Sādhanā Vol. 29, Part 2, April 2004, pp. 205–216. © Printed in India

Cluster-based centralized data fusion for tracking maneuvering targets using interacting multiple model algorithm

V VAIDEHI*, K KALAVIDYA, and S INDIRA GANDHI

Department of Electronics Engineering, Madras Institute of Technology, Anna University, Chennai 600 044, India
*e-mail: vaidehivijay@hotmail.com

Abstract. The interacting multiple model (IMM) algorithm has proved to be useful in tracking maneuvering targets. Tracking accuracy can be further improved using data fusion. Tracking of multiple targets using multiple sensors and fusing them at a central site using centralized architecture involves communication of large volumes of measurements to a common site. This results in heavy processing requirement at the central site. Moreover, track updates have to be obtained in the fusion centre before the next measurement arrives. For solving this computational complexity, a cluster-based parallel processing solution is presented in this paper. In this scheme, measurements are sent to the data fusion centre where the measurements are partitioned and given to the slave processors in the cluster. The slave processors use the IMM algorithm to get accurate updates of the tracks. The master processor collects the updated tracks and performs data fusion using 'weight decision approach'. The improvement in the computation time using clusters in the data fusion centre is presented in this paper.

Keywords. Interacting multiple model algorithm; data fusion; target tracking; cluster-based parallel processing solution.

1. Introduction

Radar antenna in Track While Scan (TWS) system scans a predetermined search volume. Depending upon its scan rate it gets returns from the region of its scan coverage. A computer is given the data coming from the radar for tracking the targets and associating new measurements with targets. The objective of multi-target tracking (MTT) (Blackman 1986) is to partition the sensor data into sets of observations or tracks. If there is no uncertainty regarding which measurement belongs to which target, all tracks can be processed independently. But in case of track crossing or merging scenarios, data association techniques should be used to update each track with the corresponding measurement. Once tracks are formed and confirmed, the number of targets can be estimated and quantities such as target velocities and accelerations can be computed for each track.

Data fusion technique combines data from multiple sensors to achieve improved accuracy and more specific inferences than that could be achieved by the use of a single sensor

(Bar-Shalom & Xiao-Rong 1993). Tracking and fusion with multiple sensors has attracted a great deal of attention recently (Liggins *et al* 1997). It deals with integration and correlation of data from various sources to arrive at an overall assessment of the situation. But there is uncertainty in associating various sensor data obtained in different time instances and locations.

Multiple sensor fusion algorithm for tracking applications combines measurements from multiple sensors in a consistent manner. There are many techniques for sensor fusion. Some of them are: Least squares methods, Dempster–Shafer method (Blackman 1986), Bayesian methods (Bar-Shalom & Xiao-Rong 1993), joint probabilities data association (Bar-Shalom & Xiao-Rong 1993), fuzzy logic, neural networks etc. The two types of architectures in which processors may be employed in performing fusion are: Distributed architecture, centralized architecture (Farina & Studer 1986). The distributed architecture is characterized by the use of a computer at each site performing the tracking function on the measurements of single radar. The monoradar tracks are then transmitted to a single data-processing centre that combines them in order to establish a single multiradar track for each target.

The centralized architecture is characterized by the use of a single data processor to which radar plots are transmitted from the sites instead of tracks. These measurements are processed so as to again obtain a single multiradar track for each target. Fundamental issues to be addressed in centralized architecture include the need of high processing capability in the fusion centre in order to produce fused outputs before the arrival of next set of measurements, transmission of plots to data fusion centre with minimum communication overhead. In this paper, building a data fusion system in centralized architecture for target tracking application using ‘clusters’ (a cluster by definition is a type of parallel or distributed processing system, which consists of a collection of interconnected stand-alone heterogeneous computers working together as a single, integrated computing resource) is proposed. This centralized architecture consists of a cluster having a master node and slave nodes. The master allocates necessary data to the slave nodes for tracking. The master collects the tracked results given by the slaves and it does the fusion of tracks. Figure 1 shows the proposed cluster-based centralized architecture.

This paper is organized as follows: Section 2 discusses the multitarget tracking using IMM, section 3 presents the cluster-based solution for centralized data fusion, section 4 presents the parallel algorithm for tracking and data fusion, section 5 presents the analytical evaluation of performance of centralized and distributed data fusion and section 6 presents the results and discussion.

