ELSEVIER

Information Fusion 3 (2002) 191–201

INFORMATION FUSION

www.elsevier.com/locate/inffus

Some remarks on Kalman filters for the multisensor fusion

J.B. Gao [a,*,1], C.J. Harris [b]

[a] School of Mathematics, Statistics and Computer Sciences, University of New England, Armidale, NSW 2351, Australia
[b] Image, Speech and Intelligent System Research Group, Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

Received 21 March 2001; received in revised form 13 December 2001; accepted 26 March 2002

Abstract

Multisensor data fusion has found widespread application in industry and commerce. The purpose of data fusion is to produce an improved model or estimate of a system from a set of independent data sources. There are various multisensor data fusion approaches, of which Kalman filtering is one of the most significant. Methods for Kalman filter based data fusion includes measurement fusion and state fusion. This paper gives first a simple a review of both measurement fusion and state fusion, and secondly proposes two new methods of state fusion based on fusion procedures at the prediction and update level, respectively, of the Kalman filter. The theoretical derivation for these algorithms are derived. To illustrate their application, a simple example is performed to evaluate the proposed methods and compare their performance with the conventional state fusion method and measurement fusion methods.

1. Introduction

Multisensor data fusion has found widespread application in diverse areas ranging from local robot guidance to global military theatre defense etc. (see [10]). The use of sensory data from a range of disparate, multiple sensors is to automatically extract the maximum amount of information possible about the sensed environment under all operating conditions. Increased performance, reliability, data rates, and autonomy, coupled with increased complexity, diverse uncertain operating environments, requires the automated intelligent combination of data from multiple sensors to derive less ambiguous/uncertain information about the desired state. In recent years there has been increasing awareness that a variety of sensors/platforms owned and operated by different agencies can be fruitfully integrated for better intelligence gathering, situation awareness, tactical missile defense, etc. For this purpose, efficient algorithms for data fusion and track-to-track association must be derived so that existing systems can be easily upgraded without imposing undue burdens on system operators, using existing hardware and software.

The purpose of data fusion is to produce a model or representation of a system from a set of independent data sources, from which a single view or perception of some external environment or system is found or detected; therefore data fusion is the continuous process of assembling a model of the domain of the interest utilizing data from disparate sources. For many application involving real time, the desired domain model is the state vector of a dynamical process, such as an observed airborne vehicle. The combination of the information from the sensors and subsequent estimation of the state of the environment should be done in a consistent, coherent manner such that the uncertainty is reduced. The algorithms must be able to deal with multiple observations, multiple sensors and multiple targets.

A common application of data fusion techniques is the estimation of target position or kinematic information from multiple measurements from a single or multiple sensors. Two essential processes are involved in the derivation of position or kinematic information: data association and state estimation. State estimation

* Corresponding author.
E-mail addresses: jbgao@mcs.une.edu.au (J.B. Gao), cjh@ecs.soton.ac.uk (C.J. Harris).
[1] This author is partially supported by the Faculty Research Grant of University of New England and Natural Science Foundation of China (Grant no.: 19871032).

1566-2535/02/$ - see front matter
PII: S1566-2535(02)00070-2

refers to the optimal estimation of those values, e.g., the position, velocity, acceleration, angular position etc of the target, from observation data.

The Kalman filter is the best known and most widely applied state estimator algorithm [7]. The Kalman filter gives a linear, unbiased, and minimum error variance recursive algorithm to optimally estimate the unknown state of a linear dynamic system from Gaussian distributed noisy observations. The Kalman filtering process can be considered as a prediction-update formulation. The algorithm uses a predefined (linear) model of the system to predict the state at the next time step. Added to this is a component to update for errors in the model using the actual observations of the system. The prediction and update are combined using the Kalman gain which is calculated to minimize the mean-square error of the state estimate.

The Kalman filter has found widespread application in data fusion problems and track fusion problems, see [4], in particular Manyika and Durrant-Whyte [13] have applied it extensively to robot localisation, guidance and navigation. Other areas of application include target detection, multisensor, multi-target tracking, automatic target recognition, collision avoidance, etc. (see [10,12, 14,16,17]).

The conventional state-vector fusion and measurement fusion are two kinds of methods for Kalman filter based data fusion, and the conventional measurement fusion [8,10] has lower estimation error but a higher computational cost. There is also another optimal fusion method based on information fusion [5,11]. This fusion method is optimal in the sense that sensors communicate each time they receive measurements or the process noise is zero, however it needs more computational costs (see the Eqs. (12) and (13) in [5]). The objective of this paper is to derive new fusion models at the state vector or measurement vector level. We propose two new state-vector fusion algorithms with respect to the prediction and update stages in the Kalman filtering procedure. The fusion models will use local fused information to give a new, possibly better, state estimate at each step. The computational cost for the first new algorithm is comparable to the conventional track-to-track fusion method, but has better performance for dissimilar sensors system.

In this paper we restrict attention to the linear state process and measurement process defined by (2.1) and (2.2) for simplicity. These linear models may have been derived from a phenomenological model or a data based linearisation procedure of an observable nonlinear process, see [8]. In Section 2 some conventional methods for state-vector fusion and measurement fusion are reviewed and two new fusion models are introduced. Section 3 provides an illustrative evaluation example for comparison with the conventional track-to-track fusion method.

2. Fusion models

For simplicity, assume that the sensors' sample rates are identical and the dynamics of the target is represented by

$$x_{k+1} = F_k x_k + \Gamma_k v_k \tag{2.1}$$

where x_k is the state vector at time k, and the state noise v_k such that

$$E[v_k] = 0; \quad E[v_k v_l^{\mathrm{T}}] = Q_k \delta_{kl}$$

The measurements corresponding to the two sensors are

$$z_k^m = H_k^m x_k + w_k^m, \quad m = 1, 2 \tag{2.2}$$

where z_k^m is the measurement of the sensor m at time k and the measurement noise sequences w_k^m are zero-mean, white, with covariance R_k^m, and mutually independent, i.e.,

$$\begin{aligned} &E[w_k^m] = 0; \quad E[w_k^m w_l^{m\mathrm{T}}] = R_k^m \delta_{kl} \\ &E[w_k^1 w_l^{2\mathrm{T}}] = E[w_k^2 w_l^{1\mathrm{T}}] = 0 \end{aligned} \tag{2.3}$$

Fusion of these tracks can now take place at either the state vector or measurement vector level.

2.1. Measurement fusion models

There are essentially two methods for measurement fusion. The first simply merges the measurements into an argumented observation vector and the second combines the measurements by using minimum mean-square estimates [10].

In the first measurement fusion technique, the measurement vectors z_k^1 and z_k^2 from the two (or more) sensors are merged into a new augmented measurement vector given by

$$z_k = [(z_k^1)^{\mathrm{T}} (z_k^2)^{\mathrm{T}}]^{\mathrm{T}} \tag{2.4}$$

Denote $H_k = [(H_k^1)^{\mathrm{T}} (H_k^2)^{\mathrm{T}}]^{\mathrm{T}}$ and $w_k = [(w_k^1)^{\mathrm{T}} (w_k^2)^{\mathrm{T}}]^{\mathrm{T}}$, then from the Eqs. (2.2) and (2.4) a new measurement equation is given by

$$z_k = H_k x_k + w_k \tag{2.5}$$

Based on the assumed statistical independence of the two sensors the covariance matrix R_k for the merged measurement noise w_k is defined as

$$R_k = \begin{pmatrix} R_k^1 & 0 \\ 0 & R_k^2 \end{pmatrix} \tag{2.6}$$

Then the estimate, $x_{k|k}$, of the state vector can be determined by the conventional Kalman filter via Eqs. (2.1) and (2.4).

