



## **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**

## 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...

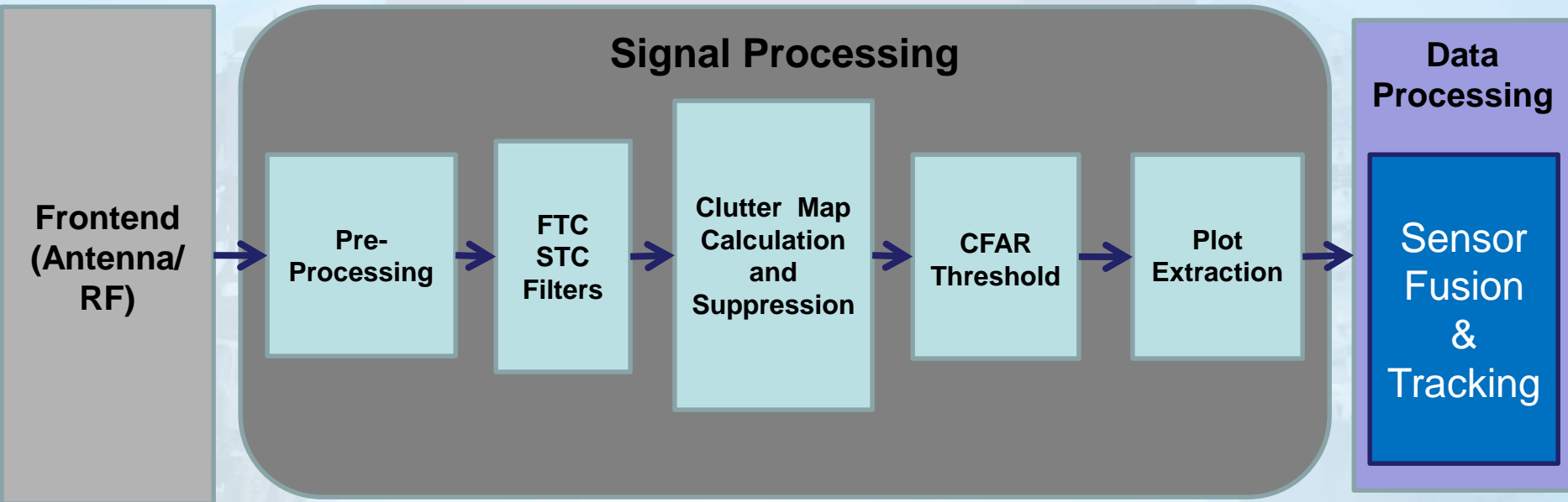
## ABOUT BEL

- ❖ 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



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

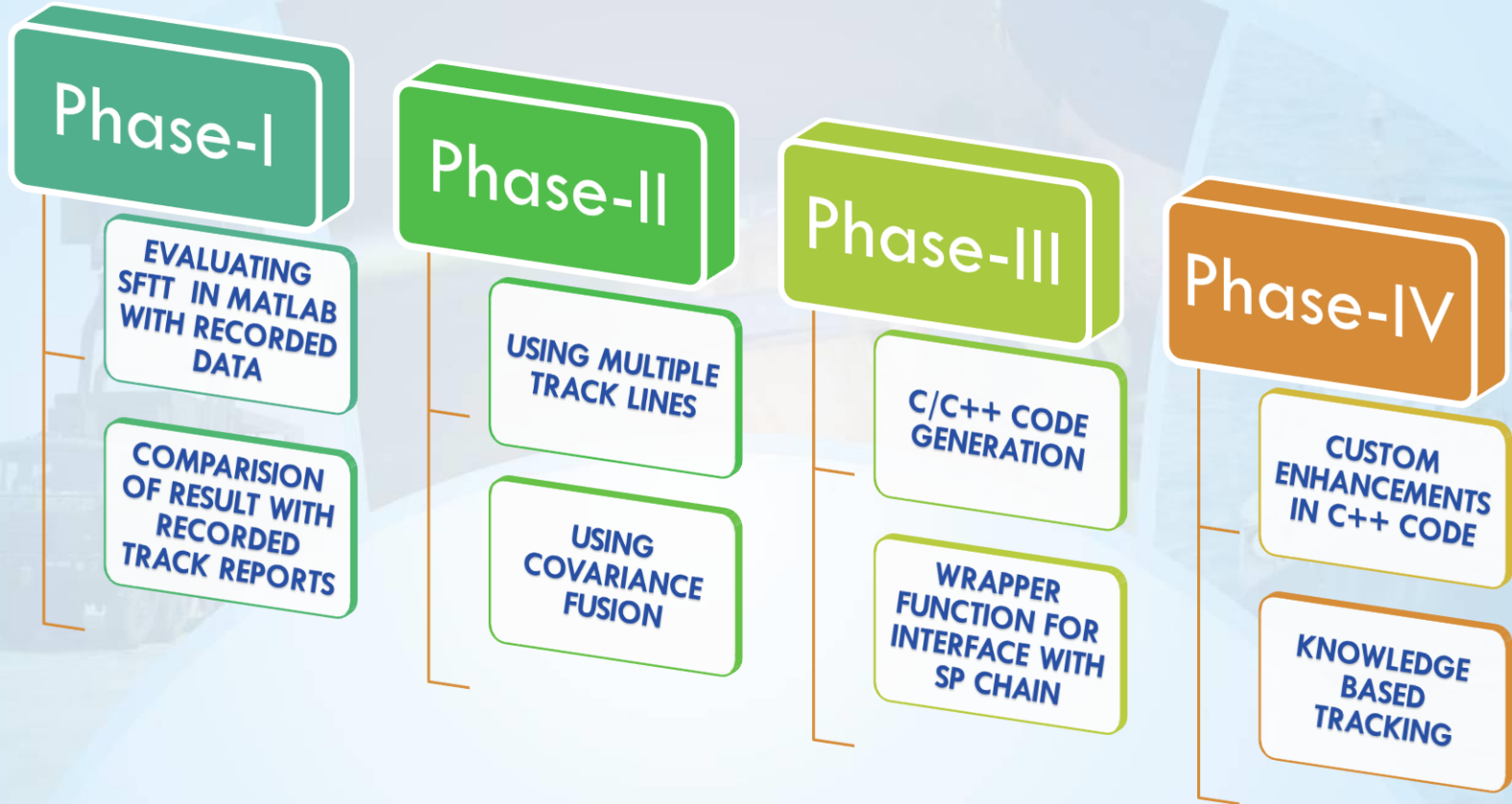
## REQUIREMENTS

- ❑ Development of **field deployable multi-target tracking module** using **Sensor Fusion Tracking Toolbox** and **MATLAB** coder.
- ❑ Evaluation of performance in **Coastal Surveillance** scenario.

