Sensor Scheduling for Target Tracking Using Approximate Dynamic Programming

ZINING ZHANG, GANLIN SHAN
Electronic Engineering Department
Shijiazhuang Mechanical Engineering College
No. 97, Hepingxilu Road, Shijiazhuang, 050003
P R CHINA
iron_zay@126.com, shanganlin@163.com

Abstract: - To trade off tracking accuracy and interception risk in a multi-sensor multi-target tracking context, we study the sensor-scheduling problem where we aim to assign sensors to observe targets over time. Our problem is formulated as a partially observable Markov decision process, and this formulation is applied to develop a non-myopic sensor-scheduling scheme. We resort to extended Kalman filtering for information-state estimation and use unscented transformation for trajectory sampling in order to reduce the number of samples required for *Q*-value approximation. We make decision using a simulation-based approximate dynamic programming method called policy rollout, which is implemented by means of receding horizon control. The effectiveness of our approach is substantiated through an example in which multiple sensors are deployed to track a single target.

Key-Words: - Non-myopic sensor scheduling, Partially observable Markov decision process, Interception risk, Policy rollout, Unscented transformation

1 Introduction

We assume that a sensor network consisting of several sensors collects target information in a battlefield. A typical purpose of sensor scheduling is how to activate sensors to trade off quality of service and usage cost for the network. Traditional scheduling methods that are generally myopic optimize the immediate reward to evaluate the instantaneous benefit resulting from a single action [1-3]. However, the non-myopic scheduling scheme outperforms the myopic scheduling scheme in some special cases because it considers the long-term benefit resulting from a sequence of actions [4-6]. In the literatures [7] and [8], the balance of tracking accuracy and power consuming has been considered. Furthermore, the work in [9] used a multi-mode sensor to track as more targets with high-priority as possible based on a presettable tracking requirement. Schneider and Chong attempt to track and discriminate targets of a specified type to desired levels using non-myopic scheduling approach [10]. Moreover, another alternative approach called multi-armed bandits (MAB) is applied to solve the non-myopic sensor-scheduling problem in recent years [11, 12]. Though MAB formulation is a special case of POMDP and its analytical solution is feasible, most sensor-scheduling problems are difficult to formulate using this MAB structure because several restrictive constraints on the scheduling problem must be satisfied.

The signal interception can betray the existence and location of the sensor to the enemy and thus increase the vulnerability of the sensor, so we regard the interception risk as the sensor usage cost and focus on the trade-off between tracking accuracy and interception risk. Our research is inspired by the works [13-15] in which sensors are scheduled by incorporating Monte Carlo sampling and particle filtering into the rollout method to trade off tracking error and power consumption, but there exist some appealing features in this work. First, we try to trade off tracking accuracy and interception risk under the POMDP framework. Second, we propose a novel sampling method based on unscented transformation to decrease the number of samples when simulating information-state trajectories for *Q*-value estimation in the rollout method.

2 Problem Formulation

For clarity and ease of presentation, we made some assumptions:

(1) All targets are moving independently, and each target must be tracked by only one sensor at each time step.

(2) All sensors are independent of each other, and each sensor can track no more than one target at each time step.
(3) There are *M* sensors located at fixed positions to track *N* targets (*M*>*N*).
(4) All targets and sensors work in a 2D plane.

The POMDP problem is a special Markov decision process (MDP) in which the underlying state is unknown and the observations can yield uncertain information about the underlying state. We usually describe a POMDP by some elements including underlying state space *S*, action space *U*, observation space *Z*, state transition law $p(S_k|S_{k-1}\ u_k)$, observation law $p(Z_k|S_k\ u_{k-1})$, initial state distribution p_0 and one-step reward function $r(S_k,u_k)$.

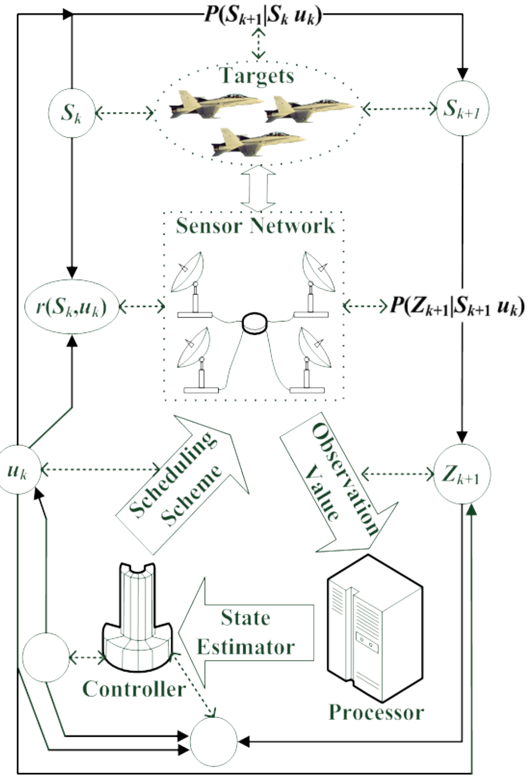


Fig.1 POMDP for sensor scheduling

We assume that past decisions, plus past observations, are available to select the current action. Let η_k ={$Z_1,\ldots,Z_k,u_0,\ldots,u_{k-1}$} denotes the information set including past information up to time step *k*. Starting from the initial state S_0 with known distribution p_0, a POMDP evolves as follows. At current time step *k*, the system state $S_k \in S$ and the observation $Z_k \in Z$ of targets are available. Select the action $u_k \in U$ based on information set η_k to obtain the next observation, and then the one-step reward $r(S_k,u_k)$ is incurred. After that the underlying state transits from S_k to S_{k+1} according to the state transition law $p(S_{k+1}|S_k\ u_k)$, and an observation Z_{k+1} can be obtained in terms of the observation law $p(Z_{k+1}|S_{k+1}\ u_k)$. The action u_k and the observation Z_{k+1} are added to the information set η_k to generate η_{k+1}. Fig.1 reveals the causal sequence of a POMDP for sensor scheduling.

We formulate our scheduling problem as a continuous-state discrete-time POMDP in which the elements are stated next. (See more details about POMDP in [16] .)

2.1 Action, State and State Transition Law

To denote the sensor assignment at time step *k*, the action $u_k=(u_{ji}^k)_{M\times N}$ is a *M*×*N* matrix, where $u_{ji}^k=1$ or $u_{ji}^k=0$ indicates scheduling decision on whether sensor *j* is activated for observing target *i* from time step *k* to time step *k*+1. Only one of the elements in each column equals to unity, and at most one of the elements in each row equals to unity.

