Data Fusion for Improved Air Picture Generation in Air Defence Systems

Terje Johnsen, Brita Hafskjold
Norwegian Defence Research Establishment
P O Box 25
2027 Kjeller
NORWAY

email: tjo@ffi.no

Svein Fagerlund, Tor Erik Strand & Arnstein Jensen
Kongsberg Defence & Aerospace AS
P O Box 1003
3601 Kongsberg
NORWAY

svein.fagerlund@kongsberg.com

SUMMARY

Fusing data from several sensors will enable the production of more complete and accurate track position and velocity estimates, the tracking of targets for longer times without track number switching, and also increasing the total coverage volume. In this paper, results from a NASAMS study have been used to illuminate important issues in the process from selecting a communication strategy of the sensor measurements/tracks to association and use in a data fusion filter used to predict the state of tracks in a multisensor-multitarget fusion system. The fusion systems sensitivity to registration errors is shown as well as the importance of accurate clocks. A bias estimator is presented that could be used to reduce the effect of bias errors and obtain a more optimal result of the association process and data fusion filters.

1.0 INTRODUCTION

In Norway, netted air defence systems have been operational since the middle of the 1980ies in the NOrwegian Adapted HAWK (NOAH) and later Norwegian Army Low Level Air Defence System (NALLADS) and the Norwegian Advanced Surface to Air Missile System (NASAMS) systems. The air picture is produced by use of input from all radars in the system. A common method for associating tracks produced by different sensors is to use the so-called Reporting Responsibility or R^2 scheme where only the responsible sensor of a track reports on the network. Network bandwidth is always a scarce resource that favours the use of such methods. When networks with higher bandwidth and nodes with increasing computational power are implemented, new and alternative methods should be evaluated for use in air defence systems. This paper will present results from a NASAMS study by Forsvarets ForskningsInstitutt (FFI) and Kongsberg Defence & Aerospace (KDA), where modern alternatives of data fusion have been studied. Results that were found are of general interest and not limited to NASAMS. Strategies, problems and alternative solutions of fusing data from a set of sources are discussed generally, with examples from the NASAMS study.

Ideally, the air picture should consist of tracks uniquely connected to separate targets for as long as one or more reporting sensors can observe each target. The tracks should represent the target manoeuvres with

Johnsen, T.; Hafskjold, B.; Fagerlund, S.; Strand, T.E.; Jensen, A. (2005) Data Fusion for Improved Air Picture Generation in Air Defence Systems. In *Design Considerations and Technologies for Air Defence Systems* (pp. 13-1 – 13-14). Meeting Proceedings RTO-MP-SCI-143, Paper 13. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

correct position and velocity at all times. Real sensors have deficiencies that would reduce the quality of the tracks in various ways depending on e.g. measurement uncertainties, update rates and biases. This and the filtering characteristics implemented in the tracking and fusion system results in an imperfect air picture. In addition, there always exists numerous ways in which measurements could be wrongly associated to a track and no method could guarantee error free behaviour of the fusion-tracking system.

Data fusion could be used with input from similar or dissimilar sensors. In this paper, equal sensor inputs from a set of simulated radars in a NASAMS battery have been used. Data fusion is in this paper restricted to focus on techniques for estimating and predicting the state of targets and do not discuss higher order data fusion related to situation and threat assessment.

To evaluate the performance of different strategies and algorithms, a set of scenarios have been developed. Terrain following cruise missiles and fighter airplanes were used as scenario targets (see example in Figure 1). Scenarios with both dense and more scattered distribution of targets were used.

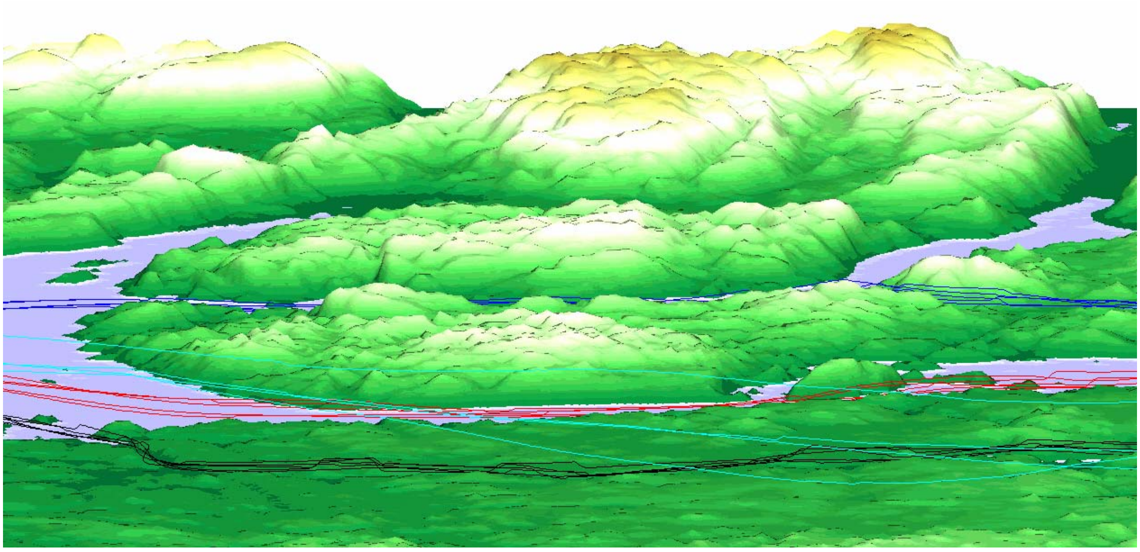


Figure 1: Typical Scenario with Low and High Altitude Targets in a Norwegian Fjord Terrain.

