

Multiple-Hypothesis Trilateration and Tracking with Distributed Radars

Jelle van Kleef, Jeroen Bergmans, Leon Kester
 TNO D&V
 {jelle.vankleef, jeroen.bergmans, leon.kester }
 @tno.nl

Frans Groen
 Informatics Institute
 University of Amsterdam
 groen@science.uva.nl

Abstract - A novel algorithm to associate and trilaterate detections from multiple distributed radars is presented. The algorithm provides for flexible track state representations. The coordinate system of a track is switched from the measurement coordinates (range-Doppler) to cartesian coordinates when a detection from another sensor is associated to the track. In the case of multiple targets and false alarms we run into the complication of multiple association possibilities. These can be resolved by using a multi-hypothesis algorithm. In general, correctly formed tracks will have more likely associations. Therefore, hypotheses describing these tracks will be favored. Simulations with one or two targets and different false alarm rates show the need to preserve multiple hypotheses of the world state. Tracking performance for various false alarms rates is evaluated.

Keywords: data association, multiple-hypothesis tracking, trilateration

1 Introduction

Recently, radar systems have become available on the consumer market, in particular due to automotive applications. To keep these radar systems cheap and robust, moving parts are avoided. The drawback of such stationary radar systems is the trade off between azimuth coverage and azimuth resolution.

A solution to overcome this problem is the use of multiple distributed radars followed by trilateration of the detections. Trilateration gives the intersection of the circles represented by the detections. This is straightforward when there is only one target, no false alarms and high probability of detection. However, in the presence of multiple targets, misdetections and false alarms, estimating the correct target locations is a complex task.

Several methods have been suggested to trilaterate multiple radar detections. Klotz [1] presents a method to trilaterate simultaneous detections of multiple FMCW radars and estimate target locations in a least squares sense. The same method is used by Rohling et al. [2].

A major problem is that it is unknown which detections of the individual sensors should be associated to each other. It is well known that when incorrect associations are made, so called *ghost* targets are found.

This paper presents a method that uses multiple

hypothesis tracking (MHT) to resolve the association problem. Each detection starts a new potential track and is associated with all existing tracks. When a track contains measurements from distributed sensors the trilateration operation can be applied and a change of tracking coordinates is made. This results in a robust tracking algorithm.

2 Trilateration

Suppose we have two sensors: one at $(x, y) = (s_1, 0)$ and one at $(x, y) = (s_2, 0)$. Figure 1 gives a schematic representation of this situation. Both sensors measure the range and Doppler shift of reflected signals. The measurement vector is denoted by $\mathbf{z} = (z, \dot{z})$. In the figure we see a detection \mathbf{z}_1 of sensor 1 and detection \mathbf{z}_2 of sensor 2, both originating from the same target. Both sensors have a beam width of 180 degrees. The likelihood of true range r and range-rate \dot{r} of a measured point target $\mathbf{r} = (r, \dot{r})$ for a measurement \mathbf{z} is denoted by $p(\mathbf{r}|\mathbf{z})$.

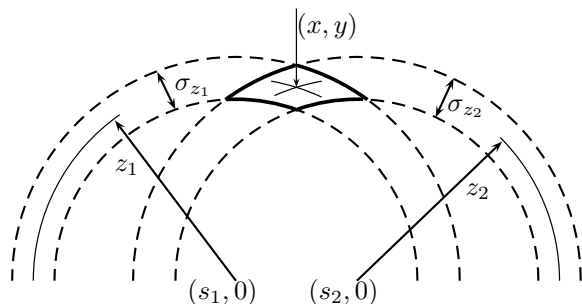


Figure 1: Trilateration

From the measurements of both sensors \mathbf{z}_1 and \mathbf{z}_2 we wish to find the position and velocity of the detected object in cartesian coordinates. It is assumed here that the targets and both radars are situated in a single plane, so their location can be described completely by their x and y coordinates. The state vector is denoted by $\mathbf{x} = (x, y, \dot{x}, \dot{y})$.

