Automatic Drone Detection with Echodyne MESA-DAA Radar

Skyler Martens, Huy Nguyen, Daniel Park, Bill(Jianfeng) Huang

Department of Electrical Engineering, University of Washington

Abstract

In collaboration with Echodyne, our team's research focused on addressing the challenge of detecting and tracking small drones in a dynamic, ever-changing airspace. In particular, our team focused on two primary areas: applying a multi-detection tracking system to predict the trajectory of objects in a cluttered environment and developing a graphical user interface (GUI) with the ability to plot relevant range-velocity, detection, and track data in real time. Our team's final design leverages Echodyne's state-of-the-art MESA-DAA radar, the C programming language to provide a means of quickly processing large quantities of data, and Java to generate an intuitive graphical interface. Our research has significant implications for providing the first low-cost and easy-to-understand method for detecting and monitoring the location of drones.

System Operation Overview

1. User specifies the radar's field of view by setting the azimuth and elevation scanning ranges.
2. The radar scans the field of view, returning range-velocity and detection data.
3. Data processing occurs, creating tracks and correctly plotting detection or range-velocity data.
4. Range-velocity map, detection, and track data are plotted in real time.

Displaying Raw Radar Data

Our system attempts to intuitively and accurately display range-velocity (RV) and detection data by applying the following transformations to the data generated by the MESA-DAA Radar:

- Apply Mercator projection and coordinate transforms to correctly scale and plot data, regardless of user location and radar orientation.
- Compress the acquired RV map by selecting the maximum signal returned by objects with positive and negative radial velocities at a given location. Negative radial velocity returns are plotted in blue and positive radial velocity returns in red.
- Update RV map and detection plots as new data is acquired.

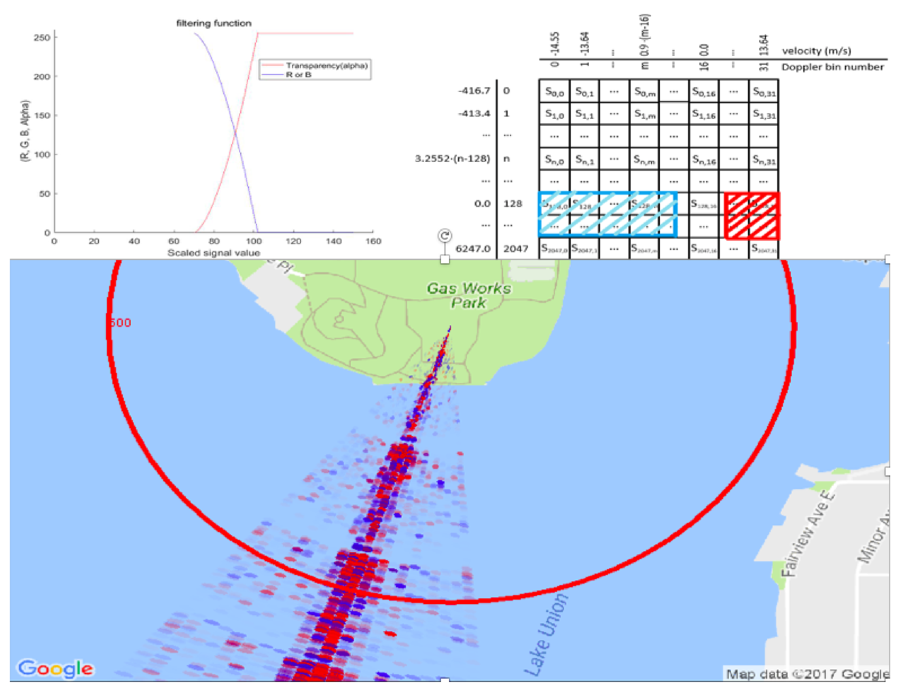

Figure 1: Process of creating and plotting a range-velocity map.

