Data Association and Tracking for a Multi-sensor Multi-target Scenario

V. P. S. Naidu*, Girija. **G*,** J. R. Raol* and R. Appavu Raj+
* Scientists, FMCD, National Aerospace Laboratories, Bangalore, India.
t Scientist, ITR, **DRDO,** India

Abstract

Tracking in multi sensor multi target (MSMT) scenario is a complex problem due to the uncertainties in the origin of observations. Solution to this problem requires appropriate gating and data association procedures to associate measurements with targets. A PC MATLAB program has been developed based on gating and data association using nearest neighborhood approach to track multiple targets with multiple ground based radars. In this paper the details of the procedure followed and tbe results of the performance of the algorithm for a typical (MSMT) situation are presented.

1. Introduction

The central problem in MSMT tracking is data association [1], – i.e. to determine from which target, if any, a particular measurement originated. There are some difficulties in performing multi sensor tracking due to uncertain data and disparate data sources. The identity of the targets responsible for each individual data set is unknown, so there is uncertainty as how to associate data from one sensor which are obtained at one time and location to those of another sensor at another point in time and location. Also, false alarms and the clutter detections may be present which are not easily distinguishable from the true target measurements.

Gating and data association enable tracking in multi-target multi sensor scenario. **Gating** helps in deciding if an observation (which includes clutter, false alarms and electronic counter measures) is a probable candidate for track maintenance or track update and **Data association** is the step to associate the measurements to the targets with certainty when several targets are in the same neighborhood.

Two commonly used approaches for multi target tracking are 'target oriented' and 'track oriented' approaches. In the target-oriented approach, the number of targets are assumed to be known and all data association hypotheses are combined into one for each target. The track oriented approach treats each track individually while they are initiated, updated and terminated based on the associated measurement history.

Track oriented approach is pursued **for** the application in **this** paper (since the other approach cannot handle track initiation and can only handle track continuation). In the track-oriented algorithm, a score is assigned to each track and is updated according to the association history. **A** track is initiated based on a single measurement, and will be eliminated when the score is below a predetermined threshold. **A** PC MATLAB program based on gating and data association using nearest neighbor **[2]** approach has been developed. The test scenario considered for validating the program includes simulated data of three targets launched fi-om different sites and nine sensors located at different locations tracking the targets. The data is generated such that a set of **3** sensors track one target. The program generates information on the target-sensor lock status in addition to the estimated target track position at the end of each scan. Details of the procedure followed in developing the program and results of tracking are presented in this paper.

2. Tracking and data association algorithms fro MSMT

The steps in the MSMT program for multi-sensor multi-target tracking and data association are shown in Fig-1. A brief description of each of the steps in the algorithm is given below.

1. **Sensor attributes** including sensor location, resolution, field of view (FOV), Detection probability (PD) and False alarm probability (Pfa) are provided.
2. **New data set:** Measurements are acquired from the sensors and converted to a common reference format.
3. **Measurement-to-track association:** Gating is performed to eliminate unlikely measurement-to-track pairs.

Assuming that the measurement vector is of dimension m, a distance d^2 representing the norm of the residual vector is computed using

$$d^2 = v^T S^{-1} v \qquad (1)$$

where the residual vector is given by $v(k) = z(k) - y(k)$ and z(k) is the measurement and y(k) is the predicted value at scan k ,
S is the residual covariance matrix given by

$$S = HPH^T + R \qquad (2)$$

where H is the measurement matrix and R is the measurement error covariance matrix given by $R = diag\lfloor \sigma_x^2 \quad \sigma_y^2 \quad \sigma_z^2 \rfloor$ - for the case where three measurements x, y, z are considered. **A** correlation between the measurement and track is allowed if the distance $d^2 \le G$, where G is the χ^2 threshold. The χ^2 threshold is obtained fi-om the tables of chi-square distribution since the validation region is chi-square distributed with number of degree of freedom equal to the dimension of the measurement as shown in Table-1 [2].

For those measurements that fall within gate, the likelihood value computed using $\log(|2\pi S|) + d^2$ is entered in the correlation matrix formed with the measurements along the rows and tracks along the columns. For those measurements that fall outside the gate a high value say 1000 is entered in the matrix. From the correlation matrix, the measurement that is nearest to the track is chosen for updating the track. Once the particular measurement-to-track association pair is chosen from the correlation matrix for updating track, both will be removed fi-om matrix and next track with the least association uncertainty will be processed. This process continues until all tracks are considered. It is to be noted that each measurement can only be associated with one track and no two tracks could share same, measurement. Measurement that has not been assigned to any track will be used to initiate a new track.

