Project Report On

REAL TIME VISUALISATION OF AERIAL OBJECTS

By:

JAGADISH DIGAL

ROLL NO: 119CS0525

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

(Submitted to Central Data Processing Unit, Defence Research and Development Organisation, Chandipur, Odisha, in completion of Summer Vocational Training from 9th June 2022 to 8th July 2022)

CERTIFICATE

This is to certify that this project report entitled **REAL TIME VISUALISATION OF AERIAL OBJECTS** submitted to Real Time Data Processing Unit, DRDO, Chandipur, is a bonafide record of work done by **JAGADISH DIGAL**, Roll No: 119CS0525, National Institute of Technology, Rourkela under my supervision from 9th May 2022 to 8th July 2022.

Dr. Sourav Kaity,

Scientist- 'F'

Group Head (RTDP)

Integrated Test Range, DRDO Chandipur

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere gratitude to all of them.

I deeply appreciate the invaluable guidance provided by Mr. H. K. Ratha, Scientist- 'G', Director and Mr. Rakesh Barua, Scientist- 'F', Group Director (CDP).

I express heartfelt gratitude to my guide, Dr. Sourav Kaity, Scientist- 'F', Group Head (RTDP), ITR, whose supervision, encouragement and meticulous observation of every detail of the work and the report has helped me reach the objective of the project. It wouldn't be exaggerating if I say sir was more concerned about the project than many students themselves.

My thanks and appreciations also go to my colleague in developing the project and people who have willing fully helped me out with their abilities.

JAGADISH DIGAL

Roll No: 119CS0525

Department of Computer Science and Engineering

National Institute of Technology, Rourkela

TABLE OF CONTENTS

1. About Organization

1.1 Defence Research and Development Organisation

1.1.1. Vision

1.1.2. Mission

1.2 Integrated Test Range

1.2.1 Vision

1.2.2 Mission

1.3 Central Data Processing Group

1.3.1 Role and Responsibility of CDP Group

2. Introduction

2.1 Problem Statement

2.2 Alternative Work

2.3 Overview of Project

2.3.1 Electro-Optical Tracking System

2.3.2 Radar Systems

2.3.3 Telemetry Systems

2.3.4 ENU Coordinate System

2.3.5 ECEF Coordinate System

2.3.6 Cartesian and Spherical Coordinates

3. Requirements:

3.1 Functional Requirements

3.2 Non-Functional Requirements

4. Architectural Design

4.1 Class Diagram

4.2 Sequence Diagram

5. Interface Design

5.1 Main View

5.2 Combo View

5.3 Text View

5.4 Menu Bar

5.5 Settings

6. Coding

6.1 Reason for choosing JAVA for Coding

6.2 Reason for choosing NetBeans for Coding

6.3 JAVA classes used

7. Testing

7.1 Receiving Packet

7.2 Plotting and Rendering All Views

7.3 Screen Snapshot and Graph Snapshot

7.4 Settings File and Sensor File

7.5 Nominal File

8. Conclusion and Future Work

9. References

1. ABOUT ORGANISATION

1.1 Defence Research and Development Organisation

DRDO is the R&D wing of Ministry of Defence, Govt of India, with a vision to empower India with cutting-edge defence technologies and a mission to achieve self-reliance in critical defence technologies and systems, while equipping our armed forces with state-of-the-art weapon systems and equipment in accordance with requirements laid down by the three Services.

1.1.1 Vision

Empowering the nation with state-of-the-art indigenous Defence and security technologies and systems.

1.1.2 Mission

- Design & develop state-of-the-art sensors, weapon systems, platforms & allied equipment in defence & Security domains of land, air, sea, space & cyber.
- Facilitate production and induction of Systems & Technologies developed through Department's R&D ecosystem.
- Provide technological solutions to the Services to enhance combat effectiveness.
- Nurture & Strengthen Defence R&D capability in Indian industry, Science & Technology(S&T) institutions & academia through collaboration.
- Development of infra-structure and test & evaluation facilities; design certification; skill development and strengthen human resources.

1.2 Integrated Test Range

Integrated Test Range (ITR), a well-equipped Test and Evaluation (T&E) centre of DRDO, is set up to provide safe and reliable launch facilities for performance evaluation of rockets, missiles and air-borne weapon system. Equipped with high performance Range Instrumentation Systems such as Electro Optical Tracking System (EOTS), Radar System, Telemetry System and Central Computer System to provide necessary data (precise location and health parameters) of airborne targets through entire course of flight.

1.2.1 Vision

To be a trusted integrated test facility delivering reliable and quality Services to all customers.

1.2.2 Mission

- To achieve excellence in service quality and reliability.
- To attain global best practices and become a world class test range.
- To be a technology driven and efficient organization.
- To encourage ideas, talents and value systems.
- To work with Vigour, Dedication and Innovation with Total Customer Satisfaction as the ultimate goal.
- To promote work culture that fosters Individual Growth, Team Spirit and Creativity to overcome challenges and attain goals.
- To contribute towards community development and Nation Building.

1.3 Central Data Processing Unit

1.3.1 Role and Responsibility of Central Data Processing Unit

- Real time acquisition and processing of data received from Various high performance Range Instrumentation Systems such as Electro Optical Tracking System (EOTS), Radar System, Telemetry System and Central Computer System deployed at various places.
- Computation of desired results for different shareholders including control system, safety unit, central time unit.
- Provision of real time data visualization and decision-making during missions.
- Range instrumentation simulator for server and RT display validation.
- Simulation for operator familiarization and safety monitoring.
- Real Time Software development for a variety of systems to meet mission requirements.
- Bias Estimation and Validation of various tracking instruments through helicopter and aircraft.
- Post mission analysis and handing over the data and report to project.

2. INTRODUCTION

2.1 Problem Statement:

The project titled REAL TIMEVISUALISATION OF AERIAL OBJECTS aims to develop a convenient and efficient interface for the various clients of ITR, DRDO and other department concern, in order to provide all the required types of display. The primary objective is to make this application user friendly, and enable user handle it with minimal knowledge technical understanding of the software and processes involved. Functionality for handling and plotting up to 100 sensors and 5 Aerial Vehicle data received simultaneously from two servers, either in unicast or multicast mode, maintaining minimal latency. Flexibility for configuring and saving the settings to achieve the best display for each mission must be provided while making use of the latest graph plotting technologies currently existing to provide certain additional facilities as per the client's needs.

2.2 Alternative Work:

Existing work for a slightly different version of the problem statement was already done using Qt GUI Application Development Framework. Although all functionalities were provided as per the client's needs, however, still it only supported a mission scenario of maximum 2 Aerial Vehicles. Certain tweaks were required in the configuration for each mission, to provide different colour scheme and appearance as per the client's demands. As a result, a new improved application was targeted, covering all the scope of improvements of the previous application in order to provide a satisfactory result to the users.

2.3 Overview of the Project:

ITR is equipped with a number of tracking instruments to cover the total flight path of test vehicles. These include the: Electro-Optical Tracking System (mobile and fixed), S- band Tracking Radar (mobile), C-band Tracking Radar (fixed) and Telemetry (fixed and mobile). These instruments track the flight vehicle and send data to the Data Processing Server (DPS) where the data is processed and meaningful information is extracted. Also, using this information, the necessary Computer Designated Mode (CDM) bearings are sent to tracking instruments for locating the flight vehicle if they lose track of it. A reliable communication network is essential to connect all the mission related stations (where tracking instruments are located) for conducting missions.

The purpose of this project is to develop a user interactive application that receives data from two chains of server using a UDP socket so that loss of a few packets of data does not affect the functioning of the whole system. The application can establish a connection with the server in multicast mode or unicast mode as per the client's input with three modes of operation provided for displaying the transmitted data in the form of a graph:

Chain 1: Data received from Server 1 displayed only

Chain 2: Data received from Server 2 displayed only

Chain Auto: Primary selection is Server 1, but if continuous 5 packets of Server 1 missed and Server 2 is active, shift to displaying data received from Server 2.

