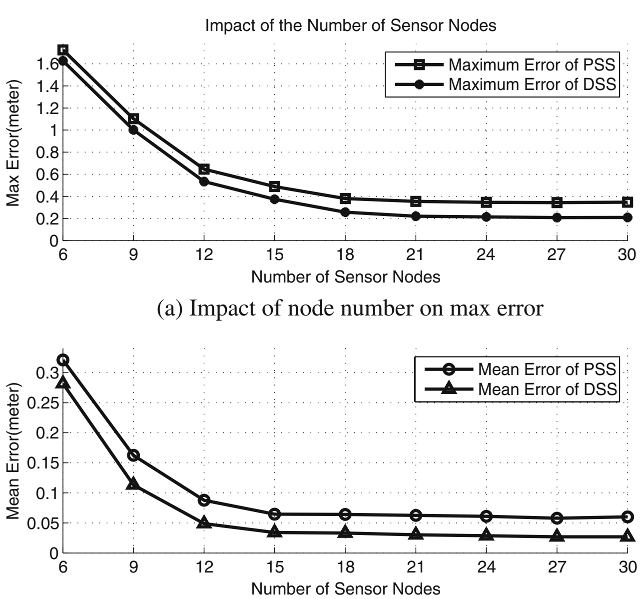
4.2 Impact of the scale of ROI

ROI means the region of interest, i.e., the monitored area. We compare the performance of both schemes under different scales of ROI, with same node density. For each scale, we define the number of nodes by

$$N_{\text{node}} = \alpha l^2$$

where α is a density factor, and l is the side length of the square monitored area. In the simulation, l increases from 6–30 m in steps of 3 m, and α is set to 1/3.

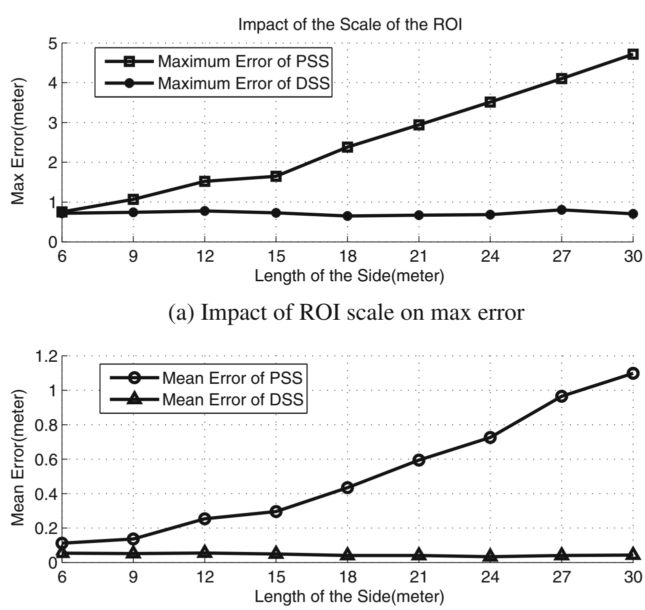
Figure 6 shows: (i) DSS scheme is very robust to the network scale (error keeps constant when the network scale increases) since it only schedules the activated nodes, while the error with the PSS scheme increases quickly because of the growing number of empty detections; and (ii) the DSS scheme is superior to the PSS scheme, especially when the network scale is very large.



(a) Impact of node number on max error

(b) Impact of node number on mean error

Fig. 5 Impact of node number on tracking error



(a) Impact of ROI scale on max error

(b) Impact of ROI scale on mean error

Fig. 6 Impact of ROI scale on tracking error

4.3 Impact of the target velocity

In the third group of simulations, we compare the performance of schemes under different target velocities. By setting the extent of the uniform distribution to 0.4, we vary the mean target velocity v from 0.3 to 1.9 m/s, i.e., in each set of simulations, the target velocity is randomly selected between $(v - 0.2) \sim (v + 0.2)$ m/s.

