論文2001-38SP-5-14

EM 알고리즘을 이용한 적응다중표적추적필터

(An Adaptive Multiple Target Tracking Filter Using the EM Algorithm)

丁 弘 *, 朴 晸 浩 **

(Hong Jeong and Jeong-Ho Park)

Abstract

Tracking the targets of interest has been one of the major research areas in radar surveillance system. We formulate the tracking problem as an incomplete data problem and apply the EM algorithm to obtain the MAP estimate. The resulting filter has a recursive structure analogous to the Kalman filter. The difference is that the measurement-update deals with multiple measurements and the parameter-update can estimate the system parameters. Through extensive experiments, it turns out that the proposed system is better than PDAF and NNF in tracking the targets. Also, the performance degrades gracefully as the disturbances become stronger.

I. Introduction

Tracking all the targets of interest within the coverage region of radar has been one of the major research fields in radar surveillance system[1]. The difficulty of this arises from the noise characteristics of the return signal that is often the output of a detection system such as moving target indication (MTI) radar[2,3].

One of the pioneering works in this field is the joint probabilistic data association filter (JPDAF)[4]. This is an expansion of the probabilistic data association filter (PDAF)[5,6], and both methods are well described in[7~9]. These algorithms introduce an association mechanism between the targets and measurements using a matrix. Within a tracking gate, the optimal estimate of a target state is obtained by the conditional mean of the states given the measurements.

Another approach for obtaining optimal association is the connectionist scheme[10~13]. In this approach, the constraints on association are represented with the connection strengths between neurons. The energy function is often realized by a Hopfield network[14,15] and the network finds minimum energy state. Sengupta and Iltis[10] used this approach, and Jeong and Lee[16,17] showed that there exists a better energy function that faithfully reflects the natural constraints for optimal association. One of the disadvantages of this approach is that the parameters of the neural network are determined by trial-and-error.

As yet another approach, there have been emerging interests in applying the expectation-maximization (EM) algorithm[18,19] to the tracking

* 正會員, 浦港工科大學校 電子工學科

(Dept. of E.E., POSTECH)

** 正會員, LG이노텍 責任硏究員

(LG Innotek Co., Ltd.)

接受日字:1999年11月16日, 수정완료일:2001年7月23日

problem[20~23]. Major works include the papers by Avitzour[20], Gauvrit[21], and Molnar[22]. The key point of Avitzour's method[20] is to calculate the target states using maximum likelihood (ML) estimation in an EM fashion. Data observed over a time interval is used and processed in block form.

All of these schemes focus only on data association and state estimation based on measurements and thus additionally need a good filter, such as the Kalman filter[24,25], for predicting the target state in the next time frame. These two operations, often denoted as measurement- and time-updates, are the two key components in a tracking filter. Integration of measurement- and time-updates in an EM approach has been successfully achieved by Molnar[22] by using a maximum a posteriori probability (MAP) estimate instead of Avitzour's ML approach.

In order to start from the most general case, we also include the system parameters, like the probability of detection, false alarm density, and measurement error variance to the parameters to be estimated, in addition to the states of targets. As a result, the formulation becomes an adaptive filter that consists of a recursive time-update in time and a recursive measurement-update in each fixed time frame. This approach leads to a more efficient filter that integrates EM and Kalman filtering so that one can recursively predict, independently for each target, the states while incoming measurements improve the states and system parameters. This scheme enables us to build a tracking filter that can be computed for dynamically changing environments in real time.

The organization of the rest of this article is as follows. Section 2 explains a tracking filter that consists of time-, measurement-, and parameter-updates. Section 3 defines the problem on measurement- and parameter-updates, introduces parametric models of probability distributions, and derives the solutions in closed forms. Separately from this, the corresponding issues of the covariance measurement update together with computational complexity are discussed in section 4. Finally, experimental results are shown in section 5.

II. Tracking Filter Structure

Fig.1 shows the three parts of the tracking filter. Time-update part predicts the target state, measurement-update refines the predicted state with the measurements, and parameter-update part tunes the system parameters.

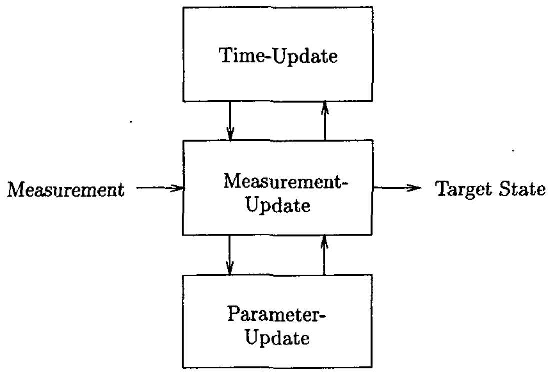


그림 1. 측정, 시간, 패러미터 갱신.
Fig. 1. Measurement-, time-, and parameter-updates.

Assuming that the number of targets N is known, we define the state as the positions and velocities of the targets. For a specific target t at time k, it becomes $\phi_t(k)=(x_t(k), \dot{x}_t(k), y_t(k), \dot{y}_t(k))^T$, where and $x_t(k)$ and $y_t(k)$ represent the target position in rectangular coordinates, and $\dot{x}_t(k)$ and $\dot{y}_t(k)$ represent the target velocity. Based on these notations, the dynamics of the target t can be expressed by the state equation

$$\phi_t(k)=\boldsymbol{F}\phi_t(k-1)+\boldsymbol{G}w(k-1) \quad (1)$$

$$\boldsymbol{y}_j(k)=\boldsymbol{H}\phi_t(k)+v(k) \quad (2)$$

where the state transition matrix $\boldsymbol{F}$ and the process noise coupling matrix $\boldsymbol{G}$ are given by

$$F=\begin{pmatrix}1 & T & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & T\\0 & 0 & 0 & 1\end{pmatrix},\quad G=\begin{pmatrix}T^2/2 & 0\\T & 0\\0 & T^2/2\\0 & T1\end{pmatrix},$$

where T is the time interval. As usual, the noise $w(k-1)$ and $v(k)$ are white Gaussian $N(0, Q)$ and $N(0, R(k))$, respectively. We assume that the measurement $y_j(k)$ is observed in rectangular coordinates for simplicity and originates from the target t. If $y_k(k)$ does not originate from the target t, $y_j(k)$ is distributed uniformly. Note that the measurement is usually available in polar coordinates form, but we limit the model to the linear observation case for simplicity.

It is well known that the time-update is given by[26]

$$\begin{cases}\phi_t(k|k-1)=F\phi_t(k-1|k-1),\\ P_t(k|k-1)=FP_t(k-1|k-1)F^T+GQG^T\end{cases}\tag{3}$$

A measurement generated from the target t is normally distributed as

$$p[yj(k)|\phi_t(k|k-1)]=N(H\phi_t(k|k-1), S_t(k))\tag{4}$$

where the measurement prediction covariance $S_t(k)$ is given by

$$S_t(k)=HP_t(k|k-1)H^T+R(k)\tag{5}$$

From these, the tracking gate is given by

$$V_t(k)\triangleq\{y(k): \nu^T(k)S_t^{-1}(k)\nu(k)\le\gamma\}\tag{6}$$

to restrict the measurement with the gate area to reduce a computational burden, where $\nu(k)=y(k)-H\phi_t(k|k-1)$ is the innovation[7] and γ is a constant that controls the gate size and the gate probability.