2.0 NETTED AIR DEFENCE SYSTEMS IN NORWAY

In Norway, there are two operating netted air defence systems, NALLADS and NASAMS. This study will focus primarily on a study related to the NASAMS system. The basic elements in NASAMS are the Fire Unit (FU) shown inside the blue dotted line in Figure 2, consisting of a multifunction radar (AN/TPQ-36A or AN/MPQ-64, Sentinel), a Fire Distribution Centre (FDC), three launchers, each with six Advanced Medium-Range Air-to-Air Missiles (AMRAAM), and an Electro Optical (EO)-sensor labelled NTAS in the figure. These units communicate either via cable or radio. The NASAMS Fire Units are usually deployed in a battery configuration of up to four FUs (see Figure 2). The FUs communicate through a battery communication net named the Battery Net Data Link (BNDL).

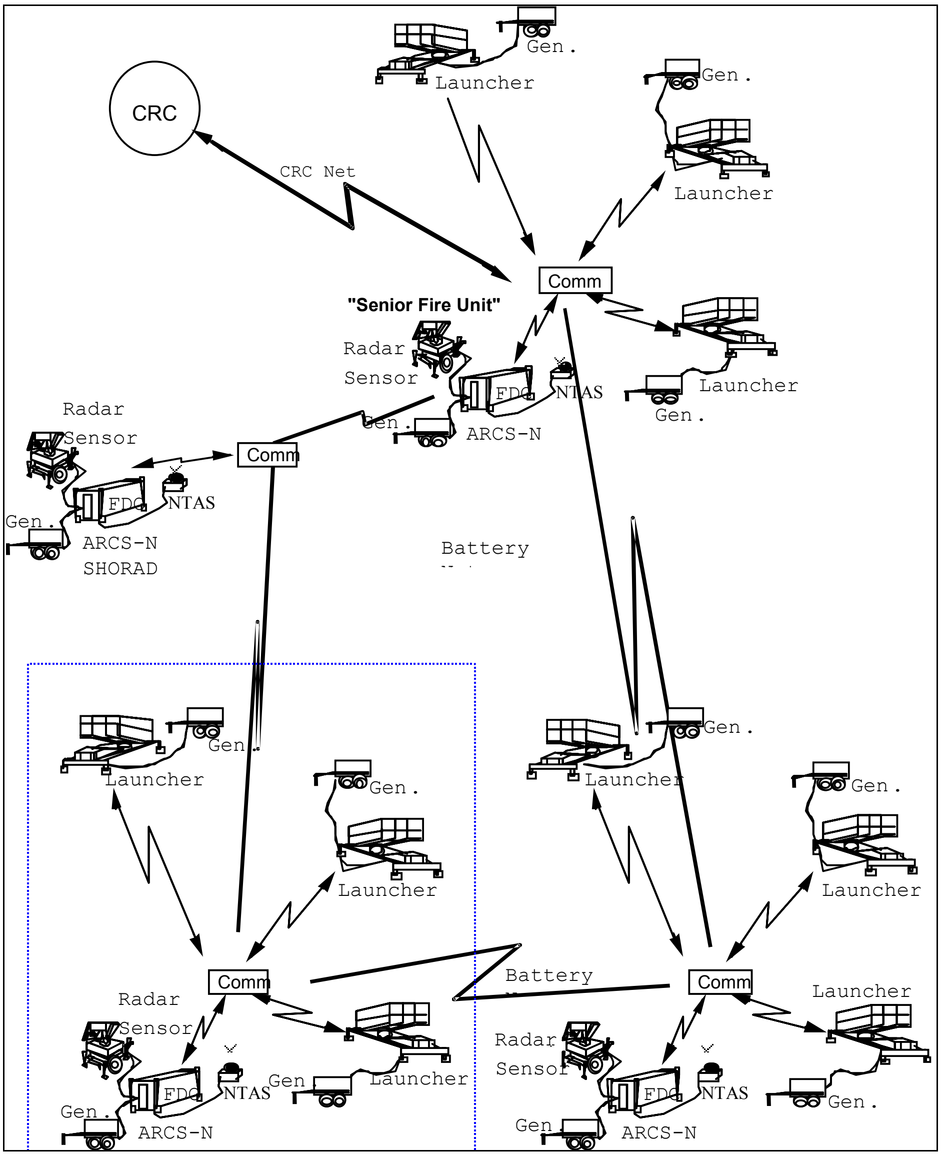


Figure 2: NASAMS Battery Configured with Four FUs.
The units included in one FU are shown within the blue dotted line.

The radar processes the radar data and produces tracks for further processing in the FDCs. Each FDC can produce an air picture based on track information from all radar sensors in the network. The fusion of sensor data will therefore consist of track-to-track association and fusion, and not processing of target plots. The tracking filter in the radars introduces some filtering lag to the radar plots during manoeuvre. If the fusion was on plot level, the lag is removed, but the bandwidth requirements for communicating all plot data to all other fusion centres would increase, and therefore makes this strategy unpractical.

3.0 DATA FUSION OF MULTIPLE SENSORS

There could be many motivations for fusing data from a set of sensors. Obviously, more sensors add more information of targets at a higher rate, which should result in a more complete and accurate track picture. The fusion of input from several sensors produces an air picture covering a larger area than a single sensor, as shown in Figure 3. The presentation of continuous tracks without number switching should span track segments contributed from a set of sensors. Rough terrain and low altitude targets would also require contribution from several sensors to increase the coverage volume and presentation of longer unbroken tracks. Different sensors with different characteristics and probabilities of detection would enhance the final track picture when fused to produce a common air picture.