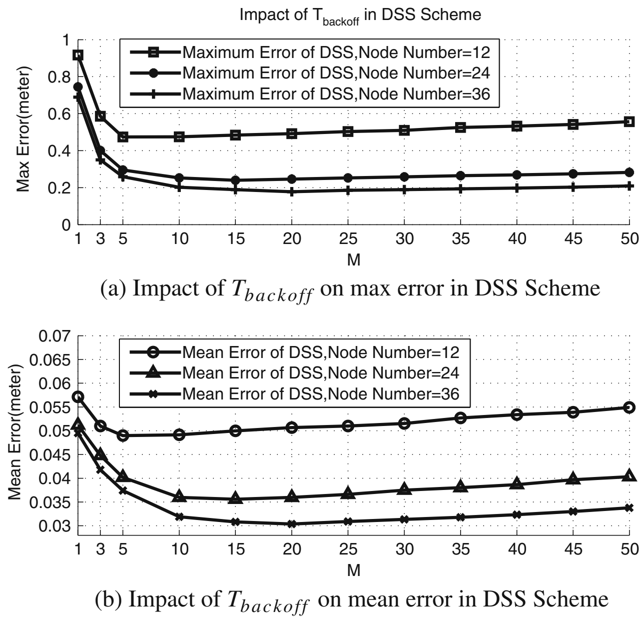
We depict the simulation results in Fig. 7. We can see: (i) the DSS scheme outperforms the PSS scheme; (ii) the advantage gets more remarkable with the increase of the target velocity since when the target moves faster, the negative effect of empty detection on tracking accuracy becomes more obvious; and (iii) The tracking error for both schemes increases linearly with the target velocity. This is because higher target velocity results in larger process noise in our model, where EKF promises higher tracking error.

4.4 Impact of M in distributed sensor scheduling scheme

Figure 8 shows the impact of M to the DSS scheme. Simulation results indicate that: (i) there exists an optimal M_o, when M equals to M_o, the tracking error is the minimum; (ii) when M is larger than M_o, the tracking error increases with the increase of M because the average sampling period increases; and (iii) when M is smaller than M_o, the tracking error reduces with the increase of M. The reason for this phenomenon is that the probability of more than one sensor nodes triggering their *delay timer* at the same time increases with the smaller M. Therefore, only nodes with the smallest ID will be selected to sense the target, which results in the increase of accumulate error; and (iv) from the simulation results,

$$M_o \propto \alpha \tag{8}$$

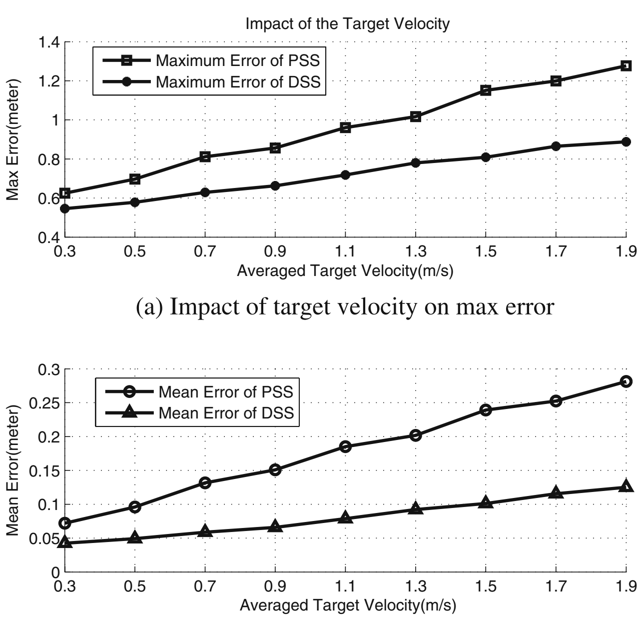
Equation 8 indicates that the optimal M_o is approximately proportional to the node density. This is because



