Design and Implementation of Radar Console Displays for Multi Object Tracking Radar using Qt-IDE

Ch.Ravindra, S.Rajkumar, M.Sreenivasa Babu.
Multi Object Tracking Radar (MOTR), Satish Dhawan Space Centre (SHAR), Sriharikota
Indian Space Research Organisation, (ISRO), AP, India.

Email: challaravi@shar.gov.in

Abstract – Design, development and deployment of Console Display Application realized at Multi Object Tracking Radar, Sriharikota. Console Display Application includes graphs such as Polar Plot, Ground Trace and Radar Parameter Display. Incremental Model of Software Development is opted for realization process. In addition, this paper describes about Console Display Performance in real time.

Keywords – Polar Plot, Ground Trace, Incremental Software Development Model

I. INTRODUCTION

This paper describes about **C**onsole **D**isplay **A**pplication (CDA) realized at Multi Object Tracking Radar (MOTR). MOTR is an L-Band Active Phased Array Radar with 4608 Transmit Receive Modules (TRMs) designed for tracking multiple targets using single agile beam with electronic Beam Steering technique. MOTR is capable of tracking multiple targets of size 0.25m^2 target up to a range of 1000 km. Main objective of MOTR is to utilize its long range skin mode tracking capability to track space debris up to an altitude of 800km. It is the first radar system in India realized for tracking space debris.

At MOTR, user can monitor and control all major subsystems of radar under one console. Console Display Application at MOTR includes, Antenna Status Display and Graphical User Interface Display. Antenna Status Display of 4608 TRMs, which are monitored for Power Output, Trigger status and Temperature. Graphical User Interface Display, to provide user with real time information of multiple targets under track.

As MOTR is a phased array radar with electronic beam steering, Azimuth sweep of MOTR antenna is +60deg to -60deg with respect to antenna Boresight, Elevation sweep of MOTR antenna is +45deg to -45deg. With the versatility in tracking of targets, multiple graphical user interface displays are required at MOTR. Graphical User Interface Display realized at MOTR includes

- Polar Plot: Polar Plot presents radar operator about targets under target with in the Boresight limit.
- Target Ground Trace Display: Target Ground Trace presents radar operator with information of targets position with respect to non-inertial earth frame.
- Radar Parameter Display: Radar Parameter Display presents radar operator with dynamics of target under track. Target dynamics such as range, range rate, azimuth and azimuth rate.

This paper mainly focuses on Graphical User Interface of CDA. Design, Development and Deployment of Console Display Application realized at MOTR are discussed in detail. Section two details about development cycle adopted for realizing, Section three details about User Requirement capturing and analysis of the requirements, Section four describes about design and implementation of user requirements, Section five and six describes about testing and deployment process carried.

II. DEVELOPMENT PROCESS

Incremental Software Development Model is opted for developing the CDA. An Incremental model involves multiple Software Development Cycles. Each developmental cycle involves following steps

- Capturing and analysis of user requirements
- Design and implementation
- Testing
- Deployment in real time.

Where-in after completion of each development cycle, application will be tested and deployed in real time at user end. Based on previous development cycle application performance, a new development cycle will be initiated. In the new development cycle, new requirements from user are gathered and the cycle continues till test and deployment. Three Versions of Console Display Application are deployed at MOTR. In the first version CDA can handle real time plotting. In second version CDA is added with offline plotting, reading from file and plotting on the displays

III. REQUIREMENTS AND ANALYSIS

The summary of requirements and its analysis for CDA are as follows

A. REQUIREMENTS FROM THE USER ARE AS FOLLOW:

A total of 3 graphical user interface pages consisting of Polar Plot, Target Ground Trace and Radar Parameter Display.

Polar Plot: MOTR operates in air surveillance mode by scanning in 360 degrees of azimuth. MOTR antenna employs electronic beam steering technique to scan and track targets. Radar operates initiates a defined scan volume to assess air situation. Polar Plot helps radar operator in giving optimized scan volume. Along the display, numerical information of targets under track is tabulated.

