DEVELOPMENT OF MULTI-TARGET TRACKER FOR SURVEILLANCE RADAR USING MATLAB

SRIHARI BR - DEPUTY MANAGER
ABHISHEK T - DEPUTY ENGINEER
D&E-CSS-NS2
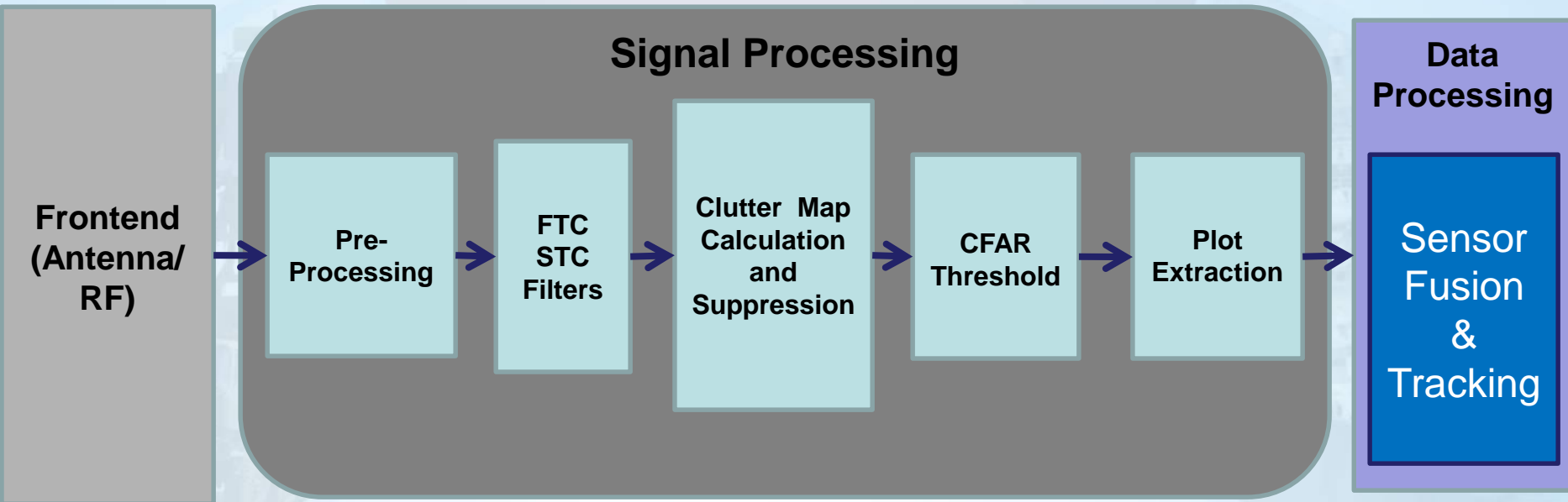BHARAT ELECTRONICS LIMITED
25-04-2019

1

Outline

- ❖ Introduction about BEL
- ❖ Requirements
- ❖ Approach
  - ➢ Tools used :MATLAB Sensor Fusion Tracking Toolbox (SFTT) , CODER
  - ➢ Method employed : Development Phase I - IV
- ❖ Results
  - ➢ Comparison with different approaches
  - ➢ Comparison with Original Equipment Manufacture (OEM)system
- ❖ Key takeaways
- ❖ Looking forward for…

ABOUT BEL

❖ BEL is PSUs under the Ministry of Defence, India.

❖ BEL design, develop and manufactures a wide range of products in the following fields

➢ Radars

➢ Electronic Warfare System

➢ Defence Communication

➢ Tank Electronics

➢ Opto Electronics and Semiconductors

➢ Missile System and Tank Electronics

➢ EVM and VVPAT and many more…

❖ Major customers of BEL comprise of Indian Army, Navy, Air Force, Paramilitary, Coast Guard and many more.

RADAR PROJECT SCHEMATIC

Frontend (Antenna/RF)

Signal Processing

- Pre-Processing
- FTC STC Filters
- Clutter Map Calculation and Suppression
- CFAR Threshold
- Plot Extraction

Data Processing

Sensor Fusion & Tracking

STC : Sensitivity Time Control
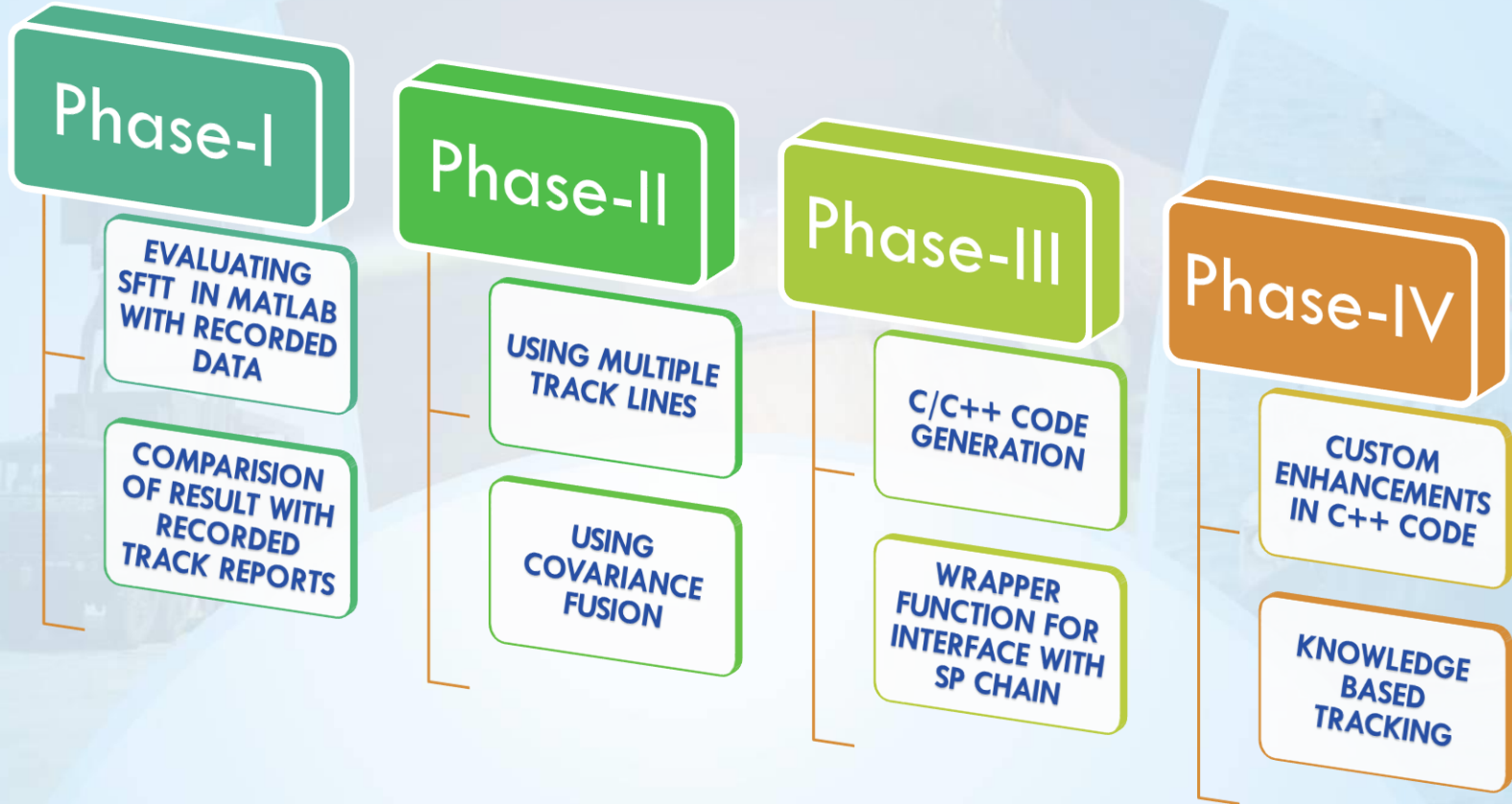FTC : Fast Time Control
CFAR : Constant False Alarm Rate

