Proceedings of the International MultiConference of Engineers and Computer Scientists 2011 Vol II,
IMECS 2011, March 16 - 18, 2011, Hong Kong

Multi-Censor Fusion using Observation Merging with Central Level Architecture

D. M. Akbar Hussain*, Zaki Ahmed, M. Z. Khan and Andrea Valente *

Abstract—**The use of multiple sensors typically requires the fusion of data from different type of sensors. The combined use of such a data has the potential to give an efficient, high quality and reliable estimation. Input data from different sensors allows the introduction of target attributes (target type, size) into the association logic. This requires a more general association logic, in which both the physical position parameters and the target attributes can be used simultaneously. Although, the data fusion from a number of sensors could provide better and reliable estimation but abundance of information is to be handled. Therefore, more extensive computer resources are needed for such a system. The parallel processing technique could be an alternative for such a system. The main objective of this research is to provide a real time task allocation strategy for data processing using multiple processing units for same type of multiple sensors, typically radar in our case.**

Keywords: Target Tracking, Data Fusion, Sensor level, Parallel Processing.

1 Introduction

Basically multi-sensor system is a net formed by number of different or similar types of sensors such as radar, sonar, infrared sensor and cameras etc. Multiple sensors can be deployed as co-located and non-co-located systems. The co-located system for example can be described by a warship which is mounted with different sensors; radar, sonar, infra-red and cameras. The netting of radars/sensors is an example of non-co-located sensors which is considered in this research. The first step in developing a multiple sensor system is the choice of the architecture of the system for the data processing. In fact only two architectures [1, 2, 3] are available for the data processing, firstly the distributed architecture which consists of a computer at each radar/sensor site performing the tracking functions as an independent single target tracking system. The tracks formed at each radar site are then sent to a command (computer) site where a single multi-sensor track for each target is maintained. Secondly, the centralized architecture consist of a single (main) computer which receive the observation/measurement data from the different sensor sites. These observations are then processed to obtain a single multi-sensor track for each target. There is a possibility of a third architecture which could be combination of the two architectures described above. In text the distributed architecture is sometimes called as sensor level tracking and the centralized architecture is known as central level tracking and we are using and investigating the later description for the architecture, similar kind of work has also been investigated in reference [4].

*Manuscript submitted January, 2011 Dr. M. Akbar Hussain is member of IEEE, IDA, IAENG, Email: akh@es.aau.dk and Dr. Andrea Valente, Email: av@es.aau.dk, both work at the Department of Electronic Systems Aalborg University, Niels Bohrs Vej 8, 6700 Esbjerg, Denmark. Dr. Zaki Ahmed is working as a chief Scientist at PILO (Pakistan Institute of Laser and Optics). Email: zaki786@ieee.org.

1.1 Sensor Level Fusion

In this setup each sensor is coupled with an independent tracking system, which is responsible for track initialization, data association, track prediction and track update. Each tracking system individually performs the above mentioned tracking functions and finally all the updated tracks are sent to a common place (computer) for fusion of tracks to obtain a single global track for each target. Fundamental to the problem of combining sensor level tracks is determining whether two tracks from different tracking systems potentially represent the same target. Consider two track estimates $\hat{\underline{x}}_i$ and $\hat{\underline{x}}_j$ with corresponding covariance matrices $\mathbf{P}_i$ and $\mathbf{P}_j$ from the two tracking systems i and j. If the estimation errors on the two estimates are assumed to be uncorrelated then the common test statistics [5]

$$\mathbf{d}_{ij}\ \mathbf{P}^{-1}\ \mathbf{d}_{ij}^T\ \leq\ D_{th} \tag{1}$$

can be used to decide whether the two estimates are from the same target. Where

$$\mathbf{d}_{ij} = (\hat{\underline{x}}_i - \hat{\underline{x}}_j) \tag{2}$$

and

$$\mathbf{P} = \mathbf{P}_i + \mathbf{P}_j \tag{3}$$

Because $\mathbf{d}_{ij}$ is assumed to have the Gaussian distribution, therefore equation 1 will have a χ^2 distribution with the number of degrees of freedom equal to the dimension of the estimate vector. Therefore, when equation 1 using a threshold D_{th} obtained from a χ^2 distribution is satisfied

