A Nonlinear IMM Algorithm for Maneuvering Target Tracking

WEN-RONG WU

PEEN-PAU CHENG
National Chiao Tung University
Taiwan

In target tracking, the measurement noise is usually assumed to be Gaussian. However, the Gaussian modeling of the noise may not be true. Noise can be non-Gaussian. The non-Gaussian noise arising in a radar system is known as glint noise. The distribution of glint noise is long tailed and will seriously affect the tracking performance. We develop a new algorithm that can effectively track a maneuvering target in the glint environment. The algorithm incorporates the nonlinear Masreliez filter into the interactive multiple model (IMM) method. Simulations demonstrate the superiority of the new algorithm.

Manuscript received July 31, 1992; revised March 4, 1993.

IEEE Log No. T-AES/30/3/16654.

Authors' address: Dept. of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

I. INTRODUCTION

The Kalman filter is widely used in the tracking problem. It can optimally estimate the target motion from noisy radar data. The optimality of the Kalman filter is based on the assumption of the Gaussian noise. If the assumption is violated, the Kalman filter is no longer the optimal filter.

In a radar system, due to the target glint, the measurement noise may present non-Gaussian behavior. This is referred to as the glint noise. Hewer, Martin, and Zeh analyzed the glint noise in [10] and concluded that the distribution of the glint noise is long tailed. It is well known that the conventional minimum mean square estimate can be seriously degraded if noise is non-Gaussian [17]. Unfortunately, very few results have been reported regarding this problem and the standard Kalman filter is continuously used in tracking applications. Hewer, et al. [10] approximated the glint noise by a mixture of a Gaussian noise and outliers and employed robust estimation techniques to preprocess (clean) the radar data. Wu [20] used a nonlinear filter that was originally proposed by Masreliez [15, 16] to track a target maneuvering in a Markov fashion. The Masreliez filter employs a nonlinear score function as the correction term in the state estimate and the results are often nearly optimal.

The implementation of score function is difficult except for simple cases. Wu and Kundu developed an efficient approximation method in [19]. This method employed an adaptive normal expansion to expand the score function and truncates the higher order terms in the expanded series. Consequently, the score function can be approximated by a few central moments of the observation prediction density. The normal expansion is made adaptive by using the concept of conjugate recentering and the saddle point method. This method was used in [20].

Here we develop a new algorithm, which is an extension of [20], to track a maneuvering target in the glint environment. There exist many maneuvering targets tracking algorithms [1–9]. Among them, the interactive multiple model (IMM) method [4] provides good performance with efficient computation. However, if the observation noise is non-Gaussian, the IMM method degrades. This degradation is due to the nonoptimal Kalman filter used in the IMM and the miscalculation of the model probabilities. To remedy the problem, we propose to use the Masrelize filter in place of the Kalman filter and correct the model probabilities. This results in a nonlinear IMM algorithm (NIMM). We show that the NIMM can significantly improve the tracking performance.

The organization of the paper is as follows. In Section II, the Masreliez's approach and the implementation issue of the score function is briefly reviewed. In Section III, we describe the IMM method and the new algorithm. In Section IV, we present some simulation results and draw the conclusion in Section V.

II. SCORE FUNCTION APPROACH

A. General Filtering Problem

The general filtering problem can be formulated as the estimation of the state given all the history of the observations. Consider a linear system described as follows:

$$x_{k+1} = \phi_k x_k + w_k \tag{1}$$

$$z_k = H_k x_k + v_k \tag{2}$$

where x_k is the state vector, w_k and v_k represent white noise sequences and are assumed to be mutually independent. The basic problem is to estimate the state x_k from the noisy observations $(z_1,\ldots,z_k)$. The probability density of the state conditioned on all the available observation data is called the *a posteriori density*. If this density is known, an estimation for any type of performance criterion can be found. Thus, the estimation problem can be viewed as the problem of determining the *a posteriori* density. For computational efficiency, one is frequently interested in performing the filtering recursively. The recursive determination of the *a posteriori* density is generally referred to as the Bayesian approach [12]. Denote $f(\cdot)$ as a density and $Z^k = \{z_0, z_1, \ldots, z_k\}$. The Bayesian approach is described by the following relations [12]

$$f(x_k \mid Z^k) = \frac{f(x_k \mid Z^{k-1}) f(z_k \mid x_k)}{f(z_k \mid Z^{k-1})} \tag{3}$$

$$f(x_k \mid Z^{k-1}) = \int f(x_{k-1} \mid Z^{k-1}) f(x_k \mid x_{k-1}) dx_{k-1} \tag{4}$$

where the normalizing constant $f(z_k \mid Z^{k-1})$ is given by

$$f(z_k \mid Z^{k-1}) = \int f(x_k \mid Z^{k-1}) f(z_k \mid x_k) dx_k. \tag{5}$$

The $f(z_k \mid x_k)$ in (3) is determined by the observation noise density $f(v_k)$ and (2). Similarly, $f(x_k \mid x_{k-1})$ in (4) is determined by the state noise density $f(w_k)$ and (1). Theoretically, knowing these densities, we can determine the *a posteriori* density $f(x_k \mid Z^k)$. However, it is generally impossible to carry out the integration in (4) for every instant. Consequently, the *a posteriori* density cannot be determined for most applications. There is only one exception, i.e., when the initial state and all the noise sequences are Gaussian. In this case, (3) and (4) are reduced to the standard Kalman filter equations.

B. Score Function Approach

In this section, we briefly review Masreliez's algorithm. Consider a linear system described in (1)–(2). w_k and v_k can be non-Gaussian. $f(z_k \mid Z^{k-1})$ denote the density of z_k conditioned on the *prior* observations. We name $f(z_k \mid Z^{k-1})$ the *observation prediction density* and assume that it is twice differentiable. Similarly, $f(x_k \mid Z^{k-1})$ is the density of x_k conditioned on *prior* observations and is named the *state prediction density*. The filtering problem is to estimate the state vector x_k from the noisy observations Z^k. Assuming that $f(x_k \mid Z^{k-1})$ is a Gaussian density with mean $\overline{x}_k$, and covariance matrix M_k, Masreliez has shown that the minimum variance state estimation $\hat{x}_k$, and its covariance matrix $P_k = E\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)^t \mid Z^k\}$ can be recursively calculated as follows [15]

$$\hat{x}_k = \overline{x}_k + M_k H_k^t g_k(z_k) \tag{6}$$

$$P_k = M_k - M_k H_k^t G_k(z_k) H_k M_k \tag{7}$$

$$\overline{x}_{k+1} = \phi_k \hat{x}_k \tag{8}$$

$$M_{k+1} = \phi_k P_k \phi_k^t + Q_k \tag{9}$$

where $g_k(\cdot)$ is a column vector with components:

$$\{g_k(z_k)\}_i = -\left[\frac{\partial f(z_k \mid Z^{k-1})}{\partial (z_k)_i}\right] [f(z_k \mid Z^{k-1})]^{-1} \tag{10}$$

and $G_k(z_k)$ is a matrix with elements

$$\{G_k(z_k)\}_{ij} = \frac{\partial \{g_k(z_k)\}_i}{\partial (z_k)_j}. \tag{11}$$

The function $g_k(\cdot)$ is called the score function of $f(z_k \mid Z^{k-1})$. It is $g_k(\cdot)$ that suggests how to modify the Kalman filter in the non-Gaussian noise. Assuming that w_k is Gaussian and v_k is non-Gaussian, we can see that the score function $g(\cdot)$ operating on the residual $z_k - H_k \overline{x}_k$ will deemphasize the influence of large residuals when the observation prediction density is long tailed, and, on the other hand, emphasize the large residuals when the observation density is short tailed. This is intuitively appealing. It is easy to check that the filter is reduced to the standard Kalman filter if the initial state, w_k, and v_k (for all ks) are Gaussian. The following procedure summarizes the implementation of the filter.