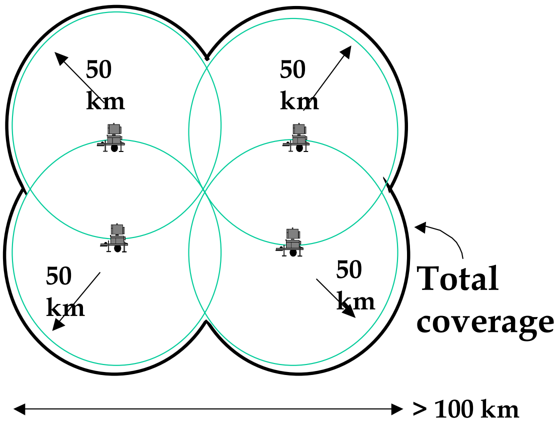


Figure 3: Maximum Coverage Positioning of Radar Sensors.

The accuracy of the measurements/tracks would also be improved by the fusion of data from two or more sensors. To illustrate this effect, we have drawn the error ellipsoid from the covariance matrix in Figure 4. Three different target positions are shown with the radar placed in origin. The measurements or tracks typically have an uncertainty in azimuth and elevation that grows proportionally with distance, while the range error remains constant. This results in an error ellipsoid forming a discos-shape for larger distances. The ellipsoid is shown in three different projections for each measurement position in the figure.

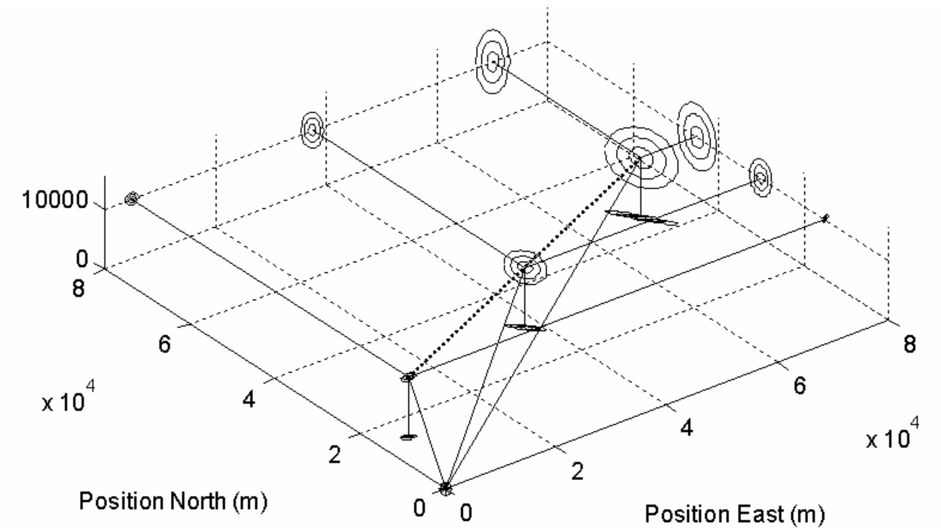


Figure 4: The Geometry of Three Single Radar Measurement Error Ellipsoids in 3D. The range error is very small compared to the errors in angle.

When several radars observe the same target, their error ellipsoids will typically not coincide. The actual overlap is dependent upon geometry. In Figure 5, three different radars labelled R_1, R_2 and R_3 observe the same target. All radars and target positions are located in the same plane. The projections of the error ellipsoids of the measurements/tracks for the radars are shown as the three outer ellipses for each measurement.

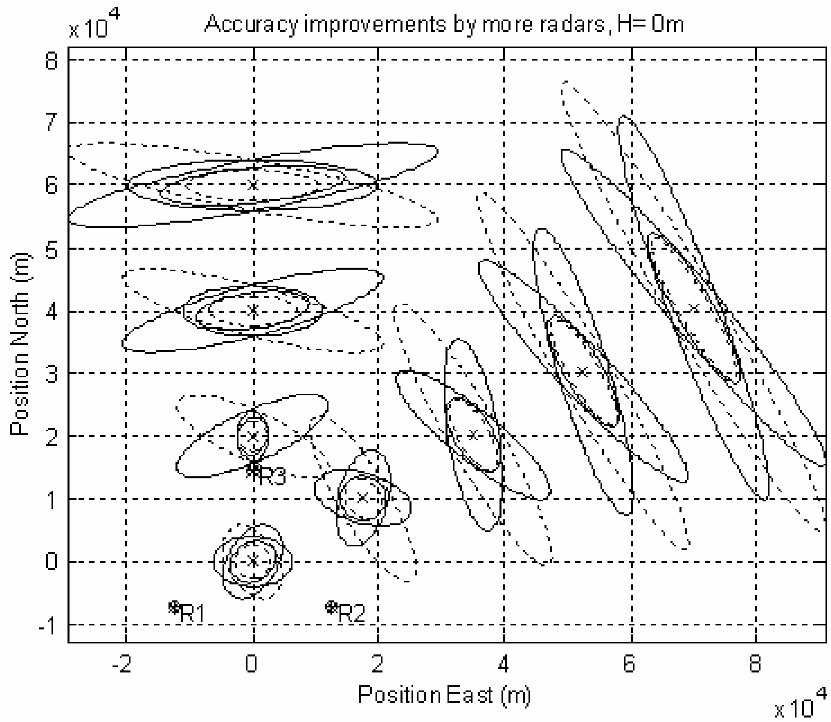


Figure 5: The Accuracy Improvement of Combining the Three Error Ellipses (outer ellipses) is Shown as the Inner Dotted Ellipse. The slightly larger solid ellipse is the combination of R_2 and R_3.

The resulting combined uncertainty can be visualised as the interior of intersections of the uncertainty ellipsoids from each radar measurement, the intersections turned into the closest matching ellipsoid. The smaller dotted ellipsoid is the accuracy of all radar measurements combined, while the solid ellipse do not use R_1. As shown by different measurement positions, the resulting accuracy is geometry dependent.