2.1 Trilateration of the state vector

For the measured ranges z_1 and z_2 of a target at (x, y) we can write:

$$z_1^2 = (x - s_1)^2 + y^2, \quad (1)$$

$$z_2^2 = (x - s_2)^2 + y^2. \quad (2)$$

Eliminating y from these equations results in

$$z_1^2 - (x - s_1)^2 = z_2^2 - (x - s_2)^2. \quad (3)$$

Solving for the x coordinate gives:

$$x = \frac{s_1^2 - s_2^2 - z_1^2 + z_2^2}{2(s_1 - s_2)}. \quad (4)$$

The y value can be found from equation 1 and 2:

$$y = \pm \sqrt{1/2 \left(z_1^2 + z_2^2 - (x - s_1)^2 - (x - s_2)^2 \right)}. \quad (5)$$

Here it is assumed that the radars have a beam width of at most 180 degrees and point in the positive y direction, so in our case the positive solution for y should be used.

The velocities $\dot{x} \equiv dx/dt$ and $\dot{y} \equiv dy/dt$ can be expressed as:

$$\dot{x} = \frac{\partial x}{\partial z_1} \dot{z}_1 + \frac{\partial x}{\partial z_2} \dot{z}_2, \quad (6)$$

$$\dot{y} = \frac{\partial y}{\partial z_1} \dot{z}_1 + \frac{\partial y}{\partial z_2} \dot{z}_2. \quad (7)$$

2.2 Trilateration of the covariance

The covariance of the trilaterated state can be calculated by multiplying the covariance of both measurements with the Jacobian of the transformation $(z_1, z_2, \dot{z}_1, \dot{z}_2) \rightarrow (x, y, \dot{x}, \dot{y})$:

$$\mathbf{C}_{\mathbf{x}\mathbf{x}} = \mathbf{J} \mathbf{C}_{z_1 z_2} \mathbf{J}^T, \quad (8)$$

where $\mathbf{C}_{z_1 z_2}$ denotes the covariance of both measurements and \mathbf{J} is the Jacobian. Note that we neglect the second and higher order terms of the (nonlinear) transformation. A location dependent representation of the covariance after trilateration can be seen in figure 2. Contours of one standard deviation σ are drawn around the transformed points represented by the asterisks. The pentagrams represent the sensor locations. Due to the scale of the plot and the range accuracy of the radars the contours are flattened to lines in the bottom pane.

Under the assumption that trilateration for x and y is locally linear, $p(\mathbf{r}_1|z_1)$ and $p(\mathbf{r}_2|z_2)$ are uncorrelated and normally distributed with covariances \mathbf{C}_{z_1} and \mathbf{C}_{z_2} respectively, the uncertainty $p(\mathbf{x}|z_1, z_2)$ can be written as function of the uncertainties $p(\mathbf{r}_1|z_1)$ and $p(\mathbf{r}_2|z_2)$:

$$p(\mathbf{x}|z_1, z_2, \mathbf{C}_{z_1}, \mathbf{C}_{z_2}) = N_d(\mathbf{x}, \mathbf{C}_{\mathbf{x}\mathbf{x}}). \quad (9)$$

Where $N_d(\mathbf{x}, \mathbf{C})$ is the d -dimensional normal (Gaussian) distribution around \mathbf{x} with covariance matrix \mathbf{C} .

