Australian Government

Department of Defence

Defence Science and
Technology Organisation

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

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

Appendices

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 μ and variance σ^2, there is, in general, no closed-form expression for the mean and variance of a nonlinear function $h(x)$ in terms of μ and σ^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, $\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(p)}$, and their weights, $W^{(0)}, \ldots, W^{(p)}$, can be chosen in a number of ways but must satisfy [1]:

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

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

$$\sum_{i=0}^{p} W^{(i)} (\mathbf{x}^{(i)} - \bar{\mathbf{x}})(\mathbf{x}^{(i)} - \bar{\mathbf{x}})^{\mathrm{T}} = \mathbf{P_x} \tag{3}$$

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

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

$$\mathbf{P_z} = \sum_{i=0}^{p} W^{(i)} (\mathbf{z}^{(i)} - \bar{\mathbf{z}})(\mathbf{z}^{(i)} - \bar{\mathbf{z}})^{\mathrm{T}} \tag{5}$$

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

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

$$\mathbf{x}^{(i)} = \bar{\mathbf{x}} + \mathbf{c}_i \tag{6}$$
$$\mathbf{W}^{(i)} = 1/(2N_\mathbf{x}) \tag{7}$$
$$\mathbf{x}^{(i+N_\mathbf{x})} = \bar{\mathbf{x}} - \mathbf{c}_i \tag{8}$$
$$\mathbf{W}^{(i+N_\mathbf{x})} = 1/(2N_\mathbf{x}) \tag{9}$$

for $i = 0, \ldots, N_\mathbf{x} - 1$, where $N_\mathbf{x}$ is the number of dimensions of $\mathbf{x}$ and $\mathbf{c}_i$ is the $(i+1)$th column of the Cholesky decomposition $\mathbf{C}$ of $N_\mathbf{x}\mathbf{P_x}$, i.e., $N_\mathbf{x}\mathbf{P_x} = \mathbf{C}\mathbf{C}^{\mathrm{T}}$. Note that $\mathbf{P_x}$ must be a symmetric matrix for the Cholesky matrix $\mathbf{C}$ to exist.

The set of $2N_\mathbf{x}$ points can be extended by adding another point equal to the mean $\bar{\mathbf{x}}$. 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 $\mathbf{C}$ are given by [6]:

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

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

$$c_{i,j} = 0, \qquad \text{for } i < j \tag{12}$$

where $c_{i,j}$ is the ith row, jth column entry of $\mathbf{C}$, and $p_{i,j}$ is the ith row, jth column entry of $\mathbf{P_x}$. Note that $\mathbf{C}$ 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:

$$r = \sqrt{x^2 + y^2 + z^2} \tag{13}$$
$$b = \arctan(x, y) \tag{14}$$
$$e = \arcsin(z/r) \tag{15}$$
$$\dot{r} = (x\dot{x} + y\dot{y} + z\dot{z})/r \tag{16}$$
$$s = \sqrt{\dot{x}^2 + \dot{y}^2} \tag{17}$$
$$c = \arctan(\dot{x}, \dot{y}) \tag{18}$$

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 2π, then the directional mean $\bar{\theta}$ and standard deviation σ_θ of this set of angles are given by [7]:

$$\bar{\theta} = \arctan\left(\sum_{i=0}^{p} W^{(i)} \sin \theta^{(i)}, \sum_{i=0}^{p} W^{(i)} \cos \theta^{(i)}\right) \tag{19}$$

$$\sigma_\theta = \sqrt{\sum_{i=0}^{p} W^{(i)} \tilde{\theta}^{(i)2}} \tag{20}$$

where

$$\tilde{\theta}^{(i)} = \begin{cases} |\theta^{(i)} - \bar{\theta}| & \text{if } |\theta^{(i)} - \bar{\theta}| \leq \pi, \\ 2\pi - |\theta^{(i)} - \bar{\theta}| & \text{if } |\theta^{(i)} - \bar{\theta}| > \pi. \end{cases} \tag{21}$$

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.

$$\mathbf{x} = \begin{pmatrix} 50 \\ 100 \\ 1 \\ -10 \\ -5 \\ 0 \end{pmatrix} \qquad \mathbf{P} = \begin{pmatrix} 5 & 4 & 3 & 2 & 1 & 0 \\ 4 & 5 & 4 & 3 & 2 & 1 \\ 3 & 4 & 5 & 4 & 3 & 2 \\ 2 & 3 & 4 & 5 & 4 & 3 \\ 1 & 2 & 3 & 4 & 5 & 4 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{pmatrix} \tag{31}$$