REQUIREMENTS

- Development of **field deployable multi-target tracking module** using **Sensor Fusion Tracking Toolbox** and **MATLAB coder**.

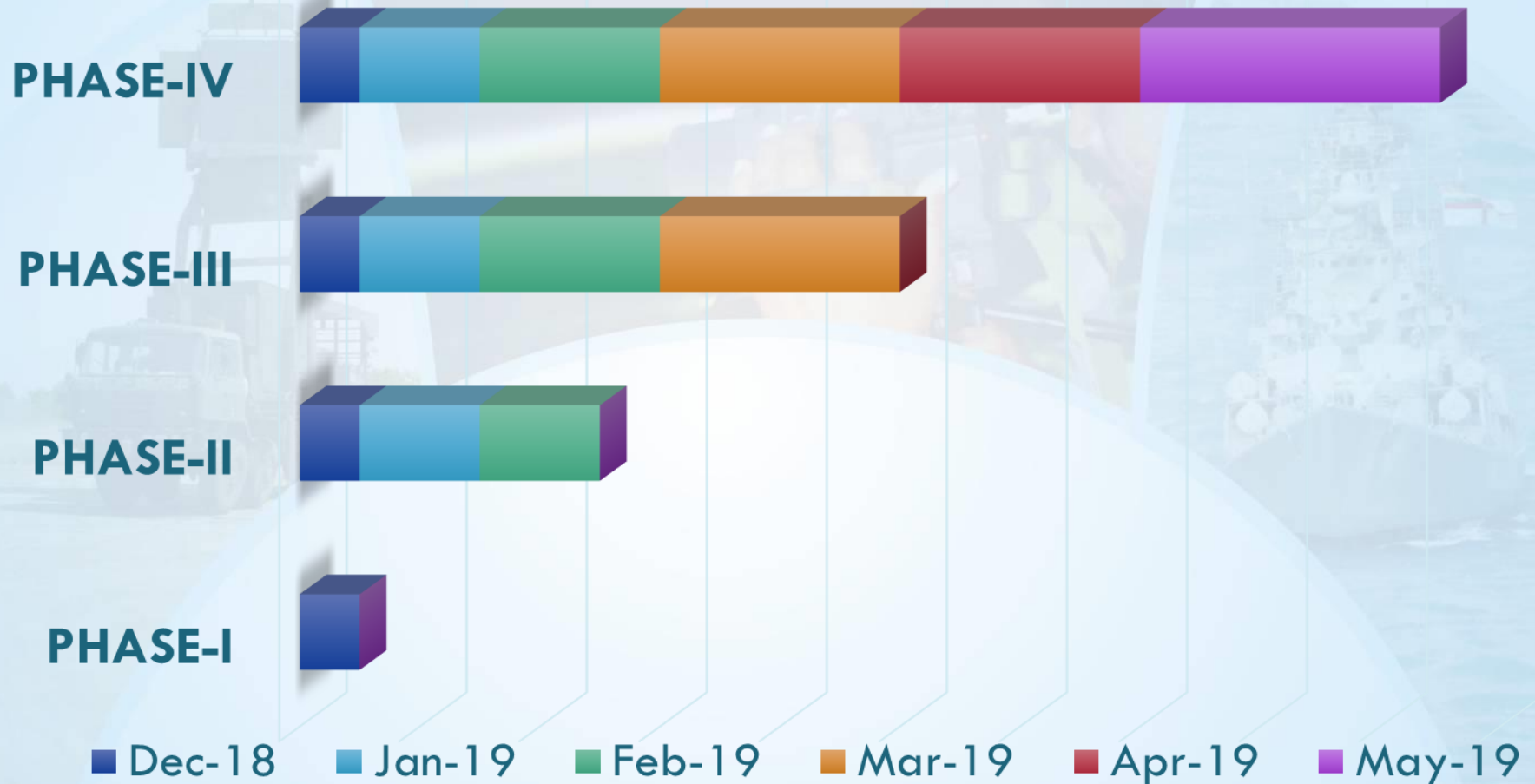- Evaluation of performance in **Coastal Surveillance** scenario.