Target Ground Trace Display: Information of targets under track will be displayed in Geodetic-WGS-84 standard. Ground Trace Display helps radar operator to identify targets flying above restricted latitude-longitude regions. MOTR track data is plotted against earth surface boundaries along latitude-longitude grids.

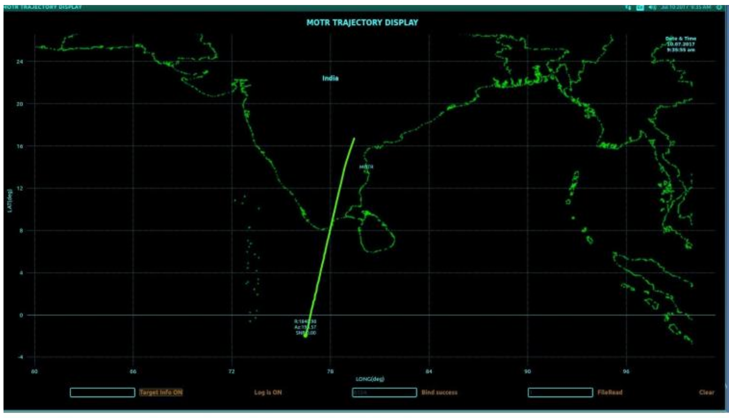


Figure 1 Ground Trace Display of MOTR Tracking PSLV_Debris

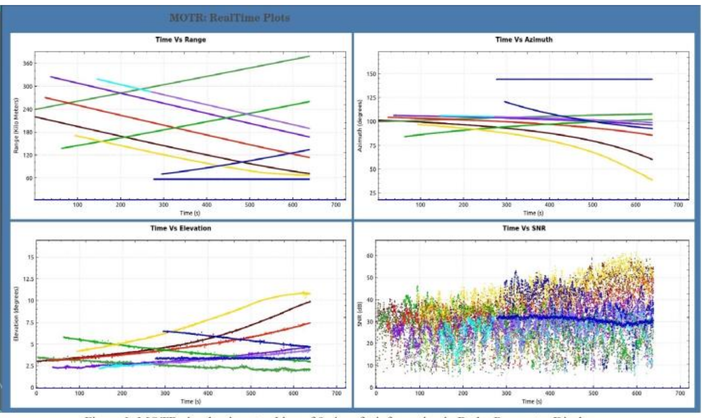


Figure 2 MOTR simultanious tracking of 9 aircrafts information in Radar Parameter Display

Radar Parameter Display: MOTR tracks multiple targets ranging from forward moving launch vehicle to separated objects from launch vehicle. In a single tracking session each object under track has variable characteristics. Radar parameter display presents radar operator with Range, Azimuth, Elevation and SNR, in four different graphs versus time scale. Radar operator can change to Range Rate, Azimuth Rate, and Elevation Rate.

Major design requirements are listed below.

- Usability across multiple screens with adaptive design, including a 1920x1080 screen, a 2560 x 1600 high-definition video wall, and a 15-inch 1024 x 768 capacitive touch screen.
- The ability to compensate for the very high pixel density on a high-definition screen so that fonts and icons are easily readable for average users.
- Console Display Application is platform independent that is, it runs on Linux, Ubuntu and Windows.
- Interactive zooming options in real time. With mouse scroll in and out, radar operator zoom to a one single detection of the target on the display in real time. Dragging of graph to desired point is provided with mouse click-hold-drag operation.
- Data need to be received from radar controller with User Datagram Protocol (UDP); four bytes of information is considered as one parameter. A total of 1000 bytes are received in each packet.
- Display application is capable of receiving data at a rate of 10*1000 bytes to 1000*1000 bytes per second. Data rate is based on the Pulse Repetition Frequency of radar.
- Plotting of logged data from a file is provided. User can select a desired file from Radar Controller log, data is read from file and updated on multiple graphs similar to real time.
- Once plotting is done, upon user selection all the graphs are converted to PDF format and send for printing in A4-sheet portrait mode.