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

then the two estimates represents the same target. When the estimates are determined to be from the same target, they are combined. The combined estimate vector $\underline{\hat{x}}_c$ which minimizes the expected error is

$$\underline{\hat{x}}_c = \underline{\hat{x}}_i + \mathbf{C}(\underline{\hat{x}}_i - \underline{\hat{x}}_j) \tag{4}$$

where

$$\mathbf{C} = \frac{\mathbf{P}_i}{\mathbf{P}_i + \mathbf{P}_j} \tag{5}$$

When the estimation errors of the two estimates are not independent the covariance defined above in equation 3 for the two estimates is not strictly valid because of the error correlation between the two sensor estimates. Bar-shalom [6] has proposed a statistical test, to account for the correlation between the two estimates, this method is summarized from references [1, 7]. In this method the difference between the two estimates given by $\mathbf{d_{ij}}$ is normalized by the covariance;

$$\begin{aligned}\mathrm{E}[\mathbf{d}_{ij}\ \mathbf{d}_{ij}^T] &= E[\{(\underline{\hat{x}}_i - \underline{x}) - (\underline{\hat{x}}_j - \underline{x})\}\{(\underline{\hat{x}}_i - \underline{x}) - (\underline{\hat{x}}_j - \underline{x})\}^T] \\ &= \mathbf{P}_i + \mathbf{P}_j - \mathbf{P}_{ij} - \mathbf{P}_{ij}^T \end{aligned} \tag{6}$$

where $\underline{x}$ is the true state (noise free) of the two estimates and

$$\mathbf{P}_{ij} \approx\ E[(\underline{\hat{x}}_i - \underline{x})\ (\underline{\hat{x}}_j - \underline{x})^T] \tag{7}$$

reflects the correlation between the two estimates. Therefore the new proposed (by Bar-Shalom) test statistics to check two estimates represent the same target is

$$(\underline{\hat{x}}_i - \underline{\hat{x}}_j)\ (\mathbf{P}_i + \mathbf{P}_j - \mathbf{P}_{ij} - \mathbf{P}_{ij}^T)^{-1}\ (\underline{\hat{x}}_i - \underline{\hat{x}}_j)^T \le D_{th} \tag{8}$$

When the estimates are determined to be from the same target they are combined using equation 4 and the value of $\mathbf{C}$ is now given as

$$\mathbf{C} = \frac{(\mathbf{P}_i -\ \mathbf{P}_{ij})}{\mathbf{P}_i + \mathbf{P}_j - \mathbf{P}_{ij} - \mathbf{P}_{ij}^T} \tag{9}$$

Initially the cross covariance matrix $\mathbf{P}_{ij}$ is assumed zero that is

$$\mathbf{P}_{ij}(0/0) = 0 \tag{10}$$

after that for $n > 0$ the values of $\mathbf{P}_{ij}(n/n)$ are computed recursively by the following relationship

$$\mathbf{P}_{ij}(n/n) = \mathbf{A}_i(n)\ \mathbf{F}(n-1)\ \mathbf{A}_j^T(n) \tag{11}$$

where

$$\mathbf{A}_i(n) = \mathbf{I} - \mathbf{K}_i(n)\ \mathbf{H} \tag{12}$$

$$\mathbf{A}_j(n) = \mathbf{I} - \mathbf{K}_j(n)\ \mathbf{H} \tag{13}$$

$$\mathbf{F}(n-1) = \mathbf{\Phi}\ \mathbf{P}_{ij}(n-1/n-1)\ \mathbf{\Phi}^T + \mathbf{\Gamma}\ \mathbf{Q}\ \mathbf{\Gamma}^T \tag{14}$$