The underlying state vector S_k is comprised of the dynamic state X_k and state estimator $\hat{X}_k$. At time step *k*, the system state is written as

$$S_k=\left[X_k\ \hat{X}_k\right]^{\mathrm{T}},\ X_k=\left[X_k^1\ \cdots\ X_k^N\right]^{\mathrm{T}}$$

Here X_k^i =[$x_k^i\ \dot{x}_k^i\ y_k^i\ \dot{y}_k^i$]$^{\mathrm{T}}$ represents the dynamic state of target *i* at time *k*, including target position and velocities in Cartesian coordinates. Notice that the state estimator $\hat{X}_k$ of X_k evolves via extended Kalman filtering (EKF) in this paper. The state transition law $p(S_{k+1}|S_k\ u_k)$ is given by

$$p(S_k|S_{k-1}\ u_{k-1})=p(X_k|X_{k-1})p\left(\hat{X}_k|\hat{X}_{k-1}\ u_{k-1}\right) \quad (1)$$

where the dynamic state transition law $p(X_k|X_{k-1})$ is defined through the target dynamics equation that is a nearly constant velocity model in our simulation.

$$X_k=F\cdot X_{k-1}+\Gamma\cdot\omega_{k-1} \quad (2)$$

$$F=diag\left(F_1,\cdots,F_N\right),\ \Gamma=diag\left(\Gamma_1,\cdots,\Gamma_N\right)$$

$$F_{i=1,\cdots,N}=\begin{bmatrix}1 & T_s & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & T_s\\ 0 & 0 & 0 & 1\end{bmatrix},\ \Gamma_{i=1,\cdots,N}=\begin{bmatrix}T_s^2/2 & 0\\ T_s & 0\\ 0 & T_s^2/2\\ 0 & T_s\end{bmatrix}$$

$$\omega_{k-1}=\left[\omega_{k-1}^1\cdots\omega_{k-1}^N\right]^{\mathrm{T}},\omega_{k-1}^i=\left[\omega_{x,k-1}^i\ \ \omega_{y,k-1}^i\right]^{\mathrm{T}}$$

where T_s is the sampling interval, and ω_{k-1} is the unrelated zero-mean Gaussian process noise with the covariance matrix *W*.

Based on EKF, the estimator transition law $p(\hat{X}_k|\hat{X}_{k-1}\ u_{k-1})$ is given by

$$\begin{aligned}&p\left(\hat{X}_k|\hat{X}_{k-1}\ u_{k-1}\right)\\ &=\int p\left(\hat{X}_k|Z_k\ \hat{X}_{k-1}\ u_{k-1}\right)p\left(Z_k|\hat{X}_{k-1}\ u_{k-1}\right)d_{Z_k}\end{aligned} \quad (3)$$

$$p\left(\hat{X}_k|Z_k\ \hat{X}_{k-1}\ u_{k-1}\right)=\begin{cases}1 & Z_k,\hat{X}_{k-1},u_{k-1}\overset{\text{EKF}}{\Rightarrow}\hat{X}_k'=\hat{X}_k\\ 0 & Z_k,\hat{X}_{k-1},u_{k-1}\overset{\text{EKF}}{\Rightarrow}\hat{X}_k'\neq\hat{X}_k\end{cases}$$

$$p\left(Z_k \mid \hat{X}_{k-1}\ u_{k-1}\right)=\int p\left(Z_k \mid X_k\ u_{k-1}\right)p\left(X_k \mid \eta_{k-1}\ u_{k-1}\right)d_{X_k}$$

where $p(Z_k|X_k\ u_{k-1})$ is the observation law, and the conditional probability $p(X_k|\eta_{k-1}u_{k-1})$ constitutes the predicted information state explained later.

2.2 Observation and Observation Law

The overall observation Z_k of dynamic state X_k can be written as

$$Z_k=\left[Z_k^1\ \cdots\ Z_k^N\right]^{\mathrm{T}}=h\left(X_k\right)+u_{k-1}^{\mathrm{T}}\otimes I\cdot\nu_k \qquad (4)$$

$$h\left(X_k\right)=\left[h_1\left(X_k^1\right)\ \cdots\ h_N\left(X_k^N\right)\right]^{\mathrm{T}},\ \ \nu_k=\left[\nu_k^1\ \cdots\ \nu_k^M\right]^{\mathrm{T}}$$

Here the symbol $\otimes$ denotes the Kronecker product, I is the identity matrix, $h_i(\cdot)$ stands for the true value of nonlinear observation of target *i*, and ν_k^j is the unrelated zero-mean Gaussian observation noise of sensor *j*. The nonlinear observation of a target is obtained in polar coordinates and can be written as

$$h_i(X_k^i)=\left[r_k^i\ \ \theta_k^i\ \ \dot{r}_k^i\right]^{\mathrm{T}} \qquad (5)$$

with

$$r_k^i=\sqrt{\left(x_k^i-sc_j\left(x\right)\right)^2+\left(y_k^i-sc_j\left(y\right)\right)^2}$$

$$\theta_k^i=\tan^{-1}\left(\frac{y_k^i-sc_j\left(y\right)}{x_k^i-sc_j\left(x\right)}\right)$$

$$\dot{r}_k^i=\frac{\left(x_k^i-sc_j\left(x\right)\right)\dot{x}_k^i+\left(y_k^i-sc_j\left(y\right)\right)\dot{y}_k^i}{\sqrt{\left(x_k^i-sc_j\left(x\right)\right)^2+\left(y_k^i-sc_j\left(y\right)\right)^2}}$$

Here r_k^i, θ_k^i and $\dot{r}_k^i$ are the range, the azimuth angle and the range rate of target *i*, and $\nu_k^{u_{k-1}^i=j}$ =[$\nu_k^{r,j}$ $\nu_k^{\theta,j}$ $\nu_k^{\dot{r},j}$]$^{\mathrm{T}}$ are the corresponding Gaussian noise. It is noted that the coordinates ($sc_j(x)$, $sc_j(y)$) are the position of sensor *j*, and the coordinates (x_k^i, y_k^i), ($\dot{x}_k^i$, $\dot{y}_k^i$) are the position and velocities of target *i*, respectively. Note that the observation law $p(Z_k|S_k\ u_{k-1})=p(Z_k|X_k\ u_{k-1})$ is defined by the expressions (4) and (5).

2.3 One-step Reward and Objective Function

The signal interception occurs as overlaps happen to multiple window functions [17, 18], so that we define three window functions as follows.

(1) The window function of pulse signal for sensor *j* is defined as $F_p^j(T_p^j,\tau_p^j)$, where T_p^j is the pulse repetition interval, and τ_p^j is the pulse width.

(2) The window function of scanning antenna for target *i* is defined as $F_a^i(T_a^i=360/\gamma_a^i,\tau_a^i=\psi_a^i/\gamma_a^i)$, where γ_a^i is the antenna rotation rate, and ψ_a^i is the antenna beam width.

(3) The window function of scanning receiver for target *i* is defined as $F_r^i(T_r^i,\tau_r^i=B_iT_r^i/f_i)$, where T_r^i is the time required for the receiver to scan across a frequency band f_i, and B_i is the receiver passband.