Step 0 Assume that at stage $k-1$, $\hat{x}_{k-1}$ and P_{k-1} are known.

Step 1 Calculate $M_k = \phi_{k-1} P_{k-1} \phi_{k-1}^t + Q_{k-1}$.

Step 2 Approximate the state prediction density $f(x_k \mid Z^{k-1})$ by a Gaussian distribution with mean $\overline{x}_k = \phi_{k-1}\hat{x}_{k-1}$ and covariance matrix M_k.

Step 3 Find the observation prediction density $f(z_k \mid Z^{k-1})$ by convolving $f(H_k x_k \mid Z^{k-1})$ with $f_{v_k}(\cdot)$.

Step 4 Find $g_k(z_k)$ and $G_k(z_k)$.

Step 5 Apply (6) and (7) to find $\hat{x}_k$ and P_k.

Step 6 Let $k \rightarrow k+1$ and start all over from Step 1.

The procedure outlined above is straightforward in principle. However, the convolution operation in Step 3 is difficult to implement in general except for very simple cases. Also, in Step 4, the differentiation operations involved in the evaluation of the score function and its derivative are not trivial. In the following, we briefly describe the score function approximation method developed in [19].

C. Evaluation of Score Function

It has been shown [25] that under certain regularity conditions, a distribution can be expanded by the normal distribution and its associated orthogonal polynomials. Let $g(\cdot)$ be a distribution with zero mean and unit variance, $M(t)$ be its moment generating function (MGF), and $K(T) = \ln[M(T)]$. The normal expansion is described as follows [22]

$$g(x) = \phi(x)\left\{1 + \frac{\rho_3}{3!}H_3(x) + \frac{\rho_4}{4!}H_4(x) + \frac{\rho_5}{5!}H_5(x) + \frac{(\rho_6 + 10\rho_3^2)}{6!}H_6(x) + \cdots\right\} \tag{12}$$

where ρ_n is the standardized cumulant that is defined as $K^{(n)}(0)/K^{(2)}(0)$, $K^{(n)}(\cdot)$ denotes the nth derivative of $K(\cdot)$, and $H_n(\cdot)$ is the *Hermite* polynomials. An approximation can be made by retaining several terms in the expanded series given by (12). This is called the normal approximation. The basic idea of the adaptive normal expansion is to use a low-order expansion at each point of the distribution rather than to use a high-order expansion for the whole distribution at a single point. Suppose that we want to approximate the distribution at a point, say, x_0. We first transform the original distribution to a distribution that has its mean at x_0. Then, we apply the normal approximation on the transformed distribution and evaluate it at x_0. Since the normal approximation is good around the mean, this approach will yield a better result than the straightforward normal approximation. The procedure for the transformation is called *recentering* [23] and the transformed distribution is called the *conjugate density* [24]. We now formally state the definition.

DEFINITION A density $g(z)$ is called the conjugate density of $f(x)$ at a point x_0 if there are constants α and T_0 such that

$$(1) \quad z = x - x_0 \tag{13}$$

$$(2) \quad g(z) = \alpha e^{T_0 z} f(z + x_0) \tag{14}$$

$$(3) \quad \int_{-\infty}^{+\infty} g(z) = 1, \qquad \int_{-\infty}^{+\infty} z g(z)\,dz = 0. \tag{15}$$

$g(z)$ can be normalized and expanded as in (12). If we only retain the first term, we have

$$f(x_0) = \alpha^{-1} g(0) \approx \frac{\phi(0)}{\alpha\sigma_z} \tag{16}$$

where σ_z is the standard deviation of $g(z)$. From [20], it is known that if we choose T_0 and α such that

$$K'(T_0) - x_0 = 0 \tag{17}$$

$$1/\alpha = e^{K(T_0) - T_0 x_0} \tag{18}$$

then, $g(s) = \alpha e^{T_0 s} f(s + x_0)$ is the conjugate density.

The solution of $K'(T) - x_0 = 0$ is referred to as the *saddle point* of $e^{-T x_0} M(T)$. It is shown in [21] that under some regularity condition a unique real saddle point exists.

The MGT of a distribution is nothing but the Laplace transform of the distribution. One of the most important properties of Laplace transform is that the convolution in time domain or spatial domain can be transformed into multiplication in frequency domain. This property is directly applicable to the MGFs. This is also the key concept that we can avoid the convolution operation in the estimation of the score function as required in Masreliez's approach.

The distribution approximation technique discussed above can be extended to find the approximation of score function. The idea is to find the expansion of $f(\cdot)$ and $f'(\cdot)$ via the conjugate recentering. From that, we find the expansion of $f'(\cdot)/f(\cdot)$ and truncate it to obtain the approximation. Thus (detailed derivation, see [20])

$$-\frac{f'(x_0)}{f(x_0)} \approx T_0 + \frac{\rho_{z,3}}{2\sigma_z} = T_0 + \frac{\mu_{z,3}}{2\sigma_z^4} \tag{19}$$

where $\mu_{z,i}$ stands for the ith central moment of $g(z)$ and T_0 is obtained from (17). To implement Masreliez's filter, we also need the derivative of the score function. One can show that

$$\frac{d}{dx}\left(-\frac{f'}{f}\right)\bigg|_{x=x_0} \approx \left\{\frac{dT}{dx} + \frac{dT}{dx}\frac{d}{dT}\left[\frac{K^{(3)}(T)}{2K''(T)^2}\right]\right\}\bigg|_{T=T_0} \tag{20}$$

$$\approx \frac{1}{\sigma_z^2}\left[1 + \frac{\mu_{z,4}}{2\sigma_z^4} - \frac{\mu_{z,3}^2}{\sigma_z^6}\right]. \tag{21}$$

In the above derivation, we assume that the distribution under consideration is univariate. It turns out that the score function approximation scheme is simple and computationally efficient. It is not difficult to extend the result to the multivariate distributions. However, in this case, the approximation formula becomes very complicated. For the target tracking problem, the distribution of the observation noise is usually not univariate. However, it is possible to decouple the distribution into univariate ones by some coordinate transformation [20]. By doing so, the scalar score function approximation method can be applied.

