Application of the Unscented Transformation to Obtain Target Range, Bearing, Elevation, Range Rate, Speed and Course Estimates and Errors From Tracks Given in Cartesian Coordinates

Hwa-Tung Ong

Information Integration Branch
Defence Science and Technology Organisation

DSTO–TN–0745

ABSTRACT

The unscented transformation is applied to tracks given in Cartesian coordinates, to obtain target range, bearing, elevation, range rate, speed and course estimates, and their associated errors. This approach is attractive because it does not require tedious closed-form mathematical derivations, is not computationally expensive, and does not require a pseudo-random number generator. Example code and results are given to demonstrate the applicability of the approach.

APPROVED FOR PUBLIC RELEASE
Application of the Unscented Transformation to Obtain Target Range, Bearing, Elevation, Range Rate, Speed and Course Estimates and Errors From Tracks Given in Cartesian Coordinates

EXECUTIVE SUMMARY

Target tracking algorithms typically produce tracks in Cartesian coordinates but target range, bearing, elevation and other estimates may be required to meet software interface specifications. In this report, an algorithm based on the unscented transformation is described that allows such estimates and their error standard deviations to be computed. This report is provided as a contribution to the Defence Science and Technology Organisation’s Virtual Maritime System Architecture (VMSA). The VMSA plays an important role in providing modelling and simulation support for Australian Defence projects such as the SEA 1448 project to upgrade the anti-ship missile defence system of the Royal Australian Navy’s ANZAC class frigates.

Under the unscented transformation approach, closed-form derivations are unnecessary and computational requirements are at least an order of magnitude less than those of the Monte Carlo method. Furthermore, unlike the Monte Carlo method, the unscented transformation does not require a pseudo-random number generator.

The basic idea of the unscented transformation is to choose a set of sigma points with assigned weights to represent a multivariate normal distribution that preserves the mean and covariance. The points are then transformed, and the transformed points are used to calculate estimates of the transformed mean and covariance.

Table 1 on page 4 of this report summarises the algorithm to obtain range, bearing, elevation, range rate, speed and course from track data given in Cartesian coordinates.

Table 2 on page 5 of this report compares example results from the unscented transformation algorithm with those obtained from Monte Carlo simulations. The estimated quantities and errors were found to be in close agreement. Java code for the example are given in the appendix.
Contents

1 Introduction 1
2 Algorithm 1
3 Example 3
4 Conclusions 5
References 5

Appendices

A Example Code 7
1 Introduction

Target tracking algorithms typically produce tracks in Cartesian coordinates but target range, bearing, elevation and other estimates may be required to meet software interface specifications. In this report, an algorithm based on the unscented transformation [1] is described that allows such estimates and their error standard deviations to be computed. This report is provided as a contribution to the Defence Science and Technology Organisation’s Virtual Maritime System Architecture (VMSA). The VMSA plays an important role in providing modelling and simulation support for Australian Defence projects such as the SEA 1448 project to upgrade the anti-ship missile defence system of the Royal Australian Navy’s ANZAC class frigates.

Track position and velocity are given in Cartesian coordinates, \((x, y, z)\) and \((\dot{x}, \dot{y}, \dot{z})\), together with track covariance which quantifies the track errors. Obtaining estimates of target range, bearing, elevation, range rate, speed and course, and their error standard deviations, involves nonlinear functions and is consequently not straightforward. The resultant statistical distribution of any nonlinear function of track variables cannot be determined by directly transforming the track mean and covariance statistics, even when the track is completely characterised by its mean and covariance. To illustrate in mathematical terms, if \(x\) is a random variable with mean \(\mu\) and variance \(\sigma^2\), there is, in general, no closed-form expression for the mean and variance of a nonlinear function \(h(x)\) in terms of \(\mu\) and \(\sigma^2\). Note that \(h(\mu)\) is not, in general, the mean of \(h(x)\).

