Performance Limits of Track-to-Track Fusion vs. Centralized Estimation: Theory and Application

H. Chen, T. Kirubarajan, and Y. Bar-Shalom*
Department of Electrical and Computer Engineering
University of Connecticut
Storrs, CT 06269-2157, USA

Abstract – *Track-to-track fusion is an important part in distributed multisensor-multitarget tracking. The centralized and distributed tracking configurations were studied in [6] using simulated air-to-air scenarios, and in [5] with analytical results based on α-β filters. The current work generalizes the results in [5] to the cases with more than 2 sensors. As the number of sensors increases, the performance of the distributed tracker is shown to degrade compared to the centralized estimation even when the optimal track-to-track fusion is used. An approximation technique for track-to-track fusion is also compared with the optimal approach with performance curves for various numbers of sensors. These performance curves can be used in designing a fusion system where certain trade-offs need to be considered.*

Keywords: Track-to-track fusion, distributed information processing, centralized information processing, α-β filter, target tracking.

1 Introduction

In a distributed multisensor environment where each sensor processes its own measurement and keeps tracks separately, an important question is to decide whether two tracks coming from different sensors represent the same target. If so, the next problem is how to combine (fuse) the two track estimates together.

To determine whether two tracks represent the same target, and if yes, how to fuse them, the information from the latest track estimates is not sufficient. The covariance between the two track estimates (the "cross-covariance") from different sensors has to be computed as in [2]. The optimal track-to-track fusion formula derived in [2] to combine the local estimates is a maximum likelihood (ML) estimator [5], and it has a larger error than the optimal method with centralized configuration. In [5] the authors point out that the performance degradation using track-to-track fusion is about 7% for the mean square error (MSE). Performance curves are also shown in [5] for various target maneuvering indices. However, these results are only for the fusion of the tracks from two sensors. In this work we derive the algorithm and present performance curves for fusing N tracks for the second order motion model (white noise acceleration — WNA — for which the steady state estimator is the α-β filter) using an exact approach.[1] We show that as the number of sensors increases, the performance of distributed tracking keeps degrading in comparison with the centralized one.[2] The results are based on steady state filter variance and they also provide the bound to compare the centralized and distributed tracking configurations through simulations as in [6].

Section 2 presents the general algorithm for fusing N tracks. The evaluation and comparison with the centralized estimator are presented in Section 3. An approximation of the cross-covariance in track-to-track fusion, presented in Section 4, can be easily implemented in real tracking systems. The performance is compared with the optimal track-to-track fusion in terms of mean square error of the fused state estimate. The comparison of a distributed estimator with 4 platforms vs. a centralized one for a realistic air-to-air encounter scenario is presented in Section 5. Conclusions are given in Section 6.

2 Track-to-Track Fusion with N Sensors

Assume we have N sensors and the state estimates at time k from sensor s_i and sensor s_j are the n-vectors $\hat{x}^{s_i}(k|k)$ and $\hat{x}^{s_j}(k|k)$, respectively. For steady state analysis, the time index is dropped in the sequel. Without loss of generality, we denote the state estimates as $\hat{x}^{s_i}$ and $\hat{x}^{s_j}$, with covariances $P^{s_i s_i}$, $P^{s_j s_j}$ and cross-covariance $P^{s_i s_j}$, respectively. Denote the true state of the target as x and assume the estimates from all sensors are purely target originated. As discussed in [5], under Gaussian assumption, the (negative) log-likelihood function is

$$\begin{aligned} L(x) &= -\ln p(\hat{x}^{s_1}, ..., \hat{x}^{s_N}|x) \\ &= c + \frac{1}{2}\left(\begin{bmatrix} \hat{x}^{s1} \\ \hat{x}^{s2} \\ \cdots \\ \hat{x}^{s_N} \end{bmatrix} - \begin{bmatrix} I \\ I \\ \cdots \\ I \end{bmatrix} x \right)' \mathbf{P}^{-1} \\ & \left(\begin{bmatrix} \hat{x}^{s1} \\ \hat{x}^{s2} \\ \cdots \\ \hat{x}^{s_N} \end{bmatrix} - \begin{bmatrix} I \\ I \\ \cdots \\ I \end{bmatrix} x \right) \end{aligned} \quad (1)$$

*Corresponding author, email: ybs@ee.uconn.edu. Research sponsored by AFRL under subcontract C/UB-2730-5A, AFOSR Grant F49620-00-1-0052 and ONR Grant N00014-97-1-0502.

[1] The fusion equations are not restricted to any particular motion model, but the results are believed to be most meaningfully illustrated on the WNA motion model.

[2] With more sensor platforms from which tracks are available for fusion, this is becoming a practical situation of interest.

where c is a constant, I is the $n \times n$ identity matrix and P is the $Nn \times Nn$ covariance matrix with blocks as follows

$$\mathbf{P} = \begin{bmatrix} P^{s_1 s_1} & P^{s_1 s_2} & \dots & P^{s_1 s_N} \\ P^{s_2 s_1} & P^{s_2 s_2} & \dots & P^{s_2 s_N} \\ \dots & & \ddots & \dots \\ P^{s_N s_1} & P^{s_N s_2} & \dots & P^{s_N s_N} \end{bmatrix} \tag{2}$$

Denoting $\mathbf{I} = [I \ I \ ... \ I]'$, an $Nn \times n$ matrix, and $\hat{\mathbf{X}} = [\hat{x}^{s_1} \ \hat{x}^{s_2} \ ... \ \hat{x}^{s_N}]'$, the ML estimator of x is given by

$$\hat{x}^{\text{ML}} = (\mathbf{I}'\mathbf{P}^{-1}\mathbf{I})^{-1}\mathbf{I}'\mathbf{P}^{-1}\hat{\mathbf{X}} \tag{3}$$

which is the general formula for track-to-track fusion from N sensors. This is a generalization of the result from [5]. The covariance corresponding to (3) is

$$P^{\text{dis}} = (\mathbf{I}'\mathbf{P}^{-1}\mathbf{I})^{-1} \tag{4}$$

where the superscript indicates that this pertains to the distributed tracker using optimal track-to-track fusion. The fused ML estimate (3) of the state is also the weighted least squares (LS) estimate where $\mathbf{P}^{-1}$ is the weighting matrix. This follows from the equivalence of LS and ML in the Gaussian case [1].