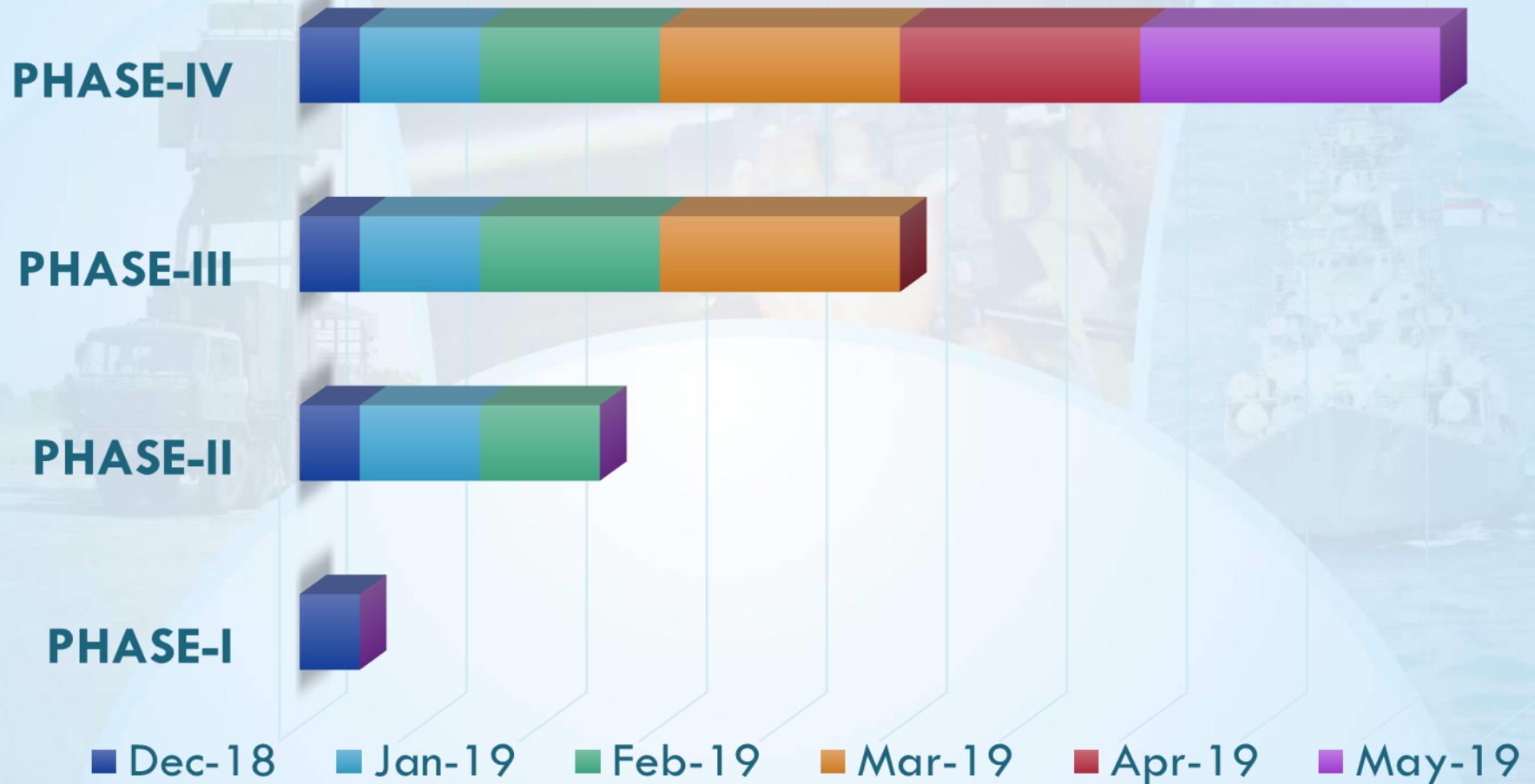
## CHALLENGES

- ❑ **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



## DEVELOPMENT OF MULTI\_TARGET\_TRACKER FOR CSS - TIMELINE



## DEVELOPMENT PHASE - I

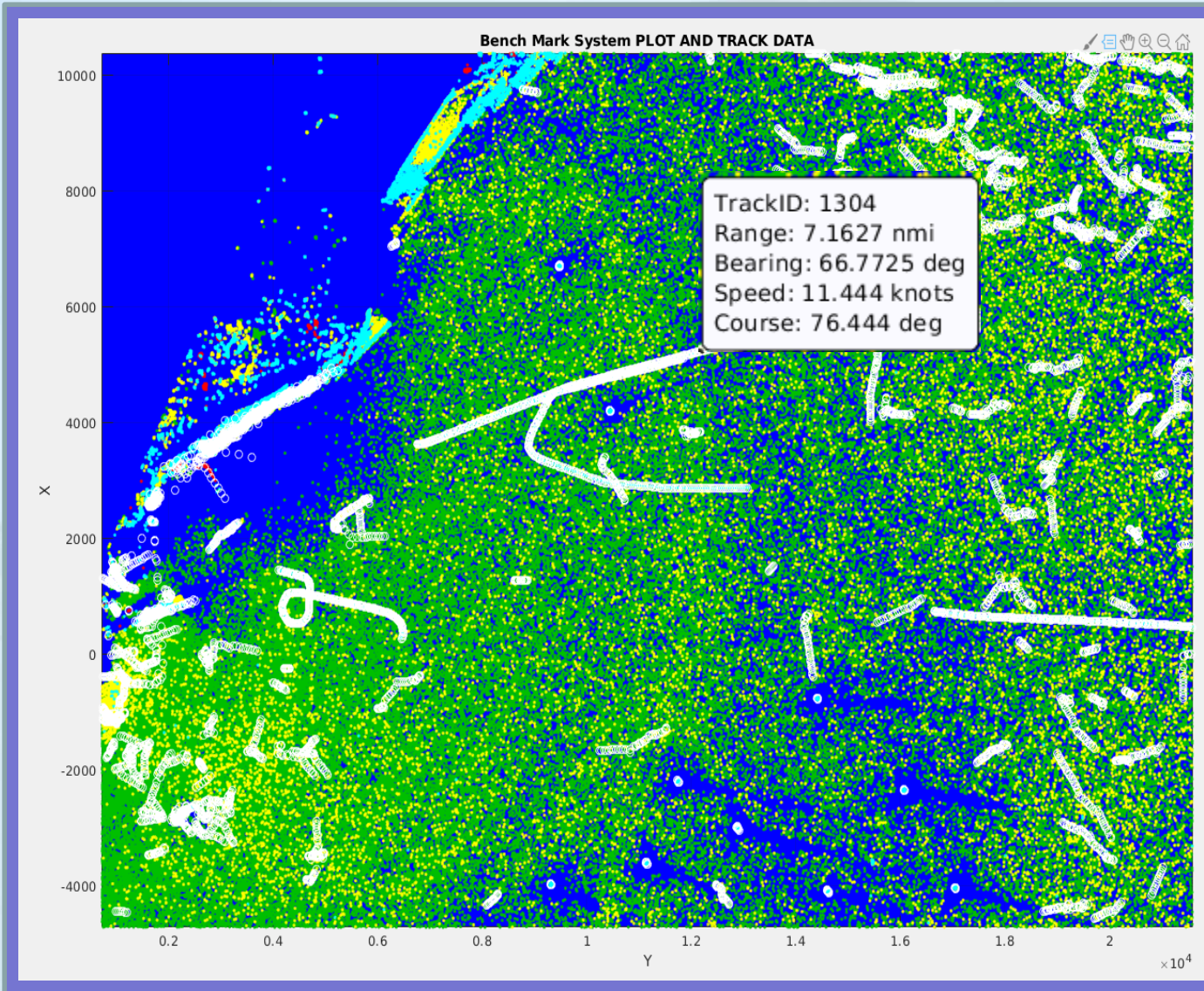
- Evaluating performance of **trackerGNN** and **trackerTOMHT** modules in **Matlab-SFTT** with **field recorded data(detections)**.