Table 1: *Steps to obtain range, bearing, elevation, range rate, speed and course from track data given in Cartesian coordinates*

Input: Track state estimate $\bar{\mathbf{x}}$ and covariance matrix $\mathbf{P}$ for $\mathbf{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z})$
Output: Range, bearing, elevation, range rate, speed and course estimates and error standard deviations

Step 1 $N_{\mathbf{x}} = 6$

Step 2 Compute Cholesky matrix entries for $N_{\mathbf{x}} \mathbf{P}_{\mathbf{x}}$:

$$c_{i,j} = \frac{1}{c_{j,j}} \left(p_{i,j} - \sum_{k=1}^{j-1} c_{i,k} c_{j,k} \right), \qquad \text{for } j = 1, \ldots, i-1 \tag{22}$$

$$c_{i,i} = \sqrt{p_{i,i} - \sum_{k=1}^{i-1} c_{i,k}^2} \tag{23}$$

$$c_{i,j} = 0, \qquad \text{for } j = i, \ldots, N_{\mathbf{x}} \tag{24}$$

for $i = 1, \ldots, N_{\mathbf{x}}$, where $p_{i,j}$ is the (i, j)th entry of $N_{\mathbf{x}} \mathbf{P}$.

Step 3 Create sigma points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(2N_{\mathbf{x}})}$:

$$\mathbf{x}^{(i)} = \bar{\mathbf{x}} + \mathbf{c}_i \qquad\qquad \mathbf{x}^{(i+N_{\mathbf{x}})} = \bar{\mathbf{x}} - \mathbf{c}_i \tag{25}$$

for $i = 1, \ldots, N_{\mathbf{x}}$, where $\mathbf{c}_i$ is the ith column of $\mathbf{C}$.

Step 4 Transform each sigma point: $\mathbf{z}^{(i)} = \mathbf{h}(\mathbf{x}^{(i)})$, where $\mathbf{h}(\mathbf{x})$ is given by

$$r = \sqrt{x^2 + y^2 + z^2} \qquad b = \arctan(x, y) \qquad e = \arcsin(z/r) \tag{26}$$

$$\dot{r} = (x\dot{x} + y\dot{y} + z\dot{z})/r \qquad s = \sqrt{\dot{x}^2 + \dot{y}^2} \qquad c = \arctan(\dot{x}, \dot{y}) \tag{27}$$

Step 5 Compute mean and standard deviation for range, range rate and speed using the formulae

$$\bar{x} = \frac{1}{2N_{\mathbf{x}}} \sum_{i=1}^{2N_{\mathbf{x}}} x^{(i)} \qquad\qquad \sigma_x = \sqrt{\frac{1}{2N_{\mathbf{x}}} \sum_{i=1}^{2N_{\mathbf{x}}} (x^{(i)} - \bar{x})^2} \tag{28}$$

and mean and standard deviation for bearing, elevation and course using the formulae

$$\bar{\theta} = \arctan \left(\sum_{i=1}^{2N_{\mathbf{x}}} \sin \theta^{(i)}, \sum_{i=1}^{2N_{\mathbf{x}}} \cos \theta^{(i)} \right) \qquad \sigma_\theta = \sqrt{\frac{1}{2N_{\mathbf{x}}} \sum_{i=1}^{2N_{\mathbf{x}}} \tilde{\theta}^{(i)2}} \tag{29}$$

where

$$\tilde{\theta}^{(i)} = \begin{cases} |\theta^{(i)} - \bar{\theta}| & \text{if } |\theta^{(i)} - \bar{\theta}| \leq \pi, \\ 2\pi - |\theta^{(i)} - \bar{\theta}| & \text{if } |\theta^{(i)} - \bar{\theta}| > \pi. \end{cases} \tag{30}$$

Table 2: *Results from Unscented Transformation (UT), Monte Carlo simulations (MC) and the track mean (TM)*

	UT	Error	MC	Error	TM
Range	111.84	2.88	111.84	2.90	111.81
Bearing	0.46	0.01	0.46	0.01	0.46
Elevation	0.01	0.02	0.01	0.02	0.01
Range Rate	-8.92	2.88	-8.95	2.88	-8.94
Course	-2.01	0.12	-2.01	0.15	-2.03
Speed	11.26	2.85	11.30	2.82	11.18

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

1. Julier, S. J. and Uhlmann, J. K. (2004) Unscented filtering and nonlinear estimation, *Proceedings of the IEEE* **92**(3), 401–422.

