



## Distributed Interactive Simulation and Its Application in Multi Sensor Tracking

**Garima Vats\****Computer Science and Engineering  
Amity University, India***Avantika Singh***Computer Science and Engineering  
Amity University, India***Nidhi Chandra***Computer Science and Engineering  
Amity University, India*

**Abstract**— *The study focuses on extraction of data from multiple entities which are spread over a different network through a technique which is known as Multi Sensor Tracking. The various functions which are involved in the process of multi sensor tracking have been explained so that the accurate information is broadcasted to the further networks such that each entity comprises of similar units with reference to a particular standard. Distributed interactive simulation, a real-time simulation of a network's information exchange, is used to generate battle space test instances that are used in evaluating the proposed framework. VR Force and VR Link are the tools which have been used for creating a scenario and simulating entities over it. MAK is the platform which has been used for simulating a scenario over a particular framework.*

**Keywords**— *Distributed Interactive Simulation, MAK, Multi sensor Tracking, VR Forces, VR Link*

### I. INTRODUCTION

War involving space, air and on surface becomes three dimensional wars in nowadays and the future. The system should be build which can track the entities with the help of multiple medium ranging RADAR's and gives out the accurate information about the entity like its ID and name. Today, the world is facing diverse threats from the surface as well as from the air hence there is a need to built such a system which can identify the particular object as a enemy or not and can take action accordingly. Before building such a system a simulator need to be built which will be incorporating all the functionalities of the required system. The Multi Sensor Tracking is used which will be providing the accurate information about the entities which will be removing the redundancy of the number of entities present over a particular scenario and will give the exact count of entities. In nowadays, as the Defence weapon becomes more and more intelligent and complex, the Defence strategic and tactical simulations based on DIS [10], AI and VR come true. Analyzing distributed interactive simulation system based on MAK and referring to the Multi Sensor Tracking, the foundation and technology of CGF battlefield situation display is illustrated. [5]

### II. DISTRIBUTED INTERACTIVE SIMULATION

Distributed Interactive Simulation (DIS) is a government/industry initiative to define an infrastructure for linking simulations of various types at multiple locations to create realistic, complex, virtual worlds for the simulation of highly interactive activities. This infrastructure brings together systems built for separate purposes, technologies from different eras, products from various vendors, and platforms from various services and permits them to interoperate. DIS exercises are intended to support a mixture of virtual entities with computer-controlled behaviour (computer-generated forces), virtual entities with live operators (human-in-the-loop simulators), live entities (operational platforms and test and evaluation systems), and constructive entities (war games and other automated simulations). DIS draws heavily on experience derived from the Simulation Networking (SIMNET) program developed by the Advanced Research Projects Agency (ARPA), adopting many of SIMNET's basic concepts and heeding lessons learned. In order for DIS to take advantage of currently installed and future simulations developed by different organizations, a means had to be found for assuring interoperability between dissimilar simulations. These means were developed in the form of industry consensus standards. The open forum (including government, industry, and academia) chosen for developing these standards was a series of semi-annual Workshops on Standards for the Interoperability of Distributed Simulations that began in 1989. [4] IEEE Std 1278.1a-1998 defines the format and semantics of data messages, also known as protocol data units (PDUs), that are exchanged between simulation applications and simulation management. The PDUs provide information concerning simulated entity states, types of entity interactions that take place in a DIS exercise, data for management and control of a DIS exercise, simulated environment states, aggregation of entities, and the transfer of control of entities. DIS is intended to support the following functional requirements:

- a) Entity information/interaction
- b) Warfare
- c) Logistics
- d) Radio communications
- e) Distributed emission regeneration

- f) Simulation management
- g) Synthetic environment
- h) Entity management
- i) Minefield
- j) Live entity information/interaction
- k) Non-real time

### III. SIMULATION BASED ON MAK

#### A. Design of DIS based Simulation System

VR-Link provides the following categories of protocol-independent classes:

- Exercise Connection – A class that serves as the application's interface to the RTI or to the DIS network.
- Object management classes – Classes that maintain state information of local and remote objects, and handle the sending and receiving of state updates.
- Interaction classes – Classes that provide a protocol-independent API to either HLA interactions or DIS PDUs that represent events.