The software handles data for 100 sensors and 5 Aerial Vehicle and plots the received data in the form of a graph/chart as shown in Figure 1, without much latency. At the same time flexibility for altering and saving the changes in configuration for each mission requirements are also provided. The appearance of textual and graphical for each mission can be easy manipulated with the help of user interaction with the application. All other necessary functionalities like screenshot, screen recording, scenario settings, etc. have been provided.

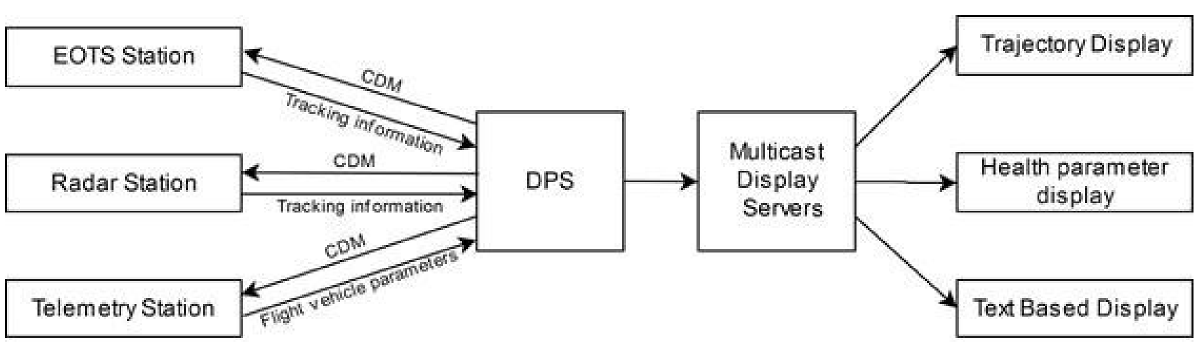


Fig 1. Block diagram of existing system

2.3.1 Electro-Optical Tracking System:

EOTS utilizes a combination of electronics and optics to generate, detect, and measure radiation from airborne vehicles in the optical spectrum. The portion of the electromagnetic spectrum used by EOTS includes infrared radiation, visible light and ultra-violet radiation. The operational requirements for EOTS are target detection, target auto track and data collection. Various tracking algorithms like Edge Tracking, Centroid Tracking and Correlation Tracking can be used to track airborne vehicles.

EOTS can operate in Manual, Designate or Auto position system. In the Manual system, the operator positions the gimbal through a positional control following the target's motion. In a Designate system, the target's trajectory is determined from a prior knowledge of the target trajectory (nominal trajectory). This data is used to drive the gimbal's position encoders to known positions. In an Auto position system, initial acquisition is accomplished by operator identification and selection of the target. The operator then initiates the auto positioning or auto track mode and the tracking processor positions the gimbal based on the calculated target position.

The tracking data for EOTS sent to the CDP Unit includes various parameters like GPS time, Range, Azimuth and Elevation.

2.3.2 Radar Systems:

Radar is an object-detection system that uses radio waves to determine the range, angle, or velocity of objects. A radar system consists of a transmitter producing electromagnetic waves in the radio or microwave domain, a transmitting antenna, a receiving antenna (often the same antenna is used for transmitting and receiving) and a receiver and processor to determine properties of the object(s). Radio waves from the transmitter reflect or scatter from the object and return to the receiver, giving information about the object's location and speed. The types of tracking radar are STT Radar (Single Target Tracking Radar), ADT Radar (Automatic Detection and Tracking Radar), TWS Radar (Track While Scan Radar), Angle Tracking Radar, Phased Array Tracking Radar and Monopulse Tracking Radar. The tracking data for Radar sent to CDP includes various parameters like GPS time, Range, Azimuth and Elevation.

2.3.3 Telemetry Systems:

Telemetry is an automated communications process by which measurements and other data are collected at remote or inaccessible points and transmitted to receiving equipment for monitoring. The word is derived from Greek roots: tele meaning remote, and metron meaning measure. Telemetry is used in testing of airborne vehicles since it allows the automatic monitoring, alerting, and record-keeping necessary for efficient and safe operation. Telemetry is vital in the development of Aerial Vehicle, satellites and aircraft because the system might be destroyed during or after the test. Engineers need critical system parameters to analyse (and improve) the performance of the system. In the absence of telemetry, this data would often be unavailable.

The tracking data for Telemetry sent to CDP unit includes various parameters like GPS time, down range, cross range, altitude, quaternion angles, roll, pitch, yaw and chamber pressures.

2.3.4 ENU Coordinate System:

The East-North-Up (ENU) coordinate system is defined with respect to a location on the earth's surface, i.e., it is a local coordinate system (Fig 2). In this system, the origin is arbitrarily fixed to a point on the earth's surface, the +X axis points to the east, the +Y axis points to the north and the +Z axis points to the vertically upward direction. The Z axis passes through the centre of the earth when using a spherical earth simplification, or is along the ellipsoid normal when using a geodetic ellipsoidal model of the earth.

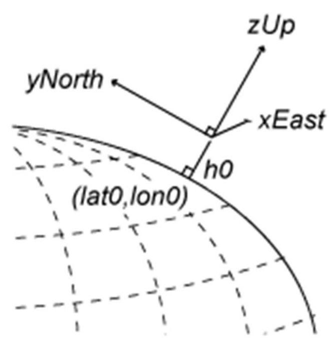


Fig 2. ENU coordinate systems

2.3.5 ECEF Coordinate System:

In the Earth-Cantered, Earth-Fixed (ECEF) coordinate system, the origin is at the centre of the earth, the X-axis intersects the sphere of the earth at 0° latitude (the equator) and 0° longitude (prime meridian in Greenwich), the Z-axis extends through true north and the Y-axis is orthogonal to both the X and Z axes following the right-hand rule (Fig 3). The ECEF axes rotate with the earth, and therefore coordinates of a point fixed on the surface of the earth do not change.

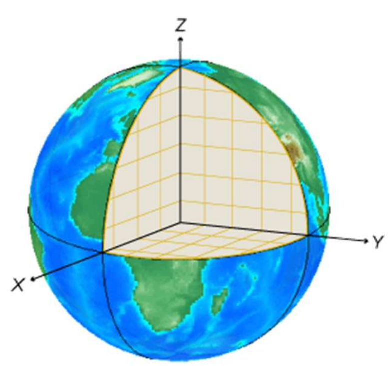


Fig 3. ECEF coordinate systems

2.3.6 Cartesian and Spherical Coordinates:

The Cartesian coordinate system specifies each point uniquely in space by three numerical coordinates, which are the signed distances to the point from three fixed perpendicular directed lines (Fig 4). Each reference line is called a coordinate axis or just axis of the system, and the point where they meet is its origin, usually at ordered pair (0, 0, 0).

The spherical coordinate system is a coordinate system where the position of a point is specified by three numbers: the radial distance of that point from a fixed origin, its polar angle measured from a fixed zenith direction, and the azimuth angle of its orthogonal projection on a reference plane that passes through the origin and is orthogonal to the zenith, measured from a fixed reference direction on that plane.

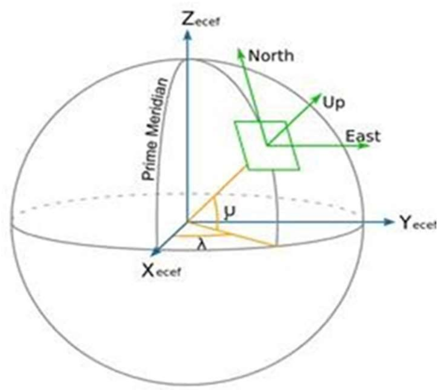


Fig 4. Cartesian and spherical coordinates

2.3.7 Azimuth-Elevation-Range Coordinates

An azimuth-elevation-range (AER) system uses the spherical coordinates (az, elev, range) to represent position relative to a local origin. The local origin is described by the geodetic coordinates (lat0, lon0, h0). Azimuth, elevation, and slant range are dependent on a local Cartesian system, for example, an ENU system.

