3

Predictive Angle Tracking Algorithm Based on Extended Kalman Filter

Sheng-Yun Hou[1], Shun-Hsyung Chang[2] and Hsien-Sen Hung[3]
[1]*Department of Electrical Engineering, Hwa Hsia Institute of Technology*
[2]*Department of Microelectronics Engineering, National Kaohsiung Marine University*
[3]*Department of Electrical Engineering, National Taiwan Ocean University*
Taiwan

1. Introduction

In the case of moving sources, various target angle tracking algorithms have been proposed and reported in the literature for multiple narrow-band targets. Yang and Kaveh proposed an iterative adaptive eigen-subspace method in conjunction with the multiple signal classification (MUSIC) algorithm to track the DOA angles of multiple targets (Yang & Kaveh, 1988). Due to the data association problem caused by multi-target tracking, the adaptive MUSIC method fails to track targets when they are moving close to each other. Although the method proposed by Sword, et al. (1990) can avoid the data association problem, errors are accumulated in each iteration, making it unable to track targets that are mutually close. Due to the nature of prediction-correction filtering process, Kalman filter (KF) can reduce estimation errors and avoid the data association problem when applied to angle tracking, as stated in several references (Javier & Sylvie, 1999; Yang, 1995; Park, et al. 1994). Rao, et al. (1994) proposed to estimate DOA angles of targets using the maximum likelihood method and feeding the results to a KF. However, it is assumed that the signal powers of the targets are all different, making the algorithm impractical. Javier and Sylvie (1999) suggested to estimate target angles using the projection approximation subspace tracking algorithm with deflation (PASTd) (Yang, 1995) and a Newton-type method (for MUSIC spectrum) for the use in the KF. It has lower computational load and better tracking performance than Rao's algorithm, but still exhibits poor tracking success rate at low signal-to-noise ratios (SNRs). Park, et al. (1994) proposed an approach, which utilizes predicted angles obtained from Sword's method. The approach also uses the constrained least-squares criterion to confine the dynamic range of angles. The choice of relevant parameters is empirical and is not suitable for various scenarios of different moving speeds and SNRs. Besides, the tracking performance degrades seriously with an increasing number of crossing targets. Later on, to improve Park's method, Ryu, et al. (1999, 2002) suggested to obtain the angle innovations of the targets from a signal subspace, instead of the sensor output covariance matrix, via projection approximation subspace tracking (PAST) algorithm (Yang, 1995). Chang, et al. (2005) modified Park's algorithm by incorporating a spatial smoothing (Shan et al., 1985) technique to overcome multipath interference, and also coherent signal-subspace (CSS) (Wang & Kaveh, 1985) processing for tracking wideband targets. All of the above algorithms are based on the sample covariance matrix or signal subspace made with multiple snapshots of data from a sensor array. However, they all fail to track multiple targets when only a single snapshot measurement is available between two consecutive time steps during the tracking process, because DOA estimation using subspace-based approach requires sample covariance matrix or signal subspace with a rank of more than one.

In the case of a single snapshot measurement within each time increment, tracking multiple targets becomes feasible if the sensor array output is directly used as the measurement data in the extended Kalman filter (EKF) (Kong & Chun, 2000). However, the EKF is an approximate nonlinear state estimation technique with first-order linearization accuracy, and is suitable for the tracking problem since the measurement model is nonlinear in terms of the angles (states) to be estimated. The algorithm proposed by Kong and Chun (2000) exhibits low tracking success rate when targets approach near the points of intersection. The reason for this weakness is the EKF can be difficult to tune and often gives unreliable estimates if the system nonlinearities are severe.

2. Tracking algorithm

For tracking non-stationary targets efficiently and effectively, the predictive angle tracking algorithm based on extended Kalman filter (PAT-EKF) is presented. In the proposed algorithm, the sensor array output is used as measurement data in EKF, since the measurement model is nonlinear in terms of angle estimates. Using the predicted angles, the PAT-EKF algorithm modifies Park's method to obtain angular innovation, from which the angle estimates are updated (smoothed) via Kalman gain.

2.1 Data model

In the data model, M targets moving (for tracking) in a plane are considered, which contain an array of L sensors (or hydrophones). The sensor positions are assumed to be known, and it takes them to be placed uniformly on a line with spacing of d between two adjacent sensors (abbreviated as ULA), measured in the unit of wavelength λ. The motion of the targets is assumed to be at constant angular speed in the presence of Gaussian disturbance, and is observed every T seconds. Let $\theta_m(t) \in [-\pi/2, \pi/2]$, measured clockwise with respect to y axis, denote the DOA angle of the mth target at time t. Assuming that these targets are located in the far field and their radiated signals are narrowband with a common angular frequency ω_0, the output of the lth isotropic sensor at time t is then

$$r_l(t) = \sum_{m=1}^{M} e^{-j\omega_o \tau_{lm}} s_m(t) + n_l(t) \tag{1}$$

where $s_m(t) \in R$ is the signal transmitted by the mth target at time t, $n_l(t)$ is a complex Gaussian white noise with zero mean and variance σ_n^2, which is uncorrelated with the target signals, and τ_{lm} is the difference in time delays of the mth target reaching the first (reference) sensor and the lth sensor.

By using vector-matrix representation, the output of the sensor array is given by

$$\mathbf{r}(t) = \mathbf{A}[\boldsymbol{\theta}(t)]\mathbf{s}(t) + \mathbf{n}(t) \tag{2}$$

where $\mathbf{r}(t)=[r_1(t),...,r_L(t)]^T$, $\mathbf{s}(t)=[s_1(t),...,s_M(t)]^T$, $\mathbf{n}(t)=[n_1(t),...,n_L(t)]^T$ are the output data, target signal, and noise vectors, respectively. $\boldsymbol{\theta}(t)=[\theta_1(t),\theta_2(t),...,\theta_M(t)]^T$ is the target DOA vector and $\mathbf{A}[\boldsymbol{\theta}(t)]$ is the array direction matrix with the direction vector of the mth target (the mth column vector)

$$\mathbf{a}_m=[1,e^{-j\frac{2\pi}{\lambda}d\sin\theta_m},...,e^{-j\frac{2\pi}{\lambda}(L-1)d\sin\theta_m}]^T,\quad m=1,\ldots,M. \tag{3}$$

Suppose that there are K measurements (snapshots) that are taken for each increment T, and the time increment is sufficiently small allowing us to approximate the target as stationary. Figure 1 shows the sensor array and source configurations in 2-D space.

