Comparison of Decentralized Tracking Algorithms

Ng Gee Wah
DSO National Laboratories
20 Science Park Drive
Singapore 118230
ngeewah@dso.org.sg

Yang Rong
DSO National Laboratories
20 Science Park Drive
Singapore 118230
yrong@dso.org.sg

Abstract – There are various algorithms in decentralized tracking. These algorithms can be categorized into three classes based on the inputs, namely track fusion, measurement fusion and information fusion. The track fusion combines individual tracks formed by different sensors. The measurement fusion generates tracks from the combined measurements detected by different sensors. The information fusion is derived from information filter performing estimation and tracking on information space. In the track fusion, the algorithms include the simple convex combination, the Bar-Shalom/Campo state vector combination, the best linear unbiased estimation (BLUE) and the covariance intersection (CI). This paper presents the comparison results of the main fusion algorithms based on the simulation data. The paper also discusses the issue on the communication cost of the fusion algorithms.

Keywords: Tracking, track fusion, measurement fusion, information fusion.

1 Introduction

The decentralized tracking system is a collection of fusion nodes worked cooperatively and independently to provide target detection and tracking through data and information sharing. Each fusion node is connected to the local sensors. Besides the advantages of more reliable and scalable, the decentralized tracking system can improve the track capability in terms of detection range and track accuracy. There are various algorithms suitable in a decentralized tracking system. These algorithms can be categorized into three classes based on the inputs, namely track fusion [1], measurement fusion [2] and information fusion [3].

In track fusion, the fusion nodes form local individual tracks using their own local detection. These individual tracks are sent to the other nodes through boardcasting or multi-casting on the network, and finally each fusion node generates the composite tracks from all the individual tracks it collected. The track fusion algorithms are widely used in decentralized architecture. This is because most of the false alarms of measurements are filtered out at the local tracking level. Thus, the network transmitting cost is reduced. In the track fusion, the algorithms include the simple convex combination (CC), the Bar-Shalom/Campo state vector combination (BC), the best linear unbiased estimation (BLUE) and the covariance intersection (CI).

In measurement fusion, all the sensors’ measurements are sent on the network directly, and the measurements associated to the same target are fused to obtain a combined measurement. Tracks are updated by the combined measurements. In measurement fusion, a large number of measurements (include false alarms) need to transmit on the network. It consumes a large amount of bandwidth. In decentralized architecture, the bandwidth requirement increases quadratically with the number of nodes. Thus, the measurement fusion is normally used in the centralized architecture, which requires less bandwidth. However, from the algorithm aspect, the measurement fusion is not limited on the centralized architecture only. In general, measurement fusion is regarded as a good algorithm, since it is possible to obtain the optimal solution. A real system in naval application showed that the measurement fusion had advantages to maintain track continuity when probability of detection was not perfect [8].

Information fusion is developed from the information filter. The information filter is called the inverse covariance form of the Kalman filter. The information

1Decentralized architecture referred to network of nodes. Each node has its own processor.
2Centralized architecture referred to all sensor data sent to a central processor.
state contribution $i$ and associated information matrix $I$ on each sensor or node, are transmitted on the network for combination.

In this paper, we compared five main fusion algorithms, namely the simple convex combination (CC), the Bar-Shalom/Campo state vector combination (BC), the covariance intersection (CI), the measurements fusion (MF) and the information fusion (IF). We also discussed some implementation issues on these algorithms. The structure of the paper is as follows. In section 2, the five decentralized tracking algorithms are reviewed. Section 3 discusses some implementation issues, which include how to select the high quality tracks for combination, how to compute $\omega$ value in the covariance intersection algorithm, and the network transmission cost for each algorithms. Section 4 presents simulation results. We conclude at section 5.

2 Decentralized tracking algorithms

In this sections, the principle of five multi-sensor tracking algorithms are reviewed.