CHALLENGES

- **Small target tracking** amidst sea clutter

- **Reduced false track initiation** in high clutter conditions

- **Faster track initiation** & better **maintenance**

- Attribute extraction for **target classification**

- **Deployable** code generation **without dependencies** on external libraries

APPROACH

Phase-I
- EVALUATING SFTT IN MATLAB WITH RECORDED DATA
- COMPARISION OF RESULT WITH RECORDED TRACK REPORTS

Phase-II
- USING MULTIPLE TRACK LINES
- USING COVARIANCE FUSION

Phase-III
- C/C++ CODE GENERATION
- WRAPPER FUNCTION FOR INTERFACE WITH SP CHAIN

Phase-IV
- CUSTOM ENHANCEMENTS IN C++ CODE
- KNOWLEDGE BASED TRACKING

DEVELOPMENT OF MULTI_TARGET_TRACKER FOR CSS - TIMELINE

PHASE-IV

PHASE-III

PHASE-II

PHASE-I

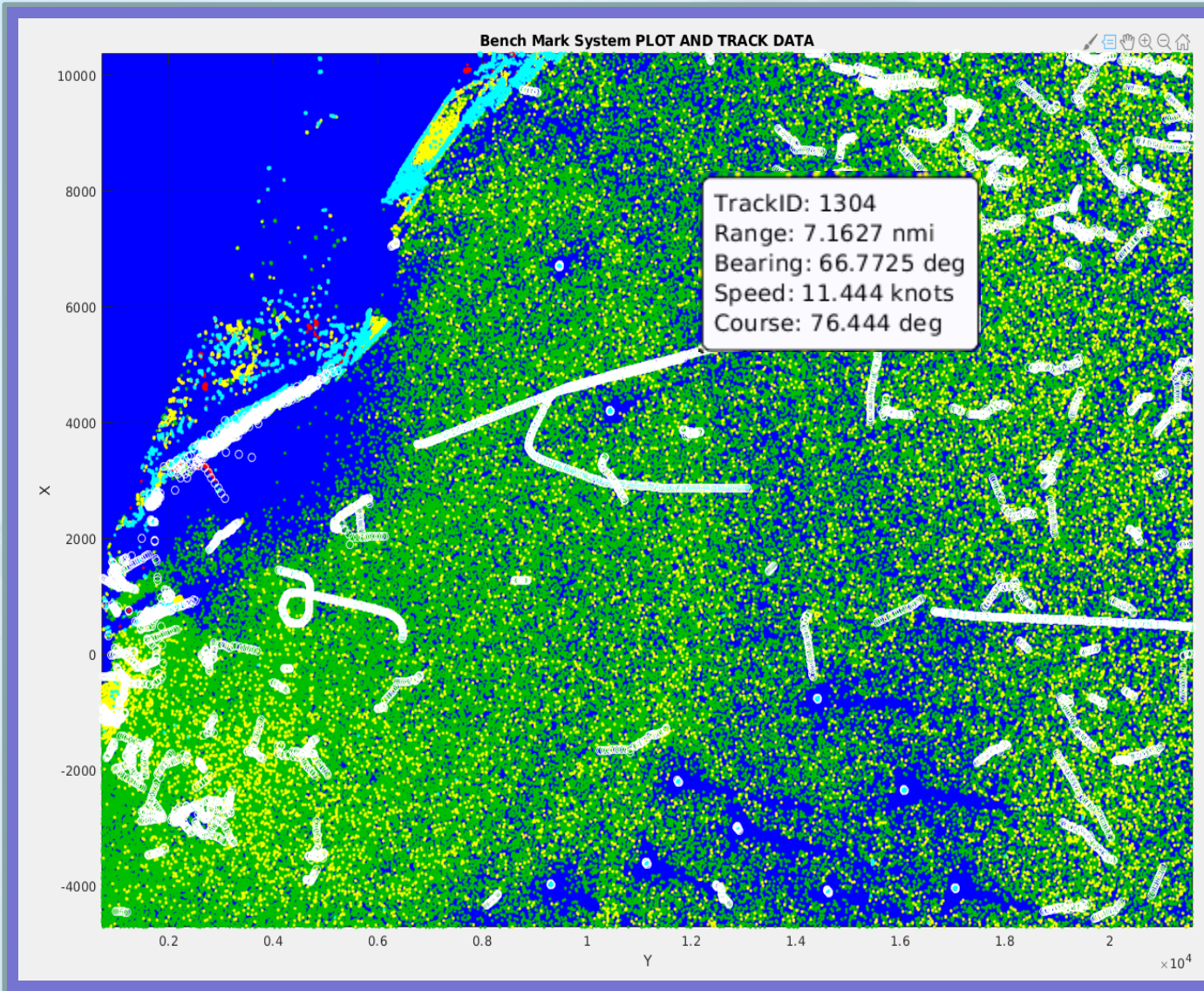Dec-18   Jan-19   Feb-19   Mar-19   Apr-19   May-19

DEVELOPMENT PHASE - I

➢ Evaluating performance of **trackerGNN** and **trackerTOMHT** modules in **Matlab-SFTT with field recorded data(detections).**