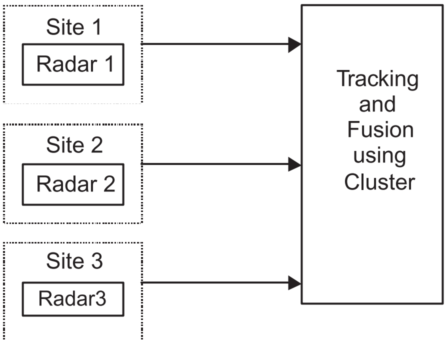

Figure 1. Cluster-based centralized architecture for multisensor data fusion.

2. Multitarget tracking using interactive multiple model

Elements of multi target tracking are gating, data association and correlation, track initiation, deletion, filtering and prediction (Blackman 1986). Gating is a correlation algorithm used to decide if an observation belongs to a previously established target or to a new target. A spherical gate is considered with the predicted position as the centre of the sphere. If the next measurement falls inside this gate it becomes one of the candidates of updation of that track. If the observation does not satisfy the gates of existing tracks it is used to initiate new track. When multiple observations fall within the same gate or when an observation falls within the gates of more than one target then data association technique is used to associate observations with tracks. There are many approaches for data association like nearest neighbour approach, all neighbour approach.

Observations not assigned to existing tracks are used to form new tentative tracks. Once a tentative track is formed confirmation logic is usually needed because the probability of a single observation being from an extraneous source is too high for immediate confirmation. A track not updated becomes degraded and must be deleted. Filtering is performed after associating measurements with corresponding targets using filtered values. Predictions are made to the time, when the next data scan is to be received.

Blom & Bar-Shalom (1988) have used a bank of filters in their IMM approach to accommodate various possible target trajectory patterns and conditions. These filters interact with each other probabilistically and adjust their parameters to provide optimum results. The system output is got by a weighted average of the filters. A simple IMM filter for target tracking consists of two models of different dimensions as shown in figure 2, a second-order (velocity)

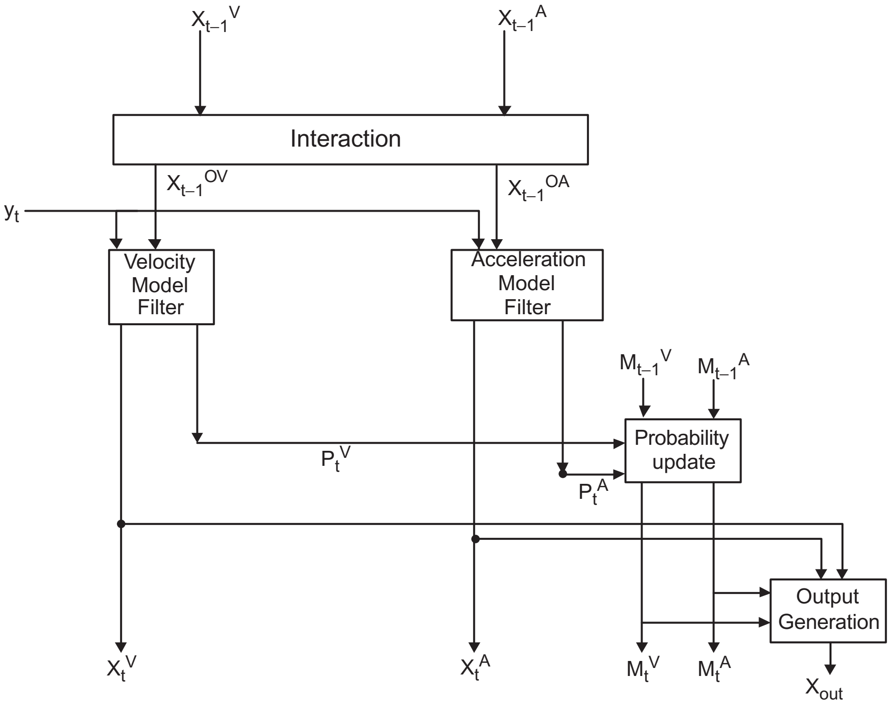

Figure 2. IMM filter.

model for the quiescent mode of the target, a third-order (acceleration) model for the maneuvering mode with appropriate process noise levels, a model probability evaluator, and an estimate combiner at the output of the filters, where each filter uses a mixed estimate at start of the cycle.