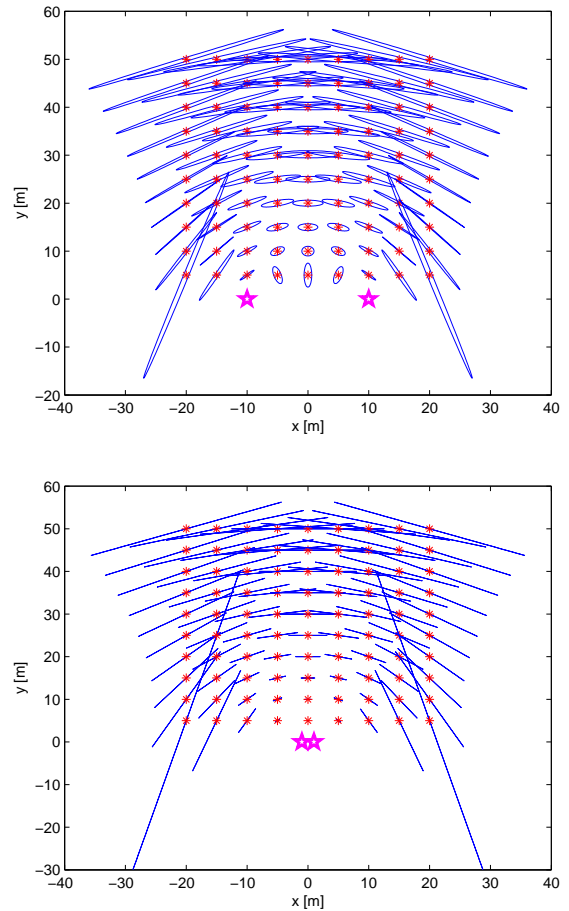


Figure 2: Location dependent covariances: Bottom: Range resolution 1 cm, sensors at -1 m and 1 m. Top: Range resolution 1 m, sensors on -10 m and 10 m.

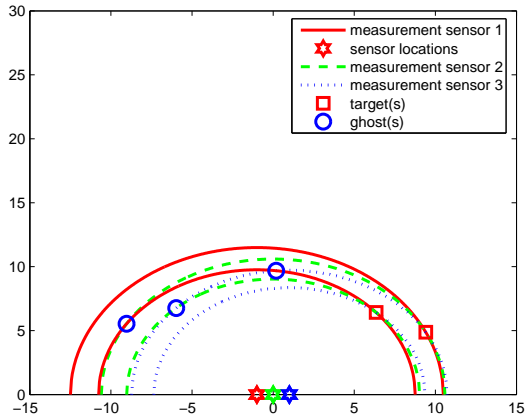


Figure 3: Target and ghost locations

3 Association

The previous section presented a simple algorithm to transform a pair of measurements or tracks from range-Doppler to cartesian coordinates. The trilateration procedure has however multiple solutions when a sensor has more than a single detection. Assume for example that there are two sensors and two targets detected by both sensors. The detections of the first sensor can be associated to those of the second sensor in two different ways, resulting in two distinct pairs of target locations. Without additional (a priori) knowledge there is no criterion to decide which two locations are correct.

A solution to this problem is to use more than two sensors. The idea is that, given point targets and an infinite resolution, only the correct associations give exactly the same triangulated target location for every pair of detections. This is graphically represented by figure 3. It can be seen that only at the two locations indicated by the squares *three* circles intersect. However, when inaccuracies are introduced, there is no longer one solution for each true target location and intersections have to be associated.

Apart from these constraints other problems arise. When for example one or more sensors do not detect the target, there will be problems to find a trilaterated estimation. In an extreme case a target would be detected by only a single sensor. Then direct trilateration is impossible. Still it is desirable to track the target using the available information. This would mean to form the track in the measurement space (in this case in range-Doppler coordinates), and switch to cartesian coordinates when a detection from another sensor is associated. This calls for a flexible state representation of tracks.

Furthermore, the tracking algorithm is required to accommodate asynchronous detections. Otherwise asynchronous measurements that are directly trilaterated lead to systematic errors in the state estimates.

The following properties are desirable for a tracking

algorithm making use of trilateration:

- flexible state representation
- flexible association decisions
- arbitrary number of sensors
- asynchronous sensor update times

Our method is designed to solve the above problems using multiple-hypothesis tracking. The algorithm is based on the measurement oriented algorithm proposed by Reid [3] which is explained comprehensively in [4]. In short, Reid’s algorithm maintains a hypothesis for each of the associations possibilities. When a detection is made, each parent hypothesis spawns a number of new hypotheses: One hypothesis that interprets the new detection as a false alarm, one that interprets the detection as a new track and N hypotheses for updates of all N tracks described by the parent hypothesis. Relative probabilities of these hypotheses are updated by multiplying with the probability of such an event.

This algorithm is very suitable for the problem at hand, because it gives the possibility to delay the decision that a certain association is correct and all others are not. However, to apply MHT to the trilateration problem, extensions have to be made to the algorithm. This is because three essentially different association types have to be performed.