Another approach to measurement fusion is to weight the individual measurements from each sensor and then track those fused measurements by a Kalman filter to obtain an estimate of the state vector [18]. Since the measurement noise is independent for sensors 1 and 2,

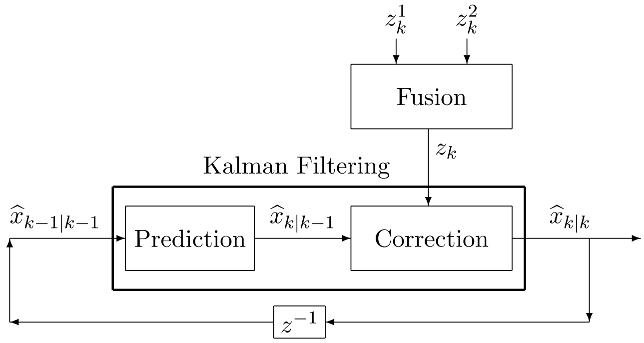

Fig. 1. The measurement fusion process.

the equation for fusing the measurement vectors z_k^1 and z_k^2, in recursive form, for a minimum mean square estimate is given by

$$z_k = z_k^1 + R_k^1(R_k^1 + R_k^2)^{-1}(z_k^2 - z_k^1) \tag{2.7}$$

where R_k^m is the covariance matrix of the measurement vector of the sensor m defined as (2.3). Then the covariance matrix of this fused measurement z_k can be derived by

$$R_k = [(R_k^1)^{-1} + (R_k^2)^{-1}]^{-1} \tag{2.8}$$

These fused measurements can then be tracked to obtain the estimate of the state vector $x_{k|k}$ via the conventional Kalman filter.

These algorithms are mathematically equivalent to the Kalman filter with a fused sensor, i.e, with a new fused measurement equation. An interesting fact is that the above two measurement fusion methods are functionally equivalent to each other when the measurement matrices satisfy the condition $H_k^1 = H_k^2$, see [9], and is independent of the state noise characteristic Q_k.

The measurement fusion procedure is illustrated in Fig. 1.

2.2. The track-to-track fusion models

In many practical situations targets are tracked by a variety of sensors. The decision process involved in associating tracks belonging to the same target is a correlation problem which has been previously examined [2]. Once the targets are correlated an algorithm is needed to provided a single target track which has less uncertainty than those of the individual tracks. This process is often referred to as track fusion (which also provides data compression).

One of the frequently used track fusion methods is the so-called track-to-track fusion algorithm which was first proposed by Bar-Shalom et al. [1,2]. Fig. 2 shows the procedure of this track-to-track fusion process. In this method, track fusion is performed by fusing the state estimates $\hat{x}_{k|k}^1$ and $\hat{x}_{k|k}^2$ from sensors 1 and 2 respectively into a new estimate of the state vector. The

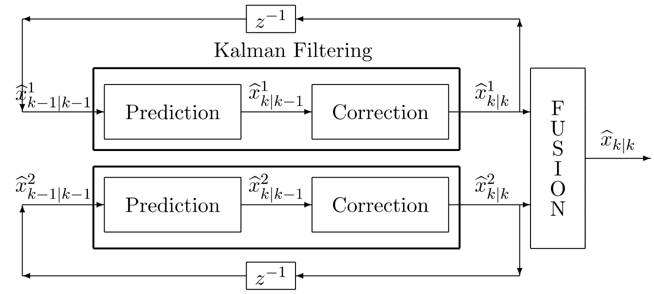

Fig. 2. The track-to-track fusion.

new fused estimate of the state vector $\hat{x}_{k|k}$ is generated from the static linear estimation equation

$$\hat{x}_{k|k} = \hat{x}_{k|k}^1 + \left[P_{k|k}^1 - P_{k|k}^{12}\right]\left[P_{k|k}^1 + P_{k|k}^2 - P_{k|k}^{12} - P_{k|k}^{21}\right]^{-1} \times \left(\hat{x}_{k|k}^2 - \hat{x}_{k|k}^1\right) \tag{2.9}$$

where $P_{k|k}^m$ is the covariance matrix for the tracked estimate $\hat{x}_{k|k}^m$ based on the measurement of sensor m ($m = 1, 2$), and $P_{k|k}^{12} = (P_{k|k}^{21})^{\mathrm{T}}$ is the cross covariance matrix between $\hat{x}_{k|k}^1$ and $\hat{x}_{k|k}^2$. The cross covariance matrix is given by the following recursive equation

$$P_{k|k}^{12} = (I - K_k^1 H_k^1)F_{k-1}P_{k-1|k-1}^{12}F_{k-1}^{\mathrm{T}}(I - K_k^2 H_k^2)^{\mathrm{T}} + (I - K_k^1 H_k^1)\Gamma_{k-1}Q_{k-1}\Gamma_{k-1}^{\mathrm{T}}(I - K_k^2 H_k^2)^{\mathrm{T}}$$

where K_k^m is the Kalman filter gain matrix for sensor m at time k.

This method of combining tracks is in general suboptimal due to the fusion Eq. (2.9) being the optimal solution of the linear estimator, [2]. The advantage of this track-to-track state fusion algorithm is a reduced computational load on the central processor. In [15], this track-to-track state fusion method was compared with the above second measurement fusion method (see Section 2.1) and an example was employed to illustrate reduction in the uncertainty achieved by the measurement fusion over the state vector fusion method.

However, since the fused state estimate $\hat{x}_{k|k}$ in (2.9) is better by definition than either of $\hat{x}_{k|k}^1$, $\hat{x}_{k|k}^2$, instead of feeding track these individual and local estimates to the predictive stage of the Kalman Filter, it is proposed here that the final fused estimate is fedback (see Fig. 3) to a single state predictor, (Eq. (2.10)), whose output is then fed to the two local observation correction equations (see Eq. (2.11)). Here a modified track-to-track fusion (MTF) method is derived in which the prediction procedure of Kalman filtering will be improved with the fused state estimate at the last time step $k-1$. This procedure is illustrated by Fig. 3.

Let $\hat{x}_{k-1|k-1}$ be the fused state estimate at time $k-1$, and take the prediction $\hat{x}_{k|k-1}$ based on $\hat{x}_{k-1|k-1}$ as

$$\hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} \tag{2.10}$$

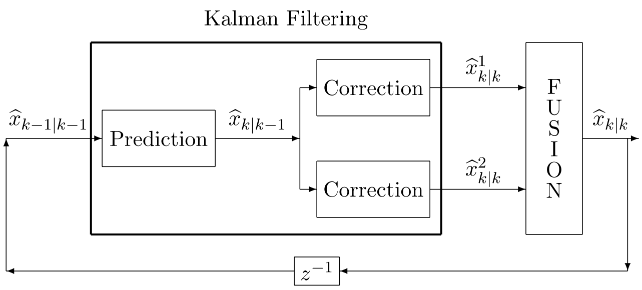

Fig. 3. The modified track-to-track fusion algorithm.