  - Tuning tracker initiation , filter and maintenance parameters
  - Evaluating performance w.r.t track kinematics accuracy and maintenance , by changing assignment methods, 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);
```



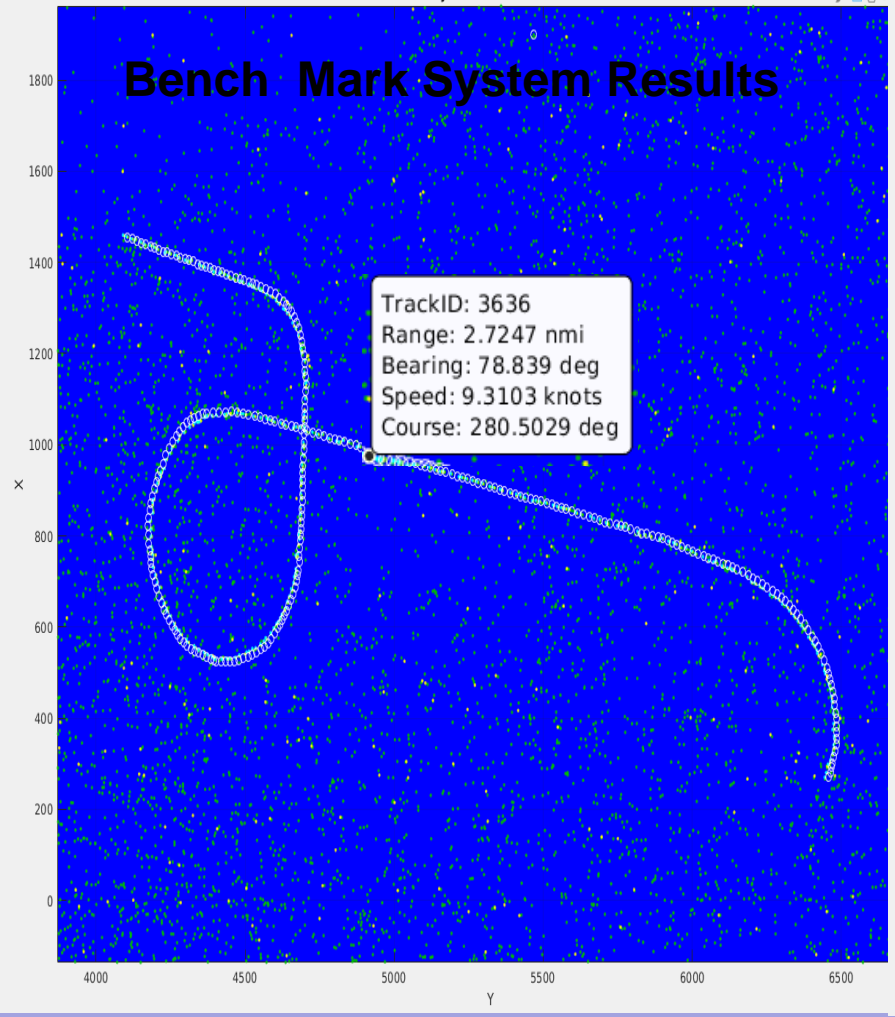
# RECORDED DATA-295 SCANS



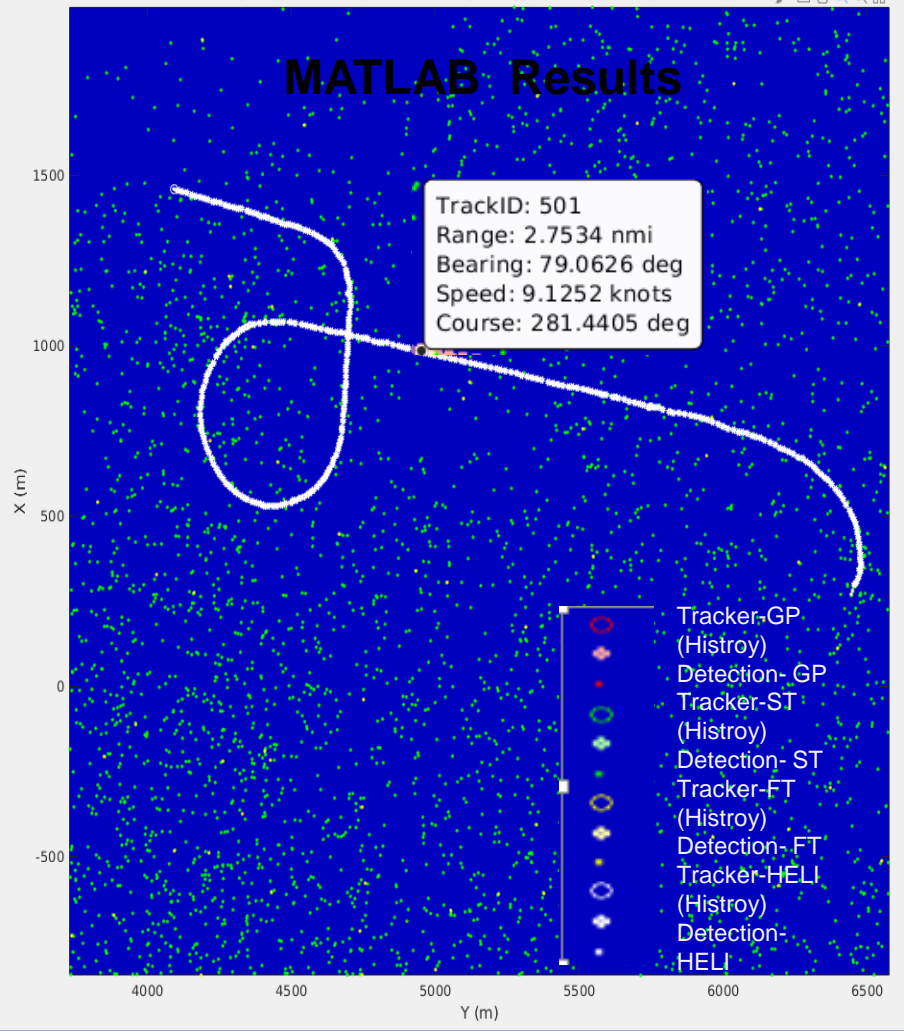
