Tracking/Fusion and Deghosting with Doppler Frequency from Two Passive Acoustic Sensors

Rong Yang, Gee Wah Ng
DSO National Laboratories
20 Science Park Drive
Singapore 118230
Emails: yrong@dso.org.sg, ngeewah@dso.org.sg

Yaakov Bar-Shalom
Department of ECE
University of Connecticut
Storrs, CT 06269, USA
Email: ybs@engr.uconn.edu

Abstract—It is known that using two passive sensors that provide bearings-only information cannot eliminate false intersections in a multi-target environment, and results in ghost tracks. In this paper, we propose a tracking/fusion and deghosting method using two stationary acoustic sensors. The existing approaches for bearings-only information require the number of sensors to be greater than or equal to three. The method proposed in this paper makes target tracking possible in a two-sensor system through making use of frequency information. This method generates tentative tracks from all possible bearing track combinations, and distinguishes target tracks from ghost tracks using a 2-D assignment algorithm. Tentative tracks are initiated using an iterated Least Squares (LS) algorithm, and are updated by an Unscented Kalman Filter (UKF) based on Doppler-Bearing Tracking (DBT) approach. The likelihood ratios of tentative tracks, which are computed based on track innovation probability density functions, are assigned as costs in the 2-D assignment problem. Simulation tests have been conducted to validate the effectiveness of the proposed method.

Keywords—Deghosting, passive tracking, Doppler-bearing tracking

I. INTRODUCTION

Passive tracking is a challenging problem, as target range cannot be obtained from sensor measurements directly. Target position has to be computed from some special methods. One approach is the so called Target Motion Analysis (TMA), which requires a sensor deployed on a maneuvering platform. The target trajectory is then estimated using a Bearings-Only Tracking (BOT) algorithm [8] [1]. The BOT was further developed to the Doppler-Bearing Tracking (DBT) algorithm in narrowband sonar applications [9] [7] [4]. The DBT algorithm tracks target position and emitted frequency from bearings and Doppler shifted frequencies, and a target can be localized even when the platform is not maneuvering. Another approach to obtain target positions is through triangulation of synchronous bearings detected by multiple sensors [10] [5] [12] [11] [6]. The triangulation technique, which generates “composite measurements” (as part of configuration III fusion) [3], needs more than one passive sensor working collaboratively. The present paper focuses on the triangulation technique.

The main challenge of the triangulation approach is deghosting. In a multi-target scenario, triangulation points from bearing lines consist of target points and ghost points, and the target points need to be distinguished from ghost points through a deghosting algorithm. One conventional deghosting method is through comparing hinge angles [11] [6]. A hinge angle is the angle between the reference plane and the target plane. The reference plane is formed by the locations of two sensors and a predefined reference point, and the target plane is constructed by two sensors and an angular vector (bearing and elevation) detected by a sensor. Since hinge angle computation needs measured elevation, this method cannot work on bearing-only measurements. A more robust approach is to consider the problem as an S-D assignment problem [10] [5]. The approach matches the bearing measurements from all the sensors, and computes the cost of each combination by the logarithm of likelihood ratio. However, this approach requires the number of sensors, \( S \), to be greater than or equal to three. This is because ghosts and targets cannot be distinguished from two-sensor triangulation. The constraint makes S-D assignment problem \((S \geq 3)\) to be an NP-hard problem. A Lagrangian relaxation was suggested to solve the problem by a series of 2-D assignment problems. To avoid the NP-hard problem, feature-aided deghosting method has been considered. A deghosting method using bearings and acoustic signals was proposed in [12]. This method can work in a two-sensor system. A simple hard condition was applied to likelihoods of acoustic signals for deghosting.

In this paper, we develop a featured-aided deghosting approach. Frequency information is utilized in tracking/fusion and deghosting in a two-stationary-sensor system. The overall tracking system architecture is shown in Fig. 1. It consists of two bearing trackers, a tentative target tracker, a deghosting component and a target track management component. A brief description for each component is given in the following:

- The bearing trackers track bearings and Doppler shifted frequencies. A Kalman filter is used to obtain Bearing Tracks (BT) consisting of estimated bearings, Doppler shifted frequencies, and their measured values. To avoid common measurements in different BTs, a 2-D assignment algorithm is applied in the track to measurement association.