- az, the azimuth, is the clockwise angle in the xEast-yNorth plane from the positive yNorth-axis to the projection of the object into the plane.
- elev, the elevation, is the angle from the xEast-yNorth plane to the object.
- range, the slant range, is the Euclidean distance between the object and the local origin.

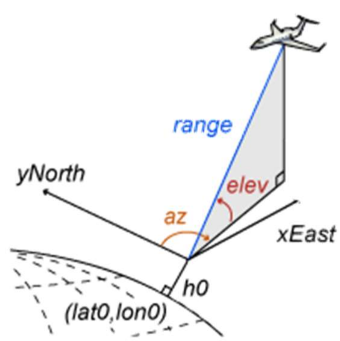


Fig 5. Illustration of AER coordinate system

3. REQUIREMENTS

3.1 Functional Requirements:

1. The application should continuously receive data from the server. The server sends the data through a UDP packet every 100ms.
2. Two modes of operation:
    a. Unicast
    b. Multicast
3. The application receives data from two servers. The second server is provided as a failsafe in case Server 1 fails.
4. Server 1 and Server 2 are manually synchronized. On the basis of the Chain Selection by the user, there are three modes:
    a. Chain 1: Forced Server 1 mode
    b. Chain 2: Forced Server 2 mode
    c. Chain Auto: Switches to Server 2 if Server 1 fails for 5 consecutive data receipt.
5. Graphical User Interface for the clients which provides 'Views' to the clients based on their requirements.
6. Plotting the data sent and displaying in the form of a graph. The graph must meet the following minimal functionalities:
    a. Auto scaling for the axis based on the received data.
    b. Manually setting the range for each axis.
    c. Zoom in and zoom out option.
    d. Control to hide or show grid lines, title for each plot, ticks and their labels along each axis.
    e. Changing the color of the plots based on user input.
7. Ability to switch between different graphs or views including VIP view, Mission Director View, Group Head View.
8. Displaying the real time data received every one second, thus enhancing user readability.
9. Update and keep track of the number of sensors and Aerial Vehicle participating in the mission.

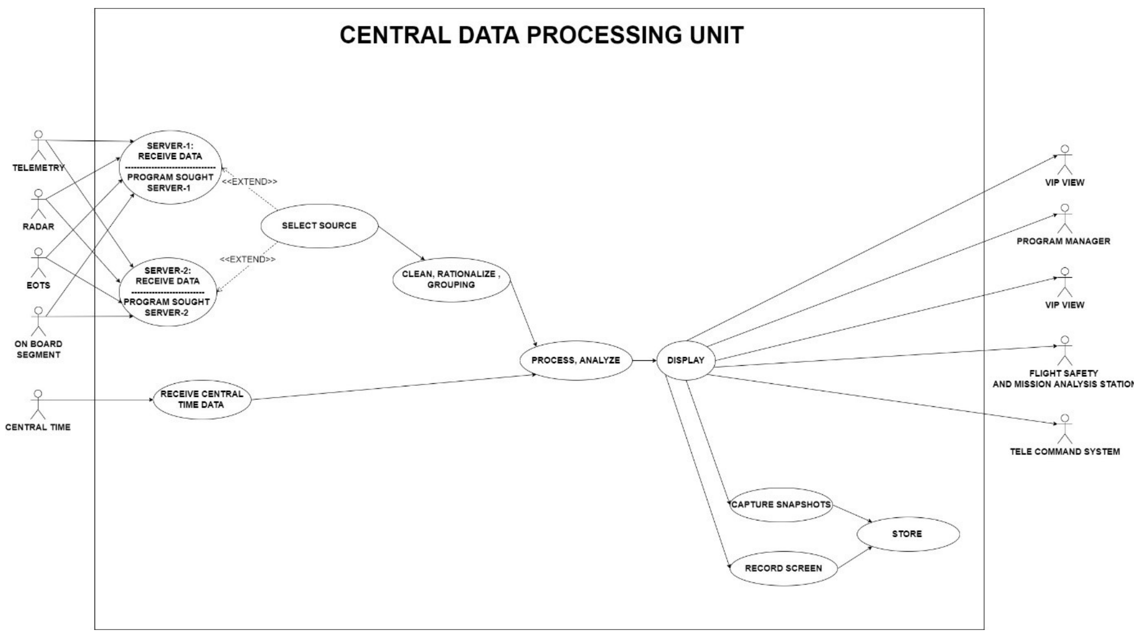
10. Selecting/ deselecting the active sensors to be displayed on the graph.
11. A text view for displaying the data without any delay.
12. Keeping track of the activity of the sensors with respect to time.
13. Displaying information on mouse hover over the graph.
14. Drawing annotations like circles, lines, points and text based on user input, to mark certain important locations on the plot.
15. Screenshot at any instant of time:
    a. Screenshot of each individual sensor active for the mission.
    b. Screenshot of the plot for all sensors active at the same time.
16. Dynamic settings for the font size based on the number of Aerial Vehicle participating in the mission.
17. Designing the appearance and colour for the data displayed to the user.

3.2 Non-Functional Requirements:

1. The application must be written in a platform independent language so that it can work on any Operating System.
2. Ability to handle large data without any issues in graphic rendering
3. Try to reconnect to the server immediately in case of network failure.