In practice, there are several ways of estimating the statistics of nonlinearly transformed random variables. The standard method is to approximate the transformation by the linear function obtained by truncating its Taylor series expanded about the mean of the random variables. Another approach, which has been applied to convert measurements from polar/spherical to Cartesian coordinates [2, 3, 4], relies on trigonometric identities. A computationally-intensive alternative is to apply the Monte Carlo method [5] which does not require tedious closed-form derivations. In this report, the unscented transformation [1] is applied. Under this approach, closed-form derivations are unnecessary and computational requirements are at least an order of magnitude less than those of the Monte Carlo method. Furthermore, unlike the Monte Carlo method, the unscented transformation does not require a pseudo-random number generator.

In the next section, the algorithm based on the unscented transformation is described. In Section 3, example results are provided and compared with Monte Carlo results. The appendix contains Java code of the algorithm for generating the results. The conclusions are given in Section 4.

2 Algorithm

The basic idea of the unscented transformation is to choose a set of sigma points with assigned weights to represent a multivariate normal distribution that preserves the mean and covariance [1]. The points are then transformed, and the transformed points are used to calculate estimates of the transformed mean and covariance.
The sigma points, \( x^{(0)}, \ldots, x^{(p)} \), and their weights, \( W^{(0)}, \ldots, W^{(p)} \), can be chosen in a number of ways but must satisfy [1]:

\[
\sum_{i=0}^{p} W^{(i)} = 1 \tag{1}
\]

\[
\sum_{i=0}^{p} W^{(i)} x^{(i)} = \bar{x} \tag{2}
\]

\[
\sum_{i=0}^{p} W^{(i)} (x^{(i)} - \bar{x}) (x^{(i)} - \bar{x})^T = P_x \tag{3}
\]

where \( \bar{x} \) and \( P_x \) are the mean and covariance of the target track state \( x \) respectively. Under these conditions, the unscented transformation approximates the mean and covariance of \( h(x) \), denoted \( z \), by [1]

\[
\bar{z} = \sum_{i=0}^{p} W^{(i)} z^{(i)} \tag{4}
\]

\[
P_z = \sum_{i=0}^{p} W^{(i)} (z^{(i)} - \bar{z}) (z^{(i)} - \bar{z})^T \tag{5}
\]

where \( z^{(i)} = h(x^{(i)}) \), \( i = 0, \ldots, p \).

For the purpose of this report, the following symmetric set of \( 2N_x \) points and weights which satisfy conditions (1)–(3) are chosen [1]:

\[
x^{(i)} = \bar{x} + c_i \tag{6}
\]

\[
W^{(i)} = 1/(2N_x) \tag{7}
\]

\[
x^{(i+N_x)} = \bar{x} - c_i \tag{8}
\]

\[
W^{(i+N_x)} = 1/(2N_x) \tag{9}
\]

for \( i = 0, \ldots, N_x - 1 \), where \( N_x \) is the number of dimensions of \( x \) and \( c_i \) is the \((i + 1)\)th column of the Cholesky decomposition \( C \) of \( N_x P_x \), i.e., \( N_x P_x = CC^T \). Note that \( P_x \) must be a symmetric matrix for the Cholesky matrix \( C \) to exist.

The set of \( 2N_x \) points can be extended by adding another point equal to the mean \( \bar{x} \). This extra point provides a parameter for controlling higher moments of the distribution through its weighting. More details are given by Julier and Uhlmann [1].

The entries of the Cholesky matrix \( C \) are given by [6]:

\[
c_{i,j} = \frac{1}{c_{j,j}} \left( p_{i,j} - \sum_{k=1}^{j-1} c_{i,k} c_{j,k} \right), \quad \text{for } i > j \tag{10}
\]

\[
c_{i,i} = \sqrt{p_{i,i} - \sum_{k=1}^{i-1} c_{i,k}^2} \tag{11}
\]

\[
c_{i,j} = 0, \quad \text{for } i < j \tag{12}
\]
where \( c_{i,j} \) is the \( i \)th row, \( j \)th column entry of \( C \), and \( p_{i,j} \) is the \( i \)th row, \( j \)th column entry of \( P_x \). Note that \( C \) is a lower triangular matrix.

