

Adaptive Filtering Algorithms in Target Tracking Applications

Igor Jovandić, Željko Djurović, and Branko Kovačević

Abstract: Comparison of several target tracking algorithms is presented. Namely, discrete noise level adjustment (DNLA), variable state dimension (VSD) and interacting multiple model (IMM) algorithms are discussed. Target trajectory, target models, filtering algorithms and simulation results are given. The cumulative estimation error criterion is used in order to compare the algorithms.

Keywords: Signal processing, adaptive filter, Kalman filter, target tracking, discrete noise level

1 Introduction

Accurate estimation of target position and velocity is the problem of significant importance in a multiple target tracking (MTT) system. Since target dynamics is subject to various changes, solutions based on only one filter with no adaptation at all rarely give satisfactory results [1,2,3]. A number of adaptive algorithms utilizing Kalman filter approach have been proposed [1],[2], three of which are to be discussed here. Namely, discrete noise level adjustment (DNLA), variable state dimension (VSD) and interacting multiple model (IMM) algorithms are presented and compared. The DNLA algorithm uses only one Kalman filter with adjustable process noise variance, while VSD and IMM algorithms require multiple filters. The VSD algorithm

Manuscript received June 22, 2003. An earlier version of this paper was presented at the 10th Telecommunications Forum TELFOR 2002, November 26-28, 2002, Belgrade, Serbia.

The authors are with the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, Serbia and Montenegro (e-mail: ijovandic@etf.bg.ac.yu, kovacevicb@kiklop.etf.bg.ac.yu).

uses two filters. Only one of these two is used for estimation purposes at a given time, while switching between two filters is done based on quality of the estimate that filter currently in use produces. The IMM algorithm uses both filters simultaneously, combining estimates produced by each of them.

Problem discussed is estimation of position and velocity of a single target in planar motion, based on a sequence of radar measurements. It is assumed that only position measurements are available.

2 Target Models

Target models used here are simple constant velocity (CV) and constant acceleration (CA) models [1]. Discrete state equation (1) describes these models in (ξ, η) plane

$$x_{k+1} = Fx_k + Gw_k, \quad (1)$$

where state vectors for constant velocity and constant acceleration model are given as $x_k = [\xi_k \dot{\xi}_k \eta_k \dot{\eta}_k]$ and $x_k = [\xi_k \dot{\xi}_k \ddot{\xi}_k \eta_k \dot{\eta}_k \ddot{\eta}_k]$, respectively. Process noise vector is given as $w_k = [w_k^\xi w_k^\eta]$. Process noise components are Gaussian zero-mean white noise sequences and are assumed to be independent. In the case of CV (CA) model, they represent velocity (acceleration) increments during k -th sampling period in ξ and η coordinates. Variances of both process noise components are assumed to be equal and are marked as σ_k^2 . Therefore process noise covariance matrix can be written as $Q_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sigma_k^2$. Matrices F and G for constant velocity model are given by equation (2)

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad G = \begin{bmatrix} \frac{T}{2} & 0 \\ 1 & 0 \\ 0 & \frac{T}{2} \\ 0 & 1 \end{bmatrix}. \quad (2)$$

Matrices F and G for constant acceleration model are given by equation (3)

$$F = \begin{bmatrix} 1 & T & \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad G = \begin{bmatrix} \frac{T^2}{6} & 0 \\ \frac{T}{2} & 0 \\ 0 & \frac{T^2}{6} & 0 \\ 1 & 0 \\ 0 & \frac{T}{2} \\ 0 & 1 \end{bmatrix}. \quad (3)$$

Since only position measurements are available, discrete measurement equation is

$$z_k = Hx_k + v_k, \quad (4)$$

where

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 01 & 0 & 0 \end{bmatrix}$$

for constant velocity model and

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (5)$$

for constant acceleration model. Therefore dimension of measurement vector z_k is $n_z = 2$. Measurement noise vector consists of two components and is given as: $v_k = [v_k^\xi \ v_k^\eta]^T$.