Matrices $\mathbf{\Phi}, \mathbf{K}, \mathbf{H}, \mathbf{Q}$ and $\mathbf{\Gamma}$ are the same State Transition, Kalman Gain, Measurement, Covariance of Measurement Noise and Excitation respectively used for standard Kalman filter [8]. It is evident that if the number of sites are more than two or if more than two tracks represent the same target the above procedure will be quite complex due to the number of calculations involved. The above described procedure has one big advantage, that is apart from the straight line motion target path, it can also be used for maneuvering target as well. The target estimates strictly for targets moving in a straight line path from two tracking systems i and j can also be combined using the following relationships [2]

$$\underline{\hat{x}}_c = \mathbf{P}_c\ (\frac{\underline{\hat{x}}_i}{\mathbf{P}_i}\ +\ \frac{\underline{\hat{x}}_j}{\mathbf{P}_j}) \tag{15}$$

where

$$\mathbf{P}_c = [\frac{1}{\mathbf{P}_i}\ +\ \frac{1}{\mathbf{P}_j}]^{-1} \tag{16}$$

1.2 Central Level Fusion

In this setup the measurements from different sensor sites are received at a central system as shown in figure 1. In contrast to a sensor level tracking system initialization of tracks, data association, track prediction and track update is processed only at a central place (computer). The data processing is done by considering all the information (measurements) available from all the sensors. The track loss rate and mis-correlations in central level tracking are fewer than in the case with sensor level tracking. Also more accurate tracking should be expected if all the data (measurements) are processed at the same place. A target track that consists of measurements from more than one sensor should be more accurate than the track which could be established on the partial data received by the individual tracking system. Finally, the approach whereby all data are sent directly to the central processor should, in principle, lead to faster and efficient computation. The overall time required to develop sensor level tracks and then to combine these tracks is generally greater than the time required for central level processing of all data at once [1]. On the other hand branching of tracks may occur in a central level tracking if similar tracks are not merged due to some reasons and multiple tracks may also be initiated at the time of track initialization. Also if the data from one of the sensor is degraded it will effect the central level tracks. In this study a straight line model for the target dynamics have been assumed, therefore using the same technique given in eqns. 15 and 16, the combined measurement vector and its covariance are obtained instead of the combined estimate vector and the covariance of the estimate. Consider for example the two measurement vectors $\underline{z}_i$ and $\underline{z}_j$ with the corresponding measurement error covariances $\mathbf{R}_i$ and $\mathbf{R}_j$ from the

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

two sensor sites, then the combined measurement vector $\underline{z}_c$ and the combined measurement error covariance matrix are

$$\underline{z}_c = \mathbf{R}_c \left(\frac{\underline{z}_i}{\mathbf{R}_i} + \frac{\underline{z}_j}{\mathbf{R}_j}\right) \quad (17)$$

where

$$\mathbf{R}_c = \left[\frac{1}{\mathbf{R}_i} + \frac{1}{\mathbf{R}_j}\right]^{-1} \quad (18)$$

The approximate statistics given by equation 1 is now modified as to test the nearness of two measurements instead of the estimates that is

$$(\underline{z}_i - \underline{z}_j)\ \mathbf{R}_a^{-1}\ (\underline{z}_i - \underline{z}_j)^T \leq D_{th} \quad (19)$$

where

$$\mathbf{R}_a = \mathbf{R}_i + \mathbf{R}_j \quad (20)$$

When the two measurements using the above statistical test equation 19 are determined to be from the same target they are merged using equations 15, 16.