The prediction $\hat{x}_{k|k-1}$ is then combined with the measurements z_k^1 and z_k^2, respectively, to update the prediction for obtaining new estimates $\hat{x}^1_{k|k}$ and $\hat{x}^2_{k|k}$ at the next iteration with local Kalman filtering. The update equations are:

$$\begin{aligned}\bar{z}^m_{k|k-1} &= H^m_k \hat{x}_{k|k-1}\\ \hat{x}^m_{k|k} &= \hat{x}_{k|k-1} + P^m_{k|k-1}(x,z)P^m_{k|k-1}(z,z)^{-1}[z^m_k - \bar{z}^m_{k|k-1}]\end{aligned} \tag{2.11}$$

where $P^m_{k|k-1}(x,z) = E[(x_k - \hat{x}_{k|k-1})(z^m_k - \bar{z}^m_{k|k-1})^{\mathrm{T}}]$ and $P^m_{k|k-1}(z,z) = E[(z^m_k - \bar{z}^m_{k|k-1})(z^m_k - \bar{z}^m_{k|k-1})^{\mathrm{T}}]$. Then the new fused estimates $\hat{x}_{k|k}$ is given by the fusion Eq. (2.9) with $\hat{x}^1_{k|k}$ and $\hat{x}^2_{k|k}$. And the covariance matrix of the fused estimate $\hat{x}_{k|k}$ is then given by

$$P_{k|k} = P^1_{k|k} - [P^1_{k|k} - P^{12}_{k|k}][P^1_{k|k} + P^2_{k|k} - P^{12}_{k|k} - P^{21}_{k|k}]^{-1}[P^1_{k|k} - P^{21}_{k|k}]$$

This matrix $P_{k|k}$ is not used in the conventional track-to-track fusion algorithm but is used in the proposed modified track-to-track fusion algorithm.

Finally, by the usual derivation of the Kalman filter, the following fusion algorithm for the modified track-to-track fusion Kalman filter algorithm can easily be derived as:

$$\begin{aligned}
&P_{0|0} = \mathrm{Var}(x_0)\\
&P_{k|k-1} = F_k P_{k-1|k-1} F_k^{\mathrm{T}} + \Gamma_k Q_k \Gamma_k^{\mathrm{T}}\\
&K^m_k = P_{k|k-1}(H^m_k)^{\mathrm{T}}[H^m_k P_{k|k-1} H^{m\mathrm{T}}_k + R_k]^{-1} \quad \text{with } m = 1,2\\
&P^m_{k|k} = [I - K^m_k H^m_k]P_{k|k-1} \quad \text{with } m = 1,2\\
&P^{12}_{k|k} = (P^{21}_{k|k})^{\mathrm{T}} = [I - K^1_k H^1_k]P_{k|k-1}[I - K^2_k H^2_k]^{\mathrm{T}}\\
&P_{k|k} = P^1_{k|k} - [P^1_{k|k} - P^{12}_{k|k}][P^1_{k|k} + P^2_{k|k} - P^{12}_{k|k} - P^{21}_{k|k}]^{-1}[P^1_{k|k} - P^{21}_{k|k}]\\
&\hat{x}_{0|0} = E(x_0)\\
&\hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1}\\
&\bar{z}^m_{k|k-1} = H^m_k \hat{x}_{k|k-1} \quad \text{with } m = 1,2\\
&\hat{x}^m_{k|k} = \hat{x}_{k|k-1} + K^m_k[z^m_k - \bar{z}^m_{k|k-1}] \quad \text{with } m = 1,2\\
&\hat{x}_{k|k} = \hat{x}^1_{k|k} + [P^1_{k|k} - P^{12}_{k|k}][P^1_{k|k} + P^2_{k|k} - P^{12}_{k|k} - P^{21}_{k|k}]^{-1}(\hat{x}^2_{k|k} - \hat{x}^1_{k|k})\\
&\quad k = 1,2,\ldots
\end{aligned}$$

It is noteworthy that the above algorithm is similar in structure to the track-to-track fusion algorithm but with different Kalman gains K^m_k. In the modified case the matrix K^m_k is related to the covariance of the fused state estimate $\hat{x}_{k-1|k-1}$. However the computation cost for the modified track-to-track fusion algorithm is identical to Bar-Shalom's track-to-track fusion algorithm.

2.3. A track fusion model with fused prediction

In this section we will propose a second state fusion algorithm. Consider the track-to-track model again. In the prediction step of the local Kalman filtering, we first fuse the predicted states $\hat{x}^1_{k|k-1}$ and $\hat{x}^2_{k|k-1}$ to arrive at a combined state prediction $\hat{x}_{k|k-1}$, then this fused prediction will be used to correct the estimates $\hat{x}^1_{k|k}$ and $\hat{x}^2_{k|k}$ at time k with the measurements z^1_k and z^2_k, respectively, which are then feedback to the prediction procedure for the next iteration. Finally a fused estimate $\hat{x}_{k|k}$ at time k is given with the local estimates $\hat{x}^1_{k|k}$ and $\hat{x}^2_{k|k}$ with fusion Eq. (2.9). The whole procedure is shown in the Fig. 4.

Let $\hat{x}^m_{k-1|k-1}$ be the estimate corresponding to the sensor m ($m = 1,2$) at time $k-1$. From the Fig. 4 the prediction $\hat{x}^m_{k|k-1}$ from time $k-1$ to k is given by

$$\hat{x}^m_{k|k-1} = F_{k-1}\hat{x}^m_{k-1|k-1} \quad \text{with } m = 1,2 \tag{2.12}$$

Then the fused prediction $\hat{x}_{k|k-1}$ is generated by the following fusion equation

$$\begin{aligned}\hat{x}_{k|k-1} = \hat{x}^1_{k|k-1} &+ [P^1_{k|k-1} - P^{12}_{k|k-1}][P^1_{k|k-1} + P^2_{k|k-1}\\ &- P^{12}_{k|k-1} - P^{21}_{k|k-1}]^{-1}(\hat{x}^2_{k|k-1} - \hat{x}^1_{k|k-1})\end{aligned} \tag{2.13}$$

where $P^m_{k|k-1}$ is the covariance matrix for the prediction $\hat{x}^m_{k|k-1}$ ($m = 1,2$), and $P^{12}_{k|k-1} = (P^{21}_{k|k-1})^{\mathrm{T}}$ is the cross covariance matrix between $\hat{x}^1_{k|k-1}$ and $\hat{x}^2_{k|k-1}$. They are defined by

$$\begin{aligned}P^m_{k|k-1} &= E[(x_k - \hat{x}^m_{k|k-1})(x_k - \hat{x}^m_{k|k-1})^{\mathrm{T}}]\\ P^{12}_{k|k-1} &= (P^{21}_{k|k-1})^{\mathrm{T}} = E[(x_k - \hat{x}^1_{k|k-1})(x_k - \hat{x}^2_{k|k-1})^{\mathrm{T}}]\end{aligned}$$