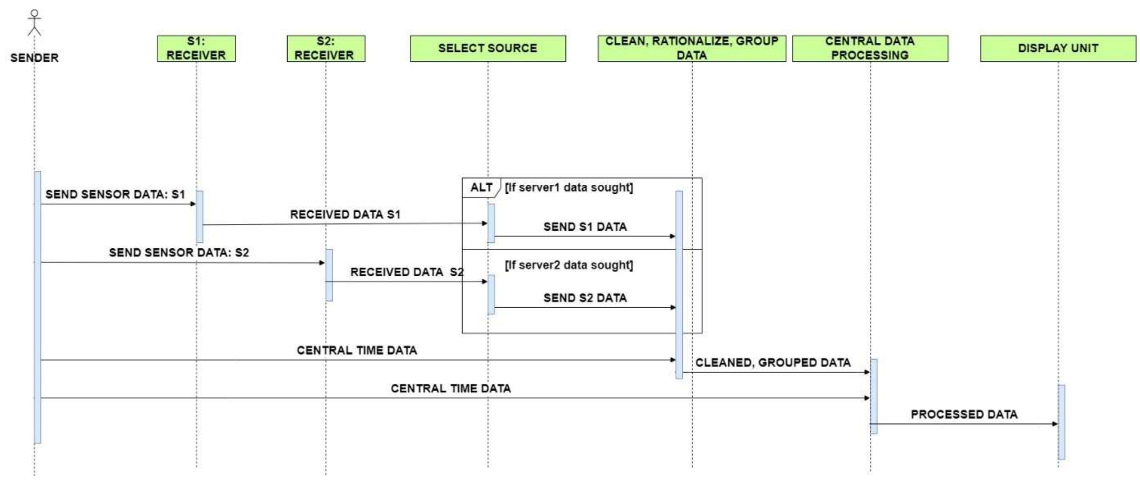
4. ARCHITECTURAL DESIGN

4.1 Use Case Diagram:

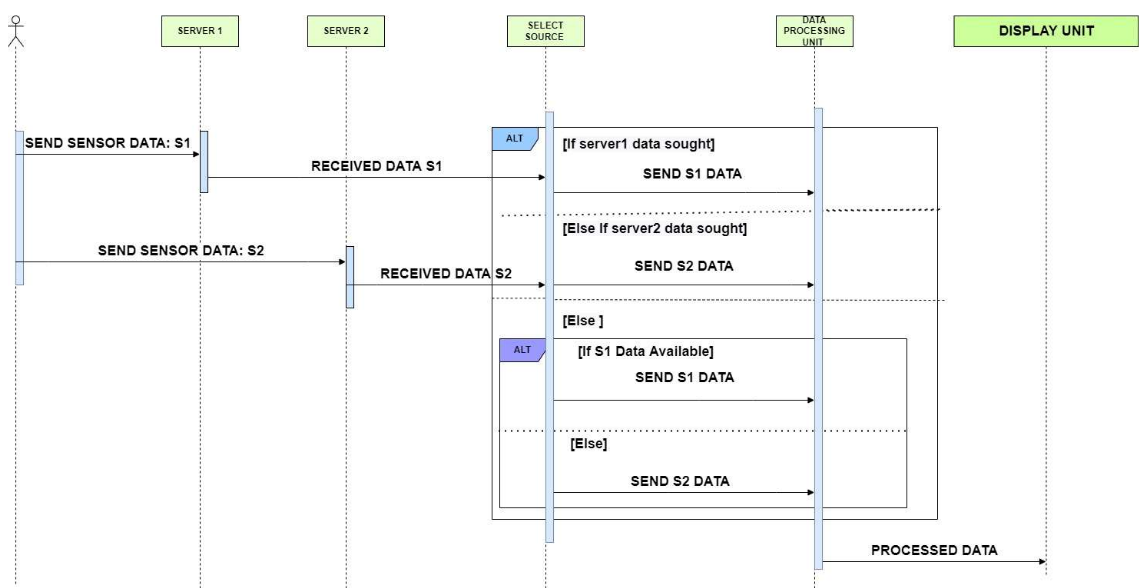


4.2 Sequence Diagram:

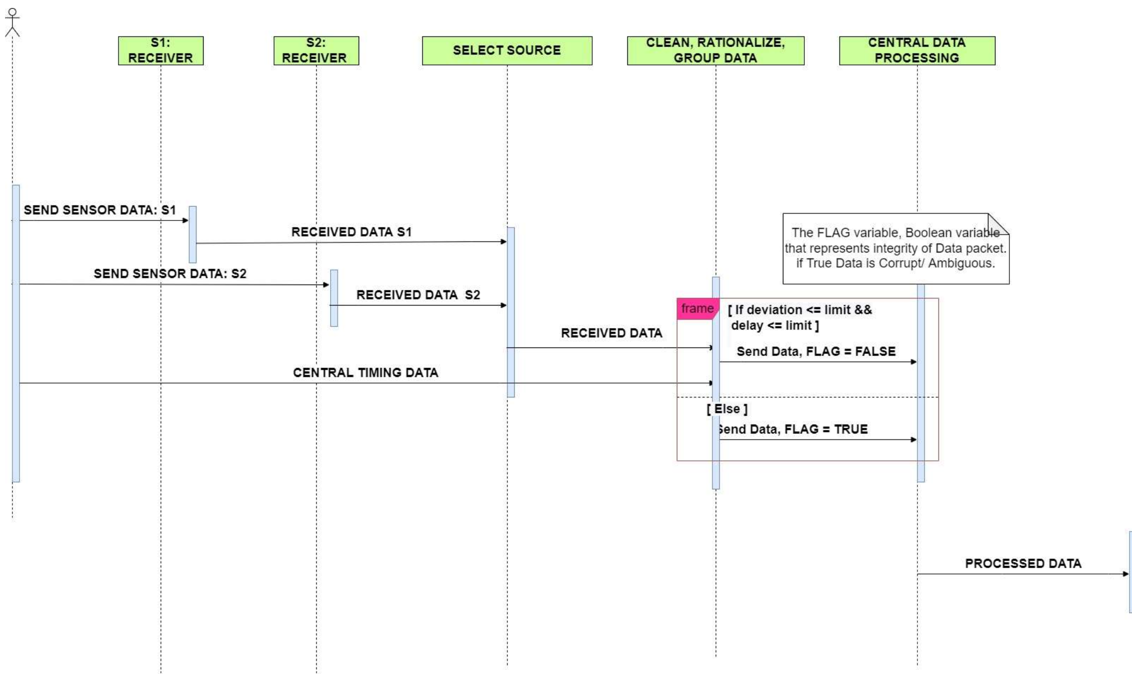
4.2.1 Overall Functionality of the Central Data Processing Centre.



4.2.2 Source Selection Algorithm Mechanism.

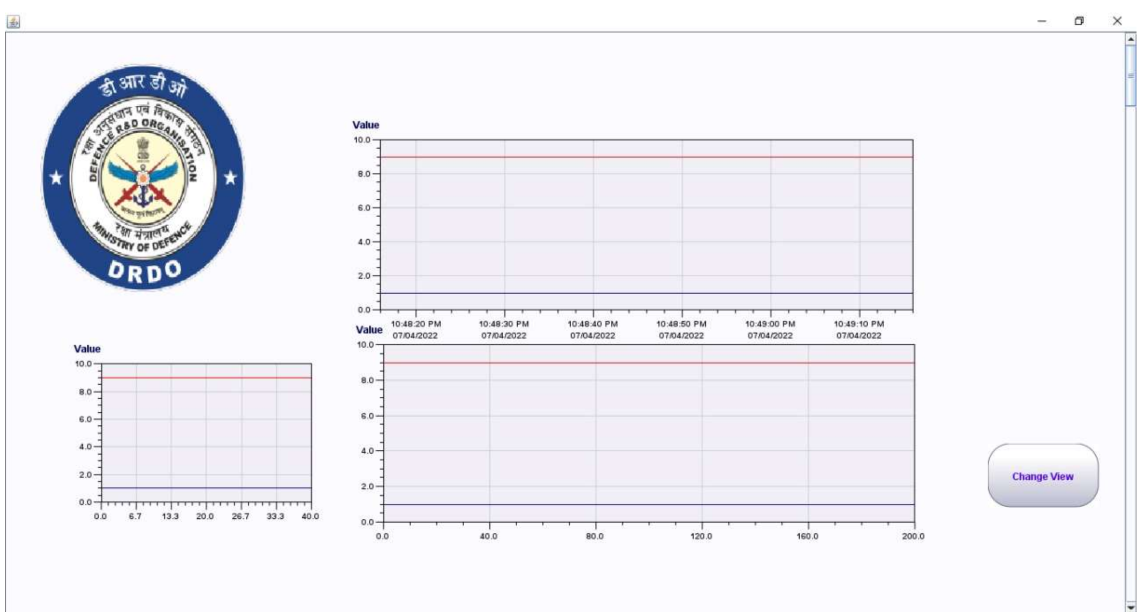


4.2.3 Integrity Check, Cleaning, Grouping of Received Data.



5. INTERFACE DESIGN

5.1 Main View



5.1.1 Graph

The main view comprises of two categories of graph: Main Graph and Zoom Graph. On the basis of selection by the user, the following views are possible in the main view:

a. Down range vs Altitude
b. Time vs Altitude
c. Zoom-in view of first Plot

5.1.2 Sensor Table.

It displays all the active sensors participating in the mission and each sensor plot can be enabled/disabled by clicking on the sensor name.

5.1.3 Aerial Vehicle Table.

It displays the crucial Aerial Vehicle parameters that are needed for inspection by the client. The values are updated every 1 second to enhance used readability.