3.1 Sensor Data

Sensors could record different measures, e.g. range, range rate, bearing, and elevation. Many types of radar such as the Sentinel radar (AN/MPQ-64) in NASAMS process the detections internally, screening much of the temporal clutter and noise data, and produce tracks based on a series of measurements as output. The radar controls the process of initiating, updating, splitting and terminating local tracks. This produces reliable data to the fusion engine at the expense of possible loss or delay of barely detectable targets. Fusion can generally be handled at plot level or track level. Plot level fusion and tracking would require a higher communication bandwidth, which might not be available. Low bandwidth could introduce delays in such a system. In addition, an increase in the processing requirements of the fusion nodes would be anticipated with plot level fusion. In future systems, the processing needs might be practicable for at least a reduced number of sensors. Simulations have been executed for various fusion methods with both plot data and track data as input. Generally, for all methods, the results showed that both a more accurate mean and a lower standard deviation were found for position and velocity when plot data were used.

The radars in the NASAMS system produce tracks that can be fused with other sensor tracks. The ideal solution would be to fuse plot data but this could be expensive to implement in a real system. In the following, radar tracks are used as sensor input to the fusion process.

3.2 Data Fusion Filters

The fusion of all sensor data should result in a more correct picture than the use of the traditional Reporting Responsibility (R^2) scheme, where radar tracks are only associated to the system tracks reported by the radar with best track quality. Due to track loss in difficult tracking environments, delays would be anticipated before responsibility takeover by other sensors. This leads to gaps in track pictures [1].

To overcome such drawbacks and improve the resulting air picture studies of different fusion algorithms were executed. The filtering and prediction of such quantities as track position and velocity can be solved by use of different methods described in a well of textbooks and papers, e.g. [2][3]. In our study, two data fusion methods were investigated in detail; use of Extended Kalman Filters (EKFs) and IMM. Both represent state-of-the art algorithms for target tracking/data fusion. Both algorithms have been used extensively to solve the target-tracking problem in numerous applications. They are both based on mathematical models of the target behaviour and have a number of design parameters that can be tuned to the tracking problem at hand, and that will change the performance of the target tracking.

In the EKF fusion, two separate EKF filters were used; one representing no manoeuvre and a second filter with a model of a manoeuvring target. In the process of switching between the filters, a manoeuvre detection algorithm was implemented, testing for changes in manoeuvre at each track update. The IMM filter in our study was also implemented with two filters. Generally, the IMM could use two or more parallel EKFs processing on the same measurements. Each EKF in the IMM-algorithm has a different mathematical model, for example one model assumes the target is moving along a straight path, and another model assuming that the target is performing a turn (manoeuvre). A study with three parallel filters in IMM has been reported for tracking in an air defence application [4]. Here, the models used were a constant velocity non-manoeuvring model, a general Singer acceleration model and a turn model with constant speed.

Based on sensor observations, the IMM-filter calculates the probability likelihood of how well each model is representing the behaviour of the target given the measurement, and generates the final state vector and covariance matrix as a weighted sum of the outputs of the models. This allows for a dynamic change between models.

Our simulations showed that the IMM filter produced a more correct estimate of target position and velocity, especially during target manoeuvres, than the use of EKF filters alone. This is in agreement with [4].

3.3 Communication Strategies

In netted air defence systems, various communication strategies can be implemented. It is common to differentiate between the two main directions; centralized and decentralized strategies.

3.3.1 Centralized Fusion

Basically, Centralized Fusion means that all the sensor data is transmitted to one central fusion node, which performs the track association/fusion and generates the system tracks. In our system, these system tracks are transmitted back to the other nodes on the network. This ensures a common air picture in all nodes of the network. We implemented two versions of Centralized Fusion, described as version #1 and #2.

3.3.1.1 Centralized Fusion Version #1

The Centralized Fusion version #1 strategy can be described as follows:

- Assume that 4 FDCs, labelled A, B, C and D are participants on the network.

- FDC B, C and D distribute their local radar tracks on the network to FDC A.
- FDC A performs the track association/fusion process to generate the official system tracks of the final Battalion Air Picture, and distributes the results to the other FDCs.
- If FDC A drops out of the network, FDC B takes over the track association/fusion process, and so on.

Since there is only one FDC "master" fusion node in the network at a time, the track picture should be identical in all nodes, given that no system tracks are lost in the network. Clearly, this strategy is vulnerable, since all updated system tracks; associations of local radar tracks to system tracks and covariance matrixes are stored in only one location. If the master fusion node is lost, the node to take responsibility need time to build up a complete air picture from knowledge of the system tracks only.

3.3.1.2 Centralized Fusion Version #2

A variation of the Centralized Fusion strategy that alleviates some of the disadvantages, and places the strategy closer to Decentralized Fusion, is the following:

- Assume that 4 FDCs, labelled A, B, C and D are participants on the network.
- All FDCs distribute their local radar tracks on the network to all other FDCs.
- FDC A performs the track association/fusion process to generate the final Battalion Air Picture, and distributes the result to the other FDCs.
- FDC B, C and D performs the same track association/fusion process in the background, however, the final Battalion Air Picture is received from FDC A.
- Each of FDC B, C and D verifies that their background association/fusion results are the same as the official fusing result generated by FDC A.
- If FDC A drops out of the network, FDC B takes over the track association/fusion process, and so on.

