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

Signal Processing

Frontend (Antenna/RF) ➔ Pre-Processing ➔ FTC/STC Filters ➔ Clutter Map Calculation and Suppression ➔ CFAR Threshold ➔ Plot Extraction ➔ Data Processing

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

Sensor Fusion & Tracking
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
- Comparison 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.

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

```matlab
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
RESULTS USING MATLAB SFTT-GNN MULTIOBJECT TRACKER

Bench Mark System Results

TrackID: 3636
Range: 2.7247 nmi
Bearing: 78.839 deg
Speed: 9.3103 knots
Course: 280.5029 deg

MATLAB Results

TrackID: 501
Range: 2.7534 nmi
Bearing: 79.0626 deg
Speed: 9.1252 knots
Course: 281.4405 deg
RESULTS USING MATLAB SFTT-TOMHT MULTIOBJECT TRACKER

Bench Mark System Results

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

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

\[
X(:,t) = \text{confTracks}_h(\text{fusionIDS}_c\text{Trk}_t).\text{State}; \\
P(:, :, t) = \text{confTracks}_h(\text{fusionIDS}_c\text{Trk}_t).\text{StateCovariance}; \\
[X_{corr}, P_{corr}] = \text{fusexcov}(X, P); \quad \%\text{Cross-covariance fusion}
\]
RESULTS (CONTD.)

WITHOUT COVARIANCE FUSION

WITH COVARIANCE FUSION

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

TrackID: 97
Range: 8.3879 nmi
Bearing: 77.6319 deg
Speed: 9.2926 knots
Course: 28.2795 deg
TRACK MAINTENANCE FOR TWO DATASETS

Track is maintained even after long run (Track ID is maintained)

TrackID: 501
Range: 2.3403 nmi
Bearing: 70.3619 deg
Speed: 8.0639 knots
Course: 284.9019 deg

TrackID: 501
Range: 3.3436 nmi
Bearing: 83.3629 deg
Speed: 9.0641 knots
Course: 294.0722 deg
**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 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 as 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

- Radar
- Plot
- Plot azimuth width
- Plot centre
- Digital echo (Strength indicated by shading)
- Radials
- Plot range width
STATUSF

Phase-I
- Evaluating SFTT in MATLAB with recorded data
- Comparison 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

In Progress
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