- The tentative target tracker fuses the BT information to generate and maintain tentative target tracks based on all possible BT combinations. The tracking problem is formulated based on the DBT approach. Tracks are initiated by an iterated LS algorithm, and are updated
The notations used in the paper are listed in the following:

\[ x, y \] Target position in \( x \) and \( y \) coordinates;
\[ \dot{x}, \dot{y} \] Target speed in \( x \) and \( y \) coordinates;
\[ x_i^e, y_i^e \] Position of sensor \( i \), and \( i \in \{1, 2\} \);
\[ x_i^e = x - x_i^e \] Target position relative to sensor \( i \) in \( x \) coordinate;
\[ y_i^e = y - y_i^e \] Target position relative to sensor \( i \) in \( y \) coordinate;
\[ r_i \] Target range to sensor \( i \);
\[ \dot{r}_i \] Target radial velocity to sensor \( i \);
\[ f^e \] Target emitted frequency;
\[ T \] Time interval;
\[ s \] Sound speed in the air (343 m/s).

The deghosting component selects target tracks from all tentative tracks. The problem is formulated as a 2-D assignment problem based on the assumption that a BT can only contribute to one target. The cost of each tentative track is the negative log Likelihood Ratio (LR) of the track.

The target track management function creates and maintains target tracks from the selected/deghosted tentative tracks. Note that tracking/estimation is not performed in this component, since it has been done in the tentative target tracker. The target track state is updated from the tentative track directly. The management logic used here includes: a target track is created if a new tentative track has been selected in the current and previous time cycles continuously in a sliding window; a target track is updated by its tentative track if the tentative track is selected at the current time; a target track is predicted if its tentative track is not selected or deleted; a target track is deleted if its tentative track is not updated for a certain period of time.

In this paper, we focus on the tentative target tracker and the deghosting. For simplicity, we track in 2-D and assume that the propagation delays are the same for all the sensors. The notations used in the paper are listed in the following:

\[ x, y \] Target position in \( x \) and \( y \) coordinates;
\[ \dot{x}, \dot{y} \] Target speed in \( x \) and \( y \) coordinates;
\[ x_i^e, y_i^e \] Position of sensor \( i \), and \( i \in \{1, 2\} \);
\[ x_i^e = x - x_i^e \] Target position relative to sensor \( i \) in \( x \) coordinate;
\[ y_i^e = y - y_i^e \] Target position relative to sensor \( i \) in \( y \) coordinate;
\[ r_i \] Target range to sensor \( i \);
\[ \dot{r}_i \] Target radial velocity to sensor \( i \);
\[ f^e \] Target emitted frequency;
\[ T \] Time interval;
\[ s \] Sound speed in the air (343 m/s).

The structure of the rest of the paper is as follows. Section II describes the tentative target tracker in detail. It includes an iterated LS track initiation and a DBT algorithm. Section III formulates the deghosting problem as a 2-D assignment problem, and describes the costs of tentative tracks. Simulation results and conclusions are in Section IV and Section V, respectively.

II. TENTATIVE TARGET TRACKING

Tentative target tracking is performed after bearing tracking in each time cycle. First of all, BT pairs are formed from all the combinations of BTs. A BT pair consists of two BTs, which are from two different sensors. A validation process is carried out to filter out those pairs which cannot triangulate or whose triangulation points are out of detection range. Tentative target tracks are then updated by the validated BT pairs. The following subsections describe the state estimation and the track initiation of tentative tracks.

A. Tentative target track state estimation

The tentative target track estimation is formulated based on the DBT approach. The state vector is defined as

\[
\mathbf{x}(k) = \begin{bmatrix} x(k) & \dot{x}(k) & y(k) & \dot{y}(k) & f^e(k) \end{bmatrix}^T
\]  
(1)

where \( k \) is the time index. The measurement vector is

\[
\mathbf{z}(k) = \begin{bmatrix} b_{1}^{m}(k) & f_{1}^{m}(k) & b_{2}^{m}(k) & f_{2}^{m}(k) \end{bmatrix}^T
\]  
(2)

where \( b_{1}^{m}(k) \) and \( f_{1}^{m}(k) \) are measured bearing and Doppler frequency in BT \( i_1 \) from sensor 1, \( b_{2}^{m}(k) \) and \( f_{2}^{m}(k) \) are measured bearing and Doppler frequency in BT \( i_2 \) from sensor 2, and BTs \( i_1 \) and \( i_2 \) belong to a valid BT pair. Note that the measured bearings and frequencies (not their estimated values) attached to BTs are used to perform state estimation. This is because the estimated bearings and frequencies have errors that are correlated over time, and using them as measurements conflicts with the assumption of Kalman-based filter on measurements.