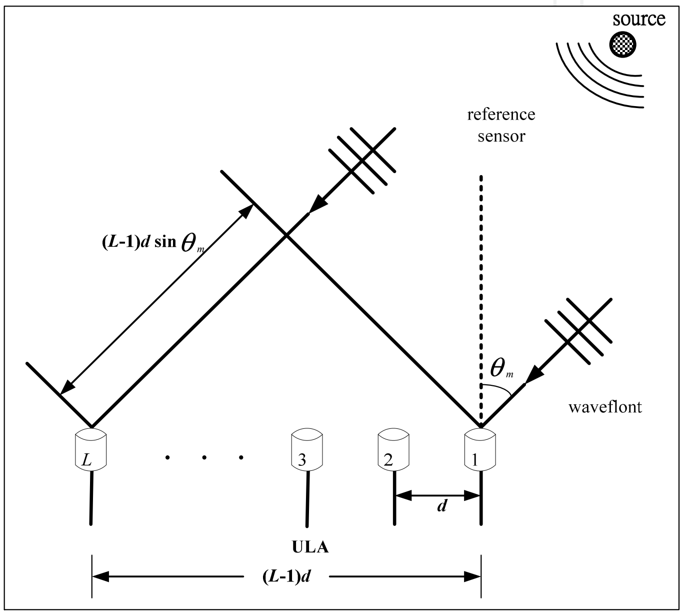


Fig. 1. Sensor array and source configurations in 2-D space

2.2 PAT-EKF algorithm

First, it depicts the discrete-time state (process) model for the target motion. For each time index k, it defines the state vector for the mth target as $\mathbf{x}_m(k)=\left[\theta_m(k)\quad \dot{\theta}_m(k)\right]^T$, consisting of its DOA and angular speed. The mth target motion can lead to the process equation (Park et al., 1994)

$$\mathbf{x}_m(k+1)=\mathbf{F}\mathbf{x}_m(k)+\mathbf{w}_m(k) \tag{4}$$

$$\mathbf{F}=\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix}$$

where the process noise vector $\mathbf{w}_m(k)$ is assumed to be Gaussian distributed with zero mean and covariance

$$\mathbf{Q}_m=\sigma_w^2\begin{bmatrix}\frac{T^3}{3} & \frac{T^2}{2}\\ \frac{T^2}{2} & T\end{bmatrix}$$

Assume that the motion of each target is mutually independent. By defining the composite state vector as $\mathbf{x}(k)=\left[\mathbf{x}_1^T(k),...,\mathbf{x}_M^T(k)\right]^T$, the system dynamics is governed by the process model

$$\mathbf{x}(k+1)=\bar{\mathbf{F}}\mathbf{x}(k)+\mathbf{w}(k) \tag{5}$$

$$\bar{\mathbf{F}}=diag(\overbrace{\mathbf{F},\cdots,\mathbf{F}}^{M})$$

The process noise vector $\mathbf{w}(k)$=$\left[\mathbf{w}_1^T(k),...,\mathbf{w}_M^T(k)\right]^T$ reflects the random modeling error, which is Gaussian distributed with zero mean vector and covariance

$$\bar{\mathbf{Q}}=diag(\mathbf{Q}_1,\cdots,\mathbf{Q}_M)$$

The matrices $\bar{\mathbf{F}}$ and $\bar{\mathbf{Q}}$ are all block diagonal. Although the process equation is a linear model, the measurement model of (2) is a vector nonlinear function of the target DOAs (and thus, of the target state vectors as well), which can be restated as

$$\mathbf{r}(k)\triangleq\mathbf{h}(\mathbf{x}(k),\mathbf{s}(k),\mathbf{n}(k))=\mathbf{A}(\mathbf{x}(k))\mathbf{s}(k)+\mathbf{n}(k) \tag{6}$$

where $\mathbf{n}(k)$ is complex Gaussian noise process with the known covariance $\sigma_n^2\mathbf{I}$, and is assumed to be uncorrelated with the process noise $\mathbf{w}(k)$. Assuming that a uniform linear array of L sensors with a half wavelength of inter-element spacing d is deployed, the partial derivative (Jacobian) matrix of the measurement model (6) is given by

$$\mathbf{H}(k)=\frac{\partial\mathbf{h}}{\partial\mathbf{x}}=[\mathbf{H}_1(k),...,\mathbf{H}_M(k)]$$

By augmenting the real and imaginary parts of each complex matrix $\mathbf{H}_m(k)$, it has the composite real matrix of dimension 2L×2M

$$\bar{\mathbf{H}}(k)=\begin{bmatrix} real(\mathbf{H}_1(k),...,\mathbf{H}_M(k)) \\ imag(\mathbf{H}_1(k),...,\mathbf{H}_M(k)) \end{bmatrix}$$

which can be expressed as

$$\bar{\mathbf{H}}(k)=\begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ g_{1,1} & 0 & \cdots & g_{M,1} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ g_{1,L-1} & 0 & \cdots & g_{M,L-1} & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ c_{1,1} & 0 & \cdots & c_{M,1} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{1,L-1} & 0 & \cdots & c_{M,L-1} & 0 \end{bmatrix}$$

where

$$g_{m,b} = -\sin(\pi b \sin(\theta_m(k)))\cos(\theta_m(k))s_m(k) \tag{7}$$

$$c_{m,b} = -\cos(\pi b \sin(\theta_m(k)))\cos(\theta_m(k))s_m(k) \tag{8}$$

m=1, …, M, b=1, …, L−1.

Initially (at k=0), the target DOA angles, $\{\hat{\theta}_m(-1)\}$ and $\{\hat{\theta}_m(0)\}$ at two successive time instants, k=−1 and k=0, are assumed to be available, which can be estimated by any subspace-based DOA angle estimation algorithm for instance the MUSIC algorithm (Schmidt, 1986). Thus, the initial state vector estimate can be set as $\hat{\mathbf{x}}(0\,|\,0) = [\hat{\theta}_1(0)\ \ (\hat{\theta}_1(0) - \hat{\theta}_1(-1))/T, \cdots,\ \ \hat{\theta}_M(0)\ \ (\hat{\theta}_M(0) - \hat{\theta}_M(-1))/T]^T$ with its covariance matrix $\mathbf{P}(0\,|\,0)$, given by

$$\mathbf{P}(0\,|\,0) = \sigma_v^2 \begin{bmatrix} 1 & \frac{1}{T} & & & & \\ \frac{1}{T} & \frac{2}{T^2} & & & 0 & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & 1 & \frac{1}{T} \\ & 0 & & & \frac{1}{T} & \frac{2}{T^2} \end{bmatrix}$$

For k=1,2,…, the proposed tracking algorithm can be summarized in the following four steps.