To simplify the problem and obtain results from which system design (information processing architecture) recommendations can be made, we assume that the N sensors are synchronized and each local tracker uses the same target kinematic model with the same measurement noise statistics. Thus we can denote the steady state covariance matrix $P^{s_i s_i}$ of each "local" tracker as P and the cross-covariance matrix between any two trackers as $P^{\times}$.

In the sequel, we will present the solution for the distributed configuration of α-β filters for tracking an object whose acceleration is modeled as white noise. The state and measurement equations are

$$\begin{aligned} x(k+1) &= Fx(k) + v(k) \\ &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} v(k) \end{aligned} \tag{5}$$

$$z(k) = Hx(k) + w(k) = [1 \quad 0]x(k) + w(k) \tag{6}$$

Denote the variance of process noise as σ_v^2, the measurement noise variance as σ_w^2. The target maneuvering index is defined as

$$\lambda = \frac{\sigma_v T^2}{\sigma_w} \tag{7}$$

Then the steady state filter gain is

$$W = \left[\alpha \quad \frac{\beta}{T}\right]' \tag{8}$$

where

$$\alpha = -\frac{1}{8}\left(\lambda^2 + 8\lambda - (\lambda+4)\sqrt{\lambda^2+8\lambda}\right) \tag{9}$$

$$\beta = \frac{1}{4}\left(\lambda^2 + 4\lambda - \lambda\sqrt{\lambda^2+8\lambda}\right) \tag{10}$$

The steady state filter covariance is

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} = \begin{bmatrix} \alpha & \frac{\beta}{T} \\ \frac{\beta}{T} & \frac{\beta(\alpha-\beta/2)}{(1-\alpha)T^2} \end{bmatrix} \sigma_w^2 \tag{11}$$

The steady state cross-covariance matrix [2] is determined by

$$P^{\times} = [I - WH][FP^{\times}F' + Q][I - WH]' \tag{12}$$

where

$$Q = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} \\ \frac{T^3}{2} & T^2 \end{bmatrix} \sigma_v^2 \tag{13}$$

The above Lyapunov matrix equation (12) can be solved numerically for any given target maneuvering index. This is done for the results to be presented next.

For the WNA motion model used in [1], when the local trackers have equal measurement noise variance, the measurement noise varaince of the centralized filter is σ_w^2/N. The centralized filter is the same as a local tracker but with target maneuvering index changing from λ to

$$\lambda_c = \sqrt{N}\lambda \tag{14}$$

The steady state filter covariance follows (11).

3 Performance Evaluation: Equal Measurement Noise Case

The covariance (4) of the distributed estimate from track-to-track fusion is obtained using (2) whose diagonal blocks follow from (11) and off-diagonal blocks from (12). The covariance P^c of the centralized estimate follows from (11) using the "centralized" maneuvering index (14).

To compare the steady state filter variances between the centralized and distributed tracker, we use the above to illustrate our methodology and evaluate the relative performance as the number of sensors increases. Assuming a sampling interval T=2 s, we compute each component of the covariance matrices P^{dis} and P^c and plot the relative differences between the centralized and distributed tracker in Figs. 1–3. An important observation is that the percentage difference of the mean square error (MSE) between the centralized and distributed tracker increases as the number of sensors increases. This quantifies the performance limits of the distributed estimation with trak-to-track fusion vs. centralized estimation.[3] The difference of the position variance is over 15% for N=4 when the target maneuvering index $\lambda < 1$. The difference of the velocity

[3] See [2] p. 589 for the explanation of the reason that the optimal track-to-track fusion is inferior to optimal centralized estimation.

variance is around 20–25% for N=4. From the performance curves, during uniform target motion period we expect that the distributed tracker should yield about 2.5% (11%) larger RMS position error and about 2% (10%) larger RMS velocity error than the centralized one for N=2 (N=4) sensors. As the target maneuvers, the performance degradation of the distributed tracker becomes smaller in terms of RMS position error while slightly larger in terms of RMS velocity error.

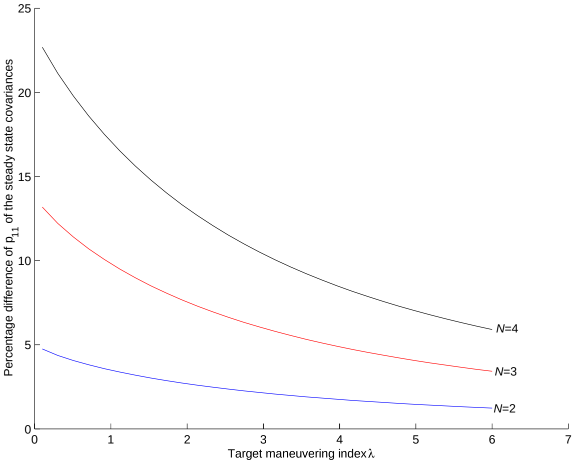

Figure 1: Percentage of the position variance difference between the distributed and the centralized trackers versus number of sensors $[(p_{11}^{\text{dis}} - p_{11}^{c})/p_{11}^{c}]$.

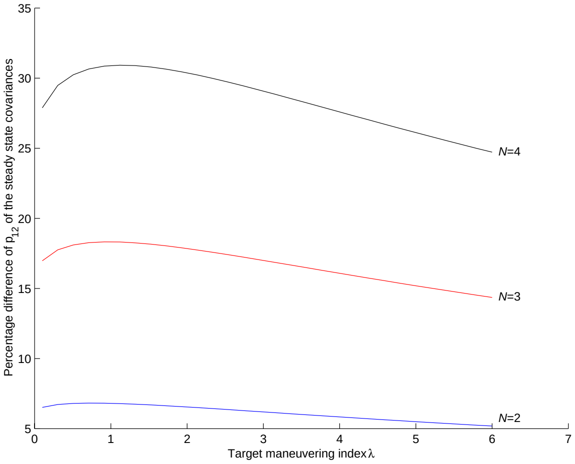

Figure 2: Percentage of the position-velocity covariance difference between the distributed and the centralized trackers versus number of sensors $[(p_{12}^{\text{dis}} - p_{12}^{c})/p_{12}^{c}]$.