4. **Track initiation:** A new track is initiated with a single measurement that is not associated with any existing track. A score is assigned **to** each initiated new track. A track is initiated by creating a Gaussian distribution from the three position measurements (i.e. x, y, z-positions) and a Gaussian distribution of the velocity vector with a priori information.
   The initial score for new track is calculated using

$$p = \frac{\beta_{NT}}{\beta_{NT} + \beta_{fa}} \tag{3}$$

   where β_{NT} = expected number of true targets
   β_{fa} = expected number of false a l m s per unit surveillance volume per scan

5. **Track updating:** If valid measurement exists, the track is updated using the validated **NN** measurement and the target dynamic state model. A score is obtained for each track based on the association history and is used in the decision of eliminating or conforming tracks
6. **Track Extrapolation:** It is possible that a track may not have any validated measurement, in which case the track will not be updated but existing tracks are just extrapolated for processing at next scan.
7. **Extrapolate tracks into next sensor FOV:** The surviving tracks in current sensor FOV are taken into next sensor FOV, because it is assumed that in MSMT scenario all sensors are tracking all targets.
8. **Extrapolate tracks into the next scan:** The surviving tracks after seen by all sensors are taken into next scan i.e. future time specified by the time difference (Δt).
9. **Track management:** Many tracks could be initiated in a clutter environment. Scoring threshold is used to eliminate the false tracks. The scoring threshold is one of the system parameter and it should be adjusted based on the scenario and performance requirement. Similar tracks are combined together to avoid redundant tracks using a distance threshold. The distance threshold is chosen from sensor resolution.
10. **Graphical display:** This module displays the true trajectory and measurements and also performance measures such as true & false track detections, number of good and false tracks, good and false tracks probabilities and also the sensor and target lock status at each instant of time.

3. Sensor and Target Models

Target model: Target dynamics use three-dimensional Cartesian co-ordinates with white acceleration process noise and are given by

$$x(k+1) = \Phi x(k) + Gw(k) \tag{4}$$

where state vector $x(k) = \left[x(k) \quad \dot{x}(k) \quad y(k) \quad \dot{y}(k) \quad z(k) \quad \dot{z}(k)\right]^T$

System noise covariance matrix $\mathbf{G} = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 & 0 \end{bmatrix}$

target dynamic state transition matrix $\Phi = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Sensor model: Sensor is modeled using measurement function, PD and Pfa. The measurement function contains the x, y and z-positions of the measurements. The PD and Pfa are assumed to be fixed since Cartesian co-ordinates are used in the algorithm. The sum of PD and Pfa is unity. The expected number of false alarms can be calculated as

$$Nfa = Pfa * \mu FOV \qquad (5)$$

where Nfa is the expected number of false alarms

μFOV is the volume of FOV.

4. Results and Discussions

An interactive program has been developed for MSMT data association and tracking as per the above steps. The program is used to identify which of the sensors in the MSMT scenario are tracking same targets using the simulated scenario of nine sensors and their measurements. The data is simulated with radars 1,2,3 tracking target 1, radars 4,5,6 and radars 7,8,9 tracking targets 2 and 3 respectively. However, this information is not utilized in the program for data association and tracking. The estimated tracks and the track association history along with the sensor lock status are provided as outputs of the program.

Fig.2 shows the trajectories as seen from respective sensor locations. The locations of sensors are shown as 'x' in the figure and are given in Table-2. The trajectories of the targets are shown as solid lines. The target identification (ID) and the sensors IDs which *are* tracking that particula target for each scan, are displayed on the screen at each scan.

It is found that initially 9 tracks survive before similar tracks are combined using a distance threshold of 10m. After this combination, it is seen that only 3 tracks survive and they have been assigned 3 target id numbers (T1, T2 and T3). The sensors, which track a particular target, are shown in Table-3 from which it is clear that three sensors track one target. Fig.3 shows the estimated trajectories of targets after combining the similar tracks.

In order to further check the tracking performance of the MSMT, clutter with a false density of $10^{-5}/m^3$ was added to the 9 sensor data. The measurements are shown in Fig.4. The estimated tracks (after combining the similar tracks) are shown in Fig.5.