The state transition is modified from the White Noise Acceleration (WNA) model [2] to include the emitted frequency

\[
\mathbf{x}(k) = F \mathbf{x}(k-1) + \Gamma v(k-1)
\]  
(3)

where \( v \) is white Gaussian process noise with covariance \( Q \), and

\[
F = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\]  
(4)
The state at time \( k \) can be iterated starting from an initial state and its error covariance. The iterated Kalman filter is selected to perform state estimation.

**B. Tentative track initiation**

The recursive state estimation described in section II-A requires an initial state and its error covariance. The measurement model should be adjusted to handle situations where both sensors have measurements. If only one sensor has detection, the measurement model should be adjusted to:

\[
Z = h_x x + w
\]

where \( w \) is white Gaussian measurement noise with covariance \( R = \text{diag}(\sigma_{m}^{2}, \sigma_{f}^{2}) \) and \( \sigma_{m} \) and \( \sigma_{f} \) are the standard deviation of measurement errors on bearing and frequency respectively, and \( h(\cdot) \) is given by:

\[
\begin{align*}
    h_1 &= \arctan\left(\frac{x_1'(k)}{y_1'(k)}\right) \\
    h_2 &= 1 - \frac{\hat{r}_1(k)}{s} \cdot f^e(k) \\
    h_3 &= \arctan\left(\frac{x_2'(k)}{y_2'(k)}\right) \\
    h_4 &= 1 - \frac{\hat{r}_2(k)}{s} \cdot f^e(k)
\end{align*}
\]

with

\[
\begin{align*}
    \hat{r}_1(k) &= \frac{\dot{x}(k)x_1'(k) + \dot{y}(k)y_1'(k)}{r_1(k)} \\
    \hat{r}_2(k) &= \frac{\dot{x}(k)x_2'(k) + \dot{y}(k)y_2'(k)}{r_2(k)}
\end{align*}
\]

Note that the measurement model mentioned above is for when both sensors have measurements. If only one sensor has detection, the measurement model should be adjusted to:

\[
h(\cdot) = [ h_1 \ h_2 ]^T
\]

or

\[
h(\cdot) = [ h_3 \ h_4 ]^T
\]

with

\[
R = \text{diag}(\sigma_{m}^{2}, \sigma_{f}^{2})
\]

Since the measurement model is nonlinear, the unscented Kalman filter is selected to perform state estimation.

The Jacobian of \( h(n) \) is defined as:

\[
H_n = \frac{\partial h(n)}{\partial x(n)} = \begin{bmatrix} \frac{\partial h_1}{\partial x(n)} & \frac{\partial h_1}{\partial \dot{x}} & \frac{\partial h_1}{\partial y(n)} & \frac{\partial h_1}{\partial \dot{y}} & \frac{\partial h_1}{\partial f^e} \\
\frac{\partial h_2}{\partial x(n)} & \frac{\partial h_2}{\partial \dot{x}} & \frac{\partial h_2}{\partial y(n)} & \frac{\partial h_2}{\partial \dot{y}} & \frac{\partial h_2}{\partial f^e} \\
\frac{\partial h_3}{\partial x(n)} & \frac{\partial h_3}{\partial \dot{x}} & \frac{\partial h_3}{\partial y(n)} & \frac{\partial h_3}{\partial \dot{y}} & \frac{\partial h_3}{\partial f^e} \\
\frac{\partial h_4}{\partial x(n)} & \frac{\partial h_4}{\partial \dot{x}} & \frac{\partial h_4}{\partial y(n)} & \frac{\partial h_4}{\partial \dot{y}} & \frac{\partial h_4}{\partial f^e} \end{bmatrix}
\]