In this case, each FDC is performing data fusion in a "shadow" – process, and if the node that generates the "official" air picture goes off the network, any of the other FDCs can take over the Centralized Fusion responsibility without any abrupt change in the air picture, and without regeneration of the state covariance matrices for the tracks. Exchange of state covariance on the network is neither an issue in this case, except for the case when a new FDC goes on the network. Then, 'the new' FDC needs to get a copy of the state covariance matrix for its "shadow"-tracking process. This strategy is less vulnerable to loss of FDCs than version #1. The use of background processing in all nodes requires the use of equal amount of CPU time in each FDC, but since all nodes could operate as the master fusion node their CPU capacity should be of the same size.

The bandwidth requirement is higher in version #2, since all measurements are transmitted to all other nodes.

3.3.2 Decentralized Fusion

Decentralized Fusion means that each node (FDC) connected to the network sends its local sensor data to all the other nodes, so that all nodes gets all the data and can generate its own air picture independently of each other, but arrive at the same result in a distributed fashion. When implemented three versions of Decentralized fusion are feasible, as described in the following.

3.3.2.1 Decentralized Fusion Version #1

The following steps can describe the strategy:

- Assume that 4 FDCs, A, B, C and D are participants on the network.
- All FDCs distribute their local radar tracks on the network to all other FDCs.
- Each FDC performs the track association/fusion process locally to generate the final Battalion air picture. Since all FDCs operate on the same datasets, the likelihood that they all arrive at the same air picture is high, but not necessarily 100%.
- No additional data is sent on the network.

If one or more FDCs drop out of the network, the remaining FDCs continue as before with the exception that the local radar track data from these FDCs are no longer available.

Since the determination of system track numbers and the association of radar tracks to system tracks will be executed in each FDC, there is a possibility for the (wrongly) establishment of different system track numbers in the different FDCs. Time delays and missing track messages will introduce differences in available data sets.

3.3.2.2 Decentralized Fusion Version #2

A variation of the Decentralized Fusion strategy that copes with the possibility that the different FDCs could arrive at a different air pictures is described in the following.

- Assume that 4 FDCs, A, B, C and D are participants on the network.
- All FDCs distribute their local radar tracks on the network to all other FDCs.
- Each FDC performs the track association/fusion process locally to generate the final Battalion Air Picture.
- One of the FDCs, e.g. FDC A, is defined as the "Master Fuser" on the network. FDC A sends a "Fusion Manuscript", describing how tracks from the different nodes in the network has been associated, out on the network to the other FDCs.
- The FDCs receiving the fusion manuscript, checks that their own fusing process has arrived at the same result as the "Master Fuser", and takes corrective actions if this is not the case.
- If FDC B, C or D drops out of the network, the remaining FDCs continue, as before, with the exception that track data from these FDCs are no longer participating.
- If FDC A (the "Master Fuser") drops out of the network, FDC B takes over as the new "Master Fuser".

The communication from FDC A of the fusion manuscript is added to the distribution of local track data to all other FDCs. Requirements for bandwidth is therefore higher for this decentralized fusion version #2, but for the fusion manuscript only changes in associations need to the transmitted. Assuming few association shifts among the total system tracks from scan to scan, the increase in bandwidth with relation to version #1 is not high.

3.3.2.3 Decentralized "Relay-Pin" – Fusion

The most advanced form of distributed track association/fusion that has been studied was named "Relay-Pin" – fusion. The "Relay-Pin" fusion strategy can briefly be described as follows: Assume the current fusion result for a track is available on the network. The first FDC on the network that has a new observation for that particular track from its local radar fuses its new sensor observation into the track solution, and sends the result out on the network. The next FDC that receives a new sensor observation on the track does the same, and so on. The track goes around as a "relay pin", and it is updated every time there is a new sensor observation on the track.

With this fusion strategy, the final fusion result is distributed, not the radar observations. However, all fusion algorithms also need the State Covariance Matrix for the track to be able to fuse new sensor observations into the track solution and update the track state vector. If the covariance matrices for the tracks were distributed together with the state vector, this would solve the problem, but would also require a lot of additional communication capacity per. track, so this alternative should be disregarded.

Another possibility is to regenerate the state covariance in the FDCs receiving a state vector from the network. This solution would not require additional network capacity. Every time a state vector for a track is received by a FDC from the network, the FDC regenerates the covariance matrix locally, based on the previously locally stored covariance matrix and state vector, and the fact that it knows the location of the last sensor that integrated an observation into the track, and the time it happened. No additional data is sent on the network, only the system track state vector. If one or more FDCs drop out of the network, the remaining FDCs continue as before, without track contributions from the lost FDCs.

This fusion strategy could have one additional advantage over the other decentralized fusion strategies: If there were other data users of the track data traffic on the network, the messages that are sent with the "relay pin" strategy are the final track results, not the local radar tracks, as was the case when using the first two decentralized fusion strategies.

The process of regenerating the covariance matrix locally requires the measurement to first be regenerated before the fusion process is repeated locally as indicated in Figure 6. This process was found to be complex and computer intensive, especially in the IMM case. In addition, the mode probabilities of the filters in the IMM filter must be communicated. Simultaneous updates of the same track by different FDCs would generate two valid track state vectors. The process of handling such cases and the additional CPU and bandwidth requirements led to the conclusion that this algorithm is not suited for use in our system.