Measurement noise components are Gaussian zero-mean white noise sequences and are assumed to be independent. Variances of both measurement noise components are assumed to be equal and constant, and are marked as σ_v^2 . Therefore measurement noise covariance matrix can be written as $R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sigma_v^2$.

3 Adaptive Filtering Algorithms

An overview of algorithms discussed is presented.

3.1 DNLA

The DNLA algorithm is based on a single, constant velocity model [1]. Kalman filter corresponding to this model is initialized with some low-level process noise variance, marked as σ_L^2 . Residual (innovation) sequence is constantly monitored. If 'large' innovation is detected it is assumed that target is maneuvering, so Kalman filter is reinitialized with new, larger process noise variance (marked as σ_H^2) in order to 'cover' the maneuver. Detection of large innovation can be performed in several ways; normalized innovation squared (NIS) approach is applied here. Normalized innovation squared is defined as

$$e_k = r_k^T S_k^{-1} r_k, \quad (6)$$

where r_k is innovation and S_k is innovation covariance matrix, both in $t = kT$. NIS is chi-square distributed, with n_z degrees of freedom (n_z is

dimension of measurement vector; here $n_z = 2$). A threshold e_{TR} can be defined based on the chosen probability that maneuver takes place when $e_k < e_{TR}$. That probability should be low, say 0.01. Process noise switching from low to high value is done when $e_k > e_{TR}$ is detected, and return from high to low process noise value when $e_k < e_{TR}$.

3.2 VSD

The VSD algorithm utilizes both constant velocity and constant acceleration model [1],[4]. Nominal (nonmaneuvering) model is the CV model. Fading memory average (FMA) of nominal model innovations is constantly monitored in order to detect a maneuver. If FMA exceeds certain threshold it is declared that maneuver has started. When choosing a value of FMA threshold one has to have in mind the definition of fading memory average given by equation (7)

$$e_k^a = ae_{k-1}^a + e_k, \quad 0 < a < 1 \quad (7)$$

where a is fading parameter and e_k is normalized innovation squared as defined in (6). FMA can be regarded as chi-square distributed with sn_z degrees of freedom, where $s = 1/(1 - a)$ is effective window length. FMA threshold value e_{TR}^a then corresponds to chosen probability that maneuver has taken place when $e_k < e_{TR}^a$. That probability should be low, say 0.01.

When maneuver is detected in $t = kT$ one should initialize Kalman filter based on CA (maneuvering) model, so future measurements (z_{k+1}, z_{k+2}, \dots) could be processed with it. That is, state estimate for maneuvering model \hat{x}_k^m and its covariance matrix P_k^m are needed. It is assumed that maneuver has started s periods before detection, so values \hat{x}_{k-s}^m and P_{k-s}^m are calculated based on nonmaneuvering state estimate \hat{x}_{k-s-1} , its covariance matrix P_{k-s-1} and measurement $z - k - s$. Relations defining these calculations (in ξ coordinate only) are given by following equations

$$\begin{aligned} \hat{\xi}_{k-s}^m &= z_{k-s}^\xi, \\ \hat{\xi}_{k-s}^m &= \frac{2}{T} [z_{k-s}^\xi - \hat{\xi}_{k-s-1} - T\hat{\xi}_{k-s-1}^m], \\ \hat{\xi}_{k-s}^m &= \hat{\xi}_{k-s-1} + T\hat{\xi}_{k-s}^m, \\ P_{11}^m &= R_{11}, \\ P_{22}^m &= \frac{4}{T^2}R_{11} + \frac{4}{T^2}P_{11} + \frac{4}{T}P_{12} + P_{22}, \\ P_{33}^m &= \frac{4}{T^4}(R_{11} + P_{11} + @P_{12}T + P_{22}T^2), \end{aligned}$$

$$\begin{aligned}
P_{12}^m &= \frac{2}{T} R_{11}, \\
P_{13}^m &= \frac{2}{T^2} R_{11}, \\
P_{23}^m &= \frac{4}{T^3} R_{11} + \frac{4}{T^3} P_{11} + \frac{6}{T^2} P_{12} + \frac{2}{T} P_{22},
\end{aligned}$$

where $P_{11}^M, P_{22}^M, P_{33}^M, P_{12}^M, P_{12}^M$ and P_{23}^M are elements of P_{k-s}^M corresponding to ξ coordinate, P_{11}, P_{12} and P_{22} are elements of P_{k-s-1} and R_{11} and R_{22} are elements of $R = \text{cov}\{v\}$.