III. MANEUVERING TARGET TRACKING

A. IMM Algorithm

Let a system be described by

$$x_k = \Phi(\theta_k)x_{k-1} + B(\theta_k)w_k \tag{22}$$

$$z_k = H(\theta_k)x_k + v_k \tag{23}$$

where θ_k is a finite state Markov process taking values in $\{\theta^1,\ldots,\theta^N\}$ according to a transition probability matrix A, and w_k, v_k are mutually independent of white Gaussian processes. For notation simplicity, we define

$$\{\theta_k = \theta^i\} \qquad \text{as} \quad \{\theta_k^i\} \tag{24}$$

$$A_{ij} = P_r(\theta_k^i \mid \theta_{k-1}^j) \tag{25}$$

$$\hat{p}_{k-1}^i = f(\theta_{k-1}^i \mid Z^{k-1}) \tag{26}$$

$$\overline{p}_k^i = f(\theta_k^i \mid Z^{k-1}). \tag{27}$$

Assume that

$$f(x_{k-1} \mid \theta_{k-1}^i, Z^{k-1}) \sim N(\hat{x}_{k-1}^i, \hat{P}_{k-1}^i) \tag{28}$$

$$f(x_{k-1} \mid \theta_k^i, Z^{k-1}) \sim N(\tilde{x}_{k-1}^i, \tilde{P}_{k-1}^i) \tag{29}$$

$$f(x_k \mid \theta_k^i, Z^{k-1}) \sim N(\overline{x}_k^i, \overline{P}_k^i). \tag{30}$$

The IMM algorithm cycle can be summarized in the following four steps.

1) Starting with the N weights $\hat{p}_{k-1}^i$, the N means $\hat{x}_{k-1}^i$ and the N associated covariances $\hat{P}_{k-1}^i$, we compute the mixed initial condition for the filter matched to θ_k^i

$$\overline{p}_k^i = \sum_j A_{ij}\hat{p}_{k-1}^j \tag{31}$$

$$\tilde{x}_{k-1}^i = \sum_j A_{ij}\hat{p}_{k-1}^j \hat{x}_{k-1}^j / \overline{p}_k^i \tag{32}$$

$$\tilde{P}_{k-1}^i = \sum_j A_{ij}\hat{p}_{k-1}^j [\hat{P}_{k-1}^j + (\hat{x}_{k-1}^j - \tilde{x}_{k-1}^i)(\cdot\,\cdot)^T]/\overline{p}_k^i. \tag{33}$$

2) Each pair of $\tilde{x}_{k-1}^i, \tilde{P}_{k-1}^i$ is used as input to a Kalman filter to obtain the predicted $\overline{x}_k^i, \overline{P}_k^i$ and measurement updated $\hat{x}_k^i, \hat{P}_k^i$.

3) The N weights $\overline{p}_k^i$ are updated from the innovations of the N Kalman filters

$$\hat{p}_k^i = c \cdot \overline{p}_k^i \cdot \|S_k^i\|^{-1/2} \exp\{-1/2\vartheta_i^T (S_k^i)^{-1} \vartheta_k^i\} \tag{34}$$

where c is a nomenclature constant.

$$\vartheta_k^i = z_k - H(\theta_k^i)\overline{x}_k^i \tag{35}$$

$$S_k^i = H(\theta_k^i)\overline{P}_k^i H(\theta_k^i)^T + R_k \tag{36}$$

where $R_k = E\{v_k v_k^T\}$.

4) The output $\hat{x}_k$ and $\hat{R}_k$ are computed according to

$$\hat{x}_k = \sum_i \hat{p}_k^i \hat{x}_k^i \tag{37}$$

$$\hat{P}_k = \sum_i \hat{p}_k^i [\hat{P}_k^i + (\hat{x}_k^i - \hat{x}_k)(\cdot\,\cdot)^T]. \tag{38}$$

B. Nonlinear IMM Algorithm

The standard IMM algorithm assumes that the measurement noise is Gaussian. When this assumption is violated, the performance is degraded. This degradation is mainly due to Step 2) and 3) in the IMM cycle. In Step 2), the standard Kalman filter is used to perform the estimation. However, as we mentioned above, when the measurement noise is non-Gaussian, the performance of the Kalman filter can be greatly affected. To remedy this problem, we propose to replace the Kalman filter by the Masreliez filter and use the score function approximation method mentioned in Section II. Let z_k be scalar and the MGF of v_k be known. In the following, we summarize the filtering procedure.

1) *Find the MGF of* $f(z_k \mid \theta_k^i, Z^{k-1})$:

Since $f(x_k \mid \theta_k^i, Z^{k-1})$ is Gaussian with mean $\overline{x}_k^i$ and covariance $\overline{P}_k^i$, we can find the MGF of $f(H(\theta_k^i)x_k \mid \theta_k^i, Z^{k-1})$ easily. Let the MGF of $f(v_k)$ and $f(H(\theta_k^i)x_k \mid \theta_k^i, Z^{k-1})$ be $M_v(T)$ and $M_x(T)$, respectively. We can obtain $f(z_k \mid \theta_k^i, Z^{k-1})$ by the convolution of $f(v_k)$ and $f(H(\theta_k^i)x_k \mid \theta_k^i, Z^{k-1})$. Thus, the MGF of $f(z_k \mid \theta_k^i, Z^{k-1})$ is $M_v(T)M_x(T)$.

2) *Find the conjugate density of* $f(z_k \mid \theta_k^i, Z^{k-1})$ *at* z_k:

Let $K(T) = \ln[Mv(T)M_x(T)]$. The conjugate density of $f(z_k \mid \theta_k^i, Z^{k-1})$ at z_k is constructed as $g(s) = \alpha_k e^{T_k s} f(s + z_k)$ where T_k is chosen as the saddle point of $\{M_v(T)M_x(T)e^{-Tz_k}\}$, i.e,

$$K'(T_k) - z_k = 0 \tag{39}$$

$$1/\alpha_k = M_v(T_k)M_x(T_k)e^{-T_k z_k}. \tag{40}$$

3) *Find the second, third, and the fourth moment of* $g(s)$:

These moments can be found by

$$\begin{aligned} \sigma_s^2 &= K^{(2)}(T_k), \\ \mu_{s,3} &= K^{(3)}(T_k), \\ \mu_{s,4} &= K^{(4)}(T_k) \end{aligned} \tag{41}$$

where $K^{(i)}(T)$ denotes the ith derivative of $K(T)$.

4) *Approximate the score function of* $f(z_k \mid \theta_k^i, Z^{k-1})$ *and its derivative*:

$$-\frac{f'(z_k \mid \theta_k^i, Z^{k-1})}{f(z_k \mid \theta_k^i, Z^{k-1})} \approx T_k + \frac{\mu_{s,3}}{2\sigma_s^2} \tag{42}$$

$$\frac{d}{dz_k}\left(-\frac{f'}{f}\right) \approx \frac{1}{\sigma_s^2}\left[1 + \frac{\mu_{s,4}}{2\sigma_s^4} - \frac{\mu_{s,3}^2}{\sigma_s^6}\right]. \tag{43}$$

5) *Perform the filtering*:

Use (6)–(9) to obtain the filtered value of $\hat{x}_k^i$ and $\hat{P}_k^i$ for $i = 1,\ldots,N$ and put the results into IMM.