2.1 Simple convex combination

The simple convex combination is one of the simplest algorithms for track fusion. It assumes each individual tracks are independent. The combined state $\hat{x}_{CC}$ and its covariance $P_{CC}$ are given by:

$$\hat{x}_{CC} = P_{CC}^{-1} \sum_{i=1}^{N} P_i^{-1} \hat{x}_i$$  \hspace{1cm} (1)

$$P_{CC} = (\sum_{i=1}^{N} P_i^{-1})^{-1}$$  \hspace{1cm} (2)

where $N$ is the number of individual tracks, $\hat{x}_i$ stands for the $i^{th}$ individual track state to fuse, and $P_i$ is the error covariance of $\hat{x}_i$.

2.2 Bar-Shalom/Campo state vector combination

In real world, the individual tracks are not totally independent. Bar-Shalom/Campo state vector combination algorithm was developed based on the track correlation on the same process noise [4]. The multiple tracks can be combined as follows [5]:

$$\hat{x}_{BC} = P_{BC}^{-1} \hat{X}$$  \hspace{1cm} (3)

$$P_{BC} = (\lambda P^{-1} I)^{-1}$$  \hspace{1cm} (4)

where $I$ is the identity matrix, $\lambda$ is the correlation on the same process noise [4]. The multiple combination algorithm was developed based on the track independent. Bar-Shalom/Campo state vector combination.

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1N} \\ P_{21} & P_{22} & \cdots & P_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N2} & \cdots & P_{NN} \end{bmatrix}$$  \hspace{1cm} (5)

$$I = \begin{bmatrix} I & I & \cdots & I \end{bmatrix}$$  \hspace{1cm} (6)

$$\hat{X} = [ \hat{x}_1 \ \hat{x}_2 \ \cdots \ \hat{x}_N ]$$  \hspace{1cm} (7)

and the cross-covariance between track $i$ and $j$ is:

$$P_{ij}(k|k) = [I - W_i(k)H_i(k)]F(k - 1)P_{ij}(k - 1)F'(k - 1) + Q(k - 1)|I - W_j(k)H_j(k)||$$  \hspace{1cm} (8)

where $W_i$, $W_j$ are Kalman gains from track $i$ and $j$ respectively.

2.3 Covariance intersection

The covariance intersection algorithm can be viewed as a weighted form of simple convex combination in state estimation. It is suitable for the unknown correlation among the tracks. The algorithm is given by:

$$\hat{x}_{CI} = P_{CI}^{-1} \sum_{i=1}^{N} \omega_i P_i^{-1} \hat{x}_i$$  \hspace{1cm} (9)

$$P_{CI} = (\sum_{i=1}^{N} \omega_i P_i^{-1})^{-1}$$  \hspace{1cm} (10)

$$\sum_{i=1}^{N} \omega_i = 1$$  \hspace{1cm} (11)

The $\omega$ is a number between 0 to 1. Normally, the $\omega$ is chosen by minimizing of the determinant or trace of the combined state covariance $P_{CI}$. However, the method cannot be applied to the case when individual tracks have the same error covariance $P$. This is because the minimized functions are independent of the $\omega$.

$$P_{CI} = (\sum_{i=1}^{N} \omega_i P_i^{-1})^{-1} = (\sum_{i=1}^{N} \omega_i P_i^{-1})^{-1}$$  \hspace{1cm} (12)

where $j$ is 1 to $N$. 

108
2.4 Measurement fusion

The measurement fusion algorithm is suitable for the centralized architecture. It is also possible to be used in the decentralized architecture when network transmission cost can be reduced by other technologies. In measurement fusion algorithm, the sensors’ measurements are fused to obtain a combined measurement, and then estimate the target state by the combined measurement. Here we use Kalman filter to illustrate the measurement fusion algorithm.

The tracking is modeled by following transition equation and measurement equation:

\[ x(k + 1) = F(k)x(k) + v(k) \]  
(13)

\[ z_i(k + 1) = H_i(k + 1)x(k + 1) + w_i(k + 1) \]  
(14)

where \( i \) is the sensor index, \( v(k) \) and \( w_i(k + 1) \) are white Gaussian noise with covariance \( Q \) and \( R_i \) respectively.