In the first case a detection $M(z, \dot{z})_i$ from sensor i is associated to a track $T(z, \dot{z})_i$, which has the same state representation. The track can be updated using standard Kalman filtering, see section 4.1.

For the second case a measurement $M(z, \dot{z})_i$ from sensor i is associated to a track $T(z, \dot{z})_j$ which, up to that point, consisted of measurements made by sensor j only. When the detection is associated, the track state at the time of the new detection is predicted. Subsequently the trilateration operation (see section 2) can be applied to this prediction and the detection. As a result the state representation of the track is changed to Cartesian coordinates, $T(x, y, \dot{x}, \dot{y})$. Note that $T(z, \dot{z})_j$ is possibly a track based on a single detection.

In the third case a measurement $M(z, \dot{z})$ from any sensor is associated to a track $T(x, y, \dot{x}, \dot{y})$ which already exists in cartesian coordinates. This time we again do a Kalman update, but because the transformation from (x, y, \dot{x}, \dot{y}) to (z, \dot{z}) is nonlinear we choose an extended Kalman filter. Section 4.3 explains the update step in more detail.

In short the associations can be listed as:

1. $M(z, \dot{z})_i, T(z, \dot{z})_i \rightarrow T(z, \dot{z})_i$
2. $M(z, \dot{z})_i, T(z, \dot{z})_j \rightarrow T(x, y, \dot{x}, \dot{y})$
3. $M(z, \dot{z}), T(x, y, \dot{x}, \dot{y}) \rightarrow T(x, y, \dot{x}, \dot{y})$

where, i and j denote the sensor number, M specifies a measurement and T specifies a track.

4 Track updating

In the previous section a method was given to trilaterate two range measurements and their accuracies. This section describes a method to update a state estimate with new measurements. In the previous section three different association types were distinguished. When a track consists of data from only one sensor, the state is represented in the range-Doppler coordinate system. A standard Kalman filter is used when a detection from the same sensor is associated to such a track. See sections 4.1 and 4.2 for a more detailed description.

Standard Kalman filtering is not appropriate when a range-Doppler detection is associated to a track that is represented in cartesian coordinates. This is a result of the non-linearities of the transformation from cartesian coordinates to range-Doppler coordinates. An extended Kalman filter is used in this case. See section 4.3 for a more detailed description of this filter.

For notational convenience in this section the complete state is represented by $\mathbf{x} = (x, y, \dot{x}, \dot{y})$. The measurement is denoted by z . First we give a short introduction to general Kalman filtering. Then the two different Kalman filters will be described.

4.1 Kalman filtering

The first step of a Kalman filter is the prediction of the target state at time $t = t_k$, based on the estimated state at time $t = t_{k-1}$ which is given by:

$$\mathbf{x}(k|k-1) = \mathbf{F}\mathbf{x}(k-1|k-1), \quad (10)$$

where \mathbf{F} is the state transition matrix and we denote the time t_k simply by its subscript k . The prediction of the accuracy at time k is given by:

$$\mathbf{P}(k|k-1) = \mathbf{F}\mathbf{P}(k-1|k-1)\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T. \quad (11)$$

Here $\mathbf{P}(k-1|k-1)$ is the estimated covariance of the previous time step and \mathbf{Q} is the covariance of the process noise (usually random accelerations for a constant velocity target model). \mathbf{G} converts \mathbf{Q} to the state coordinates, so it can be added to the state.

When a new detection arrives the estimated state is corrected. The corrected state is a weighted sum of the prediction and the measurement based on the ratio of accuracies of prediction and measurement. To express this correction, the innovation vector is introduced:

$$\mathbf{e}(k) = \mathbf{z}(k) - h_i(\mathbf{x}(k|k-1)). \quad (12)$$

This vector can be interpreted as the difference of the measurement $\mathbf{z}(k)$ and the predicted state $\mathbf{x}(k|k-1)$ projected into the measurement space. The function $h_i(\mathbf{x})$ is a non-linear function providing this projection of the state vector $\mathbf{x}(k|k-1)$ to the measurement space of sensor i .

