999

DEVELOPMENT OF ALGORITHMS FOR DATA FUSION IN NETTED RADAR SYSTEM

DISSERTATION SUBMITTED BY

PURAN AUDESH BHAGAT

IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE AND TECHNOLOGY

SCHOOL OF COMPUTER AND SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI

JANUARY 1993

CERTIFICATE

This is to certify that the dissertation entitled **"Development of Algorithms for Data Fusion in Netted Radar System"**, being submitted by me to Jawaharlal Nehru University, New Delhi in the partial fulfilment of the requirements for the award of the degree of **Master of Technology**, is a record of original work done by me under the supervision of **Dr. P. C. Saxena**, Associate Professor, School of Computer and System Sciences, Jawaharlal Nehru University during the year 1992, Monsoon Semester.

The results reported in this dissertation have not been submitted in part or full to any other University or Institute for the award of any degree or diploma, etc.

PBhagat
Puran Audesh Bhagat

Prof. R. G. Gupta
Dean, 4/1/93
School of Computer
and System Sciences,
J. N. U., New Delhi.

Dr. P. C. Saxena,
Associate Professor,
School of Computer
and System Sciences,
J. N. U., New Delhi.

ACKNOWLEDGEMENTS

I express my sincere thanks to **Dr. P. C. Saxena,** Associate Professor, School of Computer and System Sciences, Jawaharlal Nehru University, New Delhi for suggesting such a brilliant topic. I am indebted to him for his personal involvement with my work and his immense and eloquent guidance which has been indispensable in bringing about a successful completion of the dissertation.

I am also grateful to Dr. Saxena for providing me with his invaluable notes and papers related to the topic and also guiding me in my lookout for proper references.

I extend my sincere thanks to **Prof. R. G. Gupta,** Dean, School of Computer and System Sciences, Jawaharlal Nehru University for providing me with the environment and all the facilities required for the completion of my dissertation.

I also take this opportunity to thank all faculty and staff members and my friends who helped me in every way possible.

PBhagat

Puran Audesh Bhagat

INDEX

PREFACE

The problem of netted radar system arises when the search and tracking are performed by using measurments obtained from two or more spatially separated radar sets having overlapping coverages. Netting allows search for and track of targets over an area wider than the coverage of each individual sensor. It generally implies the conveying of data provided by different radar systems, to a main site. Here, a single data processor correctly combines the target reports and establishes target tracks which are of higher quality then those formed from a single radar. The inherent redundancy of multiple radars results in a high overall reliability and allows achievements of some reconfiguration capability of one or more of the radars. These advantages are obtained at the expense of spatial and time alignment of radars and more extensive computer resources and communication facilities. In the recent years, the increasing sophistication of distributed target- detection and tracking system has generated a great deal of interest in devolopment of new computational structures and strategies. The design of such spatially distributed systems involves the integration of solutions obtained by solving subproblems in data association, hypothesis testing, data fusion etc. Multisensor integration and fusion of information requires techniques to abstractly represent and integrate sensor information.

In this dissertation, we have tried to elaborate the concept of coordinate transformation, registration errors, association of data having attributes and data fusion in the case of MRT system. Then a Blackboard architecture is proposed for data fusion.

Chapter 1 of this report describes the problem with a single sensor (Radar) tracking and then gives an introduction to multisensor (Multiradar) tracking.

Chapter 2 describes the algorithm for coordinate transformation.

Chapter 3 defines the registration problem in terms of source of registration error and their subsequent implication on mulisensor tracking; and offers a solution to the registration problem.

Chapter 4 discusses the application of Bayesian and Dempster-Shafer method to the fusion of multisensor attributes and target information.

Chapter 5 discusses the two level data fusion model; three reasoning classes for data fusion and memory elements required to store the information.

Analysis of real time or time critical processing, parallelization, object oriented approaches, storage and search problems, knowledge representation issues and spatial, hierarchical and temporal reasoning has lead us to the postulation of the "ideal" Black Board system architecture, which is described in chapter six.

CHAPTER 1

INTRODUCTION

Single sensor systems have been studied for a long time. The motivation for this was their requirements in applications requiring detection and tracking of targets using a single sensor such as a radar or sonar. Despite two or more decades of intense research in the development of sensors and of diverse sorts of sensory information no single sensor can guarantee to deliver accurate information all of the time. This is because of two main reasons:

1. Associated with any sensor is a set of limits that define its useful operating range.
2. Any sensor signal is inevitably corrupted by noise.

Because of these reasons the data collected by the sensor can be incorrect. To ensure correct inferences by the program that interpret the sensor data, the sensor must be made fault tolerant. Single sensor systems can be made fault tolerant in two ways :

1. Based on the sensor's specification the sensor output can be ignored if it is unrealistic.
2. The sensor can be replicated physically or logicaly so that when one sensor fails another sensor can take over.

The first method can be used in situations in which time is not a critical factor. The second method is a trivial instance of a distributed sensor network, and has been used to monitor temperature in nuclear reactor vessels. .pa

Even though we can make a single sensor system more reliable using the above two techniques, these systems can be used in limited applications. This is because

1. Every sensor has a limited space. Hence any single sensor cannot sense the complete phenomena that must be monitored.
2. Some applications require that the observations be taken from different point of view.

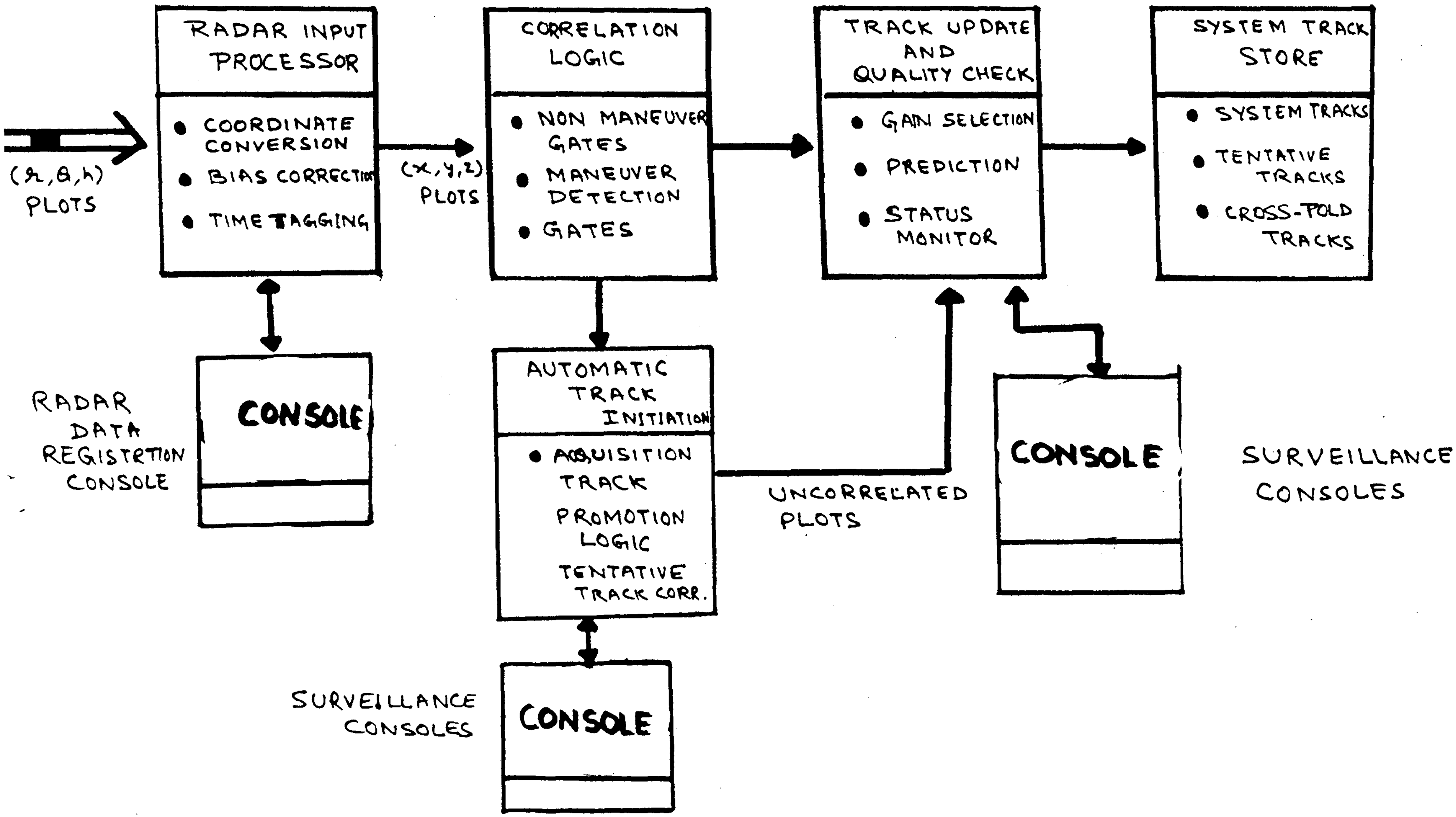
Because of the limitations of single sensor systems and increasing fault tolerance requirements of today's applications, single sensor systems are being replaced by distributed sensor networks.

1.1 Multi-Sensor Tracking

A Multisensor tracking consists of units with sensing, computational and communicational ability that are physically distributed. Here signals from several sensors are combined to derive a more acurate and reliable value of the phenomena that the sensors are monitoring. The MRT problem is a subclass problem of MST.

The interaction of many radars, display systems and communication links together with command and control systems requires the aid of computers because of the abundance of information to be handled. Computers are sited esentially in

FIG. 1. FUNCTIONAL REQ. FOR MULTPLE RADAR TRACKING



the main centre, and sometimes at the radar sites, or in the display system and in the command and control area. As a consequence, a radar network implies a computer network which ensures performance of data processing, organisation of information display (as a necessary prerequisite for taking decisions) and, finally, communication between the different system components.

1.2 Advantages of MRT :

MRT has a number of advantages over a monoradar system, namely:

- performance of target advantages and tracking over an area wider than the coverage of each individual radar.
- reduction of coverage gaps especially at low altitude.
- early initiation of new tracks.
- track continuity during the hand-over of a target between two adjacent radars.
- more precise estimation of the track parameters (due to a higher data rate) than with a single radar.
- lower probability of false correlation in the areas affected by clutter owing to smaller dimensions of the correlation gate.
- higher detection opportunities in the overlapping areas.
- reduced vulnerability to clutter and/or jamming effects because of the different siting and sensor characteristics.
- reduced occurrence of targets having blind speed or

range as a result of different waveforms, processing parameters and siting of the netted radars.

- ability to evaluate target altitude through triangulation.
- capability of estimating the total velocity vector from independent radial speed measurements.
- capability of system reconfiguration in the case of failure of one or more radars.
- reduced human intervention in supporting data processing (e.g. track initiation and track validation owing to the improved performance and reliability of the system).

1.3 LEVEL OF DATA ASSOCIATION :

In a multiple-sensor tracking system the first major conceptual issue is to define the level at which data will be combined into tracks. The choice are sensor or central-level tracking or some combination of both.

1.3.1 SENSOR LEVEL TRACKING :

The first alternative, illustrated in Fig 2 , is to have each sensor maintain its own track file. The track in these sensor track files would be established primarily upon measurements received from the individual sensor, but some communication among the sensors and between the sensors and the central track file may be used to update sensor-level track file. However, the sensor-level tracks must eventually be combined into a central track file. Thus, under this (sensor-level) approach each sensor would have separate track file and central track file would be formed as a composite.

Point cited in favor of sensor-level tracking are reduced data-bus loading, reduced computational loading (in any single processor), and higher survivability due to distributed tracking capability. Certain computational advantages may result from parallel processing that is possible using the sensor-level track approach. Also, if one sensor becomes degraded, its observation will not affect the sensor-level tracks of the other sensors. Finally, the use of sensor-level tracking allows for filter design that is specifically tailored to the individual sensors.

If sensor-level tracks are maintained, they must be combined at some point if significant benefit is to be derived from the multisensor fusion approach. The result is central-level track that are updated with sensor-level track data, instead of with sensor report data. Several problem arise. First, if a central-level track is updated with a sensor-level track, the usual assumption (valid for the case of raw measurements with uncorrelated measurements error) of error independence from one update period to another is not valid. This can be taken into account in the processing, but it forces additional complexity. Second, less accurate tracking and correlation are to be expected if indepedent sensor-level tracks are maintained. For example, there will be a higher probability of false correlation in areas affected by clutter because the gate sizes will be larger due