Step 1. Prediction of angles

The prediction of the state vector and its covariance matrix can be obtained from the existing estimates by the equations

$$\hat{\mathbf{x}}(k\,|\,k-1) = \overline{\mathbf{F}}\hat{\mathbf{x}}(k-1\,|\,k-1) \tag{9}$$

$$\mathbf{P}(k\,|\,k-1) = \overline{\mathbf{F}}\mathbf{P}(k-1\,|\,k-1)\overline{\mathbf{F}}^T + \overline{\mathbf{Q}} \tag{10}$$

The first element of each state vector $\hat{\mathbf{x}}_m(k\,|\,k-1)$ is the predicted estimate $\hat{\theta}_m(k\,|\,k-1)$ of $\theta_m(k)$. The predicted direction matrix $\mathbf{A}(k\,|\,k\text{-}1)$ can be obtained by (3) using $\hat{\theta}_m(k\,|\,k-1)$ for $\theta_m(k)$. From (2), the predicted output of the sensor array becomes

$$\mathbf{r}(k\,|\,k-1) = \mathbf{A}(k\,|\,k-1)\mathbf{s}(k) \tag{11}$$

and can be obtained once $\mathbf{s}(k)=[s_1(k),\ldots,\ s_M(k)]^T$ is estimated by invoking the maximum likelihood method as

$$\hat{\mathbf{s}}(k) = \left[\mathbf{A}^H(k\,|\,k-1)\mathbf{A}(k\,|\,k-1)\right]^{-1}\mathbf{A}^H(k\,|\,k-1)\mathbf{r}(k) \tag{12}$$

Step 2. Computations of the angle innovation

After time T, a new array output is observed and the direction matrix can be written as

$$\mathbf{A}(k) = \mathbf{A}(k \mid k-1) + \delta\mathbf{A}(k) \tag{13}$$

where $\delta\mathbf{A}(k)$ is the error matrix, which can be derived, according to (Sword *et al.*, 1990), as

$$\delta\mathbf{A}(k) = \begin{bmatrix} 0 & \cdots & 0 \\ \delta\gamma_1 & \cdots & \delta\gamma_M \\ 2\gamma_1\delta\gamma_1 & \cdots & 2\gamma_M\delta\gamma_M \\ \vdots & \ddots & \vdots \\ (L-1)\gamma_1^{L-2}\delta\gamma_1 & \cdots & (L-1)\gamma_M^{L-2}\delta\gamma_M \end{bmatrix} \tag{14}$$

wherein

$$\delta\gamma_m(k) = -j\pi\cos\theta_m(k)\gamma_m(k)\delta\theta_m(k) \tag{15}$$

Thus, the residual array output $\Delta\mathbf{r}(k)$ can be obtained and written as

$$\Delta\mathbf{r}(k) = \mathbf{r}(k) - \mathbf{r}(k \mid k-1) = \delta\mathbf{A}(k)\mathbf{s}(k) + \mathbf{n}(k) \tag{16}$$

Note that the first row vector of $\delta\mathbf{A}(k)$ in (14) is a null vector. To reduce the computation, the null vector allows us to define a $(L-1)\times 1$ vector $\Delta\tilde{\mathbf{r}}$, which is obtained by removing the first element of $\Delta\mathbf{r}$ in (16). By substituting (15) into (14), $\Delta\tilde{\mathbf{r}}$ can be represented by (dropping k temporarily)

$$\Delta\tilde{\mathbf{r}} = \mathbf{B}\delta\boldsymbol{\theta} + \tilde{\mathbf{n}} \tag{17}$$

In (17), the $(L-1)\times M$ matrix $\mathbf{B}$ is

$$\mathbf{B} = -j\pi \begin{bmatrix} \cos(\theta_1)\gamma_1 s_1 & \cdots & \cos(\theta_M)\gamma_M s_M \\ 2\cos(\theta_1)\gamma_1^2 s_1 & \cdots & 2\cos(\theta_M)\gamma_M^2 s_M \\ \vdots & \ddots & \vdots \\ (L-1)\cos(\theta_1)\gamma_1^{L-1}s_1 & \cdots & (L-1)\cos(\theta_M)\gamma_M^{L-1}s_M \end{bmatrix}$$

where θ_m is substituted with the predicted angle $\hat{\theta}_m(k \mid k-1)$, and $\delta\boldsymbol{\theta} = [\delta\theta_1(k), \delta\theta_2(k), \cdots, \delta\theta_M(k)]$ is the unknown angle innovation vector to be estimated. In general, a least-squares solution of (17) is given by $\delta\boldsymbol{\theta} = (\mathbf{B}^H\mathbf{B})^{-1}\mathbf{B}^H\Delta\tilde{\mathbf{r}}$. However, the modified solution

$$\delta\boldsymbol{\theta} = (\mathbf{B}^H\mathbf{B} + \mathbf{L})^{-1}\mathbf{B}^H\Delta\tilde{\mathbf{r}} \tag{18}$$

as suggested in Park's algorithm, will be used to constrain the absolute values of innovations for the cases of nearby targets, where $\mathbf{L}$ is a weighting matrix with diagonal form.

Step 3. Estimation of the angles

The estimated angle can be obtained as

$$\hat{\theta}_m(k) = \hat{\theta}_m(k \mid k-1) + \delta\theta_m(k) \tag{19}$$

Furthermore, $\hat{\theta}_m(k)$ and $\hat{s}_m(k)$ are substituted into (7) and (8) to update the matrix $\bar{\mathbf{H}}(k)$.

Step 4. Smoothing the estimated angles

Since the state vector is real-valued, it formulates the state estimation equation as

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{K}(k)\Delta\bar{\mathbf{r}}(k) \tag{20}$$

where $\Delta\bar{\mathbf{r}}(k)=[\Delta\mathbf{r}_R\ \Delta\mathbf{r}_I]^T$ is a real vector; $\Delta\mathbf{r}_R$ and $\Delta\mathbf{r}_I$ are the real and imaginary parts of $\Delta\mathbf{r}(k)$ from (16). **K(*k*)** is the Kalman Gain matrix, given by

$$\mathbf{K}(k) = \mathbf{P}(k \mid k-1)\bar{\mathbf{H}}^T(k)\left[\bar{\mathbf{H}}(k)\mathbf{P}(k \mid k-1)\bar{\mathbf{H}}^T(k) + \sigma_n^2\mathbf{I}\right]^{-1} \tag{21}$$

The covariance matrix of $\hat{\mathbf{x}}(k \mid k)$ is given by

$$\mathbf{P}(k \mid k) = \left[\mathbf{I} - \mathbf{K}(k)\bar{\mathbf{H}}(k)\right]\mathbf{P}(k \mid k-1) \tag{22}$$

The proposed PAT-EKF algorithm requires the number of $7LM^2+16L^2M+LMK$ real multiplications, whereas the Park's and Kong's algorithms require, respectively, the numbers of $3LM^2+K(3L^2+LM)$ and $5M^3+10LM^2+8L^2M+LMK$ real multiplications (K is the number of snapshots). Table 1 shows the comparison of computational complexity among these algorithms for M=3, L=8 and different number of snapshots. It is evident that the PAT-EKF algorithm has lower computational complexity than the Park's algorithm for $K \geq 30$, where $K \geq 30$ is often needed for acceptable tracking performance. Although the computational complexity is higher than the Kong's algorithm, the proposed method has much better performance as demonstrated by the simulations.