Therefore, the interception probability $P_I^{u_k^i}$ of sensor *j* is derived from the three window functions above.

$$P_I^{u_k^i=j}=1-\kappa e^{-\frac{T_s}{T_o}} \qquad (6)$$

$$T_o=\frac{T_p^jT_a^iT_r^i}{\left(\tau_p^j-d_{ji}\right)\left(\tau_a^i-d_{ji}\right)+\left(\tau_p^j-d_{ji}\right)\left(\tau_r^i-d_{ji}\right)+\left(\tau_a^i-d_{ji}\right)\left(\tau_r^i-d_{ji}\right)} \qquad (7)$$

Here $\kappa\approx 1$ for radar examples, and d_{ji} is the minimum interception duration to declare a valid interception. Note that u_k^i is the *i-th* column vector of u_k, denoting which sensor is assigned to observe target *i*.

By combining the tracking accuracy with the interception risk, the one-step reward is defined as

$$r(S_k,u_k)=\left\|X_k-\hat{X}_k\right\|_1-\alpha\ln\prod_{i=1}^{N}\left(1\text{-}P_I^{u_k^i}\right) \qquad (8)$$

Here $\|\cdot\|_1$ denotes the L$_1$ norm of a vector. The second term in (8) represents the interception cost of all selected sensors with no interception during the sampling interval. The coefficient α is the balance factor that adjusts the impact of interception cost on the one-step reward.

2.4 Information-state Transformation

The standard POMDP problem generally has to be converted into MDP problem for solutions using so-called information state χ_k={ $p(X_k\mid\eta_k)$ for all X_k }, which is the probability distribution of state X_k conditioned on the information set η_k including the history of past measurements and actions up to time step *k*. Consequently, the equivalent MDP with the underlying state χ_k is called as information-state Markov decision process (IS-MDP). We will depict the state transition law $p(\chi_k\mid\chi_{k-1}\ u_{k-1})$, the one-step reward $R(\chi_k,u_k)$ and the objective function *J* for our IS-MDP formulation.

The state transition law $p(\chi_k\mid\chi_{k-1}\ u_{k-1})$ of our IS-MDP problem is written as

$$\begin{aligned}&p\left(\chi_k\mid\chi_{k-1}\ u_{k-1}\right)\\&=\int p\left(\chi_k\mid Z_k\ \chi_{k-1}\ u_{k-1}\right)p\left(Z_k\mid\chi_{k-1}\ u_{k-1}\right)d_{Z_k}\end{aligned} \qquad (9)$$

$$p\left(\chi_k\mid Z_k\ \chi_{k-1}\ u_{k-1}\right)=\begin{cases}1 & Z_k,\chi_{k-1},u_{k-1}\Rightarrow\chi_k'=\chi_k\\0 & Z_k,\chi_{k-1},u_{k-1}\Rightarrow\chi_k'\neq\chi_k\end{cases}$$

$$p(Z_k|\chi_{k-1}\ u_{k-1})=\int p\left(Z_k|X_k\ u_{k-1}\right)p\left(X_k|\eta_{k-1}\ u_{k-1}\right)d_{X_k}$$

where Z_k, χ_{k-1} and u_{k-1} are devoted to acquiring χ_k' using Bayesian rule as

$$p(X_k|\eta_k)=\frac{p\left(Z_k|X_k\ u_{k-1}\right)p\left(X_k|\eta_{k-1}\ u_{k-1}\right)}{\int p\left(Z_k|X_k\ u_{k-1}\right)p\left(X_k|\eta_{k-1}\ u_{k-1}\right)d_{X_k}} \qquad (10)$$

$$p\left(X_k|\eta_{k-1}\ u_{k-1}\right)=\int p\left(X_k|X_{k-1}\right)p\left(X_{k-1}|\eta_{k-1}\right)d_{X_{k-1}} \quad (11)$$

Here the symbol $\bar{\chi}_k=\{\ p(X_k\,|\,\eta_{k-1}\ u_{k-1})$ for all X_k } is the predicted information state which is the representation of information state before obtaining observation Z_k. It is theoretically possible that the evolutional information state could be iteratively computed by the equations (9), (10) and (11). However, we can observe that the information state χ_k can be approximated to Gaussian distribution $N(\hat{X}_k,P_k)$ duo to Gaussian noise and nonlinear observation. It is feasible to represent the information state through the sufficient statistics ($\hat{X}_k$, P_k) we keep track of which using EKF. Therefore, the state transition law $p(\chi_k\,|\,\chi_{k-1}\ u_{k-1})$ is equivalent to the statistic transition law $p((\hat{\hat{X}}_k,P_k)\,|\,(X_{k-1},P_{k-1})\ u_{k-1})$ in EKF.

Derived from the one-step reward in (8), the mathematical definition of the one-step reward in IS-MDP is given by

$$\begin{aligned} R\left(\chi_k,u_k\right)&=\int r\left(S_k,u_k\right)p\left(X_k\,|\,\eta_k\right)d_{X_k} \\ &=trace\left(sqrt(P_k)\right)-\alpha\cdot\ln\prod_{i=1}^{N}\left(1\text{-}P_I^{u_k^i}\right) \end{aligned} \quad (12)$$

where P_k is the covariance matrix in EKF, and the symbol $sqrt\left(P_k\right)$ denotes the matrix consisting of the square roots of all elements in P_k. Thus, the objective function is the expected total one-step reward over a horizon of *H* time steps

$$\begin{aligned} J&=\mathop{E}_{Z_1,\cdots,Z_{H\text{-}1}}\left[\sum_{k=0}^{H-1}R\left(S_k,u_k\right)\right] \\ &=\mathop{E}_{Z_1,\cdots,Z_{H\text{-}1}}\sum_{k=0}^{H-1}\left[trace\left(sqrt(P_k)\right)-\alpha\ln\prod_{i=1}^{N}\left(1\text{-}P_I^{u_k^i}\right)\right] \\ &=\mathop{E}_{Z_1,\cdots,Z_{H\text{-}1}}\left\{\sum_{k=0}^{H-1}\left[trace\left(sqrt(P_k)\right)\right]\right\}-\alpha\ln\prod_{k=0}^{H-1}\prod_{i=1}^{N}\left(1\text{-}P_I^{u_k^i}\right) \end{aligned} \quad (13)$$

where $\prod_{k=0}^{H-1}\prod_{i=1}^{N}(1\text{-}P_I^{u_k^i})$ is the probability of all scheduled sensors with no interception over *H* time steps. We have represented our problem in the form of information state, and our goal is to find an optimal policy $\pi=\left\{\pi_0,\cdots,\pi_{H-1}\right\}$, which is a sequence of mappings $u_k=\pi_k(\chi_k)$ to minimize the objective function *J* in (13).

3 Approximate Solution

The main issue to solve our problem is the curse of dimensionality and computational complexity. We use a simulation-based approximate programming method called policy rollout [19, 20], which approximately solve a POMDP based on the *Q*-value approximation.