In Step 3) of the IMM cycle, we use (34) to compute the updated model probabilities. Equation (34) is obtained by using the Bayes' formula.

$$\hat{p}_k^i = f(\theta_k^i \mid Z^k) = \frac{f(\theta_k^i \mid Z^{k-1}) f(z_k \mid \theta_k^i, Z^{k-1})}{f(z_k \mid Z^{k-1})} \tag{44}$$

the distribution $f(z_k \mid \theta_k^i, Z^{k-1})$ is obtained by the convolution of $f(H(\theta_k^i)x_k \mid \theta_k^i, Z^{k-1})$ with $f(v_k)$. When the measurement noise is Gaussian, (44) is reduced to (34). However, when $f(v_k)$ is non-Gaussian, (34) is not valid. It is difficult to implement the convolution exactly. Here, we propose to use the distribution approximation method described in (16) to evaluate the convoluted distribution.

$$\hat{p}_k^i \approx c \cdot \frac{\overline{p}_k^i \phi(0)}{\alpha_k \sigma_s} \tag{45}$$

where $\phi(0) = 1/\sqrt{2\pi}$, α_k and σ_s are same as those in (40)–(41).

C. Non-Gaussian Glint Noise

As we have described before the glint noise is clearly non-Gaussian and long tailed. Now, the problem is how to model it. Borden and Mumford [26] consider the distribution of the glint as a student's t distribution with two degrees of freedom and develop a method to produce glint-like signals. From the empirical studies, Hewer, Martin, and Zeh [10] argue that the glint can be modeled as a mixture of a Gaussian noise and outliers. Their results are based on the analysis of normal QQ-plots of glint noise records. Indeed, from [10, Fig. 2], we can see that the QQ-plot is fairly linear around the origin. This indicates that the distribution is Gaussian-like around its mean. But, in the tail region, the plot deviates the linearity and indicates a non-Gaussian long-tailed character. The data in the tail region is essentially associated with the glint spikes and are considered to be outliers. They are modeled as a Gaussian noise with large variance. This leads to the Gaussian-mixture noise model. Although this model is simple, it is not suitable for our use. It is easy to see that the score function of the Gaussian mixture is not robust. The score function of a Gaussian mixture will increase without bound as the measurement goes to infinity. Here, we propose to model the glint as a mixture of a Gaussian and a Laplacian noise, i.e.,

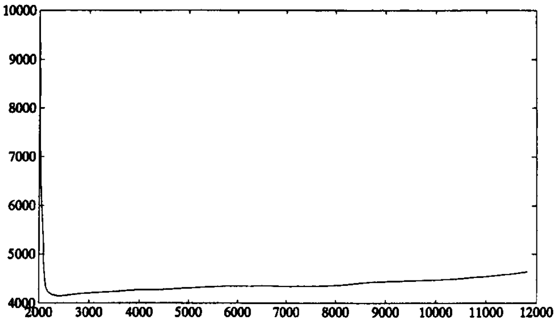


Fig. 1. Target trajectory.

$$f_t(x) = (1-\epsilon) f_g(x) + \epsilon f_l(x) \tag{46}$$

$$f_g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{x^2/2\sigma^2}, \qquad f_l(x) = \frac{1}{2\eta} e^{|x|/\eta} \tag{47}$$

where $f_t(\cdot)$, $f_g(\cdot)$, and $f_l(\cdot)$ represent the glint, the Gaussian, and the Laplacian distribution, respectively. The ϵ is a small positive number less than one. The variance of $f_l(\cdot)$ is larger than that of $f_g(x)$. One can show that the score function of this distribution is robust. It is also interesting to note that this score function is very similar to some psi-function in the robust M-estimator.

IV. SIMULATIONS

In this section, we perform some simulations to evaluate our new algorithm. For simplicity, we assume that the target is in a 2-D space and its position is sampled every $T = 10$ s. The target is moving with a constant course with a process noise $Q_k = 0.001$ until $k = 40$ when it starts a 90 deg turn. This problem is similar with those in [6, 7, 9]. The acceleration we use is

$$U_k^x = U_k^y = 0, \qquad 0 \le k \le 39 \quad \text{and} \quad 44 < k \le 100$$

$$U_k^x = U_k^y = 0.3, \qquad 40 < k \le 44.$$

The target trajectory is shown in Fig. 1. The initial condition with the state

$$X = [\dot{x}, x, \dot{y}, y] \tag{48}$$

is

$$X(0) = [0 \text{ m/s}, 2000 \text{ m}, -15 \text{ m/s}, 10000 \text{ m}]. \tag{49}$$

The IMM consists of a second order model and a third order model.

1) *Second-order model (for x axis)*:

$$X_k = [\dot{x}, x], \qquad Q_k = 0 \tag{50}$$

$$\Phi_k^1 = \begin{bmatrix} 1 & 0 \\ T & 1 \end{bmatrix}, \qquad B_k^1 = \begin{bmatrix} T \\ T^2/2 \end{bmatrix}, \qquad H_k^1 = [0,1]. \tag{51}$$

2) *Third-order model (for x axis)*:

$$X_k = [\ddot{x}, \dot{x}, x], \qquad Q_k = 0.001 \tag{52}$$

$$\Phi_k^2 = \begin{bmatrix} 1 & 0 & 0 \\ T & 1 & 0 \\ T^2/2 & T & 1 \end{bmatrix}, \quad B_k^2 = \begin{bmatrix} T \\ T^2/2 \\ T^3/6 \end{bmatrix}, \quad H_k^2 = [0,0,1]. \tag{53}$$

The probability transition matrix of two models is

$$A = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}. \tag{54}$$

The parameters for the glint noise generation are chosen as (according to (46)–(47))

$$\epsilon = 0.1, \qquad \sigma = 100, \qquad \eta = 400. \tag{55}$$

To illustrate the effect of the noise variance used in the Kalman filter, we consider two cases for IMM. In the first case, we ignore the Laplacian noise in the filtering process, i.e., the variance of the observation noise is just σ^2 (the variance of the Gaussian component). The other case is that we count in the variance of the Laplacian noise. In other words, the variance of the observation noise is $(1-\epsilon)\sigma^2 + 2\epsilon\eta^2$. A Monte Carlo simulation of 500 runs is performed. Figs. 2–3 and 5–6 show the average rms errors of the position and the velocity estimate. Figs. 4 and 7 show the average probability estimates of model one. Figs. 8–11 illustrate the results of a sample run for x axis (the results in y axis is similar and therefore is omitted). For notational simplicity, we denote the first case of IMM as IMM1, the second case of IMM as IMM2, and the nonlinear IMM as NIMM. From these figures, we can see that the performance of our new NIMM is significantly better than those of IMM1 and IMM2. The improvement is due to the nonlinear filtering and the model probability correction. Since the Kalman filter is not optimal in the non-Gaussian noise, when high variance Laplacian noise arises, it produces big error. In Fig. 8, we see that the Laplacian noise arises in steps around 38, 60, and 77. The estimation errors of both IMMs are then large in those places. On the contrary, the Masreliez filter in NIMM is nearly optimal and completely removes the Lapalcian noise. This is shown in Figs. 9 and 10. Note that due to the larger noise variance in IMM2, the Kalman gain is smaller. Consequently, the estimation error of IMM2 is smaller than that of IMM1. However, when the Gaussian noise arises, the Kalman gain in IMM2 becomes too small. This results in an overfiltering problem. This effect is hidden by the model uncertainty and is not obvious in our results. Under the Gaussian noise, the filter gain of IMM1 is optimal and that of NIMM is nearly optimal. We can say that the Masreliez filter uses a variable gain to approach the optimal performance in both noises. This may be best depicted by the score functions of the noise distributions. Though the score functions are not the real ones used in the filtering operation, the general shapes are similar. They are shown in Fig. 12 (the figure is normalized to σ). We can clearly see that the score function in NIMM is nonlinear. It almost coincides with the score function in IMM1

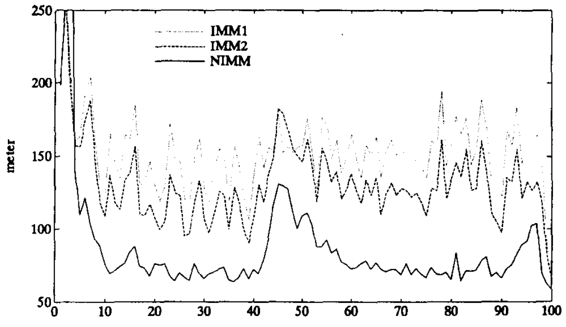


Fig. 2. RMS error of position estimate in x axis.