5.2 Combo View

Fig 7. Combo View

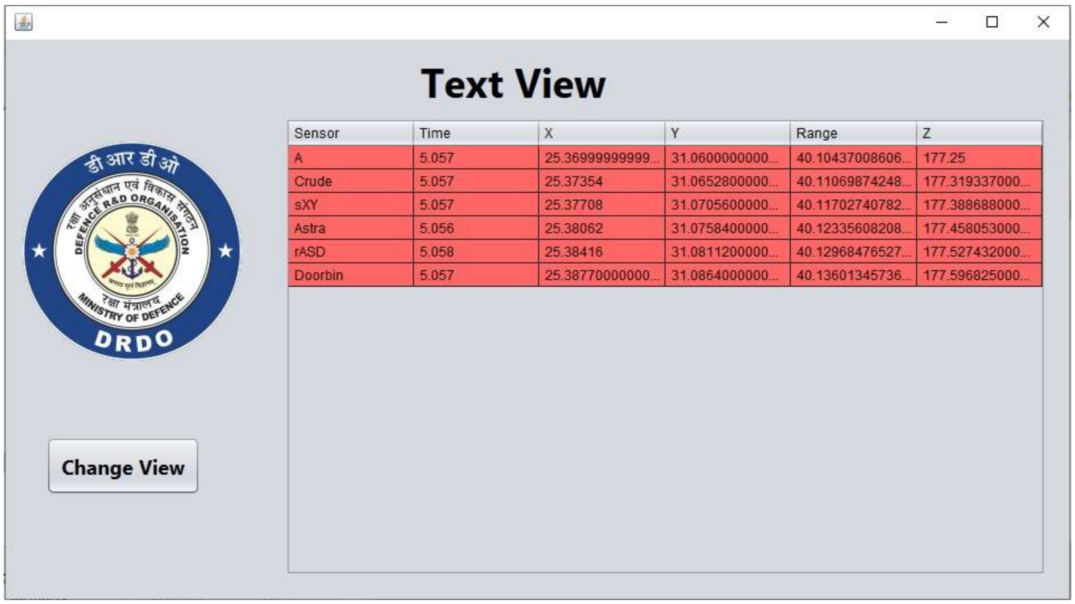
It consists of three types of plots, all represented on the same screen:

a. Top View: Downrange vs Crossrange plot
b. Side View: Altitude vs Downrange plot
c. Time Altitude View: Altitude of sensor vs Count up time

Additionally, it also consists of the sensor table and Aerial Vehicle table similar to the main view.

5.3 Text View:

This view displays the sensor data received by the application, each time the plot is updated, without any delay. The sensors are grouped together to track a particular Aerial Vehicle and to enhance readability, each group's font colour, background colour, Aerial Vehicle header colour and Aerial Vehicle background colour can be configured in the settings.



The parameters displayed are: Sensor Name, Time, Rx, Status, Range, Altitude, Azimuth, I-Azimuth, I-Elevation.

5.4 Menu Bar

5.4.1 File:

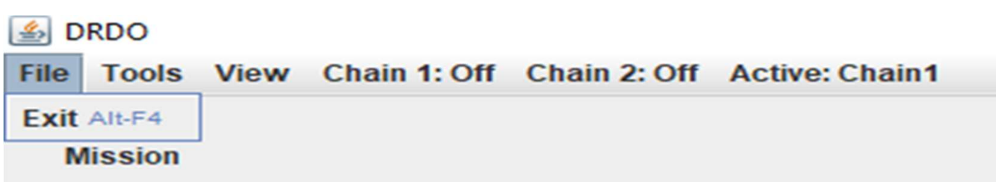


Fig 9. File Menu Bar

The menu item provided under this menu bar is Exit, which terminates and closes the application.

5.4.2 Tools

Fig 10. Tools Menu Bar

This menu bar consists of four menu items:

1. Graph Snapshot.
   It captures each graph with individual and collective sensor data and saves it in .png format.

2. Screen Screenshot.
   It captures the screenshot with individual and collective sensor data and saves it .png format.

3. Setting.
   It opens a different frame for settings. It is further described in Section 5.5.

4. Refresh Layout.
   In case the layout of the screen is hampered, the refresh layout option is provided in order to restore the layout to the original format.

5.4.3 View

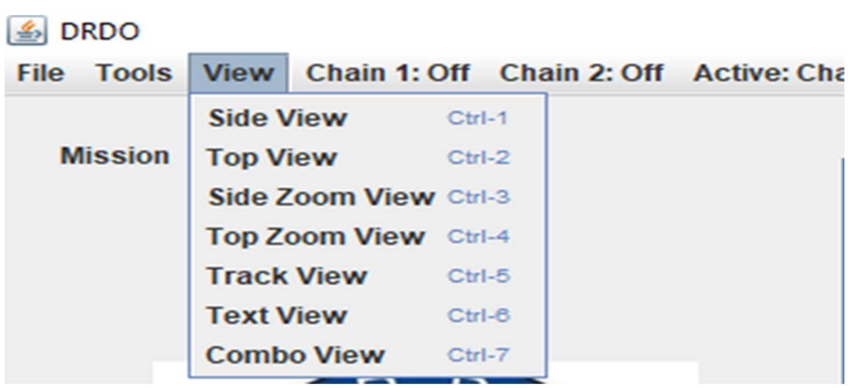


Fig 11. View Menu Bar

This menu bar allows the user to switch between different views, namely: 1- Side View

1. Top View
2. Side Zoom View
3. Top Zoom View 5- Track View

4. Text View
5. Combo View

5.4.4 Chain Information:

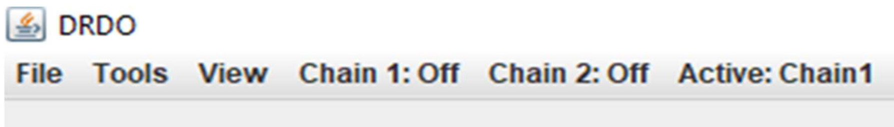


Fig 12. Chain Information Bar

There are three menu bars with no menu items, whose purpose is to display the status of the two Servers. These menu bars are-

1. Chain 1 - Displays whether Server 1 is sending data or not.
2. Chain 2- Displays whether Server 2 is sending data or not
3. Active- Displays the current Server being used for plotting the graphs

5.5 Settings.

In order to configure the appearance and scenarios for each mission differently, according to the needs of the mission and the clients, settings have been provided, which are further categorized into six types:

1. Graph Settings.
2. Network Settings.
3. Latitude- Longitude Settings.
4. Sensor Settings.
5. Appearance Settings.
6. Scenario Settings

5.5.1 Graph Settings

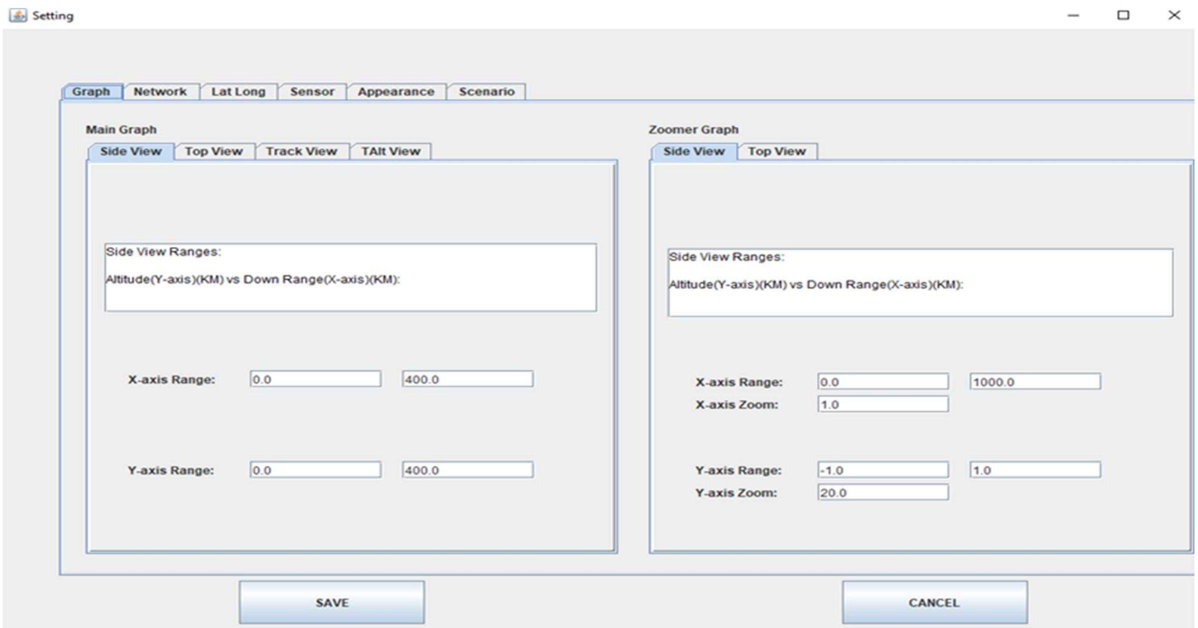


Fig 13. Graph Range Settings

The graph settings provide control over the X axis range and Y axis range for all the different types of graphs along with the title of each plot and its respective axis.