Assume there are \( N \) independent measurements from \( N \) sensors associated to the same target. The combined measurement \( z_{MF} \) and its error covariance \( R_{MF} \) are given by [2]:

\[ z_{MF}(k) = R_{MF}(k) \sum_{i=1}^{N} R_i^{-1}(k)z_i(k) \]  
(15)

\[ R_{MF}(k) = \left[ \sum_{i=1}^{N} R_i^{-1}(k) \right]^{-1} \]  
(16)

and the combined measurement matrix \( H_{MF} \) is:

\[ H_{MF}(k) = R_{MF}(k) \sum_{i=1}^{N} R_i^{-1}(k)H_i(k) \]  
(17)

The Kalman filter prediction and estimation by the combined measurement are shown as follows: Prediction:

\[ \hat{x}(k + 1|k) = F(k)\hat{x}(k|k) \]  
(18)

\[ P(k + 1|k) = F(k)P(k|k)F^*(k) + Q(k) \]  
(19)

Innovation covariance:

\[ S(k + 1) = H_{MF}(k + 1)P(k + 1|k)H_{MF}^*(k + 1) + R_{MF}(k + 1) \]  
(20)

Kalman gain:

\[ W(k + 1) = P(k + 1|k)H_{MF}^*(k + 1)S(k + 1)^{-1} \]  
(21)

Estimation:

\[ \hat{x}(k + 1|k + 1) = \hat{x}(k + 1|k) + W(k + 1)(z_{MF}(k + 1) - H_{MF}(k + 1)\hat{x}(k + 1|k)) \]  
(22)

\[ P(k + 1|k + 1) = [I - W(k + 1)H_{MF}(k + 1)]P(k + 1|k) \]  
(23)

2.5 Information fusion

Information fusion is developed from information filter. The information filter tracks the target by information state vector \( \hat{y} \) and information Matrix \( Y \). The relationship between traditional tracking and information tracking is given by:

\[ \hat{y}(i|j) = P^{-1}(i|j)\hat{x}(i|j) \]  
(24)

\[ Y(i|j) = P^{-1}(i|j) \]  
(25)

Assume \( N \) information filters running on \( N \) sensors. The prediction and estimation processes are [3]:

Propagation coefficient:

\[ L(k + 1|k) = P^{-1}(k + 1|k)F(k)P(k|k) \]  
(26)

where:

\[ P(k + 1|k) = F(k)P(k|k)F^*(k) + Q(k) \]  
(27)

Prediction:

\[ \hat{y}(k + 1|k) = L(k + 1|k)\hat{y}(k|k) \]  
(28)

\[ Y(k + 1|k) = [F(k)Y^{-1}(k|k)F^*(k) + Q(k)]^{-1} \]  
(29)

The information state contribution and the associated information matrix:

\[ i_j(k + 1) = H_j^*(k + 1)R_j^{-1}(k + 1)z_j(k + 1) \]  
(30)

\[ I_j(k + 1) = H_j^*(k + 1)R_j^{-1}(k + 1)H_j(k + 1) \]  
(31)

Estimation:

\[ \hat{y}(k + 1|k + 1) = \hat{y}(k + 1|k) + \sum_{j=1}^{N} i_j(k + 1) \]  
(32)

\[ Y(k + 1|k + 1) = Y(k + 1|k) + \sum_{j=1}^{N} I_j(k + 1) \]  
(33)

where \( j \) is the index of individual information filters. The data transmitted on the network are the information state contribution \( i \) and the associated information matrix \( I \).

3 Implementation issues

In order to make unbiased comparison, Kalman filter is applied to all the local tracks in track fusion, and all local kalman filters use the same process noise covariance \( Q \) and the same measurement noise covariance \( R \). This implementation, results in all the local tracks generate the same state error covariance \( P \), after the Kalman filter is stable. This means that a set of individual tracks with identical error covariance are going to be fused together in track fusion. There are two issues when we fuse these tracks.
• The first issue is: In the simple convex combination and the Bar-Shalom/Campo state vector combination, the combined state vector becomes the average of all the individual state vectors. It results in that good and bad tracks (based on track quality) have equal contribution to the combined track. Thus, it is necessary to develop a track selection algorithm to eliminate tracks with poor track quality.