2. Lerro, D. and Bar-Shalom, Y. (1993) Tracking with debiased consistent converted measurements versus EKF, *IEEE Transactions on Aerospace and Electronic Systems* **29**(3), 1015–1022.

3. Longbin, M., Xiaoquan, S., Yiyu, Z., Kang, S. Z. and Bar-Shalom, Y. (1998) Unbiased converted measurements for tracking, *IEEE Transactions on Aerospace and Electronic Systems* **34**(3), 1023–1027.

4. Suchomski, P. (1999) Explicit expressions for debiased statistics of 3D converted measurements, *IEEE Transactions on Aerospace and Electronic Systems* **35**(1), 368–370.

5. Wikipedia (2007) Monte Carlo method, URL http://en.wikipedia.org/wiki/ Monte_Carlo_method.

6. Wikipedia (2007) Cholesky decomposition, URL http://en.wikipedia.org/wiki/ Cholesky_decomposition.

7. Fisher, N. I. (1993) Statistical Analysis of Circular Data, Cambridge University Press.

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 TWOPI = 2*Math.PI;
        for(int i = 0 ; i < x.length ; i++) {
            s = Math.abs(x[i] - m);
            if(s > Math.PI)
                s = TWOPI - s;
            ss += s * s;
```

```
            }
            return Math.sqrt(ss / x.length);
        }

        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
```

```java
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("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

AUSTRALIA

Number of Copies

DEFENCE ORGANISATION

Task Sponsor

Director General Maritime Development	1 (printed)

S&T Program

Chief Defence Scientist	1
Deputy Chief Defence Scientist Policy	1
AS Science Corporate Management	1
Director General Science Policy Development	1
Counsellor Defence Science, London	Doc Data Sheet
Counsellor Defence Science, Washington	Doc Data Sheet
Scientific Adviser to MRDC, Thailand	Doc Data Sheet
Scientific Adviser Joint	1
Navy Scientific Adviser	1
Scientific Adviser – Army	Doc Data Sheet and Dist List
Air Force Scientific Adviser	Doc Data Sheet and Dist List
Scientific Adviser to the DMO	Doc Data Sheet and Dist List
Deputy Chief Defence Scientist Platform and Human Systems	Doc Data Sheet and Exec Summ
Chief of Intelligence, Surveillance and Reconnaissance Division	Doc Data Sheet and Dist List
Research Leader, Information Integration	Doc Data Sheet and Dist List
Head, Tracking and Sensor Fusion	1
ISRD TSF Group	1
Task Manager	1
Author	1
Gavin Puddy, Maritime Operations Division	1
Soi Do, Maritime Operations Division	1

DSTO Library and Archives

Library Edinburgh	2 (printed)

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 Garden Island Sydney NSW Deputy Director (Operations) Deputy Director (Analysis)	Doc Data Sheet and Dist List
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, Williamtown NSW 2314	Doc Data Sheet and Exec Summ
Staff Officer Science Surveillance and Response Group	Doc Data Sheet 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 and Exec Summ
SO (Science) – Special Operations Command (SOCOMD), R5-SB-15, Russell Offices Canberra	Doc Data Sheet and Exec Summ and Dist List
SO (Science), Deployable Joint Force Headquarters (DJFHQ)(L), Enoggera QLD	Doc Data Sheet

Joint Operations Command

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

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

<div align="center">

OUTSIDE AUSTRALIA

</div>

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	1. CAVEAT/PRIVACY MARKING		

2. TITLE	3. SECURITY CLASSIFICATION		
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	Document　　　　(U) Title　　　　　　(U) Abstract　　　　(U)		

4. AUTHORS	5. CORPORATE AUTHOR		
Hwa-Tung Ong	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 http://www.dsto.defence.gov.au/corporate/ reports/DSTO–TN–0745.pdf	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
Transformations	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.