where

\[
\begin{align*}
    \frac{\partial h_1}{\partial x(n)} &= \frac{y_1'}{(r_1)^2} \\
    \frac{\partial h_1}{\partial \dot{x}} &= -(n-k)T \frac{y_1'}{(r_1)^2} \\
    \frac{\partial h_1}{\partial y(n)} &= -x_1' \frac{y_1'}{(r_1)^2} \\
    \frac{\partial h_1}{\partial \dot{y}} &= -y_1' \frac{(n-k)T x_1'}{(r_1)^2} \\
    \frac{\partial h_1}{\partial f^e} &= 0 \\
    \frac{\partial h_2}{\partial x(n)} &= 0 \\
    \frac{\partial h_2}{\partial \dot{x}} &= \frac{f^e y_1'}{s(r_1)^3} (y_1' \dot{x} - x_1' \dot{y}) \\
    \frac{\partial h_2}{\partial y(n)} &= -\frac{f^e x_1'}{s(r_1)^3} \\
    \frac{\partial h_2}{\partial \dot{y}} &= +\frac{(n-k)T f^e y_1'}{s(r_1)^3} (y_1' \dot{x} - x_1' \dot{y}) \\
\end{align*}
\]

The measurement model is:

\[
z(k) = h(x(k)) + w(k)
\]

where \( w \) is white Gaussian measurement noise with covariance \( R = \text{diag}(\sigma_{m}^{2}, \sigma_{f}^{2}) \). The function \( h(n)(\cdot) \) can be obtained by the following two steps:

**Step 1:** \( x[k, x(n)] \) computes the state at time \( k \) from \( x(n) \), where \( k \in \{1, \cdots, n\} \), is the time index. It is based on the assumptions that the target is moving at a constant velocity \((\dot{x}, \dot{y})\) with fixed emitted frequency \( f^e \), and is given by:

\[
\begin{align*}
x(k) &= x(n) - (n-k)\dot{x} - x_1' \\
\dot{x}(k) &= \dot{x}(n) - \dot{x} \\
y(k) &= y(n) - (n-k)\dot{y} - y_1' \\
\dot{y}(k) &= \dot{y}(n) - \dot{y} \\
f^e(k) &= f^e(n) = f^e
\end{align*}
\]

**Step 2:** \( z[k, x(k)] \) computes \( z \) at time \( k \) from \( x(k) \). It can be obtained from the measurement model described in (8) – (11). \( Z_n \) is then built from \( z \) according to (17).
The problem can be formulated as a 2-D assignment problem, that is, to minimize a cost function with a set of constraints. The cost function [10] is defined as

\[ J = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} c_{i_1i_2} \rho_{i_1i_2} \]

and the constraints are

\[ \sum_{i=1}^{n_1} \rho_{i_1i_2} = 1 \quad \text{for all } i_2 = 1, 2, \ldots, n_2 \]

\[ \sum_{i_1=1}^{n_2} \rho_{i_1i_2} = 1 \quad \text{for all } i_1 = 1, 2, \ldots, n_1 \]

where \( n_1 \) and \( n_2 \) are the numbers of bearing tracks at sensor 1 and sensor 2 respectively, \( c_{i_1i_2} \) is the cost of the tentative track formed by BT \( i_1 \) from sensor 1 and BT \( i_2 \) from sensor 2, and \( \rho_{i_1i_2} \) is binary variable defined as

\[ \rho_{i_1i_2} = \begin{cases} 1 & \text{if the tentative track is a target track} \\ 0 & \text{if the tentative track is a ghost track} \end{cases} \]

If \( i_1i_2 \) is an invalid pair, infinity is assigned to \( c_{i_1i_2} \).

The 2-D assignment problem can be solved in quasi-polynomial time using some existing algorithms, such as Aucition or Joncker-Volgenant-Castanon (JVC) algorithm. The main focus here is to compute the track cost \( c_{i_1i_2} \), which is the cumulative negative log likelihood ratio given by

\[ c_{i_1i_2}(k) = c_{i_1i_2}(k-1) - \ln(\Lambda) \]

where \( \Lambda \) is the likelihood ratio given by [3]

\[ \Lambda = \frac{P_D}{1 - P_D} \left( \frac{\nu}{\lambda_c} \right) \]

where \( P_D \) is the probability of detection, \( \lambda_c \) is the spatial density of the extraneous measurements, \( \nu \) and \( S \) are the innovation and its covariance with two-detection update respectively (namely, both sensors have measurements), and \( \nu_1 \) and \( S_1 \) are the innovation and its covariance with one-detection update respectively (namely, only one sensor has detection).

Once the costs of all tentative tracks are computed, the target tracks are selected through solving the 2-D assignment problem. The selected tracks are then used to update target tracks or to form new target tracks.