Denote by $P_{k|k-1} = E[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^{\mathrm{T}}]$, i.e., the covariance matrix of the fused prediction state $\hat{x}_{k|k-1}$, then

$$\begin{aligned}P_{k|k-1} = P^1_{k|k-1} &- [P^1_{k|k-1} - P^{12}_{k|k-1}][P^1_{k|k-1} + P^2_{k|k-1}\\ &- P^{12}_{k|k-1} - P^{21}_{k|k-1}]^{-1}[P^1_{k|k-1} - P^{21}_{k|k-1}]\end{aligned} \tag{2.14}$$

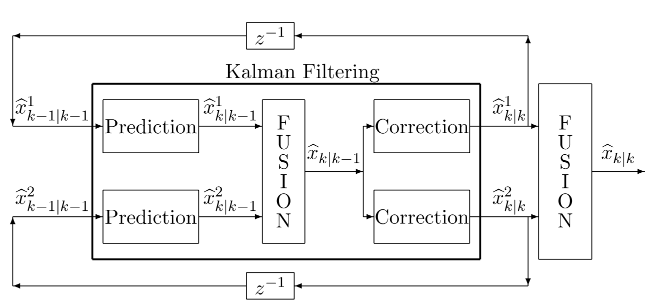

Fig. 4. The track fusion algorithm with fused prediction.

With the fused prediction, the next time state estimate $\hat{x}^m_{k|k}$ at the sensor m is then attained by the following correction equation

$$\begin{aligned}\bar{z}^m_{k|k-1} &= H^m_k \hat{x}_{k|k-1}\\ \hat{x}^m_{k|k} &= \hat{x}_{k|k-1} + P^m_{k|k-1}(x,z)P^m_{k|k-1}(z,z)^{-1}[z^m_k - \bar{z}^m_{k|k-1}]\end{aligned} \tag{2.15}$$

where $P^m_{k|k-1}(x,z) = E[(x_k - \hat{x}_{k|k-1})(z^m_k - \bar{z}^m_{k|k-1})^{\mathrm{T}}]$ and $P^m_{k|k-1}(z,z) = E[(z^m_k - \bar{z}^m_{k|k-1})(z^m_k - \bar{z}^m_{k|k-1})^{\mathrm{T}}]$. We refer to $K^m_k = P^m_{k|k-1}(x,z)P^m_{k|k-1}(z,z)^{-1}$ as the modified Kalman filter gain. It is obvious that

$$K^m_k = P_{k|k-1}(H^m_k)^{\mathrm{T}}[H^m_k P_{k|k-1}(H^m_k)^{\mathrm{T}} + R_k]^{-1}$$

where $P_{k|k-1}$ is defined by (2.14).

After that, from the local estimate $\hat{x}^m_{k|k}$ with $m = 1, 2$ which are sent to next prediction procedure, a new state estimate $\hat{x}_{k|k}$ can be fused with the equation

$$\begin{aligned}\hat{x}_{k|k} = \hat{x}^1_{k|k} &+ [P^1_{k|k} - P^{12}_{k|k}][P^1_{k|k} + P^2_{k|k} - P^{12}_{k|k} - P^{21}_{k|k}]^{-1}\\ &\times (\hat{x}^2_{k|k} - \hat{x}^1_{k|k})\end{aligned} \tag{2.16}$$

where $P^m_{k|k}$ is the covariance matrix for the local tracked estimate $\hat{x}^m_{k|k}$ based on the measurement of sensor m ($m = 1, 2$), and $P^{12}_{k|k} = (P^{21}_{k|k})^{\mathrm{T}}$ is the cross covariance matrix between $\hat{x}^1_{k|k}$ and $\hat{x}^2_{k|k}$. As in the track-to-track fusion, the fused results $\hat{x}_{k|k}$ will just be stored at the fusion site.

In this fusion procedure there are two fusion steps, one for the fused prediction (2.13) and the other for the fused state estimate (2.16), so more (cross) covariance matrices, such as $P^m_{k|k-1}$, $P^m_{k|k}$, $P_{k|k-1}$ etc, should be computed. With the mathematical derivation, a compact recursive algorithm can be obtained as:

The algorithm for track fusion model with fused prediction (TFP)

Given $P^1_{0|0}$; $P^2_{0|0}$; $P^{12}_{0|0}$; $P^{21}_{0|0}$;

$P^m_{k|k-1} = F_{k-1}P^m_{k-1|k-1}F^{\mathrm{T}}_{k-1} + \Gamma_{k-1}Q_{k-1}\Gamma^{\mathrm{T}}_{k-1}$ with $m = 1, 2$

$P^{12}_{k|k-1} = (P^{21}_{k|k-1})^{\mathrm{T}} = F_{k-1}P^{12}_{k-1|k-1}F^{\mathrm{T}}_{k-1} + \Gamma_{k-1}Q_{k-1}\Gamma^{\mathrm{T}}_{k-1}$

$P_{k|k-1} = P^1_{k|k-1} - [P^1_{k|k-1} - P^{12}_{k|k-1}][P^1_{k|k-1} + P^2_{k|k-1} - P^{12}_{k|k-1} - P^{21}_{k|k-1}]^{-1}[P^1_{k|k-1} - P^{21}_{k|k-1}]$

$K^m_k = P_{k|k-1}(H^m_k)^{\mathrm{T}}[H^m_k P_{k|k-1}H^{m\mathrm{T}}_k + R_k]^{-1}$ with $m = 1, 2$

$P^m_{k|k} = [I - K^m_k H^m_k]P_{k|k-1}$ with $m = 1, 2$

$P^{12}_{k|k} = (P^{21}_{k|k})^{\mathrm{T}} = [I - K^1_k H^1_k]P_{k|k-1}[I - K^2_k H^2_k]^{\mathrm{T}}$

$\hat{x}^1_{0|0} = E(x_0)$; $\hat{x}^2_{0|0} = E(x_0)$;

$\hat{x}^m_{k|k-1} = F_{k-1}\hat{x}^m_{k-1|k-1}$ with $m = 1, 2$

$\hat{x}_{k|k-1} = \hat{x}^1_{k|k-1} + [P^1_{k|k-1} - P^{12}_{k|k-1}][P^1_{k|k-1} + P^2_{k|k-1} - P^{12}_{k|k-1} - P^{21}_{k|k-1}]^{-1}(\hat{x}^2_{k|k-1} - \hat{x}^1_{k|k-1})$

$\bar{z}^m_{k|k-1} = H^m_k \hat{x}_{k|k-1}$ with $m = 1, 2$

$\hat{x}^m_{k|k} = \hat{x}_{k|k-1} + K^m_k[z^m_k - \bar{z}^m_{k|k-1}]$ with $m = 1, 2$

$\hat{x}_{k|k} = \hat{x}^1_{k|k} + [P^1_{k|k} - P^{12}_{k|k}][P^1_{k|k} + P^2_{k|k} - P^{12}_{k|k} - P^{21}_{k|k}]^{-1}(\hat{x}^2_{k|k} - \hat{x}^1_{k|k})$

$k = 1, 2, \ldots$

3. Numerical example

The two conventional measurement fusion algorithms covered in Section 2 have been compared in [9], where it has been shown that the two measurement fusion methods are functionally equivalent when the measurement matrices satisfy the condition $H^1_k = H^2_k$ and irrespective of the level of state noise. A previous study [15] showed that the performance of the track-to-track fusion algorithm is worse than that of the optimal method. However, Chang et al. [5] pointed out the implicit approximation made in [1] and showed that the track-to-track fusion algorithm is only optimal in the maximum likelihood sense. In this section we compare the performance of the proposed algorithms and the conventional track-to-track fusion algorithm and the measurement fusion algorithm.