B. *ANALYSIS OF REQUIREMENTS:*

- Since display should be able to run on multiple Operating Systems, as Qt-IDE is cross platform [1] (Windows to Linux migration requires minimal code changes), Qt-IDE is chosen for realizing the Display Application.
- An open source software QCustomPlot is interfaced to Qt-IDE to enhance the plotting experience, by this user interactive features like zooming and scaling is done with minimal code changes.
- Commercial off-the-shelf Hardware with one Network Interface Card will be sufficient to run the application.
- Since all the user requirements point to 2-Dimentional graphs only, no additional licensing for 3D support of Qt-IDE and graphic card hardware are not required.
- Second page of the display requires geodetic information, for this real-time conversion of radar's East North Up coordinate system parameters to WGS-84-Geodetic format in C++ is required.

IV. DESIGN AND IMPLEMENTATION

A. *DESIGNING OF USER INTERFACE*

With the aforementioned requirements and its analysis, the application is designed using Qt widget with custom styling. The user interface is written in XML; Models and logic associated to interface is implemented in C++ language; Maps and Graphs are realized using QPainter and QCustomplot.

Table1 summarizes kind of base widgets used for realising various features.

Feature	***Implemented using***
Real time page switching	Stack widget
Polar Plot	QPainter
Map, Geodetics graph	QCustomPlot
Dynamic font faces	QStylesheet and Layouting

Table 1: Base widgets used for graphical design

A brief overview of the major widgets such as QPainter and QCustomPlot used in implementing CDA, are summarized below.

QPainter: QPainter [2] class provides standard functions to draw points, lines, ellipses, arcs, Bézier curves, and other primitives. QPen and QBrush are basic classes used by QPainter. Qpen is used for drawing lines and outlines of shapes. QBrush class defines the fill pattern of shapes drawn by QPainter.

QCustomPlot: QCustomPlot [3] is a Qt C++ widget for plotting and data visualization. This plotting library focuses on making good looking, publication quality 2D plots, graphs and charts, as well as offering high performance for real time visualization applications. QCustomPlot can export to various formats such as PDF files and rasterized images like PNG, JPG and BMP. QCustomPlot is an independent third party widget with no additional packages required.

OFFLINE PLOTTING: File reading and plotting on to the graphs similar to real time is provided. User can select the file and desired fields of the file to be plotted on the graph. File will be read in line mode at data rate of 100 lines per second and displayed on the graph similar to real time.

B. RADAR ENU TO ECEF CONVERSION:

Following steps are involved while converting radar raw data to geodetic data, ENU to ECEF non-inertial frame conversion [4].

Step1: Radar's data is in East-North-Up (ENU) polar format that is in Range, Azimuth and Elevation form.

Step2: Radar data of ENU-polar data is converted to ENU-Cartesian form.

Step3: Reference radar position of MOTR, this is at latitude of 13.7358deg, longitude of 80.18473deg and altitude of 30meters, converted to Earth-Centered-Earth-Fixed (ECEF) non inertial coordinate system.

Step4: Convert Cartesian Radar data in ENU format to Earth-Centered-Earth-Fixed (ECEF) format.

Step5: Now radar data is in ECEF non inertial format. Now considering earth's curvature as per earth radius mentioned in WGS-84 standard, data of ECEF is converted to geodetic.

C. IMPLEMENTATION:

Ground Trace Display:

MOTR-Ground Trace Display is graphical display of satellite ground track or air craft ground trace. Targets information of Range, Azimuth and Elevation received in East-North-Up (ENU) coordinate system are converted to WGS-84 standard geodetic information. Geodetic information is displayed on the surface map of earth along the latitude - longitude code grids.

QCustomPlot is used for realizing the Ground Trace Display. Reference boundary information of ground is stored in a file. At the time of starting the application, boundary values in latitude-longitude are read from file and displayed on to the widget.

Section III.B describes about various steps involved in converting Radar data to geodetic information. Detailed equations used for converting the data are not listed in this paper.