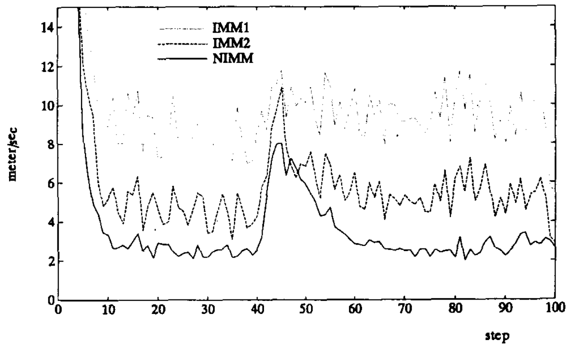


Fig. 3. RMS error of velocity estimate in x axis.

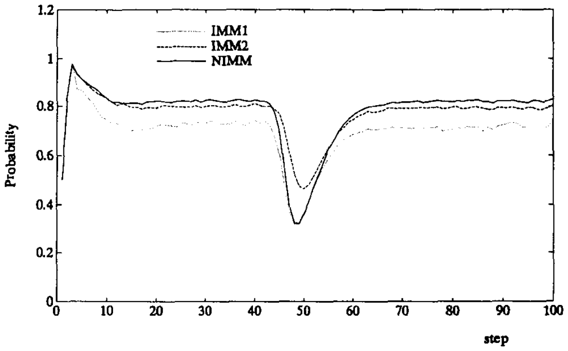


Fig. 4. Probability estimate of model 1 in x axis.

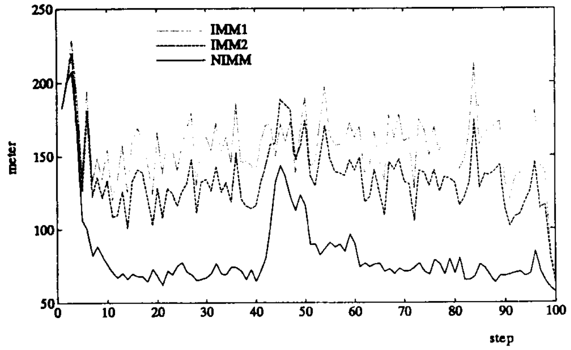


Fig. 5. RMS error of position estimate in y axis.

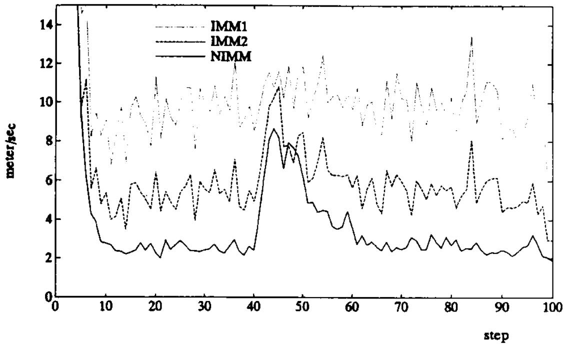


Fig. 6. RMS error of velocity estimate in y axis.

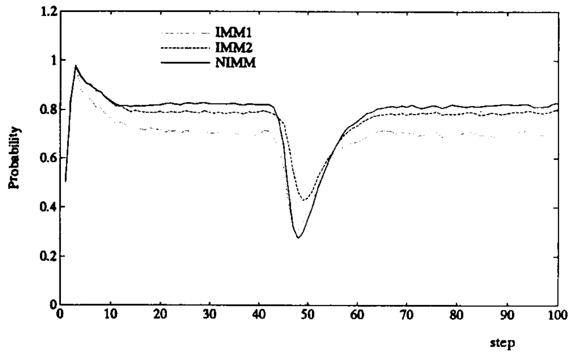


Fig. 7. Probability estimate of model 1 in y axis.

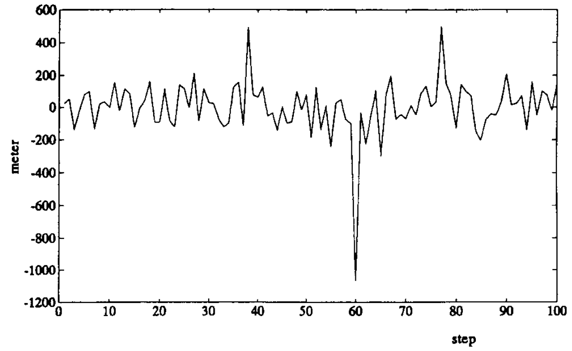


Fig. 8. Noise in x axis.

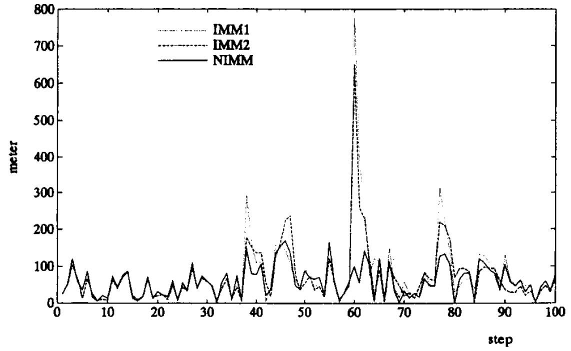


Fig. 9. RMS error of position estimate in x axis.

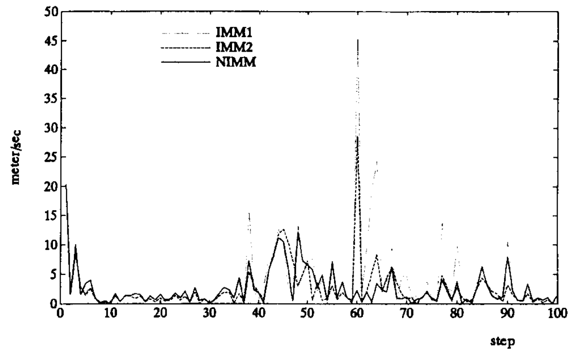


Fig. 10. RMS error of velocity estimate in x axis.

in the central region. If the residual signal falls in this regions, the Masreliez filter considers noise to be Gaussian. However, the score function descents to a small constant value in the tail region where the Masreliez filter treats noise to be Laplacian.