The correction step is:

$$\mathbf{x}(k|k) = \mathbf{x}(k|k-1) + \mathbf{K}(k)\mathbf{e}(k), \quad (13)$$

where

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}^T\mathbf{S}^{-1}(k), \quad (14)$$

is the Kalman gain and \mathbf{H}_i is the Jacobian of $h_i(\mathbf{x})$ in the point $\mathbf{x}(k|k-1)$.

Finally, the covariance matrix of the innovation vector \mathbf{e}_k is given by

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}^T + \mathbf{R}, \quad (15)$$

with \mathbf{R} the covariance matrix of the measurement \mathbf{z}_k .

4.2 Kalman filter for measurement coordinates

Kalman filtering in measurement coordinates means that state and measurement coordinates are equal. The state consists of range and range rate:

$$\mathbf{x} = (r, \dot{r}). \quad (16)$$

Therefore, the measurement matrix is given by:

$$h(\mathbf{x}) = \begin{pmatrix} r \\ \dot{r} \end{pmatrix}. \quad (17)$$

The Jacobian \mathbf{H} of this transformation is the two dimensional identity matrix:

$$\mathbf{H} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (18)$$

4.3 Kalman filter for cartesian coordinates

For this Kalman filter the state is represented as:

$$\mathbf{x} = (x, y, \dot{x}, \dot{y}). \quad (19)$$

The transformation to measurement coordinates is given by:

$$h_i(\mathbf{x}) = \begin{pmatrix} r_i \\ \dot{r}_i \end{pmatrix} = \begin{pmatrix} \sqrt{(x-s_i)^2 + y^2} \\ \frac{\partial r_i}{\partial x}\dot{x} + \frac{\partial r_i}{\partial y}\dot{y} \end{pmatrix}. \quad (20)$$

Here i identifies the sensor and s_i gives the x coordinate of the sensor.

The Jacobian of this transformation is given by:

$$\mathbf{H} = \begin{pmatrix} x/r_i & y/r_i & 0 & 0 \\ \frac{y(\dot{x}-x\dot{y})}{r_i^3} & \frac{x(x\dot{y}-y\dot{x})}{r_i^3} & x/r_i & y/r_i \end{pmatrix}. \quad (21)$$

5 Experiments

Simulations have been carried out in order to evaluate the performance of the tracker. The number of targets, the number of sensors as well as the number of false alarms have been varied. The labels of contacts and the true trajectories of the targets are used to evaluate the performance of the tracker. An adapted version of an MHT implementation called $M^6 T$ [5], [6] is used for the experiments.

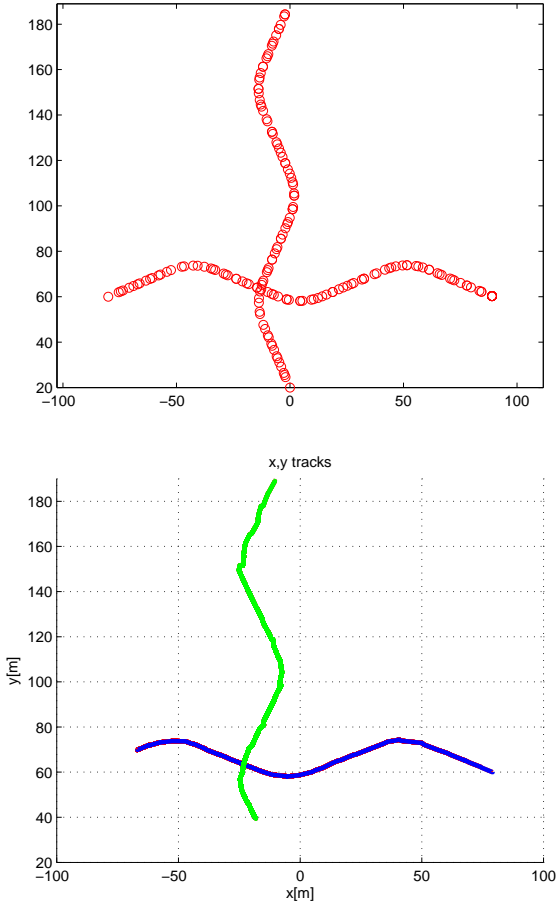


Figure 4: Ground truth (top) and tracks (bottom) of the target trajectories.