For a target track where \( \mathbf{x} \) is defined as \((x, y, z, \dot{x}, \dot{y}, \dot{z})\), the target track range \( r \), bearing \( b \), elevation \( e \), range rate \( \dot{r} \), speed \( s \) and course \( c \) are given by:

\[
\begin{align*}
    r & = \sqrt{x^2 + y^2 + z^2} \\
    b & = \arctan(x, y) \\
    e & = \arcsin(z/r) \\
    \dot{r} & = (x\dot{x} + y\dot{y} + z\dot{z})/r \\
    s & = \sqrt{\dot{x}^2 + \dot{y}^2} \\
    c & = \arctan(\dot{x}, \dot{y})
\end{align*}
\]

where \( \arctan(\cdot, \cdot) \) is the four quadrant arctangent function. Note that, in this report, the arctangent returns bearing and course relative to the \( y \)-axis.

The equations above define the nonlinear transformation \( \mathbf{z} = \mathbf{h}(\mathbf{x}) = (r, b, e, \dot{r}, s, c) \). To obtain the mean and covariance of \( \mathbf{z} \), the unscented transformation is applied with modifications to accommodate the directional data, i.e., bearing, elevation and course. If \( \theta^{(0)}, \ldots, \theta^{(p)} \) represents a set of angles falling within an interval of length \( 2\pi \), then the directional mean \( \bar{\theta} \) and standard deviation \( \sigma_\theta \) of this set of angles are given by [7]:

\[
\begin{align*}
    \bar{\theta} & = \arctan \left( \sum_{i=0}^{p} W(i) \sin \theta^{(i)} , \sum_{i=0}^{p} W(i) \cos \theta^{(i)} \right) \\
    \sigma_\theta & = \sqrt{\sum_{i=0}^{p} W(i) \bar{\theta}^{(i)}{}^2}
\end{align*}
\]

where

\[
\bar{\theta}^{(i)} = \begin{cases} 
    |\theta^{(i)} - \bar{\theta}| & \text{if } |\theta^{(i)} - \bar{\theta}| \leq \pi, \\
    2\pi - |\theta^{(i)} - \bar{\theta}| & \text{if } |\theta^{(i)} - \bar{\theta}| > \pi.
\end{cases}
\]

Table 1 summarises the steps to obtain range, bearing, elevation, range rate, speed and course from track data given in Cartesian coordinates.

### 3 Example

The algorithm in Table 1 was implemented in Java and applied to the following track mean and covariance, as an example.

\[
\begin{align*}
    \mathbf{x} & = \begin{pmatrix} 50 \\ 100 \\ 1 \\ -10 \\ -5 \\ 0 \end{pmatrix} & \mathbf{P} & = \begin{pmatrix} 5 & 4 & 3 & 2 & 1 & 0 \\ 4 & 5 & 4 & 3 & 2 & 1 \\ 3 & 4 & 5 & 4 & 3 & 2 \\ 2 & 3 & 4 & 5 & 4 & 3 \\ 1 & 2 & 3 & 4 & 5 & 4 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{pmatrix}
\end{align*}
\]
**Table 1:** Steps to obtain range, bearing, elevation, range rate, speed and course from track data given in Cartesian coordinates