IV. SIMULATIONS

Two scenarios shown in Fig. 3 and Fig. 4 are used to verify the proposed tracking/fusion and deghosting algorithms. The scenarios simulate helicopters being detected by a two-acoustic-sensor system. The helicopters emit constant frequencies of 20Hz from the main rotors. The emitted frequency is used in simulation only, and it is unknown to the detection and tracking system. There are three targets in each scenario. Scenario 1 consists of a stationary target and two targets moving with different velocities. The three targets in scenario 2 have the same velocity. Scenario 2 is a difficult scenario, as the target and ghost tracks have similar motion and Doppler shift, and it is difficult to distinguish them.

In the simulation tests, the sampling interval of the two synchronized sensors is \( T = 1s \). The frequency error is Gaussian with standard deviation \( \sigma_{f_m} = 0.1Hz \). The Gaussian bearing errors are set to three levels, namely \( \sigma_{\nu} \in \{0.1^\circ, 0.3^\circ, 0.5^\circ\} \). The process noise covariance \( (3 \times 3) \) is

\[ Q = \text{diag}(1, 1, 0.01) \]
The false alarm rate for each sensor is set to 2 per second over 180°. The tentative track initial length \( n \) is set to 4. The memory coefficient is set to 0.7.

### TABLE I. DEHOSTING AND TRACKING PERFORMANCE IN SCENARIO 1

<table>
<thead>
<tr>
<th>( \sigma_{f,m} ) (°)</th>
<th>( P_D )</th>
<th>num. truth (s)</th>
<th>dur. (s)</th>
<th>dur. (%)</th>
<th>trk. brk. (s)</th>
<th>num. dur. (s)</th>
<th>dur. (s)</th>
<th>dur. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.0</td>
<td>188</td>
<td>3.1</td>
<td>183.6</td>
<td>97.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.8</td>
<td>188</td>
<td>3.3</td>
<td>182.1</td>
<td>96.8</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.3</td>
<td>188</td>
<td>3.7</td>
<td>179.8</td>
<td>95.6</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.5</td>
<td>188</td>
<td>3.3</td>
<td>183.6</td>
<td>97.6</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.8</td>
<td>188</td>
<td>3.4</td>
<td>182.9</td>
<td>97.3</td>
<td>0.4</td>
<td>0.0</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>0.5</td>
<td>188</td>
<td>3.5</td>
<td>183.5</td>
<td>97.6</td>
<td>0.5</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>0.8</td>
<td>188</td>
<td>3.8</td>
<td>179.8</td>
<td>95.7</td>
<td>0.8</td>
<td>0.1</td>
<td>1.1</td>
<td>0.6</td>
</tr>
<tr>
<td>average</td>
<td>188</td>
<td>3.4</td>
<td>182.4</td>
<td>97.0</td>
<td>0.4</td>
<td>0.0</td>
<td>0.2</td>
<td>0.1</td>
</tr>
</tbody>
</table>

The simulation tests show the performance obtained from 100 Monte Carlo runs. The performance metrics listed below are obtained from average of 100 runs:

- target/false track num.: average of total number of target (or false) tracks per run;
- target/false track dur.(s): average of total duration of target (or false) tracks in seconds per run;

### Table II. DEHOSTING AND TRACKING PERFORMANCE IN SCENARIO 2

<table>
<thead>
<tr>
<th>( \sigma_{f,m} ) (°)</th>
<th>( P_D )</th>
<th>num. truth (s)</th>
<th>dur. (s)</th>
<th>dur. (%)</th>
<th>trk. brk. (s)</th>
<th>num. dur. (s)</th>
<th>dur. (s)</th>
<th>dur. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.0</td>
<td>188</td>
<td>3.8</td>
<td>174.9</td>
<td>93.0</td>
<td>0.8</td>
<td>2.5</td>
<td>23.0</td>
</tr>
<tr>
<td>0.8</td>
<td>188</td>
<td>4.1</td>
<td>173.6</td>
<td>92.4</td>
<td>1.1</td>
<td>0.0</td>
<td>3.0</td>
<td>28.9</td>
</tr>
<tr>
<td>0.5</td>
<td>188</td>
<td>4.2</td>
<td>172.0</td>
<td>91.5</td>
<td>1.2</td>
<td>0.0</td>
<td>3.0</td>
<td>31.1</td>
</tr>
<tr>
<td>0.3</td>
<td>188</td>
<td>4.8</td>
<td>167.9</td>
<td>89.3</td>
<td>1.8</td>
<td>0.0</td>
<td>4.3</td>
<td>47.8</td>
</tr>
<tr>
<td>0.5</td>
<td>188</td>
<td>4.9</td>
<td>167.3</td>
<td>89.0</td>
<td>1.9</td>
<td>0.0</td>
<td>4.1</td>
<td>50.4</td>
</tr>
<tr>
<td>0.8</td>
<td>188</td>
<td>5.2</td>
<td>164.9</td>
<td>87.7</td>
<td>2.2</td>
<td>0.0</td>
<td>4.3</td>
<td>55.9</td>
</tr>
<tr>
<td>average</td>
<td>188</td>
<td>5.1</td>
<td>165.6</td>
<td>88.1</td>
<td>2.1</td>
<td>0.0</td>
<td>4.2</td>
<td>50.9</td>
</tr>
</tbody>
</table>