To illustrate the proposed fusion methodology of this paper, a simple example involving a 2D state vector fusion employing two sensors is discussed. In this example, the target tracked by two sensors is modeled by the kinematic model

$$x_{k+1} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} x_k + \begin{pmatrix} T^2/2 \\ T \end{pmatrix} v_k \tag{3.1}$$

with sampling time $T = 1$ and zero-mean white process noise with variance Q and two dimensional state vector $x_k = [x^1_k, x^2_k]^{\mathrm{T}}$. The measurement of the two sensors is modelled either as

$$z^m_k = [1 \quad 0]x_k + w^m_k, \quad m = 1, 2 \tag{3.2}$$

for similar sensors, or as

$$\begin{cases} z^1_k = [1 \quad 0]x_k + w^1_k \\ z^2_k = [0 \quad 1]x_k + w^2_k \end{cases} \tag{3.3}$$

for dissimilar sensors, where the two measurement noises w^1_k and w^2_k are independent with variance $R^m_k = R^m = 0.5$.

We take $x_0 = [0 \quad 0.5]^{\mathrm{T}}$ as initial condition and the number of samples is 500. The tracking algorithm are evaluated at different state noise levels of $Q = 10^{-2}$, 10^{-1}, 1, 10, 20, 50, 100. For the MTF algorithm and track TFP, the derivation of the exact corresponding steady-state covariance is quite involved, so instead of using steady-state covariance we compare the estimated covariance matrix $\widehat{P}_{k|k}$ of the algorithms via Monte Carlo runs [3].

3.1. Simulation of two new fusion methods with respect to the optimal steady-state filters

To analyse the performance of the two new fusion methods, the steady-state ($\alpha - \beta$) filter for this target is determined by a maneuvering index [3],

$$\lambda = \sqrt{Q}T^2/\sqrt{R}$$

With the model described by (3.1) and (3.2), 1000 Monte Carlo runs were conducted. In each Monte Carlo run, the target trajectory was simulated over 500 samples. Two sensor models were considered, i.e. the similar sensors, (3.1) with (3.2), and the dissimilar sensors, (3.1) with (3.3). For simplicity denote

$$P = \widehat{P}_{k|k} = \begin{pmatrix} p_1 & p_2 \\ p_2 & p_3 \end{pmatrix}$$

the covariance matrix P of algorithms and P^* is the optimal steady-state error covariance matrix with a single sensor which can be computed by the formulas given in [3].

3.1.1. Similar sensors

Fig. 5(a)–(c) illustrate the ratio (p_i/p_i^*) of the estimated covariance matrices of the MTF algorithm to the optimal single sensor covariance matrix for a wide range of process or state noise Q in case of similar sensor model (3.2).

Note: throughout the section, Figs. 5–9 have effectively six curves on each corresponding to the time $k =$ 5 $(+)$, 10 $(\times)$, 50 $(\bigcirc)$, 100 $(*)$, 300 $(\square)$ and 500 $(\triangle)$. For the majority most are congruent, i.e., the estimated covariance convergences a steady-state value.

Fig. 6(a)–(c) show the similar results for the TFP algorithm.

It can be seen from the figures that the fusion results of both MTF and TFP algorithms are relatively insensitive to process noise, except for the element p_2 (covariance between the estimated position and the estimated speed). This is mainly due to the additional fusion procedures in the algorithms (see, Figs. 3 and 4). The state space model described by (3.1) and (3.2) (or (3.3)) satisfies the conditions under which the state estimate covariance $\widehat{P}_{k|k}$ of the track-to-track fusion algorithm will converge to a steady-state value [3]. Note that the curves corresponding to $k = 50$, 100, 300 and 500 are coincident in plots (a), (b) and (c) demonstrating that the estimated covariance of the modified track-to-track fusion algorithm also converges to a steady-state value.

3.1.2. Dissimilar sensors

In the case of dissimilar sensors, i.e., the model (3.1) with (3.3), a similar simulation is conducted and the

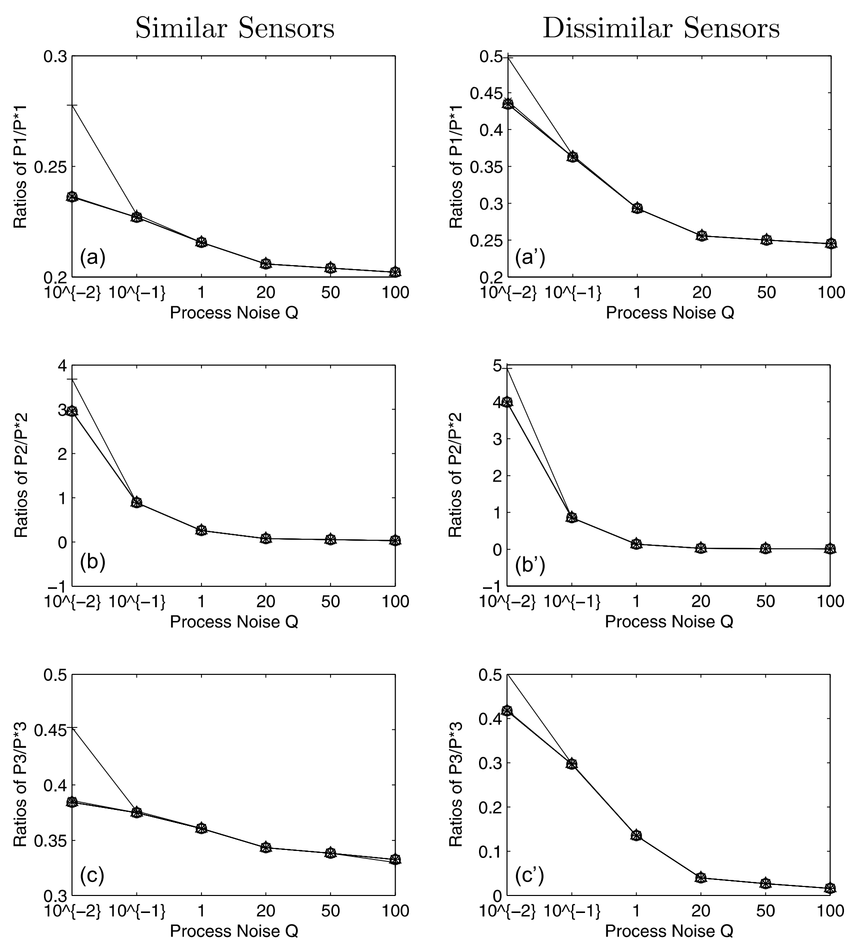

Fig. 5. The results for measurement models (3.2) and (3.3) by the MTF algorithm.