4 An Approximation Technique for Track-to-Track Fusion

4.1 Case 1: Equal Measurement Noise

For a real distributed tracking system, the calculation of the cross-covariance using (12) is not practical. The following approximation is considered. From (11) and (12), denoting the ratios between the cross-covariance and the covariance terms as

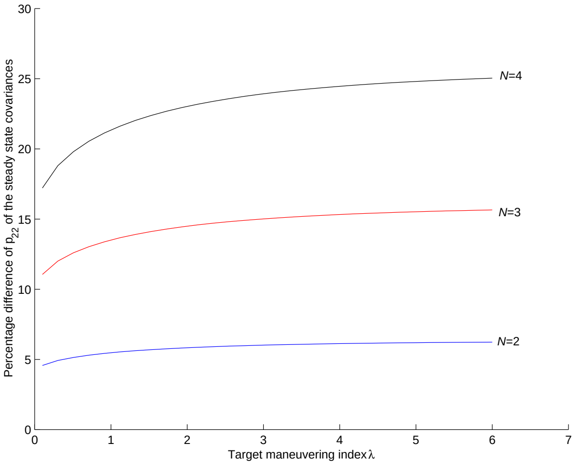

Figure 3: Percentage of the velocity variance difference between the distributed and the centralized trackers versus number of sensors $[(p_{22}^{\text{dis}} - p_{22}^{c})/p_{22}^{c}]$.

$$\rho_{ij} = p_{ij}^{\times}/\sqrt{p_{ii}p_{jj}} \qquad i = 1, 2, \; j = 1, 2 \qquad (15)$$

For target with uniform motion, we use $\rho_{11} = 0.20$, $\rho_{12} = 0.30$, $\rho_{22} = 0.73$ to approximate the cross-covariance matrix. If the target maneuvering index is known (or can be estimated from the track history), we have the following quadratic approximations

$$\rho_{11} = 0.0058\lambda^2 - 0.0567\lambda + 0.1724 \qquad (16)$$

$$\rho_{12} = 0.0070\lambda^2 - 0.0730\lambda + 0.2698 \qquad (17)$$

$$\rho_{22} = 0.0037\lambda^2 - 0.0418\lambda + 0.7102 \qquad (18)$$

valid for $\lambda \in [0.1, \; 6]$. Notice that ρ_{11} and ρ_{12} are much smaller than ρ_{22}.

In order to know the performance of the approximation technique, we calculate the exact mean square error (MSE) of the track-to-track fusion obtained with the approximate cross-covariance matrix based on (15). Denote

$$\bar{\mathbf{P}} = \begin{bmatrix} P^{s_1 s_1} & \bar{P}^{s_1 s_2} & \ldots & \bar{P}^{s_1 s_N} \\ \bar{P}^{s_2 s_1} & P^{s_2 s_2} & \ldots & \bar{P}^{s_2 s_N} \\ \ldots & & \ddots & \ldots \\ \bar{P}^{s_N s_1} & \bar{P}^{s_N s_2} & \ldots & P^{s_N s_N} \end{bmatrix} \qquad (19)$$

where $\bar{P}^{s_i s_j}$ are the approximate cross-covariance matrix between sensor s_i and s_j. Using the same fusion technique given by (3) but with the covariance matrix $\bar{\mathbf{P}}$, the fused state estimate is given by

$$\hat{x}_{\text{a}}^{\text{ML}} = (\mathbf{I}'\bar{\mathbf{P}}^{-1}\mathbf{I})^{-1}\mathbf{I}'\bar{\mathbf{P}}^{-1}\hat{\mathbf{X}} \qquad (20)$$

The fusion center will assume the covariance of the fused state estimate to be

$$\bar{P}^{\text{dis}} = (\mathbf{I}'\bar{\mathbf{P}}^{-1}\mathbf{I})^{-1} \qquad (21)$$

while the exact MSE of the fused state estimate using approximate cross-covariance matrix is

$$P_{\mathrm{a}}^{\mathrm{dis}} = (\mathbf{I}'\bar{\mathbf{P}}^{-1}\mathbf{I})^{-1}(\mathbf{I}'\bar{\mathbf{P}}^{-1}\mathbf{P}\bar{\mathbf{P}}^{-1}\mathbf{I})(\mathbf{I}'\bar{\mathbf{P}}^{-1}\mathbf{I})^{-1} \qquad (22)$$

Notice that the approximate track-to-track fusion is indeed optimal i.e.,

$$\hat{x}_{\mathrm{a}}^{\mathrm{ML}} = \hat{x}^{\mathrm{ML}} \quad \text{and} \quad P_{\mathrm{a}}^{\mathrm{dis}} = P^{\mathrm{dis}} \qquad (23)$$

for any ρ_{ij} given by (15) when all the sensors have the same estimation error covariance and cross-covariance matrices. But the covariance obtained by the fusion center is different from the exact MSE given by (22) depending on the target maneuvering index.

4.2 Case 2: Unequal Measurement Noise

When the local trackers have different covariance and cross-covariance matrices, the MSE obtained using approximate track-to-track fusion is only suboptimal and the MSE difference depends on the approximation coefficients. To simplify the problem, we only consider a two-sensor case. Sensor 1 maintains a local tracker with target maneuvering index λ while sensor 2 with target maneuvering index $\sqrt{r}\lambda$, meaning the variance of the measurement noise from sensor 2 is $\frac{1}{r}$ that of sensor 1 for the same target motion equation. We want to find the ratio r that achieves the maximum MSE difference between the distributed and centralized trackers. First, we consider the position variance difference under various λ. Figs. 4–5 show the ratios at which the corresponding MSE position differences between the distributed and centralized trackers achieve maximum under different target maneuvering indices. Notice that the ratio r at which the difference is maximum is close to 1 when the target undergoes uniform motion and slightly decreases as the target maneuvering index increases. The MSE position difference shown in Fig. 5 is quite close to that of the 2 sensor case with r=1. Figs. 4–5 are a "worst case" analysis for the distributed tracker. In general, the MSE position difference between the distributed and centralized trackers does not change significantly for the unequal measurement noise case. Similar results hold for velocity.