- RADAR – POSITION
- NAVIGATIONAL-TL PLOTS
- SLOW-TL-PLOTS
- FAST-TL-PLOTS
- HELICOPTER-TL-PLOTS
- TERMA- TRACKS

# RESULTS USING MATLAB SFTT-GNN MULTIOBJECT TRACKER

Bench Mark System PLOT AND TRACK DATA



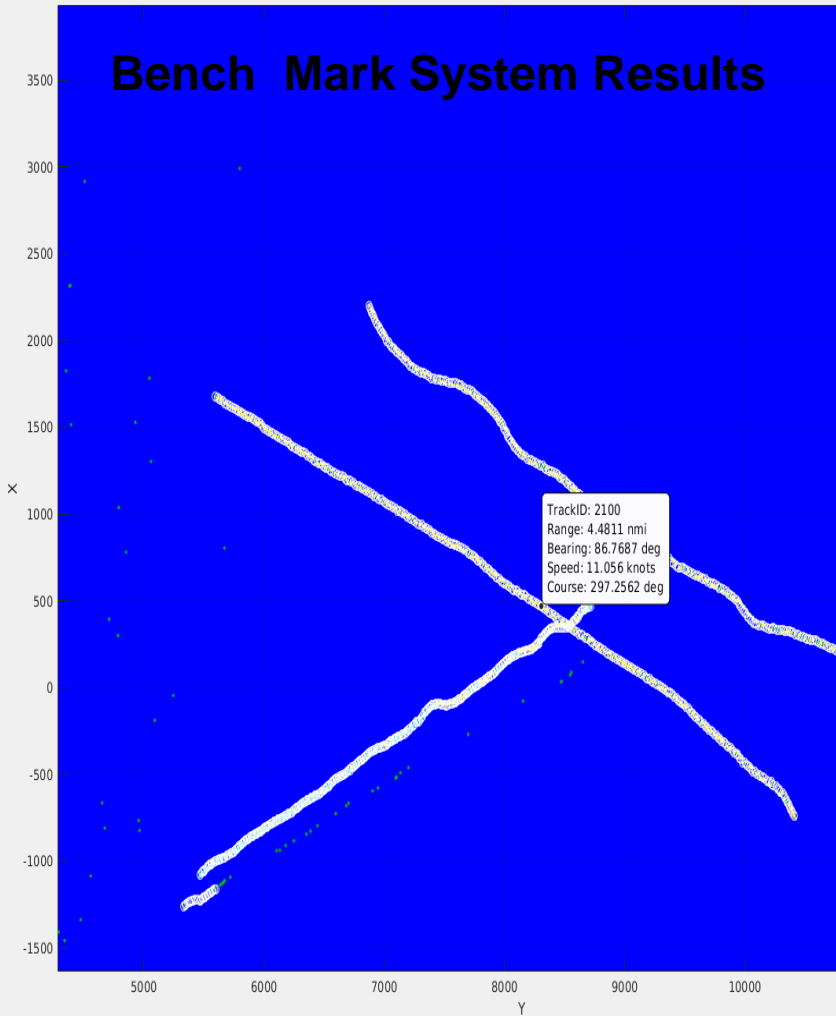
MULTICHANNEL- GNN-EKF-TRACKER OUTPUT. SCAN-NUMBER:295



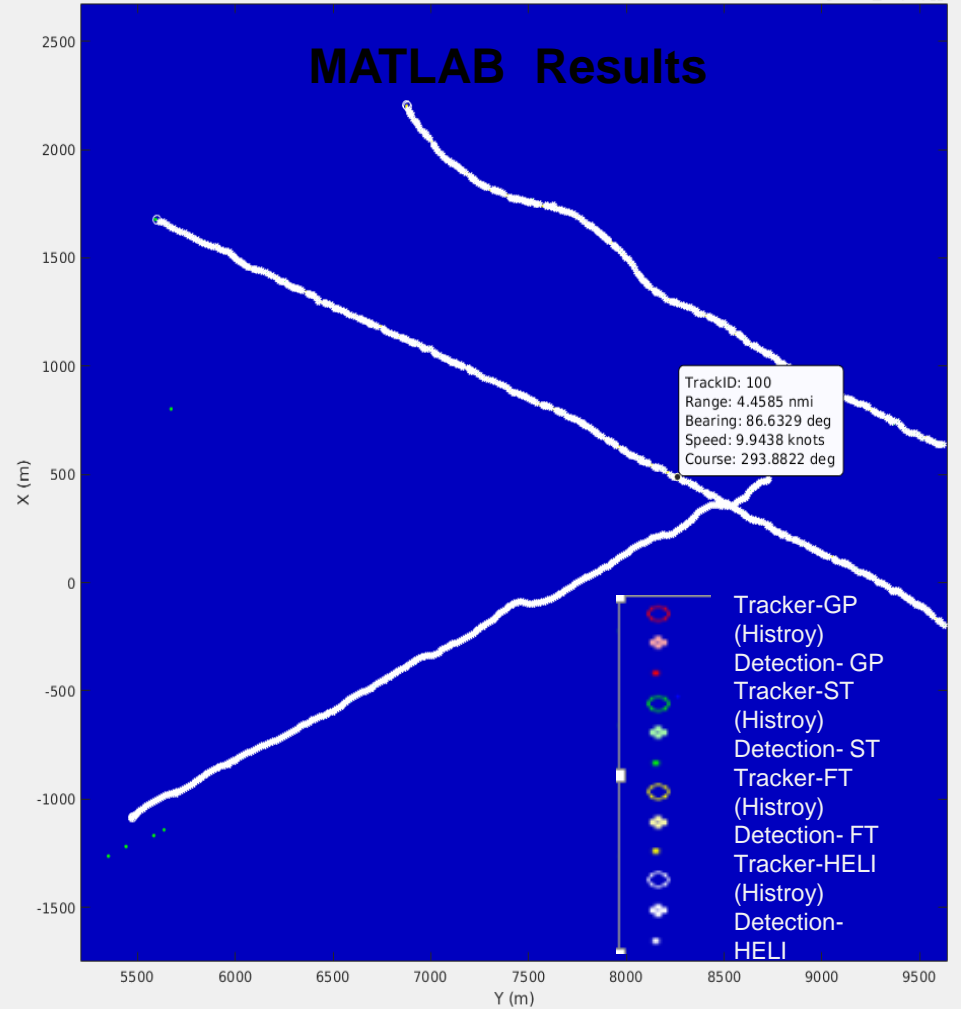