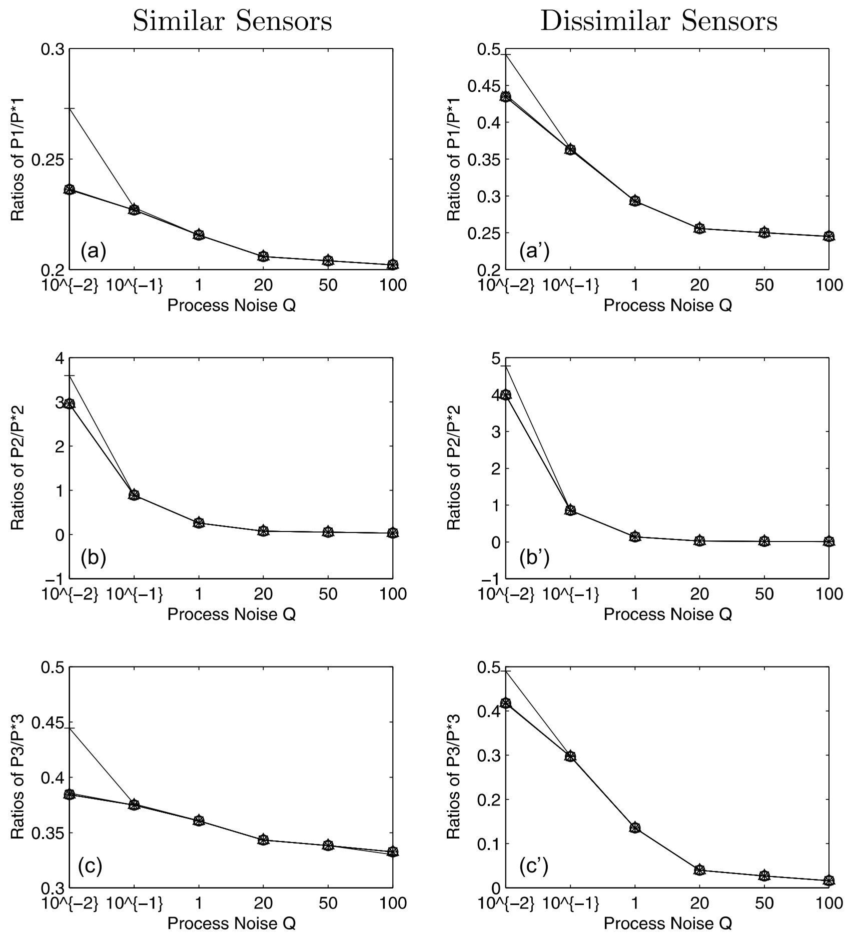

Fig. 6. The results for measurement models (3.2) and (3.3) by TFP algorithm.

corresponding simulated results are shown in plots (a′), (b′) and (c′) of Figs. 5 and 6. We choose the ratios of estimation corresponding to the time intervals $k = 5$, 10, 50, 100, 300, 500 with the same markers. Note that the fusion results in the case of dissimilar sensors are sensitive to the state noise (particularly for p_2 and p_3).

From these simulations we conclude that for lower levels of state noise the optimal steady-state fusion method is a good choice because it provides better estimates, and for larger state noise, the MTF algorithm is a reasonable selection with good performance at the same computational cost as the conventional track-to-track fusion method.

3.2. Performance of the two new fusion methods

Here we compare the performances of the two new algorithms, i.e., the MTF algorithm (see Section 2.2) and the TFP algorithm (see Section 2.3). The simulation results are shown in Fig. 7. Plots (a), (b) and (c) are corresponding to measurement model (3.2), whilst plots (a′), (b′) and (c′) for measurement model (3.3). In these plots the ratio is the state estimate covariance matrix of the MTF algorithm to that of the TFP algorithm. Fig. 7 illustrates almost equivalent performance between the two new proposed methods for both measurement models (3.2) and (3.3). However we have found that the track fusion algorithm with fused prediction can be numerically unstable for similar sensors. Alternatively, the TFP algorithm is computationally more expensive than the MTF algorithm at each estimation step. Hence we will just consider the MTF algorithm in the following subsections.

3.3. Comparison between the modified track-to-track fusion and the conventional track-to-track fusion

The conventional track-to-track fusion (TTF) algorithm (see Section 2.2) is a frequently used for the track-to-track fusion method. With the models described in (3.1) and (3.2) or (3.3), 1000 Monte Carlo runs were conducted. Fig. 8 shows the experimental results in cases for both similar sensors and dissimilar sensors. Plots (a), (b) and (c) show the ratios of the elements of the covariance matrix of MTF algorithm to that of TTF

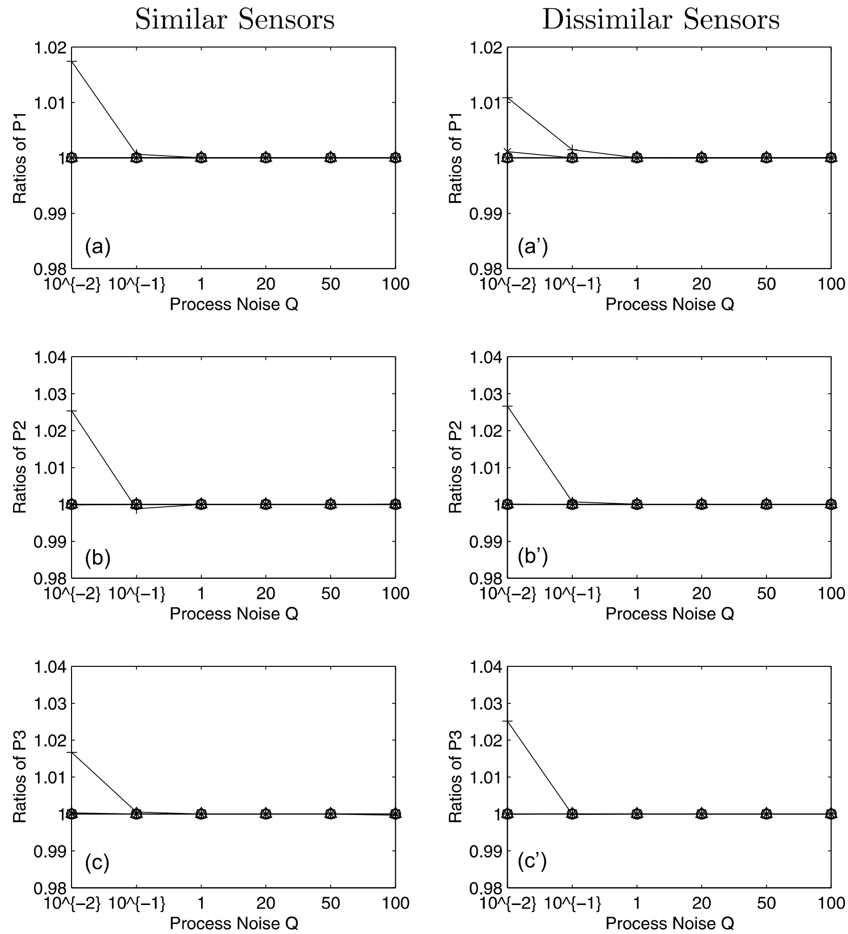

Fig. 7. The results for measurement models (3.2) (a)–(c) and (3.3) (a′)–(c′) by MTF and TFP algorithms.

algorithm. The simulation results at different time intervals are represented by the distinct markers as previously used. As can be seen from the plots, the TTF algorithm is superior to the MTF algorithm in the case of similar sensors (see (a), (b) and (c)), whilst the MTF algorithm is better than the TTF algorithm in the case of dissimilar sensors. As there is an additional fusion procedure step in MTF algorithm, the information from dissimilar sensors is fused and fedback for the next step of estimation, however this procedure will not provide any further fused information in the case of similar sensors. Also note that the computational costs for these two methods are the same. That means that it is possible to obtain a better estimation by the MTF algorithm in the case of dissimilar sensors without increasing computational cost compared to the conventional track-to-track fusion algorithm.