We are also interested in the performance using approximate track-to-track fusion with various r. Notice that the cross-covariance matrix is not symmetric for $r \neq 1$. We found that there is no fixed ρ_{ij} suitable for all cases in the approximate track-to-track fusion. For a real tracking system, the following approximation of the cross-covariance matrix $P^{s_i s_j}$ can be used. To make it suitable for a general estimation problem, we assume the dimension of the state vector is n. Each element of $P^{s_i s_j}$ is approximated by

$$p_{lm}^{s_i s_j} = \rho\,\mathrm{sign}(\rho_{lm}^{s_i}\rho_{lm}^{s_j})\left[|\rho_{lm}^{s_i}\rho_{lm}^{s_j}| \cdot p_{ll}^{s_i} p_{mm}^{s_j}\right]^{\frac{1}{2}} \qquad (24)$$
$$l, m = 1, 2, ..., n$$

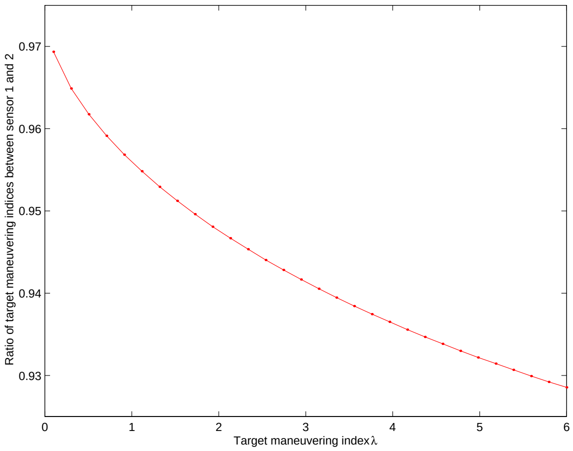

Figure 4: Ratio between the target maneuvering indices of the two sensors at which one has maximum MSE position difference between the distributed and centralized trackers for various target maneuvering indices.

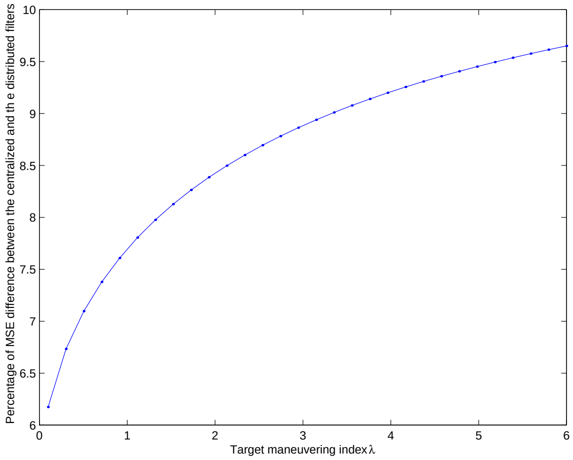

Figure 5: Percentage of the position covariance difference between the distributed and the centralized trackers $[(p_{11}^{\mathrm{dis}} - p_{11}^{c})/p_{11}^{c}]$ at ratio r for which it is maximum.

where

$$\rho_{lm}^{s_i} = \frac{p_{lm}^{s_i}}{\sqrt{p_{ll}^{s_i} p_{mm}^{s_i}}} \qquad (25)$$

and ρ is a constant chosen within [0.1, 0.4].

5 Comparison of the Centralized and Distributed Tracker Using a Realistic Air-to-Air Scenario

In this section we compare the performance between centralized and distributed tracking systems using a realistic air-to-air encounter scenario. The target and platform trajectories are designed to simulate fighter aircraft trajectories pursued by other fighters which use measurements from a radar as well as inertial and

GPS navigation. The scenarios used in this paper are chosen from the 24 recommended scenarios in distributed tracking algorithm (DTA) evaluation plan [3] with the assistance of DTA Trajectory Generator and DTA Sensor Simulator[4]. We consider up to 4 platforms pursuing 3 targets.

5.1 Platform and Target Trajectories, Sensor Models and Filter Configurations

5.1.1 Platform and Target Trajectories

The platform trajectories consist of parallel straight lines with variable initial position offsets. The position offsets can be scaled to provide the desired angular differences in the respective lines of sight to the targets. In [3] each platform trajectory is divided into 3 main course segments, separated by turning and climbing maneuvers. The initial platform position is around −86° in longitude and 44° in latitude. The initial altitude is 20 kft. In dense platform situations, the distance between two platforms is 0.02° difference in geodetic longitude, that is, approximately 7.3 kft. The total time span of DTA scenarios is 2000 s (approximately 33 min). The platforms perform a climb from 20 kft up to 30 kft starting around 930 s and a descent to 21 kft starting around 1390 s. The maximum speed of a platform in the horizontal plane is around 620 ft/s and around 110 ft/s along the vertical axis. The maximum acceleration in climbing is around $1g$ and when descending around $1.5g$. The maximum turn rate is approximately 20°/s.

All platform trajectories have 2 course changes of approximately 30° each (which start around 730 s and 1390 s, respectively), with a maximum bank angle of about 17° as determined by a low turn acceleration level of $0.3g$. Each platform also has 2 vertical maneuvers with altitude changes of up to 10 kft at around 10° flight path angle with respect to the horizontal plane.

The target trajectories also consist of parallel straight lines running perpendicular to the (average) platform trajectories, with initial position offsets to achieve the desired spacing, and located so as to achieve desired distances of closest approach. The target position offsets can be scaled in proportion to the track sensor range and angular measurement errors, so as to make the targets more or less easy to separate via the measurements. The initial target positions are around −82.5° in longitude and 47.5° in latitude. The initial altitude is 27 kft. In dense target situations, the distance between two targets is 0.01° difference in longitude and latitude, that is, approximately 5.2 kft. The medium separation has a 0.02° difference in longitude and latitude while the sparse situation has a 0.03° difference. The maximum speed of a target in the horizontal plane is around 600 ft/s and around 100 ft/s along the (local) vertical axis. The maximum acceleration in diving is $2.14g$ and in jinking it is $3.5g$.

Figure 6 illustrates the general geometry of the target and platform trajectories as viewed from above. For the purposes of illustration, we have shown 3 platforms and 3 targets. Each platform trajectory has 3 course segments as described earlier. The target trajectories also have 3 course segments. The target turns allow near-constant horizontal accelerations to be introduced at prescribed levels. The final segments include 200 s of jinking maneuvers (linked S-turns) with time-varying accelerations at similarly prescribed levels. Target altitude changes (not shown) introduce vertical accelerations as well as changes in target elevation angles as viewed from the platforms. The targets perform two dives at around 1100 s to 22 kft and at around 1450 s to 20 kft. One 30° course change occurs at around 600 s and other course changes (30°, 40°, 50° respectively for the 3 targets) occur around 1400 s. Jinking motion starts around 1500 s.