3.1 *Q*-value Approximation

It is known that the *Q*-value [21] is defined as

$$\begin{aligned} &Q_{H-k}\left(\chi_k,u_k\right) \\ &=R\left(\chi_k,u_k\right)+E\left(J_{H-k-1}^{*}\left(\chi_{k+1}\right)|\,\chi_k,u_k\right) \\ &=R\left(\chi_k,u_k\right)+\int J_{H-k-1}^{*}\left(\chi_{k+1}\right)p\left(\chi_{k+1}\,|\,\chi_k,u_k\right)d_{\chi_{k+1}} \end{aligned} \quad (14)$$

where $Q_{H-k}(\chi_k,u_k)$ is the *Q*-value in a horizon of *H*-*k* time steps, and $J_{H-k-1}^{*}(\chi_{k+1})$ is the optimal value over *H*-*k*-1 time steps given the next information state χ_{k+1}. The first term in the *Q*-value expresses the immediate reward at time step *k* and the second term denotes the expected reward in future. The rollout method is to give *Q*-value approximation by surrogating $J_{H-k-1}^{*}(\chi_{k+1})$ with $J_{H-k-1}^{\pi_b}(\chi_{k+1})$ incurred by a base policy π_b. The base policy is a suboptimal heuristic mapping that is easy to implement.

Suppose that the horizon length *H* is sufficiently large, the remaining horizon is still *H* steps away regardless of the time step *k*. This leads to the receding horizon control [22], which estimates the *Q*-value at each time step *k* and chooses an optimal action u_k^*.

$$Q_H\left(\chi_k,u_k\right)\approx R\left(\chi_k,u_k\right)+E\left(J_{H-1}^{\pi_b}\left(\chi_{k+1}\right)\middle|\chi_k,u_k\right) \quad (15)$$

$$u_k^*=\arg\min_{u_k}\ Q_H(\chi_k,u_k) \quad (16)$$

Here the base policy is the closest distances policy (CDP), which assigns the sensors with the minimum sum of distances between the sensor positions and the estimated positions of the targets

$$\pi_b\left(\chi_k\right)=\mathop{\arg\min}_{u_k=\left[u_k^1\cdots u_k^N\right]}\left(\sum_{i=1}^{N}\sqrt{\left(sc_{u_k^i}(x)-\hat{x}_k^i\right)^2+\left(sc_{u_k^i}(y)-y_k^i\right)^2}\right) \quad (17)$$

where $\hat{x}_k^i$ and $\hat{y}_k^i$ are the position estimators of target *i* in *x* and *y* directions.

3.2 Rollout Based on UT Sampling

The rollout method is implemented with ease, but computationally intractable because there must be enough samples to evaluate the expected future rewards. In fact, the computational requirement of the rollout method depends on the length of the simulation runs and the number of samples required for the *Q*-value approximation. Traditional rollout methods employ Monte Carlo sampling to estimate the *Q*-value by averaging the cumulative rewards from the *N* Monte Carlo simulation runs. The resulting action minimizes

$$Q_H(\chi_k,u_k)\approx R(\chi_k,u_k)+N^{-1}\cdot\sum_{n=1}^{N}J_{H-1}^{\pi_b}(\chi_{k+1}^n|\chi_k\ u_k)$$

where χ_{k+1}^n is the *n-th* information state sample derived from the current state χ_k. Note that Monte Carlo sampling has to collect a large number of random samples to evaluate the *Q*-value without considering the horizon length and the number of the targets. Moreover, Monte Carlo sampling is a time-consuming random sampling method, and the use of more samples in the rollout method induces expensive computation. Unscented transformation (UT) was developed as a method to propagate mean and covariance information through nonlinear transformations [23], but we use it to afford the initial state samples in the rollout method. Unlike Monte Carlo sampling, unscented transformation sampling could engender numbers of deterministic samples according to the concrete tracking scenario, resulting in large decrease on the samples especially in a tracking mission with a small number of targets to track. Fig.2 illustrates the rollout algorithm based on unscented transformation sampling, which we name as UTR for short (see Fig. 2 in Appendix).

Starting from the initial state sample $\tilde{X}_k^n$, we propagate the state sample and observe it to simulate the evolutional information state by EKF [24]. Each initial state sample gives birth to a trajectory over H-1 steps to produce the accumulated future rewards. Specifically, the candidate action u_k is selected at the first step of the H-length horizon, and the base policy π_b is used for the remaining time steps to generate the future action sequence. We summarize the UTR algorithm systematically in Algorithm 1.

```
Algorithm 1 Rollout based on UT sampling
1. Given $\chi_k$ at current time step k, use (18) and (19) to sample the integrated sigma points $\varepsilon_k^n$ and the sigma weights $\beta_n$.
2. Select a candidate action $u_k$ and compute immediate reward using (12).
3. Let t=k, $(\hat{\tilde{X}}_k^n, \tilde{P}_k^n) = (\hat{X}_k, P_k)$ for $n = 0, \cdots, 2\sigma$.
4. Process all sigma points in parallel through the following steps.
    Propagate system state samples via (20).
    Acquire the measurements by (21).
    Update information state for sigma point n using EKF in (22) and (23).
5. Starting at time step k+1 from $(\hat{\tilde{X}}_{k+1}^n, \tilde{P}_{k+1}^n)$ and $\tilde{X}_{k+1}^n$, transact the following procedure for $n = 0, \cdots, 2\sigma$.
    for t= k+1 to k+H-1
        Determine the action $u_t$ through CDP.
        Calculate the one-step reward in (12).
        Implement step 4.
    end
6. Evaluate the approximate Q-value for information state trajectory in (24)
7. Enumerate all actions and choose the best action minimizing the Q-value.
```

Notice that we implement Algorithm 1 at each time step k to make scheduling decision in real time. The equations mentioned in Algorithm 1 are given as follows.

$$\varepsilon_k^n = \begin{bmatrix} \tilde{X}_k^n & \underbrace{\tilde{\omega}_k^n \ \cdots \ \tilde{\omega}_{k+H-1}^n}_{\tilde{\omega}_n} & \underbrace{\tilde{v}_{k+1}^n \ \cdots \ \tilde{v}_{k+H}^n}_{\tilde{v}_n} \end{bmatrix}^{\mathrm{T}}$$

$$= \begin{cases} \varepsilon_k^0 + (\sqrt{(\sigma+\lambda)\rho_k^0})_n, & n = 1, \cdots, \sigma \\ \varepsilon_k^0 - (\sqrt{(\sigma+\lambda)\rho_k^0})_n, & n = \sigma+1, \cdots, 2\sigma \end{cases} \tag{18}$$

$$\varepsilon_k^0 = \begin{bmatrix} \hat{X}_k & \underbrace{0 \ \cdots \ 0}_{E(\tilde{\omega}_n)} & \underbrace{0 \ \cdots \ 0}_{E(\tilde{v}_n)} \end{bmatrix}^{\mathrm{T}}$$

$$\rho_k^0 = diag\left(P_k, \underbrace{W, \cdots, W}_{E\left(\tilde{\omega}_n \cdot \tilde{\omega}_n{}^{\mathrm{T}}\right)}, \underbrace{V, \cdots, V}_{E\left(\tilde{v}_n \cdot \tilde{v}_n{}^{\mathrm{T}}\right)} \right)$$

where σ is the dimension of the sigma point, $(\sqrt{(\sigma+\lambda)\rho_0^k})_n$ denotes the *n-th* row of the matrix square root and $\lambda = \xi^2(\sigma+\varphi)-\sigma$, ξ is the spreading parameter, and φ is a factor usually chosen to be $3-\sigma$ [25].