5.5.2 Network Settings

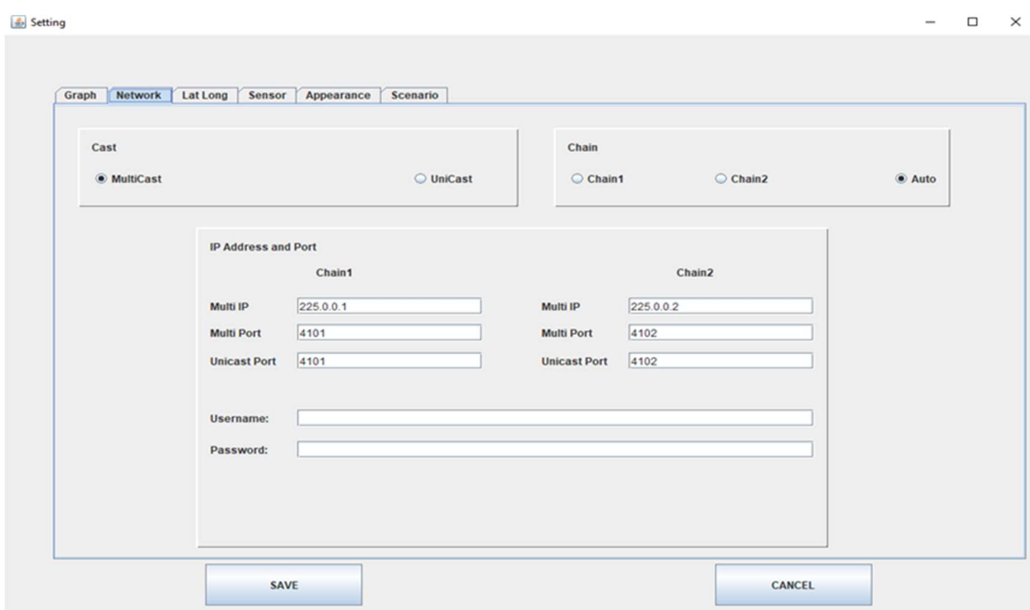


Fig 14. Network Settings

This tab enables the selection of unicast or multicast mode of connection with the server, along with the preferred chain for plotting the graph. For each mode the IP address (if multicast) and port number must be given as input for both the servers transmitting data.

5.5.3 Latitude-Longitude Settings

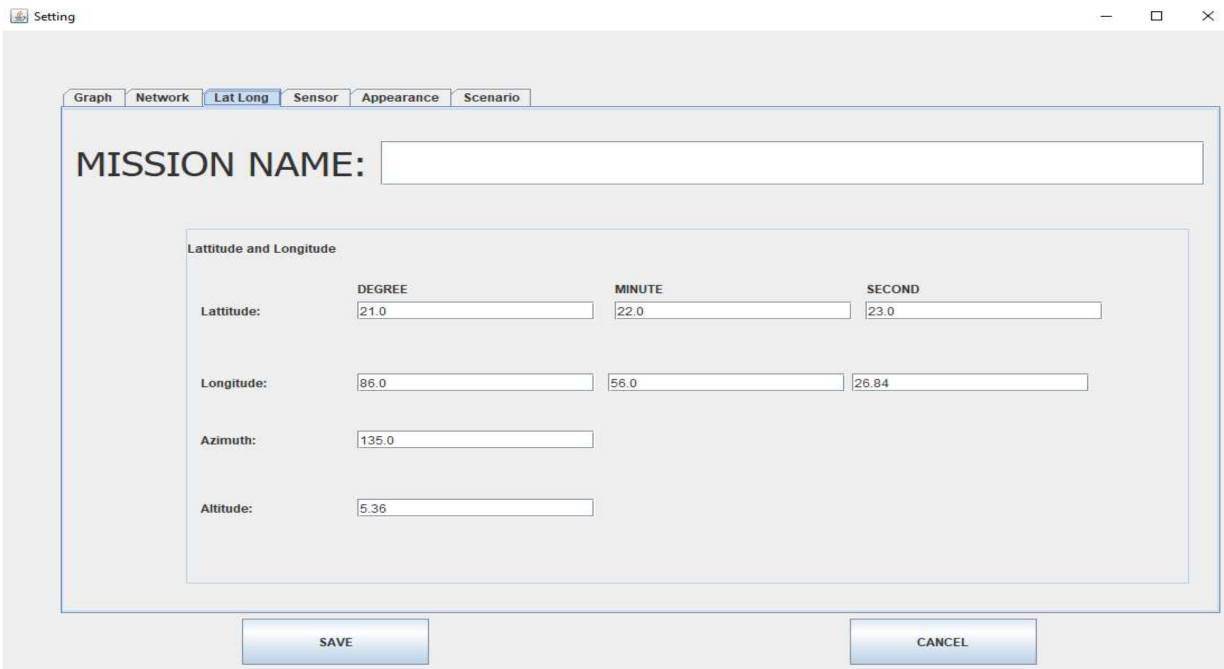


Fig 15. Latitude-Longitude Settings

This tab allows configuring the mission’s name and assigning the values of latitude, longitude, Azimuth and Altitude for the mission.

5.5.4 Sensor Settings

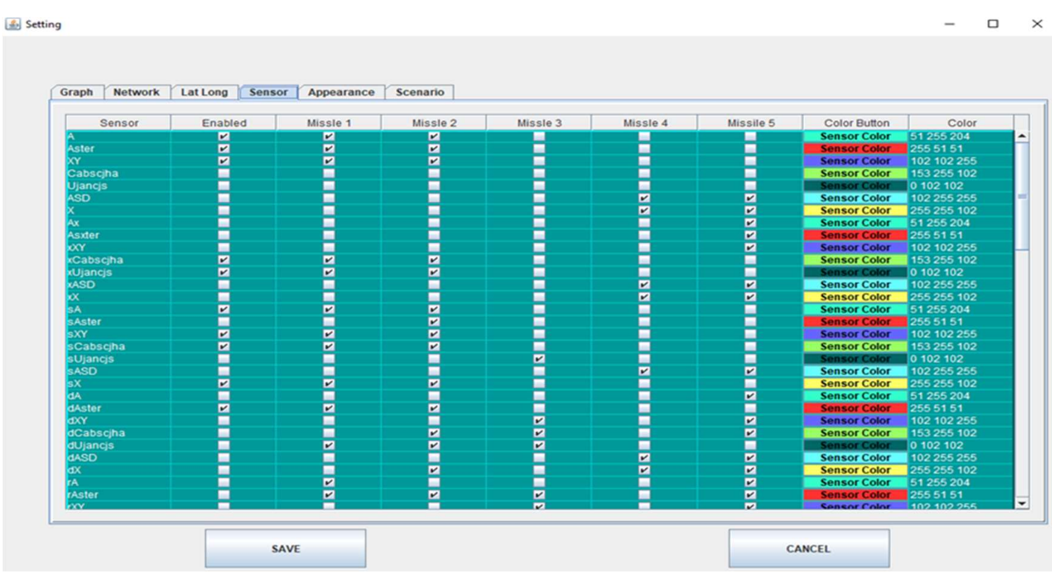


Fig 16. Sensor Settings

This tab allows the selection of the active sensors that are supposed to track a group of Aerial Vehicles and also the colour of each sensor can be adjusted.