The estimate of model probability in IMM is based on (35), (36). We see that the residual signal and the noise variance play the main roles. Large residual signal can be caused by Lapalcian noise or target maneuvering and IMM cannot distinguish them. Increasing variance implies that the probability estimate favors model one. It is likely to treat large residual signal as the cause of noise instead of maneuvering. Consequently, the capability of maneuvering detection is decreased. On the contrary, decreasing variance implies that the probability estimate favors model two. As shown in Figs. 4 and 7, the probability estimate of model one of IMM2 in unmaneuvering period is better than that of IMM1. However, in the maneuvering period, the results are opposite. From Fig. 11, we see that in the unmaneuvering period, the probability estimate is smooth for IMM2. However, it is more noise-like for IMM1. This reflects the maneuvering detection sensitivity of IMM1. The probability estimate of NIMM retains the high maneuvering detection

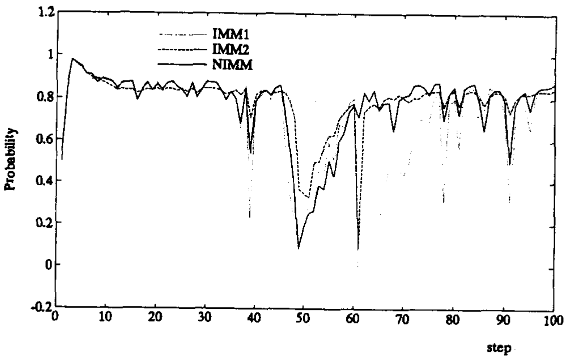


Fig. 11. Probability estimate of model 1 in x axis.

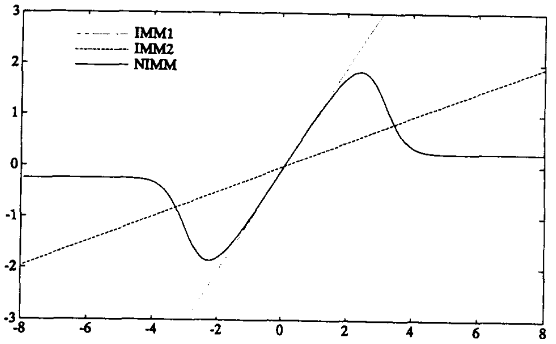


Fig. 12. Score functions of noise distributions.

sensitivity of IMM1 while reduces the effect of Laplacian noise. Through (45), NIMM can makes a proper decision that weather the residual signal is caused by noise or maneuvering.

Since our glint model uses more parameters, it will be more difficult to identify. Therefore, it is informative to study the sensitivity of NIMM to the parameter variation. Our study is based on the simulation analysis. In our experiments, we purposely set the wrong parameters and observe the performance variation. From (46), we know that in our model, three parameters are involved namely, ϵ, σ, and η. There are numerous ways to set the parameters. To simplify the experiments, we make a constrain for the parameters, i.e., the variance is a constant, i.e.,

$$\epsilon\sigma^2 + 2(1-\epsilon)\eta^2 = \text{constant}. \qquad (56)$$

The constrain is practically meaningful. Given noise records, the variance is the value most easily obtained. The other advantage is that in all cases, the performance of the standard IMM remains the same (since the variance is the same). This facilitates comparison. Here, we use the same parameters as those in previous experiments (55) to generate observations. Therefore, the variance is 4.1×10^4, which is the one used in IMM2. Two sets of experiments are conducted. In the first one we fix σ and vary ϵ and the other one we fix ϵ and vary σ. In both cases, due the constrain in (56), η is also varied. For each experiment, we perform four Monte Carlo simulations for four sets of parameters (each 500 runs). They are denoted as NIMM1,NIMM2,...,NIMM8. The parameters for experiments are listed in Table I. Since the performance in two axes is similar, we only show the results in x-axis. The average rms errors (from Step 10 to 90) for position and velocity estimate in x-axis are listed in Table II. In the table, the average rms errors of IMM1, IMM2, and NIMM are also included. The third and the fifth column of the table show the performance variation (PV) which are defined as follows.

$$\text{PV} = \frac{\text{rms}_{\text{NIMM}} - \text{rms}}{\text{rms}_{\text{NIMM}}} \qquad (57)$$

Negative PV indicates the performance degradation (compare to NIMM). From the table, we surprisingly find that the rms errors of NIMM3, NIMM4, NIMM5, and NIMM6 are smaller than that of NIMM. Does this violate filtering theory? We will discuss this later. The performances of NIMM1, NIMM2, NIMM7 and NIMM 8 are degraded. The worst case is NIMM1. The rms error increases about 20% for the position estimate and 30% for the velocity estimate. However, note that in this case the variations of η and ϵ are extremely large (from 0.1 to 0.01 and from 400 to 1247). Even in this worst case, the performance of NIMM is still much better than that of IMM2. The other three cases, the degradation is small. In terms of rms error, we conclude that NIMM is not sensitive to the parameter variation. Figs. 13–18 show the rms errors of position estimate, the rms errors of velocity estimate, and the probability estimates of model one for NIMM1, NIMM4, NIMM5, and NIMM8. We also plot the results of IMM2 and NIMM for the comparison purpose. If we look at the figures carefully, we find that the behavior of the rms errors of NIMMs are different in the unmaneuvering and the maneuvering period (Figs. 13–14 and 16–17). For IMM4-5, the errors are smaller than those of NIMM in the unmaneuvering period and bigger in the maneuvering period. Also, the amount of error increased in the maneuvering period is bigger than that decreased in the unmaneuvering period. However, since the maneuvering period is short, the average rms errors of NIMM4 and NIMM5 are smaller. This is also true for NIMM3 and NIMM6. In (54), we implicitly assume that the maneuvering and the unmaneuvering period have same length. For the particular trajectory we used, it seems that the assumption is not valid. This is why wrong parameters produce better results. If the switching of two models is based on (54) and the trajectory is sufficiently long, the performance of NIMM will be better than those of NIMM3-6. Contract to NIMM4 and NIMM5, the errors of NIMM1 and NIMM8 are bigger in the unmaneuvering

TABLE I
Parameters Used in NIMMs

	ϵ	σ	η		ϵ	σ	η
NIMM1	0.01	100	1247	NIMM5	0.10	140	341.8
NIMM2	0.05	100	561.3	NIMM6	0.10	120	374.4
NIMM3	0.20	100	287.2	NIMM7	0.10	80	419.8
NIMM4	0.30	100	238.1	NIMM8	0.10	60	434.5

TABLE II
RMS Errors of IMMs and NIMMs

	Position (m)	P. V.	Velocity (m/sec)	P. V.
IMM1	149.3	-91.71%	9.337	-188.2%
IMM2	127.3	-63.46%	5.466	-68.70%
NIMM	77.88	—	3.240	—
NIMM1	93.44	-19.98%	4.238	-30.80%
NIMM2	80.82	-3.780%	3.515	-8.490%
NIMM3	76.03	+2.380%	3.000	+7.410%
NIMM4	75.65	+2.860%	2.895	+10.65%
NIMM5	77.30	+0.740%	3.031	+6.450%
NIMM6	77.27	+0.780%	3.109	+4.040%
NIMM7	79.56	-2.160%	3.466	-6.980%
NIMM8	83.08	-6.680%	3.863	-19.23%

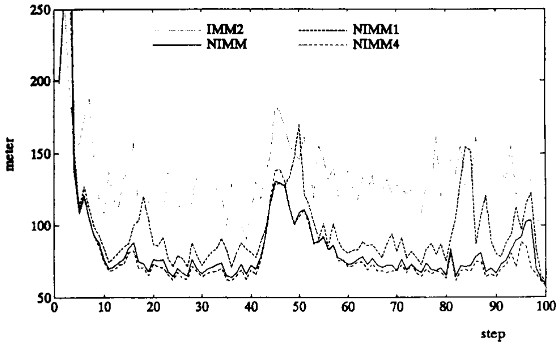


Fig. 13. RMS error of position estimate in x axis.

period and smaller in the maneuvering period (Figs. 13–14 and 16–17). The amount of error decreased in the maneuvering period is smaller than that increased in the unmaneuvering period. Thus, if the maneuvering condition described above is satisfied, the performance of NIMM1 and NIMM8 will be better than their present results, but still worse than that of NIMM. Again, this is also true for NIMM2 and NIMM7.