After finding the gate for each of the N targets, the number of measurements in each gate are counted and denoted with $M_t(k)$. We define the validation vector for target t as follows:

$$w_t(k)\triangleq\{w_j(k)|j\in[1, M_t(k)]\},\quad \text{for } t\in[1, N]\tag{7}$$

Since measurements located outside of the gate are simply discarded, the elements in this vector are all one. This definition differs from that of the ordinary validation matrix where some elements may be 0. This case occurs when all the measurements are globally counted. Similarly, we define an association vector as

$$z_t(k)\triangleq\{z_j(k)|j\in[1, M_t(k)]\},\quad \text{for } t\in[1, N]\tag{8}$$

If the measurement j originates from the target, $z_j(k)=1$. If not, $z_j(k)=0$. Somehow, this quantity must be estimated from the given validation vector. This problem is a kind of inverse mapping where many solutions may exist. To limit the possible solution space, one should rely on natural constraints of the measurement-target association.

Ⅲ. Measurement- and Parameter-Updates

Among the components of the tracking filter shown in Fig.1, the structure of the time-update is already clear as given in (3). Hence we have only to investigate the structure of the other parts. To begin with, we define the problem within the context of stochastic estimation and derive the necessary probability density functions. From these, it is possible to derive the functions to be maximized such as the Q- and M-functions. The final goal is to derive optimal solutions for both the states and the parameters.

1. Problem Formulation

In the EM formulation we define the combination of measurements $Y(k)$ and validation vectors $\Omega(k)$ as the *incomplete data* $W(k)$; $W(k)=\{Y(k), \Omega(k)\}$. The association vectors $Z(k)$ is considered to be the *missing data*. As a combination of the two quantities, the *complete* *data* $X(k)$ becomes $X(k)=\{Y(k),\Omega(k),Z(k)\}$. We further define the quantity, $X^k=\{X(k),X(k-1),\cdots,X(0)\}$ that is an accumulation of the present and all the past data.

A parameter that controls the complete data is the state $\Phi(k)$ as used by Molnar[22]. In addition, we define a parameter $\theta(k)$ that includes the various sources of external influence. The most important sources of $\theta(k)$ is the covariance matrix $\boldsymbol{R}(k)$ of the measurement noise. The diagonal components of $\boldsymbol{R}(k)$ are $\sigma_x^2(k)$ and $\sigma_y^2(k)$, and the off-diagonal components are all zeros. As we will see later, the other important sources include the false alarm density $L_d(k)$ and the gate detection probability $\alpha(k)$. Putting these altogether, we define $\theta(k)\triangleq\{\sigma_x^2(k),\sigma_y^2(k),L_d(k),\alpha(k)\}$. As a result of our extended model, the two quantities, the state $\Phi(k)$ and the parameter $\theta(k)$, jointly form the parameter space $(\Phi(k),\theta(k))$. Incidentally, the upper and lower cases of the variables denote respectively the random variables and realizations.

The goal is to determine the unknowns $(\Phi(k),\theta(k))$, given the observations $(Y(k),\Omega(k))$. Let us observe the posterior pdf

$$p(\phi,\boldsymbol{\theta}|\boldsymbol{x}^k)=p(\boldsymbol{x}|\phi,\boldsymbol{\theta},\boldsymbol{x}^{k-1})p(\phi|\boldsymbol{\theta},\boldsymbol{x}^{k-1})(p\,\boldsymbol{\theta}|\boldsymbol{x}^{k-1})$$
$$=\frac{1}{C}p(\boldsymbol{x}|\phi,\boldsymbol{\theta})p\{\phi|\boldsymbol{\theta},\phi(k|k-1)\}p\{\boldsymbol{\theta}|\boldsymbol{\theta}(k-1)\} \qquad (9)$$

where $p(\boldsymbol{x}|\phi,\boldsymbol{\theta},\boldsymbol{x}^{k-1})=p(\boldsymbol{x}|\phi,\boldsymbol{\theta})$ since the current parameter implies all the past complete data, $p(\phi|\boldsymbol{\theta},\boldsymbol{x}^{k-1})=p[\phi|\boldsymbol{\theta},\phi(k|k-1)]$ since the information $\boldsymbol{x}^{k-1}$ is contained in $\phi(k|k-1)$, $p(\boldsymbol{\theta}|\boldsymbol{x}^{k-1})=p[\boldsymbol{\theta}|\boldsymbol{\theta}(k-1)]$, and C is the normalization constant, because $p(\boldsymbol{x}|\boldsymbol{x}^{k-1})$ is constant in terms of the parameter. In this section, we omit (k) in the equations for simplicity if there is no possibility of confusion.

This posterior pdf enables us to employ penalized EM[27]. To get the estimates of ϕ and $\boldsymbol{\theta}$ simultaneously, ECM method is normally used[28]. However, here we use a different method to solve two parameters individually. For a fixed system parameter, the state variable is determined with the penalized EM method. Only after a stable solution is reached, the system parameter is determined. That is, if $\phi(k|k)\triangleq\lim_{p\to\infty}\phi^{(p)}(k|k)$, this scheme becomes:

$$\begin{cases}\text{E-step: } Q\{\phi|\phi^{(p)}(k|k),\boldsymbol{\theta}^0\}=E\{\log p[\boldsymbol{x}|\phi,\boldsymbol{\theta}^0]|\boldsymbol{y},\boldsymbol{w},\phi^{(p)}(k|k),\boldsymbol{\theta}^0\}\\ \text{M-step: } \phi^{(p+1)}(k|k)=\arg\max_\phi M[\phi,\boldsymbol{\theta}^0|\phi^{(p)}(k|k),\boldsymbol{\theta}^0]\\ \text{MAP: } \boldsymbol{\theta}=\arg\max_\theta M[\phi(k|k),\boldsymbol{\theta}|\phi(k|k),\boldsymbol{\theta}^0]\end{cases} \qquad (10)$$

where the M-function is defined as

$$M[\phi,\boldsymbol{\theta}|\phi^{(p)}(k|k),\boldsymbol{\theta}^0]\triangleq E\{\log p[\phi,\boldsymbol{\theta}|\boldsymbol{x}^k]|\boldsymbol{y},\boldsymbol{w},\phi^{(p)}(k|k),\boldsymbol{\theta}^0\}$$
$$=Q[\phi,\boldsymbol{\theta}|\phi^{(p)}(k|k),\boldsymbol{\theta}^0]-K(\phi,\boldsymbol{\theta}) \qquad (11)$$

and the penalty function is given by

$$K(\phi,\boldsymbol{\theta})\triangleq-\log p[\phi|\boldsymbol{\theta},\phi(k|k-1)]-\log p[\boldsymbol{\theta}|\boldsymbol{\theta}(k-1)] \qquad (12)$$

Here, $\boldsymbol{\theta}^0=\boldsymbol{\theta}(k-1)$. It is important that the EM loop and the parameter are separately computed. Note also that the parameter determined in the current process will be used again in the EM loop of the next frame.

2. Parametric Models

We assume the N targets are statistically independent and thus a derivation for only one target will suffice. The other assumption is that the x and y components of the state are mutually independent and thus a derivation in one direction is enough. In this subsection we derive probability density functions to be used to get the Q-function.