5.5.5 Appearance Settings

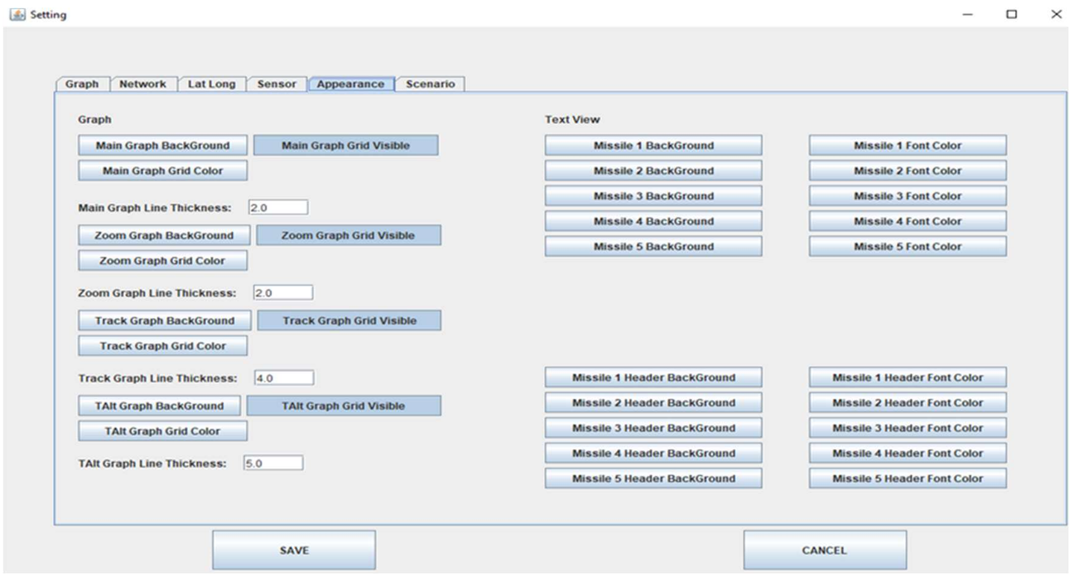


Fig 17. Appearance Settings

This tab enables the user to change the of the plot background colour, grid colour, plot thickness and togging between displaying or hiding grid lines for each graph. It also provides functionality for adjusting the background colour and font colour for a group of sensors tracking one Aerial Vehicle, as well as the background colour and font colour of each headers for the Aerial Vehicle.

5.5.6 Scenario Settings

It enables annotating different parts of the graph with different shapes and text like circles, lines and points. Addition of a nominal file to the plot helps in defining the predicted path to be followed by the Aerial Vehicle. Thus, the deviation between the predicted path and the actual path followed by the Aerial Vehicle during the mission can be observed.

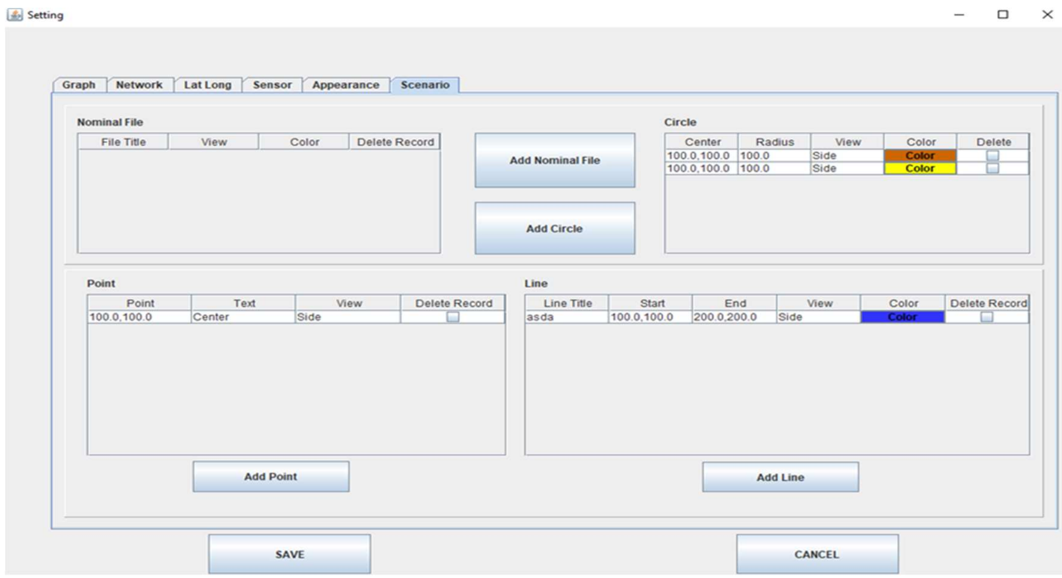


Fig 18. Scenario Settings

6. CODE DESCRIPTIONS

6.1 Reason for choosing Java

Java is an object-oriented programming language. Object Oriented Programming has great advantages over other programming paradigms.

1. Code Reuse and Recycling:
   Objects created for Object Oriented Programs can easily be reused in other programs.

2. Encapsulation:
   Once an Object is created, knowledge of its implementation is not necessary for its use. In older programs, coders needed understand the details of a piece of code before using it. Also, implementation details of objects are abstracted and not available for modification. This prevents accidental/unnecessary modification of values.
3. Design Benefits
   Large programs are very difficult to write. Object Oriented Programs force designers to go through an extensive planning phase, which makes for better designs with lesser flaws. In addition, once a program reaches a certain size, Object Oriented Programs are easier to program than non-Object-Oriented ones.
4. Software Maintenance:
   Programs are not disposable. Legacy code must be dealt with on a daily basis, either to be improved upon (for a new version of an exist piece of software) or made to work with newer computers and software. An Object-Oriented Program is much easier to modify and maintain than a non-Object-Oriented Program. So, although a lot of work is spent before the program is written, less work is needed to maintain it over time.

6.2 Reason for choosing NetBeans, as development environment

1. Free and Open Source: NetBeans is a free and open-source software with thousands of users ready to help and contribute.
2. Powerful GUI Builder: The GUI Builder (formerly known as Project Matisse) makes creation of GUIs simpler.
3. Profiling and Debugging Tools: With NetBeans IDE profiler, real time insight into memory usage and potential performance bottlenecks is obtained. The Heap Walker tool helps evaluate Java heap contents and find memory leaks.
4. Extensible Platform: Extending the platform and its Swing-based foundation saves development time and can optimize performance.
5. Customizable Projects: Through the NetBeans IDE build process, which relies on industry standards such as Apache Ant, make, Maven, and rake, rather than a proprietary build process, projects can be customized and functionalities can be added. Projects to servers outside of the IDE can be built, run, and deployed.

6.3 JAVA classes used

1. **Datagram Packet**
   This class represents a datagram packet. Datagram packets are used to implement a connectionless packet delivery service. Each message is routed from one machine to another based solely on information contained within that packet. Multiple packets sent from one machine to another might be routed differently, and might arrive in any order. Packet delivery is not guaranteed.

2. **Datagram Socket**
   This class represents a socket for sending and receiving datagram packets. A datagram socket is the sending or receiving point for a packet delivery service. Each packet sent or received on a datagram socket is individually addressed and routed. Multiple packets sent from one machine to another may be routed differently, and may arrive in any order.

3. **IO (Input-Output)**
   This class provides for system input and output through data streams, serialization and the file system. Unless otherwise noted, passing a null argument to a constructor or method in any class or interface in this package will cause a Null Pointer Exception to be thrown. This class is used in the project for reading and writing of files.

4. **Thread**
   Thread a line of execution within a program. Each program can have multiple associated threads. Each thread has a priority which is used by the thread scheduler to

determine which thread must run first. Java provides a thread class that has various method calls in order to manage the behaviour of threads by providing constructors and methods to perform operations on threads.