3.4. Comparison between the measurement fusion and modified track-to-track fusion

As shown in [9] the two measurement fusion algorithms described in Section 2.1 are functionally equivalent in the case of similar sensors, and it was also demonstrated by [15] that the measurement fusion algorithm (MF) is superior to the track-to-track fusion (TTF) algorithm. That means that the MF method is optimal in the case of similar sensors. So here we just compare the performance of the conventional measurement fusion (MF) with our MTF algorithm in the case of dissimilar sensors. Fig. 9(a), (b) and (c) show the ratios of the elements p_i of the estimate covariance matrices P by MTF algorithm to that of the estimate covariance matrix by MF method in the case of the measurement model (3.3) for different values of process noise Q. There are six curves in each of plots (a), (b) and (c) respectively, corresponding to the time $k = 5$ $(+)$, 10 $(\times)$, 50 $(\bigcirc)$, 100 $(*)$, 300 $(\square)$ and 500 $(\triangle)$. And plot (d) shows the estimates of velocity x_k^2 at the process noise $Q = 10$.

From Fig. 9 we note that the MTF algorithm outperforms the MF method with less estimated variances of p_1 and p_3 in the case of dissimilar sensors, see [9]. But the element p_2 in MTF algorithm is more sensitive to the process noise levels than MF method due to the additional fused procedure.

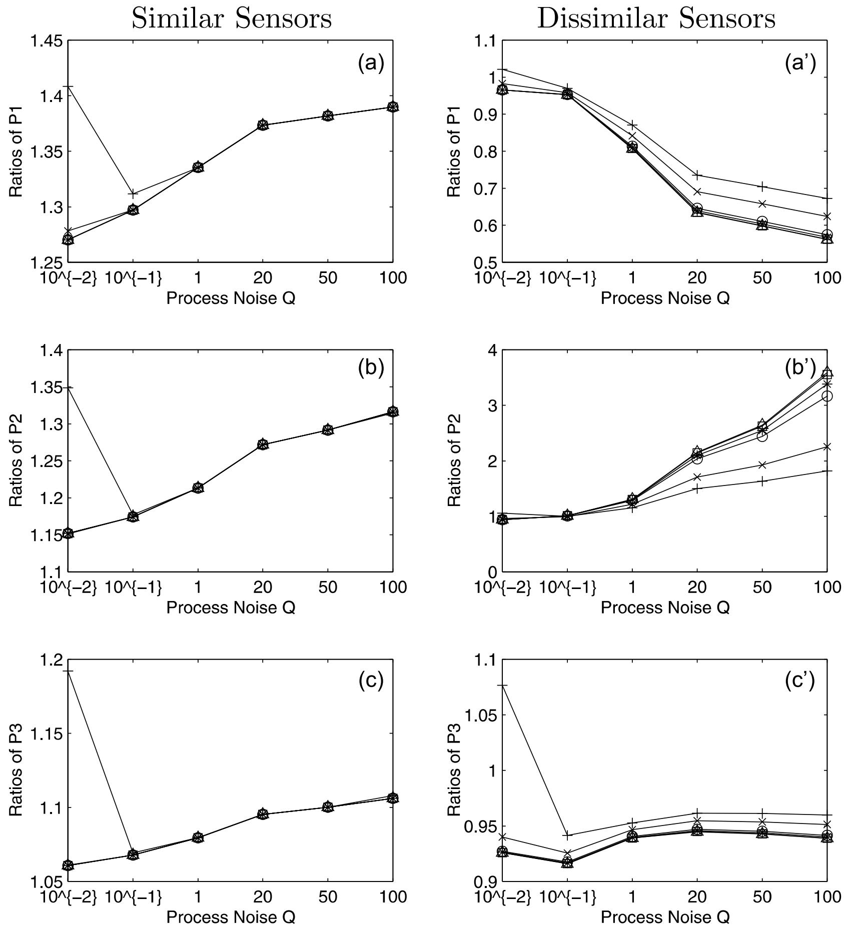

Fig. 8. The results for measurement models (3.2) and (3.3) by MTF and TTF algorithms.

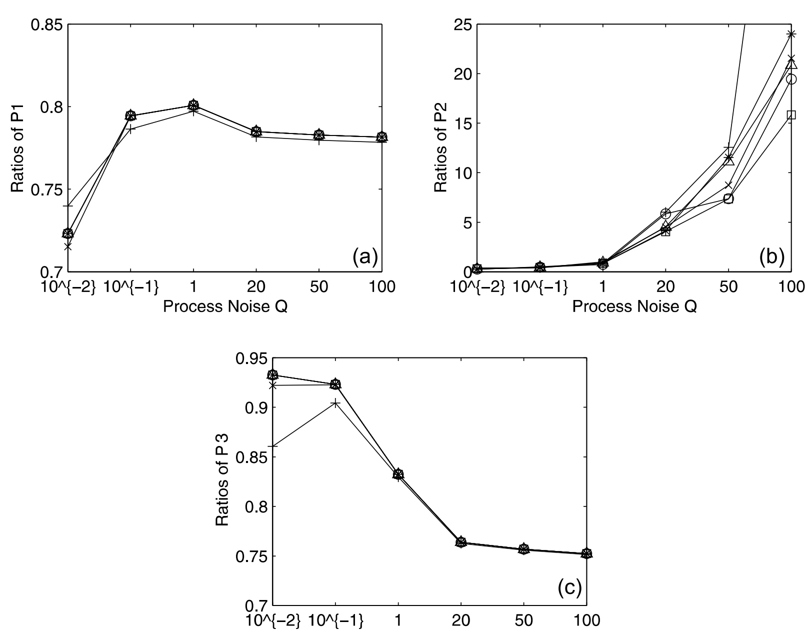

Fig. 9. The results for measurement models (3.3) by MF and TFP algorithms.