The complete data log-likelihood function is decomposed into

$$\log p(\boldsymbol{x}|\phi,\boldsymbol{\theta})=\log p(\boldsymbol{y}|\boldsymbol{w},\boldsymbol{z},\phi,\boldsymbol{\theta})+\log p(\boldsymbol{w}|\boldsymbol{z},\phi,\boldsymbol{\theta})$$
$$+\log p(\boldsymbol{z}|\phi,\boldsymbol{\theta})) \qquad (13)$$

In the first term, measurement $\boldsymbol{y}$ only depend upon $(\boldsymbol{z},\phi,\boldsymbol{\theta})$. In the second term, $\boldsymbol{w}$ depends directly upon $(\boldsymbol{z},\boldsymbol{\theta})$. The third term can be rewritten as $p(\boldsymbol{z}|\boldsymbol{\theta})$, since $\boldsymbol{z}$ does not depend upon

ϕ. Incorporating these facts, (13) becomes

$$\log p(\mathbf{x}|\phi, \boldsymbol{\theta}) = \log p(\mathbf{y}|\mathbf{z}, \phi, \boldsymbol{\theta}) + \log p(\mathbf{w}|\mathbf{z}, \boldsymbol{\theta}) + \log p(\mathbf{z}|\boldsymbol{\theta}) \quad (14)$$

The first term denotes the relationship between the measurements and a target given the data association. Naturally, this relationship must hold separately for each measurement. By definition, $z_j=1$ means that $\mathbf{y}_j$ is an observation of ϕ and it is natural to model this relationship with the Gaussian distribution $y_j \sim N(\mathbf{H}\phi, \mathbf{R}(k))$, as defined in (2). If $z_j=0$, the measurement has a uniform distribution. Therefore, we can write

$$\log p(\mathbf{y}|\mathbf{z}, \phi, \boldsymbol{\theta}) = -\sum_{j=1}^{M_t}\frac{z_j}{2}(\mathbf{y}_j - \mathbf{H}\phi)^T\mathbf{R}(k)^{-1}(y_j - \mathbf{H}\phi) - \frac{\log[(2\pi)^n|\mathbf{R}(k)|]}{2}\sum_{j=1}^{M_t}z_j - \log P_g\sum_{j=1}^{M_t}z_j - (M_t - \sum_{j=1}^{M_t}z_j)\log V, \quad (15)$$

where n is the dimension of the observation vector $\mathbf{y}_j$, i.e. $n=2$, P_g is the gate probability, and V is the area of the gate.

The second term in (14) denotes the relationship between the validation and association vectors given the system parameter. In detection theory, it is well known that the number of false alarms has a Poisson distribution, therefore, one can write

$$p(\mathbf{w}|\mathbf{z}, \boldsymbol{\theta}) = e^{-L}\frac{L^{M_t - \Sigma_{j=1}^{M_t}z_k}}{(M_t - \Sigma_{j=1}^{M_t}z_j)!}, \quad (16)$$

where $L=L_dV$ is the expected number of false alarms in the gate, $M_t - \Sigma_{j=1}^{M_t}z_j$ is the number of false alarms, and L_d is the false alarm density. By taking the logarithm of (16) and discarding all the terms that do not depend on either ϕ or $\boldsymbol{\theta}$, we get

$$\log p(\mathbf{w}|\mathbf{z}, \boldsymbol{\theta}) = -L + \left(M_t \sum_{j=1}^{M_t}z_j\right)\log L \quad (17)$$

The final term in (14) describes the natural *constraints on the association vector*. The constraint is that the number of detected targets in the gate can not exceed 1 according to the definition of the gate. $\Sigma_{t=1}^{M_t}z_j=0$ means that a miss error has occurred. With a parameter $\alpha \triangleq P_dP_g$ called the gate detection probability, the quantity $\mathbf{z}$ has the distribution:

$$\log p(\mathbf{z}|\boldsymbol{\theta}) = \delta\left(\sum_{j=1}^{M_t}z_j\right)\log(1-\alpha) + \delta\left(\sum_{j=1}^{M_t}z_j - 1\right)\log\frac{\alpha}{M_t} \quad (18)$$

where $\delta(x)$ is the Kronecker delta function and P_d is the detection probability[29].

As can be seen in (10), $p[\mathbf{x}|\mathbf{y}, \mathbf{w}, \phi^{(p)}(k|k), \boldsymbol{\theta}^0]$ is needed for the expectation of (13). However, this quantity is a function of $\mathbf{z}$, and thus knowing $p[\mathbf{z}|\mathbf{y}, \mathbf{w}, \phi^{(p)}(k|k), \boldsymbol{\theta}^0]$ is sufficient for the expectation. To be more specific, the conditional pdf is

$$p[\mathbf{z}|\mathbf{y}, \mathbf{w}, \phi^{(p)}(k|k), \boldsymbol{\theta}^0] = \frac{1}{Z}p[\mathbf{y}|\mathbf{z}, \phi^{(p)}(k|k), \mathbf{R}^0]p[\mathbf{w}|\mathbf{z}, L_d^0]p[\mathbf{z}|\alpha^0] \quad (19)$$

where Z is a normalization constant. To simplify this equation, we replace $\boldsymbol{\theta}^0$ with actual values, and delete independent variables.

The pdfs in (19) are very similar to those of the log-likelihood function (13), except that ϕ and $\boldsymbol{\theta}$ have been replaced with $\phi^{(p)}(k|k)$ and $\boldsymbol{\theta}^0$. Therefore, all the pdfs derived for the log-likelihood, can be reused here. From (15), we have

$$p[\mathbf{y}|\mathbf{z}, \phi^{(p)}(k|k), \mathbf{R}^0] = V^{-(M_t - \Sigma_{j=1}^{M_t}z_j)}P_g^{-\Sigma_{j=1}^{M_t}z_j}\{(2\pi)^n|\mathbf{S}(k|k)\}^{-\frac{\Sigma_{j=1}^{M_t}z_j}{2}} \times \exp\left\{-\sum_{j=1}^{M_t}\frac{z_j}{2}[\mathbf{y}_j - \mathbf{H}\phi^{(p)}(k|k)]^T\mathbf{S}(k|k)^{-1}[\mathbf{y}_j - \mathbf{H}\phi^{(p)}(k|k)]\right\} \quad (20)$$

where

$$\mathbf{S}(k|k) = \mathbf{HP}(k|k)\mathbf{H}^T + \mathbf{R}^0(k) \approx \mathbf{HP}(k|k-1)\mathbf{H}^T + \mathbf{R}^0(k) \quad (21)$$

Because we can not use $P(k|k)$ during this calculation, we use approximation. (16) and (18) are easily converted to

$$p[\boldsymbol{w}|\boldsymbol{z}, L_d^0] = e^{-L^0} \frac{L^{0(M_t - \Sigma_{j=1}^{M_t} z_j)}}{(M_t - \Sigma_{j=1}^{M_t} z_j)!}, \tag{22}$$

and

$$p[\boldsymbol{z}|\alpha^0] = (1-\alpha^0)\delta\left(\sum_{j=1}^{M_t} z_j\right) + \frac{\alpha^0}{M_t}\delta\left(\sum_{j=1}^{M_t} z_j - 1\right) \tag{23}$$

where $L^0 = L_d^0 V$.

3. M Function

So far, we have derived the complete data log-likelihood function and the conditional pdf of the complete data. From these and (10), we have

$$\begin{aligned} & Q[\phi, \boldsymbol{\theta}|\phi^{(p)}(k|k), \boldsymbol{\theta}^0] \\ &= -\frac{1}{2}\sum_{j=1}^{M_t} z_j^{(p)} \log[(2\pi)^n |R^0(k)|] - \frac{1}{2}\sum_{j=1}^{M_t} z_j^{(p)} \\ &\quad [\boldsymbol{y}_j - \boldsymbol{H}\phi]^T \boldsymbol{R}^0(k)^{-1}[\boldsymbol{y}_j - \boldsymbol{H}\phi] - L^0 + \left(M_t - \sum_{j=1}^{M_t} z_j^{(p)}\right) \\ &\quad \log L^0 + E(0)\log(1-\alpha^0) + E(1)\log\frac{\alpha^0}{M_t} \end{aligned} \tag{24}$$

where $z_j^{(p)}$ and $E(x)$ are the expectations of z_j and $\delta(\Sigma_{j=1}^{M_t} z_j - x)$, respectively. Note that z_j is binary but its expectation $z_j^{(p)}$ is not. Also, $E(x)$ is the probability that the number of ones in the association vector $\boldsymbol{z}$ is x. These two quantities, $z_j^{(p)}(k)$ and $E(x)$, must be determined next.