• The second issue is: The computation of $\omega$ in the covariance intersection algorithm. The value of $\omega$ cannot be decided through general methods when all the individual tracks have the same $P$ (see subsection 2.3). $\omega$ selection has to be based on other features.

We proposed some solutions to the two problems in our algorithm implementation. Another issue on decentralized tracking is the communication cost. The communication cost issue is discussed in the subsection 3.3.

3.1 Track selection

The purpose of the track selection is to choose good quality tracks to fuse. The track quality can be computed based on the normalized distance function [6]:

$$d^2(k+1) = v'(k+1)S(k+1)^{-1}v(k+1)$$

where the measurement residual $v(k+1)$ is:

$$v(k+1) = z(k+1) - H(k+1)F(k)x(k|k)$$

and the innovation covariance $S(k+1)$ is:

$$S(k+1) = H(k+1)P(k+1|k)H'(k+1) + R(k)$$

Then, the track quality $U(k+1)$ is computed by:

$$U(k+1) = \alpha U(k) + (1 - \alpha)d^2(k+1)$$

where $\alpha$ is the weight on the history. $\alpha$ can be chosen from 0 to 1. The value of $U$ represents the track quality. The smaller the $U$, the better the quality.

Next step is to eliminate those tracks whose qualities are relatively poor. The smallest $U_{\text{min}}$ (the best quality) is selected among the individual tracks, and the relative quality $A_i$ of the track $i$ is computed by:

$$A_i(k+1) = \frac{U_i(k+1)}{U_{\text{min}}(k+1)}$$

Only tracks that meet the following condition are selected to fuse:

$$A_i(k+1) < T_a$$

where $T_a$ is the threshold. $T_a$ is a number greater than 1.

3.2 $\omega$ computation in CI

The covariance intersection algorithm can be viewed as a weighted simple convex combination. If we assume that the detection accuracy of the sensor improves when the target is nearer to the sensor, then, one approximation approach for computing the $\omega$ can be derived, that is: the $\omega$ is inversely proportional to the distance between the target and the sensor. This is shown as follows:

$$\omega_i = \frac{1}{d_i}$$

where $d_i$ is the distance from the sensor to the target in the track $i$.

3.3 Communication cost

Communication cost is an important issue in decentralized tracking system. In decentralized architecture, the communication cost is naturally higher compared to the centralized architecture. Therefore, the fusion algorithm with less communication cost will be preferred.

Assume that the size of state vector is 4, and the size of measurement vector is 2. All numbers are defined as real, which need 8 bytes space. Table 1 shows the data and the number of bytes required to be sent in the decentralized architecture.

<table>
<thead>
<tr>
<th>Algo.</th>
<th>Unit</th>
<th>Data</th>
<th>Type</th>
<th>Byte</th>
<th>Total (byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>per track</td>
<td>x</td>
<td>4 real</td>
<td>16 real</td>
<td>32</td>
</tr>
<tr>
<td>BC</td>
<td>per track</td>
<td>P</td>
<td>4 real</td>
<td>16 real</td>
<td>32</td>
</tr>
<tr>
<td>CI</td>
<td>per track</td>
<td>x</td>
<td>4 real</td>
<td>8 real</td>
<td>32</td>
</tr>
<tr>
<td>MF</td>
<td>per measurement</td>
<td>z</td>
<td>2 real</td>
<td>4 real</td>
<td>16</td>
</tr>
<tr>
<td>IF</td>
<td>per track</td>
<td>i</td>
<td>4 real</td>
<td>16 real</td>
<td>32</td>
</tr>
</tbody>
</table>

The measurement fusion shows less communication cost in per measurement. However, the sensors measurements include large amounts of false alarms in the real applications. The number of measurements needed to be transmitted across the network could be much bigger than the number of tracks. The overall communication cost of the measurement fusion are still higher than the others.
4 Simulation results

Experimental results based on 2D simulation are obtained in this section. The results show the RMS position errors of the local individual tracks and fused tracks.