- Tuning tracker initiation , filter and maintenance parameters
- Evaluating performance w.r.t track kinematics accuracy and maintenance , by changing assignment methods, cost matrix computation, etc.

```
tracker =
trackerGNN('FilterInitializationFcn',...
@initcaekf,...
'MaxNumTracks',1000,...
'Assignment','Jonker-Volgenant',...
'AssignmentThreshold',50,...
'TrackLogic','History',...
'DeletionThreshold',[6 8], ...
'ConfirmationThreshold',[8 10] ,...
'HasCostMatrixInput',true);
```

```
tracker =
trackerTOMHT('FilterInitializationFcn',...
@custcaekf, ...
'ConfirmationThreshold', 25,...
'MaxNumSensors', 1, ...
'DeletionThreshold', -5, ...
'AssignmentThreshold',[0.5,0.7,1]*50,...
'MaxNumHypotheses', 2,...
'MaxNumTracks',1000,...
'MaxNumHypotheses',10,...
'NScanPruning','Hypothesis',...
'HasCostMatrixInput',true,...
'FalseAlarmRate',1e-5);
```
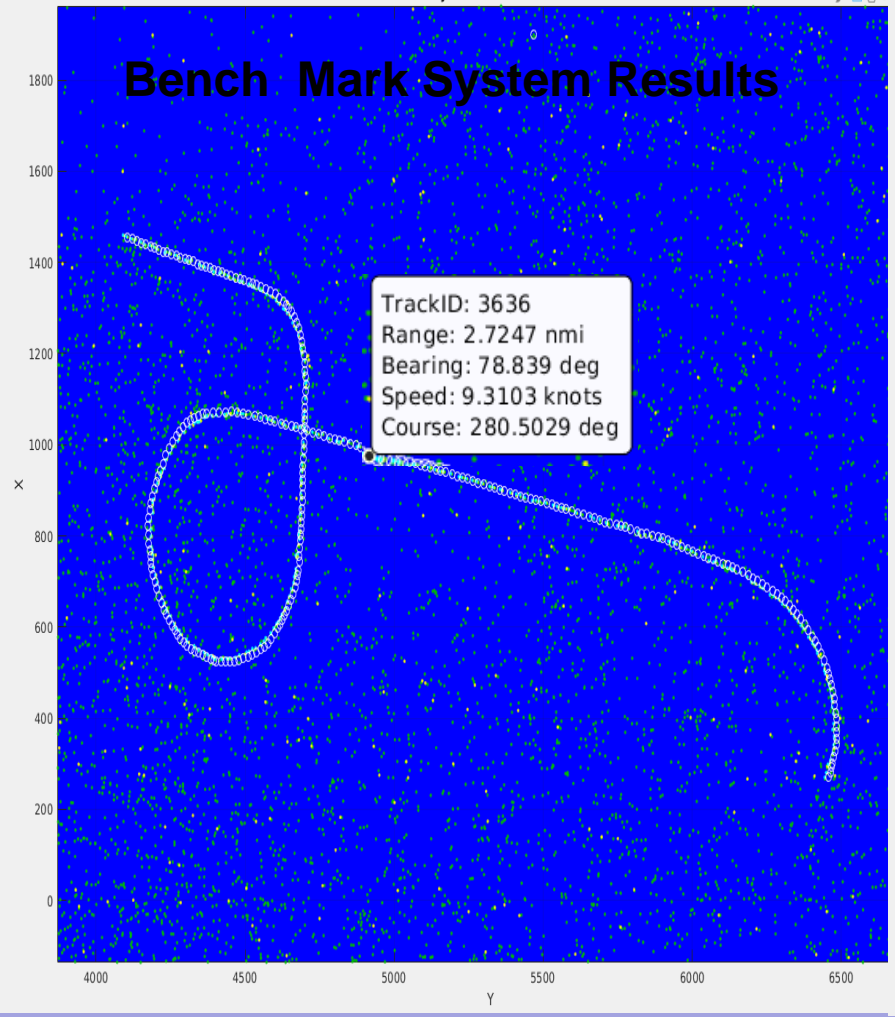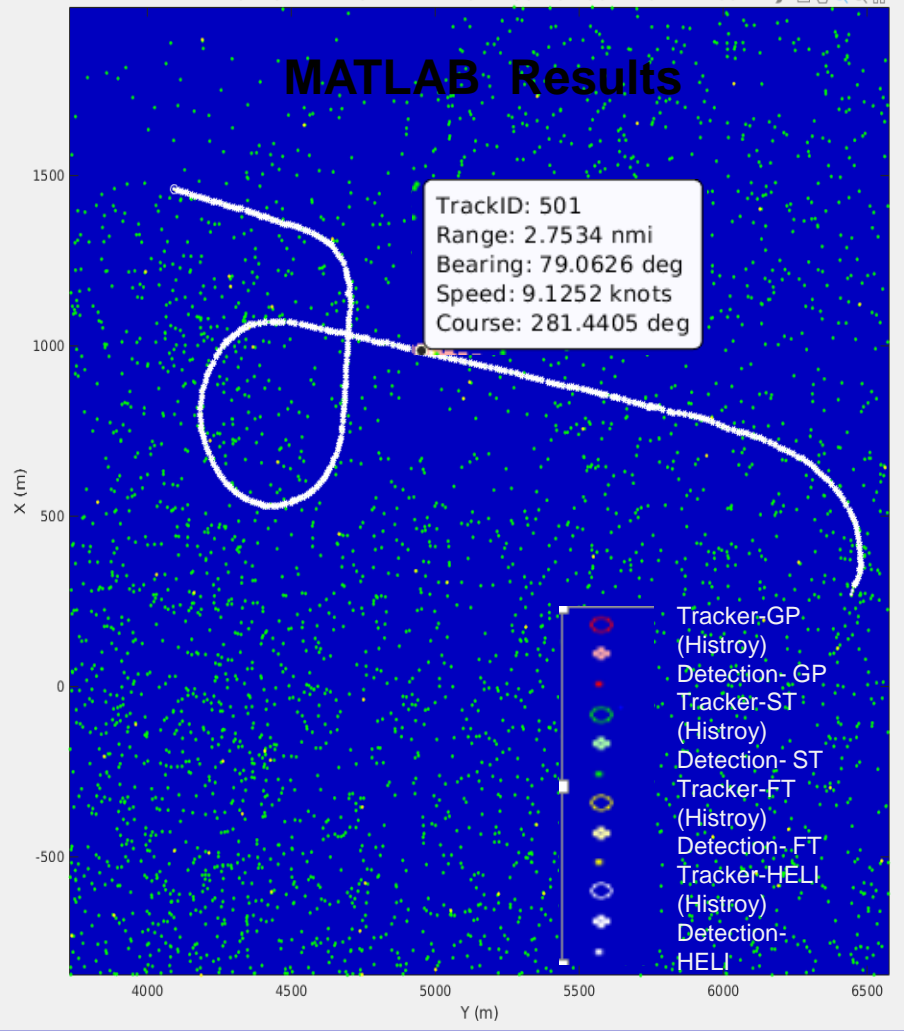
RECORDED DATA-295 SCANS

Bench Mark System PLOT AND TRACK DATA

TrackID: 1304
Range: 7.1627 nmi
Bearing: 66.7725 deg
Speed: 11.444 knots
Course: 76.444 deg