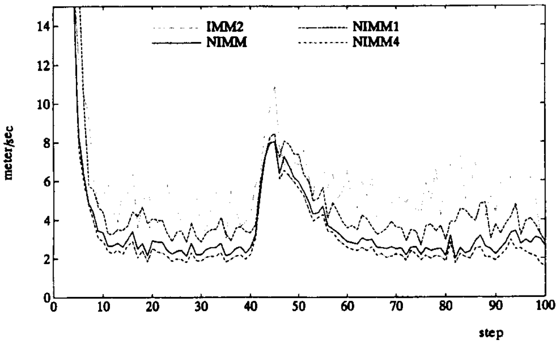


Fig. 14. RMS error of velocity estimate in x axis.

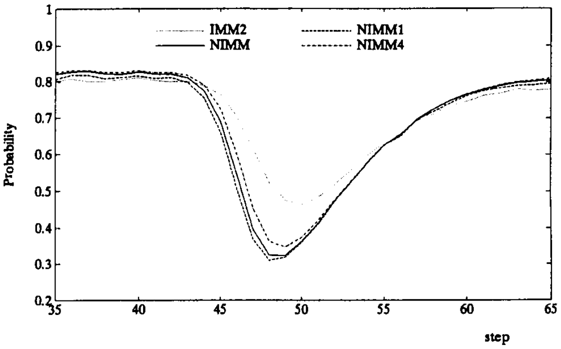


Fig. 15. Probability estimate of model 1 in x axis.

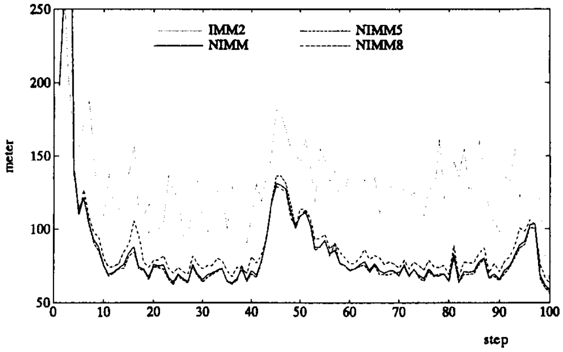


Fig. 16. RMS error of position estimate in x axis.

To understand why NIMM1-8 have different performance in the maneuvering and the unmaneuvering period, let's take a look the score functions of the noise distributions that are shown in Figs. 19–20. In Fig. 19, we see that the linear regions of the score functions for NIMM1 and NIMM2 are larger than that of NIMM while they are smaller for NIMM3 and NIMM4. In addition, the slopes of the score functions are all the same in the central region. This means that less Laplacian noise is filtered in NIMM1 and NIMM2 and that produces larger error in the unmaneuvering period. However, in the maneuvering period, due to the smaller ϵ, large residual signal tends to be seen as the cause of maneuvering. The capability of maneuvering tracking is then better. On the contrary, more Laplacian noise is assumed in NIMM3 and NIMM4. When maneuvering occurs, large residual signal tends to be seen as the cause of noise. This results in a poorer "catch up". This argument can be verified by seeing the probability estimate of model one as shown in Fig. 15. Fig. 20

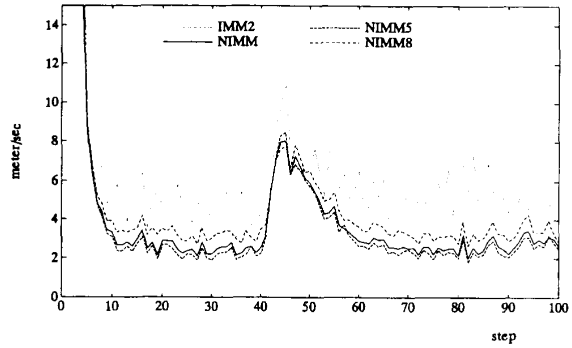


Fig. 17. RMS error of velocity estimate in x axis.

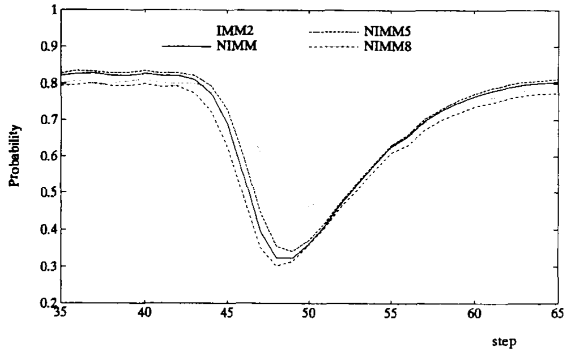


Fig. 18. Probability estimate of model 1 in x axis.

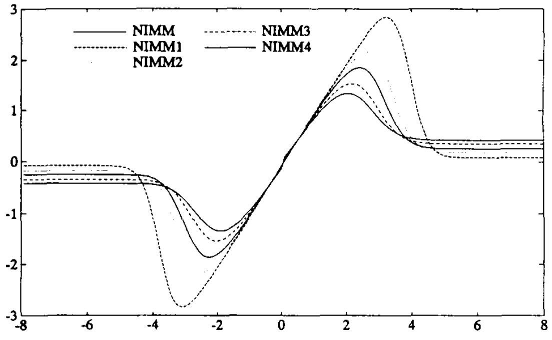


Fig. 19. Score functions of noise distributions.

shows the score functions of the noise distribution for NIMM5-8. Here, the slopes in the central region are all different. This means that gains are different in the region. The gains for NIMM5 and NIMM6 are smaller than those of NIMM while they are bigger for NIMM7 and NIMM8. As we discussed before, large gain is good for maneuvering detection but bad for noise filtering. Thus, NIMM7 and NIMM8 have better performance in the maneuvering period and worse performance in the unmaneuvering period. On the contrary, NIMM5 and NIMM6 have the opposite results. This also can be verified by seeing the probability estimate of model one as shown in Fig. 18.

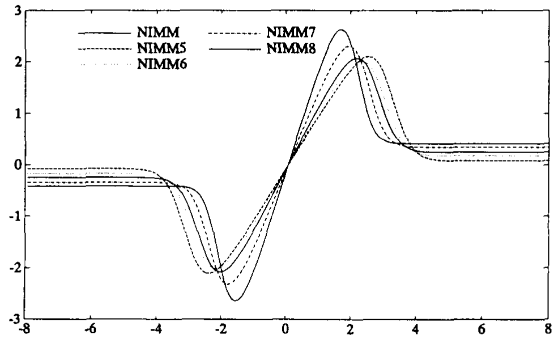


Fig. 20. Score functions of noise distributions.