$$\beta_{n=0} = \frac{\lambda}{\lambda+\sigma}, \ \beta_{n=1,\cdots,2\sigma} = \frac{1}{2(\lambda+\sigma)} \tag{19}$$

$$\tilde{X}_{t+1}^n = F \cdot \tilde{X}_t^n + \Gamma \cdot \omega_t^n \tag{20}$$

$$\tilde{Z}_{t+1}^n = h\left(\tilde{X}_{t+1}^n\right) + u_t^{\mathrm{T}} \otimes I \cdot \tilde{v}_{t+1}^n \tag{21}$$

$$\bar{\tilde{X}}_{t+1}^n = F \cdot \hat{\tilde{X}}_t^n, \ \bar{\tilde{P}}_{t+1}^n = F \cdot \tilde{P}_t^n \cdot F^{\mathrm{T}} + \Gamma \cdot W \cdot \Gamma^{\mathrm{T}} \tag{22}$$

$$\hat{\tilde{X}}_{t+1}^n = \bar{\tilde{X}}_{t+1}^n + K\left(\tilde{Z}_{t+1}^n - h\left(\bar{\tilde{X}}_{t+1}^n\right)\right), \ \tilde{P}_{t+1}^n = \left(I - KH\right)\bar{\tilde{P}}_{t+1}^n \tag{23}$$

$$H = \left.\frac{\partial h}{\partial X}\right|_{X=\bar{\tilde{X}}_{t+1}^n}, \ K = \bar{\tilde{P}}_{t+1}^n H^{\mathrm{T}}\left(H\bar{\tilde{P}}_{t+1}^n H^{\mathrm{T}} + V_{u_t}\right)^{-1}$$

where V_{u_t} is the covariance matrix of observation noise of all the assigned sensors.

$$Q_H(\chi_k, u_k) \approx R(\chi_k, u_k) + \sum_{n=0}^{2\sigma}\left[\beta_n \sum_{t=k+1}^{k+H-1} R(\tilde{\chi}_t^n, u_t)\right] \tag{24}$$

4 Simulation Experiment

We aim to assign redundant sensor assets to track multiple targets, but it is burdensome to enumerate all the candidate actions in Algorithm 1. Clearly, the number of the candidate actions is $\prod_{i=1}^{N}(M-i+1)$ for the exhaustive search. To reduce the number of the candidate actions, we plan to adopt a partition strategy in which M sensors are divided into N groups and each group consisting of m_i sensors is responsible for tracking a single target. This strategy only needs to search for $\prod_{i=1}^{N} m_i$ candidate actions. It is obvious that m_i>1 and $\sum_{i=1}^{N} m_i = M$, so that the following inequation holds

$$M - i + 1 - m_i = \sum_{i'=1,\ i'\neq i}^{N} m_{i'} - i + 1 > N - i \geq 0 \tag{25}$$

Note that (25) has proven that $\prod_{i=1}^{N}(M-i+1) > \prod_{i=1}^{N} m_i$.

That is to say, the partition strategy is more tractable. Fig.3 shows the multi-target tracking scenario of the partition strategy for *M*=10, *N*=3.

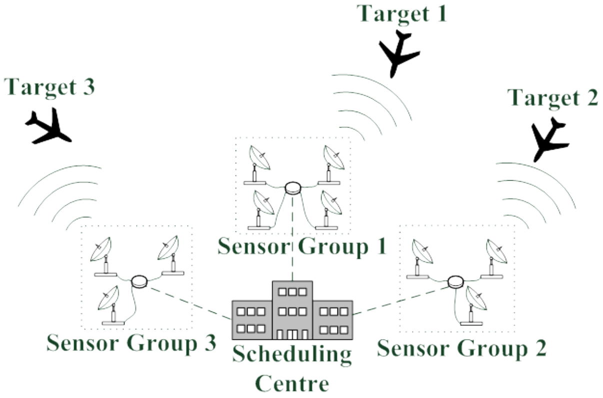


Fig.3 Illustration of the partition strategy

In view of the partition strategy, we verify the UTR algorithm via an example in which four sensors (Sensor A, B, C and D) are located at the fixed positions to track a single target whose process noise is $\sigma_{\ddot{x}} = \sigma_{\ddot{y}}$ =7 m/s^2. The two window functions of the target are F_r (T_r =100 ms, τ_r =0.05 ms) and F_a (T_a =200 ms, τ_a =27.8 ms). The particular parameters in the simulation are displayed as follows: the horizon length *H*=8, the spreading parameter ξ =0.1 and the balance factor α =200. To demonstrate the performance of our approach, the CDP and the minimum one-step reward (MOR) method are involved. We use two scenarios for the performance analysis. All sensors have the same observation noise in scenario 1 but Sensor A is more prone to be intercepted than the other sensors, while all sensors have the same interception probability in scenario 2 but Sensor C has the smallest observation noise. The sensor parameters in the experiment are shown in Table 1.

Table 1 Sensor parameters in the simulation

	Sensor Parameter	Scenario 1	Scenario 2
Error Statistics	σ_r =200 m σ_θ =2° $\sigma_{\dot{r}}$ =30 m/s	Sensor A, B, C, D	Sensor A, B, D
	σ_r =100 m σ_θ =1° $\sigma_{\dot{r}}$ =10 m/s		Sensor C
Window Function	F_p(T_p=2, τ_p=0.02) ms	Sensor A	
	F_p(T_p=1.88, τ_p=0.1) ms	Sensor B,C,D	Sensor A, B, C, D
Sampling Interval	T_s=1 s	Sensor A	
	T_s=2 s	Sensor B, C, D	Sensor A,B, C, D

The accumulated tracking errors and interception costs from the CDP, MOR and rollout policies are shown in Fig.4 (see Fig. 4 in Appendix), which has demonstrated that our rollout policy outperforms the other two methods.. What is significant here is that our rollout policy is able to automatically trade off tracking accuracy and interception risk. The UTR largely decreases the interception risk in scenario 1 despite of a little sacrifice on the tracking accuracy, while the rollout policy reduces the tracking error with no increase in interception cost in scenario 2.