VR-Link also provides protocol-specific classes and features. However, in most cases, your code does not need to interact directly with them. Figure 1 shows how classes in a VR-Link application interact. Your application code can interact with classes at any level and with the RTI or DIS network directly. However, for most applications, you can rely on the protocol-independent classes, which are generally easier to use. VR-Link utilities, such as the coordinate conversion utilities are also available to application code and VR-Link classes. [7]

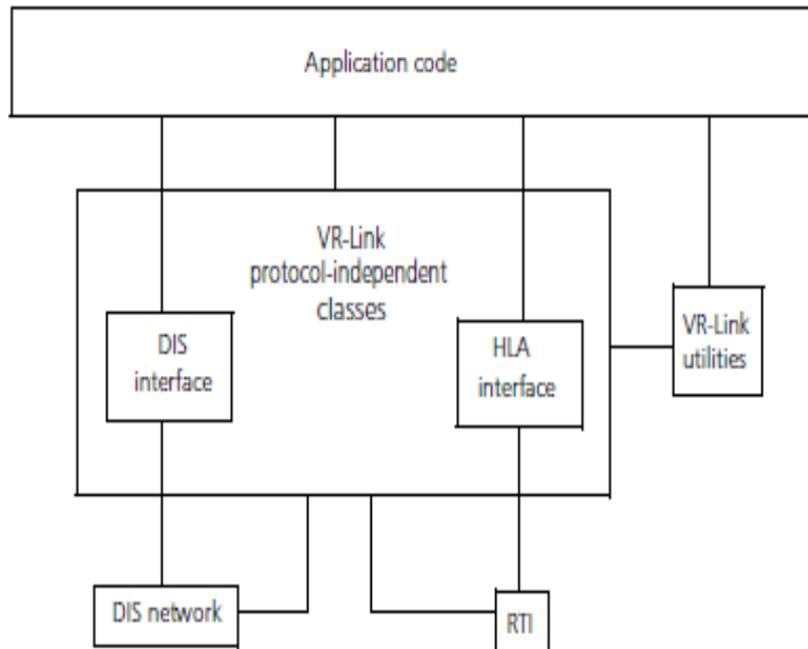


Fig. 1 Structure of Typical VR Link Application

#### B. The CGF Foundation Based on VR Forces

- 1) *Introduction to VR Forces:* VR-Forces is a development toolkit for CGF of MAK. It supplies an easy-to-use and concise GUI and rich object oriented API [2]. The user can easily make scenario editing and operation simulation by the entity and object provided by the software. Also, the user can make the user-defined design to meet the practical requirement by the API, such as program Recompilation. Moreover, the operation region models, weapon entity models and operation rules of our Air Force can be imported into the software so as to meet the practical requirement of aviation equipment combat simulation [6].
- 2) *Structure of VR Forces:* VR-Forces can realize two functions which are application and development of CGF [3]. So its structure includes software structure and API structure.
  - a) *Software structure*  
VR-Forces include to executable program, they are vrfSim and vrfGui. vrfGui is Graphical User Interface which creates entities, displays scenario and controls the simulation. vrfSim is a simulation engine which could respond the operation at vrfGui and send data to vrfGui. At same time, other application can control the VR-Forces. [1]
  - b) *API structure*  
In order to meet requirement, we must use the VR-Forces's API to execute the second development. VR-Forces provide four API functions, which are Simulation API VrfSim, Remote Control API, Terrain API and Graphical User Interface (GUI) API VrfGui. [2]