Since each element of the association vector is binary, and at most one element of the vector can be one, $Z_j^{(p)}$ is simplified to

$$z_j^{(p)} = p[\boldsymbol{z} = \boldsymbol{e}_j | \boldsymbol{y}, \boldsymbol{w}, \phi^{(p)}(k|k), \boldsymbol{\theta}^0] \tag{25}$$

where $\boldsymbol{e}_j$ is the unit vector whose jth element is one. For notational convenience, we introduce a set of auxiliary variable $\{a_j : j = 0, \cdots, M_t\}$:

$$a_j \triangleq \begin{cases} P_g(1-\alpha^0)L_d^0, & \text{for } j=0 \\ \alpha^0[(2\pi)^n |S(k|k)|]^{\frac{1}{2}} \\ \quad \times \exp\left\{-\frac{1}{2}[\boldsymbol{y}_j - \boldsymbol{H}\phi^{(p)}(k|k)^T S(k|k)]^{-1}[\boldsymbol{y}_j - \boldsymbol{H}\phi^{(p)}(k|k)]\right\} & \text{otherwise} \end{cases} \tag{26}$$

From (19), (25), and (26), we have the expectation of z_j :

$$z_j^{(p)} = \frac{a_j}{\Sigma_{j=0}^{M_t} a_j}, \tag{27}$$

and

$$\begin{cases} E(0) = \dfrac{a_0}{\Sigma_{j=0}^{M_t} a_j} = z_0^{(p)}, \\ E(1) = \Sigma_{j=1}^{M_t} z_j^{(p)} \end{cases} \tag{28}$$

Note that the possible values of $z_j(k)$ are only 0 and 1, but the expectation $z_j^{(p)}(k)$ can naturally have a value in the range [0,1].

The predicted state pdf $p[\phi(k)|\boldsymbol{\theta}(k), \phi(k|k-1)]$ is Gaussian, with a mean $\phi(k|k-1)$ and covariance $\boldsymbol{P}(k|k-1)$, and from (12), the penalty function becomes

$$\begin{aligned} K(\phi) &= -\log p[\phi|\phi(k|k-1)] \\ &= \frac{1}{2}\{\log[(2\pi)^d |\boldsymbol{P}(k|k-1)|] + [\phi(k|k-1) - \phi]^T \\ &\quad \boldsymbol{P}^{-1}(k|k-1)[\phi(k|k-1) - \phi]\} \end{aligned} \tag{29}$$

where d is the dimension of the state vector ϕ. As defined in (11), we obtain the M-function by subtracting (29) from (24).

4. Solution of Measurement- and Parameter-Updates

As we see in (10), the optimal parameter can be obtained by maximizing the M-function. In the M-step equation, Molnar[22] ignored the partition function Z for simplicity. However the partition function is not constant so this approximation can reduce the performance. Therefore we derive the M-function without removing the partition function.

From (10), setting the gradient $\nabla_\phi M$ to zero[30] and reorganizing and simplifying it by the matrix identity due to Woodbury[31,32], one obtains the closed form:

$$\phi^{p+1}(k|k) = \phi(k|k-1) + \boldsymbol{P}(k|k-1)$$

$$H^T S^{-1} \frac{\Sigma_{j=1}^{M_t} z_j^{(p)} [y_j - H\phi(k|k-1)]}{\Sigma j = 1^{M_t} z_j^{(p)}} \quad (30)$$

where

$$S = R^0(k) / \sum_{j=1}^{M_t} z_j^{(p)} + HP(k|k-1) H^T \quad (31)$$

It is easy to see that if one sets $\Sigma_{j=1}^{M_t} z_j^{(p)}(k) = 1$, state measurement update (30) is reduced to the standard measurement update. Also, if $\Sigma_{j=1}^{M_t} z_j^{(p)}(k) = 0$, then $\phi(k|k) = \phi(k|k-1)$. This means that if the target is lost, no change occurs for the predicted state. Therefore, our formulation is general in that it covers a system with multiple candidates.

The results of the measurement-update appear as the values of $\phi(k|k)$ and $\tilde{z}(k)$ in the equilibrium state, where

$$\tilde{z}(k) = \lim_{p\to\infty} z^{(p)}(k) \quad (32)$$

The system parameter $\Theta = \{\sigma_x^2, \sigma_y^2, L_d, \alpha\}$ consists of three types of variables. The first among them, (σ_x^2, σ_y^2) consists the two diagonal elements of the covariance matrix R.

Since a necessary condition for the maximizer is that the gradient must be zero at equilibrium, we obtain the necessary condition as

$$\begin{cases} \sigma_x^2 = \dfrac{\Sigma_{j=1}^{M_t} \tilde{z}_j (x_j - \phi_x)^2}{\Sigma_{j=1}^{M_t} \tilde{z}_j}, \\ \sigma_y^2 = \dfrac{\Sigma_{j=1}^{M_t} \tilde{z}_j (y_j - \phi_y)^2}{\Sigma_{j=1}^{M_t} \tilde{z}_j} \end{cases} \quad (33)$$

Similarly, we obtain the necessary conditions for L_d and α as

$$L_d = \frac{1}{V}\left(M_t - \sum_{j=1}^{M_t} \tilde{z}_j\right), \quad (34)$$

and

$$\alpha = \sum_{j=1}^{M_t} \tilde{z}_j \quad (35)$$

In (33), if we use $\phi_x(k|k)$ and $\phi_y(k|k)$ instead of ϕ_x and ϕ_y, these estimates are biased, since $\phi_x(k|k)$ and $\phi_y(k|k)$ are estimates. This bias can be corrected by rearranging (33) to use $\phi_x(k|k)$ instead of ϕ_x as follows:

$$\begin{aligned} \sigma_x^2 &= \frac{\Sigma_{j=1}^{M_t} \tilde{z}_j [x_j - \phi_x(k|k) + \phi_x(k|k) - \phi_x]^2}{\Sigma_{j=1}^{M_t} \tilde{z}_j} \\ &\approx \frac{\Sigma_{j=1}^{M_t} \tilde{z}_j [x_j - \phi_x(k|k)]^2}{\Sigma_{j=1}^{M_t} \tilde{z}_j} + [\phi_x(k|k) - \phi_x]^2, \\ &\approx \frac{\Sigma_{j=1}^{M_t} \tilde{z}_j [x_j - \phi_x(k|k)]^2}{\Sigma_{j=1}^{M_t} \tilde{z}_j} + P_{xx} \end{aligned} \quad (36)$$

where P_{xx} is the variance of ϕ_x, i.e., the first diagonal component of the covariance estimate P. To obtain (36), we assume that the measurement and estimate errors are mutually independent. The value σ_y^2 can be obtained in similar manner.

In order to get more stable estimates, it is better to average the parameters over time. Hence, we add one more step for time averaging with a window of size D, given by

$$\theta^0(k) = \frac{1}{D} \sum_{i=0}^{D-1} \theta(k-1), \quad (37)$$

where $\theta^0(k) = (\sigma_x^{20}(k), \sigma_y^{20}(k), L_d^0(k), \alpha^0(k))^T$.

Ⅳ. Covariance Matrix Update

The original EM method mentioned nothing about the covariance matrix of the estimate. Meng and Rubin[33] introduced an algorithm called SEM to resolve this issue. With the aid of their method, we can derive here the exact covariance matrix to meet our needs.