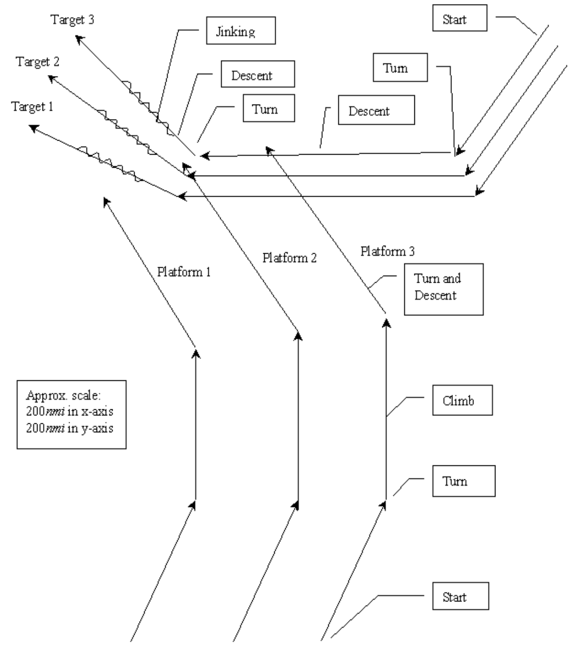

Figure 6: Platform-Target Trajectory Geometry Map for DTA Scenarios (from [4]).

5.1.2 Target Motion Models and Filter Design Parameters

In this subsection we define the terms and state the assumptions used throughout the rest of the section. It is assumed that the surveillance system is implemented at each platform with a local tracker and in the centralized configuration a centralized processor receives measurements from each platform to form global state estimates. The measurements from different platforms are synchronized and the tracker using the measurements

[4] The software for platform and target trajectory generation, sensor measurement simulation was developed by Integrity Systems [3] for the Air Force Research Laboratory.

from all platforms is called the centralized global estimator. In the distributed configuration a fusion center receives the local state estimates from each platform and fuses these estimates to form a global estimate. The tracker using state estimates from all platforms is called distributed estimator. The motion models are defined in the horizontal plane and along the vertical axis of the target's local reference frame. The filters for the horizontal and vertical motion are decoupled.

Two filtering algorithms, namely the Kalman filter and the Interacting Multiple Model (IMM) estimator, are chosen to compare their performance for both centralized and distributed trackers. The motion model used in the Kalman filter is white noise acceleration (WNA) with high process noise (WNA-H). It captures target maneuvering to the highest expected level. The motion models used in the IMM estimator are WNA model with high process noise, WNA model with low process noise (WNA-L, which models uniform target motion), and coordinated turn (CT) model in the horizontal plane. The motion models used in the IMM estimator along the vertical axis are a WNA-L and a WNA-H.

The targets may have different acceleration levels in lateral and axial direction during the maneuvering phase. To model this acceleration discrepancy, we use directional process noise for both the Kalman filter and the IMM estimator. Assume the variance of the process noise is σ_l^2 in the lateral direction, and σ_a^2 in the axial direction. At certain time k, the filter estimated course is denoted as $\phi(k)$ (obtained from the velocity components). The process noise covariance in the horizontal plane is taken as

$$
\begin{aligned}
Q_h = & \begin{bmatrix} -\cos\phi(k) & \sin\phi(k) \\ \sin\phi(k) & \cos\phi(k) \end{bmatrix} \\
& \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_a^2 \end{bmatrix} \begin{bmatrix} -\cos\phi(k) & \sin\phi(k) \\ \sin\phi(k) & \cos\phi(k) \end{bmatrix}
\end{aligned}
\tag{26}
$$

In [6] we found that an IMM estimator using 3 models (WNA-L, WNA-H and CT) in the horizontal plane and 2 models (WNA-L, WNA-H) along the vertical axis yields better estimation accuracy than other choices of target motion models for the IMM estimator[5]. The process noise parameters for the Kalman filter and the IMM estimator modules are listed in Table 1.

The mode transition probabilities for horizontal and vertical motions of the IMM estimator are additional design parameters [1] to provide the trade-off between the peak estimation error at the onset of maneuver and the noise reduction factor during uniform motion. The mean sojourn times of the two modules for the vertical filter are 40 s and 10 s, respectively. The probability transition matrix for (the end of the) interval T is given by

$$
\Pi_v = \begin{bmatrix} \pi_{v11} & \pi_{v12} \\ \pi_{v21} & \pi_{v22} \end{bmatrix} \tag{27}
$$

where $\pi_{v11} = \max\{.5, 1 - \frac{T}{40}\}$, $\pi_{v22} = \max\{.5, 1 - \frac{T}{10}\}$, and $\pi_{v12} = 1 - \pi_{v11}$, $\pi_{v12} = 1 - \pi_{v11}$. Similarly, the mean sojourn times of the three modules for the horizontal filter are 40 s, 20 s and 20 s, respectively. The probability transition matrix for (the end of the) interval T is then given by [2]

$$
\Pi_h = \begin{bmatrix} \pi_{h11} & \pi_{h12} & 0 \\ \pi_{h21} & \pi_{h22} & \pi_{h23} \\ \pi_{h31} & 0 & \pi_{h33} \end{bmatrix} \tag{28}
$$

where $\pi_{h11} = \max\{.5, 1 - \frac{T}{40}\}$, $\pi_{h22} = \pi_{h33} = \max\{.5, 1 - \frac{T}{20}\}$, and $\pi_{h12} = 1 - \pi_{h11}$, $\pi_{h21} = \pi_{h23} = 0.5(1 - \pi_{v11})$, $\pi_{h31} = 1 - \pi_{h33}$. For the measurement interval $T = 2\,\mathrm{s}$ (medium measurement rate) we have

$$
\Pi_h = \begin{bmatrix} 0.95 & 0.05 & 0 \\ 0.05 & 0.9 & 0.05 \\ 0.1 & 0 & 0.9 \end{bmatrix} \tag{29}
$$

$$
\Pi_v = \begin{bmatrix} 0.95 & 0.05 \\ 0.2 & 0.8 \end{bmatrix} \tag{30}
$$

for the horizontal and vertical IMM estimators, respectively. For the CT model, we use an extended Kalman filter (EKF) module. The detailed equations of the target motion models can be found in [8].