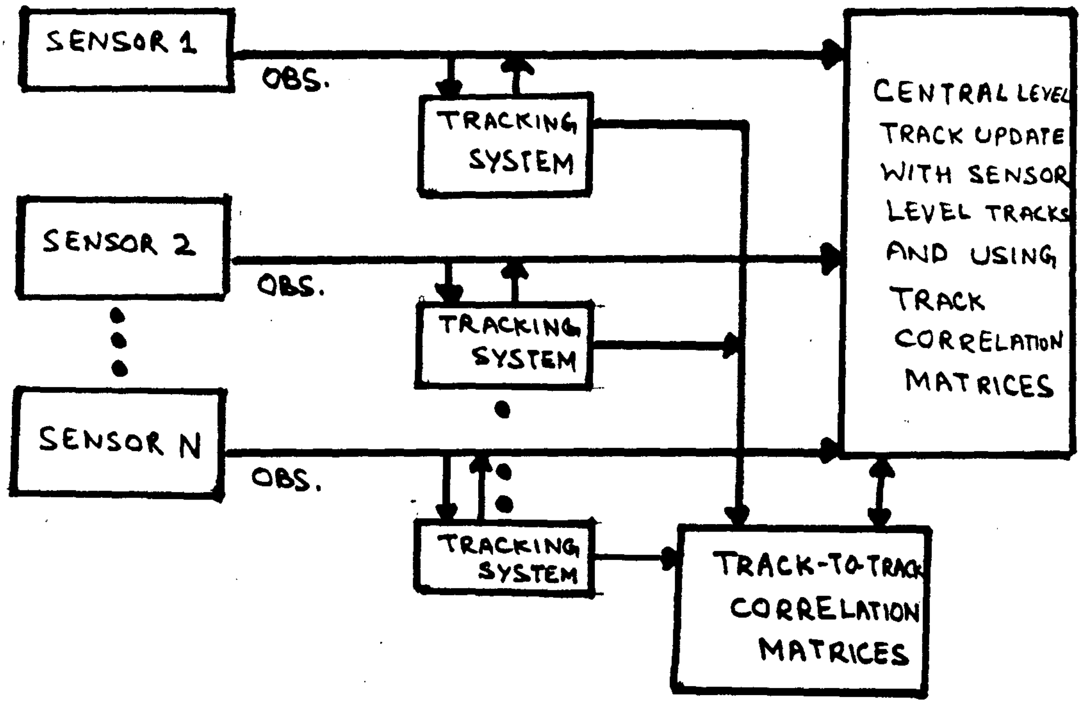


Fig. 2 SENSOR LEVEL TRACKING

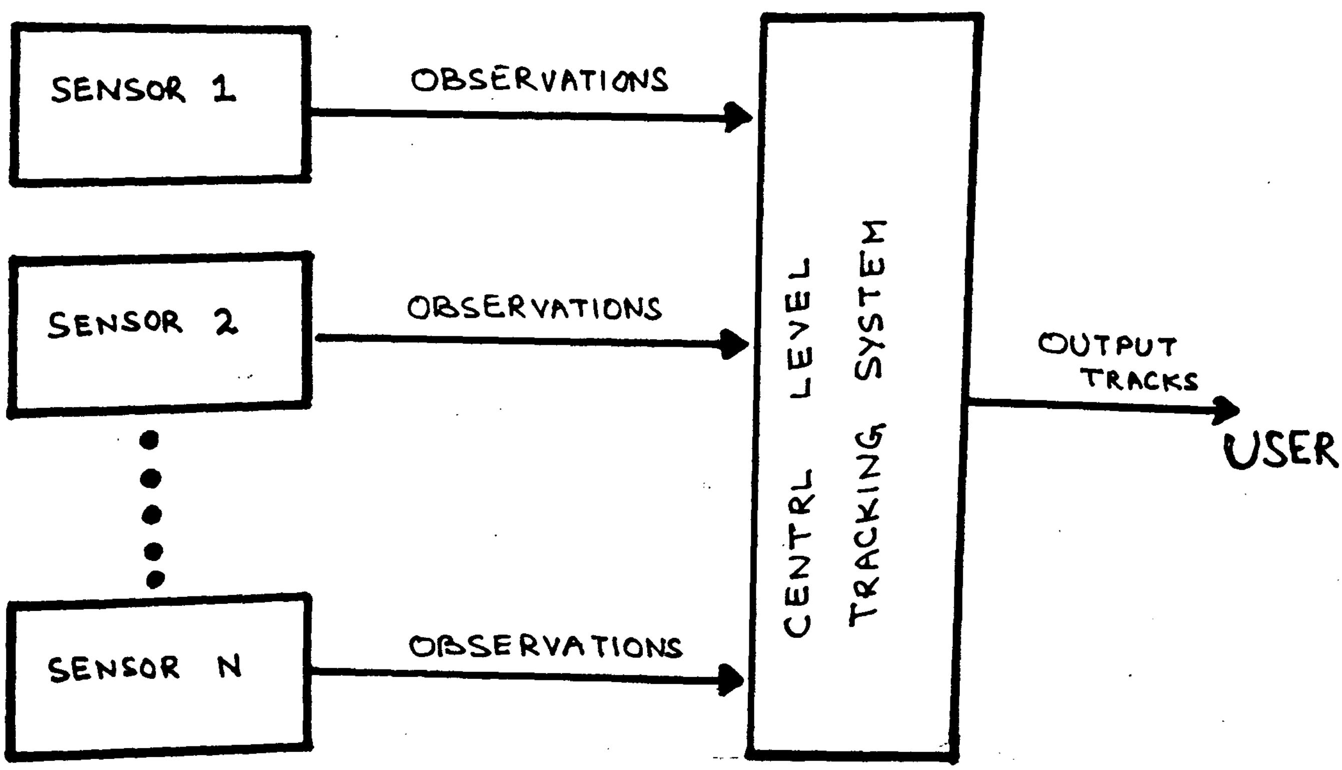


Fig. 3 CENTRAL LEVEL TRACKING

to less frequent track updates. Finally, if the multiple hypothesis tracking approach is taken at the central level, when sensor-level tracks are combined, it would also be desirable that this approach be taken at the sensor level as well. However, the maintenance of a single hypotheses tree at a central level is simpler than the maintenance of amany hypotheses at each sensor as well as additional hypotheses at the central level to combine the sensor-level tracks.

1.3.2 CENTRAL LEVEL TRACKING :

The alternative to maintaining sensor-level tracks is for all report data to be sent directly to a central processor where a master file is maintained. This approach, illustrated in Fig. 3 , also has a number of advantages. First, more accurate tracking should be possible if all data are processed at the same place. A target track that consists of observations from more than one sensor should be more accurate than the tracks which could be established on the partial data received by the individual sensors.

Thus, the central processing track approach should lead to fewer miscorrelations. Second, by processing sensor reports directly, the difficulties associated with combining sensor-level tracks are avoided. These dificulties include correlating sensor-level tracks are determining an efficient scheme for combining these tracks. Track confirmation and continuity should also be improved with central-level tracking.

The various sensors will under different circumstances, have varying ability to confirm and sustain a track. Thus, by using detections from all sensors for each track, the probability of confirming and sustaining a track can be improved over that for a single sensor. For example, in a system with radar and an infrared (IR) sensor. IR detections can maintain a track that might otherwise be lost during a fade in the radar return due to radar cross section scintillation. Also, various sensors can be used synergistically, for example, as the radar provides range and range rate while the IR provides more accurate angle measurements. Finally, the approach whereby all data are sent directly to central processor should, in principle, lead to faster, more 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.

There can, however, be a major drawback if pure central-level tracking is used. This problem occurs when the data from one sensor can become degraded and thus lead to poor central-level tracking. In this case, the possible combination of good data from undegraded sensor with bad data, in effect, will negate the value of the good data. However, if sensor-level tracks are maintained, the good sensor-level tracks will not be corrupted by the bad data. Then, when the sensor with the poor data is finally recognized , the central-level tracks can be formed using only sensor-level tracks for undegrated tracks.

1.4 FUNCTIONS PERFOMED IN CENTRAL LEVEL

At the central tracking processor, the plots from the multiple radars are used to update existing system tracks or initiate the new tracks as appropriate. Specifically, the central tracking processor must perform the following five functions :

1. Transformation of the plots from local radar coordinates to system coordinates, which usually are Cartesian coordinates in 3-d space.
2. Correlation or association of radar plots with the appropriate system tracks. (Note that, because there are multiple radars in the system, more than one plot may correlate with the same track over a nonzero time interval).
3. Initiation of new tracks with the uncorrelated plots and rejection of clutter plots. (Note, again, that this is not a straightforward task as there are no simple criteria with which valid aircraft detection can be distinguished from clutter return).
4. Tracks filtering (or updating with correlated plots) and track prediction.
5. Track monitoring and system track management (including association with tracks from external sources).

CHAPTER 2

CO-ORDINATE TRANSFORMATION

The problem of co-ordinate transformation is a direct consequence of the control of a wide portion of airspace at a single facility on the basis of data acquired from a multiple-radar tracking network. The plots reported by each radar have to be reffered to a common co-ordinate reference system. The measurements of target slant range, azimuth and altitude (or elevation angle) available at each radar site are transformed into a point of a common cartesian plane. On this plane, the air picture detected by the netted radar system is represented. The spatial congruence between plots, tracks and topological maps should be maintained in all the different co-ordinate systems.

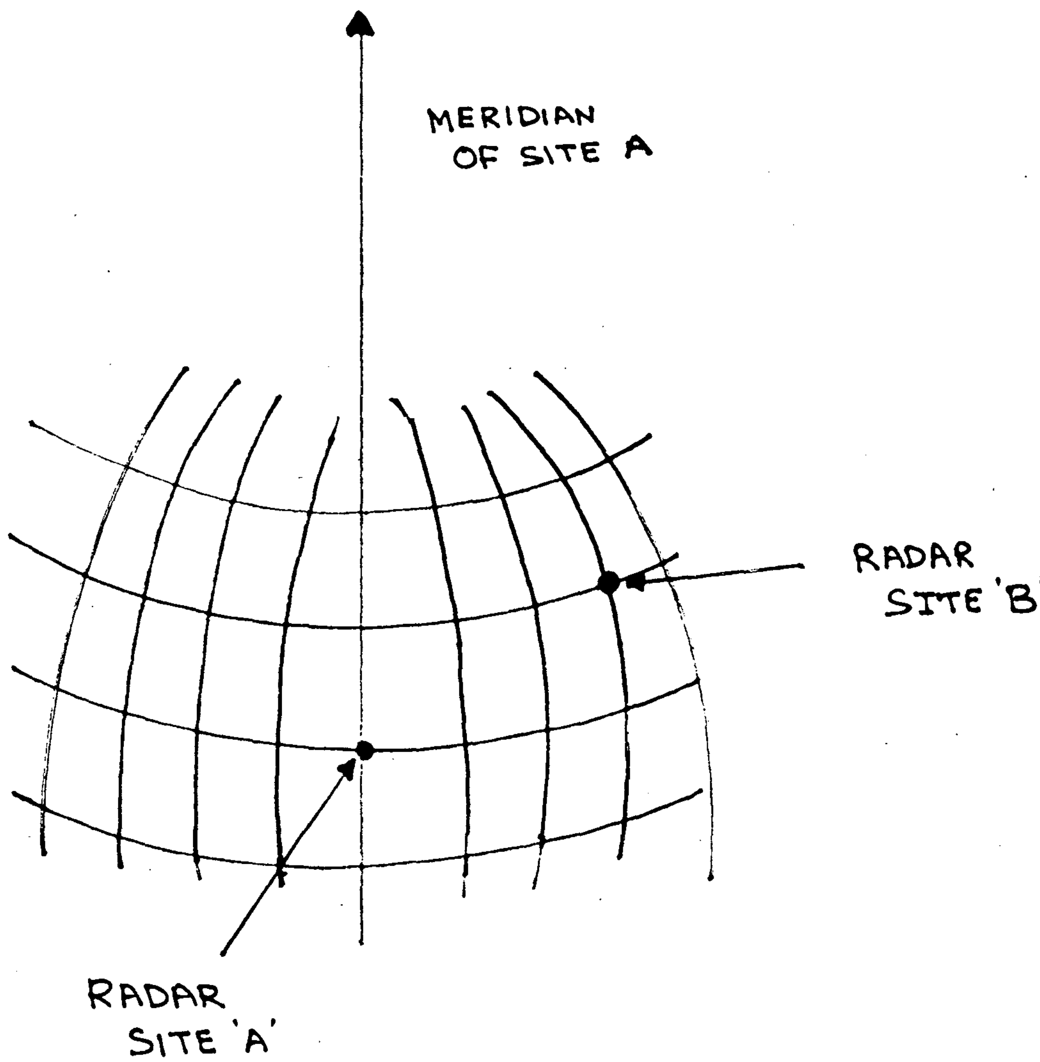
The coordinate conversion is achieved by generating a set of transformation equations that can be map any point in the frame of reference of the radar onto the frame of reference of the CRC. With the locations of the radar and CRC known in terms of their latitude and longitude, the respective 3d X-Y-Z coordinates with reference to the earth's centre (the absolute reference) are found using equation 1,2 and 3 .

$$ZR = R \cdot \sin(\text{latitude}) , \qquad .. 1$$

$$YR = R \cdot \cos(\text{latitude}) \cdot \sin(\text{longitude}) \text{ and} \qquad .. 2$$

$$XR = R \cdot \cos(\text{latitude}) \cdot \cos(\text{lonitude}) \qquad .. 3$$

XR, YR and ZR so obtained also form the direction ratios of the normal to the plane tangential to the earth's surface at the radar location under consideration .



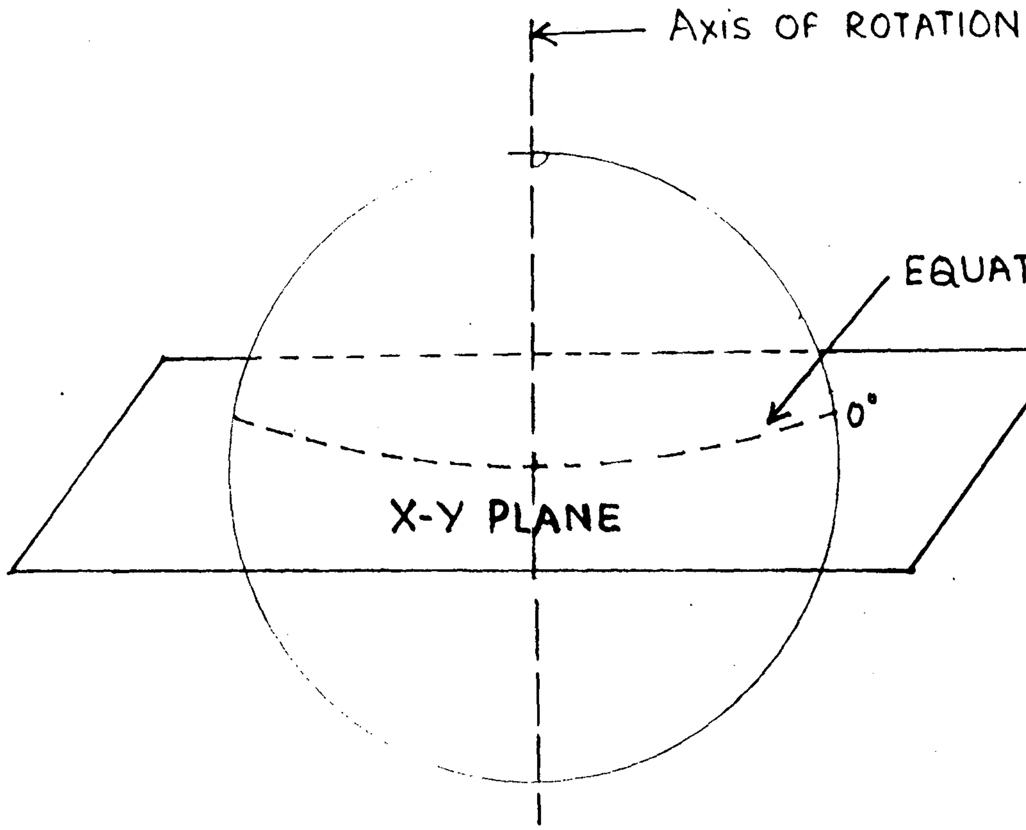
(X_A, Y_A) CO-ORDINATE REFERANCE SYSTEM OF SITE 'A'

(X_B, Y_B) CO-ORDINATE REFERENCE SYSTEM OF SITE 'B'

Figure - 4.