(a) Impact of $T_{backoff}$ on max error in DSS Scheme

(b) Impact of $T_{backoff}$ on mean error in DSS Scheme

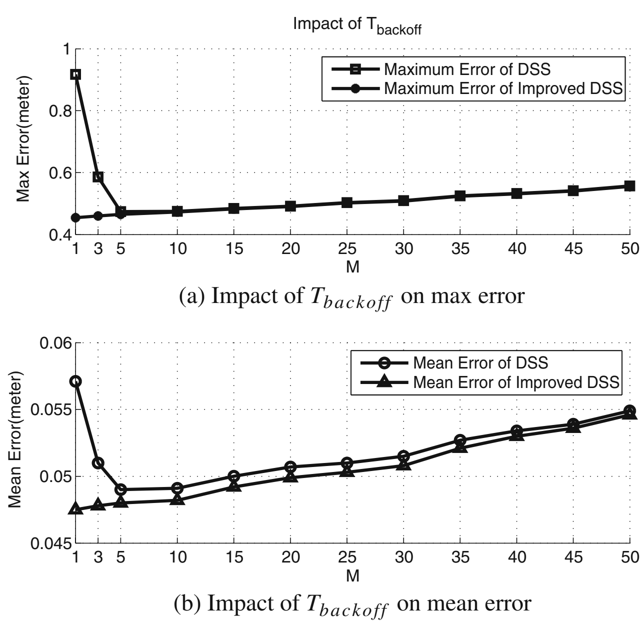
Fig. 8 Impact of T_{backoff} on tracking error in DSS scheme



(a) Impact of target velocity on max error

(b) Impact of target velocity on mean error

Fig. 7 Impact of target velocity on tracking error



(a) Impact of $T_{backoff}$ on max error

(b) Impact of $T_{backoff}$ on mean error

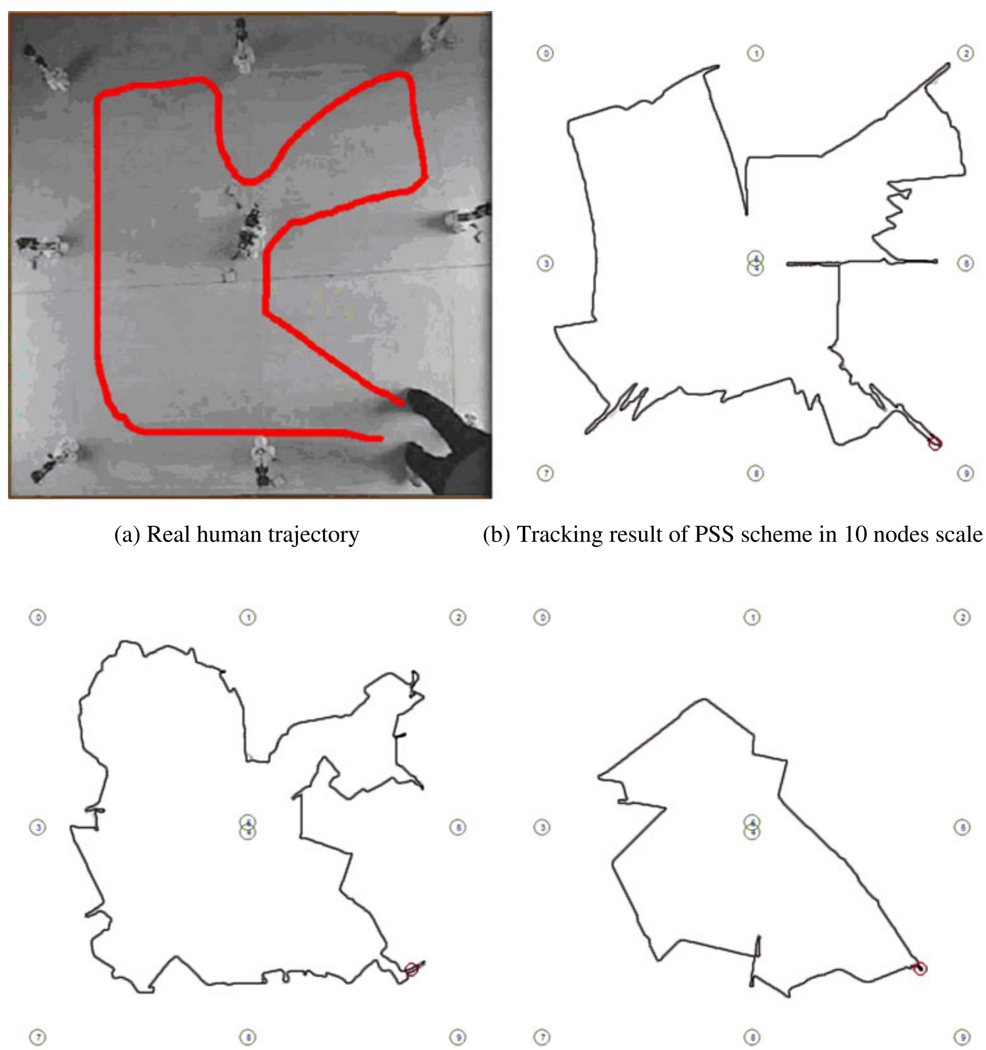
Fig. 9 Impact of T_{backoff} on tracking error larger node density creates chances for more than one sensor nodes triggering their *delay timer* at the same time, so M should be properly chosen according to the node density when implementing the DSS scheme.