The EM algorithm implicitly defines a mapping

$$\phi^{p+1}(k|k) = M[\phi^{(p)}(k|k), \theta^0(k)], \text{ for } p = 0, 1, \cdots, \quad (38)$$

from the parameter space $\phi(k)$ to itself, where

$M[\phi^{(p)}(k|k), \theta^0(k)] \triangleq \arg\max_{\phi(k)}$

$$\{Q[\phi(k)|\phi^{(p)}(k|k),\ \theta^0(k)]-K[\phi(k)]\},\quad (39)$$

Using the missing information principle by Orchard and Woodbury[34] and Theorem 4 given by Dempster[18], Meng and Rubin[33] showed that

$$P(k)=[I_d-J\{\phi(k|k)\}]^{-1}P_c(k),$$
$$=P_c(k)+\triangle P(k) \quad (40)$$

where I_d is a $d\times d$ identity matrix, $J[\phi(k|k)]$ is the Jacobian matrix for $M[\phi(k|k),\ \theta^0(k)]$ having the (i,j)th element

$$J_{i,j}[\phi(k|k)]\triangleq\frac{\partial M_i[\phi(k|k),\ \theta^0(k)]}{\partial\phi_j(k|k)} \quad (41)$$

and $P_c(k)$ is the complete data covariance matrix. $\triangle P(k)$ is given by

$$\triangle P(k)=[I_d-J\{\phi(k)\}]^{-1}J\{\phi(k|k)\}P_c(k) \quad (42)$$

$P_c(k)$ represents the ordinary prediction error in Kalman filtering, where only single observation is available, and $\triangle P(k)$ represents a newly added variance, where multiple candidates are available.

As derived in[19], the complete data covariance matrix can be written as

$$P_c(k)=\left\{I_c\{\phi(k)\}+\frac{\partial^2 K[\phi(k)]}{\partial\phi(k)\partial\phi^T(k)}\right\}^{-1} \quad (43)$$

where the expected complete data information matrix is given by

$$\mathcal{I}_c\{\phi(k)\}=E\{I_c[\phi(k)|X(k)]|w(k)\} \quad (44)$$

and the complete data information is

$$I_c(\phi|x)=\frac{\partial^2\log p[x(k)|\phi(k),\ \theta(k)]}{\partial\phi(k)\partial\phi^T(k)]} \quad (45)$$

To compute $P_c(k)$, we must know the complete data log-likelihood and the penalty function. By discarding the terms do not have $\phi(k)$, the complete data log-likelihood becomes simpler:

$$\log[p(x(k)|\phi(k),\ \theta(k)]=-\sum_{k=1}^{M_i(k)}\frac{Z_j(k)}{2}\{[y_j(k)-H\phi(k)]^T R^{-1}(k)[y_j(k)-H\phi(k)]\} \quad (46)$$

From this equation, we get

$$\mathcal{I}_c\{\phi(k)\}=H^TR^{-1}H\sum_{j=1}^{M_i(k)}\tilde{z}_j(k) \quad (47)$$

From (29), the Hessian matrix of the penalty function becomes

$$\frac{\partial^2 K[\phi_t(k)]}{\partial\phi(k)\partial\phi^T(k)}=P^{-1}(k|k-1) \quad (48)$$

Substituting (47) and (48) into (43), we finally get

$$P_c(k)=[P(k|k-1)-P(k|k-1)H^TS^{-1}(k)HP(k|k-1)] \quad (49)$$

where

$$S(k)=R(k)/\sum_{j=1}^{M_i(k)}\tilde{z}-j(k)+HP(k|k-1)H^T. \quad (50)$$

Here, we have once again used the matrix identity due to Woodbury[31]. If the problem is reduced to a single measurement case, then $\Sigma_{j=1}^{M_i(k)}\tilde{z}_j(k)=1$. It is easy to verify that the resulting equation becomes the covariance update in Kalman filtering.

We assume that the number of targets, measurements, and EM iterations are N, M, and P, respectively. For the time-update, from (3), the state update needs a multiplication between the state transition matrix and the state vector. The number of multiplications of the matrix/vector multiplication is d^2, where d is the dimension of the state, and we assume d is fixed. This multiplication is done for each target, so the computational complexity for the time-update is $O(N)$. In the covariance update part of the time-update, the number of multiplication for each target is determined by d only. This leads to $O(N)$.

The measurement-update consists of the E- and M-steps. In the E-step, the number of multiplications for each target is proportional with M. This is also same for the M-step. The E- and

M-steps iterate P times to converge, hence the resulting computational complexity is $O(MNP)$.

The covariance-update is done only once after finishing the EM loop. The number of multiplications is proportional with M to calculate the Jacobian for each target. Calculation of $P_c(k)$ and $P(k)$ do not depend on the problem size. As a result, the computational complexity for the covariance update is $O(NM)$.

Finally for the parameter-update, $\sigma_x^2(k)$ and $\sigma_y^2(k)$ each need $2M$ multiplications for each target. $L_d(k)$ and $\alpha(k)$ need $M-1$ additions for each target. So the computational complexity for parameter- update is $O(MN)$.

Note that most of the time must be spent in the measurement-update in this filter. If this filter is realized in parallel for each target, however, only $O(MP)$ computations are needed for each processor.

V. Experimental Results

In this section, the performance of the proposed tracking filter is examined and is compared to that of PDAF and the nearest neighbor filter (NNF). This experiment is focused on the analysis of the measurement-update with EM and parameter-update with MAP.

Some conditions and parameters are kept constant throughout the experiments, as described here. The time interval T is set at 1 s. We do not apply the process noise to the target movement generation, but $\boldsymbol{Q}=1.2106\times10^{-5}\boldsymbol{I}$ is used for the filter parameter. The threshold used for the validation gate is $\gamma=g^2=9.2$ and the corresponding gate probability P_g is 0.99.

The initial state setting is very important for proper statistics with the Monte Carlo method. In these experiments, the initial value of the state estimation covariance matrix is

$$\boldsymbol{P}(0|0)=\begin{pmatrix} \sigma^2 & \sigma^2/T & 0 & 0 \\ \sigma^2/T & 2\sigma^2/T^2 & 0 & 0 \\ 0 & 0 & \sigma^2 & \sigma^2/T \\ 0 & 0 & \sigma^2/T & 2\sigma^2/T^2 \end{pmatrix} \quad (51)$$

The initial state is generated randomly with a Gaussian distribution

$$\phi(0|0)\sim N\{\phi(0),\boldsymbol{P}(0|0)\} \quad (52)$$

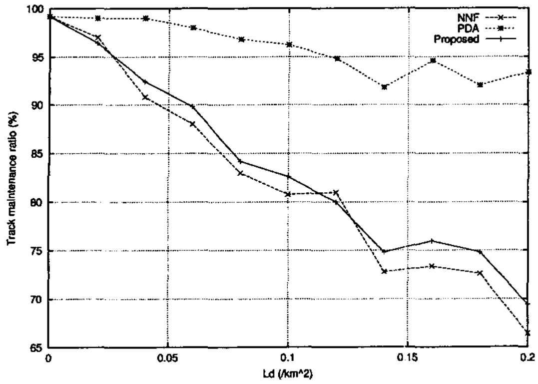
For the scenario of crossing targets, $\phi(0)=(-16.0$ km, 0.20 km · s^{-1}, 4.0 km, -0.05 km · s$^{-1})^T$ is given for the first target and $\phi(0)=(-16.0$ km, 0.20 km · s^{-1}, -4.0 km, 0.05 km · s$^{-1})^T$ for the second target. The time period during which data are collected is 120T. For a single target scenario, the initial value of the first target described above is used.