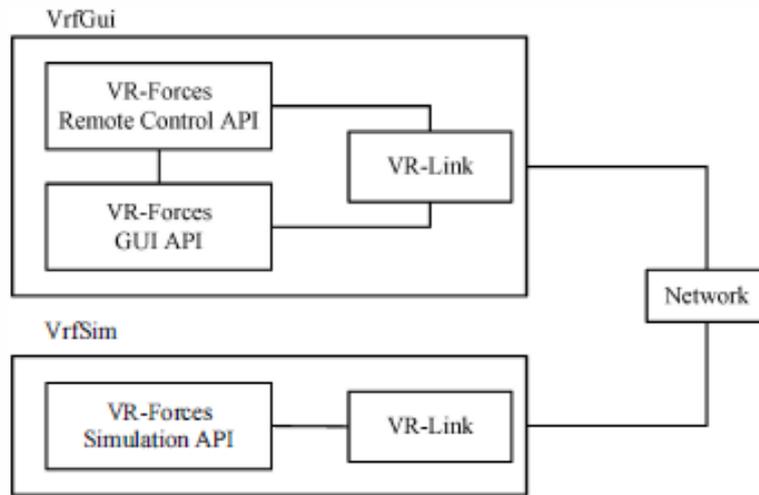


Fig. 2 VR Forces API Implementation

The Simulation API provides a wide variety of classes and functions needed to implement a CGF simulation engine. The API gives you access to libraries of code for the elements required to execute a simulation. The Graphical User Interface (GUI) API provides the code libraries that implement the VR-Forces GUI. You can use this API to customize or extend vrfGui, or to embed its functionality into an instructor/operator station or other application. The GUI API uses Qt - a cross-platform GUI development toolkit from TrollTech. The Terrain API is a set of classes that enable applications to read, write, and query terrain databases. The VR-Forces Remote Control API is a set of classes that allow an application to control one or more remote VR-Forces applications. The VR-Forces GUI (the vrfGui executable) uses the VR-Forces Remote Control API to control the vrfSim application or other VR-Forces applications. The Remote Control API can also be used in custom VR-Forces front-ends, simulation managers, or instructor/operator stations. [3]

### C. Design of Component

Entities in VR-Forces are constructed based on the component idea [8]. Component is an integrated technology on application level, fundamental of which is that practical application is divided into every independent unit to convert process of software development to the constructing process similar to building blocks, and software is integrated by packaging different component units. And thus, expanding for entities is actually a recomposing process of existing components and rebuilding of new components. They communicate with each other through ports [4]. The entity structure is showed in Fig.3.

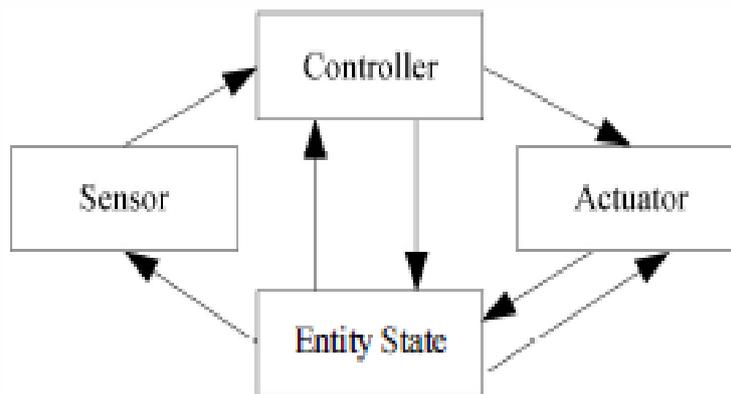


Fig. 3 The Entity Structure

Controller component is the command and control system which will be detecting the entity. Battlefield information from sensor component will be filtered and classified using the state machine mechanism and here the technique of multi sensor tracking is used. And then, referencing object parameter database, decision information will be ultimately made, which may be used to control movement, detection and fighting. Generally speaking, controller component consists of sending the orders to weapon control system which after receiving the particular transmission order will be sending the acknowledgement orders to the command control system. Action function of entity is completed in actuator component, in which physical model of entity, entity state change, and interaction with other objects, are provided. Actuator component mainly consist of movement component, communication component, detector using component, and attacking weapon using component.

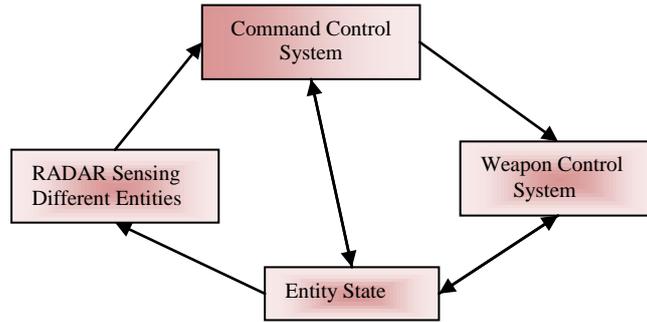


Fig. 4 Flowchart of System

It is explained that, proprietary, total logical relationship, port data, and interaction time of components need to be considered all together before designing simulation case in order to ensure sequential developing and normal running of components [9].