We also compare the tracking performance of the improved DSS scheme with the DSS scheme under different M. Figure 9 shows that: (i) the improved DSS scheme reduces the tracking error of the DSS scheme, especially when M is small, which also reveals the efficiency of our new collision avoidance strategy; and (ii) the tracking error of the improved DSS scheme is approximately proportional with M.

5 Testbed experiment

A 5.2×5.2 m testbed has been built, as shown in Fig. 10a, for testbed-scale experiments. We mount an overhead camera right up on the ceiling of the testbed to capture the experiments or extract accurate positions of interested objects. Each sensor node is integrated with one main board, one PIR sensor and three active ultrasonic sensors to realize the tracking. The main board, which is mainly composed of Atmel128L [17] as core unit and CC2420 [18] as communication chip is used as an communication/computation module. The PIR sensor can detect the changes of infrared energy radiation from the environment due to the movement of the target and outputs a high level signal which is used as an external interrupt to activate the node. The ultrasonic sensor has a maximum range of 3 m and the effective angle is $\pi/3$. All the three ultrasonic sensors are connected with the main board via UART connection.

In our experiments, ten sensor nodes are deployed in the testbed using 3×3 grid form with 2.2 m



(a) Real human trajectory

(b) Tracking result of PSS scheme in 10 nodes scale

(c) Tracking result of DSS scheme

(d) Tracking result of PSS scheme in 40 nodes scale

Fig. 10 Testbed experiments for PSS and DSS schemes displacement. Specifically, two nodes are placed in the middle of the area to provide with $\pm\pi$ coverage. The three ultrasonic sensors are fixed on the pipe with two different configurations, to cover either $\pi/2$ or π angle range. The node that covers $\pi/2$ is placed at the corner of the field, and The nodes cover π are placed in other positions.

In the experiments, a moving person is considered as the mobile target to track. Figure 10a, taken by the overhead camera, shows the sensor node deployment and target trajectory. We put a sequence of marks on the testbed as the trails to follow. And the person puts his feet on the marks at one pace per second, to generate identical target trails for each test. The estimated trajectories of PSS scheme and DSS scheme are shown in Fig. 10b and c, respectively. Compared with the true trajectory, both estimated trajectories are very close to the true one. The overall tracking error is about 19 cm for DSS and 28 cm for PSS. Due to the relatively small network scale, using PSS scheme gives enough effective measurements for target updates. Next we evaluate the number of effective measurements. The PSS is 97 effective measurements and the DSS is 216, which reveals the existence of empty detection for PSS scheme. When the target moves within the field, DSS scheme always activates sensors in the vicinity of the target and thus makes most of the measurements effective. The PSS scheme, however, emphasizes fairness among sensors regardless of the target position and yields large number of empty detections.

We further test the performance of PSS scheme under larger scale network settings. Besides the ten deployed sensors, we add another 30 *virtual* nodes working around the testbed with the same deployment and sharing the channel. The tracking result is shown in Fig. 10d. Noticeably, the tracking error of PSS is about 1.15 m, which is very large and the corresponding number of effective measurements is only 27.