EARTH WITH TWO RADARS AT DIFFERENT LOCATIONS

Fig. 5 Absolute frame of reference



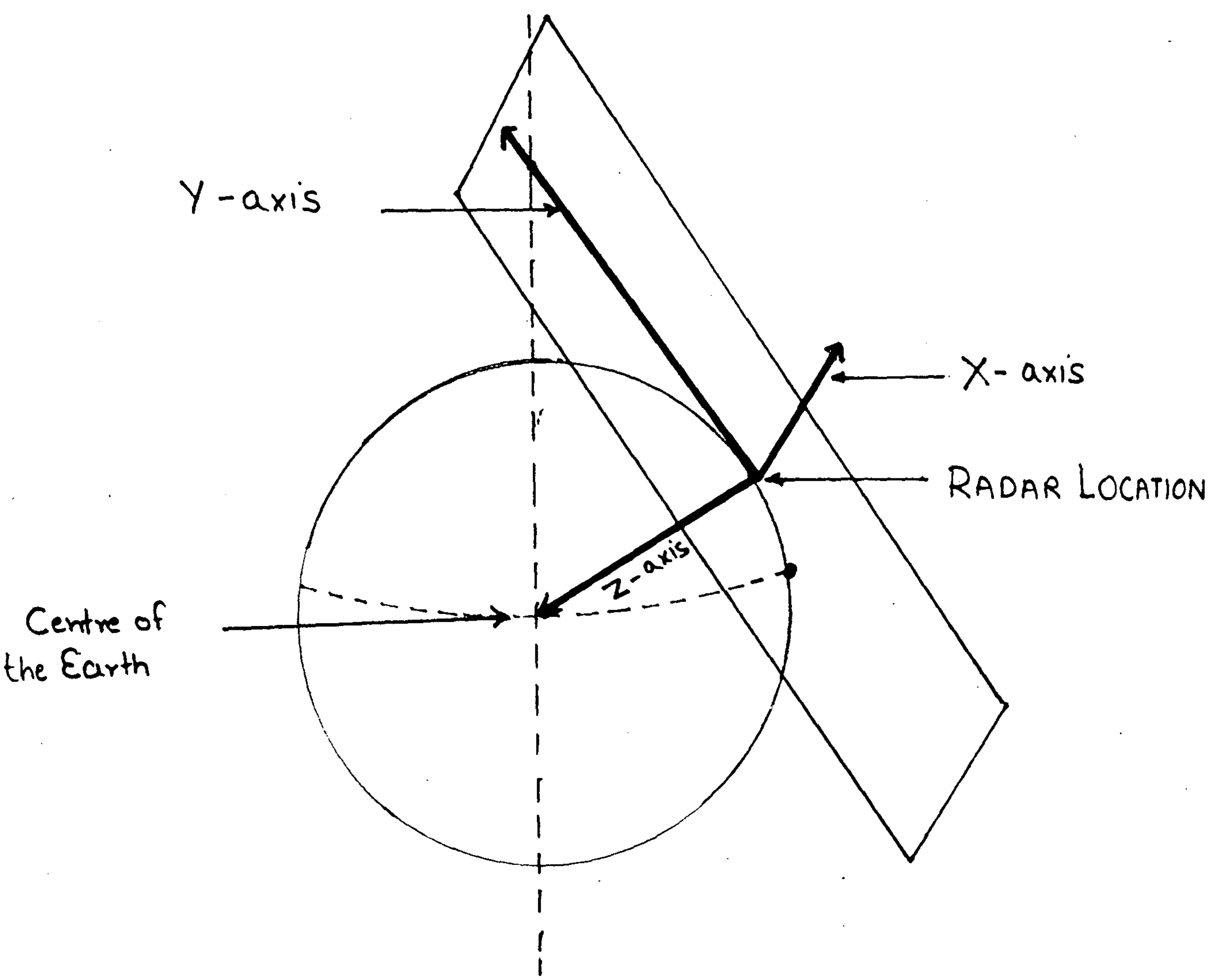


Fig. 6 Radar frame of reference

In the next step, transformation equations for each radar are found. These transformation equations can be used to map any point in the absolute reference frame onto that of a radar. To achieve this, the radar's frame of reference needs to be defined in that of the absolute one. The origin is kept as the radar location itself. The new 'X' and 'Y' axis, shall be in the direction EAST-WEST and NORTH-SOUTH of the radar, respectively. The 'Z' axis shall point towards the zenith that is the extension of the line joining the earth's centre and the radar location point (new origin), with its portion above the earth surface being the positive region and that inside earth the negative one .

Next we express these planes in proper equations that represent them in the absolute coordinate reference. The general equation of a plane in the 3d coordinate system is

$$A . X + B . Y + C . Z + D = 0 \qquad .. \ 4$$

where A, B, and C are the direction ratios of the normal to the plane.

Thus , if (XR, YR, ZR) represents the location of radar, the direction ratios A, B and C for the new Z axis (or the X-Y plane) shall be

$$AZ = XR - 0 = XR \qquad .. \ 5$$

$$BZ = YR - 0 = YR \text{ and,} \qquad .. \ 6$$

$$CZ = ZR - 0 = ZR \qquad .. \ 7$$

Substituting for AZ, BZ and CZ and knowing that (XR, YR, ZR) is a point on it, the value of DZ is computed as

$$DZ = - (AZ . XR + BZ . YR + CZ . ZR)$$

$$= - (XR \cdot XR + YR \cdot YR + ZR \cdot ZR)$$

$$= - R^2 \qquad .. 8$$

Therefore equation of plane is :

$$AZ \cdot X + BZ \cdot Y + CZ \cdot Z + DZ = 0$$

i.e., $XR \cdot X + YR \cdot Y + CZ \cdot Z - R \cdot R = 0$ .. 9

Similarly, the 'Y' axis shall be the line joining (XR,YR,ZR) and the point of intersection of the axis of rotation with a plane tangential to the earth's surface at the radar location. Using equation 9 the location of such a point (0,0,ZN), where ZN = R*R /ZR, can be found . The direction ratios A, B and C for the new Y axis (or the Z-X plane) and the corressponding D shall be

$$AY = XR - 0 , \qquad .. 10$$

$$BY = YR - 0 , \qquad .. 11$$

$$CY = ZR - ZN \text{ and} \qquad .. 12$$

$$DY = - (AY \cdot XR + BY \cdot YR + CY \cdot ZR)$$

$$= - (XR \cdot XR + YR \cdot YR + (ZR - ZN) \cdot ZR)$$

$$= - R^2 + ZN*ZR$$

$$= - R^2 + R^2$$

$$= 0 \qquad .. 13$$

The 'X' axis of the new coordinate reference shall be in the direction EAST-WEST of the radar location. However, unlike in case of other two axes, no point other than the radar location itself, is known on this axis. Hence, 'X' axis being perpendicular to both 'Y' and 'Z' axis, the computation of vector product of their equation has been resorted to. Consequently, for 'X' axis A, B, C and D are found as

$$AX = BY \cdot CZ - BZ \cdot CY, \quad .. 14$$

$$BX = CY \cdot AZ - CZ \cdot AY, \quad .. 15$$

$$CX = AY \cdot BZ - AZ \cdot BY \text{ and} \quad .. 16$$

$$DX = -(AX.XR + BY.YR + CX.ZR) \quad .. 17$$

With this, we complete the defining of new coordinate reference at the radar location. In order to map a point from the absolute frame to this new frame one needs to compute the distance of that point from the three planes. A general formula for computing the distance 'D_o' of a point (X_o ,Y_o ,Z_o) from a plane

(A . X + B . Y + C . Z + D = 0) is

$$D_o = \left| \frac{A.X_o + B.Y_o + C.Z_o + D}{(A^2 + B^2 + C^2)^{1/2}} \right| \quad .. 18$$

The '**absolute function**' used in this formula is only to signify the absurdity of negative distances. However, in the current context a negative values refers to the negative domain of the coordinate axis. The absolute function, must therefore be dropped. The coordinate transformation equations, as a result, look like :

$$X_{out} = \frac{AX \cdot X_{in} + BX \cdot Y_{in} + CX \cdot Z_{in} + DX}{(AX^2 + BX^2 + CX^2)^{1/2}} \quad .. 19$$

$$Y_{out} = \frac{AY \cdot X_{in} + BY \cdot Y_{in} + CY \cdot Z_{in} + DY}{(AY^2 + BY^2 + CY^2)^{1/2}} \quad .. 20$$

$$Z_{out} = \frac{AZ \cdot X_{in} + BZ \cdot Y_{in} + CZ \cdot Z_{in} + DZ}{(AZ^2 + BZ^2 + CZ^2)^{1/2}} \quad .. 21$$

Suffices **in** and **out** refer to input and output parameters respectively. The transformation equations 19, 20 and 21 can be used to transform any point in the absolute frame of reference onto that of the radar. We, however, required coordinate mapping from each radar to the CRC. To achieve this, a CRC location reference is to be first defined. After its location cartesian coordinates are known, they are mapped onto the radar's reference frame using the above equations. The other two points namely and for the CRC location are also similarly mapped. As was shown earlier, these three points could now be used to define the CRC's coordinate frame in that or the radar's and the transformation equations derived. With location of radars and CRC known, the conversion coefficients can be calculated offline resulting in a reduction in computational load in the real time process. The equations 19, 20 and 21 can be expressed as :

$$\begin{bmatrix} O_X \\ O_Y \\ O_Z \end{bmatrix} = \begin{bmatrix} T_{XX} & T_{XY} & T_{XZ} \\ T_{YX} & T_{YY} & T_{YZ} \\ T_{ZX} & T_{ZY} & T_{ZZ} \end{bmatrix} * \begin{bmatrix} I_X \\ I_Y \\ I_Z \end{bmatrix} + \begin{bmatrix} R_X \\ R_Y \\ R_Z \end{bmatrix}$$

Or $\mathbf{O} = \mathbf{T} * \mathbf{I} + \mathbf{R}$ .. 21

Where

O is the output (transformed) coordinates vector

T is the transformation coefficients matrix

I is the input (untransformed) coordinates vector

and **R** is the radar's location coordinates vector

To summarise , the various steps involved are :

Non real time :-

1. Find the location cartesian coordinates of each radar and the CRC from the respective latitude and longitude and the radius of the earth.

2. For each radar location proceed as follows :

   2.1 Designate the radar's location as the new origin.

   2.2 Get a point (0,0,ZN) in the plane of the radar and intersecting the Z axis of the absolute reference.

   2.3 Compute the direction ratios for the directions :

   (a) ZENITH (Z) - Using the centre of earth ,

   (b) NORTH (Y) - Usin the point (0,0,ZN) and

   (c) EAST (X) - By the vector product .

   2.4 Use these direction ratios and the origin to get the equations of the three planes that forms the new frame of reference.

   2.5 Get the transformation coefficients to map any point in the absolute reference onto the radar's reference.

   2.6 Using these coefficients in the distance formula to map the location of CRC centre of earth and the point (0,0,ZN) in plane of CRC, onto the new reference frame at the radar location.

   2.7 Designate the CRC location as the new origin and compute the direction ratios of the directions ZENITH, NORTH and EAST.

2.8 Use the equations of planes that forms the CRC's coordinate system.

2.9 Get the transformation coefficients and store them for ready use.

Real time :-

On receipt of the coordinates, find the new transformed coordinates using the pre-computed transformation coefficients and the incoming coordinates using equations 18, 19 and 20.

The above coordinate transformation method perform significantly less processing in real time while meeting the accuracy requirements as well. Also it is best suited for three dimensional radars. For 2D radars where the height component is not available Z_{in} may either be kept zero or a default value of height chosen. The inaccuracy so introduced shall not be significant.

ALGORITHM FOR COORDINATE CONVERSION

Input for Initiation :

Latitude and lagnitude from each radar location

Input for processing :

Plots for coordinate transfer from absolute frame of referance to central referance coordinate system

Algorithm :

```
location  = record
     x,y,z : real ;
end;
loc : array [1..n] of location;
d_ratio_1, d_ratio_2, d_ratio_3 :array [1..3] of real;
con_d                           :array [1..3] of real;

for i=1 to n do
begin
     getloc(lat,long);
     loc[i].x := R * cos(lat) * cos(long);
     loc[i].y := R * cos(lat) * sin(long);
     loc[i].z := R * sin(lat);
end;
getlocCRC(lat,long);
crc.x := R * cos(lat) * cos(long);
crc.y := R * cos(lat) * sin(long);
crc.z := R * sin(lat);
Zn := (R * R)/loc[i].z ;
point.x := point.y := 0;
point.z := Zn ;
d_ratio_3[1] := loc[i].x ;
d_ratio_3[2] := loc[i].y ;
d_ratio_3[3] := loc[i].z ;
con_d[3]    := - R * R ;
d_ratio_2[1] := loc[i].x ;
```

```
d_ratio_2[1] := loc[i].y ;
d_ratio_2[1] := loc[i].z - Zn ;
con_d[2]   := 0 ;
d_ratio_1[1] := loc[i].y * Zn ;
d_ratio_1[1] := - Zn * loc[i].x ;
d_ratio_1[1] :=  0;
con_d[1]  := 0 ;
d1 := sqrt(sqr(d_ratio_1[1]) + sqr(d_ratio_1[2]) + sqr(d_ratio_1[3]);
d2 := sqrt(sqr(d_ratio_2[1]) + sqr(d_ratio_2[2]) + sqr(d_ratio_2[3]);
d3 := sqrt(sqr(d_ratio_3[1]) + sqr(d_ratio_3[2]) + sqr(d_ratio_3[3]);
 for j:=1 to 3 do
 begin
      d_ratio_1[j] := d_ratio_1[j]/d1 ;
      d_ratio_2[j] := d_ratio_2[j]/d2 ;
      d_ratio_3[j] := d_ratio_3[j]/d3 ;
 end;
 con_d[1] := con_d[1] / d1 ;
 con_d[2] := con_d[2] / d2 ;
 con_d[3] := con_d[3] / d3 ;
store_trans_from_abs_to_radar (i,d_ratio_1, d_ratio_2, d_ratio_3,
crc.x := crc.x * X[1] + crc.y * X[2] + crc.z * X[3] ;
origin.x := origin.x * X[1] + origin.y * X[2] + origin.z * X[3] ;
point.x := 0 * X[1] + 0 * X[2] + (R*R)/crc.z * X[3] ;
  Y[1] := crc.x - point.x ;
  Y[2] := crc.y - point.y ;
  Y[3] := crc.z - point.z ;
  D[2] := -( Y[1] * crc.x + Y[2] * crc.y + Y[3] * crc.z) ;
  Z[1] := crc.x - origin.x ;
```

```
Z[2] := crc.y - origin.y ;
Z[3] := crc.z - origin.z ;
D[3] := -( Z[1] * crc.x + Z[2] * crc.y + Z[3] * crc.z) ;
X[1] := Y[2] * Z[3] - Z[2] * Y[3];
X[2] := Y[3] * Z[1] - Z[3] * Y[1];
X[3] := Y[1] * Z[2] - Z[1] * Y[2];
D[1] := -( X[1] * crc.x + X[2] * crc.y + X[3] * crc.z) ;
e1 := sqrt(sqr(X[1]) + sqr(X[2]) + sqr(X[3]));
e2 := sqrt(sqr(Y[1]) + sqr(Y[2]) + sqr(Y[3]));
e3 := sqrt(sqr(Z[1]) + sqr(Z[2]) + sqr(Z[3]));
```

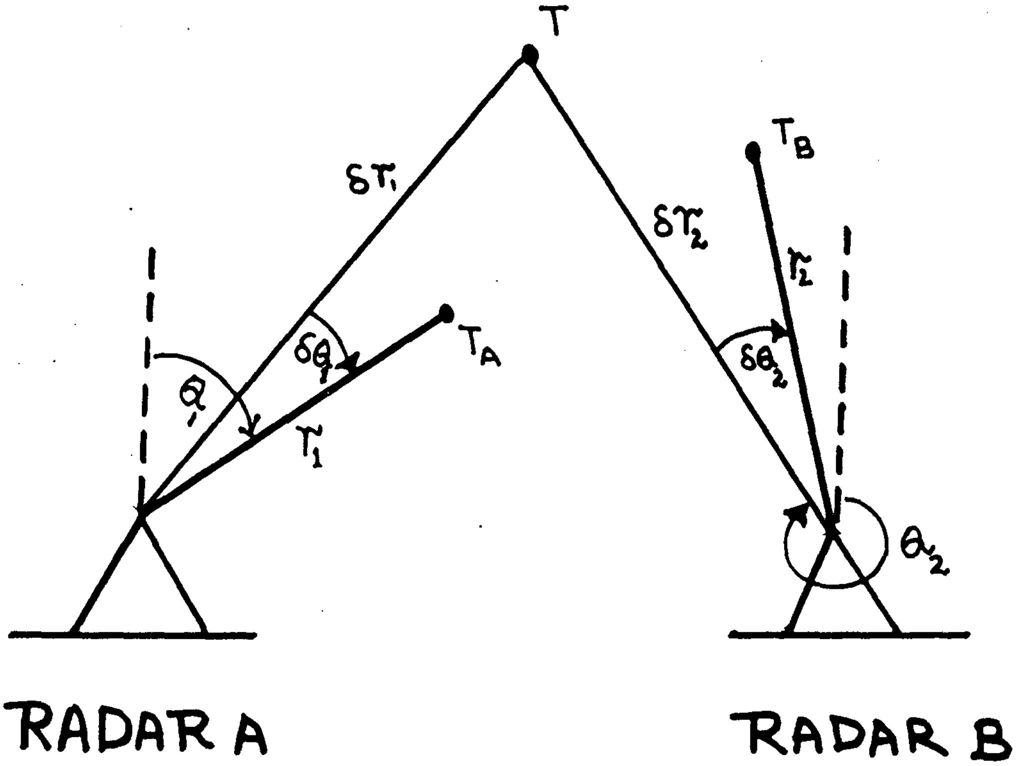
```
  for j := 1 to 3 do
  begin
       X[j] := X[j] / e1 ;
       Y[j] := Y[j] /e2 ;
       Z[j] := Z[j] /e3 ;
  end ;
  D[1] := D[1] /e1 ;
  D[2] := D[2] /e2 ;
  D[3] := D[3] /e3 ;
  store_trans_from_radar_to_crc (i,X, Y,Z, D) ;
end.
```

Run time :-

```
 for ( radar i ) do
if input plot is in loc_plot then
begin
      Plot := trans_from_abs_to_radar ( i, loc_plot ) ;
      Plot := trans_from_abs_to_radar ( i, Plot )
end ;


procedure trans_from_abs_to_radar (i, Plot)
begin
   retreive_from_abs_to_frame (i, X, Y, Z, D) ;
   out[1] := X[1] * Plot[1] + X[2] * Plot[2] + X[3] * Plot[3] + D[3];
   out[2] := Y[1] * Plot[1] + Y[2] * Plot[2] + Y[3] * Plot[3] + D[3];
   out[3] := Z[1] * Plot[1] + Z[2] * Plot[2] + Z[3] * Plot[3] + D[3];
   return (out) ;
 end ;
```

Fig. 7 REGISTRATION ERROR VERSUS REPORTED AIRCRAFT POSITION.