The working of the IMM algorithm involves interaction, prediction, model probability update, filtering and output generation. At time t, during interaction step, before the measurements y_t are received, mixed state vector X^{Oi}, and mixed error covariance P^{Oi} are obtained using the previous filtered estimate $X^i_{t-1/t-1}$, and the filtered error covariance $P^i_{t-1/t-1}$, where i represents V for velocity model and A for acceleration model. During time update stage, using mixed state vector, and mixed error covariance, the best estimate of the states $X^i_{t/t-1}$, and the corresponding $P^i_{t/t-1}$, are obtained. Once the prediction is completed, the innovation covariance S^i_t, and the Kalman gain K^i_t are evaluated. As soon as the measurements are available, the innovations e^i_t, the residuals d^i_t, and the model probability weights M^i are determined. During the filtering phase, the innovations are weighted by their respective gain K^i_t to correct the predicted state estimates, and to get the filtered state estimate $X^i_{t/t}$ and filtered error covariances $P^i_{t/t}$. The resultant filter estimate X_{out} is computed as a weighted sum of individual filter estimates as given above. The algorithm then awaits the next measurement at time $t+1$, and the above process is repeated for each of the subsequent measurements.

Performance of the acceleration model is good during maneuvering, and filtering is not perfect when there is no maneuvering. The reason is, that the acceleration model always anticipates the maneuvering. In the IMM model, the filtering is perfect throughout the track both in presence and absence of maneuvering as it contains the best features of both the velocity model and the acceleration model.

3. Cluster-based solutions for centralized data fusion

Fusion is necessary to integrate the data from the different sensors and extract the relevant information on the targets. The synergistic use of overlapping and complementary data sources provides that is otherwise not available from individual sources. Fusion for target tracking consists of three functions namely, Alignment, Association and Updation of tracks (Liggins *et al* 1997). Data alignment is the first step in fusion process. In this step tracked data is aligned in common time instances. This is achieved by extrapolating the tracks to the same time as measurement times, so that they can be compared with each other. Association is responsible for partitioning the tracks into sets that could have originated from the same targets. Fusing the associated tracks using weight decision method does updation.

Tracking of multiple targets using multiple sensors and fusing them at a central site using centralized architecture involves communication of large amount of measurements to a common site. This results in heavy processing requirement in the central site. Moreover tracks updates have to be obtained in fusion centre before the next measurement arrives. For solving this computational complexity, a cluster-based parallel processing solution is proposed in this paper.

The first step of designing cluster is to choose the architecture (i.e) number of nodes (processor), connectivity etc. The processor power, speed, memory and disk space of each node, as well as the communication bandwidth between the nodes, are taken into consideration in designing the architecture. In the present design the cluster consists of one master and four slaves, where the master node sends the required files to the slave nodes and assigns jobs for each node. Cluster nodes are tightly coupled, (i.e.) part of the master's primary memory is

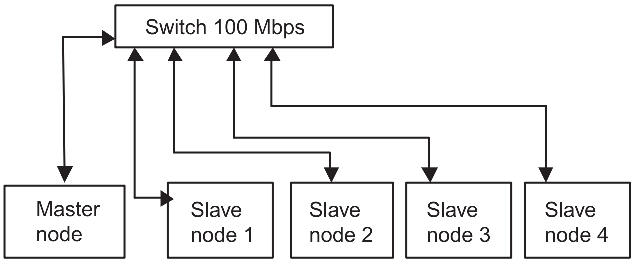

Figure 3. Cluster system.

shared to all the nodes connected in the network. The slave nodes accept the data and necessary files from the master and perform the necessary action (the task assigned for it) over the data. The slaves transmit results back to the master processor. Figure 3 shows the cluster system.

Figure 4 explains the work spawning between the master and slave nodes. The master node is made of a Pentium III processor of speed 1·5 GHz, hard disk of 40 GB and main memory of 512 MB. Slave nodes are of similar type of machines as the master, but the main memory is only 256 MB. All the machines are equipped with fast Ethernet card of 100 Mbps. These nodes are connected in a network by 8 ports 100 Mbps duplex switch. All the machines are installed with a software tool called PVM (parallel virtual machine), which enables collection of computers to be cooperatively used for concurrent or parallel computation.