#### IV. MULTI SENSOR TRACKING

The purpose of MST is to fuse the tracks sent by individual radar tracker for a system using multiple Network Systems. MST Server receives the track data from individual system trackers periodically. Two or more systems may report targets in common area of coverage. To extract the tracks representing the same target, tracks have to be correlated and fused. Before this all the tracks should be brought under common time frame. Multi Sensor Tracker sends the fused track data to central track file. The MST processing combines the positional data from multiple System to establish a database of target tracks and un-correlated raw data. [11]

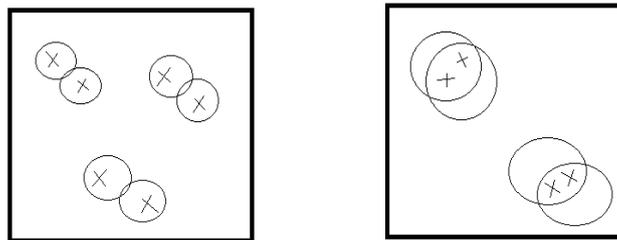


Fig. 4 Entities Being Tracked By RADAR

The MST processing combines the positional data from multiple system to establish a database of target tracks and un-correlated raw data. The processing may be partitioned into three functions: -

- Data alignment
- Data association
- Estimation for fusion
- 

##### A) Data Alignment

Data alignment functions transform data received from multiple sensors into a common spatial and temporal reference frame. Specific alignment functions include the following:

- Co-ordinate transformations
- Time transformations
- Unit conversions.

Data alignment involves following steps: -

- Spatial alignment
- Temporal alignment

##### 1) Spatial Alignment

The spatial alignment refers to transforming data representation from different co-ordinate systems & different units in to one system. The temporal alignment refers to transforming data captured at different points of time to a common point in time.

The data reported from sensors will be in following co-ordinate system.

- Cartesian co-ordinate system (x, y, z).
- Polar Co-ordinate system (r,  $\theta$ ).
- In the form of Latitude & longitude.
- Cylindrical co-ordinate system (r,  $\theta$ , z)

The purpose of data alignment is to convert the data from different co-ordinate systems into a common reference co-ordinate system. The following cases arise: -

- Polar to Cartesian co-ordinate conversion

- Cylindrical to Cartesian co-ordinate conversion
- Lat/Long to Cartesian co-ordinate conversion

2) Temporal Alignment

Temporal alignment transforms data received from multiple sensors at different times into a common time frame. Spatially aligned positional data of different sensors at different times is extrapolated.

**Expected Inputs:**

- Speed of the entity (speed)
- Course of the entity (course)
- X & Y position of the entity as reported by the sensor (X,Y) at time t.
- Current time (T).

**Output:**

- Extrapolated (time aligned) X & Y position of the entity

*B) Data Association*

A fundamental problem in multi-radar tracking is data association. In particular, association is the intermediate-processing step between data alignment and positional estimation. If there is n number of observations from m different types of sensors, then determining which observation belongs to the other observation or represents which of the existing entity is the main task of data association. The data association process compares the new observations against known entities (represented by a state vector) and/or previous observations. The new observation may represent the observed position of an entity, parameters related to position (azimuth, elevation, etc), identity information, or parameters that can be related to identity. The association process systematically compares new observation against known entities or to other observations, and determines whether or not these observation-entity or observation-observation pairs are related. To determine the association between observation pairs, or observation-track pairs, six steps are followed:

- Retrieval of candidate entities from a database.
- Updating of candidate entities to the observation time
- Computation of predicted observations for each candidate track
- Gating to eliminate unlikely observation-observation or observation-track pairs
- Computation of an association matrix
- Implementation of assignment logic.

The following are the steps involved in data association:-

- 1) *Retrieval of Candidate Entities from Database:* This step involves retrieving candidate entities from a database containing either previous observations, or previously determined state vectors, which represent the current estimate of an entity's state (estimate of an entity's position, velocity, or identity). Entities are retrieved using a Boolean query. Formulation of the database query depends on the incoming observational data, and the internal representation of the stored entity data

Retrieval of candidate entities by a query may be limited to the following factors:

- Specified geographical boundaries
- Parametric boundaries
- Entities having a specified identity