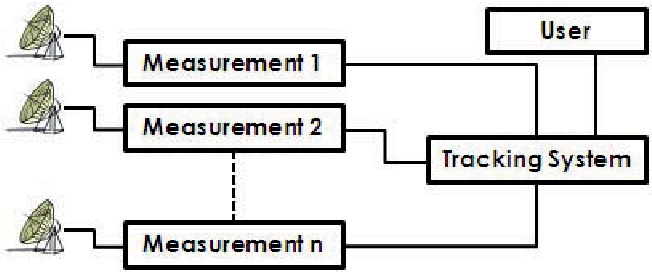

Figure 1: Central Level Architecture

2 Implementation

In this study central level architecture is simulated on a network of computers for the measurement to measurement fusion. For simplicity only two sensors are considered for the system typically sensors provide data in polar coordinates that is the range and bearing of the targets. The tracking however, is performed in Cartesian coordinates so the data is transformed from polar to Cartesian coordinates. The dynamics of the target considered in this study are modeled as straight line motion target. The data from the host computer is sent in the form of array called the measurement vector to the other processing units. The first processor which is directly connected with the host computer has seven modules/processes namely merge measurements, initialization of tracks, distribution of data (measurements and tracks), state estimation update, local similarity, storage of tracks and unused measurements and finally sending of track estimates to host. Merging of measurements and data distribution modules are explained in the following paragraphs.

2.1 Merging of Measurement

After receiving the two measurement vectors corresponding to the two radars the merge measurement task first of all sets the used/unused flag of each measurement in the two measurement vectors to 0.0, then it starts performing the statistical test described in equation 19 for each measurement pair in the two measurement vectors. For a successful test the two measurements are merged using eqns. 17 and 18 provided the used/unused flags of each measurement is 0.0. After that the used/unused flags of the two merged measurements are set to 1.0 which indicates that these measurements have been used in merging. Therefore a measurement from a sensor (1) can be merged only once with another measurement of sensor (2), this option was used to avoid merging of a single measurement from sensor (1) with a number of measurements in the second sensor (2) when multiple measurements from other targets occurs in the same neighborhood. For example, consider a case of four crossing targets and assume that the used/unused flag for the two measurement vectors are not set to 1.0, when these measurements are merged. As long as the targets are distinct (separate from each other) there will be no problem, because only the two corresponding measurements from a common target will be merged. However, consider the situation at the time of crossing, the merge measurement task will take the first measurement from sensor (1) and starts performing the statistical test given in equation 19 with every measurement in the second sensor (2) and because the measurements are very close to each other therefore possibly every measurement of sensor (2) will be likely candidate for merging with the first measurement of sensor (1). But if the used/unused flags are set to 1.0 after each successful comparison it will prevent merging of the same measurement with another measurement. However, the setting of used/unused (1.0) flag do not guarantee for the correct merging of two measurements when ambiguity occurs, but probably provides equal chance to other measurements belonging to the two sensors for merging. For situations when the two measurements from the same target do not satisfy the statistical test equation 19 both measurements from the two sensors are kept, this means track splitting will occur but the similarity criterion should take care of such situations. Finally all the merged measurements are stored in a new measurement vector.

2.2 Distribution of Tracks and Observations

For the distribution of observations and tracks among the different processors an intelligent procedure described in reference [4] is used. In which the new measurement vector is checked for data ambiguity and if there is no ambiguity data distribution task divides the number of tracks among the available processors as equally as possible and sends all measurements of the new measure-

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 1: Parameters for the Two Radars

Parameters	Values
Initial position of radar 1 (X_{p1}, Y_{p1})	(20.0, -30.0)
Initial velocity of radar 1 (V_{p1})	0.0 Km/sec
Initial position of radar 2 (X_{p2}, Y_{p2})	(-20.0, -30.0)
Initial velocity of radar 2 (V_{p2})	0.0 Km/sec
Probability of Detection (1 & 2)	1.0
Scan Sector w.r.t y-axis (1 & 2)	$\pm 90^0$
Bearing Resolution (1 & 2)	5.0^0
Range Resolution (1 & 2)	0.02 Km
Clutter Density (1 & 2)	0.0
Maximum Range (1 & 2)	50.0 Km
Minimum Range (1 & 2)	5.0 Km
Scan Interval (1 & 2)	1.0 sec.
Range variance σ_r^2 (1 & 2)	0.001 K_m^2
Bearing variance σ_θ^2 (1 & 2)	0.004 $Radians^2$