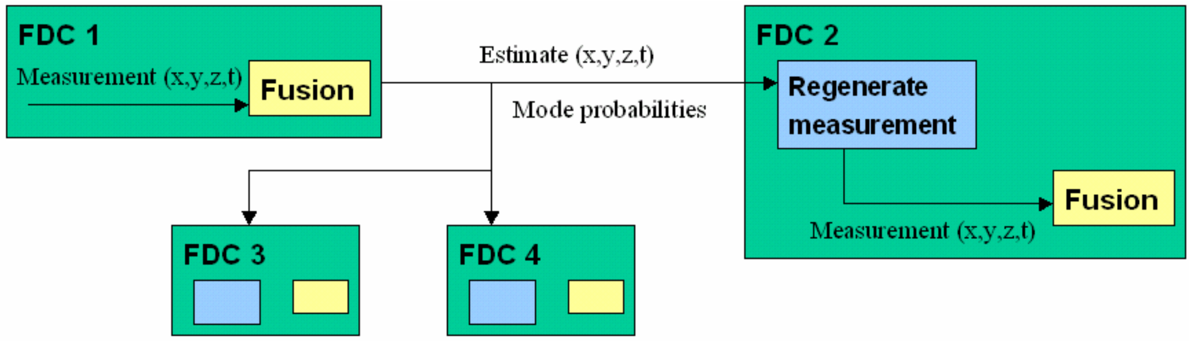
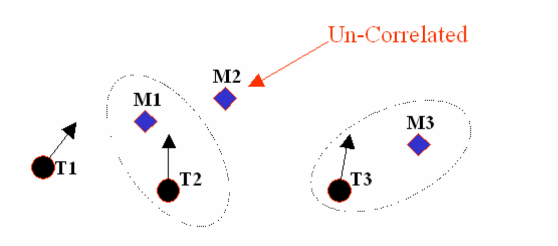


Figure 6: Illustration of Track Parameters on the Network and Processing Requirements in FDCs for Regeneration of Covariance Matrix.

3.4 Association

A very important and often time-consuming process in multisensor data fusion is the process of associating sensor tracks/measurements to system tracks. Two different association algorithms were investigated in simulations; Sub optimal Nearest Neighbour (SNN) and a Global Nearest Neighbour (GNN) algorithm implemented by the effective JVC algorithm [5]. The difference in association for a selected example is shown in Figure 7. Here, the SNN illustration to the left shows how the system tracks T_2 and T_3 correlates to the new measurements M_1 and M_3, leaving the track T_1 without any updating measurement and M_2 as an un-correlated measurement. The illustration of GNN to the right shows how the global approach solves the global minimization of track to measurement distance. In this example, all system tracks were updated with measurements. Results from simulations showed that the GNN algorithm performed best in scenarios where targets are relatively close to each other and with sensor registration error present.



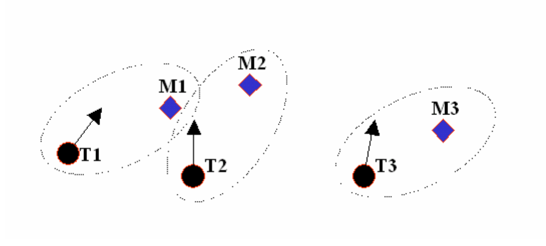


Figure 7: Illustration of Association by SNN and GNN.

3.4.1 JV and JVC Association Algorithms

An efficient algorithm for solving the association of local radar tracks to system tracks is important to make it tractable to implement. In the literature, the JVC algorithm [5], which is an optimisation of the JV algorithm [4], is known to be very fast. In our implementation they produced similar results for the association but JVC was more efficient, using considerably less CPU time. The main difference between the JV and the JVC algorithms are listed in the following:

- The JVC algorithm solves unsymmetrical problems (different number of system tracks and radar tracks). The JV algorithm requires a symmetrical input matrix; hence artificial column or rows must be added to fill this requirement.
- The JVC algorithm uses gating to reduce the number of elements in the compressed cost matrix. Hence the compressed cost matrix used by the JVC algorithm only contains cost values for feasible combinations, while the JV algorithm uses a cost matrix that contains cost values for all combinations, both feasible and infeasible associations.

The differences listed above reduce the number of combinations that the JVC solver must check while searching the minimum global cost. This makes the JVC algorithm the best choice.

3.5 Estimation of Registration Errors

Our knowledge about the position and orientation of each radar sensor in the system will always have constant errors, called registration errors or biases. These errors result in un-modelled errors in the filters and subsequently affect their potential of estimating correctly as well as introducing more wrong associations. Their presence reduces the potential of many of the algorithms mentioned. Such biases may give a saw tooth shape of the fused track, as explained in section 3.5.1. Thus, we need to remove their influence by measuring or estimating these biases. As illustrated in Figure 8, a bias estimator could consist of two distinct parts; a track segment selector that is used to find suitable track segments, see section 3.5.4 and a bias estimation algorithm, discussed in section 3.5.3.

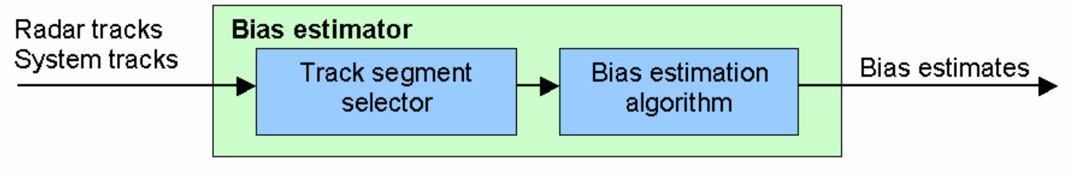
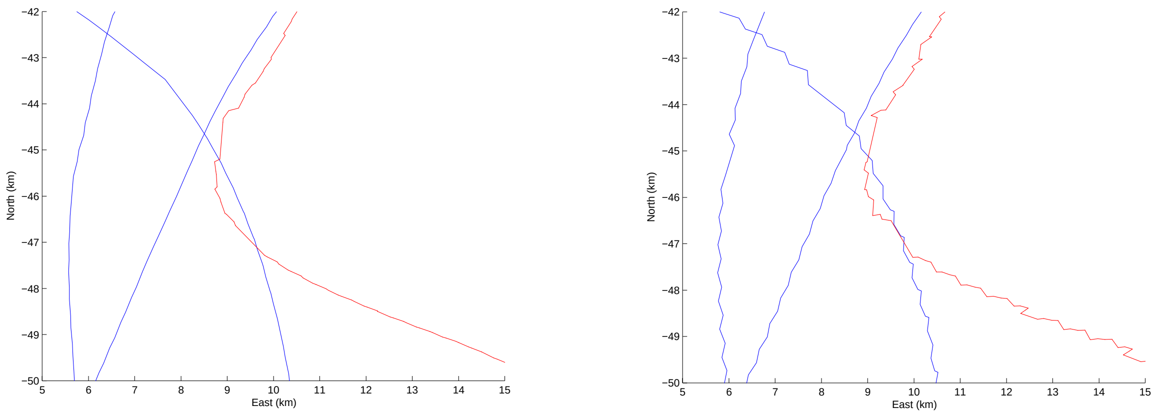