5. Concluding Remarks

A PC MATLAB program has been developed based on gating and data association using nearest neighborhood approach to track multiple targets with multiple ground based radars. The tracking performance of the algorithm for a typical (MSMT) simulated situation are presented and it is clear that the tracking performance is satisfactory even in the presence of clutter in the radar data.

References:

1. K.C. Chang and **Y.** Bar-Shalom. FUSEDAT: a software package for fusion and data association and tracking with multiple sensors. In Proc. SPIE, Vol. **2235,** April 1994.
2. **Y.** Bar Shalom. Multi target multi sensor tracking: Principles and Techniques. Academic Press, 1990.

Table-1. χ^2 threshold and gate probability for a given degree of freedom (m)

χ^2 threshold / m	1	4	6.6	9	9.2	11.4	16	25
1	0.683	0.954	0.99	0.997	------	-----	0.99994	1
2	0.393	0.865	------	0.989	0.99	-----	0.9997	1
3	0.199	0.739	------	0.971	------	0.99	0.9989	0.99998

Table-2. Sensor location w.r.t common reference point

Sl. No.	Sensor	X (meters)	Y (meters)
1	S1	-2.635e3	4.7818e3
2	s 2	-1.8085	2.212e3
3	s3	2.038e3	-6.461e4
4	S4	-3.069	-6.461e4
5	s5	1.849e3	4.449e3
6	S6	7.147e2	-5.76e4
7	s7	1.8056	-6.444e4
8	S8	7.149e3	8.386e3
9	s 9	1.853e3	4.427e3

Table-3. Track Ids and corresponding Sensors Ids.