Notice that the filters may not reach the steady state even during the target uniform motion for a long period of time since the measurement noise statistics change with time. The target maneuvering index is smaller at the beginning than in the end since the range of the target to the platform becomes closer. The platforms are close enough so that the pairs of platform-target geometries are similar, meaning that the local trackers have similar state estimation error statistics. We will see whether the distributed tracker yields the performance degradation predicted using an α-β filter.

5.1.3 Sensor Models

We assume that each platform is equipped with a radar that measures range, azimuth, and elevation. For the platform and target trajectories illustrated in Figure 6, the sequences of platform and target courses keep the targets within the platform sensor fields of view. The radar, specified in [3], has the maximum detection range of $1.2 \cdot 10^6$ ft (approximately 197 nmi) and maximum angular field of view ±1.047 rad (60°) in azimuth, ±0.524 rad (30°) in elevation. The measurement noise has the statistics given in Table 2.

The probability of detection is modeled as follows. At every scan, if a target is in the radar surveillance

measurement	unit	high	medium	low
range (σ_r)	ft	200	400	1000
azimuth (σ_θ)	mrad	2.0	4.0	1.0
elevation (σ_ϵ)	mrad	2.0	4.0	1.0

Table 2: The standard deviations of the measurement noises.

[5] For civilian air traffic surveillance [8], 2 models (WNA-L and CT) were enough for horizontal motion.

region, the target detection probability depends only on its range. A reference range r_0 is set to $7.6 \cdot 10^5$ ft and the detection probability P_D at r_0 is 0.9. The detection probability at range r is given by

$$P_D = 0.9^{(r/r_0)^4} \tag{31}$$

The false alarm rates can be set to zero or at three levels (low, medium and high). They are represented, respectively, by mean values of 0.2, 1, and 5 false alarms per scan. In [3] the P_{FA} per-cell is set to 10^{-6} for medium accuracy sensor (that is, the mean per-frame values divided by 10^6). The false position measurements are governed by a uniform distribution for range, azimuth and elevation within the surveillance volume. The radar has four options for the setting of measurement rate (high, medium, low, and very low: 1 Hz, 0.5 Hz, 0.2 Hz and 0.1 Hz). The medium rate ($0.5\,Hz, T{=}2$ s) is the nominal rate for all testing scenarios.

Each platform is also equipped with an inertial navigation unit (INU) and a navigation filter (NAV) that simulates GPS outputs of the platform position and velocity. The navigation filter computes GPS-derived Kalman filter corrections to the INU state. The output files are generated using the DTA Trajectory Generator and DTA Sensor Simulator [4].

5.2 Coordinate Systems and Measurement Transformations

The local and global trackers, as presented in Subsection 5.1.2, are implemented in the target local reference frame for any initiated track. The target local reference frame (referred to in the sequel by the subscript *loc*, whenever necessary) is defined as ENU (East, North, Up) centered at its last position estimate in the earth-centered earth-fixed (ECEF) coordinate system. That is, at certain time k, assuming the latest position estimate is $\{x_p(k-1), ECEF\}$, the target local reference frame is specified by the 3×3 rotation matrix A_{loc} and the origin O_{loc}, denoted as

$$\{A_{loc}, O_{loc}\} = AXES\{x_p(k-1), ECEF\} \tag{32}$$

The AXES procedure to derive the *loc* reference frame can be found in [7].

The sensor measurements consist of target range z_r, azimuth z_ϕ and elevation z_θ, i.e., $z_s = [z_r \; z_\phi \; z_\theta]'$. The azimuth and elevation are obtained in the sensor body frame. The measurement z_s is first transformed to ECEF coordinates and then to the target local reference frame. The platform position in ECEF coordinates derived from NAV filter output is used in the first transformation. An issue to be considered here is the possible bias introduced by measurement conversion from spherical (located in the sensor body frame) to Cartesian frame. We need to calculate the bias significance as defined in [2], Section 1.6, to see whether measurement debiasing is required. For low- medium- and high- accuracy radars with parameters given above, the bias significance factors are all below 0.2 at target range of 100 nmi. Thus the standard measurement conversion from polar to Cartesian coordinates is used in our DTA implementation. Denote the measurement in ECEF coordinates as $z_{ECEF} = [z_x \; z_y \; z_z]'$. With the platform position $z^p_{ECEF} = [z^p_x \; z^p_y \; z^p_z]'$, z_{ECEF} is given by

$$\begin{aligned} z_x &= z^p_x + z_r \cos z_\phi \cos z_\theta \\ z_y &= z^p_y + z_r \sin z_\phi \cos z_\theta \\ z_z &= z^p_z + z_r \sin z_\theta \end{aligned} \tag{33}$$

The measurement covariance R_{ECEF} is given by

$$R_{ECEF} = \begin{bmatrix} \cos z_\phi \cos z_\theta & -z_r \sin z_\phi \cos z_\theta & -z_r \cos z_\phi \sin z_\theta \\ \sin z_\phi \cos z_\theta & z_r \cos z_\phi \cos z_\theta & -z_r \sin z_\phi \sin z_\theta \\ \sin z_\theta & 0 & z_r \cos z_\theta \end{bmatrix} \begin{bmatrix} \sigma_{z_r}^2 & & \\ & \sigma_{z_\phi}^2 & \\ & & \sigma_{z_\theta}^2 \end{bmatrix} \begin{bmatrix} \cos z_\phi \cos z_\theta & -z_r \sin z_\phi \cos z_\theta & -z_r \cos z_\phi \sin z_\theta \\ \sin z_\phi \cos z_\theta & z_r \cos z_\phi \cos z_\theta & -z_r \sin z_\phi \sin z_\theta \\ \sin z_\theta & 0 & z_r \cos z_\theta \end{bmatrix}^T \tag{34}$$