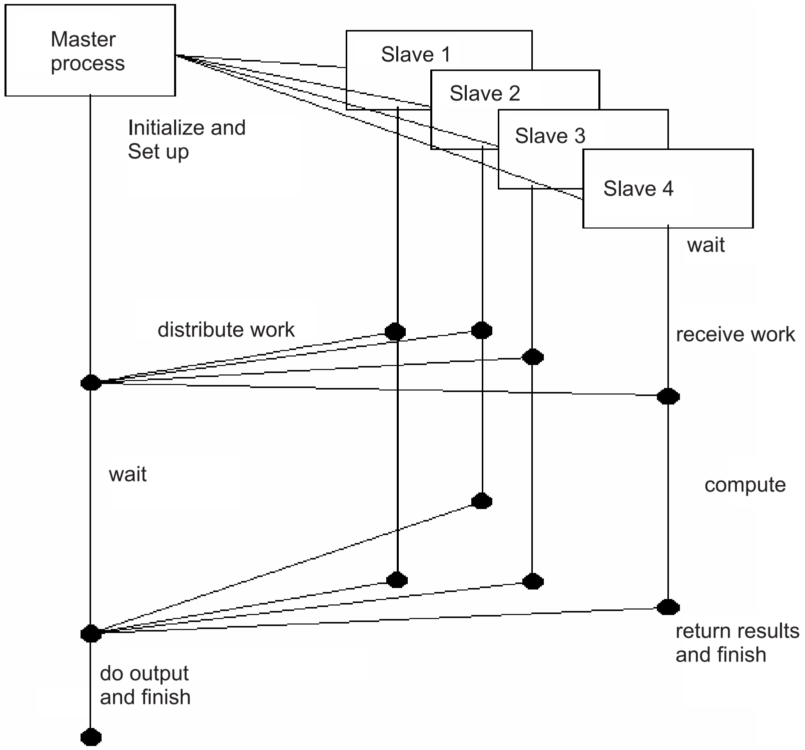

Figure 4. Work spawning.

In the current work, a scenario where four radars are connected in a network is considered for implementing data fusion. The objective is to use clusters in the fusion centre of the centralized architecture to get the fused tracks of the targets using the measurements from these four radars in lesser time than using a single processor in the fusion centre.

4. Parallel algorithm for tracking and data fusion

```
Master
BEGIN
   Get RADAR inputs (plots) namely range, azimuth, elevation
REPEAT for time $t1$ and $t2$ (sampling times)
   BEGIN
        Allocate plots to node 3 & 4 at time $t1$
        Allocate plots to node 1 & 2 at time $t2$.
   END
   BEGIN
        Get filtered data form node 3 & 4 at time $t1$.
        Get filtered data form node 1 & 2 at time $t2$.
        Fusion of tracks
   END
END
END
Slave
REPEAT for time $t1$ and $t2$
BEGIN
   Node 3 & 4 get plots from master
   Node 1 & 2 get plots from master
   Node 3 & 4 give filtered data after tracking using IMM
   Node 1 & 2 give filtered data after tracking using IMM
   Node 3 & 4 send filtered data to master.
   Node 1 & 2 send filtered data to master.
END
END
```

5. Analytical evaluation of performance of centralized and distributed data fusion

This section presents the simplified analytical expression for estimating the number of tracks of those that can be fused using the sensor measurements both in the centralised architecture and distributed architecture.

5.1 *Centralised architecture*

5.1a *Analytical evaluation of the transmission time:* In the centralized architecture the terms associated with the information transmission are defined as follows: X_m, Y_m, Z_m – measurement data; t_m – scan time.

Therefore for each radar site, the average number of terms per scan time is given by

$$nw_i = 4(nt_i),$$

where nw_i is the average number of terms to be transmitted per scan time, and nt_i is number of targets relevant to the ith site per antenna scan. If the transmission cable can support a bit rate of 64 kbps then the time taken to transmit all the information to the fusion centre is:

$$T_{\mathrm{tr}} = n_b/64k + T_{\mathrm{pd}},$$
$$T_{\mathrm{pd}} = l \times t,$$

where n_b is the number of bits required to represent nw_i (where ith radar is considered to take the longest transmission time among all the other radars) + number of bits required for the header to transmit in TCP/IP protocol, T_{pd} = propagation time, l = length of the cable, t = travel time per metre, which is 1·5 ns for copper wire, and T_{pd} is negligible.