Algorithm		PAT-EKF	Park's	Kong's
Number of real multiplications	K=1	3600	432	2415
	K=10	3816	2376	2631
	K=30	4296	6696	3111
	K=50	4776	11016	3591

Table 1. Computational complexity comparison for M=3, L=8 and different K values.

2.3 PAT-EKF algorithm for tracking targets in 3-D space

The PAT-EKF algorithm is now extended to track narrow-band targets in 3-D space, where the system of sensor array and source configurations is shown in Figure 2, where $s_m(k)$ is the signal transmitted by the mth target, of which φ_m and θ_m are the azimuth and elevation respectively. ρ_m is the range from the mth target to the first (reference) sensor in the uniform linear array. As explained later, the number of sensors L must satisfy the condition $L \geq 3M+1$, where M is the number of targets. All the targets can be located in the near field or far field. In the following formulations, far-field targets are treated. The output of the lth sensor for the kth sampling interval can be expressed as

$$r_l(k) = \sum_{m=1}^{M} s_m(k - \tau_{lm}(k)) + n_l(k) \qquad l = 1, 2, \cdots, L \tag{23}$$

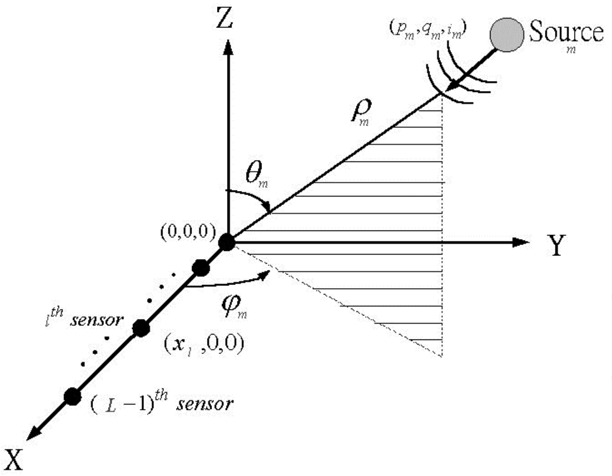


Fig. 2. Sensor array and source configurations in 3-D space.

From the array and source configurations shown in Figure 2, $\tau_{lm}(k)$ can be expressed as

$$\tau_{lm}(k) = \frac{1}{c}\left[\sqrt{(p_m - x_l)^2 + (q_m - y_l)^2 + (i_m - z_l)^2} - \rho_m\right]$$

where (x_l, y_l, z_l) is the lth sensor position relative to the reference sensor. Here $x_l=(l-1)d$, $y_l=0$, and $z_l=0$. The location coordinate of the mth signal source is given by

$$p_m = \rho_m \sin\theta_m \cos\varphi_m$$

$$q_m = \rho_m \sin\theta_m \sin\varphi_m$$

$$i_m = \rho_m \cos\theta_m$$

Assume that all the signal sources are narrowband with a common angular frequency ω. Then

$$s(k - \tau_{lm}(k)) \simeq s(k)e^{-j\omega\tau_{lm}(k)}$$

Therefore, (23) becomes

$$r_l(k) = \sum_{m=1}^{M} e^{-j\omega\tau_{lm}(k)} s_m(k) + n_l(k) \qquad l = 1,2,\cdots,L$$

In vector-matrix notation, the received output vector of the sensor array is $\mathbf{r}(k)= \mathbf{A}(k)\mathbf{s}(k) + \mathbf{n}(k)$, where

$$\mathbf{A}(k)=\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \gamma_{21}(k) & \gamma_{22}(k) & \cdots & \gamma_{2M}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{\mathrm{L}1}(k) & \gamma_{\mathrm{L}2}(k) & \cdots & \gamma_{LM}(k) \end{bmatrix}$$

and

$$\gamma_{lm}(k) = e^{-j\omega\tau_{lm}(k)}$$

For the use of the EKF, it redefines the state vector for the *m*th target as $\mathbf{x}_m(k) = \left[\rho_m(k) \quad \dot{\rho}_m(k) \quad \varphi_m(k) \quad \dot{\varphi}_m(k) \quad \theta_m(k) \quad \dot{\theta}_m(k)\right]^T$ in 3-D space. By augmenting the real and imaginary parts of each complex matrix $\mathbf{H}_m(k)$, it has the composite real matrix of dimension $2L\times6M$

$$\bar{\mathbf{H}}(k) = \begin{bmatrix} real(\mathbf{H}_1(k),...,\mathbf{H}_M(k)) \\ imag(\mathbf{H}_1(k),...,\mathbf{H}_M(k)) \end{bmatrix}$$

which can be expressed as

$$\bar{\mathbf{H}}(k) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ g_{21}\frac{\partial\tau_{21}}{\partial\rho_1} & 0 & g_{21}\frac{\partial\tau_{21}}{\partial\varphi_1} & 0 & g_{21}\frac{\partial\tau_{21}}{\partial\theta_1} & 0 & \cdots & g_{2M}\frac{\partial\tau_{2M}}{\partial\theta_M} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ g_{L1}\frac{\partial\tau_{L1}}{\partial\rho_1} & 0 & g_{L1}\frac{\partial\tau_{L1}}{\partial\varphi_1} & 0 & g_{L1}\frac{\partial\tau_{L1}}{\partial\theta_1} & 0 & \cdots & g_{LM}\frac{\partial\tau_{LM}}{\partial\theta_M} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ c_{21}\frac{\partial\tau_{21}}{\partial\rho_1} & 0 & c_{21}\frac{\partial\tau_{21}}{\partial\varphi_1} & 0 & c_{21}\frac{\partial\tau_{21}}{\partial\theta_1} & 0 & \cdots & c_{2M}\frac{\partial\tau_{2M}}{\partial\theta_M} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{L1}\frac{\partial\tau_{L1}}{\partial\rho_1} & 0 & c_{L1}\frac{\partial\tau_{L1}}{\partial\varphi_1} & 0 & c_{L1}\frac{\partial\tau_{L1}}{\partial\theta_1} & 0 & \cdots & c_{LM}\frac{\partial\tau_{LM}}{\partial\theta_M} & 0 \end{bmatrix}$$