T = TRUE TARGET POSITION

T_A = RADAR A REPORTED POSITION (r_1, θ_1)

T_B = RADAR B REPORTED POSITION (r_2, θ_2)

$\delta\theta_1, \delta\theta_2$ = AZIMUTH BIAS OF RADARS A AND B

$\delta r_1, \delta r_2$ = RANGE BIAS OF RADARS A AND B

```
procedure trans_from_radar_to_crc (i, Plot)
  begin
  retreive_from_radar_to_crc (i, X, Y, Z, D) ;
  out[1] := X[1] * Plot[1] + X[2] * Plot[2] + X[3] * Plot[3] + D[1];
  out[2] := Y[1] * Plot[1] + Y[2] * Plot[2] + Y[3] * Plot[3] + D[2];
  out[3] := Z[1] * Plot[1] + Z[2] * Plot[2] + Z[3] * Plot[3] + D[3];
  return (out) ;
 end.
```

CHAPTER 3

3.1 THE RADAR REGISTRATION PROBLEM

In case of Multi Radar Tracking the individual radar data must be expressed in a common coordinate system in which the errors due to site uncertainties, antenna orientation, and improper calibration of range and time have been minimized so they do not cause a significant degradation of the system operation. The process of ensuring the requisite "error free" (or, more precisely, controlled error) coordinate conversion of radar data is called registration. Thus, registration is an absolute prequisite for multiple radar tracking or sensor netting in general.

3.2 REGISTRATION ERRORS :

The objective of this section is twofold : first, the major sources of registration error in multiple radar systems will be identified; second, the possible impact of these errors on the data association and track processes will be discussed on a qualitative level to asses their significance.

3.2.1 SOURCES OF REGISTRATION ERROR

The major sources of registration error for radars are given in the left-hand column of Table 1 ,and some possible corrective actions in the right-hand column.

TABLE - 1

Error source	Corrective measure
Range :	
offset	Test targets
scale	Factory calibration
Atmospheric refraction	Tabular corrections
Azimuth :	
offset	Solar allignment Electronic North reference modules
Antenna tilt	Electrónic leveling
Elevation :	
offset	Test targets
Antenna tilt	Electronic leveling
Time :	
offset	Common electronic time reference
scale	Factory calibration
Radar location	Electronic position location
Coordinate conversion :	
radar plane	3D radars
system plane	Exact or second order stereographic proj.

Of the sources of registration error listed in Table 1, four sources have proved to be major problems in current defense and air traffic control systems :

(1) position of the radar with respect to the system coordinate origin ,

(2) alignment of the antennas with respect to a common North reference ,

(3) range offset errors, and

(4) coordinate conversion with 2D radars.

The other errors very well may exist in the current radar systems; however, they have not been significant problems in the past. As communications and radar technology improves, these other error sources could not become more significant in the future.

The potential effects of range and azimuth offset errors are illustrated in Fig 7 . Registration errors are systematic, not random, errors in the reported aircrafts position; large errors will result in two apparent aircraft when only one real aircraft exists. Fig.7 shows the expected or average reports for a common target from two radars, each of which consistently reports **(1)** a range less than the true range by a fixed amount (i.e.,the offset) and **(2)** an azimuth (measured clockwise from North) less than the true azimuth by a fixed offset. For any specific set of measurements, the random measurement errors will be superimposed on the bias or offset errors.

3.2.2 EFFECT OF REGISTRATION ERROR ON TRACKING

The effect of of systematics errors is to introduce biases into the track estimation process. Therefore, failure to register adequately a multiple radar system can result in varying degree of performance degradation, depending on the magnitude of the biases w.r.t. the random measurement errors and the track correlation gates. The level of degradation ranges, at worst, from the formation of multiple redundant tracks for a single aircraft to reduced track accuracy and stability, when the bias is relatively small. In between the benefits of a multiple radar system can be negated and the system, in effect, reduced to a single radar tracking system.

3.3 REGISTRATION PROCEDURE :

System registration may be considered as a two-phase process : sensor initialization and relative alignment. The objective of the initial registration procedure is to register each sensor independently, with respcet to absolute coordinates. Once the position of the sensor has been estimated, the range measurements have been calibrated, and an initial alignment with respect to true North has been completed, we can initiate the procedures for relative alignment of the system sensors. The initialization procedure is generally straightforward; the difficult part of registration is the relative alignment of the sensors.

Techniques for relative registration depends on common targets. Generally ,data are collected until a sufficient number of paired reports have been obtained, and then a set of bias corrections are computed. The usual technique for obtaining the solutions is either to formulate the problem as an ordinary **least-square estimation** (LSE) problem or to rely on simple averaging to remove the random error components. The major limitation of either approach is that radar report is treated equally when, in fact, the measurement errors are a function of both the individual radar parameters and target range.

3.4 BIAS ESTIMATION :

Three alternative approaches for registration have been sugested by Ficsher, Muehe, and Cameron [Fischer,80]; specially, the **generalizeg linear least-squres estimation (GLSE)** technique and two numerical optimization methods, one based on a grid search technique and the other on Powell's method for steepest descent.

Commercial array processors or special purpose coprocessors now can perform the large scale matrix operation required by the GLSE aproach (see [Anderson,58] and [Mardia,79]). This technique developed by Wax [Wax,83] for aircraft location with sensors at uncertain locations can be applied to formulate the generalized Gauss-Markov problem. Moreover, the solutiom can be reduced to a computationally tractable algorithm, as we will demonstrate.

3.4.1 MATHEMATICAL DEVELOPMENT :

In the following derivation, assume that master radar R_A is located at the origin of the coordinate system and that a subordinate radar R_B is located at coordinates (u,v). Also assume that there are N targets in the intersection of the respective fields of view, denoted by $\{T_1, T_2, \ldots, T_N\}$.

The basic problem is to determine the range and azimuth biases at each radar from the measurements of the set of common targets $\{T_1, T_2, \ldots, T_N\}$. That is, we need to estimate the azimuth biases (offsets) $\Delta\theta_A$ and $\Delta\theta_B$ at $.R_A$ and $.R_B$, respectively, and the range biases Δr_A and Δr_B at $.R_A$ and $.R_B$. denote the vector of biases by

$$\tilde{A} = (\Delta r_A, \Delta\theta_A, \Delta r_B, \Delta\theta_B)^T \qquad . . (3.1)$$

Where the superscript. T denote the transposition operator.

For each target T_k, define the vector of radar measurements

$$\Psi_k = (r_{AK}, \tilde{\theta}_{AK}, r_{BK}, \tilde{\theta}_{BK})^T \qquad . . (3.2)$$

Where $(r_{AK}, \tilde{\theta}_{AK})$ and $(r_{BK}, \tilde{\theta}_{BK})$ denote the range and azimuth measurements from radar R_A and radar R_B, respectively, and the index k denote the sequence of independent measurement over time.

For each set of measurements, Ψ_k , the observations are the seperations in the system (x,y) plane of the reported target positions.

These are

$$x_K = [r_{AK} + r_A] \cos[\tilde{\theta}_{AK} + \tilde{\theta}_A] - u - [r_{BK} + r_B] \cos[\tilde{\theta}_{BK} + \tilde{\theta}_B]$$
$$y_K = [r_{AK} + r_A] \sin[\tilde{\theta}_{AK} + \tilde{\theta}_A] - v - [r_{BK} + r_B] \sin[\tilde{\theta}_{BK} + \tilde{\theta}_B]$$

$$... (3.3)$$

To apply the theory of generalized least-squre estimation, we need to represent the observations as a linear function of parameters to be estimated, namelly $\tilde{A}$. This can be accomplished by defining a function **f** as follows :

$$\mathbf{f}(\Psi_K, \tilde{A}) = [\Delta x_K, \Delta y_K]^T \qquad . . (3.4)$$

Further, let Ψ_K' and $\tilde{A}'$ denote the actual measurement sets and an initial esimate of $\tilde{A}$, respectively. Now, Taylor's theorm can be used to approximate the function **f** at the true values of

Ψ_K and $\tilde{A}$ in terms of the measurements Ψ'_K and the initial estimate $\tilde{A}'$:

$$\mathbf{f}(\Psi_K, \tilde{A}) = \mathbf{f}(\Psi'_K, \tilde{A}') + \nabla_{\tilde{A}}[\mathbf{f}(\Psi_K, \tilde{A})](\tilde{A} - \tilde{A}') + \nabla_{\Psi}[\mathbf{f}(\Psi'_K, \tilde{A})](\Psi_K - \Psi'_K) \quad . . (3.5)$$

Where the differential operators $_{\tilde{A}}$ and are defined as follows :

$$\nabla_{\Psi}[f(\Psi'_K, \tilde{A}')] = \begin{bmatrix} \frac{\partial(\Delta x_K)}{\partial r_{AK}} & \frac{\partial(\Delta x_K)}{\partial \theta_{AK}} & \frac{\partial(\Delta x_K)}{\partial r_{BK}} & \frac{\partial(\Delta x_K)}{\partial \theta_{BK}} \\ \frac{\partial(\Delta y_K)}{\partial r_{AK}} & \frac{\partial(\Delta y_K)}{\partial \theta_{AK}} & \frac{\partial(\Delta y_K)}{\partial r_{BK}} & \frac{\partial(\Delta y_K)}{\partial \theta_{BK}} \end{bmatrix} = F_K$$

$$\nabla_{\tilde{A}}[f(\Psi'_K, \tilde{A}')] = \begin{bmatrix} \frac{\partial(\Delta x_K)}{\partial(\Delta r_A)} & \frac{\partial(\Delta x_K)}{\partial(\Delta\theta_A)} & \frac{\partial(\Delta x_K)}{\partial(\Delta r_B)} & \frac{\partial(\Delta x_K)}{\partial(\Delta\theta_B)} \\ \frac{\partial(\Delta y_K)}{\partial(\Delta r_A)} & \frac{\partial(\Delta y_K)}{\partial(\Delta\theta_A)} & \frac{\partial(\Delta y_K)}{\partial(\Delta r_B)} & \frac{\partial(\Delta y_K)}{\partial(\Delta\theta_B)} \end{bmatrix} = G_K \quad . . (3.6)$$

For convenience later in the development, the 2 X 4 matrices $\nabla_{\Psi}[\mathbf{f}(\Psi_K', \tilde{A}')]$ and $\nabla_{\tilde{A}}[\mathbf{f}(\Psi_K', \tilde{A}')]$ has been labeled $\mathbf{F}_k$ and $\mathbf{G}_k$, respectively. If the errors $(\Psi_K - \Psi'_K)$ and $(\tilde{A} - \tilde{A}')$ are sufficiently small that the higher-order terms can be neglected, the approximation in (4.5) may be regarded as an equality. Note that

$$[\mathbf{f}(\Psi_K, \tilde{A})] = 0 \quad . . (3.7)$$

by definition; therefore,

$$\mathbf{G}_k\tilde{A} + \mathbf{F}_k\partial\Psi_K = \mathbf{G}_k\tilde{A}' - \mathbf{f}(\Psi'_K, \tilde{A}') \quad . . (3.8)$$

where $\partial\Psi_K = (\Psi_K - \Psi'_K)$. Also, note that the matrix $\mathbf{G}_k$ is a matrix of known parameters, $\mathbf{F}_k\,\partial\Psi_K$ is error due to the measurement noise, and that the terms on the right hand side of (3.8) now represent the observations.

With all this notation and the approximation of (3.5),(3.4) (3.3) may now be reformulated as the classical model of **GLSE** theory (see, for example, [Anderson,58] or [Mardia,79]).

$$\mathbf{X}\breve{A} + \xi = \mathbf{Y} \quad . . (3.9)$$

by setting

$$\mathbf{X} = [\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, \ldots, \mathbf{G}_N]^T \quad . . (3.10)$$

$$\xi = [F_1 \partial\psi_1, F_2 \partial\psi_2, \ldots F_N \partial\psi_N]^T \quad . . (3.11)$$

$$= [G_1 \breve{A}' - f(\psi_1', \breve{A}'), G_2 \breve{A}' - f(\psi_2', \breve{A}') \ldots G_N \breve{A}' - f(\psi_N', \breve{A}')]^T$$

$$. . (3.12)$$

Note that parameter matrix **X** is of dimension 2N X 4 whereas the error vector ξ and the observation vector **Y** are of dimension 2N.

The last step in the application of the Gauss-Markov model is to develop the covariance Σ_ξ matrix for the error vector ξ. To this end, define a 2N X 2N matrix :

$$\Sigma_\xi = E[\xi\xi^T] = \{F_i E[(\partial\psi_i)(\partial\psi_j)^T] F_j^T \mid i, j = 1, 2, \ldots N\}$$

$$. . (3.13)$$

The terms in (3.13) can be simplified if it is noted tha the measurement vectors or sets ψ_i' and ψ_j' are independent therefore,

$$E[(\partial\psi_i)(\partial\psi_j)^T] = 0 \quad . . (3.14)$$

if i ≠ j. if i = j, then

$$\Sigma_\psi = E[(\partial\psi_i)(\partial\psi_i)^T] = \begin{bmatrix} \sigma_\gamma^2(A) & 0 & 0 & 0 \\ 0 & \sigma_\theta^2(A) & 0 & 0 \\ 0 & 0 & \sigma_\gamma^2(B) & 0 \\ 0 & 0 & 0 & \sigma_\theta^2(B) \end{bmatrix} \quad . . (3.15)$$

Now note that F_k is a 2 X 4 matrix and that Σ_ψ is a 4 X 4 matrix; therefore,

$$\Sigma_k = F_k \Sigma_\psi F_k^T \qquad \ldots (3.16)$$

is a 2 X 2 matrix. This implies that, finally, Σ_ξ is a block diagonal matrix with the 2 X 2 blocks $\{\Sigma_1, \Sigma_2, \ldots, \Sigma_N\}$ along the main diagonal and zeros in the off-diagonal positions.