RADAR – POSITION
NAVIGATIONAL-TL PLOTS
SLOW-TL-PLOTS
FAST-TL-PLOTS
HELICOPTER-TL-PLOTS
TERMA- TRACKS

RESULTS USING MATLAB SFTT-GNN MULTIOBJECT TRACKER

RESULTS USING MATLAB SFTT-TOMHT MULTIOBJECT TRACKER
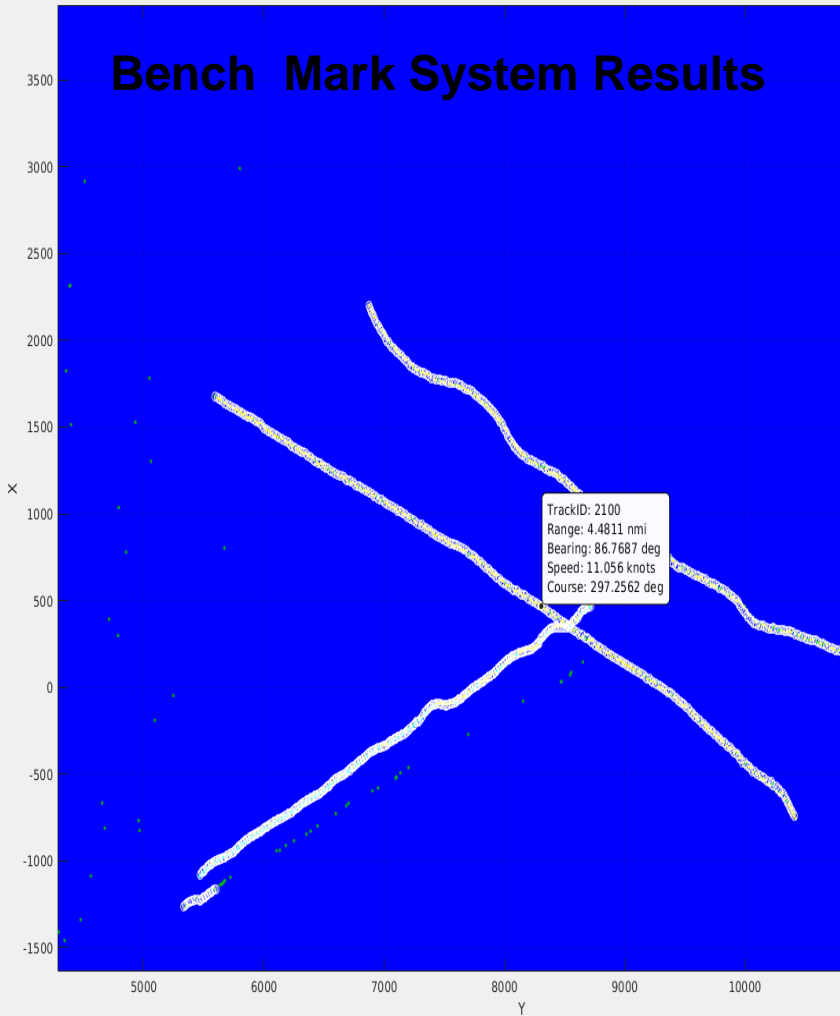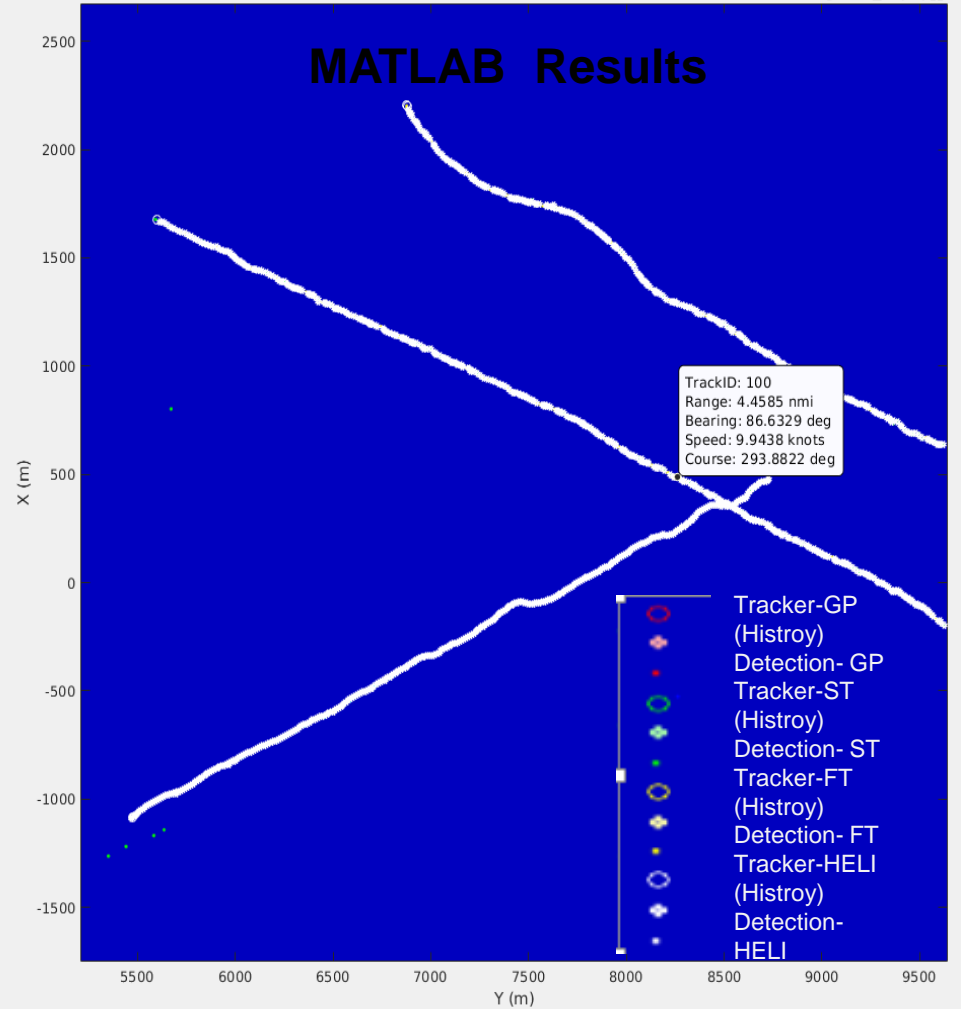
DEVELOPMENT PHASE - II

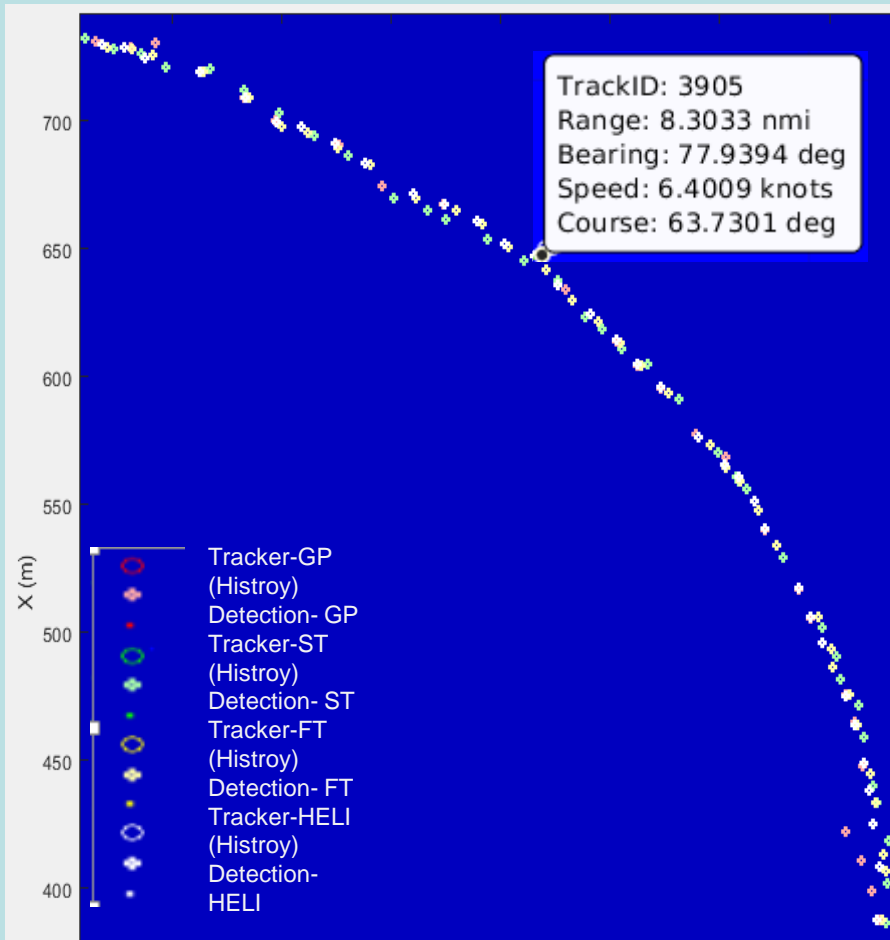➢ Use of **Multiple-Track Lines** for load reduction