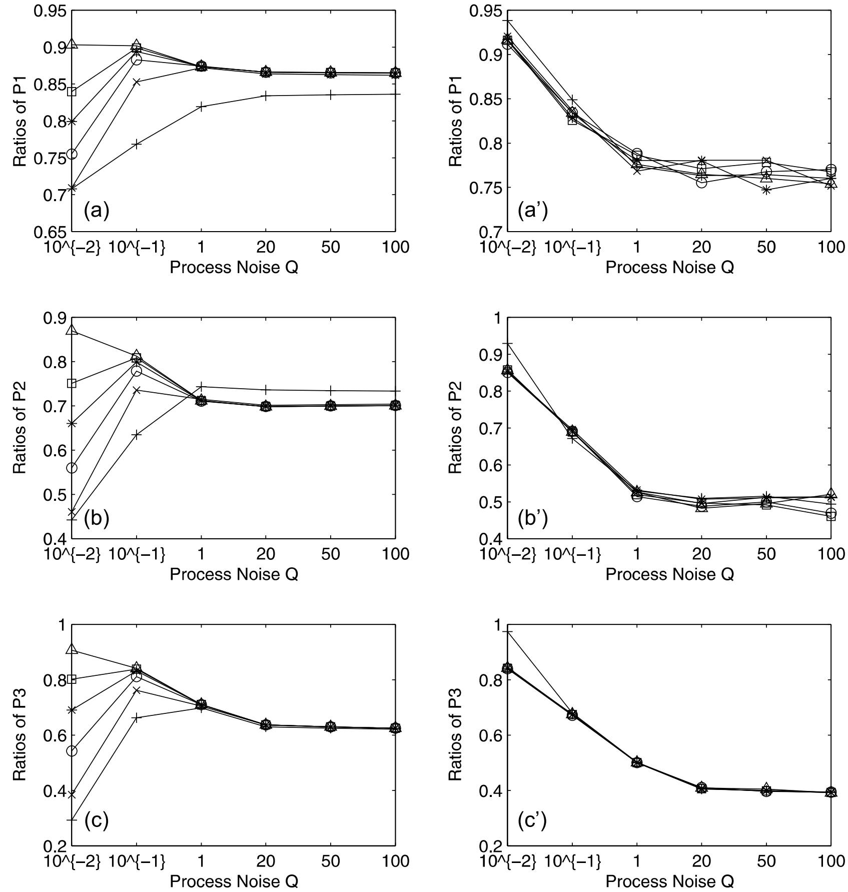

Fig. 10. The results for measurement models (3.3) by TFP and OIF algorithms.

3.5. Comparison between the modified track-to-track fusion and the optimal fusion method

As discussed above, the MTF algorithm offers better performance in the case of dissimilar sensors. Here we compare the performance of the MTF algorithm and the optimal information fusion (OIF) method for just the case of dissimilar sensors. We conducted 1000 Monte Carlo runs, and in order to give additional comparison the fused estimated covariance matrix was also estimated by the average root mean squares (RMS) errors of the Monte Carlo runs. In Fig. 10 the ratios of the elements of the the covariance matrices of the MTF algorithm to the OIF algorithm for a different process noise are given for the different time intervals. Plots (a), (b) and (c) show the ratios of the element of covariance matrix computed in the algorithms, and plots (a′), (b′) and (c′) present the ratios of the element of covariance matrix estimated from the RMS errors of the states. It can be seen from the plots of Fig. 10 that the MTF algorithm performs better than the OIF algorithm. This conclusion also follows from the simulation results in the above subsection as the performance results of the OIF (full-rate communication) method are same as that one obtained by MF method [6] and the MTF is better than MF in the case of dissimilar sensors.

4. Conclusion

In this paper a brief review of various Kalman-filter-based multisensor data fusion algorithms has been given, and two new fusion algorithms of state-based fusion are introduced. In order to compare these algorithms benchmark simulations are made. In general, the MTF can be viewed as a sub-optimal state fusion method, but, compared to the measurement fusion methods and track-to-track fusion algorithm etc. MTF algorithm proposed in Section 2 for dissimilar sensors is more effective both in computational cost (compared to MF algorithm) and for high levels of process noise.

Acknowledgements

The authors gratefully acknowledge the financial support of EPSRC. The authors would like to thank the reviewers for their helpful and constructive comments.

References

[1] Y. Bar-Shalom, L. Campo, The effect of the common process noise on the two-sensor fused-track covariance, IEEE Transactions on Aerospace and Electronic Systems 22 (6) (1986) 803–805.

[2] Y. Bar-Shalom, T. Fortmann, Tracking and data association, in: Mathematics in Science and Engineering, vol. 179, Academic Press Inc., New York, 1988.

[3] Y. Bar-Shalom, X. Li, Estimation and Tracking: Principles, Techniques, and Softwares, Artech House, Boston, London, 1993.

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

[5] K. Chang, R. Saha, Y. Bar-Shalom, On optimal track-to-track fusion, IEEE Transactions on Aerospace and Electronic Systems 33 (4) (1997) 1271–1275.

[6] K. Chang, Z. Tian, R. Saha, Performance evaluation of track fusion with information filter, in: FUSION'98 International Conference, 1998, pp. 648–655.

[7] C. Chui, G. Chen, Kalman filtering with real-time applications, in: Springer Series in Information Sciences, vol. 17, Springer-Verlag, New York, 1987, pp. 23–24.

[8] Q. Gan, C. Harris, Linearisation and state estimation of unknown discrete-time nonlinear dynamic systems using recurrent neurofuzzy networks, Research report, ISIS Research Group, Department of Electronics and Computer Science, University of Southampton, 1998.

[9] Q. Gan, C. Harris, Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion, Research report, ISIS Research Group, Department of Electronics and Computer Science, University of Southampton, 1999.

[10] C. Harris, A. Bailey, T. Dodd, Multisensor data fusion in defense and aerospace, Journal of Royal Aerospace Soceity 162 (1015) (1998) 229–244.

[11] M. Liggins, C. Chong, I. Kadar, M. Alford, V. Vannicola, S. Thomopoulos, Distributed fusion architectures and algorithms for target tracking, Proceedings of IEEE 85 (1) (1997) 95–107.

[12] R. Luo, M. Kay, Multisensor integration and fusion in intelligent systems, IEEE Transactions on Systems, Man and Cybernetics 19 (5) (1989) 901–931.

[13] J. Manyika, H. Durrant-Whyte, Data Fusion and Sensor Management: A Decentralized Information—Theoretic Approach, Ellis Horwood, London, 1994.

[14] F. Martinerie, Data fusion and tracking using HMMS in a distributed sensor network, IEEE Transactions on Aerospace and Electronic Systems 33 (1) (1997) 11–28.

[15] J. Roecker, C. McGillem, Comparison of two-sensor tracking methods based on state vector fusion and measurement fusion, IEEE Transactions on Aerospace and Electronic Systems 24 (4) (1988) 447–449.

[16] M. Stevens, J. Beveridge, Precise matching of 3d target models to multisensor data, IEEE Transactions on Image Processing 6 (1) (1997) 126–142.

[17] H. Wang, M. Kung, T. Lin, Multimodel adaptive Kalman filter design for manoeuvring target tracking, International Journal of Systems Science 25 (11) (1994) 2039–2046.

[18] D. Willner, C. Chang, K. Dunn, Kalman filter algorithms for a multi-sensor system, in: Proceedings of the IEEE Conference on Decision and Control, 1976, pp. 570–574.