The solution of the Gauss-Markov equation, (3.9), is simply

$$\tilde{A}^* = (X^T \Sigma_\xi^{-1} X)^{-1} X^T \Sigma_\xi^{-1} Y \qquad \ldots (3.17)$$

where

$$\mathrm{Cov}(\tilde{A}^*) = (X^T \Sigma_\xi^{-1} X)^{-1} \qquad \ldots (3.18)$$

because Σ_ξ is a 2N X 2N block-diagonal matrix, we thus have

$$X^T \Sigma_\xi^{-1} X = \sum_{k=1}^{N} G_k^T \Sigma_k^{-1} G_k \qquad \ldots (3.19)$$

where the individual terms of the sum are 4 X 4 matrices. Similarly,

$$X^T \Sigma_\xi^{-1} Y = \sum_{k=1}^{N} G_k^T \Sigma_k^{-1} [G_k \tilde{A}' - f(\psi_k', \tilde{A}')] \qquad \ldots (3.20)$$

If the individual radar measurement error are normally distributed $\partial\Psi_k$ is a normally distributed vector; $F_k \Psi_k$ is a linear combination of normal variables, so it is normally distributed. Thus is distributed as **N(0 , Σ_ψ)**.

Equation (3.17) is the minimum variance solution under any error distribution. For the normal distribution (that is, distributed as **N(0, Σ_ψ)**), $\tilde{A}^*$ also is the maximum- likelihood solution. By these criteria, $\tilde{A}^*$ in (3.17) is the "best" solution to the minimum variance problem as defined by (3.9) .

The estimation technique just outlined attempts to minimize the effects of registration errors in the system plane with corrections in the radar measurement plane. If the transformation (3.3) of the data from the radar to the radar system plane introduces an error of the same order of magnitude as the observation errors, then the utility of the solution is open to question. Fortunately, the issue can be resolved by use of the second-order stereographic transformation between planes. As shown by Burke [Burke,73], the error induced by the second-order stereographic transformation is less than 2 m over any realistic sensor geometry. This transformation error is at least one order of magnitude less than the random measurement errors of modern surveillance radars.

CHAPTER 4

DATA FUSION INCLUDING ATTRIBUTES

4.1 FUSION AND CORRELATION FOR DATA INCLUDIN ATTRIBUTES

This section develops a Bayesian mathematical structure based upon [Atkinson,80], under which observation containing attribute data as well as kinematics can be combined and update estimates thereby formed. Attributes are sensed target quantitie that are associated with a particular type or class of target These may include such quantities as wheel type for ground targets engine type for aircraft, type of emitting radar for either ground or aircraft targets, or target image shape. Also, the class or type of target may itself be considered an attribute.

Most previously developed MTT system have only use kinematics quantities such as position, range rate etc. However with the use of sensors other than radar and with advances in radar signal processing techniques, the efficient use of other types of attribute data now becomes important. In particular, future military MTT systems will use a wide variety of sensors that will measure a number of different attributes. The problem is to correlate these different types of data, to make inference on the important attributes such as target type, and to assign confidence to these inferences.

Pattern recognition is one method of using multiple sensor data to determine target identification. Using this approach, we would determine the appropriate set of features to be formed from multiple sensor observation data and the best weighting (confidence level) to use with these data. This is a complex process an probably feasible for only a limited number of sensors. Also, redefinition of the features and weightings will be required when another sensor is added to the system. The approach discussed her will assume that each sensor first processes its own data. The each sensor will produce its current best estimate of target attributes. The confidence associated with such output also will be assumed to be transmitted or known from previous experience. Thus the problem becomes one of combining sensor attribute data to specify target type and the associated confidence level.

As an example of multiple sensor attribute data, fig 8 show the information that may be available to an airborne interceptor system using multiple sensors and advanced processing methods. This target attribute information may include type of engine, type of radar, target shape, response (or lack of) to interrogation friend or foe (IFF), and radar cross section (RCS). Finally, other information such as flight path characteristics and possible intended target destination also may be added from other sources.

In general, it is best to keep estimates at the attribute level. Directly converting attributes to target type may keep to some inaccuracy. For example, consider the case where radar type r1 can be carried by target types a_1 and a_2 whereas radar type r2 can be carried by target types a_2 and a_3. We assume that the target carries a single type of radar. If a return indicating the target to be carrying radar type r_1 is converted directly to target type (a_1 or a_2), the incorrect conclusion can be made that these return both correspond to the same target (type a_2). Thus, we assume than

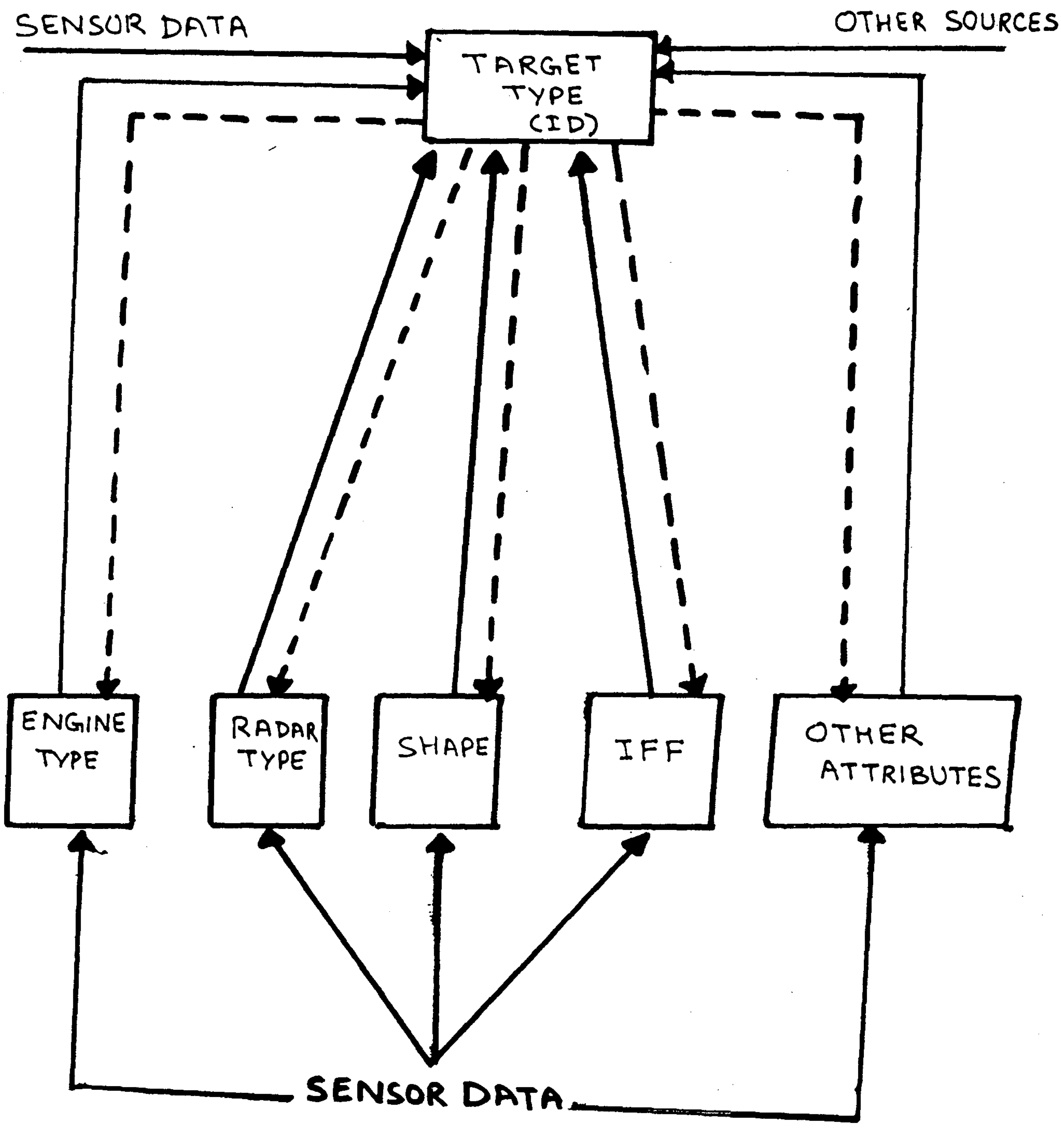


Figure - 8 Target ID Information Flow

estimates are kept on the attribute level and that these estimate are continually updated as new attribute data are received. Then attribute estimates are combined, and recombined as new data are received, to form estimates of the target type. We next outline three approaches to the process of combining target attribute and identification information.

4.2 BAYESIAN APPROACH

Application of a Bayesian approach to the attribute and target identification problem requires a priori information and conditional probabilities. First, the measurement process is defined by the folowing relationship:

$P(X_m/X)$ = probability of receiving measurement X_m given that the true quantity is X which is assumed to be known.

Then, whenever measurement data are received, the updated probabilities can be computed using bayes's rule :

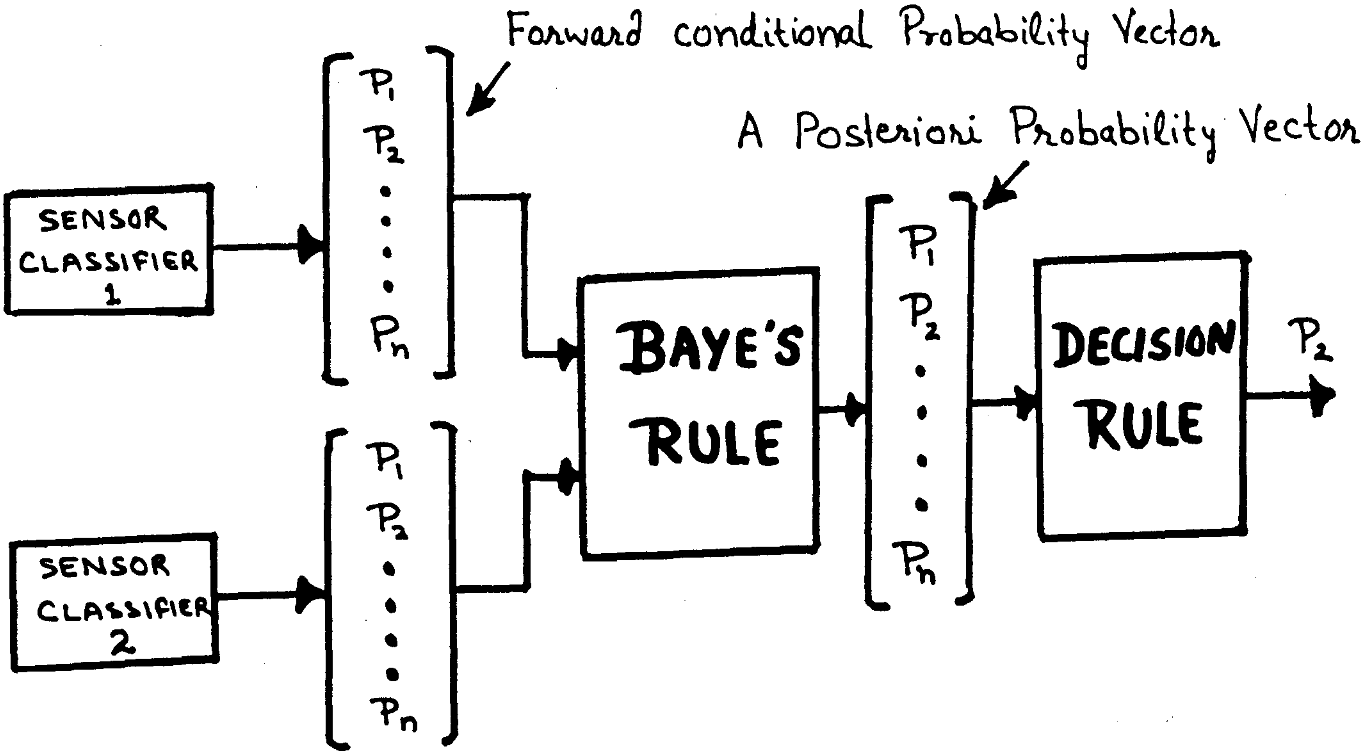


Fig. 9 Bayesian combination

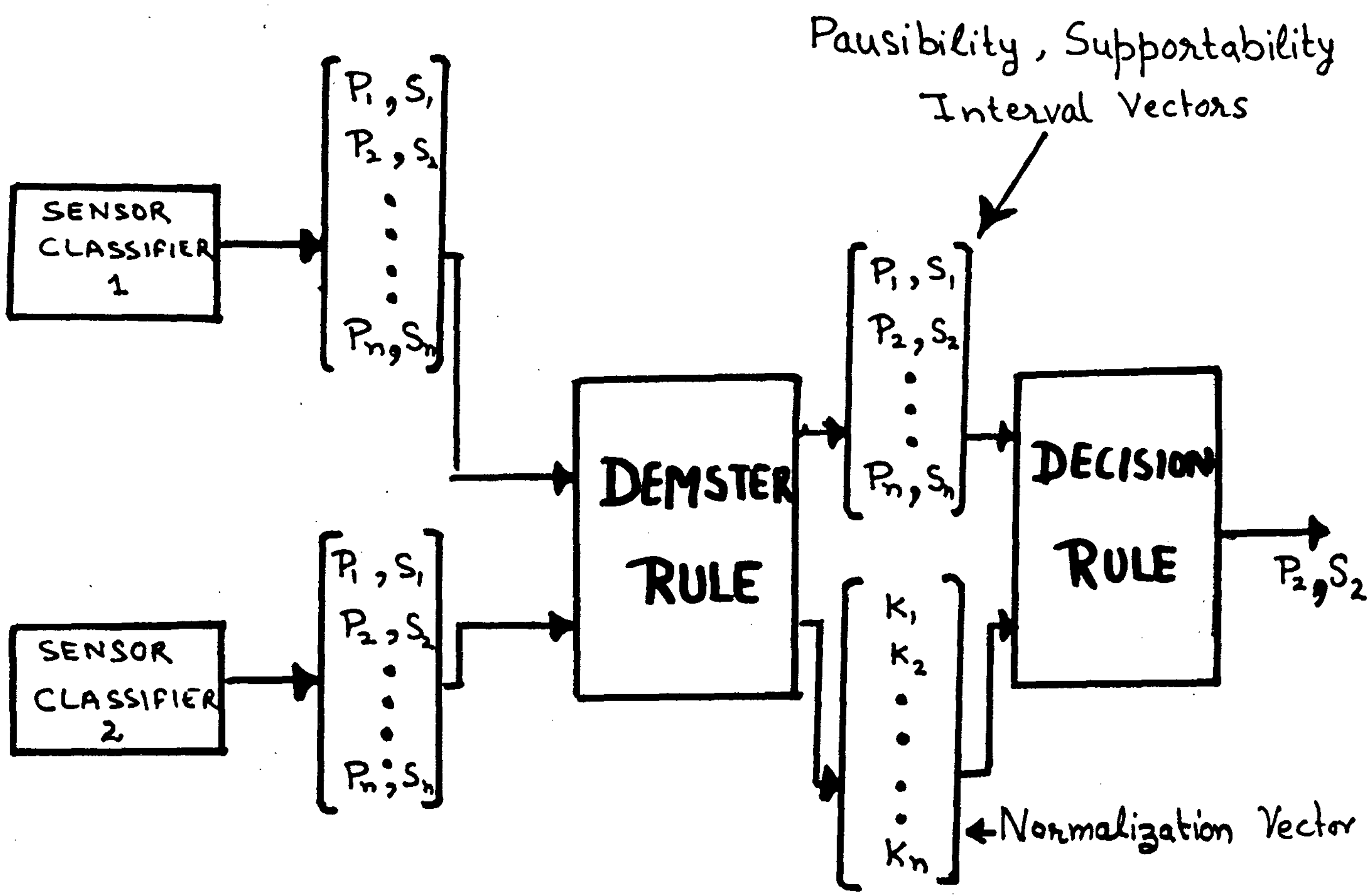


Fig. 10 Dempster - Shafer combination

$$P(X/X_m) = \frac{P(X_m/X)\,P(X)}{P(X_m)}$$

Where

$P(X)$ = prior probability associated with X

$P(X_m) = \sum_m P(X_m/X)\ P(X)$

The process continues as $P(X/X_m)$ becomes the new prior probability for use when further data is received. To initiate the process, before any received measurements, Bayesian approach require an initial estimate of the probability of X.