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1    | **Input:** Track state estimate $\bar{x}$ and covariance matrix $P$ for $x = (x, y, z, \dot{x}, \dot{y}, \dot{z})$  
**Output:** Range, bearing, elevation, range rate, speed and course estimates and error standard deviations  
**Step 1** $N_x = 6$  
**Step 2** Compute Cholesky matrix entries for $N_x P_x$:  
\[ c_{i,j} = \frac{1}{c_{j,j}} \left( p_{i,j} - \sum_{k=1}^{j-1} c_{i,k} c_{j,k} \right), \quad \text{for } j = 1, \ldots, i - 1 \]  
\[ c_{i,i} = \sqrt{p_{i,i} - \sum_{k=1}^{i-1} c_{i,k}^2} \]  
\[ c_{i,j} = 0, \quad \text{for } j = i, \ldots, N_x \]  
for $i = 1, \ldots, N_x$, where $p_{i,j}$ is the $(i,j)$th entry of $N_x P$.  
**Step 3** Create sigma points $x^{(1)}, \ldots, x^{(2N_x)}$:  
\[ x^{(i)} = \bar{x} + c_i \quad \quad \quad x^{(i+N_x)} = \bar{x} - c_i \]  
for $i = 1, \ldots, N_x$, where $c_i$ is the $i$th column of $C$.  
**Step 4** Transform each sigma point: $z^{(i)} = h(x^{(i)})$, where $h(x)$ is given by  
\[ r = \sqrt{x^2 + y^2 + z^2} \quad b = \arctan(x, y) \quad e = \arcsin(z/r) \]  
\[ \dot{r} = (x\dot{x} + y\dot{y} + z\dot{z})/r \quad s = \sqrt{\dot{x}^2 + \dot{y}^2} \quad c = \arctan(\dot{x}, \dot{y}) \]  
**Step 5** Compute mean and standard deviation for range, range rate and speed using the formulae  
\[ \bar{x} = \frac{1}{2N_x} \sum_{i=1}^{2N_x} x^{(i)} \quad \quad \sigma_x = \sqrt{\frac{1}{2N_x} \sum_{i=1}^{2N_x} (x^{(i)} - \bar{x})^2} \]  
and mean and standard deviation for bearing, elevation and course using the formulae  
\[ \bar{\theta} = \arctan \left( \frac{\sum_{i=1}^{2N_x} \sin \theta^{(i)}}{\sum_{i=1}^{2N_x} \cos \theta^{(i)}} \right) \quad \sigma_\theta = \sqrt{\frac{1}{2N_x} \sum_{i=1}^{2N_x} \tilde{\theta}^{(i)}^2} \]  
where  
\[ \tilde{\theta}^{(i)} = \begin{cases} \theta^{(i)} - \bar{\theta} & \text{if } |\theta^{(i)} - \bar{\theta}| \leq \pi, \\ 2\pi - |\theta^{(i)} - \bar{\theta}| & \text{if } |\theta^{(i)} - \bar{\theta}| > \pi. \end{cases} \]
Table 2: Results from Unscented Transformation (UT), Monte Carlo simulations (MC) and the track mean (TM)

<table>
<thead>
<tr>
<th></th>
<th>UT Error</th>
<th>MC Error</th>
<th>TM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>111.84</td>
<td>111.84</td>
<td>111.81</td>
</tr>
<tr>
<td>Bearing</td>
<td>0.46</td>
<td>0.46</td>
<td>0.46</td>
</tr>
<tr>
<td>Elevation</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Range Rate</td>
<td>-8.92</td>
<td>-8.95</td>
<td>-8.94</td>
</tr>
<tr>
<td>Course</td>
<td>-2.01</td>
<td>-2.01</td>
<td>-2.03</td>
</tr>
<tr>
<td>Speed</td>
<td>11.26</td>
<td>11.30</td>
<td>11.18</td>
</tr>
</tbody>
</table>

The code is given in the appendix of this report. Table 2 compares the results from the algorithm with that obtained using Monte Carlo simulations (10,000 samples) and with estimates based on the track mean. From the results, it appears that the unscented transformation yields results that are in close agreement with those from Monte Carlo simulations. The estimates based on the track mean are also in close agreement.

4 Conclusions

The unscented transformation can be applied to tracks given in Cartesian coordinates, to obtain target range, bearing, elevation, range rate, speed and course estimates, and their associated errors. This approach is attractive because it does not require tedious closed-form mathematical derivations, is not computationally expensive, and does not require a pseudo-random number generator. Example code and results have been given to demonstrate the applicability of the approach.