# RESULTS USING MATLAB SFTT-TOMHT MULTIOBJECT TRACKER

TERMA PLOT AND TRACK DATA

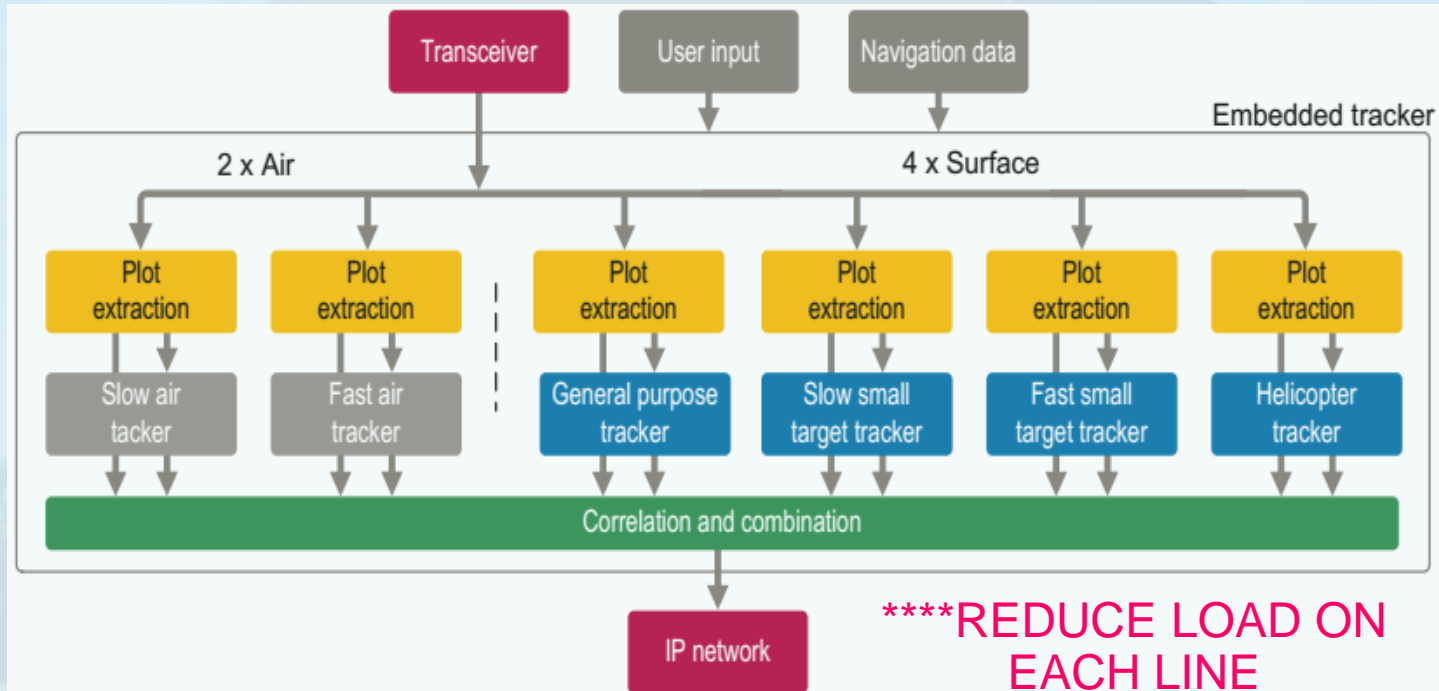


MULTICHANNEL-TOMHT-customcaEKF-TRACKER CORRELATED OUTPUT. SCAN-NUM



## DEVELOPMENT PHASE - II

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



- Use of **covariance fusion** for correlating tracks from all track lines

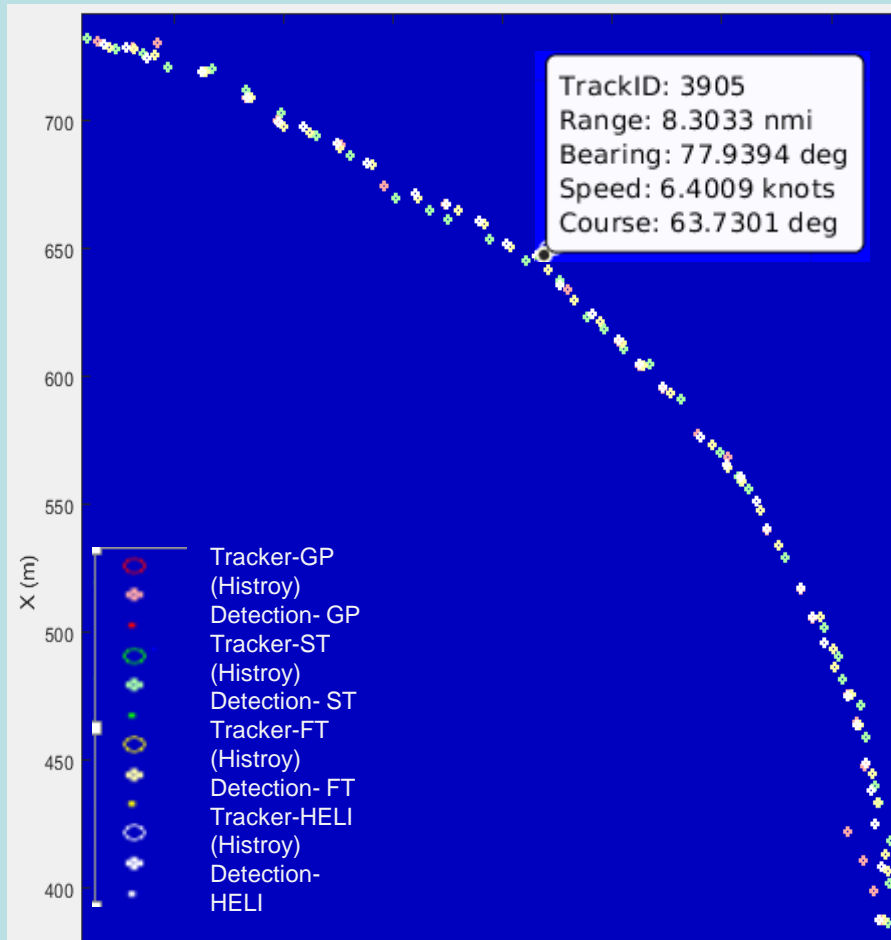
```