with the following equations

$$\mathrm{g_{lm}} = -\omega\sin(\omega\tau_{\mathrm{lm}})\, \mathrm{s_m(k)}$$

$$\mathrm{c_{lm}} = \omega\cos(\omega\tau_{\mathrm{lm}})\, \mathrm{s_m(k)}$$

$$\begin{aligned} \frac{\partial\tau_{lm}}{\partial\rho_m} &= \frac{1}{c}\left\{\left[\rho_m\sin\theta_m\cos\varphi_m-(l-1)d\right]^2+\left(\rho_m\sin\theta_m\sin\varphi_m\right)^2+\left(\rho_m\cos\theta_m\right)^2\right\}^{-\frac{1}{2}} \\ &\quad \times\left[\rho_m-(l-1)d\sin\theta_m\cos\varphi_m\right]-\frac{1}{c} \\ &= \tau_{lm}\times\left[\rho_m-(l-1)d\sin\theta_m\cos\varphi_m\right]-\frac{1}{c} \end{aligned}$$

$$\begin{aligned} \frac{\partial\tau_{lm}}{\partial\varphi_m} &= \frac{1}{c}\left\{\left[\rho_m\sin\theta_m\cos\varphi_m-(l-1)d\right]^2+\left(\rho_m\sin\theta_m\sin\varphi_m\right)^2+\left(\rho_n\cos\theta_n\right)^2\right\}^{-\frac{1}{2}} \\ &\quad \times(l-1)d\rho_m\sin\theta_m\sin\varphi_m \\ &= \tau_{lm}\times(l-1)d\rho_m\sin\theta_m\sin\varphi_m \end{aligned}$$

$$\begin{aligned}\tfrac{\partial \tau_{lm}}{\partial \theta_m} &= \frac{1}{c}\left\{\left[\rho_m \sin\theta_m \cos\varphi_m - (l-1)d\right]^2 + \left(\rho_m \sin\theta_m \sin\varphi_m\right)^2 + \left(\rho_m \cos\theta_m\right)^2\right\}^{-\frac{1}{2}} \\ &\quad \times -(l-1)d\rho_m \cos\theta_m \cos\varphi_m \\ &= \tau_{lm} \times -(l-1)d\rho_m \cos\theta_m \cos\varphi_m\end{aligned}$$

For the 3-D PAT-EKF algorithm, the recursive equations of (9)-(13), (16) and (20)-(22) remain unchanged and the recursive equations of (14)-(15) and (17)-(19) need to be changed as stated in the following context.

Let $\delta\rho(k)$, $\delta\varphi(k)$, and $\delta\theta(k)$ be the unknown innovations of $\rho(k)$, $\varphi(k)$, $\theta(k)$ respectively, from time k-1 to time k. The (l,m) element of $\delta\mathbf{A}(k)$ can be derived as

$$[\delta\mathbf{A}(k)]_{(l,m)} = -j\omega\gamma_{lm}(k)\left[\tfrac{\partial \tau_{lm}}{\partial \rho_m}\delta\rho_m + \tfrac{\partial \tau_{lm}}{\partial \varphi_m}\delta\varphi_m + \tfrac{\partial \tau_{lm}}{\partial \theta_m}\delta\theta_m\right]$$

and the residual array output with the first row removed can be expressed as

$$\Delta\tilde{\mathbf{r}}(k) = \mathbf{B}\cdot\begin{bmatrix}\delta\boldsymbol{\rho}(k)\\ \delta\varphi(k)\\ \delta\boldsymbol{\theta}(k)\end{bmatrix} + \tilde{\mathbf{n}} \tag{24}$$

where the matrix $\mathbf{B}$ is a $(L-1) \times 3M$ matrix, given by

$$-j\omega\begin{bmatrix}\gamma_{21}\tfrac{\partial\tau_{21}}{\partial\rho_1}s_1 & \cdots & \gamma_{2N}\tfrac{\partial\tau_{2M}}{\partial\rho_M}s_M & \gamma_{21}\tfrac{\partial\tau_{21}}{\partial\varphi_1}s_1 & \cdots & \gamma_{2M}\tfrac{\partial\tau_{2M}}{\partial\varphi_M}s_M & \gamma_{21}\tfrac{\partial\tau_{21}}{\partial\theta_1}s_1 & \cdots & \gamma_{2M}\tfrac{\partial\tau_{2M}}{\partial\theta_M}s_M \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \gamma_{L1}\tfrac{\partial\tau_{L1}}{\partial\rho_1}s_1 & \cdots & \gamma_{LM}\tfrac{\partial\tau_{LM}}{\partial\rho_M}s_M & \gamma_{L1}\tfrac{\partial\tau_{L1}}{\partial\varphi_1}s_1 & \cdots & \gamma_{LM}\tfrac{\partial\tau_{LM}}{\partial\varphi_M}s_M & \gamma_{L1}\tfrac{\partial\tau_{L1}}{\partial\theta_1}s_1 & \cdots & \gamma_{LM}\tfrac{\partial\tau_{LM}}{\partial\theta_M}s_M\end{bmatrix}$$

Thus, a modified least-squares solution of (24) yields the innovations $\delta\boldsymbol{\rho}(k)$=[$\delta\rho_1(k),\cdots,\delta\rho_M(k)]^T$, $\delta\varphi(k)$=[$\delta\varphi_1(k),\cdots,\delta\varphi_M(k)]^T$, and $\delta\boldsymbol{\theta}(k)$=[$\delta\theta_1(k),\cdots,\delta\theta_M(k)]^T$, given by

$$\begin{bmatrix}\delta\boldsymbol{\rho}(k)\\ \delta\varphi(k)\\ \delta\boldsymbol{\theta}(k)\end{bmatrix} = (\mathbf{B}^H\mathbf{B} + \mathbf{L})^{-1}\mathbf{B}^H\Delta\tilde{\mathbf{r}}$$

These innovations are then used to update the state estimation according to

$$\hat{\theta}_m(k) = \hat{\theta}_m(k\,|\,k-1) + \delta\theta_m(k)$$

$$\hat{\varphi}_m(k) = \hat{\varphi}_m(k\,|\,k-1) + \delta\varphi_m(k)$$

$$\hat{\rho}_m(k) = \hat{\rho}_m(k\,|\,k-1) + \delta\rho_m(k)$$

There is one limiting condition, i.e., $L-1 \geq 3M$, under which the $L-1$ linear equations are used for solving $3M$ unknown variables.