References


Appendix A  Example Code

The following example code is written in Java. Note that the covariance matrix must be symmetric and positive definite for the Cholesky decomposition. Divide by zero can occur if the covariance matrix is degenerate.

```java
class Example {

    public static double mean(double[] x) {
        double m = 0;
        for(int i = 0 ; i < x.length ; i++)
            m += x[i];
        return m / x.length;
    }

    public static double stddev(double[] x, double m) {
        double ss = 0;
        double s;
        for(int i = 0 ; i < x.length ; i++) {
            s = x[i] - m;
            ss += s * s;
        }
        return Math.sqrt(ss / x.length);
    }

    public static double dmean(double[] x) {
        // Mean for directional data given in radians
        double s = 0;
        double c = 0;
        for(int i = 0 ; i < x.length ; i++) {
            s += Math.sin(x[i]);
            c += Math.cos(x[i]);
        }
        return Math.atan2(s, c);
    }

    public static double dstddev(double[] x, double m) {
        // Standard deviation for directional data given in radians
        // Data points are assumed to all fall in the same 2*pi interval
        double ss = 0;
        double s;
        double TWOP = 2*Math.PI;
        for(int i = 0 ; i < x.length ; i++) {
            s = Math.abs(x[i] - m);
            if(s > Math.PI)
                s = TWOP - s;
            ss += s * s;
        }
    }
}
```
public static void compute_results(double[] x, double[][] P) {

    // Number of dimensions
    int n = x.length;

    // Weight covariance matrix
    // Note: the lower triangular part of the covariance matrix
    // is over-written in the process
    for(int i = 0 ; i < n ; i++)
    for(int j = 0 ; j <= i ; j++)
        P[i][j] = n * P[i][j];

    // Compute lower-triangular Cholesky decomposition using the
    // Cholesky-Banachiewicz algorithm
    for(int i = 0 ; i < n ; i++) {
        for(int j = 0 ; j < i ; j++) {
            for(int k = 0 ; k < j ; k++)
                P[i][j] = P[i][j] - P[i][k] * P[j][k];
            P[i][j] = P[i][j] / P[j][j];
        }
        for(int k = 0 ; k < i ; k++)
            P[i][i] = P[i][i] - P[i][k] * P[i][k];
        P[i][i] = Math.sqrt(P[i][i]);
    }

    // Create sigma points
    double[][] z = new double[n][2*n];
    for(int i = 0 ; i < n ; i++) {
        for(int j = 0 ; j <= i ; j++) {
            z[i][j] = x[i] + P[i][j];
            z[i][n+j] = x[i] - P[i][j];
        }
        for(int j = i + 1; j < n ; j++) {
            // Note that for Cholesky decomposition, entries above
            // diagonal are zero
            z[i][j] = x[i];
            z[i][n+j] = x[i];
        }
    }

    // Transform sigma points
    // Sigma points are assumed to be 3D Cartesian position and velocity
    double[] r = new double[2*n]; // Slant range
double[] b = new double[2*n]; // Bearing relative to y-axis (radians)
double[] e = new double[2*n]; // Elevation (radians)
double[] rr = new double[2*n]; // Slant range rate
double[] c = new double[2*n]; // Course relative to y-axis (radians)
double[] s = new double[2*n]; // Speed in x-y plane
for(int i = 0 ; i < 2*n ; i++) {
    r[i] = Math.sqrt(z[0][i]*z[0][i] + z[1][i]*z[1][i] + z[2][i]*z[2][i]);
    b[i] = Math.atan2(z[0][i], z[1][i]);
    e[i] = Math.asin(z[2][i] / r[i]);
    rr[i] = (z[0][i]*z[3][i] + z[1][i]*z[4][i] + z[2][i]*z[5][i]) / r[i];
    c[i] = Math.atan2(z[3][i], z[4][i]);
    s[i] = Math.sqrt(z[3][i]*z[3][i] + z[4][i]*z[4][i]);
}

// Determine slant range and error (standard deviation)
double range = mean(r);
double rangeError = stddev(r, range);

// Determine bearing and error (standard deviation)
double bearing = dmean(b);
double bearingError = dstddev(b, bearing);

// Determine elevation and error (standard deviation)
double elevation = dmean(e);
double elevationError = dstddev(e, elevation);

// Determine slant range rate and error (standard deviation)
double rangeRate = mean(rr);
double rangeRateError = stddev(rr, rangeRate);