6 Conclusion

In this paper, we have proposed a distributed sensor scheduling (DSS) scheme for target tracking in ultrasonic sensor network. With the scheme, the computation burden for each sensor can be reduced significantly and the scalability can be guaranteed. It has been demonstrated that DSS scheme outperforms the PSS scheme, especially with large scale network. Since DSS scheme focuses on the effective measurement, which only considers the target presence, more efficient sensor scheduling scheme could be designed with the estimation/prediction techniques using the historical information of the target. Furthermore, DSS scheme requires sensor nodes to keep negotiating with others, therefore, energy efficiency may be a critical issue for real system implementation. In addition, tracking and adaptive sensor scheduling for multiple targets using multiple modalities are also very interesting.

Acknowledgements The research is supported in part by NSFC-Guangdong joint Project grant no. U0735003; NSFC grant nos. 60736021 and 60974122; and Zhejiang Provincial Natural Science Foundation of China under Grant No. R1100324.

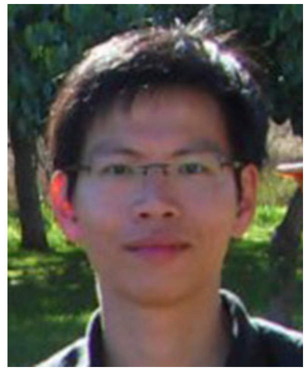
References

1. Zhang F, Chen J, Li H, Sun Y, Shen X (2010) Distributed interfering sensor scheduling for target tracking. In: Proceedings of IEEE ChinaCom 2010. Beijing, China
2. Chipara O, Lu C, Bailey TC, Roman G-C (2010) Reliable clinical monitoring using wireless sensor networks: experience in a step-down hospital unit. In: Proceedings of ACM SenSys 2010. Zurich, Switzerland
3. Kim DH, Kim Y, Estrin D, Srivastava MB (2010) SensLoc: sensing everyday places and paths using less energy. In: Proceedings of ACM SenSys 2010. Zurich, Switzerland
4. Wang C, Chen J, Sun Y (2010) Sensor network localization using kernel spectral regression. Wirel Commun Mob Comput (Wiley) 10(8):1045–1054
5. Chen J, Cao K, Li K, Sun Y (2011) Distributed sensor activation algorithm for tracking with binary sensor networks. Cluster Comput (Springer) 14(1):55–64
6. Markham A, Trigoni N, Ellwood SA, Macdonald DW (2010) Revealing the hidden lives of underground animals using magneto-inductive tracking. In: Proceedings of ACM SenSys 2010. Zurich, Switzerland
7. Kusy B, Ledeczi A, Koutsoukos X (2007) Tracking Mobile Nodes Using RF Doppler Shifts. In: Proceedings of ACM SenSys 2007. Sydney, Australia, pp 29–42
8. Klingbeil L, Wark T (2008) A wireless sensor network for real-time indoor localisation and motion monitoring. In: Proceedings of ACM/IEEE IPSN 2008. St. Louis, Missouri, pp 39-50
9. Li H, Miao D, Chen J, Sun Y, Shen X (2009) Networked ultrasonic sensors for target tracking: an experimental study. In: Proceedings of IEEE GlobeCom 2009. Honolulu, Hawaii
10. He Y, Chong KP (2004) Sensor scheduling for target tracking in sensor networks. In: Proceedings of IEEE CDC 2004. Atlantis, Paradise Island, Bahamas, pp 743–748
11. Chen J, Cao K, Sun Y, Shen X (2009) Adaptive sensor activation for target tracking in wireless sensor networks. In: Proceedings of IEEE ICC 2009. Dresden, Germany
12. Huber MF, Hanebeck UD (2008) Priority list sensor scheduling using optimal pruning. In: Proceedings of the 11th international conference on information fusion. Cologne, Germany
13. Gribben J, Boukerche A, Pazzi R (2010) Scheduling for scalable energy-efficient localization in mobile ad hoc networks. In: Proceedings of IEEE SECON 2010. Boston, MA
14. Gupta V, Chung TH, Hassibi B, Murray RM (2006) On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. Automatica 42(2):251–260