1. The Performance of Measurement-Update

In any event when a tracking filter misses the target, the magnitude of the deviation is arbitrary, hence it is useless to evaluate the tracking accuracy in this case. Even though there is no clear definition for target missing, we define that a track miss happens when the true measurement, that is known in advance in the simulator, does not appear in the gate for 20 contiguous time frames. Measured in this way, the track maintenance ratio (TMR) is defined as the ratio of tracks that are successfully traced during the observation interval. The RMS position error for the complete Monte Carlo simulation is defined as the RMS value of the differences between the estimated and true target positions for all time frames in all successful trials.

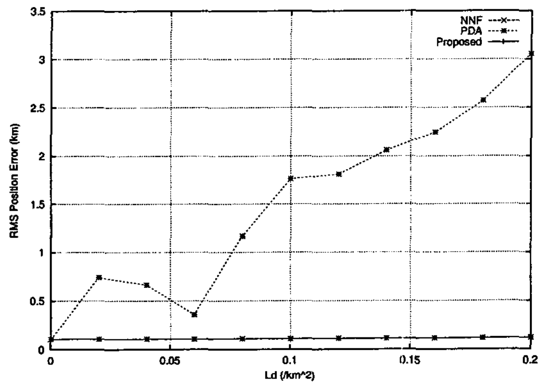
Based on these figure-of-merits, we examine the effect of L_d for fixed σ^2 and P_d on the single target scenario. For reliable statistical figures, Monte Carlo simulation was carried out for 500 runs. Given $\sigma^2=0.0225\text{km}^2$, Fig. 2 show the simulation results with $P_d=0.8$.

As far as TMR is concerned, PDAF is the best of all, NNF is the poorest, and the proposed method is in-between, though it is closer to NNF.



(a) Track maintenance ratio

(a) 트랙유지율



(b) RMS position error

(b) RMS 위치오차

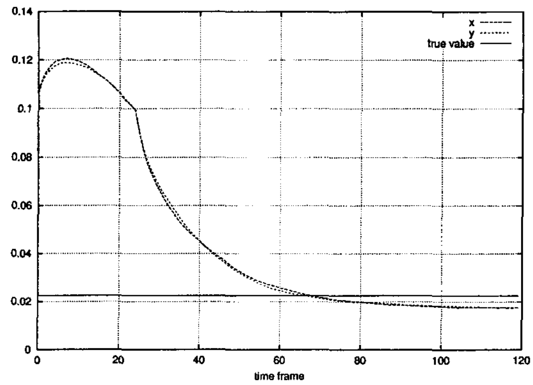
그림 2. L_d에 대한 성능변화

Fig. 2. The performance with respect to L_d, given $\sigma^2=0.0225\text{km}^2$ and $P_d=0.8$.

This fact can be expected since PDAF controls the gate size according to the known operating environment[7]. This situation becomes more prominent as L_d increases. However, the proposed method, together with NNF, are the best in term of the RMS performance. As can be seen from Fig. 2, the RMS error for PDAF increases rapidly but increases extremely slowly for the other two, looking almost constant because the error scales are so different. It is obvious that the optimal condition of high TMR and low RMS error cannot be achieved simultaneously and thus must be compromised as in the case of the new filter.

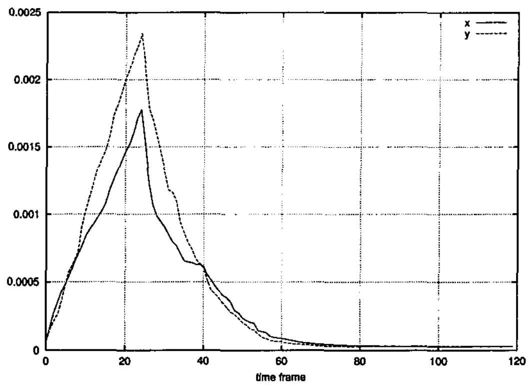
2. The Performance of Parameter-Update

All the measurements in this experiment are generated with the particular values, $\sigma^2=0.0225\text{km}^2$, and $P_d=0.9$, $L_d=0.1\text{km}^{-2}$ and $\alpha=0.891$ in the single target scenario. However, rather different values are used for the initial values of the filters: $\sigma^2=0.1\text{km}^2$, $P_d=0.8$, $L_d=0.2\text{km}^{-2}$ and $\alpha=0.792$ to investigate the effect of parameter-update. Also, the size of moving average window is D = 25 time frames.



(a) Means of $\sigma_x^2(k)$ and $\sigma_y^2(k)$

(a) $\sigma_x^2(k)$와 $\sigma_y^2(k)$의 평균



(b) Variances of $\sigma_x^2(k)$ and $\sigma_y^2(k)$

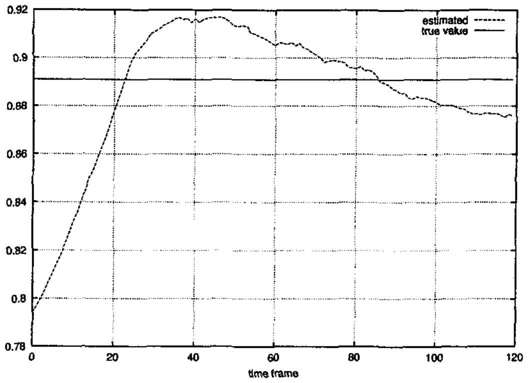
(b) $\sigma_x^2(k)$와 $\sigma_y^2(k)$의 분산

그림 3. σ_x^2 와 σ_y^2의 갱신

Fig. 3. The update of σ_x^2 and σ_y^2.

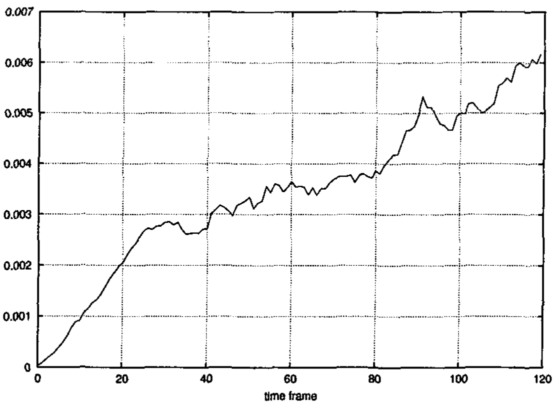
During a Monte Carlo simulation of 500 runs, we obtained the parameter estimates for each time frame, and calculated means and variances of the parameters only from the parameter estimates of the successful tracking. The variations of the mean and variance of the estimated parameter over time are plotted in Figs. 3, 4, and 5.

In Fig. 3, the means and the variances of $\sigma_x^2(k)$ and $\sigma_y^2(k)$ are drawn together. The true parameter



(a) Mean of $\alpha(k)$

(a) $\alpha(K)$의 평균

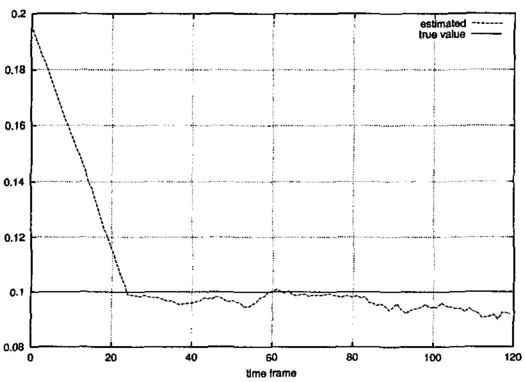


(b) Variance of $\alpha(k)$

(b) $\alpha(K)$의 분산

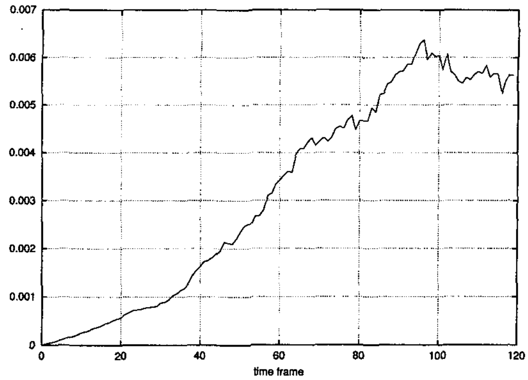
그림 4. 게이트 탐지확률 α의 갱신.