5.1 Simulations

The target contacts have been generated with a detection probability of unity. Simulations with a single target, as well as with two targets have been done. Both target scenarios have been simulated with two sensors and three sensors. For the target trajectories a shape of an 'M' is chosen. A plot of both target trajectories can be seen in figure 4. For single target simulations only the horizontal trajectory is used. The speed of the targets is constant 10 m/s and the lateral acceleration is equal to the Q value of the process noise model in the Kalman filter, i.e. 4g. The standard deviation of the range and radial velocity measurements are 5 cm and 2 cm/s, respectively. Random values drawn from a normal distribution with these standard deviations are added to the target contacts to simulate the sensor inaccuracies. The total simulation time is 20 s. For all simulations the tracker was configured to maintain maximally 50 most likely hypotheses.

The false detections have been randomly drawn from a multivariate (range, Doppler) uniform distribution. The number of false detections is constant for each simulation and given per sensor per scan. The scan periods of the three simulated sensors are 0.5, 0.6, and 0.7 s. The three sensors are located at $(x, y) = (-10 \text{ m}, 0 \text{ m})$, $(x, y) = (0 \text{ m}, 0 \text{ m})$ and $(x, y) =$

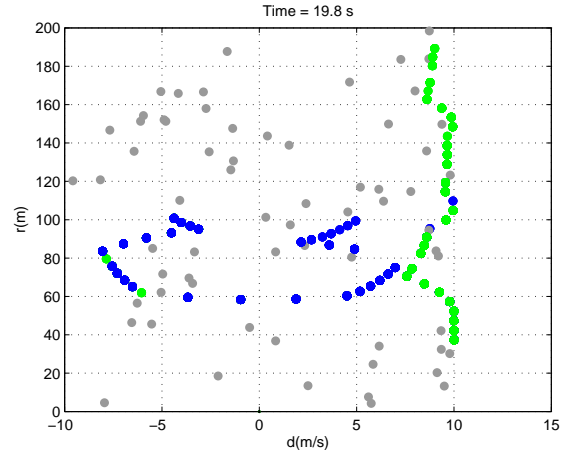


Figure 5: Detections of sensor 1

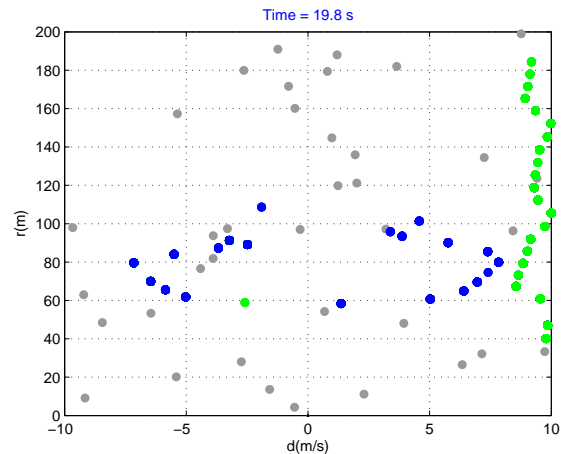


Figure 6: Detections of sensor 2

$(10 \text{ m}, 0 \text{ m})$. An example of false and target contacts can be seen in figures 5, 6, and 7. The unassociated contacts are plotted in grey, other colors are used to indicate which detections are associated together to form tracks.