Fig.5 displays the estimated trajectories and the sensor sequences (see Fig. 5 in Appendix). Notice that the true target trajectories are shown by the solid lines, and the estimated target trajectories are shown using the marks whose shapes represent the selected sensors. In scenario 1, our rollout method avoids selecting Sensor A to decrease the interception risk. In scenario 2, our rollout method prefers selecting Sensor C to reduce tracking error.

5 Conclusion

In this paper, our multi-sensor scheduling problem is how to assign redundant sensors to track multiple targets over time to trade off tracking accuracy and interception risk. This scheduling problem is formulated as a POMDP to develop a non-myopic scheme. We parameterize the information state as Gaussian distribution and use EKF to keep track of it. To solve this continuous-state discrete-time POMDP, we introduce unscented transformation into the rollout method to sample numbers of the information-state trajectories. We choose CDP as the base policy and implement our rollout policy in the manner of RHC. At the end of this paper, we try to demonstrate the effectiveness of our scheduling scheme through a simulation, which involves multiple sensors tracking a single target. The simulation results indicate that our non-myopic scheme outperforms the myopic schemes in trading off tracking accuracy and interception risk.

References:

[1] C. Kreucher, K. Kastella, A.O. Hero, A Bayesian Method for Integrated Multi-target Tracking and Sensor Management, *Proceedings of the 6th International Conference on Information Fusion*, 2003, pp. 132–139.

[2] A. Logothetis, A. Isaksson, On Sensor Scheduling via Information Theoretic Criteria, Proceedings of the American Control Conference, 1999, pp. 2402–2406.

[3] Y. Oshman, Optimal Sensor Selection Strategy for Discrete-time State Estimators. *Aerospace*

and Electronic Systems, Vol.30, No.2, 1994, pp. 307–314.

[4] S. Ji, R. Parr, L. Carin, Nonmyopic Multiaspect Sensing with Partially Observable Markov Decision Processes, *IEEE Transactions on Signal Processing*, Vol.55, No.6, 2007, pp. 2720-2730.

[5] C. Kreucher, A.O. Hero, Non-myopic Approach to Scheduling Agile Sensors for Multistage Detection, Tracking and Identification, *Proceedings of the International Conference on Acoustics Speech and Signal*, 2005, pp. 885-888.

[6] C. Kreucher, K. Kastella, A.O. Hero, Sensor Management Using an Active Sensing Approach, *Signal Processing*, Vol. 85, 2005, pp. 607–624.

[7] H.J. Rad, B. Abolhassani, M.T. Abdizadeh, A New Adaptive Prediction-based Tracking Scheme for Wireless Sensor Networks, *Proceedings of the 7th Annual Communication Networks and Services Research Conference*, 2009, pp. 335-341.

[8] A. Pinto, Z. Zhang, X. Dong, S. Velipasalar, M.C. Vuran, M.C. Gursoy, Analysis of the Accuracy-latency-energy Tradeoff Wireless Embedded Camera Networks, *Proceedings of IEEE Conference on Wireless Communications and Networking*, 2011, pp. 2101-2106.

[9] A. Nedich, M.K. Schneider, R.B. Washburn, Farsighted Sensor Management Strategies for Move/Stop Tracking, *Proceedings of the 7th International Conference on Information Fusion*, 2005, pp. 566-573.

[10] M.K. Schneider, C. Chong, A Rollout Algorithm to Coordinate Multiple Sensor Resource to Track and Discriminate Target, *Proceedings of SPIE Signal Processing, Sensor Fusion and Target Recognition XV*, 2006.

[11] V. Krishnamurthy, Emission Management for Low Probability Intercept Sensors in Network Centric Warfare, *IEEE Transactions on Aerospace and Electronic Systems*, Vol.41, No.1, 2005, pp. 133-151.

[12] R.B. Washburn, M.K. Schneider, J.J. Fox, Stochastic Dynamic Programming Based Approaches to Sensor Resource Management, *Proceedings of the International Conference on Information Fusion*, 2002, pp. 608-615.

[13] Y. He, E.K.P. Chong, Sensor Scheduling for Target Tracking in Sensor Networks, *Proceedings of the 43rd IEEE Conference on Decision and Control*, 2004, pp. 743-748.

[14] Y. He, E.K.P. Chong, Sensor Scheduling for Target Tracking: A Monte Carlo Sampling Approach, *Digital Signal Processing*, Vol.16, 2006, pp. 533-545.

[15] Y. Li, L.W. Krakow, E.K.P. Chong, K.N. Groom, Dynamic Sensor Management for Multisensor Multitarget Tracking, *Proceedings of the 40th Annual Conference on Information Sciences and Systems*, 2006, pp. 1397-1402.

[16] O. Hernández-Lerma, *Adaptive Control Processes*, Springer, 1980.

[17] S. Stein, D. Johansen, A Statistical Description of Coincidences among Random Pulse Trains, *Proceedings of the IRE*, 1958, pp. 827-830.

[18] A.G. Self, B.G. Smith, Interception Time and Its Prediction, *IEE Proceedings of Communications, Radar and Signal Processing*, 1985, 215-220.

[19] G. Wu, E.K.P. Chong, R. Givan, Congestion Control Using Policy Rollout, *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 4825-4830.

[20] D.P. Bertsekas, D.A. Castanon, Rollout Algorithms for Stochastic Scheduling Problems, *Journal of Heuristics* Vol.5, 1999, pp. 89–108.

[21] E.K.P. Chong, C.M. Kreucher, A.O. Hero, Partially Observable Markov Decision Process Approximations for Adaptive Sensing, *Discrete Event Dynamic Systems*, Vol.19, 2009, pp. 377-422.

[22] D.Q. Mayne, H. Michalska, Receding Horizon Control of Nonlinear Systems, *IEEE Transactions on Automatic Control*, Vol.35, No.7, 1990, pp. 814-824.

[23] S.J. Julier, J.K. Uhlmann, Unscented Filtering and Nonlinear Estimation, *Proceedings of the IEEE*, 2004, pp. 401-422.

[24] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking, *IEEE Transactions on Signal Processing*, Vol.50, No.2, 2002, pp. 174-188.

[25] S. Haykin, *Kalman Filtering and Neural Networks*, John Wiley and Sons, 2001.