Measurements z_{k-s+1}, \dots, z_k are reprocessed with filter based on CA model, so values \hat{x}_k^m and P_k^m are obtained. Measurements arriving after $t = kT$ are processed with filter based on CA model.

Moving sum of normalized estimates of acceleration, given by (8), is monitored in order to detect maneuver termination

$$MS_k = \sum_{j=k-p+1}^k \delta_k^a, \quad (8)$$

where $\delta_k^a = \begin{bmatrix} \hat{\xi} \\ \hat{\eta}_k^m \end{bmatrix} \begin{bmatrix} P_{33}^m(k) & 0 \\ 0 & P_{66}^m(k) \end{bmatrix}^{-1} \begin{bmatrix} \hat{\xi}_k^m \\ \hat{\eta}_k^m \end{bmatrix}$ and p is sliding window length.

Moving sum (MS) is chi-square distributed with pn_z degrees of freedom, so threshold for MS can be set so probability that maneuver has ended when $MS_k > MS_{TR}$ is low. If MS falls below this threshold it is assumed that maneuver is terminated and measurements to come should be processed with filter based on CV model. Initialization of CV filter is done by simple truncation of acceleration states in state vector and corresponding elements in state covariance matrix.

3.3 IMM

One cycle of the IMM algorithm is depicted in Figure 1. Short description of algorithm will be given [1]-[5]. Two Kalman filters (filter M1 set according to CV model and filter M2 set according to CA model) operate simultaneously producing state estimates \hat{x}^1 and \hat{x}^2 with corresponding covariance matrices P^1 and P^2 . In addition, both filters produce their likelihood function values defined as

$$\Lambda(k) = \frac{1}{\sqrt{(2\pi)^n |S_k|}} e^{-\frac{1}{2} r_k^T S_k^{-1} r_k},$$

where r_k is innovation and S_k is innovation covariance matrix, both in $t = kT$. Estimates \hat{x}^1 and \hat{x}^2 , together with corresponding covariance ma-

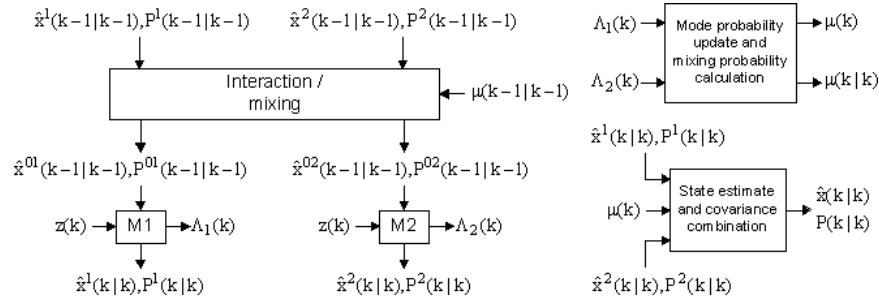


Fig. 1. The IMM algorithm.

trices P^1 and P^2 (all in $t = (k-1)T$, that is resulting from previous cycle) are mixed to form initial conditions \hat{x}^{01} and \hat{x}^{02} and corresponding covariance matrices P^{01} and P^{02} for processing to come. Variables marked as $\mu_{i|j}$ are called mixing probabilities. Results of processing (state estimates and corresponding covariance matrices) are combined to form final (output) estimate. Combining is done according to mode probabilities μ_j . Algorithm is adaptive since mode probabilities μ_1 and μ_2 are changed based on quality of estimates produced; here, likelihood functions are taken as measures of estimate quality. The IMM algorithm assumes that model jump process is a Markov process with known transition probabilities

$$p_{ij} = P\{M(k) = M_j | M(k-1) = m_i\}, \quad i, j \in \{1, 2\}$$

where $M(k)$ is model (mode) at time $t = kT$. These mode transition probabilities are assumed to be time invariant and independent of the base state. Since only two models are used, transition probability matrix P_{TRAN} is given by (9)

$$P_{TRAN} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}. \quad (9)$$

One cycle of the algorithm consists of the following five steps (number of filters is marked as r):