****REDUCE LOAD ON EACH LINE
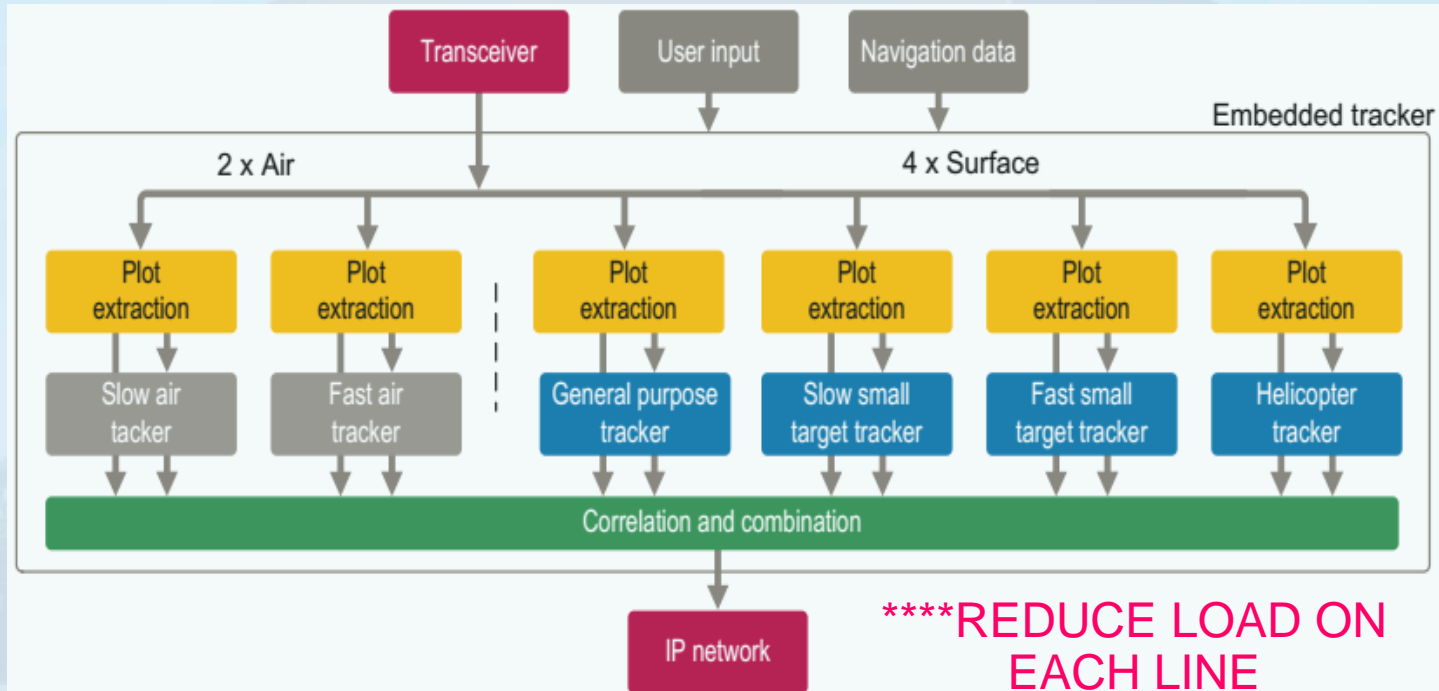
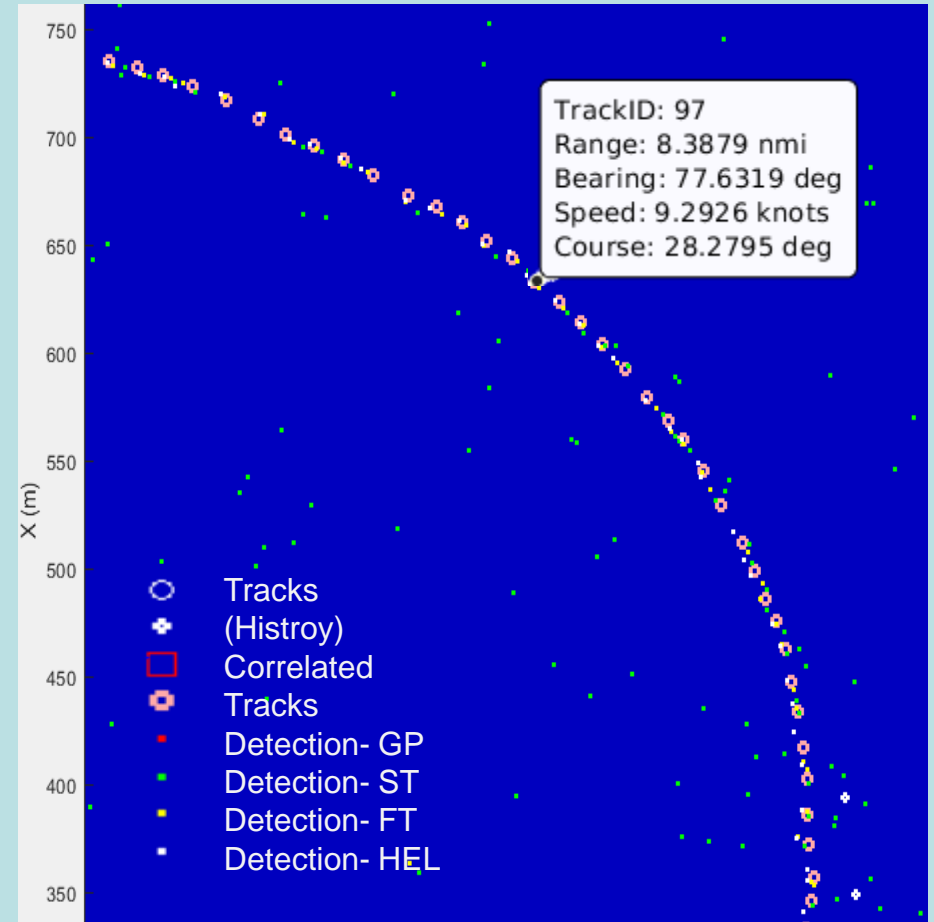➢ Use of **covariance fusion** for correlating tracks from all track lines

```
X(:,t) = confTracks{h}(fusionIDS{cTrk}(t)).State;
P(:,:,t) = confTracks{h}(fusionIDS{cTrk}(t)).StateCovariance;
[Xcorr,Pcorr] = fusexcov(X,P);  %Cross-covariance fusion
```

RESULTS (CONTD.)