5.2 Track performance evaluation

To evaluate the performance of the tracker a subset of the metrics defined in [7] is used. The metrics consist of:

- Track probability of detection (T_PD)
- False track rate (T_FAR) [track/s]
- position error (pos error)[m]
- velocity error (vel error)[m/s]

We give a short overview of the chosen metrics. For a more elaborate discussion of the metrics, the reader is referred to [7] and [8]. A track is classified 'true' when more than 50% of the contacts associated to the track originate from the same target, otherwise it is

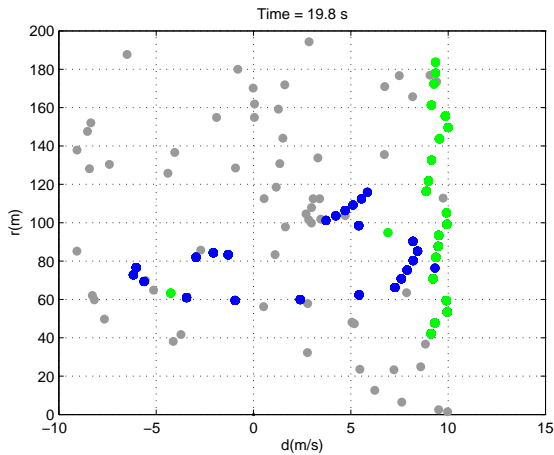


Figure 7: Detections of sensor 3

classified as a 'false' track. Only the tracks described by the most likely hypothesis are considered as tracker output. From this hypothesis only tracks with three or more updates of at least two different sensors, thus cartesian tracks, are taken into account.

The track probability of detection is defined as the total time a target was in track, normalized by the total time a target was present in the scenario.

The false track rate is defined as the cumulative number of false tracks divided by the total time of the scenario. Thus, if false tracks are present the false track rate increases when the rate increases at which the tracker output is evaluated. Note that 'ghost' tracks fall into this category if the percentage of associated detections of each target do not exceed 50%, else the tracks will be labeled 'true' and estimation errors will increase. This metric is given in tracks/s.

The position error calculated here is defined as the average of the absolute error between the ground truth position and the position of 'true' tracks. The velocity error is defined as the average of the absolute error between the ground truth velocity and the velocity of the 'true' tracks.

5.3 Results

Different datasets have been generated, fed to the tracker and then the results were evaluated. Figures 4–7 give some graphical information of the tracker output after the simulation with two targets and two false alarms per sensor per scan. The first figure shows the tracks in cartesian coordinates. The other three figures show the contacts of each sensor in the range-Doppler domain. The color of the contacts indicates to which track they are associated. Grey indicates no association.

Table 1 and 2 present the numerical values of the metrics, defined in the previous subsection, as a result of the various simulations that have been carried out. Table 1 presents the results of the single target scenario. The results show that the track probability of detection T_{PD} decreases as the number of false

alarms per scan increases. This can be explained by the longer time it takes a hypothesis that describes a true track to become the most likely hypothesis. Thus, the correct hypothesis has to compete with increasingly more hypotheses interpreting the true contacts as false alarms as the false alarm rate increases. This effect also explains why almost no false tracks were formed.

Another interesting phenomenon that can be observed is that when the hypothesis describing a true track becomes the most likely one, the track already had a number of updates. This causes the estimation of the errors of both velocity and position to decrease. This can be seen at the result for four false alarms. With rates of 7 false alarms per scan per sensor and above, no tracks are found.

The values of T_{PD} seem rather low. This is partly caused by the fact that a true track first has to reach the 'confirmed' stage, before the target is taken into account in T_{PD} .

	False Alarms per scan				
	0	1	2	4	6
T_{PD}	0.80	0.75	0.73	0.53	0.35
T_{FAR}	0	0	0.1	0	0
pos err	0.4	0.7	0.7	0.4	0.4
vel err	0.8	0.8	0.9	0.6	0.7

Table 1: metrics of tracking one simulated target

	False Alarms per scan				
	0	1	2	3	4
T_{PD}	0.65	0.66	0.56	0.51	0.33
T_{FAR}	0	1.6	2.6	7	20
pos err	0.58	0.63	2.4	10	54
vel err	0.90	0.73	28.3	50	18

Table 2: metrics of tracking two simulated targets

The results of the simulations with two targets show slightly lower detection probabilities. This is a result of an increased number of hypotheses that the correct hypothesis has to compete with. The values for false track rates are higher than in the single target case because not only false tracks, but also ghost tracks and combinations of the two are counted. It can be seen that during the simulation with two false alarms per sensor per scan incorrect associations were made. This did not result in false tracks, but in higher estimation errors. This will happen if less than 50% of the contacts is associated incorrectly. Examples of this are the simulations with two, three and four false alarms per scan.