To summarize, Bayes's rule is applied recursively as new date recieved. This relatively simple relationship provides the estimated probablities of target attribute based directly on the measurement Xm of involved quantities. The estimated probabilitie can be improved using known interrelationships, as expressed by conditional probabilities between attributes and target types.

4.3 Dempster-Shafer Evidential Reasoning

Dempster [Demster,68]] and Shafer have developed a method that generalizes Bayesian inference and which has been denoted the Dempster-Shafer or evidential reasoning method. The evidential reasoning approach is more general than the Bayesian. Also, its development has been based upon several perceived weaknesses of the standard Bayesian formulation. A weakness of the Bayesian approach is the lack of a convenient representation for ignorance of uncertainty. The evidential reasoning method handles this situation quite simply by allowing the assignment of a probabilty mass value directly to uncertainty.

The evidential reasoning method also handles the problem of incomplete or uncertain sensor measurements. First, sensor error can be conveniently represented by a probability mass assignment directly to uncertainty. Also the use of evidential reasoning allows a more convenient and accurate representation of the information from certain sensors.

4.3.1 IMPLEMENTATION OF EVIDENTIAL REASONING

Assume that we have a set of "n" mutually exclusive and exhaustive propositions, such as that the target is type $a_1, a_2, \ldots, a_n$. The method of evidential reasoning can assign a probability mass (denoted by $m(a_1)$) to any of the original n propositions or to disjunctions of the propositions. Note that this more general form of representation differs from the standard Bayesian approach in which probabilities are assigned only to the original "n" propositions - disjunctions are not considered.

The representation of uncertainty is mass assignment to the disjunction of all the original propositions and is denoted

$$m(\tilde{\theta}) = m(a_1 \vee a_2 \vee \, . \, . \vee a_n)$$

Finally, mass can be assigned to the negation of a proposition. For example, the mass assigned to the negation of a_1 (the target is not of type a_1) is denoted

$$m(\sim a_1) = m(a_1 \vee a_2 \vee \, . \, . \vee a_n)$$

To summarize, probability masses may be assigned to individual propositions $m(a_1)$, to disjunction $m(a_1 \vee a_2)$, to uncertainty $m(\tilde{\theta})$, or to the negation of a given proposition, $m(\sim a_1)$. The sum of these masses must equal unity.

Another interesting feature associated with the Dempster-Shafer method is the concept of support and plausible for propositions. the support for a given proposition is the sum of all the masses assigned directly to that proposition. To illustrate, consider the target type example, The support($spt(a_1)$) for the basic proposition the target type is a_1 is just the mass associated with a_1 (i.e. $spt(a_1) = m(a_1)$). For a more complex proposition such as that the target is either a_1, a_2, or a $_3$, we have.

$$spt(a_1 \vee a_2 \vee a_3) = m(a_1) + m(a_2) + m(a_3) + m(a_1 \vee a_2) + m(a_2 \vee a_3) + m(a_1 \vee a_3) + m(a_1 \vee a_2 \vee a_3)$$

The plausiblity of a given proposition is the sum of all mass not assigned to its negation. Thus,

$$pls(a_1) = 1 - spt(\sim a_1)$$

Alternatively, $pls(a_1)$ can be computed by summing all masses associated with a_1 and all disjuctions, including θ that contain a_1 . For example ,

$$pls(a_1) = m(a_1) + m(a_1 v a_2) + \ldots + m(\theta)$$

The plausibility of a_1 defines the mass that is free to move to the support of a_1. The interval $[spt(a_1), pls(a_1)]$ represents the uncertaintyinterval with [0,1] representing total ignorance and [0.6,0.6] representing a certain probability of 0.6.

The manner in which data are combined from multiple sensors is through Dempster's rule of combination. this rule is an extention of Bayes' rule and its application is explained through the following example.

4.3.2 An Example Using Evidential Reasoning

Consider an example where there are four target aircraft types as defined :

a_1 = friendly interceptor

a_2 = friendly bomber

a_3 = hostile interceptor

a_4 = hostile bomber

Assume, we start the target type identification problem by noting the aircraft behavior appears to be that of the class of interceptor. However, this information is not certain so that the following mass assignment vector is defined :

$$\mathbf{m}_1 = \begin{bmatrix} m_1(\theta) = 0.4 \\ m_1(a_1 v a_2) = 0.6 \end{bmatrix}$$

The assignment of 0.4 to $m_1(\theta)$ represents the uncertainty associated with the rules used to determine that the behavior is that of the interceptor aircraft class.

Next, assume that the target does not respond to the IFF interrogation. We would expect a response from a friendly aircraft. So this indicates that the target is probably hostile, but again this is not certain. Thus , we assign to this knowledge source the following mass values :

$$\mathbf{m}_2 = \begin{bmatrix} m_2(\theta) = 0.3 \\ m_2(a_3 v a_4) = 0.7 \end{bmatrix}$$

Fig. 10a APPLICATION OF DEMPSTER'S RULE

$m_1(\theta)=0.4$	$m(a_3 v a_4) = 0.28$	$m(\theta) = 0.12$
$m_1(a_1 v a_3)=0.6$	$m(a_3) = 0.42$	$m(a_1 v a_3)=0.18$
	$m_2(a_3 v a_4)=0.7$	$m_2(\theta) = 0.3$

Dempster's rule can be used to combine m_1 and m_2 as illustrated in the Fig 10a .. The resulting mass vector is

$$\begin{bmatrix} m(\theta)=0.12 \\ m(a_1 v a_2) = 0.18 \\ m(a_3) = 0.42 \\ m(a_3 v a_4) = 0.28 \end{bmatrix}$$

Referring to Fig. 10a, Demspter's rule is implemented by forming a matrix with the probability mass assignments that are to be combined given along the first column and last row. Then, the computed elements of the matrix are the product of the probability mass values in the same row of the first column and the same column of the last row. For example, for (2,2) element of the matrix shows in Fig. 10a.

$$m(a_3) = m_1(a_1 v a_3) m_2(a_3 v a_4) = 0.6(0.7) = 0.42$$

The assignment of these elements to the resulting vector is according to the following principles :

1. The product of mass assignments to two propositions that are consistent leads to an assignment to another proposition combined within the two original propositions. For example,

$$m_1(a_1)\ m_2(a_1) = m(a_1)$$

$$m_1(a_1 \vee a_3)\ m_2(a_3 \vee a_4) = m(a_3)$$

2. Multiplying the mass assignments to uncertainty by the mass assignments to any other proposition leads to a contribution to that proposition,

$$m_1(\tilde{\theta})\ m_2(a_3 v a_4) = m(a_3 v a_4)$$

3. Multiplying uncertainty by uncertainty leads to a new assignment to uncertainty.

$$m_1(\tilde{\theta})\ m_2(\tilde{\theta}) = m(\tilde{\theta})$$

CHAPTER 5

DATA FUSION

5.1 AIM OF DATA FUSION

Data fusion has been defined as " a process dealing with the association, correlation, and combination of data and information from multiple source to achieve refined position and identity estimates for entities, and complete and timely assessments of related situations and threatts, and their significance."

The goal of the data fusion is the devlopment of a complete, accurate, concise and timely picture of a environment based on sensors that provide only limited observables, coverage, resolution and accuracy. The environment can be very complex, consisting of potentially large numbers of many classes of both stationary and moving entities. Since the analysis of individual sensor reports can lead to ambiguous, inconsistent and highly local interpretations, the fusion of multiple sensor data tends to enhance the situation understanding process. Although a spatially distributed network of hetrogeneous sensors can increase the total information available, the non- deterministic nature of the domain and the largely expectation-based character of the reasoning process effectively guarantees a degree of uncertainty in the fusion product. The uncertainty can be minimized by synergistically utilizing all available sensor-derived information and relevantly priori domain knowledge; the former provides dynamic situation information, while the latter supports real world, context-sensitive reasoning.

5.2 FUNCTIONAL LEVEL FUSION MODEL

An abstract, two level functional model of the tactical data fusion process is :

5.2.1 Level-1 fusion

Level-1 fusion represents predominantly **information extraction** related to the detection, association, classification and attribute refinement, normally associated with single entities, based on the analysis of single sensor and multiple sensor measurements. Lavel-1 processing is largely numeric in character since the measurements are metric in nature and amenable to the application of algorithms(typically statistical and estimation/optimization methods).

5.2.2 Level-2 fusion

Level-2 fusion is based primarily on the current situation description, available a priori(static) domain knowledge and expectation-based world models. Levels-2 fusion extends and enhance the completeness, consistency, and level-of-abstraction of the situation description produced by level-1 to :

1. compensate for the information-deficient and errorful measurement space,
2. resolve ambiguity in the level-1 products and
3. develop higher level interpretations of the current based on reasoning in context .

The first stage of level-2 fusion performs **situation abstraction** which include both **situation generalization** and

situation specialization. **Situation generalization** allows bottom-up abstraction of entities or events that are either not directly measurable, or not observed, to be either inferred or the sensor network task to provide critical missing information that supports such inference. **Situation specialization** is a form of top-down reasoning where subordinate elements are deduced or inferred. Situation generalization and specialization develop the structural organizational and functional relationship among domain elements and suppots developments of a consistent, complete and higher leve-of-abstraction situation description. Level-2 processing is distinguished from Level-1 processing by a significant shift in emphasis to symbolic rather than numeric processing.

Once a new target is detected, the fusion system charcterize by the multiple level model just described attempts to varify the targets presence, refine its location and trajectory, it's measurable attributes, its probable classification, its association with a perticular unit, the association between that unit, the current enemy organisation and the preceived situation. Thus,the overall fusion process consist of numerous dependent and independent multiple level of abstraction functional processes. Since individual sensors may observe the same target at different times and have very different processing and reporting delays, and since estimates of the situation description are asynchronous with respect to the sensor acquisition process, fusion process are often inherently asychronous. Both the sensor systems and the processing nodes can be spatially distributed. Real time performance is mandated by time-critical nature of the tactical environment. The dynamic charcter of domain and the limited information content of sensor reports insures a degree of uncertainty in the situation development process.

5.3 Data fusion reasoning classes

Data fusion is supported by three underlying reasoning classes **spatial, hierarichical** and **temporal**. Spatial reasoning deals with the spatial relationships among entities (e.g.,distance metrics, relative locations, doctrinal patterns, apparent goal states), as well as their association with geographic and cultural domain features (e.g.,suppotability, mobility, visibility and communicability). Hierarchical reasoning is required due to 1) the predominantly vertical organisation of military entities (e.g. vehicles, companies, battalions), 2) thé multiple level of abstraction nature of the reasoning process (e.g., local strategies vs more global strategies), 3) the inherent efficienct of top-down problem solving.

Because of the dynamic nature of the tacticle domain and the asynchronous nature of the sensor reports, temporal reasoning underlies the entire fusion process.

5.4 Memory elements

The fusion of multiple sensors information to enhance situation understanding in a tacticle environment is metaphorically similar to the fusion of multiple sensory (e.g., sight and sounds information for perception enhancement in biological systems. Based on an extention of the human memory metaphor, the primary elements of the fusion process can be associated with short term, medium term and long term memory [Antony,87]. Short term memory holds highly transient short term knowledge, medium term memory holds dynamic, but less transient medium term knowledge and long term memory holds relatively static long term knowledge. Long term knowledge in biological system as well as data fusion systems represents factual and interpretive reasoning knowledge.

A primary objective of an automated tactical situation understanding system is the development of the current relevant perception in the fusion system's metphorical medium term memory element. For automated fusion system, current emphasizes the character of the dynamically changing scene under observation, as well as the time- evolving interaction among a network of distributed processes. Because of memory limitations and the critical role medium term memory can play in system survivability, only relevent states are maintained by tactical fusion system. For a frog, danger and moving insects represent highly relevent states, while for a human driving a car, the relative location of nearby vehicles, stoplights and pedestrians represent highly relevent states. The notion of relevence proves to be crucially important to supporting decision-making under stress, and is a challenge to the fusion system designer to develop systems which prevent degradation in reasoning and sub-optimal stress-related coping patterns.

Summary :

Tactical data fusion systems requires fusion among :

1) sensor derived information,
2) the current situation understanding and
3) relatively static factual and procedural information as dipicted in figure 11.

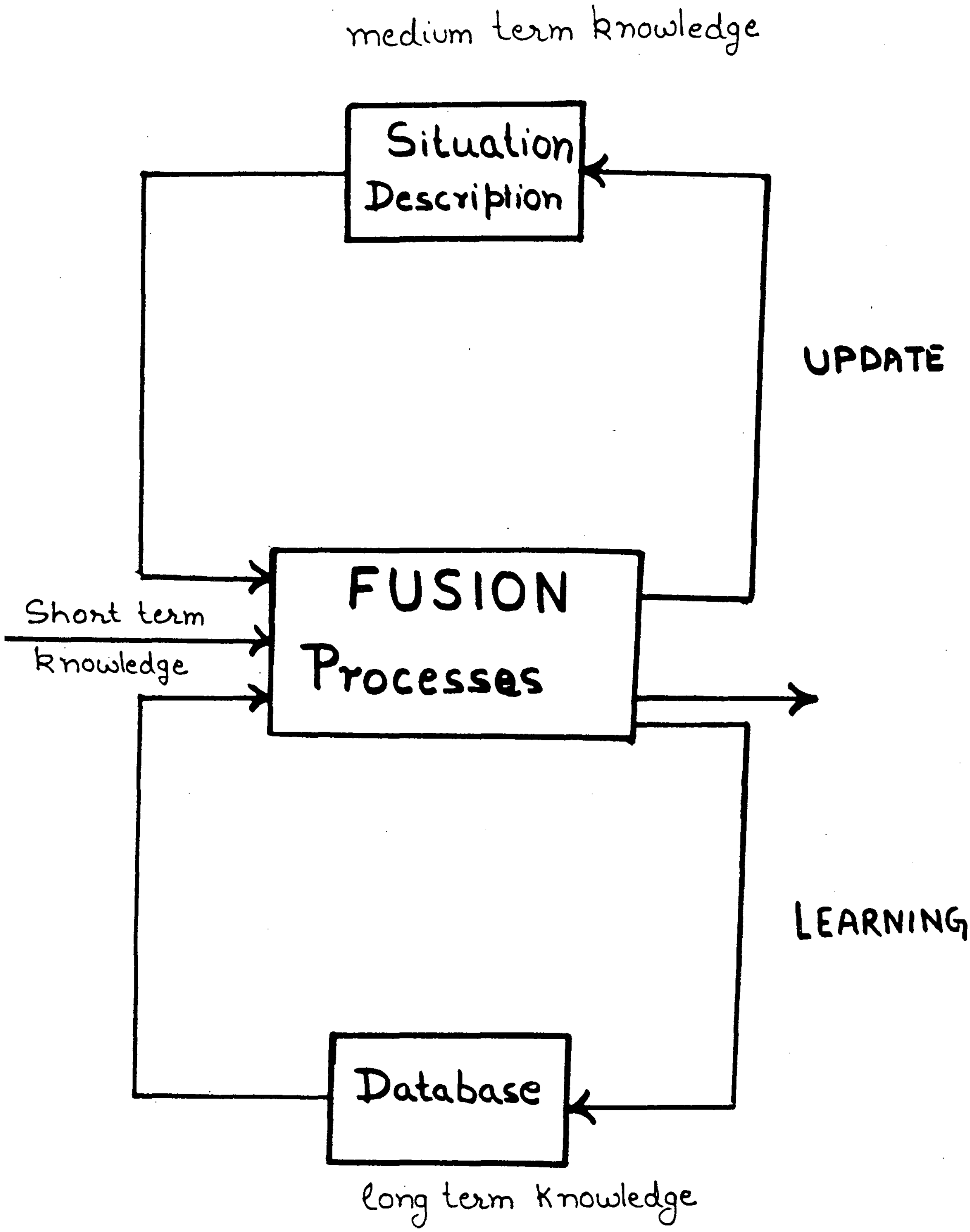


Fig. 11 Biologically - Based Metaphor of Data FUSION PROCESS

CHAPTER 6

BLACK BOARD ARCHITECTURE FOR IMPLENTATION

6.1 Black Board Architecture

The opportunistic collaboration of a group of human experts from multiple disciplines seated in front of a chalkboard provide the problem solving metaphor for the BB paradigm. As their expertise permits, individual experts participates in the evolutionary development of the problem solution that is maintained on the chalkboard. The true human metaphor is opportunistics however, it has proven difficult to achieve this explicit behavior in computer- based systems due to concurrency, consistency, and control factors. Thus, in most cases, the computational BB, to varying degrees dependent on the detailed control paradigm reflects a "moderated" problem-solving paradigm. Thus, in the computational analog, a conceptual framework for communication and result-sharing permits a group of independent processes (specialized knowledge sources) under some degree of centralize control (the moderator) to cooperatively and/or competitively interact with the evolving problem solution state (held by the BB).