Target id	Sensorids
T1	S1. s2. s 3
T2	S4, S5, S6
T3	s7. S8. s 9

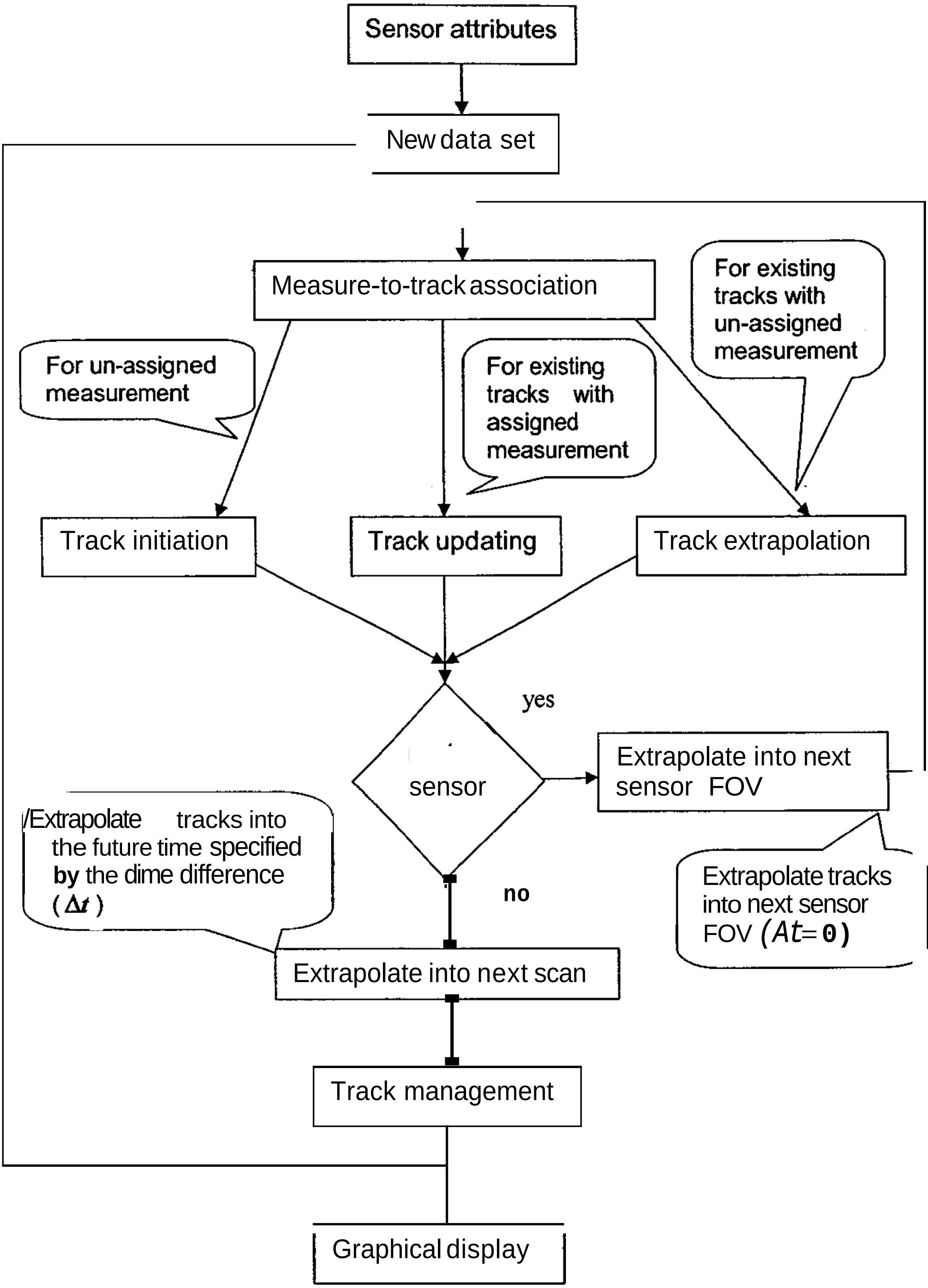

Fig-1. Multi-sensor **multi-target** algorithm

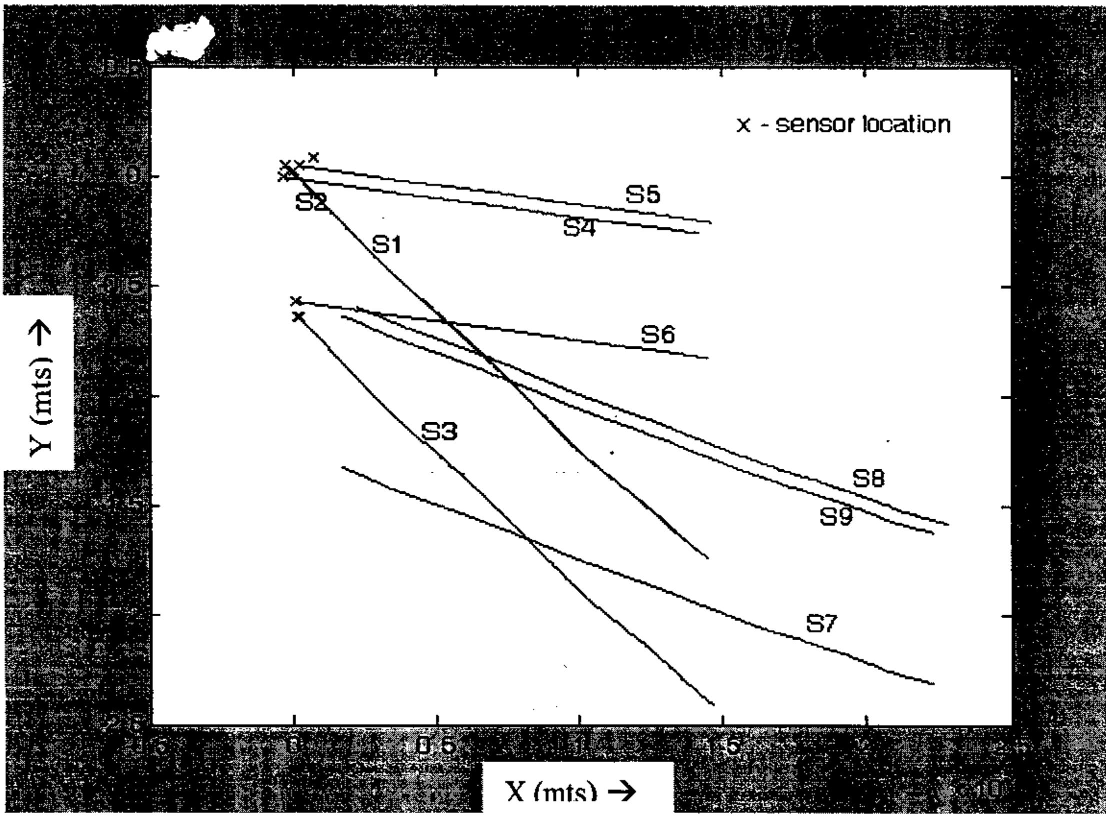