V. CONCLUSIONS

Conventional tracking algorithms are all based on an idealized assumption, i.e., Gaussian observation noise. In real applications, however, the noise is not necessarily Gaussian. In those cases, the conventional algorithms are not sufficient. Due to the target glint, the observation noise that is called glint noise in a radar system, is non-Gaussian. Here, we extend the idea in [20] to develop a new maneuvering target tracking algorithm. The algorithm substitutes the Kalman filter in the IMM algorithm with the nonlinear Masreliez filter and modifies the estimate of model probabilities. Simulations show that our new algorithm not only can filter the glint noise efficiently, but also respond to maneuvering quickly. Conventional IMM algorithm performs poorly in the glint environment. To use our new algorithm, some parameters for the glint have to be known. We have shown that our new algorithm is not sensitive to the parameter variation. One set of parameters may be good for many situations. In fact, if the parameters are carefully chosen, the algorithm can perform in a "robust" way. The main computation cost of the Masreliez filter is to evaluate the score function. For a scalar case, this computation requirement is small. The practical implementation of our algorithm is then feasible.

REFERENCES

[1] Gholson, N. H., and Moose, R. L. (1977)
Maneuvering target tracking using adaptive state estimation.
IEEE Transactions on Aerospace and Electronic Systems, **AES-13** (May 1977), 310–317.

[2] Moose, R. L., et al. (1979)
Modeling and estimation for tracking maneuvering targets.
IEEE Transactions on Aerospace and Electronic Systems, **AES-15** (May 1979), 448–456.

[3] Singer, R. A. (1970)
Estimating optimal tracking filter performance for manned maneuvering targets.
IEEE Transactions on Aerospace and Electronic Systems, **AES-6** (July 1970), 473–483.

[4] Blom, H. A. P., and Bar-Shalom, Y. (1988)
The interactive multiple model algorithm for system with markovian switching coefficients.
IEEE Transactions on Automatic Control, **30** (Aug. 1988), 780–783.

[5] Bar-Shalom, Y., and Birmiwal, K. (1982)
Variable dimension filter for maneuvering target tracking.
IEEE Transactions on Aerospace and Electronic Systems, **AES-18** (Sept. 1982), 621–628.

[6] Chan, Y. T., Hu, A. G. C., and Plant, J. B. (1979)
A Kalman filter based tracking scheme with input estimation.
IEEE Transactions on Aerospace and Electronic Systems, **AES-15** (Mar. 1979), 237–244.

[7] Chan, Y. T., Plant, J. B., and Bottomley, J. R. (1982)
A Kalman tracker with a simple input estimation.
IEEE Transactions on Aerospace and Electronic Systems, **AES-18** (Jan. 1982), 235–240.

[8] Bogler, P. L. (1987)
Tracking a maneuvering target using input estimation.
IEEE Transactions on Aerospace and Electronic Systems, **AES-23** (May 1987), 298–310.

[9] Bar-Shalom, Y., Chang, K. C., and Blom, H. A. P. (1989)
Tracking a maneuvering target using input estimation versus interactive multiple model algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **25** (Mar. 1989), 296–300.

[10] Hewer, G. A., Martin, R. D., and Zeh, J. (1987)
Robust preprocessing for Kalman filtering of glint noise.
IEEE Transactions on Aerospace and Electronic Systems, **AES-23** (Jan. 1987), 120–128.

[11] Anderson, B. D. O., and Moore, J. B. (1979)
Optimal Filtering.
Englewood Cliffs, NJ: Prentice-Hall, 1979.

[12] Sorenson, H. W., and Stubberud, A. R. (1968)
Non-linear filtering by approximation of the *a posteriori* density.
International Journal of Control, **18** (1968), 33–51.

[13] Sorenson, H. W., and Alspach, D. L. (1971)
Recursive Bayesian estimation using Gaussian sums.
Automatica, **7** (1971), 465–479.

[14] Alspach, D. L., and Sorenson, H. W. (1972)
Nonlinear Bayesian estimation using Gaussian sum approximations.
IEEE Transactions on Automatic Control, **AC-17**, 4 (Aug. 1972), 465–478.

[15] Masreliez, C. J. (1972)
Robust recursive estimation and filtering.
Ph.D. dissertation, University of Washington, Seattle, 1972.

[16] Masreliez, C. J. (1975)
Approximate non-Gaussian filtering with linear state and observation relations.
IEEE Transactions on Automatic Control, **AC-20** (1975), 107–110.

[17] Masreliez, D. J., and Martin, R. D. (1977)
Robust Bayesian estimation for the linear model and robustifying the Kalman filter.
IEEE Transactions on Automatic Control, **AC-22** (1977), 361–371.

[18] Tollet, I. H. (1975)
Robust Forecasting.
Ph.D. dissertation, University of Washington, Seattle, 1975.

[19] Wu, W.-R., and Kundu, A. (1989)
Kalman filtering in non-Gaussian environment using efficient score function approximation.
In *Proceedings of 1989 IEEE international Symposium on Circuits and Systems*, 413–416.

[20] Wu, W.-R. (1993)
Target tracking with glint noise.
IEEE Transactions on Aerospace and Electronic Systems, **29** (1993), 1–12.

[21] Daniels, H. E. (1954)
Saddle point approximations in statistics.
Annals of Mathematical Statistics, **25** (1954), 631–650.

[22] Barndorff-Nielsen, O., and Cox, D. R. (1979)
Edgeworth and saddle point approximations with statistical applications.
J. R. Statist. Soc. B, **41** (1979), 279–312.

[23] Hampel, F. R. (1973)
Some small sample asymptotics.
In J. Hajek (Ed.), *Proceedings of the Prague Symposium on Asymptotic Statistics*, Vol. II, Prague, 1973, 109–126.

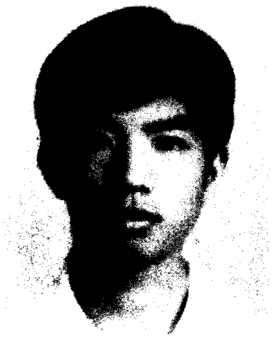
[24] Khinchin, A. I. (1949)
Mathematic Foundation of Statistical Mechanics.
New York: Dover, 1949.

[25] Cramer, H. (1946)
Mathematical methods of Statistics.
Princeton, NJ: Princeton University Press, 1946.

[26] Borden, B. H., and Mumford, M. L. (1983)
A statistical glint/radar cross section target model.
IEEE Transactions on Aerospace and Electronic Systems, **AES-19** (Sept. 1983), 781–785.

Wen-Rong Wu was born in Taiwan, R.O.C., in 1958. He received his B.S. degree in mechanical engineering from Tatung Institute of Technology, Taiwan, in 1980, M.S. degrees in mechanical and electrical engineering, and Ph.D. degree in electrical engineering from State University of New York at Buffalo in 1985, 1986, and 1989, respectively.

Since August 1988, he has been a faculty member in the Department of Communication Engineering in National Chiao Tung University, Taiwan. His research interests include estimation theory, digital signal processing, and image processing.



Peen-Pau Cheng was born in Taiwan, R.O.C., in 1968. He received his M.S. degree in control engineering from National Chiao Tung University, Taiwan, in 1992.

His research interests include control theory and radar signal processing.