6.2 The Generalized Blackboard (BB) Problem Solving Paradigm

The requirements of the tacticle data fusion process were summarized in column 1 of Table 2;the second column suggests natural strategies for dealing with each of the fusion problem characteristics. In this section , the capabilities of a generalized blackboard (BB) - based paradigm are shown to be ideally matched to tactical fusion system requirements as suggestes by column 3 of Table 2.

TABLE - 2

Fusion system character	Strategy	BB Paradigm Implication
Complex, large scale problem ..	.. Task Decomposition ..	.. BB partitioning
Distributed Sensor & nodes ..	.. Spatially distributed process	.. Communication & control
Realtime requirements ..	.. Distributed processing	.. Communication & control
Multiple level of abstraction	.. Task decomposition..	.. Hierachical reasoning
Realtime requirements..	.. Efficient processing.	...Parallelization
Dynamic situation ...	.. Dynamic process model.	...Temporal reasoning
Uncertainty (data & decision)	Multiple hypothesis management	Uncertainty management

The BB paradigm represents a special form of Object- Orientet (OO) reasoning. In general, objects possess arbitrary characteristics and degree of autonomy. Objects can either directly or indirectly interact. Indirect interaction occurs if individual objects both monitor and modify the overall problem solution state Modifications include posting locally generated solutions, partia solutions, hypotheses and unsolved problems. From this viewpoint the conventional BB is merely a specialized object that maintain the problem solution state.

The functional-level fusion model supports arbitrary implementation architectures in software and hardware consisting, for example, of potentially large numbers of homogeneous and/or non-homogeneous spatially-distributed processing nodes. The individual processes can represent a mix of technical disciplines and problem solving approaches. If each separate process is treated as an object, an OO problem solving environment supports the required process encapsulation, BB interaction, and overall process control.

Sophisticated control may be required due to the potentially complex mix of dependent and independent process operating across multiple levels of abstraction. In an OO environment, distributed hierarchical control allows each object to maintain its own local control element. In addition, higher level of abstraction objects can task lower level objects. Thus, individual processes can operate synchronously, be triggered by other processes, or operate fully asynchronously responding autonomously to the state of the BB. The overall processing net supports both data- driven (local process originated) and goal-driven (higher level process-directed problem solving, as well as elements of both centralized and distributed control.

Thus, a generalized BB-based problem solving paradigm clearly supports the first four requirements of the tactical data fusion problem listed in table 2 . While an argument could be made than an OO problem solving model intrinsically support all seven problem characteristics, the last three items require further consideration. As will be argued next, data representation is a key aspect of 1) the efficiency of individual fusion processes and 2) the maintenance of competing BB hypotheses.

6.3 Problem solving efficiency

Because of the limited computational resources and typically time-critical requirements for fusion products in the tactical environment, highly efficient fusion processing required in most applications. Since even an intrinsically efficient algorithm can become I/O bound, efficient problem solving requires both efficient algorithms and efficient database access. Efficient algorithms imply efficient support for spatial, hierarchical and temporal reasoning. In terms of database acess efficiency, rapid location of appropriate database information requires support for efficient problem- dependent search dimensions.

In the "target detection-to-track assignment" problem, for example, all candidate track files must be evaluated before making a track assignment decision. Candidate tracks are those tracks than are "close" to the new sensors- derived report based on metrics such as spatial/temporal proximity, speed, velocity, cross section or radio frequency. Rather than evaluating all possible tracks database search dimensions should permit direct access to candidate track files. If "close" is defined by a Euclidean distancemeasure, for instance, efficient data acess demands a database representation for both static and dynamic data for preserves the Euclidean metric. Single-dimensional representaion of 2-D data such as vector, boundary and linked lists do not preserve the spatial distance metric, and thus do not support true 2-D indexing (i.e.,data that is "near" in 2-D space may not be "near" in the database regardless of the indexing scheme). On the other hand, a true 2-D spatial data organization (points are stored as points, lines as lines, and regions as the enclosed area) can preserve the Euclidean metric among 2-D spatial features, and leads itself to associative processing.

Thus, both algorithm efficiency and database access efficiency can be highly sensitive to data representation. In many cases, the use of "natural" data representations (e.g.,2-D spatial data stored as true 2-d representations, solid objects as true 3-d representations, hierarchies as hierarchies, tables as tables, rules as rules) offer inherently powerful and efficient database search dimensions and reduce the requirement for "artificial indexing. In addition to efficient search, natural data representations tends to preserve key relationships among the data (e.g., spatial distance metrics) that enhance algorithm efficiency.

Although the relational database is the current industrial standard, the relational model fundamentally supports only table based data. Regardless of the number of fields that are conjoined only linear search dimensions are supported [Antony,91]. Since efficient 2-D spatial reasoning requires support for true 2-d spatial search dimensions, and efficient hierarchical is enhance by true hierarchical representations, the relational data mode tends to be suboptimal for both of these reasoning classes.

The search for BB objects that are close both in time and space to a new target report is fundamental to the fusion process. If in the 2-D case, stationary and moving entities are maintained as

X_a	Y_a	t_a
X_c	Y_c	t_c
X_b	Y_b	t_b

(X_d, Y_d, t_d)

Fig. 12 Three-Tuple database sorted by time

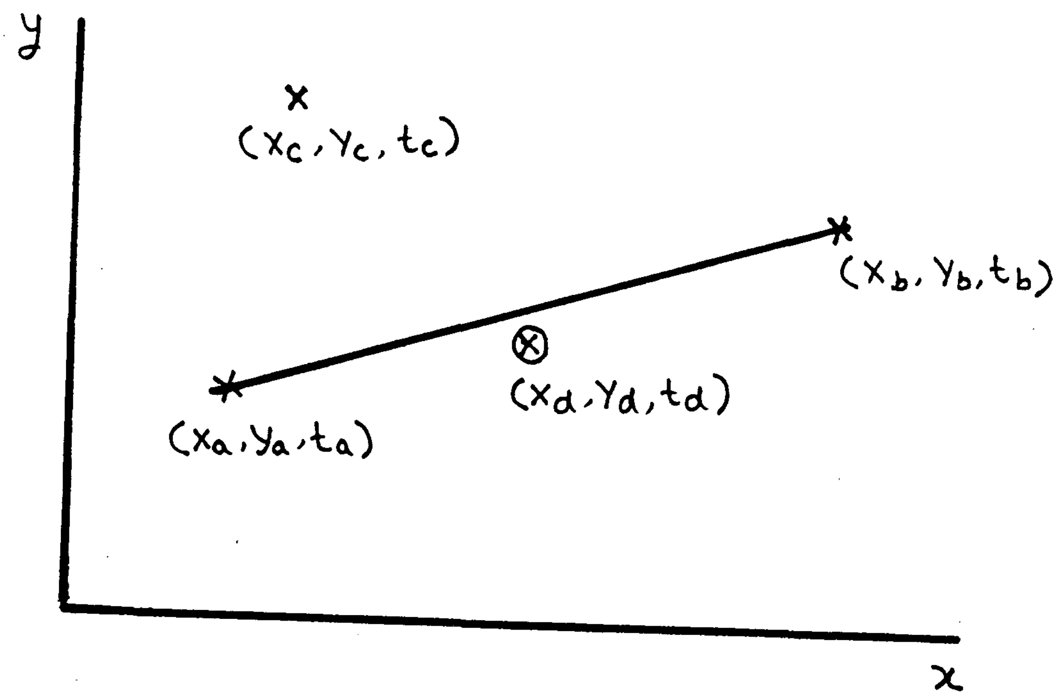


Fig. 13 Example demonstrating the advantage of a temopal-coded true 2-D spatial rep. over a temporally-indexed

discrete 3-tuples (i.e.,xi,yi,zi), indexing or sorting along an individual dimension is straightforward. A spatially-coded temporal history file is a typical example single dimension indexing (i.e. 3-tuples indexed by t).

Although a moving target is often observed only at discrete points in time, the actual trajectory of the entity is a continuous function in both space and time. Thus, the data fusion applications, indexing (or search) along the continuous dimension of space and time is ideally required. Consider the analysis of sensor reports using the discrete data represented in fig. 12 and by the diagram in fig.13 . If the database is indexed by time, for t_a=1, t_b=10, t_c=t_d=5, 1) query point **d** will appear close to point **c** since t_c=t_d and 2) point **d** will appear far from either **a** or **b**. In reality, query point **d** is far from point **c** and very close to trajectory represented by line **a-b**. Although the discrete database can be analysed to produce the desired product, a high computational overhead may result for inappropriately organizes real world size databases.

Although it does not independently index all three dimensions, a time-coded true 2-D representation (fig.12) of a target track effectively captures the true continuous character in all three dimensions and at the same time preserve spatial relationship between the track and the other database elements. Rather than an extensive computational approach, **d** can be determined to be spatially near line **a-b** by highly localised spatial search [Antony,90]. Next, for all spatially "near" candidate tracks, the time attribute can be efficiently tested by interpolation between

t_a and t_b. Search efficiency can be similarly enhanced for time varying point and region data. Rather than an exception to the argument in favor of natural search dimensions, the time dimension can often be subordinated to other natural search and problem solving dimensions (e.g., spatial dimensions). Thus, "time" become an attribute for data stored along its "primary" natural dimensions.

In general, data fusion requires efficient association between dynamic (i.e., vehicles, units, events) and static domain elements (i.e., buildings, geographic and cultural features). The efficienct of spatial, temporal and hierarchical reasoning is enhanced if both the dynamic (situation description) databases and static (long term factual and reasoning knowledge) are maintained in fully compatible representations within the same database. Thus, the overall efficiency of the data fusion process can be enhanced if the BB is maintained within the system database.

6.4 BLACKBOARD MANAGEMENT

Due to the dynamic, evolving nature of the tactical environment, time-late information, limited sensor-derive information, deliberate deception, ignorance of an adversary' intent and imperfect reasoning knowledge, uncertainty in perception is inevitable. The dynamic, sequential nature of the decision process leads inevitably to the maintainance of multiple hypothese, which, over time, can generate a combinatorial explosion of hypotheses. Thus, in order to support the tactical fusion process the BB must 1) hold the confidence in all existing BB hypothese (or states), 2) maintain multiple conflicting hypotheses, 3) support multiple views and 4) permit efficient retraction of belief and all associated backtracking operations.

The hypothesis explosion problem can be minimized by 1) powerful, context sensitive algorithms that establish only relevent hypotheses in the first place and 2) representations that support efficient pruning of low confidence hypotheses, as well as straightforward hypothesis retraction. Reasoning in context depends heavily on the association among entities and events and with respect to extensive natural and cultural domain feature databases. The most plausible hypotheses tend to be "close" along spatial,organizational or other natural problem solving dimensions. Hypothesis pruning occurs as additional information becomes available that reduces the overall situation uncertainty. Thus, a data representation that supports efficient, highly localizes spatial and semantic search facilitates both complex contextual reasoning and the development and maintenance of multiple hypotheses. Consider the basic track assignment problem depicted in Fig. 14. At time t_1 the detection represented by object is supports three hypotheses: 1) **A** is an extension of Track **T1**, 2) **A** is an extention of Track **T2** or 3) **A** represents the initiation of a new track, **T3**. As discussed previously, candidate tracks can by efficiently located if data represention preserves the appropriate distance metrics. At time t_2, detection **B** supports the extention of **T2** to point **B** and refutes hypothesis (**T2, A**). At time t_3, detection **C** supports four seperate hypotheses as depicted in Fig. 15.