// Determine course and error (standard deviation)
double course = dmean(c);
double courseError = dstddev(c, course);

// Determine speed and error (standard deviation)
double speed = mean(s);
double speedError = stddev(s, speed);

System.out.format("Range = %7.2f, ", range);
System.out.format("Error = %7.2f\n", rangeError);
System.out.format("Bearing = %7.2f, ", bearing);
System.out.format("Error = %7.2f\n", bearingError);
System.out.format("Elevation = %7.2f, ", elevation);
System.out.format("Error = %7.2f\n", elevationError);
System.out.format("Range Rate = %7.2f, ", rangeRate);
System.out.format("Error = %7.2f\n", rangeRateError);
System.out.format("Course = %7.2f, ", course);
System.out.format("\n");
System.out.format("Error = %7.2f%n", courseError);
System.out.format("Speed = %7.2f, ", speed);
System.out.format("Error = %7.2f%n", speedError);
System.out.println();
System.out.println("Cholesky Decomposition");
for(int i = 0 ; i < n ; i++) {
    for(int j = 0 ; j <= i ; j++)
        System.out.format("%7.2f ", P[i][j]);
    System.out.println();
}
System.out.println();
System.out.println("Sigma Points");
for(int i = 0 ; i < n ; i++) {
    for(int j = 0 ; j < 2*n ; j++)
        System.out.format("%7.2f ", z[i][j]);
    System.out.println();
}
}

public static void main(String[] args) {

    // Example mean
    double[] x = {50, 100, 1, -10, -5, 0};

    // Example covariance matrix
    // This must be a symmetric matrix
    double[][] P = {
        {5, 4, 3, 2, 1, 0},
        {4, 5, 4, 3, 2, 1},
        {3, 4, 5, 4, 3, 2},
        {2, 3, 4, 5, 4, 3},
        {1, 2, 3, 4, 5, 4},
        {0, 1, 2, 3, 4, 5}
    };

    compute_results(x, P);
}
}
Application of the Unscented Transformation to Obtain Target Range, Bearing, Elevation, Range Rate, Speed and Course Estimates and Errors From Tracks Given in Cartesian Coordinates

Hwa-Tung Ong
Defence Archives 1 (printed)

Capability Development Executive
Director General Capability and Plans Doc Data Sheet
Assistant Secretary Investment Analysis Doc Data Sheet
Director Capability Plans and Programming Doc Data Sheet

Chief Information Officer Group
Head Information Capability Management Division Doc Data Sheet
Director General Australian Defence Simulation Office Doc Data Sheet
AS Information Strategy and Futures Doc Data Sheet
Director General Information Services Doc Data Sheet

Strategy Executive
Assistant Secretary Strategic Planning Doc Data Sheet
Assistant Secretary International and Domestic Security Policy Doc Data Sheet

Navy
Maritime Operational Analysis Centre, Building 89/90 Doc Data Sheet
Garden Island Sydney NSW and Dist List
Deputy Director (Operations)
Deputy Director (Analysis)
Director General Navy Capability, Performance and Plans, Navy Headquarters Doc Data Sheet
Director General Navy Strategic Policy and Futures, Navy Headquarters Doc Data Sheet

Air Force
SO (Science) – Headquarters Air Combat Group, RAAF Base, Williambtown NSW 2314 Doc Data Sheet
Staff Officer Science Surveillance and Response Group and Exec Summ

Army
Australian National Coordination Officer ABCA (AS NCO ABCA), Land Warfare Development Centre, Puckapunyal Doc Data Sheet
J86 (TCS GROUP), DJFHQ Doc Data Sheet
SO (Science) – Land Headquarters (LHQ), Victoria Barracks, NSW Doc Data Sheet
SO (Science) – Special Operations Command (SOCOMD), R5-SB-15, Russell Offices Canberra and Exec Summ

SO (Science), Deployable Joint Force Headquarters (DJFHQ)(L), Enoggera QLD and Dist List
Joint Operations Command