At scan scan k denote the measurement vector as $z_{ECEF}(k)$ and its covariance $R_{ECEF}(k)$. Given the platform position $z^p_{ECEF}(k)$ at time k, the measurement $z_{ECEF}(k)$ with its covariance $R_{ECEF}(k)$ are derived as above. The measurement $z_{ECEF}(k)$ and its covariance $R_{ECEF}(k)$ are further converted to $z_{loc}(k)$ with its covariance $R_{loc}(k)$ in the target local reference frame for the filter update. The transformation is

$$\begin{aligned} z_{loc}(k) &= \begin{bmatrix} z_{h,loc}(k) \\ z_{v,loc}(k) \end{bmatrix} \\ &= A_{loc}(k) z_{ECEF}(k) + O_{loc}(k) \end{aligned} \tag{35}$$

$$\begin{aligned} R_{loc}(k) &= \begin{bmatrix} R_{h,loc}(k) & * \\ * & R_{v,loc}(k) \end{bmatrix} \\ &= A_{loc}(k) R_{ECEF}(k) A_{loc}(k)^T \end{aligned} \tag{36}$$

where $*$ denotes the omitted cross-covariance terms. After filter update, new state estimates are formed and the target position $x_p(k)$ in ECEF coordinates is derived, which will be the origin of the target local reference frame for the measurement conversion at the next scan. The flowchart of the coordinate system used for filter update is shown in Fig. 7 and the detailed equations for the coordinate conversion can be found in [7].

The global tracker with centralized configuration combines the measurements from different sensors in the target local reference frame according to the flowchart shown in Fig. 7. Each platform converts its measured range, azimuth and elevation to the position measurement of the target in ECEF coordinates and the centralized filter converts the position measurement to the measurement in target local reference frame with the origin derived using the last filter update.

For the distributed configuration, the state estimates from different platforms may use target reference frames with different origins. We first convert all the state estimates into a common target reference frame and then do the track to track fusion using the approximate technique (20) based on (24). The origin of the target reference frame for the next scan is derived using the fused state estimate of the target. Since the rotation of the coordinate system of a certain target derived by a local tracker to the common reference frame is quite small, the covariance matrices of the state estimates are left unchanged. The position and velocity estimates are transformed from one local reference frame to the common reference frame before doing track to track fusion.

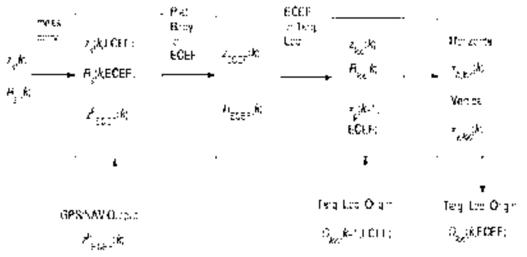

Figure 7: Flowchart of coordinate systems and conversions used in filter implementation.

5.3 Simulation Results

5.3.1 Comparison of the Tracking Accuracy in Different Segments

We choose scenario A3 from [3] to compare the tracking accuracy between the centralized and distributed (global) tracking accuracy. Our major focus here is to evaluate the performance degradation of the distributed tracker using a realistic scenario and see if it is close to the theoretical limit derived earlier. In this scenario, we have zero false alarm rate and the 3 target spacings are medium.

In order to compare the global tracking accuracy for different information processing configurations, we assume each platform maintains its own local track list with perfect data association. Then only the filtering algorithm determines the estimation accuracy. We compare the Kalman filter and IMM estimator with the design parameters given in Section 5.1.2. The RMS position and velocity errors are averaged over 6 target segments, namely constant velocity far from the platforms (far CV, F-CV) starting at 270 s and ending at 550 s, a turn far from the platforms (far turn, F-T) starting at 590 s and ending at 640 s, a descent at medium range from the platforms (medium descent, M-D) starting at 1090 s and ending at 1200 s, a turn at medium range (medium turn, M-T) starting at 1370 s and ending at 1440 s, a second descent starting at 1490 s and ending at 1540 s, a jinking motion at close range (close jinking, C-J) starting at 1550 s and ending at 1750 s[6], and a uniform motion at close range (close constant velocity, C-CV)[7] starting at 1800 s and ending at 2000 s. For all the test scenarios, the radar scan rate is 0.5 Hz. Table 3 shows the RMS errors obtained using the Kalman filter and the IMM estimator for scenario A4 for the various segments (all but the very short second descent). We can see that the IMM estimator yields less RMS errors for all the segments of the target trajectories. Notice that target 2 yields larger RMS errors than target 1 during the jinking period and target 3 yields larger RMS errors than target 2 during the jinking. Comparing the results among all the segments of each target between centralized and distributed trackers, we can see that the distributed track yields around 10%-15% larger RMS position and velocity errors than the centralized one. The results are close to but slightly larger than the theoretical bounds of the distributed tracker.

To test the estimation accuracy of global tracker versus local tracker under false alarm environment, we use scenario D3 from [3] with 3 platforms and 3 targets. For simplicity, the tracking accuracy is interpreted as the average RMS errors over the whole track life. 100 Monte Carlo runs are used and the results are listed in Table 4. We can see that the tracking accuracy does not degrade very much when 2-D assignment is used for data association. The IMM estimator yields less RMS errors than Kalman filter for both local and global estimators. These RMS errors for global tracker with distributed configuration are around 7%-10% worse than those with centralized configuration, which is close to the theoretical bound of the distributed tracker.

5.3.2 Comparison of the Computational Load

We want to compare the computational load of the Kalman and the IMM estimator using one critical scenario. Scenario A4 from [3] is chosen for comparison in a single run (centralized and distributed global estimator, with assignment, no false alarms). We have logged the Matlab flops and the simulation time for the two filtering schemes. The results are listed in Table 5. Clearly, the 3-model IMM estimator has only 2 times the computation time of the Kalman filter for this scenario. This is due to the fact that the computation of coordinate conversion and data association takes a great part in the overall computation time. Another important observation is that the centralized filter is computationally *less expensive* than the distributed one.

6 Concluding Remarks

In this paper we have investigated the relationship of the steady state variances of the centralized and distributed trackers assuming the trackers use the same target kinematic model. Exact results have been derived for α-β filters corresponding to the commonly used white noise acceleration models, with various target maneuvering indices, suitable for nearly constant velocity or maneuvering targets. The performance

[6] For the low acceleration target, the jinking period is shorter.