The results of scenario 1 and scenario 2 are shown in Table I and Table II, respectively. We use the first case in Table I, which is tested under \( \sigma_{f,m} = 0.1^° \) and \( P_D = 1.0 \), to further illustrate the performance metrics. In this case, we obtained 310 tracks over the whole duration of the scenario from 100 runs, and all of them are target tracks. The average number of target tracks per run is then 3.1, and the average number of false tracks per run is 0.0. There are total 10 track breaks in the 100 runs, so the average target track breaks per run is then 0.1. The percentage of target/false track duration are also displayed in Fig. 5 and Fig. 6 for scenario 1 and scenario 2, respectively.

- target/false track dur.(%): average ratio of the total duration of target (or false) tracks to the total duration of ground truth per run;
- target track brk.: average number of breaks in target tracks per run.

From the results, we can observe that the targets are well tracked in scenario 1. The targets are almost fully covered by tracks. The target track duration is from 95.7% to 97.7%. The target track continuity is very good as the number of track breaks is less than 1. The number of false tracks is close to 0. We can say that the proposed algorithm has very good performance in scenario 1.
However, the performance is not that good in scenario 2. The average target track duration decreases to 88.1%, and the false track duration increases to 27.1%. The average number of track breaks also increases to 2.1. The performance becomes worse when the bearing error increases and $P_D$ decreases. This is because the tentative tracks in this scenario have similar velocities and Doppler shifts. It results in similar innovation pdfs, and leads to similar assignment costs in deghosting. However, the differences among the innovation pdfs become larger when targets get closer to the sensors, and the algorithm yields better deghosting performance. To validate this, the durations of target and false tracks for the last 36 time cycles are computed from 100 Monte Carlo runs. The results are shown in Fig. 7. It can be seen the performance is improved significantly. The targets are well tracked with very small amount of false tracks.

To further illustrate the results intuitively, Figs. 8–11 show target tracking screen shots at time 8, 28, 33 and 62 in one run of scenario 2. The results are obtained with the following parameter settings: $P_D = 1$, false alarm rate is 2 per second, $\sigma_{p_m} = 0.5^\circ$, and $\sigma_{f_m} = 0.1$Hz. We can see that three false tracks are generated at time 8. Deghosting performance is poor at this range. At time 28, the three targets at range around 5km can be tracked by tracks 6, 8 and 9. Four false tracks exist due to wrong deghosting before. At time 33, the false tracks are all dropped, and the three targets tracks are well maintained until end of the simulation run.

From the results, we can conclude that the proposed deghosting and tracking algorithms can track targets very well when tentative tracks have large differences in innovation pdfs, which leads to large differences in deghosting assignment costs. False tracks and track loss are seldom observed in this
case. However, when scenarios have similar assignment costs, the tracking and deghosting performance drops, especially when bearing error is large and $P_D$ is low.

V. Conclusions

In this paper, we proposed a frequency-aided passive tracking approach using two acoustic sensors. The approach fuses bearing tracks from two sensors to generate tentative tracks, and obtains target tracks through deghosting. The tentative tracks are initiated by the iterated LS algorithm and tracked by UKF based on the DBT method. The deghosting is formulated as a 2-D assignment problem with costs computed using the track innovation pdf. Simulation tests have been conducted, and the results show that targets can be tracked very well when the assignment costs are different. However, tracking performance drops when the assignment costs are similar.

References