Figure 8: Total Bias Estimator.

3.5.1 Errors in Radar Position and Orientation

The positions of the radars can be found by different means, e.g. by the use of GPS. The sensor positioning will have a constant error in each of the three directions; north, east and height. The orientation of the radars can be represented by three angles; roll, pitch and azimuth. Radar operators typically align the radars to make the roll and pitch angles as small as possible. The azimuth angle can be found by different means, e.g. by a north finding unit. Since the alignment is not perfect, each of the three angles will have a constant error. Bias errors might also change slowly e.g. due to ground instabilities. If one or more of the sensors were on a mobile platform, constant monitoring and correction of biases must take place.

We have found that unknown biases in radar position and orientation lead to large fluctuations in the position for fused tracks, as can be seen from the position plots in Figure 9.



Small bias Large bias

Figure 9: The Result of Radar Position Bias on Fused Target Positioning.

The same scenario has here been simulated with both small and large biases. We observe that the track quality is sensitive to these errors. Different sensors enter their contribution to the system track at different offset positions, leading to the saw tooth shape of system track found for large bias errors in Figure 9 and explained in the illustration in Figure 10. Biases also result in fluctuations in heading for fused tracks, as can be seen from the heading plots in Figure 11. When prediction of future target position is based upon erroneous heading, the probability of loosing the target grows.

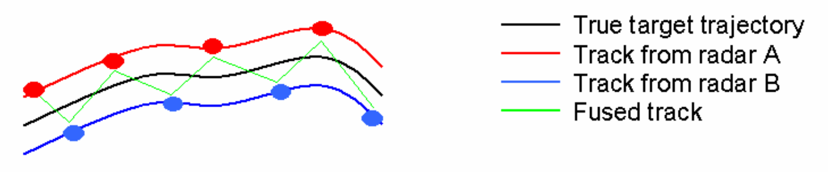
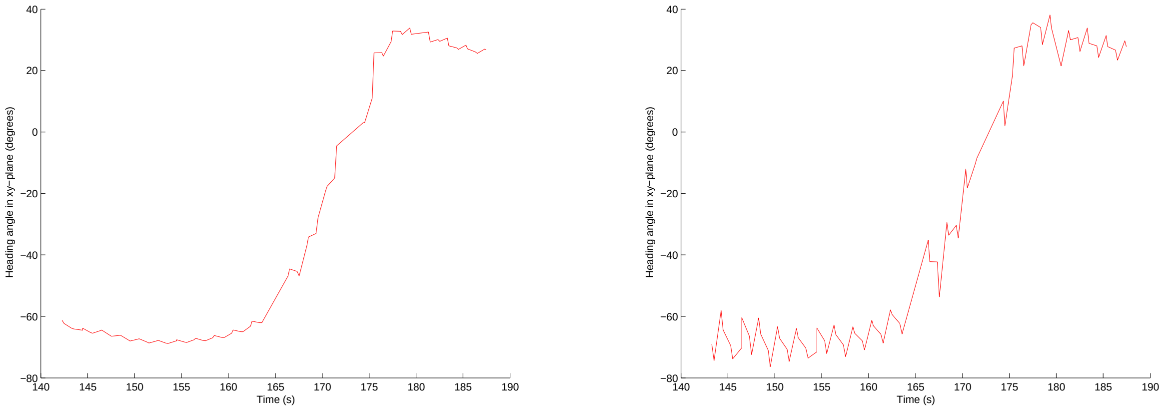


Figure 10: Illustration: Due to Bias, the Fused Track has a Saw Tooth Shape.



Small bias Large bias

Figure 11: Target Heading of the Red Track from Figure 9.

3.5.3 Bias Estimation

For a system with t targets and r radars, a bias estimator has been implemented by a Kalman filter that estimates the following state vector:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p}_{\text{target 1}} & \boldsymbol{v}_{\text{target 1}} & \cdots & \boldsymbol{p}_{\text{target t}} & \boldsymbol{v}_{\text{target t}} & (\boldsymbol{p}_{\text{radar 1}} - \boldsymbol{p}_{\text{radar 2}}) & \cdots & (\boldsymbol{p}_{\text{radar r-1}} - \boldsymbol{p}_{\text{radar r}}) & \boldsymbol{e}_{\text{radar 1}} & \cdots & \boldsymbol{e}_{\text{radar r}} \end{bmatrix}$$

where $\boldsymbol{p}_{\text{target i}}$ is the three dimensional position vector and $\boldsymbol{v}_{\text{target i}}$ is the three dimensional velocity vector of target number i, and $\boldsymbol{p}_{\text{radar j}}$ is the three dimensional position vector and $\boldsymbol{e}_{\text{radar j}}$ is the three dimensional orientation vector of radar number j. Thus, the Kalman filter estimates the orientation and the relative position of the radars, in addition to the position and velocity of the targets. The biases must be removed before association and fusion of the data. The observability of the bias estimator has been found for one simulation to be as shown in Figure 12. Here, a minimum of three targets observed by three radars is found to fully observe the parameters. For this simulation, there were five slightly curved tracks observed by four radars for ten consecutive scans. Radars and flight tracks had all good separation of each other in this scenario. This simulation is only the first attempt, and a more systematic study of radar and track geometries should be carried out to determine the capacity of this bias estimator for various scenarios.