5. **Timer**
   This class fires one or more Action Events at specified intervals. An example use is an animation object that uses a Timer as the trigger for drawing its frames. Setting up a timer involves creating a Timer object, registering one or more action listeners on it, and starting the timer using the start method.

6. **JFrame**
   A frame, implemented as an instance of the JFrame class, is a window that has decorations such as a border, a title, and supports button components that close or iconify the window. Applications with a GUI usually include at least one frame.

6.4 Several GLG Bean classes are provided

a. GlgBean- AWT-based Java bean.

b. GlgJBean- Heavyweight Swing-based Java bean.

c. GlgJLWBean- Lightweight Swing-based Java bean for use with JDesktopPane and JInternalFrame.

6.5 GLG Graphics Server for JSP / Java

A separate GLG class library for the JSP environment is provided in the GlgServer.jar file. This class library provides the GLG Extended API, and also supports headless operation as well as a multi-threaded access by the servlet threads. When used in a servlet, the GlgObject classes should be used instead of the GLG Bean. The examples_jsp directory contains elaborate examples of the GLG servlets with the source code.

1. **GlgHListener**
   public interface GlgHListener, Interface for setting the initial values of resources of the drawing displayed in the GlgBean or GlgControl.

   Methods:

   public void HCallback (GlgObject viewport)

   This callback is invoked after the component's drawing is loaded, but before the drawing hierarchy is set up. It may be used to set the initial resources of the drawing, such as the number of samples in a GLG graph object. The viewport parameter is the top viewport of the drawing displayed in the GlgBean or GlgControl.

2. **GlgVListener**
public interface GlgVListener

Interface for setting the initial values of resources of the drawing displayed in the GlgBean or GlgControl.

Methods

public void VCallback(GlgObject viewport)

This call back is invoked after the component's drawing is loaded and set up, but before it is displayed for the first time. It may be used to set the initial resources of the drawing. For example, it may be used to supply data for the initial appearance of a GLG graph object. The viewport parameter is the top viewport of the drawing displayed in the GlgBean or GlgControl.

3. **GlgReadyListener**
public interface GlgReadyListener

Interface for detecting the Ready event.

Methods

public void ReadyCallback(GlgObject viewport)

This callback is invoked after the component's drawing is loaded, setup and displayed for the first time. It may be used to detect when the GlgBean or GlgControl is ready and begin updating the drawing with data. The viewport parameter is the top viewport of the drawing displayed in the GlgBean or GlgControl.

4. **GlgInputListener**
It is used for handling with the user.

6.6 Standard API Methods of GlgBean and GlgControl Containers

1. public boolean SetDrawingObject (GlgObject viewport)
Sets a new viewport object as a component's drawing and displays the new drawing.

2. public void InputCallback (GlgObject top_viewport, GlgObject message_object)
The default InputCallback invoked every time the user interacts with input objects in the drawing. More information about the type of the input activity may be extracted from the message_object parameter.The component's AddListener method may be

used to change the default callback. The top_viewport parameter is the top level viewport of the component's drawing.

7. TESTING

7.1 Receiving Packet:

Packets of data were sent using the Tracking Instrument Simulator (TIS) and the application was tested in different chain mode in order to verify that no sent packet of data was missed. The test cases for testing the chain mode and packet receiving were:

- Server 1 on, Server 2 off
- Server 1 off, Server 2 on
- Server 1 off, Server 2 off
- Server 1 on, Server 2 on

For each case all 3 chain modes of selection were tested, namely, Chain 1, Chain 2 and auto mode. The results were accurate and satisfactory.

Server 1	Server 2	Forced Chain 1	Forced Chain 2	Auto
On	On	Chain 1 data	Chain 2 data	Chain 1 data
On	Off	Chain 1 data	No data	Chain 1 data
Off	On	No data	Chain 2 data	Chain 2 data
Off	Off	No data	No data	No data

7.2 Plotting and Rendering All Views:

As soon as the packet of data was received, the corresponding plots were updated and all views were properly rendered. The number of active sensors were increased and the latency in plotting and displaying was tested, which gave satisfactory results for even large number of sensors.

7.3 Screen snapshot and Graph snapshot:

Ideally the snapshot functionality is to be used at the end of the mission, but for testing purpose, it was also done while plotting the graphs. Although the snapshots took some seconds to complete, but this did not hamper or interfere with the process of plotting.

7.4 Settings File and Sensor File:

Rigorous testing of the Settings file was required because the configuration and appearance for each mission is stored in the Settings file. The following test cases were used:

- Settings file did not exist: Error message shown on screen and default values are assigned to variables.
- Settings file exists but some error in between: In this case the file was read and variables were updated till the error is not encountered. Updating the settings file through settings menu resolved the error.
- Same settings file exists for different packet size: The application worked as intended and just skipped reading the lines which were not meant for the concerned packet size application.
- File deleted after reading and updating variables: Updating the settings file was not possible as a result of which any changes intended for the application was not be rendered.
- File exists throughout runtime: The application ran as intended and encountered no errors.

7.4 Web Server Performance Test:

Web testing, or web application testing, is a software practice that ensures quality by testing that the functionality of a given web application is working as intended or as per the requirements. Web testing allows you to find bugs at any given time, prior to a release, or on a day-to-day basis.

- Does the web server return a valid response? A response of code of 200 is expected from a properly functioning web server. Responses such as 401, 404 and 503 may indicate a problem with the server. Individuals within your organization may not be seeing server error responses, but users outside of your network may be encountering server errors. In these cases, users are not likely to report these issues and the issue could go unchecked for a long period of time, potentially impacting the bottom line.
- Do all the elements on the page or application form return valid responses? Even if a page itself is loading properly, there could be individual elements called by the page that do not successfully load due to several factors, including third party hosts, CDNs, network latency, and DNS resolution issues.
- How fast does the website respond? How long does it take until the user is able to begin interacting with the page? A long time to interactive (TTI) can frustrate users and make it seem that your website is being unresponsive. This is especially true for mobile devices where long load times may seem more noticeable to users.
- What is the total time to load of the entire page? How long does it take to completely load all the content specified in the HTML of the given page? There may be third-party elements on your page that is causing the page to load longer than necessary.

7.5 Nominal File

Different scenario settings were established throughout testing and each time, the application as well as the nominal file was updated as per the user input.

- In case nominal file was missing, a dialogue box appears on the screen indicating that there was an error in reading nominal file. However, the plot scenario could still be

updated on all the views. Only the nominal file could not be updated
- Error in nominal file: Scenario settings only updated until error is not encountered. This error could not be resolved through the program and required manual checking of the nominal file because updating nominal file involves only appending to the file.
- Nominal file exists without error: The application executed correctly.

8. CONCLUSION AND FUTURE WORK

The application was tested and verified through testing and all the requirements of the clients were fulfilled. The application can now be used for a mission scenario within the constraints of 5 Aerial Vehicles and 100 sensors. It supports receiving data from 2 servers.

Future work involves improving the scenario of 5 Aerial Vehicles and 100 sensors to accommodate more Aerial Vehicles and sensors if required. Furthermore, this application gives accurate view on a screen resolution of 1600 X 900 or greater. Work can be done to support lower resolutions if the clients does not meet the minimum hardware requirements to support this application.

9. REFERENCES

1. https://www.drdo.gov.in/citizen-charter
2. https://www.drdo.gov.in/labs-and-establishments/integrated-test-range-itr
3. https://en.wikipedia.org/wiki/Radar
4. https://www.mathworks.com/help/map/choose-a-3-d-coordinate-system.html
3. http://www.rfwireless-world.com/Terminology/Tracking-Radar-basics-and-types.html
4. https://en.wikipedia.org/wiki/Telemetry
5. https://en.wikipedia.org/wiki/Axes_conventions
6. https://en.wikipedia.org/wiki/Cartesian_coordinate_system
7. https://en.wikipedia.org/wiki/Spherical_coordinate_system