Fig.2 Trajectories as seen from respective sensor locations

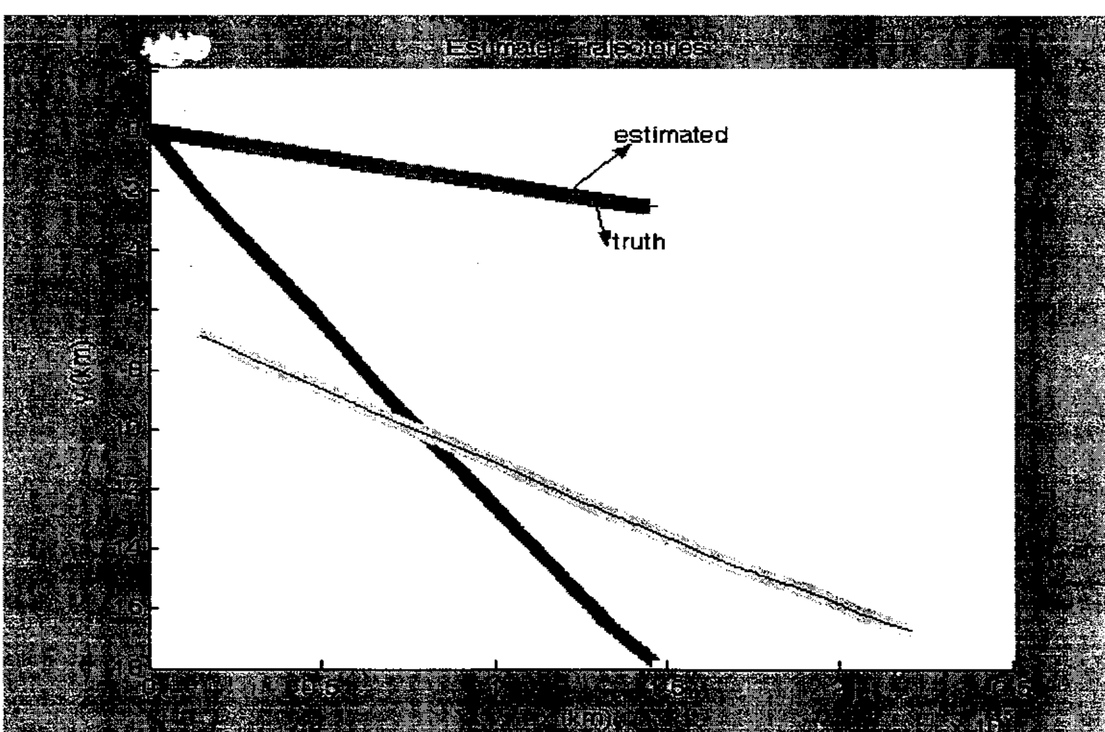

Fig.3 Estimated trajectories of targets after combining similar tracks