WITHOUT COVARIANCE FUSION


TrackID: 3905
Range: 8.3033 nmi
Bearing: 77.9394 deg
Speed: 6.4009 knots
Course: 63.7301 deg

Tracker-GP (Histroy)
Detection- GP
Tracker-ST (Histroy)
Detection- ST
Tracker-FT (Histroy)
Detection- FT
Tracker-HELI (Histroy)
Detection- HELI

WITH COVARIANCE FUSION

TrackID: 97
Range: 8.3879 nmi
Bearing: 77.6319 deg
Speed: 9.2926 knots
Course: 28.2795 deg

○ Tracks (Histroy)
✛ (Histroy)
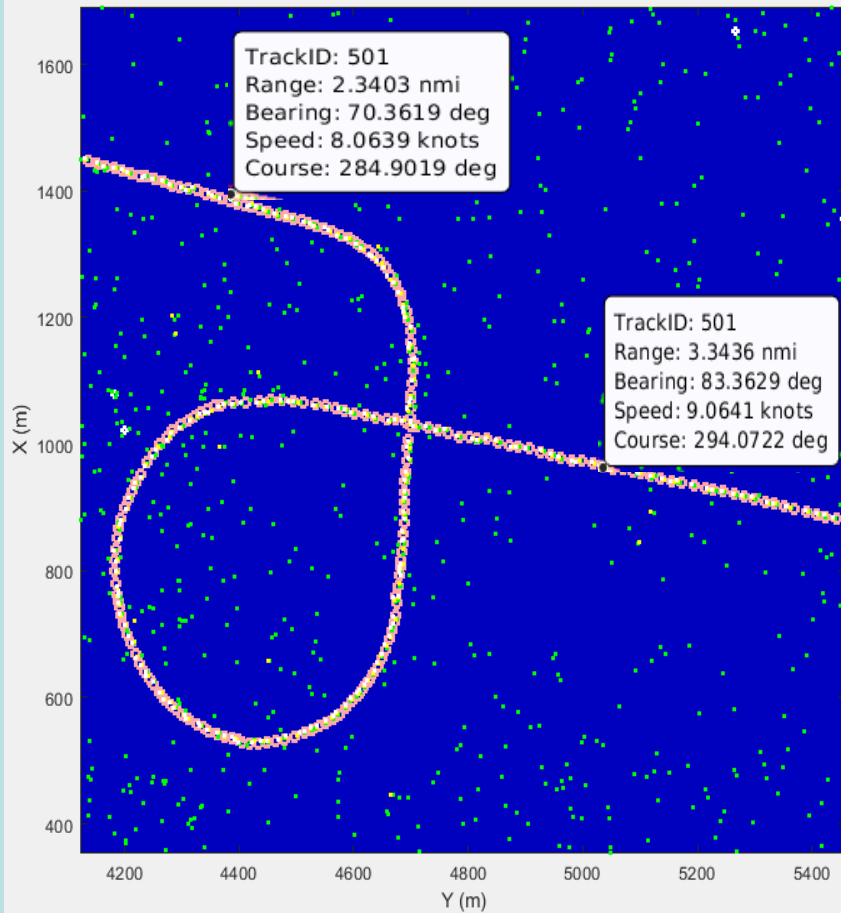☐ Correlated
⊛ Tracks
. Detection- GP
. Detection- ST
. Detection- FT
. Detection- HEL

TRACK MAINTENANCE FOR TWO DATASETS

Track is maintained even after long run (Track ID is maintained)
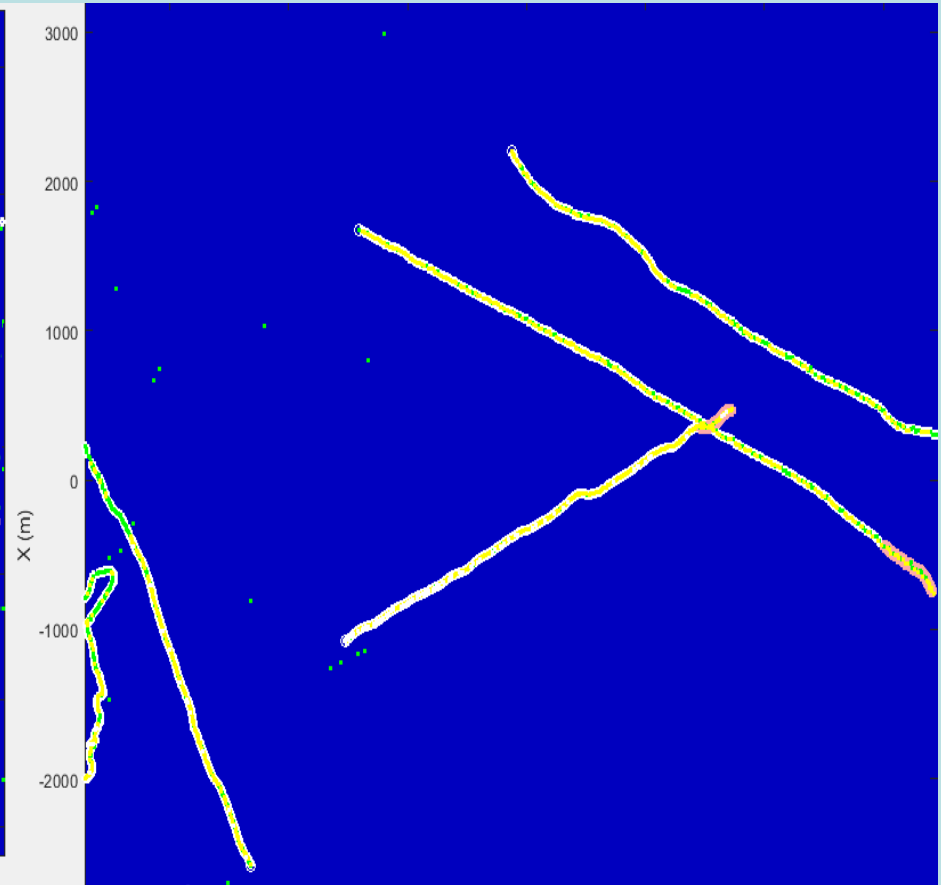
DEVELOPMENT PHASE - III