4.1 Scenario Generation

The scenario simulates the 2D radar with detection of range and azimuth. There are eight nodes in the scenario, each node is connected to a radar. The target and the nodes deployment are shown on Figure 1.

The target moves from northwest to the southeast, and then makes a right turn, and moves from north to south, and finally moves from northwest to southeast again after a left turn. The target speed is 100m/s, and turning rate is 5°/s. The radar measurement noises are simulated by zero mean Gaussian distributions. The noise of range is with standard deviation of 100m, and the azimuth’s noise is with standard deviation of 2°.

4.2 Test result

In order to track target position in x-y space. The measurements are converted to Cartesian coordinates unbiasedly using [7]:

\[ x_m = \lambda_b^{-1} r_m \cos b_m \] (41)
\[ y_m = \lambda_b^{-1} r_m \sin b_m \] (42)
\[ \lambda_b = e^{-\sigma_b^2/2} \] (43)

where \( \lambda_b \) is the bias compensation factor, \( r_m \) is the measured range, \( b_m \) is the measured bearing, and \( \sigma_b^2 \) is the error variance of the measured bearing, which is set to 4 in our scenario.

The state vector \( \mathbf{x} \) in the simulation is defined as \( [x\ y\ \dot{x}\ \dot{y}]^T \), and measurement vector \( \mathbf{z} \) is \( [x\ y]^T \). The state transition equation and the measurement equation are given by:

\[
\begin{bmatrix}
1 & 0 & T & 0 \\
0 & 1 & 0 & T \\
0 & 0 & 1 & 0 \\
T^2/2 & 0 & 0 & T^2/2 \\
0 & T & 0 & T
\end{bmatrix}
\begin{bmatrix}
x(k) \\
y(k) \\
\dot{x}(k) \\
\dot{y}(k)
\end{bmatrix}
+ \begin{bmatrix}
0 \\
0 \\
0 \\
-2T^2/2 \\
T
\end{bmatrix}
\begin{bmatrix}
a(k)
\end{bmatrix}
\] (44)

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x(k) \\
y(k)
\end{bmatrix}
+ w(k)
\] (45)

where \( a(k) = [\dot{x} \ \dot{y}]^T \) at time \( k \). It is the process noise used to handle uncertain acceleration in the constant velocity transition model. In our simulation, \( a(k) \) is set to \( [10 10]^T \). \( w(k) \) is the measurement noise, and its covariance \( R \) is set to 160000.

Four simulation tests were carried out. These tests are:

1) Only two nodes are active. they are N1 and N8.
2) Four nodes are active, they are N1, N3, N6 and N8.
3) Six nodes are active, they are N1, N2, N3, N6, N7 and N8.
4) Eight nodes are all active.

Figure 2 shows the RMS position errors vs. time. It is quite obvious that multiple sensor fusion algorithms have the less RMS errors than the local individual Kalman filters.

Table 2 shows the RMS position errors for five decentralized tracking algorithms. The results were obtained from 100 runs. The covariance intersection algorithm outperforms the other fusion algorithms in these particular experiments. The covariance intersection \( \omega \) is computed using the assumption as stated in subsection 3.2.

We also test our track selection method on the simple convex combination algorithm from 100 runs. The test result is shown on Table 3. We can see that the RMS position errors are further reduced by the track selection algorithm.