Appendix

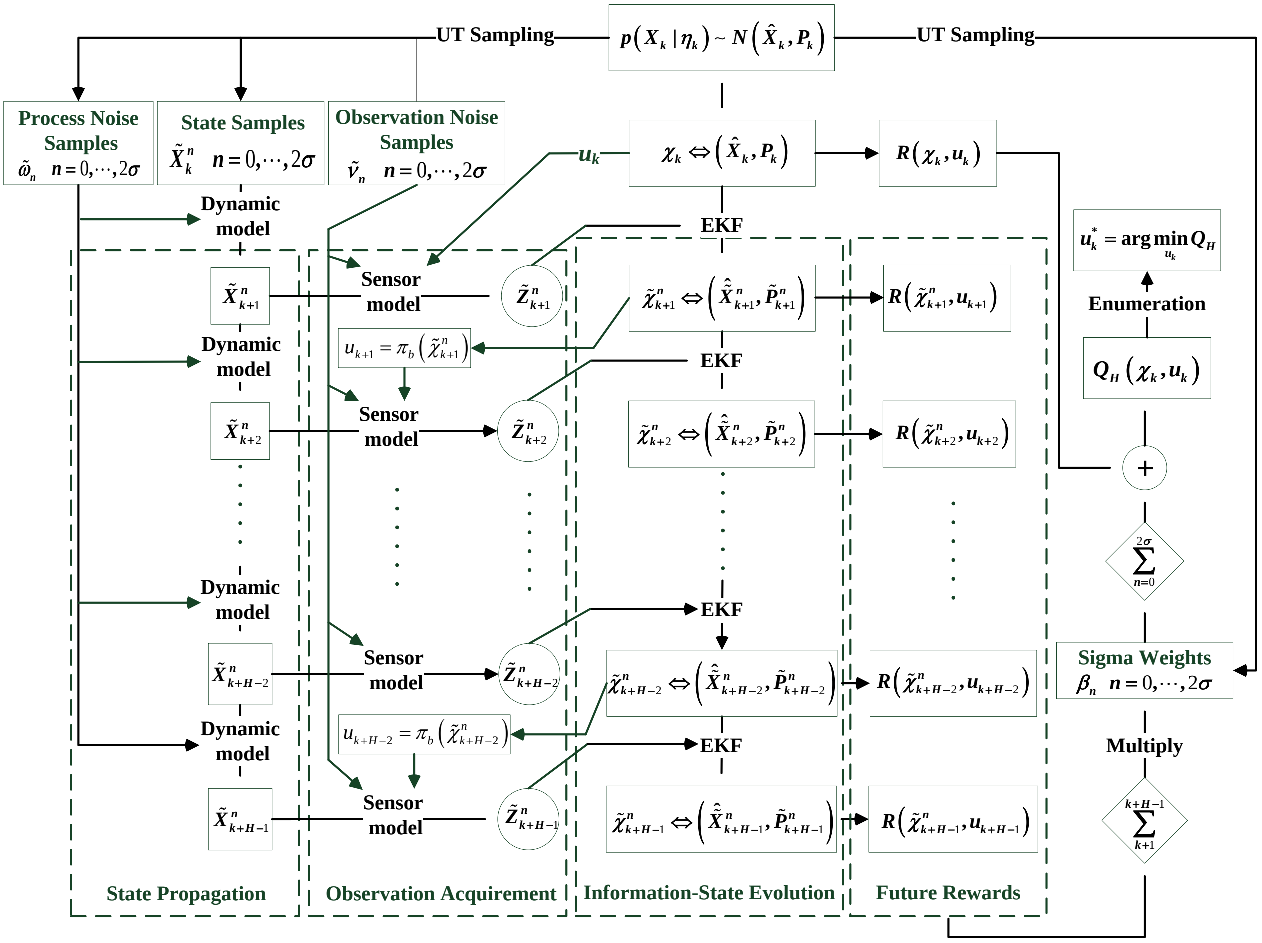
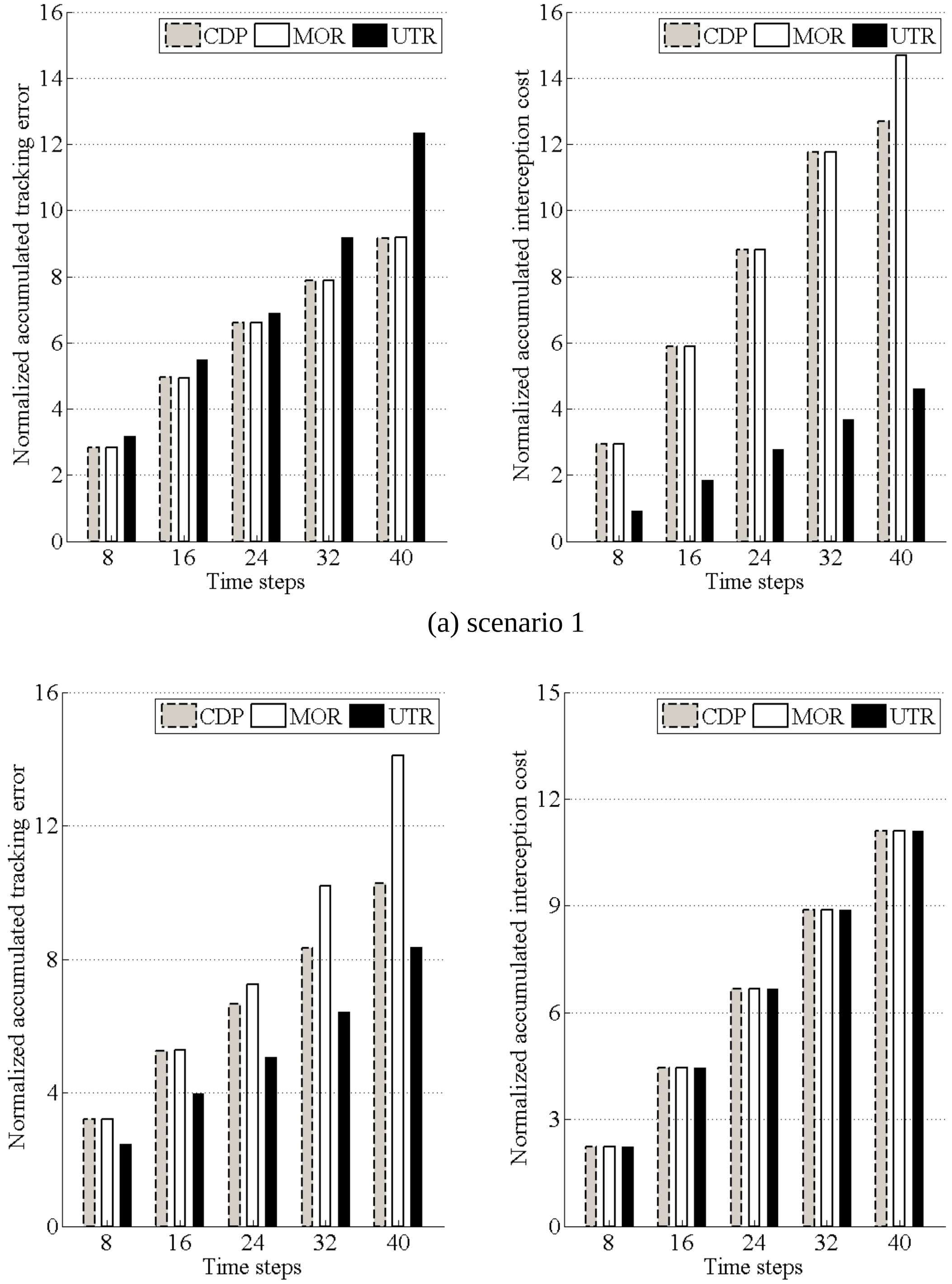