Designation of new hypotheses by "attaching" detection objects (e.g.,**A,B,C**) to the established tracks (i.e., high confidence

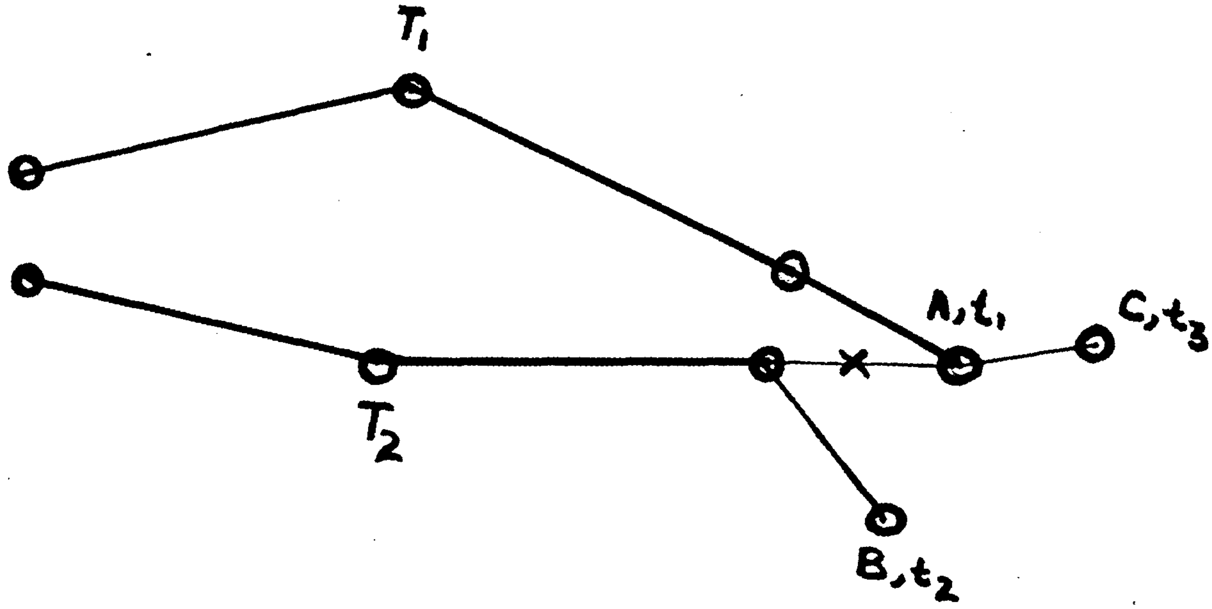


Fig. 14 Extensions to existing target tracks Supported and Refuted by directions A, B, C

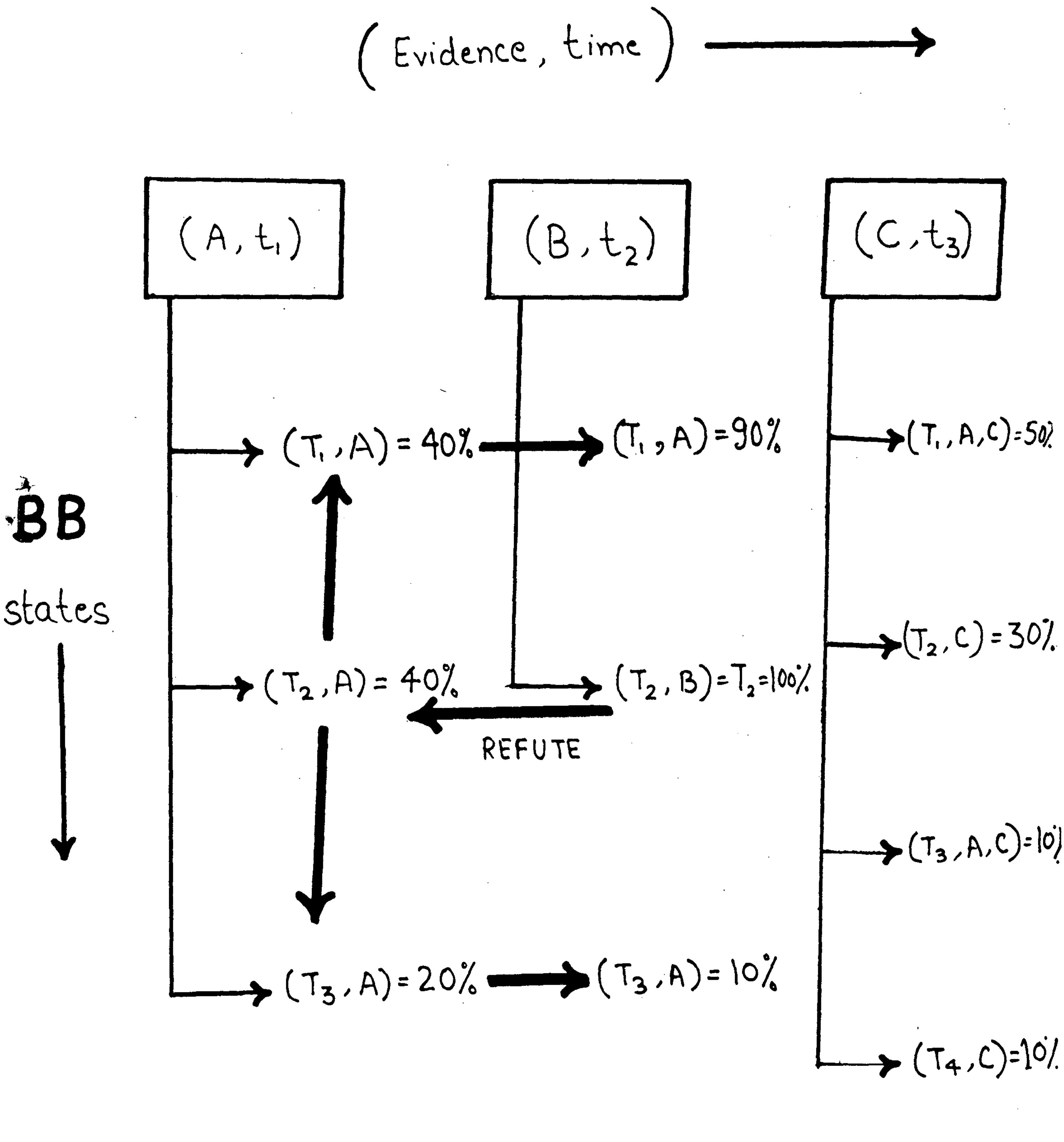


Fig. 15 Multiple hypothesis management for the target tracking example depicted in fig.

hypotheses) permits straighforward hypothesis retraction. Each time new evidence appears (either from a sensor report or from a change in the stste of the BB), all hypotheses that have depended on this hypothesis are reevaluated. Fig 15 depicts both the external, as well as internal BB operations. Sensor-derived evidence **A,B and C** directly support the inferences denoted by the thin lines, while hypothesis retraction and internal BB modification operations are denoted by thick lines. For instance, at time t_2, sensor data refutes hypothesis **(T2,A)**, which leads to the redistribution of the support for competing hypotheses **(T1,A)** and **(T3,A)**. Once adequate confidence in a hypothesis has been developed and no further hypothesis retraction is required, the supporting object extentions to the track name can be deleted. Thus, at time t_2, trace hypothesis **(T2,B)** would be renamed **T2** since evidence **B** supports only a single, high confidence hypothesis.

In summary, although the potential for a combinatorics explosion of hypotheses is an inherent characteristic of multiple hypothesis system, the impact can be minimized by 1) powerfull context-sensitive algorithms that establish only relevant hypotheses and 2) representations that support efficient pruning of low confidence hypotheses and straightforward hypothesis retraction. Data representations that supports efficient, highly localised spatial and semantic search facilitates both complete contextual reasoning and the development and maintenance of multiple hypotheses.

6.5 ARCHITECTURAL ISSUES

The key BB design aspects are associated with the partitioning and allocation steps in the classical system engineering methods and in the use of **moderate-grained** parallelization within the de-coupled functions in order to achieve balance between speed-up reliable behaviors, and goal convergence.

There are some other summary observations on architectural issues that form an analysis of the survey data :

- a typical system employs many type of knowledge
- Data consistency and control issues are very important these issues are magnified with parallelization since the BB system is then analogous to a shared, distributed data base system.

  -> Design fector : Maximum data de-coupling and information-hiding techniques are required; private and public data sets; intelligent BB data manager.

Choice of **hardware** architecture are also very important.

-> Design fector : In general, high processing power is catgorically desired. Shared-memory multiprocessors should be avoided.

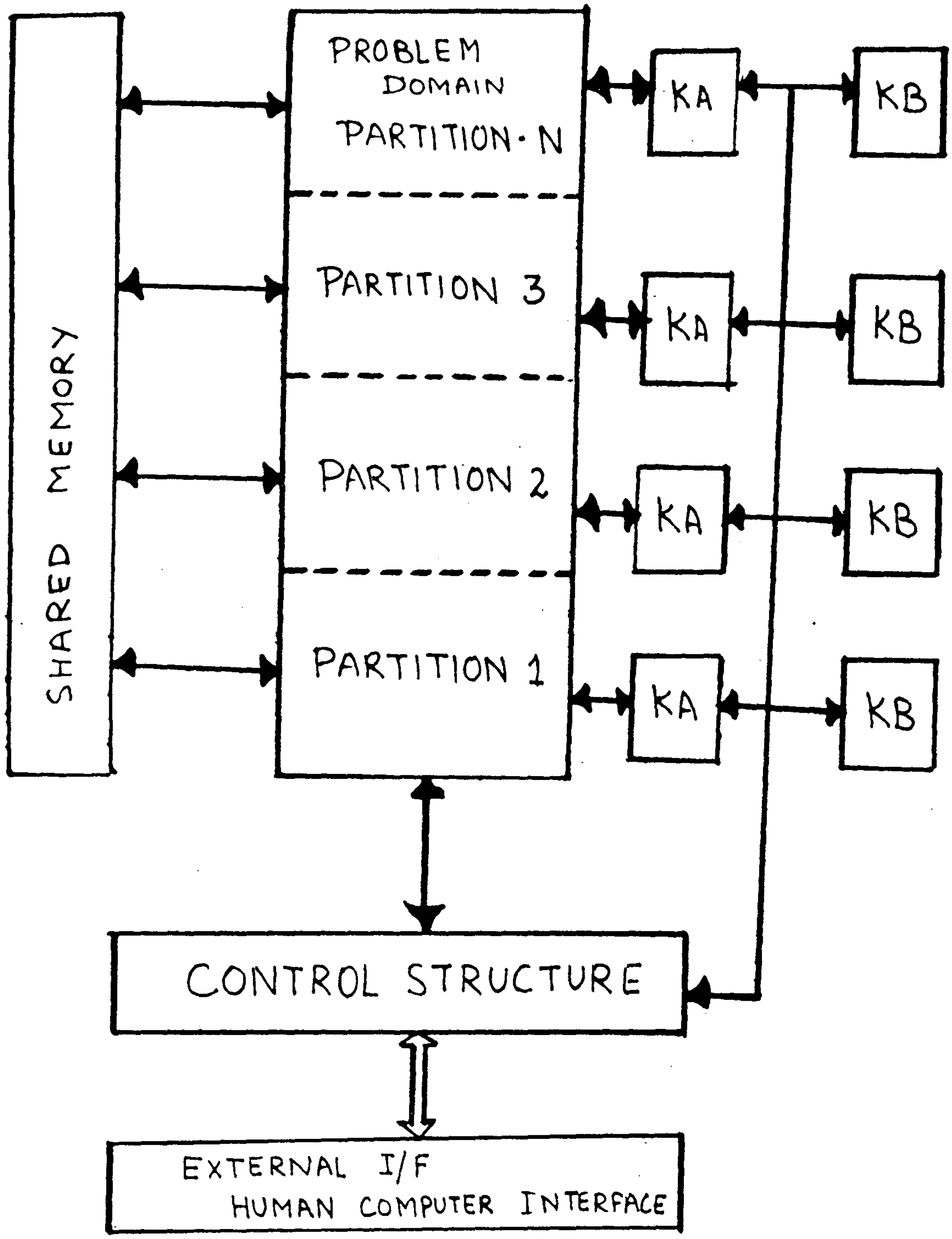
Other design features and system qualities would include :

-> Design factor : An object-oriented approach,
A truth maintenance system to accomodate non-monotonic reasoning processes, Some type of uncertainty propogation and management approach, and Intelligent use of demon-concepts for process control and data management.

6.6 Blackboard concepts for Data Fusion

The concept of blackboard architecture is illustrated in Fig 16. This figure shows a problem space or domain partitioned into subproblems. Each subproblems has a separate knowledge base comprising rules, frames, networks, scripts, etc. which define the knowledge which pertains to the subproblem. For each subproblem an associated knowledge base, (**KB**), a knowledge agent (**KA**) actively searches the shared memory and knowledge base to evolve solutions. Thus, while each portion of the problem is being addressed by logically separte **KA**, evolving solutions are shared via a share memory, accessable by all knowledge agents. A control structure seeks to balance the separate knowledge agents to achieve an overall solution to the problem and provide access to enternal data or human control.

The fundamental power of the blackboard architecture is twofold. First, by partitioning a problem into subproblems, a complex problem may be reduced to more readily soluable component problems. Second, blackboard logical architecture may by implemented by a variety of physical implimantations, i.e., all **KA**'s,**KB**'s, control structure,etc. can be implemented on a single computer (viz. single instruction, single memory (**SISM**), or on a computer having a single processors with multiple memories (single instruction, multiple data (**SIMD**)), or multiple processors and multiple memories (multiple instruction, multiple data (**MMD**)). Very complex problem may be solved in an efficient way, approaching real time performance. On a **SISM** architecture, a single inference engine operates, as required, on each knowledge base.



KA = KNOWLEDGE AGENT

KB = KNOWLEDGE BASE

Fig. 16 THE BLACKBOARD ARCHITECTURE CONCEPT

FURTHER STUDY

FURTHER STUDY

This section describe new areas of research in the field of netted systems. Two major items are considered, namely :

1) the netting of other types of sensors in addition to radars and,

2) advaced concept for the design on-line management and control of netted systems.

Development of modern air-defence system motivates the introduction of such advanced concepts in sensor netting. The novelty of these topics has meant that the theory is not yet well established.

By a multisensor system is meant a net formed of both active and passive sensors (such as monostatic and bistatic radars laser, radiometer and identification systems) covering the whol frequency spectrum together with acoustic sensors. The different data provided by sensors are synergistically fused into a high quality and reliable estimation of the surrounding scene. This estimate should be maintained at the best quality during the time even though the environment changes and technical failure and/or destruction of part of the system may occure.

The fundamental issues which arise is the development of a multisensor net are :

- the design of multisensor architectures.
- the algorithmic procedure for data fusion.
- the performance evaluation of a multisensor configuration.
- the on-line management/control of an implemented mulisensor net.

REFERENCES

REFERENCES

[Anderson, 58] An introduction to Multivariate Statistical Analysis, John Wiley & sons, New York, 1958.

[Antony,87] A framework for Automated Tactical Data Fusion, Proceedings of the 1987 Tri-service Data Fusion Symposium, Johns Hopkins University Applied Physics Laboratory, Laurel MD, 9-11 June 1987.

[Antony, 89] A generalized Data Fusion Process Model and Database Implications, Proceedings of the 1989 Tri-service Data Fusion Symposium, Johns Hopkins University Applied Physics Laboratory, Laurel, MD, 16-18 May 1987.

[Antony, 90] A hybrid Spatial/Object-Oriented DBMS to support Automated Spatial, Hierarchical and Temporal Reasoning, Chapter 3, vol. 1,Advances in Spatial Reasoning, ed. Su Shing Chan, Ablex Press, NJ, 1990.

[Antony, 91] Eight Canonical Forms of Fusion : A proposed Model of the Data fusion process, Proceddings of the 1989 Joint Service Data Fusion Symposium, Johns Hopkins University Applied Physics Laboratoty, Laurel, MD, 7-11 Oct., 1991

[Atkinson, 80] "A Bayesian Analysis of Surveillance Attribute Data," Proceedings of the 1980 IEEE confer. on Decision and Control, Albuquerque, NM, Dec. 10-12, 1980, pp. 205-247.

[Bar-shalom, 80] Tracking and Data Association, Academic Press, Orlando, FL, 1988.

[Bar-Shalom, 81] "On the Target-To-Target Correlation Problem," IEEE Trans. on Automatic Control. AC-26, April 1981, pp. 571-572. [Burke,73] "Stereographic Projection of Radar Data in a Netted Radar System." MITRE Technical Report 2580 Nov. 1973.

[Cantrell, 80] "Multiple Site Radar Tracking System." Proceedings of the IEEE 1980 International Radar Conference, Arlington, VA, April 28-30,1980, pp 348-354.

[Dempster, 68] "A Generalization of Bayesian Inference," Journal of the Royal Statistical Society, Series B, Vol. 30, 1968, pp. 205-247

[Farina, 81] "Introduction to Multiradar Tracking Systems," Rivista Technical Selenia, Vol. 8, No. 1, 1981, pp. 14-26.

[Fischer, 80] "Registration Errors in a Netted Air Surveillance System"MIT Lincoln laboratory Technical Note 1980-40, sep. 1980.

[Kanyuck, 70] "Correlation of Multiple Site Track Data", IEEE Trans. on Aerospace and Elect. Systems, AES-6, March 1970, pp. 180-187.

[Mardia, 79] Multivariate Analysis, Academic Press, New York, 1979

[Wax, 83] "Position location from Sensors with Position Uncertainty." IEEE Trans. Aerospace Electron. Syst., Vol. AES- 19, Sep 1983.