5 Conclusions

This paper presents the simulation results on the comparison of five main decentralized tracking algorithms, namely, the simple convex combination, the Bar-shalom/ Campo state vector combination, the intersection covariance algorithm, the measurement fusion and the information fusion. The covariance inter-
Table 2: Decentralized tracking algorithms comparison

<table>
<thead>
<tr>
<th>Node</th>
<th>Algo.</th>
<th>RMS error (m)</th>
<th>Noise reduction(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 nodes</td>
<td>-</td>
<td>654.78</td>
<td>Noise</td>
</tr>
<tr>
<td>CC</td>
<td>193.92</td>
<td>70%</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>193.92</td>
<td>70%</td>
<td></td>
</tr>
<tr>
<td>CI</td>
<td>122.95</td>
<td>81%</td>
<td></td>
</tr>
<tr>
<td>IF</td>
<td>202.55</td>
<td>69%</td>
<td></td>
</tr>
<tr>
<td>MF</td>
<td>202.30</td>
<td>69%</td>
<td></td>
</tr>
<tr>
<td>4 nodes</td>
<td>-</td>
<td>660.93</td>
<td>Noise</td>
</tr>
<tr>
<td>CC</td>
<td>147.08</td>
<td>78%</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>147.08</td>
<td>78%</td>
<td></td>
</tr>
<tr>
<td>CI</td>
<td>106.63</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>IF</td>
<td>157.75</td>
<td>76%</td>
<td></td>
</tr>
<tr>
<td>MF</td>
<td>157.80</td>
<td>76%</td>
<td></td>
</tr>
<tr>
<td>6 nodes</td>
<td>-</td>
<td>623.32</td>
<td>Noise</td>
</tr>
<tr>
<td>CC</td>
<td>119.64</td>
<td>81%</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>119.64</td>
<td>81%</td>
<td></td>
</tr>
<tr>
<td>CI</td>
<td>87.38</td>
<td>86%</td>
<td></td>
</tr>
<tr>
<td>IF</td>
<td>128.06</td>
<td>79%</td>
<td></td>
</tr>
<tr>
<td>MF</td>
<td>128.24</td>
<td>79%</td>
<td></td>
</tr>
<tr>
<td>8 nodes</td>
<td>-</td>
<td>605.45</td>
<td>Noise</td>
</tr>
<tr>
<td>CC</td>
<td>105.57</td>
<td>83%</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>105.57</td>
<td>83%</td>
<td></td>
</tr>
<tr>
<td>CI</td>
<td>82.79</td>
<td>86%</td>
<td></td>
</tr>
<tr>
<td>IF</td>
<td>111.51</td>
<td>82%</td>
<td></td>
</tr>
<tr>
<td>MF</td>
<td>111.52</td>
<td>82%</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Track selection method test result

<table>
<thead>
<tr>
<th>Node</th>
<th>Track selection (Yes/No)</th>
<th>RMS error (m)</th>
<th>Noise reduction(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 nodes</td>
<td>-</td>
<td>654.87</td>
<td>Noise</td>
</tr>
<tr>
<td>No</td>
<td>193.92</td>
<td>70%</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>141.99</td>
<td>78%</td>
<td></td>
</tr>
<tr>
<td>4 nodes</td>
<td>-</td>
<td>660.93</td>
<td>Noise</td>
</tr>
<tr>
<td>No</td>
<td>147.08</td>
<td>78%</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>122.86</td>
<td>81%</td>
<td></td>
</tr>
<tr>
<td>6 nodes</td>
<td>-</td>
<td>623.32</td>
<td>Noise</td>
</tr>
<tr>
<td>No</td>
<td>119.64</td>
<td>81%</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>101.01</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>8 nodes</td>
<td>-</td>
<td>605.45</td>
<td>Noise</td>
</tr>
<tr>
<td>No</td>
<td>105.57</td>
<td>83%</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>97.65</td>
<td>84%</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: RMS position errors
section algorithm performs less RMS error when reasonable \( \omega \) values are chosen. The paper proposes a simple approximation approach to compute the \( \omega \) based on the distance from the sensor to the target. The simulation results also shows that multiple sensor fusion algorithms have the less RMS error than the local Kalman filters, and the RMS error is reducing when increasing the number of nodes. However, the RMS error cannot be reduced infinitely by increasing the number of nodes. The saturating state is not investigated in our simulation.

The paper discusses the issue of selecting tracks for combination, and presents a track selection method based on the track quality. Simulation results show the method could further reduce the RMS error.

Communication cost is shown for each decentralized tracking algorithm.

References