Figure 1 describes about Ground Trace Display realized at MOTR. In figure1 PSLV-PS4 spent stage is under track by MOTR. Ground Trace Display of MOTR has full zoom and off-centre control feature. Geodetic information of CDA is compared and plotted with Systems Tools Kit Geodetic information.

Radar Parameter Display (RPD):

MOTR-Radar Parameter Display involves representation of target's Range, Azimuth, Elevation and Signal to Noise Ratio (SNR) information in four different graphs. Each target is identified with an ID tag given by Radar Controller. Using the Target ID as reference, different colours are assigned to each target trace.

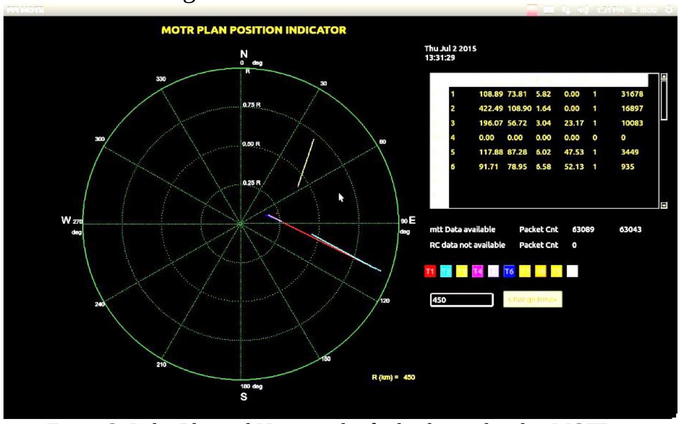
QCustomPlot is used for realizing the Radar Parameter Display. A total of four QCustomPlot widgets are imported to main designer UI, each widget is dedicated for different target information under track. Before start of track session user can select positions of the packet information to be plotted; That is, user can select either range information or range rate information in first graph, similarly for rest three graphs.

After completion of a track session export to PDF option is provided to the user. Upon user selection all the four graphs are exported to PDF in A4-sized portrait mode, ready for printing. Radar Parameter Display of MOTR has full zoom and off-centre control feature. User can pause plotting session and zoom to single detection in real time, in pause mode data is buffered and plotted on to screen once plotting is resumed.

Figure 2 describes about Radar Parameter Display realized at MOTR. Real-time performance of MOTR tracking nine targets simultaneously is displayed. Here Range, Azimuth, Elevation and Signal to Noise Ratio of nine targets under track simultaneously are being displayed.

Polar Plot:

MOTR-Polar Plot displays is a polar-coordinate (slant range and azimuth) display. Origin of the sweep represents the radar. Circles represent slant range information, outer most circle represent longest range of target under track. Rather than conventional PPI circles which represent height of target, MOTR-Polar Plot represents slant range information. MOTR- Polar Plot circles are fixed zero degree elevation circles.



Thu Jul 2 2015
13:31:29

1	108.89	73.81	5.82	0.00	1	31678
2	422.49	108.90	1.64	0.00	1	16897
3	196.07	56.72	3.04	23.17	1	10083
4	0.00	0.00	0.00	0.00	0	0
5	117.88	87.28	6.02	47.53	1	3449
6	91.71	78.95	6.58	52.13	1	935

mtt Data available Packet Cnt 63089 63043

RC data not available Packet Cnt 0

Figure 3: Polar Plot and Numerical info display realized at MOTR

True North is represented on top of the circles. A 180 degrees sweep from top indicates south direction. New target information is displayed in dot format. In this display along with graphical representation, numerical information of targets under track is tabulated.

QPainter feature of Qt-IDE is used for realizing Polar Plot. Figure 3 is the realized Polar Plot display at MOTR, different targets are identified with colour. Complete target trace of a target is held for one complete tracking session.

V. TESTING

Console Display Application is tested exhaustively at MOTR using offline simulators from Radar Controller and Multi Target Tracking Filter. Following are Major tests carried out on the display application.