1. **Calculation of the mixing probabilities** ($i, j = 1, \dots, r$):

$$\mu_{i|j}(k-1|k-1) = \frac{1}{\bar{c}_j} p_{ij} \mu_i(k-1),$$

where normalizing constants are $\bar{c}_j = \sum_{i=1}^r p_{ij} \mu_i(k-1)$

2. **Mixing** ($j = 1, \dots, r$):

$$\begin{aligned}\hat{x}^{0j}(k-1|k-1) &= \sum_{i=1}^r \hat{x}(k-1|k-1)\mu_{i|j}(k-1|k-1), \\ P^{0j}(k-1|k-1) &= \sum_{i=1}^r \mu_{i|j}(k-1|k-1)\{P^i(k-1|k-1) \\ &\quad + err_{i_j}^0(k-1|k-1)err_{i_j}^0(k-1|k-1)^T\},\end{aligned}$$

where $err_{i_j}^0(k-1|k-1) = \hat{x}^j(k-1|k-1) - \hat{x}^{0j}(k-1|k-1)$.

3. **Mode matched filtering** ($j = 1, \dots, r$): starting with $\hat{x}^{0j}(k-1|k-1)$, $P^{0j}(k-1|k-1)$ and $z(k)$, Kalman filter matched to model m_j calculates the estimate $\hat{x}^j(x|x)$, its covariance matrix $P^j(k|k)$ and likelihood function $\lambda_j(k)$.

4. **Mode probability update** ($j = 1, \dots, r$):

$$\mu_j(k) = \frac{1}{c} \Lambda_j(k) \bar{c}_j,$$

where $c = \sum_{j=1}^r \Lambda_j(k) \hat{c}_j$.

5. **Estimate and covariance combination**: this step is used for output purposes only; it is not a part of algorithm recursions

$$\begin{aligned}\hat{x}(k|k) &= \sum_{j=1}^r \hat{x}^j(k|k)\mu_j(k), \\ P(k|k) &= \sum_{j=1}^r \mu_j(k)\{P^j(k|k) + err_j(k|k)err_j(k|k)^T\},\end{aligned}$$

where $err_j(k|k) = \hat{x}^j(k|k) - \hat{x}(k|k)$.

4 Experimental results

Algorithms described are compared using computer simulation. Target trajectory in (ξ, η) plane is defined. Initial target state is described by following equations:

$$\begin{aligned}\xi_0 &= -10 \times 10^3 \text{ m}, & \eta_0 &= -15 \times 10^3 \text{ m}, \\ \dot{\xi}_0 &= 1 \text{ m/s}, & \dot{\eta}_0 &= 120 \text{ m/s}, \\ \ddot{\xi}_0 &= 0 \text{ m/s}^2 & \ddot{\eta}_0 &= 0 \text{ m/s}^2\end{aligned}$$

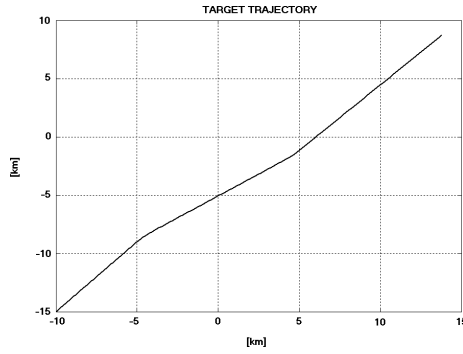


Fig. 2. Target trajectory.

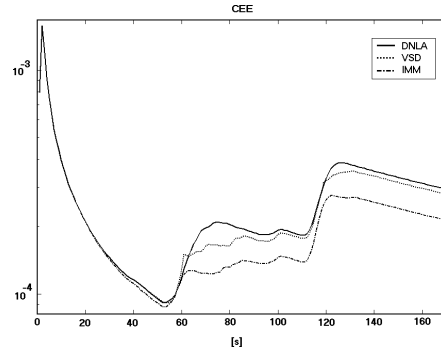


Fig. 3. CEE plots.