Fig. 4. The update of the gate detection probability α.



(a) Mean of $L_d(k)$

(a) $L_d(k)$의 평균



(b) Variance of $L_d(k)$

(b) $L_d(k)$의 분산

그림 5. 오경보확률 L_d의 갱신

Fig. 5. The update of the false alarm density L_d.

The means of the estimates $\sigma_x^2(k)$ and $\sigma_y^2(k)$ converge to 0.0176km^2 and the variances become $2.8\times10^{-5}\text{km}^4$. As one might notice, the estimates are slightly biased by about 0.0049km^2 below the true value. This error occurs in (36), where we discarded the term $\Sigma_{j=1}^{M_r(k)}\tilde{z}_j(k)\{x_j(k)-\phi_x(k|k)\}\{\phi_x(k|k)-\phi_x(k)\}/\Sigma_{j=1}^{M_r(k)}\tilde{z}_j(k)$. Since the true values $\phi_x(k)$ is unknown and the difference $\phi_x(k)-\phi_x(k|k)$ seems to be negligible, it was ignored.

The mean and the variance of the gate detection probability $\alpha(k)$ are drawn in Fig. 4. The estimate converges to 0.876 with variance 0.006. Also note that α is slightly underestimated by about 0.015, that is 1.7% of the true value.

Fig. 5 shows the adaptation performance with respect to the false alarm density L_d. The mean value converges to 0.092km^{-2} with variance 0.0056km^{-4}. The bias is 0.008km^{-2}, that is 8% error with respect to the true value.

Correcting the bias seems to be difficult, because estimates are based upon other estimates in a chain manner. Also, there are limits in adaptation capabilities particularly in terms of adaptation speed and parameter range.

3. The Performance of Tracking Crossing Targets

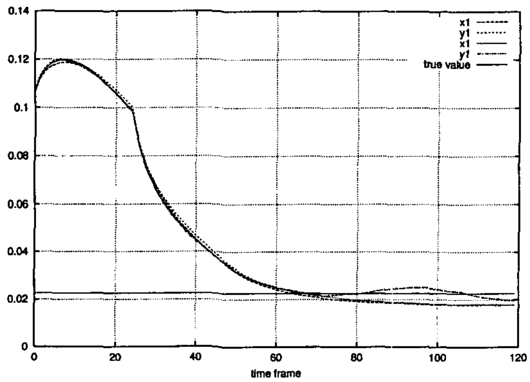
In this experiment, we examine the

parameter-update performance of the proposed filter for the crossing target case. All the settings are same as those of section 5.2.

Following a Monte Carlo simulation of 500 runs, the ensemble averages and variances have been calculated and are depicted in Figs. 6, 7, and 8. The parameter variations for both targets are shown on the same graphs.

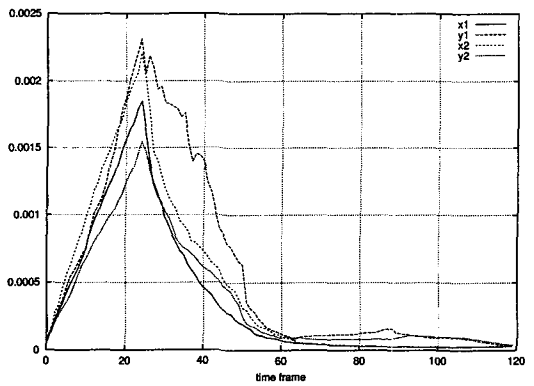
The means of the estimates $\sigma_x^2(k)$ and $\sigma_y^2(k)$ converge to 0.018km^2 and 0.020km^2 for the both targets. The variance of the estimates converge to $2.8\times10^{-5}\text{km}^4$ and $4.1\times10^{-4}\text{km}^4$. There is a tendency for the average and variance of $\sigma_y^2(k)$ to increase slightly after the targets are cross at 80T. This observation is natural since the parameter estimation must be influenced by nearby targets. Incidentally, only $\sigma_y^2(k)$ tends to rise because the two targets are separated in y axis.

In Fig. 7, the mean of the estimate $\alpha(k)$ has a variation about ±0.025 around 0.891, which amounts to 2.8% of the true value. The variance converges to 0.005. According to this graph, it seems that $\alpha(k)$ is not affected by the event of target crossing. Fig. 8 shows that the mean of the estimate $L_d(k)$ increases when targets are crossing, that is, the filter cannot discern the nearby target from the false alarms. Also, we can observe the effect of adjacent targets when they



(a) Means of $\sigma_x^2(k)$ and $\sigma_y^2(k)$

(a) $\sigma_x^2(k)$와 $\sigma_y^2(k)$의 평균

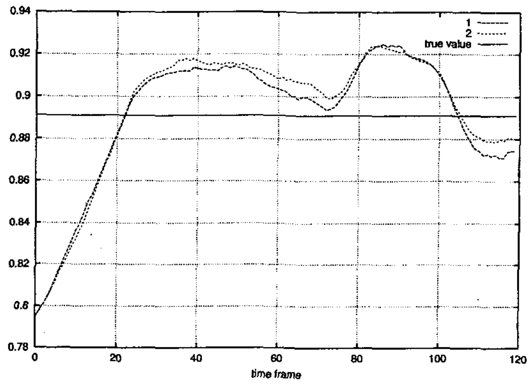


(b) Variances of $\sigma_x^2(k)$ and $\sigma_y^2(k)$

(b) $\sigma_x^2(k)$와 $\sigma_y^2(k)$의 분산

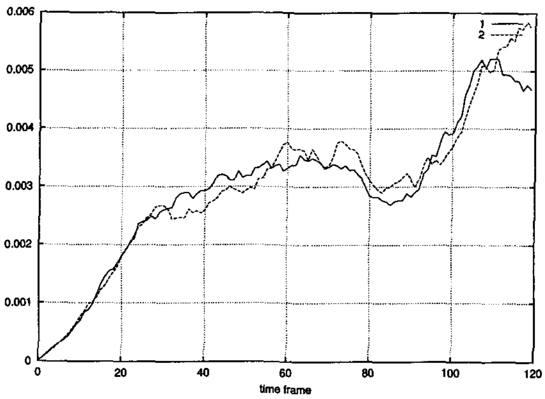
그림 6. σ_x^2와 σ_y^2의 갱신

Fig. 6. The update of σ_x^2 and σ_y^2..