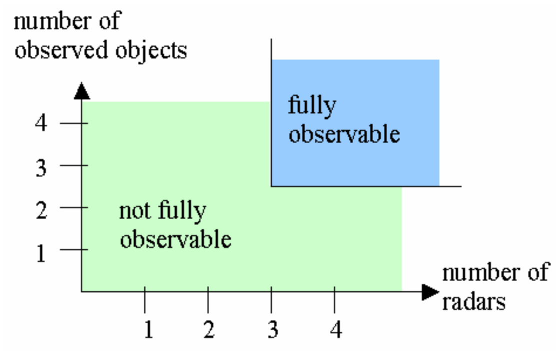


Figure12: Observability for the Bias Estimator, for a Simulation with Five Targets and Four Radars.

To illustrate how it is possible to estimate the constant azimuth angle bias for a set of radars, a simple example is presented, where the elevation biases of the radars are neglected. By using measurements of the same object from a minimum of three radars, the azimuth angle can be found. This is illustrated in Figure 13, where we assume a possibly very large azimuth error, but a more correct range measurement in each of the radars. We also assume that the constant radar position error is very small compared to the distance between the radars. From the measurement from radar 1, we know that the observed object is positioned along the blue circle. After having received a measurement also from radar 2, we know that the observed object is placed in one of the cross-points between the red and the blue circle. When we receive the measurement from radar 3, we find the position of the observed object to be in the cross-section of the three circles. For each of the radars, we know the assumed (but wrong), and the correct azimuth angles to the observed object. The constant azimuth error can thus be estimated and corrected for in the system.

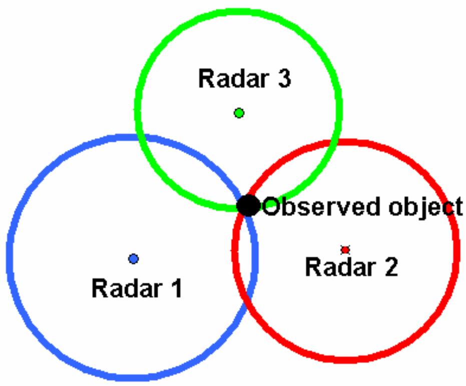


Figure 13: Simple Illustration of using Observations from Three Radars to find Azimuth.

3.5.4 Track Segment Selection

The track segments to be sent into the bias estimation algorithm should be carefully chosen. Some of the requirements for suitable tracks are:

- The association between the radar track and the system track must be correct.
- The tracks should be as straight as possible (e.g,. they should have slow curvatures), to minimize the target positioning error.
- The track segment should have measurements from at least two radars

3.6 Timing Errors

Measurements communicated over the network could suffer from latencies related to the network and the nodes in the network. In our study, small and large timing errors were simulated for measurements. The results showed that timing errors result in track behaviour that could look somewhat similar to what was observed for bias errors. Time tagging of measurements/tracks with synchronized clocks of high accuracy is therefore important. This could for instance be implemented by the use of GPS, that typically has a reported uncertainty of 100ns for many receivers. Alternatively, network and processing delays should be known with high precision.

4.0 CONCLUSION

The fusion of data from a set of distributed radars result in an improved air picture. Coverage volume is increased, tracks could be formed that remain unbroken over a larger area, and the accuracy of the estimated track position and velocity could be improved. The geometry of the combined error ellipsoids would generally be reduced by input from several sensors. State of the art data fusion filtering with both EKF and IMM was used in a simulation study, where IMM gave best overall performance. Different communication strategies have been discussed for centralized and decentralised methods. The important process of associating tracks from different sensors to system tracks has been simulated by use of SNN and GNN methods. GNN algorithms gave the least number of wrong associations, but they are CPU intensive and require effective algorithms such as the implemented JVC algorithm to be tractable in a real system. To get the optimal result from use of data fusion and tracking with the algorithms above, registration errors as well as timing errors should be minimized. Data fusion has been shown to be sensitive to the presence of these errors. A bias error estimator that uses an EKF filter to estimate position and orientation biases of sensors have been presented. To minimize timing errors in data tagging and their influence on the fused system tracks, the use of GPS or similar synchronized clocks are strongly advised.

5.0 REFERENCES

[1] Y. Bar-Shalom, W.D. Blair, “Multitarget-Multisensor Tracking: Applications and Advances, Volume III, Artech House, Bosten, 2000.

[2] S. Blackman, R. Popoli, “Design and Analysis of Modern Tracking Systems”, Artech House, Boston, 1999.

[3] D.L. Hall, J. Llinas, “Handbook of Multisensor Data Fusion”, CRC Press, London, 2001.

[4] M. Busch, S. Blackman, “Evaluation of IMM filtering for an air defence system application”, Proceedings- SPIE The International Society for Optical Engineering, Vol 2561, pp. 435-447, 1995.

[5] O.E. Drummond, D.A. Castanon, M.S. Bellovin, “Comparison of 2-D Assignment Algorithms for Sparse, Rectangular, Floating Point, Cost Matrices”, J. of the SDI Panels on Tracking, Institute for Defense Analysis, Alexandria, VA, No. 4, pp. 4-81 to 4-97, 1990.

[6] R. Jonker, A. Volgenant, “A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems”, J. Computing, Vol 38, pp. 325-340, 1987.