UT Sampling
$p(X_k \mid \eta_k) \sim N(\hat{X}_k, P_k)$
UT Sampling
Process Noise Samples $\tilde{\omega}_n \quad n=0,\cdots,2\sigma$
State Samples $\tilde{X}_k^n \quad n=0,\cdots,2\sigma$
Observation Noise Samples $\tilde{v}_n \quad n=0,\cdots,2\sigma$
u_k
$\chi_k \Leftrightarrow (\hat{X}_k, P_k)$
$R(\chi_k, u_k)$
$u_k^* = \arg\min_{u_k} Q_H$
Enumeration
$Q_H(\chi_k, u_k)$
+
$\sum_{n=0}^{2\sigma}$
Sigma Weights $\beta_n \quad n=0,\cdots,2\sigma$
Multiply
$\sum_{k+1}^{k+H-1}$
Dynamic model
Sensor model
EKF
$\tilde{X}_{k+1}^n$
$\tilde{Z}_{k+1}^n$
$\tilde{\chi}_{k+1}^n \Leftrightarrow (\hat{\tilde{X}}_{k+1}^n, \tilde{P}_{k+1}^n)$
$R(\tilde{\chi}_{k+1}^n, u_{k+1})$
$u_{k+1} = \pi_b(\tilde{\chi}_{k+1}^n)$
$\tilde{X}_{k+2}^n$
$\tilde{Z}_{k+2}^n$
$\tilde{\chi}_{k+2}^n \Leftrightarrow (\hat{\tilde{X}}_{k+2}^n, \tilde{P}_{k+2}^n)$
$R(\tilde{\chi}_{k+2}^n, u_{k+2})$
$\tilde{X}_{k+H-2}^n$
$\tilde{Z}_{k+H-2}^n$
$\tilde{\chi}_{k+H-2}^n \Leftrightarrow (\hat{\tilde{X}}_{k+H-2}^n, \tilde{P}_{k+H-2}^n)$
$R(\tilde{\chi}_{k+H-2}^n, u_{k+H-2})$
$u_{k+H-2} = \pi_b(\tilde{\chi}_{k+H-2}^n)$
$\tilde{X}_{k+H-1}^n$
$\tilde{Z}_{k+H-1}^n$
$\tilde{\chi}_{k+H-1}^n \Leftrightarrow (\hat{\tilde{X}}_{k+H-1}^n, \tilde{P}_{k+H-1}^n)$
$R(\tilde{\chi}_{k+H-1}^n, u_{k+H-1})$
State Propagation
Observation Acquirement
Information-State Evolution
Future Rewards

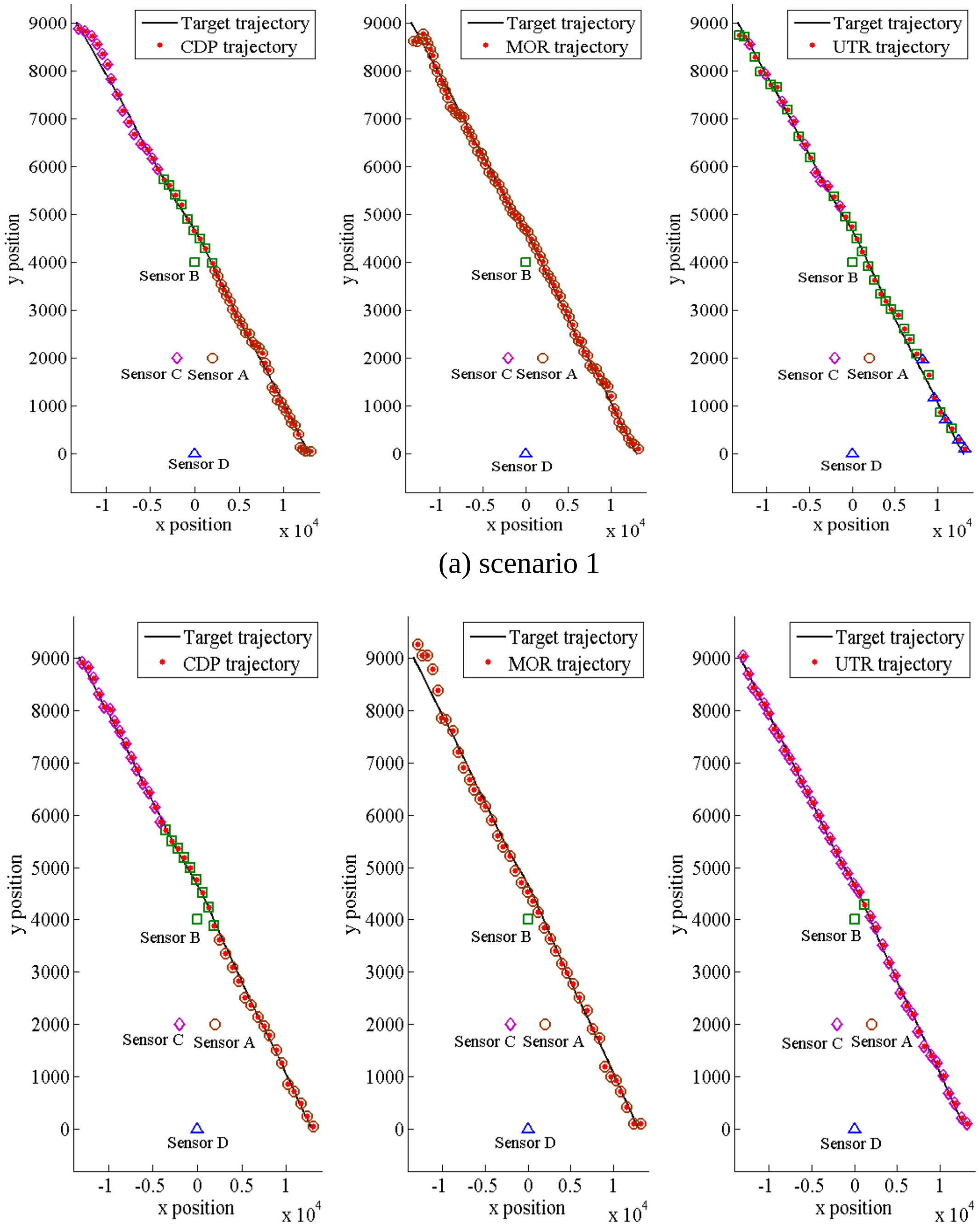
Fig.2 Diagram of the UTR algorithm



(a) scenario 1

(b) scenario 2

Fig.4 Comparison of accumulated tracking errors and interception costs



(a) scenario 1

(b) scenario 2

Fig.5 Estimated trajectories and sensor sequences