Hence

$$T_{\mathrm{tr}} = n_b/64\,\mathrm{K}.$$

5.1b *Analytical evaluation of the processing time:* The overall processing time in centralized architecture is give by

$$T_{\mathrm{pr}} = \Delta t \sum_{k=1}^{N}(n_k),$$

where T_{pr} = overall processing time, Δt = time necessary to process each plot, N = number of radars, n_k = number of targets relevant to the kth site per antenna scan.

So the total time taken (from the instant radar detects the target to the time all the data have been processed) at the fusion centre is given by

$$T_{\mathrm{tot}} = T_{\mathrm{tr}} + T_{\mathrm{pr}}.$$

Fusion in centralized architecture is effective, if

$$T_{\mathrm{tot}} < T_{\mathrm{sp}}$$

where T_{sp} = scan period of radar which has lowest scan period among all other radars.

If T_{tot} exceeds T_{sp} then fusion cannot be done in centralized architecture. In that case we have to go for distributed architecture.

5.1c *Analytical evaluation on centralized architecture with cluster:* If cluster is used in centralized architecture the processing time is given by

$$T_{\mathrm{pr,c}} = T_{\mathrm{pr/s}},$$

where $T_{\mathrm{pr,c}}$ = overall processing time using cluster, T_{pr} = overall processing time using single processor, s = speed up achieved using the cluster.

So the total time taken using the cluster is given by

$$T_{\mathrm{tot,c}} = T_{\mathrm{tr}} + T_{\mathrm{pr,c}}.$$

If $T_{\mathrm{tot,c}} > T_{\mathrm{sp}}$ then the usage of cluster cannot achieve fusion in the centralized architecture.

The total time taken for transmitting the measurements from local sites and to finish the processing of plots in the single processor environment is analytically calculated and the

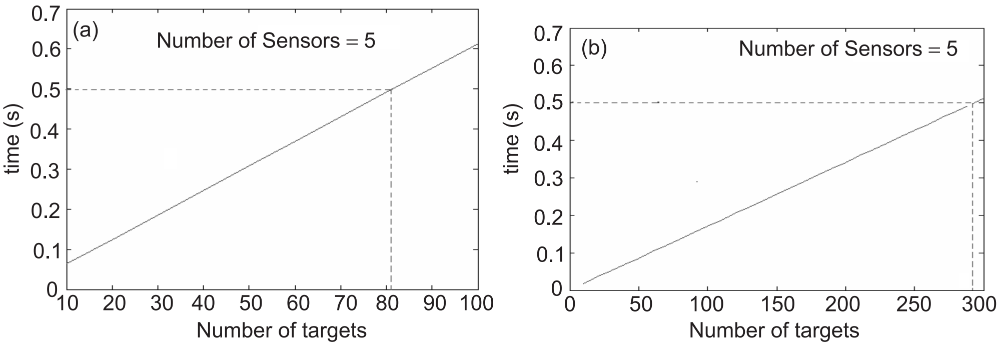

Figure 5. Number of targets vs time taken for getting the fused output using a single processor **(a)**, and a cluster **(b)**.

values are presented as graphs. In figure 5a, a scenario with five sensors is considered. It is observed that the time for getting the fused output linearly increases for increasing number of targets. It can be seen that if the scan time is 0·5 s then, it is possible to handle only up to 80 targets (neglecting communication overhead time, data association, clutter etc) using a single processor. From figure 5b it is observed that if we use cluster in the centralized architecture upto 290 targets can be processed for same no of sensors and scan time. Figure 6a presents the time taken for getting the fused output for varying number of sensors with each sensor handling 10 targets. It is shown that, if scan period is 0·5 s, it is possible to accommodate up to 45 sensors using a uni-processor. In figure 6b it is shown that if we use a cluster, a maximum 160 sensors can be handled.