Test1 Launch Vehicle tracking: Display of forward moving target and separated objects from launch vehicle are sent to CDA.
Test2 Space debris tracking: PSLV Rocket body debris program mode simulation is carried out. Data is displayed in all three graphical displays.
Test3 Geodetic information of targets data in ground trace is compared with Systems Tool Kit (STK-AGI) data. Latitude, Longitude and Altitude are displayed with minimal errors when compared with STK data.
Test4 Offline Plotting: Various Radar Controller log files are read and updated on display systems.

VI. DEPLOYMENT

CDA is integrated to MOTR sub-systems. CDA passed extensive operation modes of MOTR such as launch vehicle tracking, aircraft tracking and satellite tracking. Each tracking mode of operation has various data rates, CDA successfully updated data with variable data reception rates.

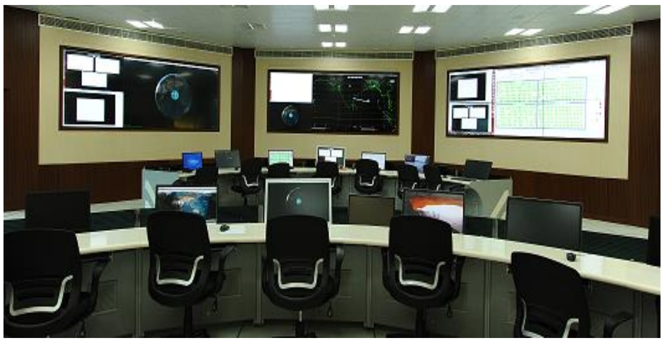


Figure 4: Console Terminal Realized at MOTR

CDA can be connected to legacy type radars as it is realized on COTS hardware. CDA is cost effective, easy to install, requires no special hardware.

VII. CONCLUSION

Console Display Application developed at MOTR involves Polar Plot, Ground Trace and Radar Parameter Display. Display Application is successfully tested and deployed for various tracking modes at MOTR. CDA is a complete ready-to-run cross-platform application. CDA can be integrated to any commercially available radar instantaneously. The realized application is compared with commercially available radar displays and found to be efficient and viable [5]. CDA displays are realized with open source software and COTS Hardware, by which CDA bares a viable cost for radar developers.

ACKNOWLEDGEMENT:
The authors would like to thank Shri.P.Kunhikrishnan (Director, SDSC SHAR) and SDSC SHAR management for their continuous guidance and encouragement to MOTR project related developmental activities and bringing out this paper. The authors also wish to acknowledge the contributions of all their colleagues in SDSC/ISRO for their valuable support in the activities related to MOTR project development.

REFERENCE:

[1] J.Blanchette, M.Summerfield, C++ GUI programming with Qt4, 2nd ed., Prentice Hall 2008.
[2] Qt programing help Available: doc.qt.io/qt-5/qthelp-index.html.
[3] Qcustomplot tutorials available: www.qcustomplot.com/index.php/tutorials
[4] Roger R. Bate, Fundamentals of Astrodynamics, Dover Books on Aeronautical Engineering.
[5] Milovan Stamatovic, Milos Jevtic, Una Kisic and Miloš Tatarevic "Design and Implementation of a Modern Radar Display for Air Surveillance Applications" 20th Teleconference Forum TELFOR 2012

AUTHORS:

Ch. Ravindra did his B.E - Electronics and Communication. He has joined ISRO in 2014 as Scientist/Engineer, worked as systems engineer at Infosys from 2013 to 2104. His field of interest includes, developing of front-end Graphical User Interface for Radar systems. Tracking, Cataloguing and Orbit Determination of Space Debris with Phased array radar.

S. Rajkumar completed his B.E degree in electronics and communication engineering from PSG College of Technology, Coimbatore in the year 2004. He has joined ISRO as Scientist/Engineer in 2004. His area of work includes development of Radar Controller, target tracking algorithms and target data association algorithms for phased array radars.

M. SreenivasaBabu did his graduation in B.E – E.C.E from JNTU, Ananthapur in 1982. He joined ISRO in 1984. He developed Tele-Command and Telemetry ground systems for various launch vehicles missions. Developed various radar sub systems for tracking radars and Active Phased array radars. He is currently designated as Project Director for MOTR project.