3. Simulation results and discussion

In this section, the tracking performance of the three tracking algorithms are compared for narrow-band sources in 2-D space. A uniform linear array of eight sensors L=8 with half wavelength as the inter-element spacing is used. Three moving targets on the plane are tracked over an interval of 180s with T=1 s. During each T interval, K(=1, 10, 30, 50) snapshots of sensors data are generated. For comparison, the algorithms developed by Park *et al*. (1994), Kong and Chun (2000) were simulated. The Monte Carlo simulations of 100 runs were carried out for each algorithm with various SNRs. The parameters used in the system model for all algorithms to be compared are σ_v^2 =3, σ_w^2 =1, and σ_n^2 =3. The weighting factors to constrain the absolute values of innovations in (18) are set to be l_m= $\frac{1}{20}\times$ (mth diagonal element of $\mathbf{B}^H\mathbf{B}$), which is the same as in Park's algorithm (Park *et al*., 1994). The SNR is defined as $10log\,(\mathrm{s}\,/\,\sigma_n^2)$ in dB, where $\boldsymbol{s}$ is the signal power.

Table 2 gives the tracking results for various SNRs at K=30 snapshots. The PAT-EKF algorithm shows the highest tracking success rate (true angle ±5°) for each SNRs. Table 3 presents the tracking results for various number of snapshots at SNR=10dB. Again, the proposed algorithm shows the highest tracking success rate for each number of snapshots.

SNR (dB)	Tracking success rate (%)		
	PAT-EKF	Park’s	Kong’s
0	28	14	11
5	62	44	34
10	86	62	60

Table 2. Tracking performance for various SNRs at K=30

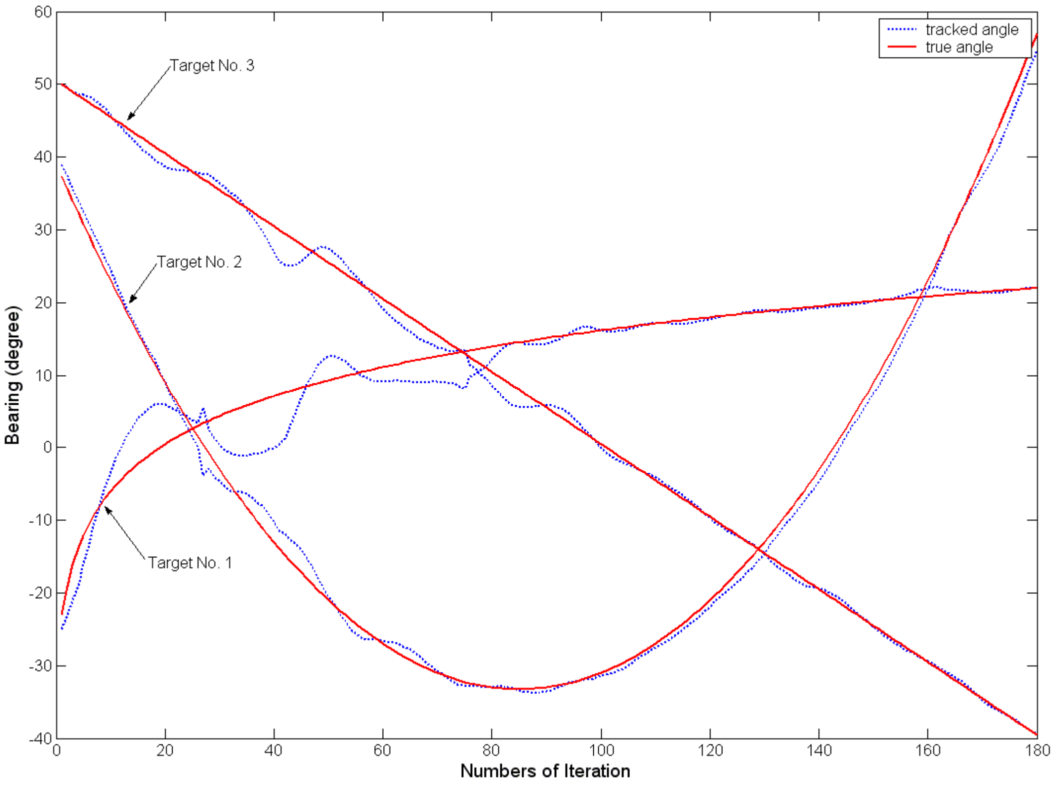
Number of Snapshots	Tracking success rate (%)		
	PAT-EKF	Park’s	Kong’s
1	70	45	43
10	83	69	55
50	88	81	66

Table 3. Tracking performance for various number of snapshots at SNR=10dB

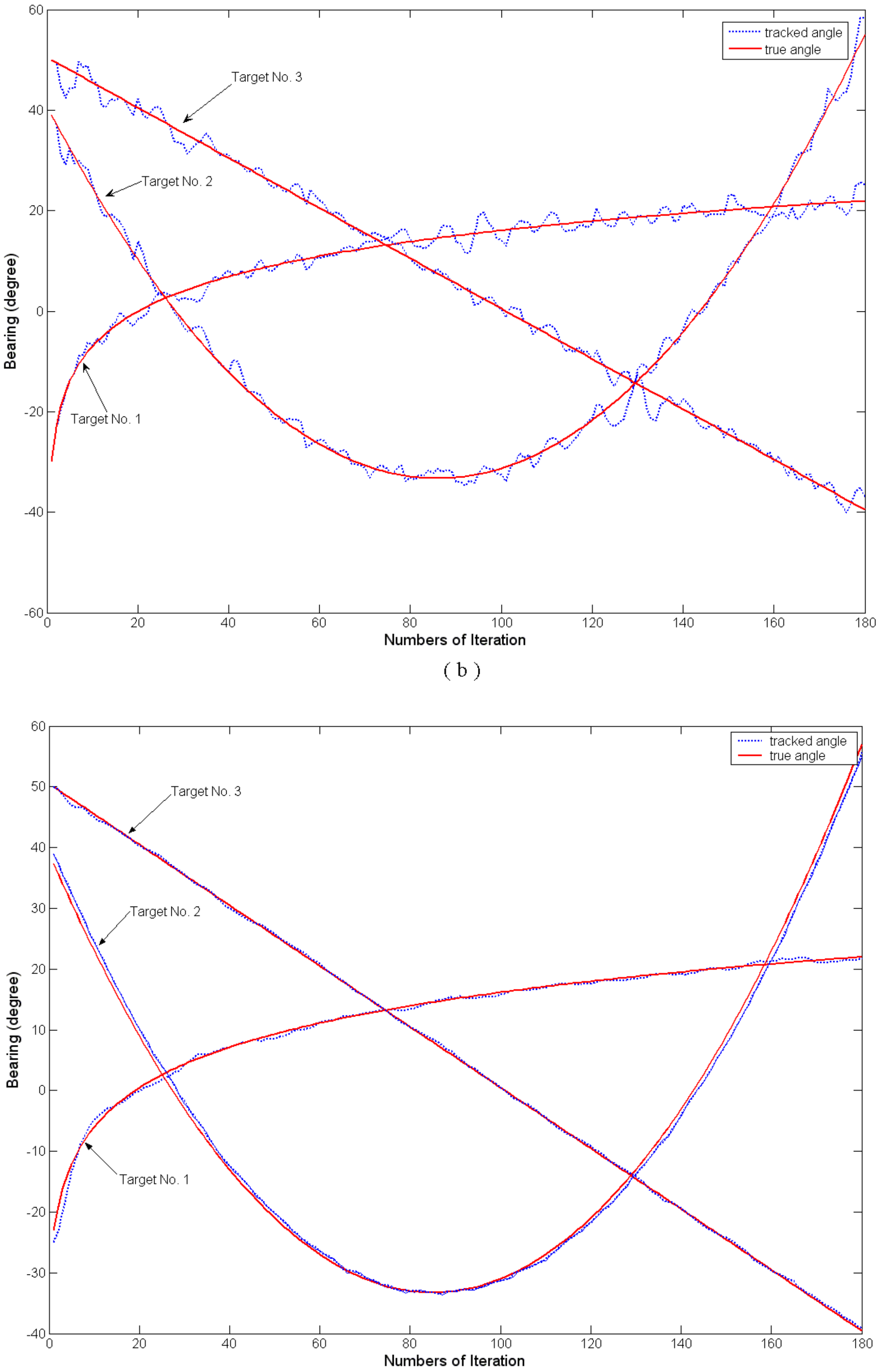
Figure 3 shows typical sample run for crossing tracks, all based on a single snapshot of data vector (*K*=1) at SNR=10dB of each target. The PAT-EKF and Park's algorithms exhibit much better tracking capability than Kong's algorithm (Kong & Chun, 2000) especially at the cross points in the trajectory.