Using the analytical results it is found that in centralized architecture having five sensors networked, with 0·5 s scan time, number of targets that can be handled is increased from 80 upto 290 using cluster in data fusion centre. Number of sensors each with maximum 10 targets can be increased from 45 to 160 using cluster. The analytical results are summarized in table 1. However, these results ignore the effect of communication overhead, assume no clutter and no data association problem.

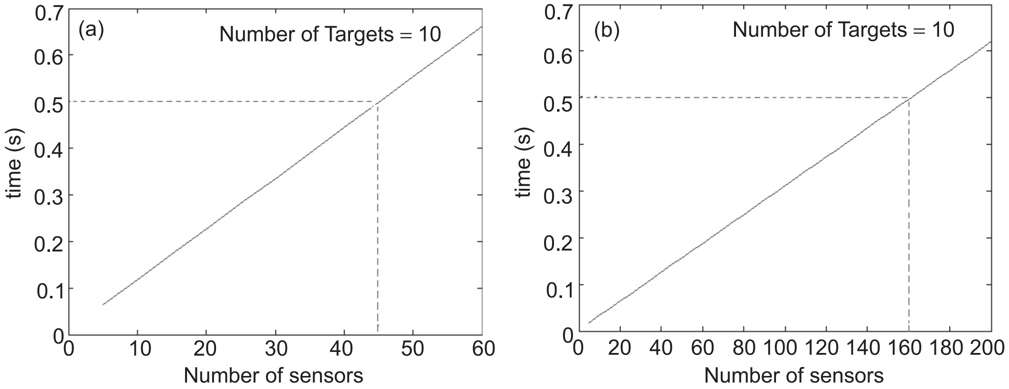

Figure 6. Number of sensors vs time taken for getting the fused output using a single processor **(a)** and a cluster **(b)**.

Table 1. Comparison of distributed and centralised architecture–analytical results.

Scenario considered	Uniprocessor	Cluster
Scan rate = 0·5 s No. of sensors = 5	No. of targets that can be tracked = 80	No. of targets that can be tracked = 290
Scan rate = 0·5 s No. of targets/sensors = 10	No. of sensors handled = 45	No. of sensors handled = 160

5.2 *Distributed architecture*

5.2a *Evaluation of the processing time in the local site:* Let TL_{pr} be the total processing time in local site i. Time taken to get the filtered outputs from site i is given by

$$TL_{\text{pr}} = \Delta t_f n_i,$$

where Δt_f = time taken to get the filtered output for each target, n_i = total number of targets visible to radar at site i (where radar at site i is considered to track large number of targets among all other radars).

5.2b *Analytical evaluation of the transmission time:* In the decentralized distributed architecture the four terms associated with the information that are to be transmitted per target are:

$$X_f, Y_f, Z_f \text{ – filtered data, } t_m \text{ – scan time.}$$

The average number of terms per scan time at each radar site i is given by

$$nw_i = 4(n_i),$$

where n_i is the number of targets relevant to the ith site per antenna scan. If the transmission cable can support a bit rate of 64 kbps then the time taken to transmit all the information to the fusion centre is given by:

$$T_{\text{tr}} = n_b/64\,\text{k} + T_{\text{pd}}$$
$$T_{\text{pd}} = l \times t,$$

where n_b is the number of bits required to represent nw_i (where ith radar is considered to take the longest transmission time among all the other radars) + number of bits required for the header to transmit in TCP/IP protocol, where T_{pd} = propagation time, l = length of the cable, t = travel time per meter and it is 1·5 ns for copper wire. However T_{pd} is negligible.

Hence

$$T_{\text{tr}} = n_b/64\,\text{k}.$$

So the total time taken to transmit all the filtered updates from site i to fusion centre, from the instant the measurements received by radar at site i is given by,

$$TL_{\text{pr}} + T_{\text{tr}}.$$

5.2c *Analytical evaluation of the processing time in fusion centre:* The processing time in fusion centre is give by

$$TF_{\mathrm{pr}} = \Delta t \times N,$$

where TF_{pr} = time taken to get fused output of all the targets; Δt = time necessary to process each track, N = total number of targets.