(a) Mean of $\alpha(k)$

(a) $\alpha(k)$의 평균

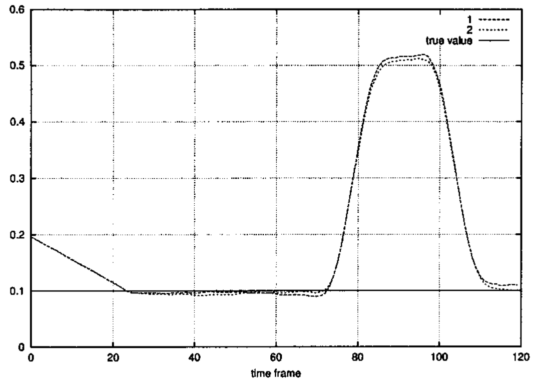


(b) Variance of $\alpha(k)$

(b) $\alpha(k)$의 분산

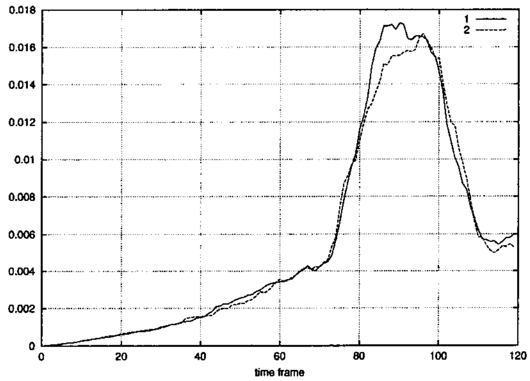
그림 7. 게이트 탐지 확률 α의 갱신

Fig. 7. The update of the gate detection probability α.



(a) Mean of $L_d(k)$

(a) $L_d(k)$의 평균



(b) Variance of $L_d(k)$

(b) $L_d(k)$의 분산

그림 8. 오경보확률 L_d의 갱신
Fig. 8. The update of the false alarm density L_d.

are approaching each other in space. There are some errors in parameter estimation, especially $L_d(k)$, when targets are crossing.

VI. Conclusion

We introduced an adaptive tracking filter following an approach based on the EM method. We improve the measurement-update by introducing validation vectors as an additional observation in EM estimation. As a result, the measurement-target association is optimally decided under the given conditions of the environment. The other contribution is the *introduction* of parameter-update that relieves us of the need to manually decide some critical values. Based on MAP, estimates of the parameters have been derived. In this manner, the filter can track multiple targets that move in a complicated manner and in variable environments.

The advantages of computational structures using this scheme are parallelism and recursiveness. Being fully parallel, each target can be monitored independently from the others. Even the system parameters can be computed separately. Also, the computation is recursive so that the previous state amounts to the entire history which reduces computation drastically. When realized in parallel for each target, only $O(MP)$ multiplications are needed for each target, where M is the number of measurements and P is the number of iterations.

Acknowledgements

This work has been supported by grants from Microwave Applications Research Center (MARC) and Agency for Defense Development (ADD) during 1999. The authors would like to thank LG Innotek Co., Ltd. for their financial support.

References

[1] S. S. Blackman. *Multiple-Target Tracking with Radar Applications*. Artech House, Inc, 1986.

[2] M. I. Skolnik. *Introduction to Radar Systems*. McGraw-Hill, New York, NY, USA, 1980.

[3] N. Levanon. *Radar Principles*. A Wiley-Interscience Publication, New York, NY, USA, 1988.

[4] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, OE-8(3):173-- 183, Jul 1983.

[5] Y. Bar-Shalom. Extension of the probabilistic

data association filter in multi-target tracking. In *Proceedings of the 5th Symposium on Nonlinear Estimation*, pages 16--21, Sep. 1974.

[6] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automation*, 11:451--460, Sep. 1975.

[7] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, Inc, 1988.

[8] Y. Bar-Shalom and X. R. Li. *Estimation and tracking: principles, techniques, and software*. Artech House, Inc, 1993.

[9] Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, Storrs, CT, 1995.

[10] D. Sengupta and R. A. Iltis. Neural solution to the multitarget tracking data association problem. *IEEE Trans. Aerosp. Electron. Syst.*, AES-25:96--108, Jan. 1989.

[11] Henry Leung. Neural network data association with application to multiple-target tracking. *Opt. Eng.*, 35(3):693--700, Mar. 1996.

[12] Gilles P. Mauroy and Edward W. Kamen. Multiple target tracking using recurrent neural networks. In *Proceedings of ICNN'97*, 1997.

[13] M. Winter and G. Favier. A neural solution for multitarget tracking based on a maximum likelihood approach. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1141--1144, May 1998.

[14] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Science*, pages 2554--2558, 1982.

[15] J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52:141--152, 1985.

[16] Y. W. Lee and H. Jeong. A MAP estimate of optimal data association for multi-target tracking. In *Proceedings of ICSPAT*, 1997.

[17] H. Jeong and J. H. Park. Multiple target tracking using constrained MAP data association. *Electronics Letters*, 35(1):25--26, Jan. 1999.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser.* B, 39(1):1--38, 1977.

[19] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. A Wiley-Interscience Publication, 1996.

[20] D. Avitzour. A maximum likelihood approach to data association. *IEEE Trans. Aerosp. Electron. Syst.*, 28(2):560--565, Apr. 1992.

[21] H. Gauvrit, J. P. Le Cadre, and C. Jauffret. A formulation of multitarget tracking as an incomplete data problem. *IEEE Trans. Aerosp. Electron. Syst.*, 33(4):1242--1257, Oct. 1997.

[22] K. J. Molnar and J. W. Modestino. Application of the EM algorithm for the multitarget/multisensor tracking problem. *IEEE Trans. Signal Processing*, 46(1):115-- 129, Jan 1998.

[23] L. Frenkel and M. Feder. Recursive expectation-maximization (EM) algorithms for time-varying parameters with applications to multiple target tracking. IEEE Trans. Signal Processing, 47(2):306--320, Feb. 1999.

[24] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, (J. Basic Eng.,)}*, 82:34--45, Mar 1960.

[25] R. E. Kalman and R. Bucy. New results in linear filtering and prediction theory. *Trans. ASME, (J. Basic Eng.,)}*, 83:95--108, Mar 1961.

[26] M. Grewal and A. Andrews. *Kalman Filtering : Theory and Practice.* Prentice-Hall, Englewood Cliffs, NJ 07632, USA, 1993.

[27] P. Green. On use of the {EM} algorithm for penalized likelihood estimation. J. *Roy. Statist. Soc. Ser.* B, 52(3):443--452, 1990.

[28] X. L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika,* 80:267--278, 1993.

[29] M. Skolnik, editor. *Radar Handbook.* McGraw-Hill, New York, NY, USA, 1990.

[30] D. G. Luenberger. *Optimization by Vector Space Methods.* John Wiley & Son, Inc., 1969.

[31] Max Woodbury. Inverting modified matrices. Memorandum Report 42, Statistical Research Group, Princeton, 1950.

[32] Alston S. Householder. *The Theory of Matrices in Numerical Analysis.* Dover Publications, New York, 1975.

[33] X. L. Meng and D. B. Rubin. Using EM to obtain asymptotic variance-covariance matrices: the SEM algorithm. *Journal of the American Statistical Association*, 86:899--909, 1991.

[34] T. Orchard and M. A. Woodbury. A missing information principle: Theory and application. In *Proceedings of the Statistical Computing Section*, pages 41--45. American Statistical Association, 1972.

저 자 소 개

丁 弘(正會員)

He received the BS degree from the EE Dept. at the Seoul National University in 1977. In 1979, he received the MS degree from the EE Dept. at KAIST. During 1984-1988, he received the SM, EE, and PhD degrees in EECS Dept. from MIT. During the period of 1979-1982, he taught at the Kyung Book National University. Since 1988, he has worked at POSTECH, where he is now an associate professor. During 1996-1997, he worked in the Bell Labs at Murray Hill. His major research area is the digital signal processing for image, speech, and radar signals.

朴 昺 浩(正會員)

He received the BS degree from the EE Dept. at the Yonsei University, in 1988, and the MS degree from the EE Dept. at POSTECH in 1990. He received PhD degree in EE Dept. at POSTECH in 2001. Since 1990 he has worked at the LG Innotek Co. in developing advanced radar system, where he is now a principal engineer. His current research interests include radar signal processing and target tracking.