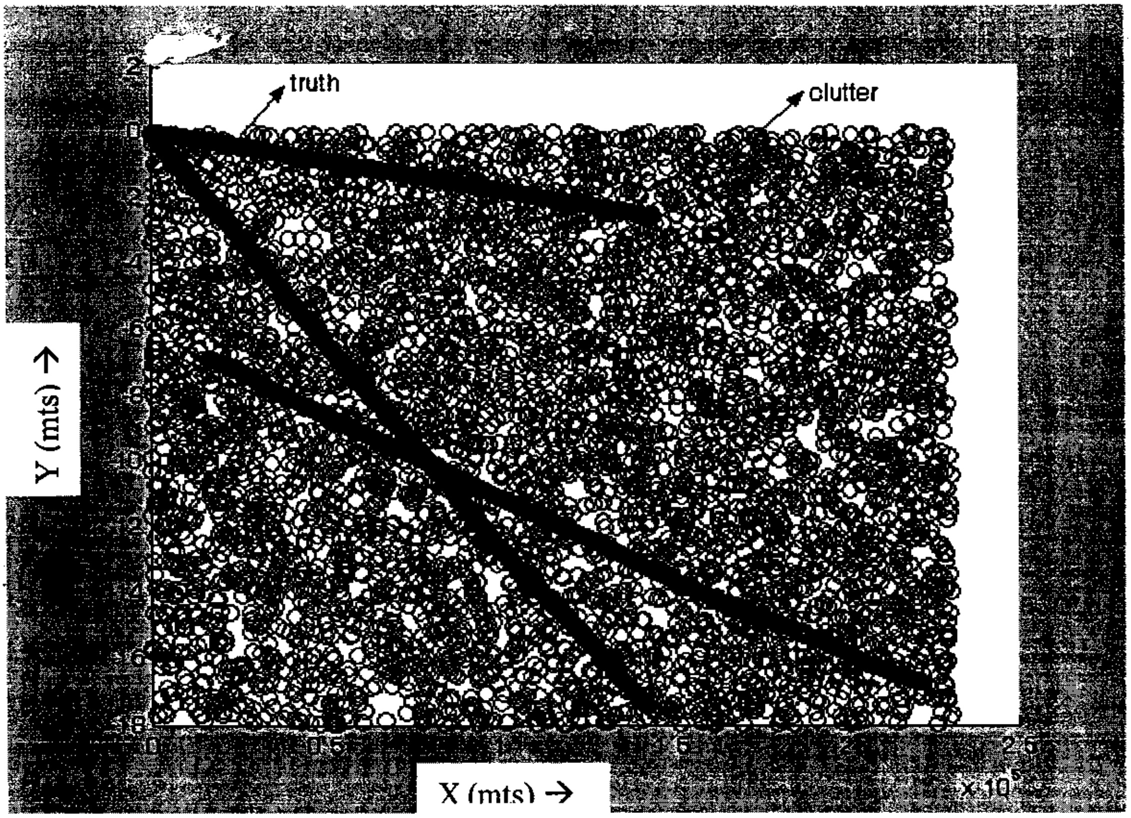

Fig.4. Trajectories with **simulated clutter**

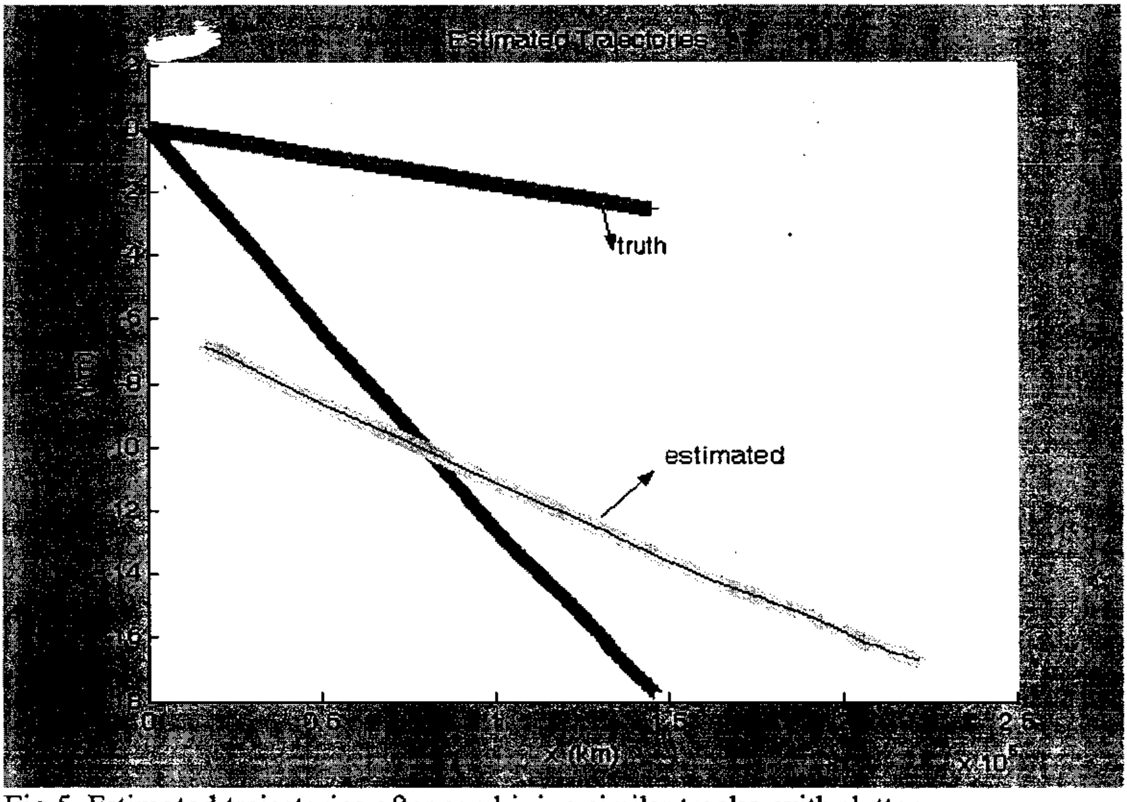

Fig.5. Estimated trajectories after combining similar tracks with clutter