Two moving targets are tracked over an interval of 20s with *T*=1 in 3-D space. During each *T* interval, *K*(=160) snapshots of sensors data are generated. Figure 4 shows the tracking performances of the 3-D PAT-EKF algorithm for the combinations of range, elevation, and azimuth at 3dB of SNR. In Figure 4, dot represents the true angle and line represents the tracked angle. The 3-D PAT-EKF algorithm is very effective in tracking the targets in 3-D space, even at low SNR.

Note that the PAT-EKF algorithm is excluded for performance comparison, simply because it fails to track the angle of the first signal source for the simulation example illustrated in Figure 5. In this example, the trajectory of the first signal source reveals its significantly changing behavior of angles. This also indicates that tracking capability of the PAT-EKF algorithm is rather limited when there exists a signal source with significant rates of angle variation.



(a)



(b)

(c)

Fig. 3. Typical sample run for crossing tracks with three targets at SNR=10dB. (a) Kong's algorithm, (b) Park's algorithm, (c) PAT-EKF algorithm

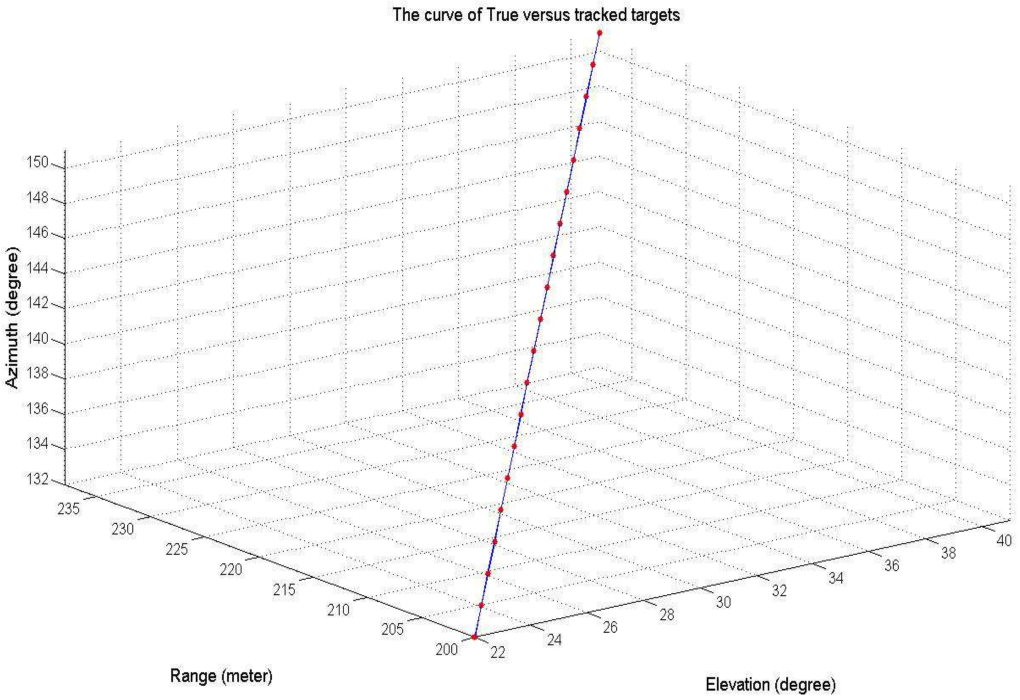


Fig. 4. Typical sample run for crossing tracks with three targets at SNR=10dB. (a) Kong's algorithm, (b) Park's algorithm, (c) PAT-EKF algorithm