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

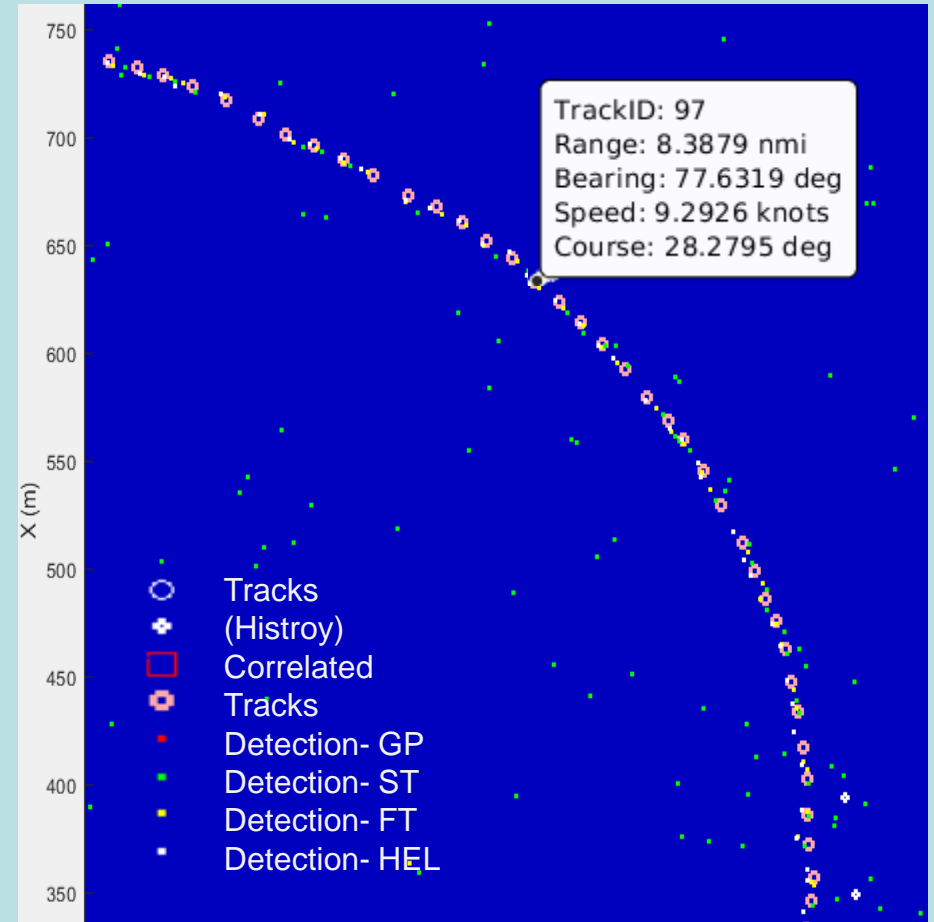


## RESULTS (CONTD.)

### WITHOUT COVARIANCE FUSION

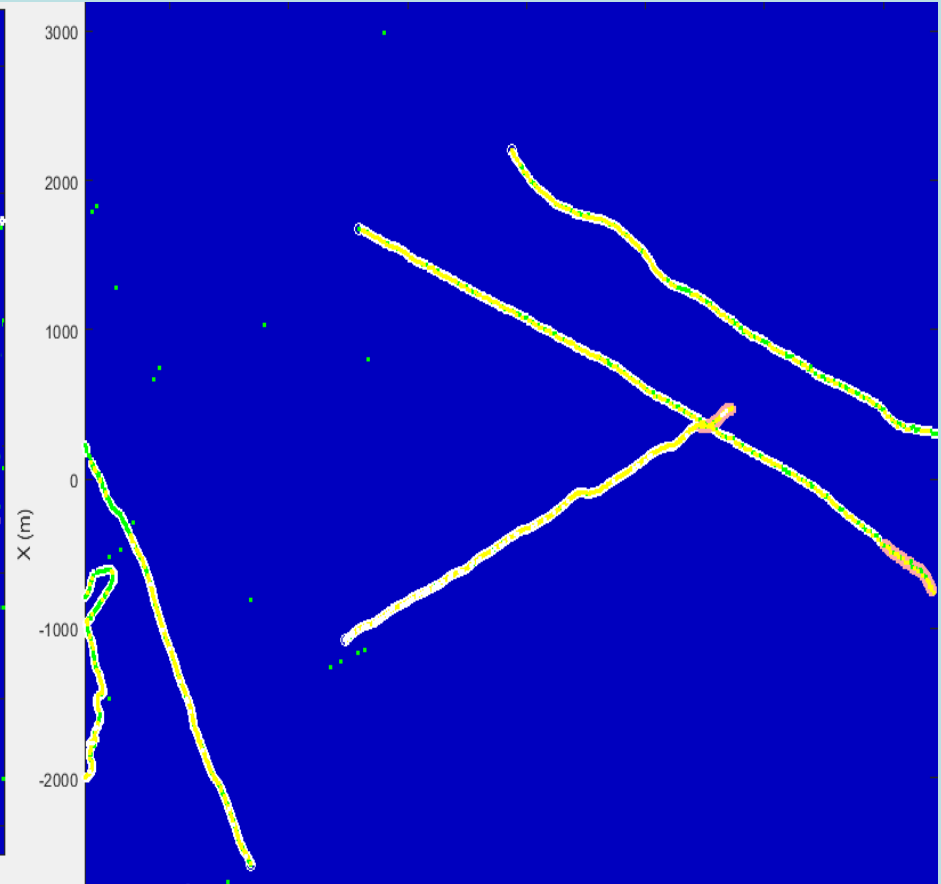
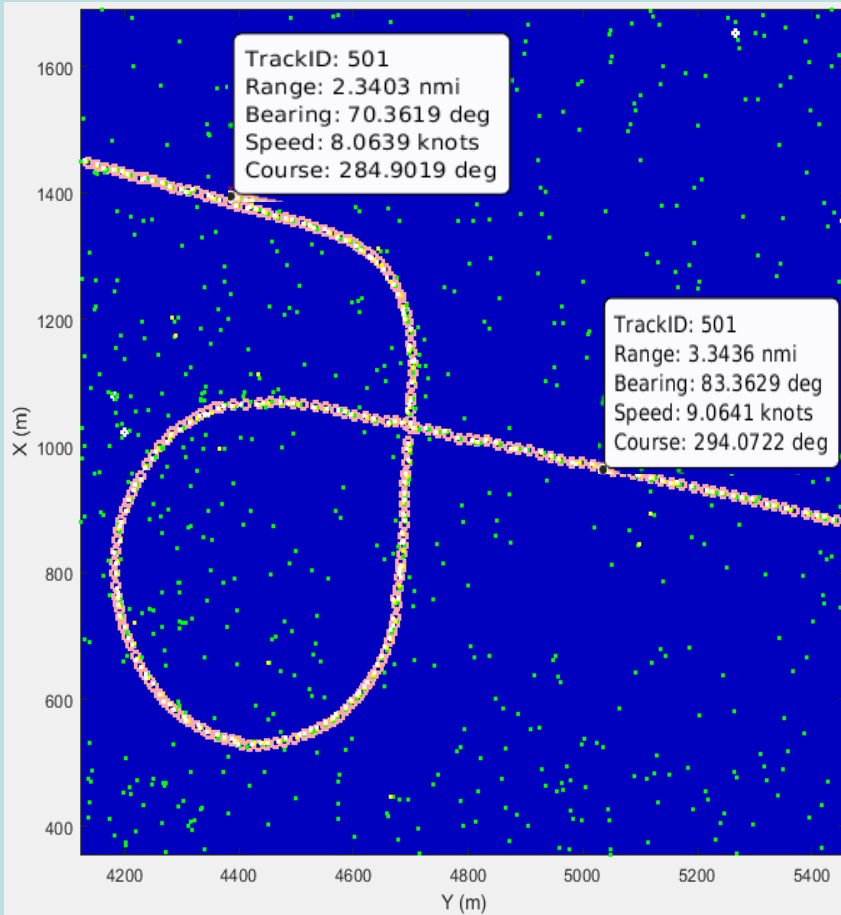