So the total time taken to get the accurate track of each target after fusion in the fusion centre is given by

$$T_{\mathrm{tot}} = TL_{\mathrm{pr}} + T_{\mathrm{tr}} + TF_{\mathrm{pr}}.$$

5.2d *Comparison of centralised and distributed data fusion architectures:* In the centralized architecture processing time in the fusion centre is given by

$$T_{\mathrm{pr}} = \sum_{k=1}^{s} TL_{\mathrm{pr,k}} + TF_{\mathrm{pr}},$$

where s is the total number of sites.

So the total time taken to get the accurate track of each target after fusion in the fusion centre in centralized architecture is given by

$$T_{\mathrm{pr}} = \sum_{k=1}^{s} TL_{\mathrm{pr,k}} + T_{\mathrm{tr}} + TF_{\mathrm{pr}}.$$

Hence, if the number of sites increases, the total time taken to get accurate fused track in centralized architecture is very high comparing to decentralized architecture.

6. Results and discussion

In the present work, the radars are assumed to have different scan rates. To perform fusion, the measurements from different radars are extrapolated to a common time instant. Then association is performed using nearest neighbour approach (Blackman 1986). Weightages are given to the tracked data depending on the accuracy of the radar to perform data fusion. Thus by using track-to-track fusion at the fusion centre, the global scenario and an accurate track for each target are obtained. Consider the scenario of a target with racecourse pattern track and two radars. Only a portion of the target track is visible to both the radars. The two tracks corresponding to the same target are fused at the fusion centre to get the complete track of the target. In the overlapping region, tracks are fused using weighted decision method where weightages are given depending upon the scan rates of the radars.

Figure 7a shows the portion of the track visible to radar 1. Figure 7b shows the portion of the track visible to radar 2. Figure 7c shows the complete track of the target got by fusion algorithm.

Next, for illustrating the use of cluster, four radars connected in a network are considered. The master gets measurement data for tracks 1 and 2 from radars 1 and 3, measurement data for tracks 3 and 4 from radars 2 and 4. For the first two radars, the standard deviation of errors in range, azimuth and elevation are assumed to be 50 m, 0·005759 rad and 0·0171 rad respectively, and its scan time is taken as 1 s. For the third and fourth radars, the standard