ment vector to all the processors. In the case of ambiguity, measurements of the new measurement vector are divided as equally as possible to the available processors and all the tracks are sent to every processor. After the distribution of data, the state estimation update is done using the standard Kalman filter and after updating, tracks are compared to eliminate similar tracks. Finally, last module store the tracks from all processors and also sends the position estimates to the host computer for real time display.The performance of the multi-sensor central level (measurement measurement fusion) algorithm implemented is investigated by considering the speed up achieved using multiple processing units as parallel processors. Therefore, simulations were performed by considering typical scenarios and the processing time with a single as well as with four processing units were obtained for comparison. Initial parameters for the two radars to simulate the data for targets are given in table 2.2. A number of scenarios were considered to test the proposed algorithm, each simulation was performed 5 times by generating data with 5 random seeds. A sample scenario is shown in figure 2. The processing time with a single and a network of 4 processors was obtained for these scenarios and it was discovered that a speed up of over 50 % is achieved with the current setup of architectures having 4 processors.

3 Summary

In this investigation a multi-sensor system using two similar kinds of radars was implemented on a network. The main object was to demonstrate that parallel architecture can be used for a real time development of a multi-sensor system. Although a very simple approach was used for the implementation but in a similar way a more general data association procedure can be developed for the fusion of data coming from different type of sensors. The merging of measurements instead of tracks is better approach when crossing target scenarios are under consideration, because if the later approach is used the merging process at the crossing point can merge all the target tracks into one global track which can lead to track instability (loss). The simulations have shown that a reasonable speed up of upto 50 % is achievable.

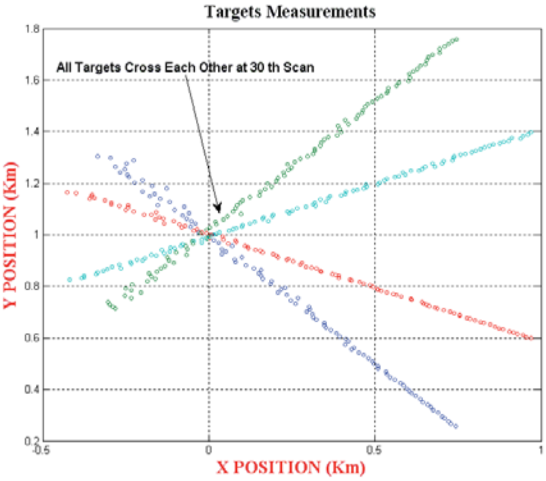

Figure 2: Example Scenario

References

[1] S. S. Blackman. Multiple target tracking with radar applications, 1986 publisher: Academic press inc.

[2] A. Farina and F. A. Studer. Radar data processing volume 2, 1986 research studies press ltd.

[3] David L. Hall and Sonya A. H. McMullen. Mathematical techniques in multisensor data fusion, 2004 artech house inc, 685 canton street, norwood ma 02062 usa.

[4] D. M. Akbar Hussain. Some implementations of multiple traget tracking algorithms on transputers, phd thesis 1991, uk.

[5] Singer R. A. and Stein J. J. An optimal tracking filter for processing sensor data of imprecisely determined origin in surveillance systems" prods. of the 1971 ieee conference on decision and control, dec. 1971, 171 - 175, miami beach, fl usa.

[6] Y. Bar-Shalom. On the track to track correlation problem, ieee transaction on automatic control, ac-26, april 1981, pp 571-572.

[7] Y. Bar-Shalom and T. E. Fortmann. Tracking and data association, 1988 publisher: Academic press inc.

[8] A. W. Bridgewater. Analysis of second and third order steady state tracking filters, agard conference proceedings no. 252.

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)