15. Xiao W, Wu J, Xie L, Dong L (2006) Sensor Scheduling for Target Tracking in Networks of Active Sensors. Acta Automatica Sinica 32(6):922–928
16. Johnson DB, Maltz DA (1996) Dynamic source routing in ad hoc wireless networks. In: Imielinski T, Korth H (eds) Mobile computing, vol 353. Kluwer Academic Publishers
17. Atmel Inc. Atmel Atmega128L Low-Power Mricrocontroller. Available at: http://www.atmel.com/dyn/resources/proddocuments/2467S.pdf. Accessed 25 Oct 2008
18. Texas Instruments. CC2420 Single-Chip 2.4 GHz RF Transceiver. Available at: http://focus.ti.com/docs/prod/folders/print/cc2420.html. Accessed 25 Oct 2008

Fan Zhang received his B.Sc degree in Control Science and Engineering from Zhejiang University at 2009. Now he is working towards the Master degree under the supervise of Prof. Jiming Chen. His current research interests are in target tracking and active sensor scheduling. zhangfan0510@gmail.com.

Jiming Chen (M'08) received B.Sc degree and Ph.D degree both in Control Science and Engineering from Zhejiang University in 2000 and 2005, respectively. He was a visiting researcher at INRIA in 2006, National University of Singapore in 2007, and University of Waterloo from 2008 to 2010. Currently, he is a full professor with Department of control science and engineering, and the coordinator of group of Networked Sensing and Control in the State Key laboratory of Industrial Control Technology at Zhejiang University, China. His research interests are estimation and control over sensor network, sensor and actuator network, target tracking in sensor networks, optimization in mobile sensor network. He currently serves associate editors for several international Journals, e.g., Wireless Communication and Mobile Computing (Wiley). He also serves as a Co-chair for Ad hoc and Sensor Network Symposium, IEEE Globecom 2011, IEEE MASS 2011 Publicity Co-Chair, IEEE DCOSS 2011 Publicity Co-Chair, etc., and TPC member for IEEE ICDCS 2010, IEEE MASS 2010, IEEE INFOCOM 2011, etc. E-mail: jmchen@ieee.org.



Hongbin Li received his B.E in 2005 and Ph.D in 2010 both in Control Science and Engineering from Zhejiang University. He is currently working at MicroStrategy China Technology Center as a software performance engineer. E-mail: hungbun.lee@gmail.com.

Youxian Sun received the Diploma from the Department of Chemical Engineering, Zhejiang University, China, in 1964. He joined the Department of Chemical Engineering, Zhejiang University, in 1964. From1984 to1987, he was an Alexander Von Humboldt Research Fellow, and Visiting Associate Professor at University of Stuttgart, Germany. He has been a full professor at Zhejiang University since 1988. In 1995, he was elevated to an Academician of Chinese Academy of Engineering. His current research interests include modeling, control and optimization of complex systems, robust control design and its application. He is author and co-author of 450 journal and conference papers. He is currently the director of institute of industrial process control and national engineering research center of industrial automation, Zhejiang University. He is President of Chinese Association of Automation, also served as Vice-Chairman of IFAC Pulp and Paper Committee, and Vice-President of China Instrument and Control Society. E-mail: yxsun@iipc.zju.edu.cn.

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, UWB wireless communications networks, wireless network security, wireless body area networks and vehicular ad hoc and sensor networks. He is a co-author of three books, and has published more than 400 papers and book chapters in wireless communications and networks, control and filtering. Dr. Shen has served as the Technical Program Committee Chair for IEEE VTC'10, the Tutorial Chair for IEEE ICC'08, the Technical Program Committee Chair for IEEE Globecom'07, the General Co-Chair for Chinacom'07 and QShine'06, the Founding Chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He has also served as a Founding Area Editor for IEEE Transactions on Wireless Communications; Editor-in-Chief for Peer-to-Peer Networking and Application; Associate Editor for IEEE Transactions on Vehicular Technology; Computer Networks; and ACM/Wireless Networks, Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, and a Distinguished Lecturer of IEEE Communications Society. E-mail: xshen@bbcr.uwaterloo.ca.