- 2) *Updation of Candidates Entities to Observation Time:* In case of dynamic situations, the state of entity changes in time. Therefore, for dynamic objects the second step required for association is to update candidate entities' state vector to the observation time. Let an entity has some value  $x_i$  at time  $t_i$ , and let this entity is a dynamic entity, then its predicted value at the observation time (say  $t_j$ ) must be determined. Equations of motion must be solved to determine this predicted value. [11]
- 3) *Computation of Predicted Observations for each Candidate Track:* In some association process, where sensors are not able to observe directly an entity's state, this step is required to predict an observation, resulting from an entity.
- 4) *Gating:* The purpose of gating is to reduce, the number of possible combinations of observation-observation or observation-track pairs that must be further considered for association. Gating is a technique that starts to solve the problem of associating observations to predicted track positions by eliminating unlikely observations to track pairings. This procedure is done as early as possible in order to reduce the computational load in the remaining steps of data association. A gate is formed around the predicted track position.
  - **RECTANGULAR GATES-** Simplest Gating Technique  
Calculations performed are not so hard  
Opportunity to choose various sizes of gates in the different measurement dimension  
Less computational resources are required
  - **ELLIPSOIDAL GATES -** Gate size decreases if the uncertainty of the predicted track position increases.  
Required Gating volume decreases, thus decreasing the probability of an extraneous observation falling within the gate.

- a) *Association Metrics*: An association metric is defined to quantify the similarities between observation pairs or observation-track pairs. This metric provides a means to quantify whether observations are similar or dissimilar. There are three different types of association metrics (association measures) which are widely used. These are:
- Distance measures
  - Probabilistic measures
  - Correlation coefficients
- b) *Assignment*: Gating only begins to solve the problem of associating observations with tracks. Additional logic is required when the observation falls within the gates of multiple target tracks or when multiple observations fall within the gates of a target track.

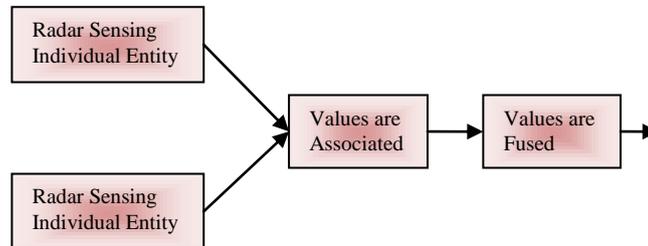


Fig. 5 Flowchart Of Multisensor Tracking

### C) Estimation for Fusion

The estimation problem refers to a process in which multiple observations of positional data are combined to determine an estimate of the target/entity position, velocity etc.

- Mean Squared Error method: This method aim at attaining zero mean error for the parameter under consideration.
- Weighted Average Method: Based on weight of inputs data or sensor behavior the fused parameters are computed.

## V. CONCLUSION AND PROSPECT

Simulation of entities over a scenario is a difficult simulation style which is only the way to realize the simulation based on MAK platform. Multi sensor Tracking technique is a ongoing research area which has been used here and its various functionalities have been explored. It is a very advantageous technique which provides us with the accurate information about the entities. And these research fruits will ultimately enrich defence operation simulation, optimize operational plan and increase operational effectiveness. VR Force here proves to be an efficient tool for simulating entities and gives out the true results of our findings.

## REFERENCES

- [1] MAK Technologies Inc, MAK VR Forces User's Guide
- [2] MAK Technologies Inc, MAK VR Forces Getting Started Guide
- [3] MAK Technologies Inc, MAK VR Forces Scenario Management Guide
- [4] IEEE Standard for Distributed Interactive Simulation-Application Protocols, IEEE Std 1278.1a-1998
- [5] Li Yaoyu, Wang Wenguang, Hou Hongtao and Wang Weiping, "Research of the Anti Submarine Tactical Visualization Simulation System Based on Vega Prime", System Simulation Technology & Application, vol. 30, p. 603.
- [6] MAK Technologies Inc, MAK VR Forces Developers Guide
- [7] MAK Technologies Inc, MAK VR Link Developers Guide, 2008, pp. 35-39
- [8] Wang Bo, Pan Changpeng, Zhang Yuan, et ai, VR-Forces Development, National Defense Industry Press, Beijing, 2011, pp. 98-100.
- [9] Gao Huameng, Li Zhi and Zhao Xinguo, "The User-defined and Function-strengthened for CGF of VR-Forces", Computer Simulation, vol 24(06), pp. 212- 215
- [10] G.J. Jense, N.H.L. Kuijpers, and A.C.M. Dumay, DIS AND HLA: Connecting People, Simulations and Simulators in the military, Space And Civil Domains
- [11] C. M. Rogers QinetiQ Ltd, Methods Of Multi sensor Tracking