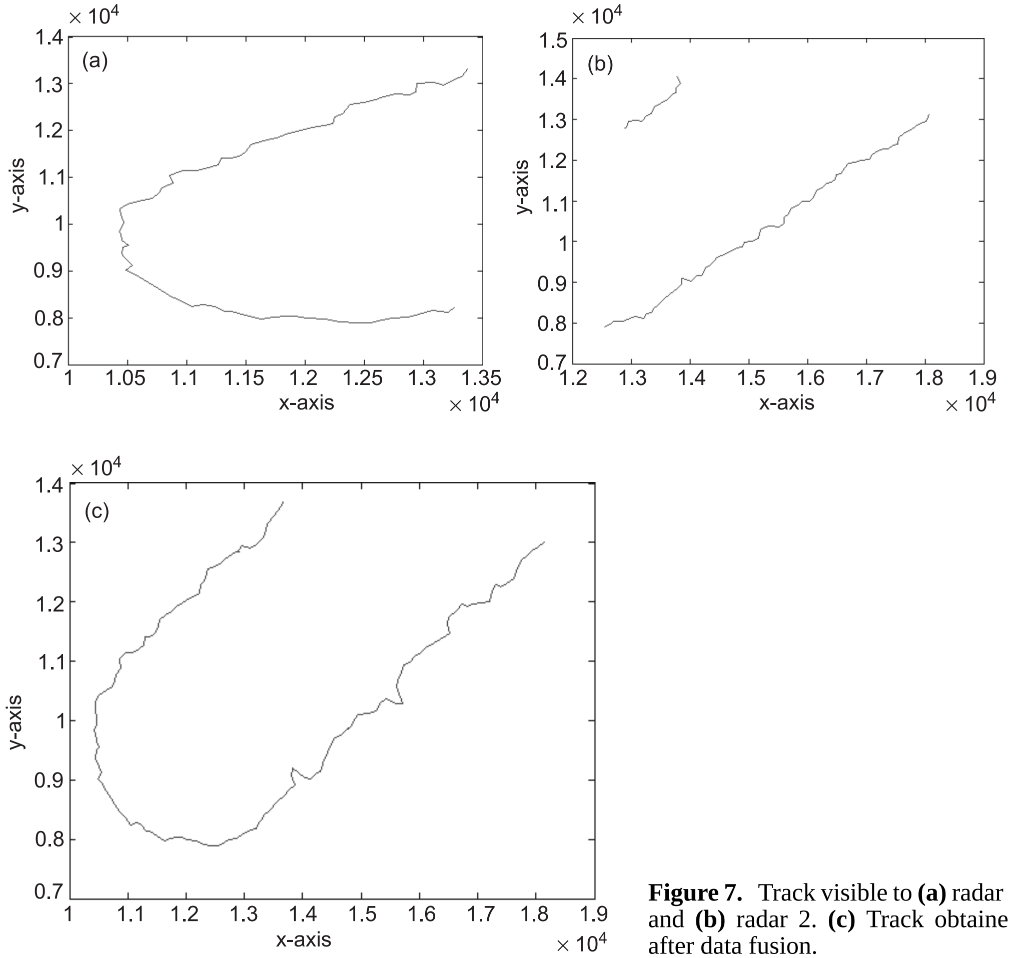

Figure 7. Track visible to **(a)** radar 1 and **(b)** radar 2. **(c)** Track obtained after data fusion.

deviation of errors in range, azimuth and elevation are assumed to be 60 m, 0·005759 rad and 0·0171 rad respectively and its scan time is taken as 0·5 s.

Table 2 shows task allocation to the master and slave nodes. The master allocates the plots from the four radars to the slaves. The slaves do the tracking and give filtered data (tracks) to the master for data fusion. Computational time for tracking multiple targets, and data fusion in

Table 2. Details of work done by the nodes in the cluster.

Slave 1	Tracking plots from RADAR 3 for every 1 s for targets 1 and 2
Slave 2	Tracking plots from RADAR 4 for every 1 s for targets 3 and 4
Slave 3	Tracking plots from RADAR 1 for every 0·5 s for targets 1 and 2
Slave 4	Tracking plots from RADAR 2 for every 0·5 s for targets 3 and 4
Master	Allocating the plots to 4 nodes and fusion of tracks

Table 3. Performance using cluster.

Computation time using cluster	0·27811 s
Computation time using single processor	0·97916 s
Speed up using cluster	3·521
Efficiency of the cluster	88%

the cluster environment and in a single processor environment are experimentally determined under simulation. The speedup using the cluster is defined as the ratio of the time taken by the uni-processor system to the time taken by the cluster. The efficiency of the cluster is defined as the ratio of speedup to the number of nodes in the cluster. The speedup and efficiency are found to be 3·5 and 88% respectively and the results are presented in table 3.

7. Conclusion

Centralized data fusion using a cluster-based parallel processing system is presented in this paper. The proposed method is useful in solving the high computational requirement in the fusion centre of the centralized architecture. Moreover it is useful if accurate tracking results are received. Using multiple sensors connected to the network, more coverage of the target's track is possible using data fusion. The performance of the cluster-based centralized data fusion is presented in terms of speedup.

References

Bar Shalom Y, Xiao-Rong L 1993 *Estimation and tracking* (Boston, MA: Artech House)

Blackman S 1986 *Multiple target tracking with radar applications* (Dedham, MA: Artech House)

Blom H A P, Bar-Shalom Y 1988 The interacting multiple model algorithm for systems with Markovian switching coefficient. *IEEE Trans. Autom. Control* 33: 780–783

Chang K C 1994 Tracking and fusion using multiple sensors. *Control Eng. Practice* 2: 889–897

Farina A, Studer F A 1986 *Radar data processing. Vol. II – Advanced topics and applications* (New York: Wiley)

Hall D L, Llinas J 1997 An introduction to multisensor data fusion. *Proc. IEEE* 85: 6–23

Indira Gandhi S, Kalavidya K, Vaidehi V 2003 Centralized data fusion using cluster for tracking maneuvering targets using interacting multiple model. Presented at the NCAC 2003 , Third National Conference on Advanced Computing, Coimbatore

Liggins M E II, Kadar I, Vannicola V 1997 Distributed fusion architectures and algorithms for target tracking. *Proc. IEEE* 85: 95–107

Varadarajan S, Suresh S D 1999 Multitarget tracker for heterogenous radar network. *Proc. Int. Radar Symposium*, Bangalore, pp 480–487