[7] The range depends on specific platform-target pair at the end of the scenario. Certain platform has medium range to the targets in set A.

curves are presented for up to 4 sensors. As the number of sensors increases, the MSE difference between the centralized and distributed trackers increases. In the 4 sensor case, using distributed estimation with optimal track-to-track fusion yields around 10% RMS error larger than the optimal centralized approach.

References

[1] Y. Bar-Shalom and X. Li, *Estimation and Tracking: Principles, Techniques and Software*, Artech House, 1993, Reprinted by YBS Publishing, 1998.

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

[3] N. A. Carlson, *Core Set of DTA Scenarios, Technical Memorandum*, DTAS-99-05, Integrity Systems, Inc., Belmont, MA, April, 1999.

[4] N. A. Carlson, *User Manual for DTA Trajectory Generator (DTATRAJ 2.1)* and *User Manual for DTA Sensors Simulator (DTASENS 2.1)*, TR-99-01/TR-99-02, Integrity Systems, Inc., Belmont, MA, April, 1999.

[5] K. Chang, R. Saha and Y. Bar-Shalom, *On Optimal Track-to-Track Fusion*, *IEEE Trans. Aerospace and Electronic Systems*, 33(4), pp. 1271-1276, October, 1997.

[6] H. Chen, T. Kirubarajan, and Y. Bar-Shalom, *Centralized vs. Distributed Tracking Algorithms for Air to Air Scenarios*, Proc. of SPIE Conf. on Signal and Data Processing of Small Targets, Vol. 4048, April 2000.

[7] M. Yeddanapudi, Y. Bar-Shalom, and K. R. Pattipati, *IMM Estimation for Multitarget-Multisensor Air Traffic Surveillance*, Proc. of IEEE Conf. on Decision and Control, New Orleans, LA, December, 1995.

[8] H. Wang, T. Kirubarajan, and Y. Bar-Shalom, *Precision Large Scale Air Traffic Surveillance Using IMM/Assignment Estimators*, IEEE Trans. Aerospace and Electronic Systems, 35(1), pp. 255-266, January, 1999.

Filter type	Target motion model	Process noise standard deviation			
		horizontal lateral	horizontal axial	turn rate	vertical
Kalman filter	WNA-H	$3g$	$1g$	—	$1g$
IMM estimator	WNA-L	$0.05g$	$0.05g$	—	$0.05g$
	WNA-H	$3g$	$1g$	—	$1g$
	CT-L	$0.05g$	$0.05g$	$5.0°/s^2$	—

Table 1: Kalman filter and IMM estimator module design parameters

Scenario A4, Filter Conf.	Performance Criterion	Kalman filter						IMM estimator					
		F-CV	F-T	M-D	M-T	C-J	C-CV	F-CV	F-T	M-D	M-T	C-J	C-CV
Cen-T1	RMS Pos. Err.	991	679	370	258	220	181	873	575	299	201	180	132
Cen-T2	RMS Pos. Err.	987	668	382	255	238	222	838	551	317	226	215	166
Cen-T3	RMS Pos. Err.	1015	697	391	274	285	236	931	635	319	248	251	172
Dis-T1	RMS Pos. Err.	1092	748	415	287	244	207	961	636	335	227	197	149
Dis-T2	RMS Pos. Err.	1099	739	424	285	260	243	923	611	351	249	237	186
Dis-T3	RMS Pos. Err.	1108	777	431	298	314	261	1042	698	354	274	277	193
Cen-T1	RMS Vel. Err.	315	174	99	95	98	98	135	84	46	47	56	32
Cen-T2	RMS Vel. Err.	317	195	103	105	103	97	138	102	52	58	89	33
Cen-T3	RMS Vel. Err.	298	212	104	105	175	96	143	126	54	69	155	35
Dis-T1	RMS Vel. Err.	349	193	110	106	108	109	148	93	52	53	63	35
Dis-T2	RMS Vel. Err.	352	218	115	116	116	108	152	112	58	65	99	37
Dis-T3	RMS Vel. Err.	330	236	115	117	193	108	159	139	60	76	171	38

Table 3: Global estimation: Comparison of RMS errors (position in ft, velocity in ft/s) between centralized and distributed estimators using Kalman filter and IMM estimator with perfect data association, scenario A4, 100 runs.

Scenario D3	Performance Criterion	Centralized Kalman filter	Distributed Kalman filter	Centralized IMM filter	Distributed IMM filter
Global Est., Tar. 1, perfect association	RMS Pos. Err.	1042	1123	786	841
	RMS Vel. Err.	135	151	73	78
Global Est., Tar. 2, perfect association	RMS Pos. Err.	1077	1151	793	856
	RMS Vel. Err.	161	173	79	84
Global Est., Tar. 3, perfect association	RMS Pos. Err.	1003	1084	853	912
	RMS Vel. Err.	143	160	87	95
Global Est., Tar. 1, 2-D assignment	RMS Pos. Err.	1225	1368	1091	1174
	RMS Vel. Err.	176	195	98	107
Global Est., Tar. 2, 2-D assignment	RMS Pos. Err.	1297	1382	1045	1168
	RMS Vel. Err.	188	204	96	108
Global Est., Tar. 3, 2-D assignment	RMS Pos. Err.	1276	1370	1074	1187
	RMS Vel. Err.	183	198	98	108

Table 4: Global estimation: Comparison of RMS errors between Kalman filter and IMM estimator with centralized and distributed configurations for scenario D3 (with high false alarm rate), 3 platforms, 100 runs.

Configuration	Performance criteria	Centralized Kalman filter	Distributed Kalman filter	Centralized IMM estimator	Distributed IMM estimator
Without data association	Computational Time (sec)	724	1388	1587	3162
	Matlab(flops)	$5.4 \cdot 10^7$	$8.3 \cdot 10^7$	$1.12 \cdot 10^8$	$2.53 \cdot 10^8$
With data association	Computational Time (sec)	869	1607	1793	3564
	Matlab(flops)	$7.0 \cdot 10^7$	$1.03 \cdot 10^8$	$1.82 \cdot 10^8$	$2.95 \cdot 10^8$

Table 5: Comparison of computation load between the Kalman filter and the IMM estimator (centralized vs. distributed), scenario A4.