Director General Joint Operations
Chief of Staff Headquarters Joint Operation Command
Commandant ADF Warfare Centre
Director General Strategic Logistics
COS Australian Defence College

Intelligence and Security Group

AS Concepts, Capabilities and Resources 1
DGSTA, Defence Intelligence Organisation 1
Manager, Information Centre, Defence Intelligence Organisation 1
Director Advanced Capabilities Doc Data Sheet

Defence Materiel Organisation

Deputy CEO Doc Data Sheet
Head Aerospace Systems Division Doc Data Sheet
Head Maritime Systems Division Doc Data Sheet
Program Manager Air Warfare Destroyer Doc Data Sheet
Guided Weapon & Explosive Ordnance Branch (GWEO) Doc Data Sheet
CDR Joint Logistics Command Doc Data Sheet

OTHER ORGANISATIONS

National Library of Australia 1
NASA (Canberra) 1

UNIVERSITIES AND COLLEGES

Australian Defence Force Academy

Library 1
Head of Aerospace and Mechanical Engineering 1
Hargrave Library, Monash University Doc Data Sheet

OUTSIDE AUSTRALIA

INTERNATIONAL DEFENCE INFORMATION CENTRES

US Defense Technical Information Center 1
UK Dstl Knowledge Services 1
Canada Defence Research Directorate R&D Knowledge & Information Management (DRDKIM) 1
NZ Defence Information Centre 1
ABSTRACTING AND INFORMATION ORGANISATIONS

Library, Chemical Abstracts Reference Service 1
Engineering Societies Library, US 1
Materials Information, Cambridge Scientific Abstracts, US 1
Documents Librarian, The Center for Research Libraries, US 1

SPARES 5 (printed)

Total number of copies: printed 9, pdf 27
**DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION**

**DOCUMENT CONTROL DATA**

<table>
<thead>
<tr>
<th>1. CAVEAT/PRIVACY MARKING</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. TITLE</td>
</tr>
<tr>
<td>Application of the Unscented Transformation to Obtain Target Range, Bearing, Elevation, Range Rate, Speed and Course Estimates and Errors From Tracks Given in Cartesian Coordinates</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. SECURITY CLASSIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document (U)</td>
</tr>
<tr>
<td>Title (U)</td>
</tr>
<tr>
<td>Abstract (U)</td>
</tr>
</tbody>
</table>

| 4. AUTHORS                 |
| Hwa-Tung Ong               |

| 5. CORPORATE AUTHOR        |
| Defence Science and Technology Organisation |
| PO Box 1500                |
| Edinburgh, South Australia 5111, Australia |

| 6a. DSTO NUMBER            |
| DSTO–TN–0745              |

| 6b. AR NUMBER              |
| 013-862                    |

| 6c. TYPE OF REPORT         |
| Technical Note             |

| 7. DOCUMENT DATE           |
| March, 2007                |

| 8. FILE NUMBER             |
| 2005/1054074/1             |

| 9. TASK NUMBER             |
| NAV 05/213                 |

| 10. SPONSOR                |
| DGMD                       |

| 11. No OF PAGES            |
| 10                        |

| 12. No OF REFS             |
| 7                         |

| 13. URL OF ELECTRONIC VERSION |

| 14. RELEASE AUTHORITY      |
| Research Leader, Information Integration Branch |

| 15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT |
| Approved For Public Release |

| OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SOUTH AUSTRALIA 5111 |

| 16. DELIBERATE ANNOUNCEMENT |
| No Limitations |

| 17. CITATION IN OTHER DOCUMENTS |
| No Limitations |

| 18. DSTO RESEARCH LIBRARY THESAURUS |
| Target tracking Algorithms Tracking |

| 19. ABSTRACT |
| The unscented transformation is applied to tracks given in Cartesian coordinates, to obtain target range, bearing, elevation, range rate, speed and course estimates, and their associated errors. This approach is attractive because it does not require tedious closed-form mathematical derivations, is not computationally expensive, and does not require a pseudo-random number generator. Example code and results are given to demonstrate the applicability of the approach. |