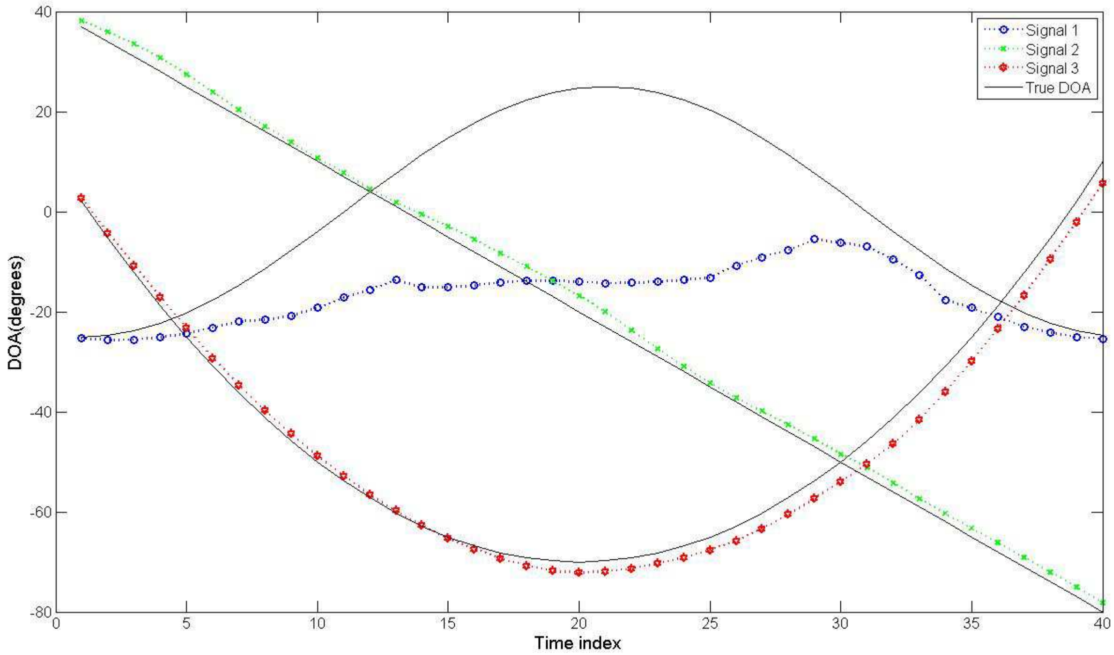


Fig. 5. The averaged tracking trajectories, using the PAT-EKF algorithm, for three equi-powered moving sources based on 25 snapshots at SNR=10dB.

4. Concluding remarks

This chapter has presented the PAT-EKF algorithm for tracking multiple targets. The proposed algorithm modified Park's algorithm by using the sensor array output vector rather than the sample covariance matrix and incorporating EKF instead of KF. These modifications allow the proposed algorithm to lower computational load, and also improve the tracking success rate particularly at lower snapshots. The PAT-EKF algorithm is then extended to track the azimuth, elevation, and range of multiple targets in 3-D space. Through computer simulations, the effectiveness of each proposed algorithm has been demonstrated. The drawback of the PAT-EKF algorithm is that it fails to track any target with a significant rate of angle variations.

5. Acknowledgment

This work was supported by National Science Council of Taiwan under contracts NSC 99-2923-E-022-001-MY3.

6. References

Chang, S.-H.; Hou, S.-Y.; Chang, S.-C. & Hung, H.-S. (2005). Underwater Wideband Signal Tracking based on Predictive Angle Tracking Algorithm. *Journal of Marine Science and Technology*, Vol. 13, No. 1, pp. 46-53

Javier, S. A. & Sylvie, M. (1999). An Efficient PASTd-Algorithm Implementation for Multiple Direction of Arrival Tracking. *IEEE Trans. Signal Processing*, Vol. 47, No. 8, pp. 2321-2324

Kong, D. & Chun, J. (2000). A fast DOA tracking algorithm based on the extended Kalman filter. Proceeding of the IEEE *National Aerospace and Electronics Conference NAECON*, pp. 235-238, ISBN 0-7803-6262-4, Dayton, Ohio, USA, Oct. 10-12, 2000

Park, S. B.; Ryu, C. S. & Lee, K. K. (1994). Multiple target tracking algorithm using predicted angles. *IEEE Trans. Aerosp. Electron Syst.*, Vol. 30, No. 2, pp. 643-648

Rao, C. R.; Sastry, C. R.; & Zhou, B. (1994). Tracking the Direction of Arrival of Multiple Moving Targets. *IEEE Trans. Signal Processing*, Vol. 42, No. 5, pp. 1133-1144

Ryu, C. S.; Lee, S. H. & Lee, K. K. (1999). Multiple target angle tracking algorithm using innovation extracted from signal subspace. *IEE Electronics Letters*, Vol. 35, No. 18, pp. 1520-1522

Ryu, C. S.; Lee, J. S. & Lee, K. K. (2002). Multiple target angle tracking algorithm with efficient equation for angular innovation. *IEE Electronics Letters*, Vol. 38, No. 10, pp. 483-484

Schmidt, R. O. (1986). Multiple emitter location and signal parameter estimation. *IEEE Trans. Antennas, Propagation*, Vol. 34, No. 3, pp. 276-280

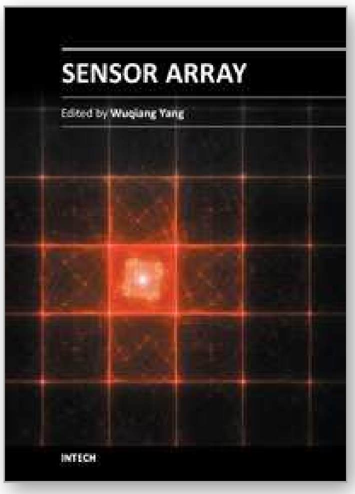
Shan, T. J.; Wax, M. & Kailath, T. (1985). On spatial smoothing for direction -of-arrival estimator of coherent signal. *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-33, pp. 806-811

Sword, C. K.; Simaan, M. & Kamen, W. W. (1990). Multiple target angle tracking using sensor array output. *IEEE Trans. Aerosp. Electron Syst.*, Vol. 26, No. 3, pp. 367-372

Wang, H. & Kaveh, M. (1985). Coherent Signal-Subspace Processing for the Detection and the Estimation of Angles of Arrival of Multiple Wide-band Sources. *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-33, pp. 823-831

Yang, B. (1995). Projection approximation subspace tracking. *IEEE Trans. Signal Processing*, Vol. 43, pp. 95-107

Yang J.-F. & Kaveh, M. (1988). Adaptive eigensubspace algorithms for direction or frequency estimation and tracking, *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 36, pp. 241-251



Sensor Array

Edited by Prof. Wuqiang Yang

ISBN 978-953-51-0613-5
Hard cover, 134 pages
Publisher InTech
Published online 23, May, 2012
Published in print edition May, 2012

Sensor arrays are used to overcome the limitation of simple and/or individual conventional sensors. Obviously, it is more complicated to deal with some issues related to sensor arrays, e.g. signal processing, than those conventional sensors. Some of the issues are addressed in this book, with emphasis on signal processing, calibration and some advanced applications, e.g. how to place sensors as an array for accurate measurement, how to calibrate a sensor array by experiment, how to use a sensor array to track non-stationary targets efficiently and effectively, how to use an ultrasonic sensor array for shape recognition and position measurement, how to use sensor arrays to detect chemical agents, and applications of gas sensor arrays, including e-nose. This book should be useful for those who would like to learn the recent developments in sensor arrays, in particular for engineers, academics and postgraduate students studying instrumentation and measurement.

Sheng-Yun Hou, Shun-Hsyung Chang and Hsien-Sen Hung (2012). Predictive Angle Tracking Algorithm Based on Extended Kalman Filter, Sensor Array, Prof. Wuqiang Yang (Ed.), ISBN: 978-953-51-0613-5, InTech, Available from: http://www.intechopen.com/books/sensor-array/ekf-based-predictive-angle-tracking-algorithm

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821