During first 50 scans target exhibits constant velocity motion. First maneuver (constant acceleration motion) starts in 51st scan and lasts 10 scans. Constant velocity motion follows and lasts 50 scans. Second maneuver is of the same type as the first one, starts in 111th scan and lasts only 5 scans. During the rest of 55 scans target has constant velocity motion. Accelerations in m/s^2 during first and second maneuver are $[a_\xi \ a_\eta] = [6 \ 0]$ and $[a_\xi \ a_\eta] = [0 \ 12]$, respectively. Sampling period is $T = 1 \text{ s}$. Target trajectory is shown in Figure 2.

Measurement noise standard deviation is in both coordinates. Same random numbers (representing measurement noise samples) are used for the simulation of all three algorithms.

Parameters used in each algorithm are given. Initial state estimate is chosen based on position measurements in all three cases (initial velocity and acceleration estimates are zero). NIS threshold in the DNLA algorithm is $e_{TR} = 5$, while low and high process noise variances are $\sigma_L^2 = 10^{-3}$ and $\sigma_H^2 = 20$, respectively. Parameters used in the VSD algorithm are $a = 0.89$, $p = 2$, $e_{TR}^a = 30$ and $MS_{TR} = 0.1$. Transition probability matrix in the IMM algorithm is

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix},$$

while initial mode probabilities are $\mu_1 = 0.9$ and $\mu_2 = 0.1$; process noise variances are $\sigma_{CV}^2 = \sigma_{CA}^2 = 0.1$. Criterion for comparison of algorithms is the value of cumulative estimation error (CEE) defined as follows

$$CEE(k) = \frac{1}{k} \sum_{i=1}^k \frac{(x_k - \hat{x}_k)^T W (x_k - \hat{x}_k)}{x_k^T W x_k} \quad (10)$$

where W is diagonal weighting matrix whose values are chosen in order

to equalize the influence of position and velocity estimation errors; here $W = \text{diag}[1 \ 10 \ 1 \ 10]$.

Figure 3 gives CEE plots for each of described algorithms; logarithmic scale is used.

5 Conclusion

By inspecting CEE plots one concludes that the IMM algorithm is superior compared to the other two, both during maneuver and in steady state. The VSD algorithm is slightly better than the DNLA algorithm in a view of overall performance. All algorithms have some maneuver detection delay (a few scans only). The estimation error during maneuver is largest in the DNLA case; this result shows that an algorithm utilizing one filter only hardly covers the maneuver in spite of adaptation. Nevertheless, rather good results were obtained in the case of the second (fast and short) maneuver compared to results of the VSD algorithm in the same case. That reveals a drawback of the VSD algorithm: the mere necessity to initialize the maneuvering filter. The IMM algorithm does not encounter this problem since both filters operate simultaneously. However, one should account for the price of such estimation quality. The price is here expressed in terms of computational complexity of an algorithm. The subject of computational complexity is not discussed here, but one could conclude (based on description of algorithms) that the DNLA algorithm is the simplest one.

One should note that the IMM algorithm is not limited on two models only; equations given describe algorithm in general case where r models are used.

References

- [1] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking Principles Techniques and Software*. Boston MA: Artech House, 1993.
- [2] S. Blackman and R. Popoli, *Design and Analysis of Modern, Tracking Systems*. Boston MA: Artech House, 1999.
- [3] B. K. cević and Z. Djurović, *Fundamentals of Stochastic Signals, Systems and Estimation Theory with Worked Examples*. Beograd: Akademska misao, 1999.
- [4] Y. Bar-Shalom and K. Birniwal, "Variable dimension filter for maneuvering target tracking," *IEEE Trans. on Aerospace and Electronic Systems*, vol. AES-18, pp. 621–629, September 1982.

- [5] X. Li and Y. Bar-Shalom, "Design of interacting multiple model algorithm for air traffic control tracking," *IEEE Trans. Cont. Sys. Tech.*, vol. 1, September 1993.