➢ **Deployable C++ code** generation using **MATLAB Coder**

```matlab
load detectiondata.mat
compInputs  = {detections simTime};
tracker_kernel(compInputs{1},simTime);
codegen tracker_kernel -args compInputs;

function [confirmedTracks, numTracks,~] =
tracker_kernel.m(detections,time)
    persistent tracker
    if isempty(tracker)
        tracker = trackerTOMHT('FilterInitializationFcn',
    @initcaEKF,...
        'MaxNumHypotheses', 5,...
        'MaxNumTracks',1000,...
        'MaxNumSensors',1,...
        'NScanPruning','Hypothesis');
    end
    [confirmedTracks,~,~,information] = tracker(detections,time);
    numTracks = tracker.NumTracks;
end
```

➢ **Wrapper function** is developed for **interfacing** the C++ Code with the signal processer chain .

DEVELOPMENT PHASE - IV

➢ **Custom Enhancements** in C++ code for introducing **Knowledge based tracking methods such as:**

    ❑   Utilization of **zones** and **map** information

    ❑   Using plot attributes such **range spread** ,**azimuth spread** and **plot amplitude** for association

    ❑   Adaptation in Confirmation , Association and Deletion Thresholds w.r.t variable clutter density

    ❑   Optimizing memory allocations and introducing parallel processing architecture for multiple track lines with correlation.

PLOT ATTRIBUTES
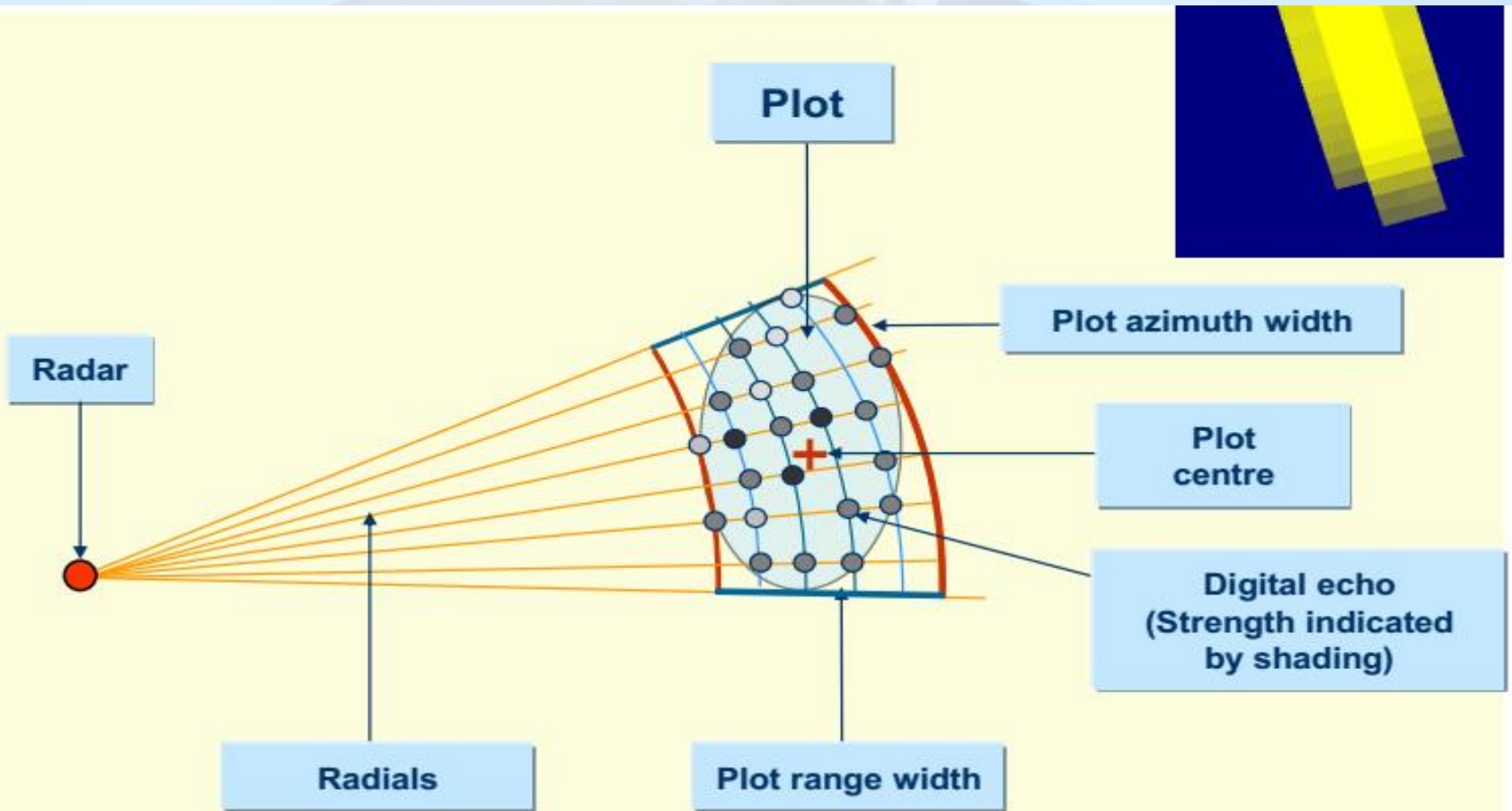
STATUSF

BENEFITS OF USING MATLAB

❖ Object Oriented Approach

❖ The ability to auto-generate C code, using MATLAB Coder

❖ Wide variety of readily available packages and Toolbox

❖ Complex Tracker Algorithms are available out of the box.

❖ Easy to prove and evaluate the concepts

❖ Saves money and time

❖ User friendly

KEY TAKE AWAYS

❑ Using **MATLAB SFTT** for evaluating **Multi-Target Tracker** and **Estimation Filters**

❑ Using **cross-covariance fusion** for track-to-track **correlation**

❑ Using **MATLAB CODER** for **deployable** C++ code generation

❑ Employing **Knowledge based tracking** methods for improving track maintenance

LOOKING FORWARD FOR

- ❑   **Multi-sensor data fusion** (viz. Radar, AIS etc)

- ❑   **Attribute based Tracking** and data fusion

- ❑   **Target classification**

- ❑   **Deployable code generation** for these modules

" *Looking forward to a long and fruitful association* "

THANK YOU