Tracking with two sensors performed similarly for the single target case, up to 3 false alarms. With higher rates, no cartesian tracks were found. In the simulations with two targets, only zero and one false alarm per sensor per scan gave satisfying results.

Without multiple-hypotheses the tracking performed very poor with rates from two or more false

alarms per sensor per scan, resulting in large false track rates and large estimation errors.

An important conclusion that can be drawn from these results is that for the simulations with false alarm rates of less than five false alarms per scan per sensor, at least 50 % of the time tracks classified as 'true' were in track, for the single target scenario. The position error and the velocity error did not exceed 0.7 m respectively 0.9 m/s. For simulations with two targets and rates under 3 false alarms per sensor per scan, the detection probabilities were over 50 %, although some association errors were made, resulting in higher estimation errors.

6 Conclusions and future work

A new algorithm to trilaterate and track multiple targets has been presented. Tracks are represented in different coordinate systems, depending on the available information. A multiple hypothesis tracker is used to find the correct tracks. The method can be applied to other applications where state representations depend dynamically on available information, for example multi-static sensor environments.

Experiments indicate that tracks are detected at least 50% of the time for single target scenarios with rates of up to 5 false alarms per scan per sensor and up to 3 false alarms per scan per sensor for simulations with two targets.

The method described here considers all possible associations. Gating techniques to limit the number of hypotheses in MHT do not apply when trilateration associations are made. Thus, the computational costs grow rapidly as the number of contacts per scan grows. A possible solution to this problem is to form tracks in sensor coordinates first and prohibit direct inter-sensor associations. Then a check is made if there are hypotheses satisfying certain conditions to fuse the tracks from different sensors. An example for such a condition could be that the hypothesis should contain at least one track for each sensor of a certain length. The tracks can then be combined sequentially using trilateration and Kalman filtering. Of course, also the hypotheses that those tracks originate from separate targets have to be generated.

In the work presented here a priori probabilities and probability densities (detection probability, false alarm density and new target density) are presumed known and constant. Future research will involve a refinement of the probabilistic models that are used and learning these models from the data. Two approaches can be taken here. Firstly, all hypotheses together contribute to an overall interpretation of the world and probabilities are updated accordingly. Secondly, each hypothesis learns its own world models which are based on interpretations of data in that hypothesis.

7 Acknowledgments

The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

References

- [1] M. Klotz, *An Automotive Short Range High Resolution Pulse Radar Network*, PhD thesis, Technische Universitat Hamburg- Harburg, 2002.
- [2] H. Rohling, A. Hoess, U. Luebbert, M. Schiementz, *Multistatic Radar Principles for Automotive RadarNet Applications*, 2002 German Radar Symposium, Bonn, Germany, Sept. 2002, pp. 181–185
- [3] D.B. Reid. *An algorithm for tracking multiple targets*, IEEE Transaction on Automatic Control, 24(6): pp. 843–854, December 1979.
- [4] S.S. Blackman. *Multiple-target tracking with radar applications*, Artech House, Norwood, MA, 1986.
- [5] L.J.H.M. Kester *Multisensor fusion using an adaptive multi-hypothesis tracking algorithm*, Proceedings of SPIE – Volume 5099, april 2003, pp. 164–172
- [6] P.A.M. de Theije, L.J.H.M. Kester, J. Bergmans, *Application of the M⁶T Tracker to Simulated and Experimental Multistatic Sonar Data*, submitted to the 9th International Conference on Information Fusion, July 2006, Florence, Italy.
- [7] S.Coraluppi, D. Grimmett and P.A.M. de Theije, *Benchmark Evaluation of Multistatic Trackers*, submitted to the 9th International Conference on Information Fusion, July 2006, Florence, Italy.
- [8] A. Theil, L.J.H.M. Kester, É. Bossé *On Measures of Performance to Assess Sensor Fusion Effectiveness* Proceedings of the Third International Conference on Information Fusion, 2000, THB5/3–THB5/7.