### WITH COVARIANCE FUSION



## 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

```
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
```

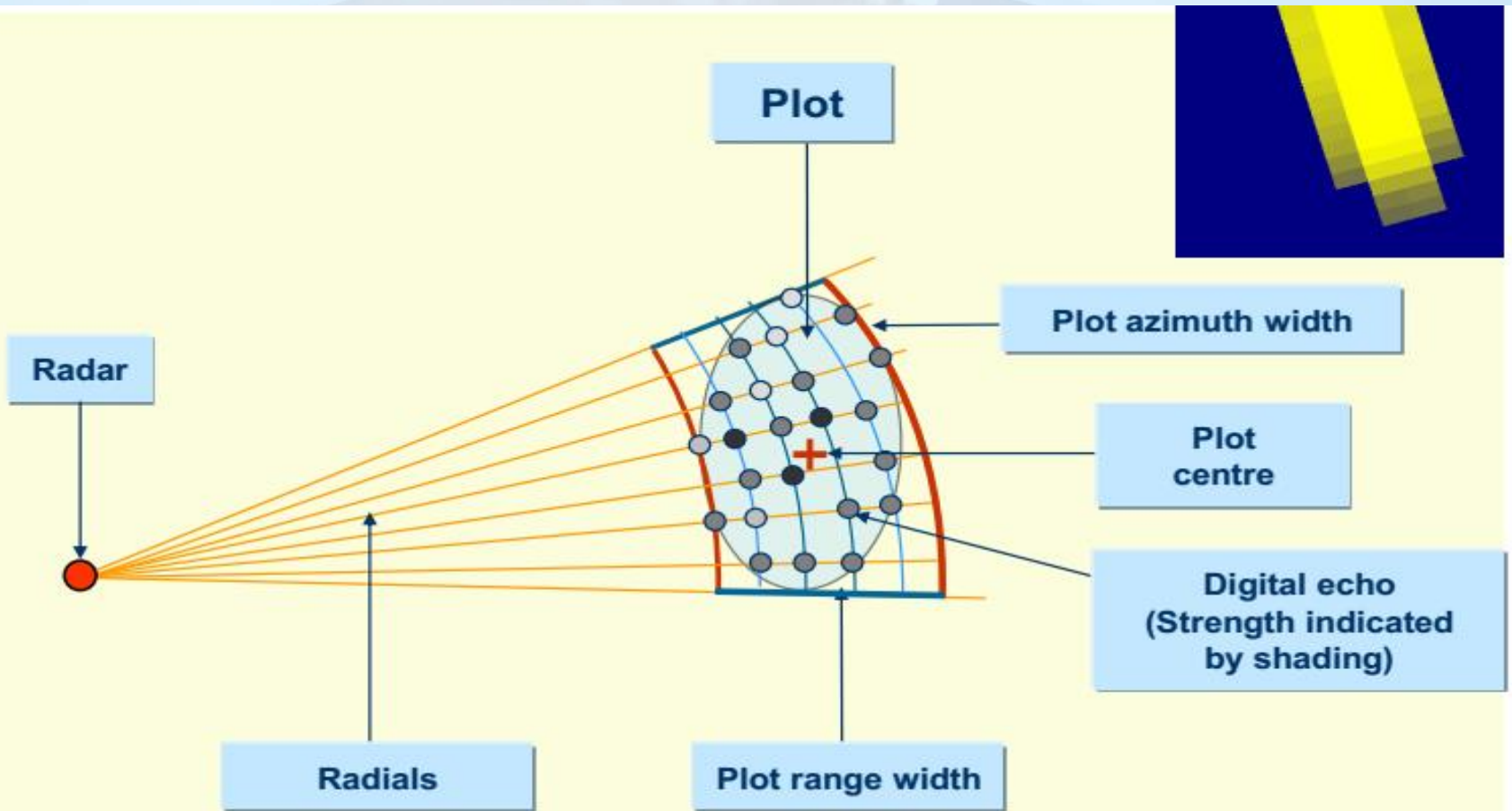
➤ Wrapper function is developed for **interfacing** the C++ Code with the signal processor chain .

## DEVELOPMENT PHASE - IV

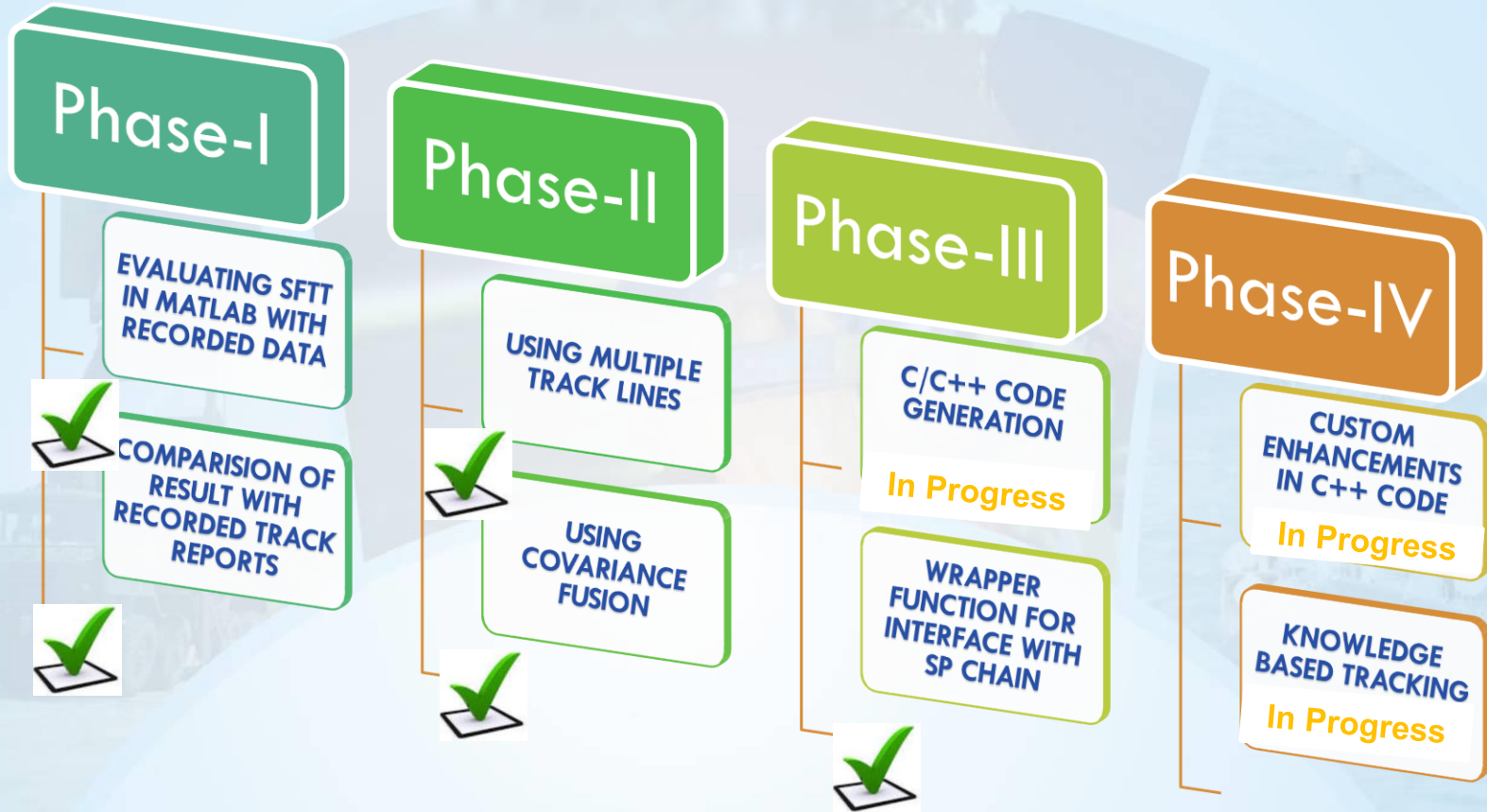
- **Custom Enhancements** in C++ code for introducing **Knowledge based tracking methods** such as:
  - ❑ Utilization of **zones** and **map** information
  - ❑ Using plot attributes such **range spread** ,**azimuth spread** and **plot amplitude** for association
  - ❑ Adaptation in Confirmation , Association 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**