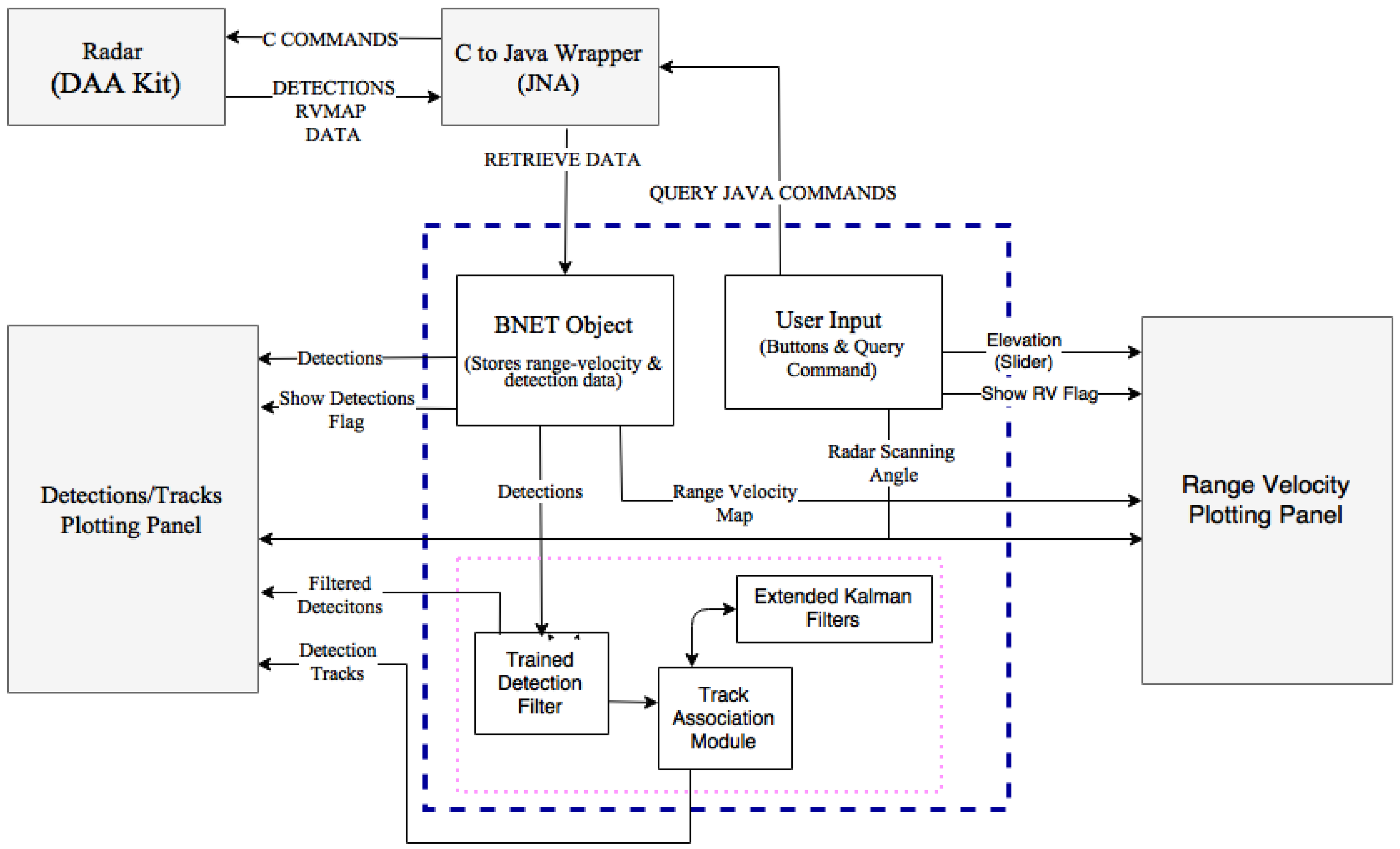

Figure 2: System Diagram of Automatic Drone Detection Module

Multi-Detection Tracking System

System Requirements:

- Must estimate the current location of an object based on past measurements made in spherical coordinates.
- Must be able to handle measurements with limited precision.
- Must be able to track multiple objects at the same time.
- Must correctly associate detections with tracks in a cluttered environment.

Solution:

- Filter out homogeneous detections that are likely caused by environmental clutter such as trees.
- Manage a list of Extended Kalman Filters (EKFs) to concurrently track multiple objects in a non-linear manner.
- Associate a detection with an EKF based on the detection's Mahalanobis distance from the predicted track location. The formula used to calculate the Mahalanobis distance is:

$$D = \sqrt{(x_1 - u_1)^T S_1^{-1} (x_1 - u_1)} \tag{1}$$

where x_1 represents the location of the detection, u_1 represents the predicted location of the track, and S_1 represents the covariance matrix of the track.

- If a detection is not associated with an EKF, create a new EKF initialized based on the given detection.
- Delete EKF's with no recent associated detection.

Tracking System Test Results

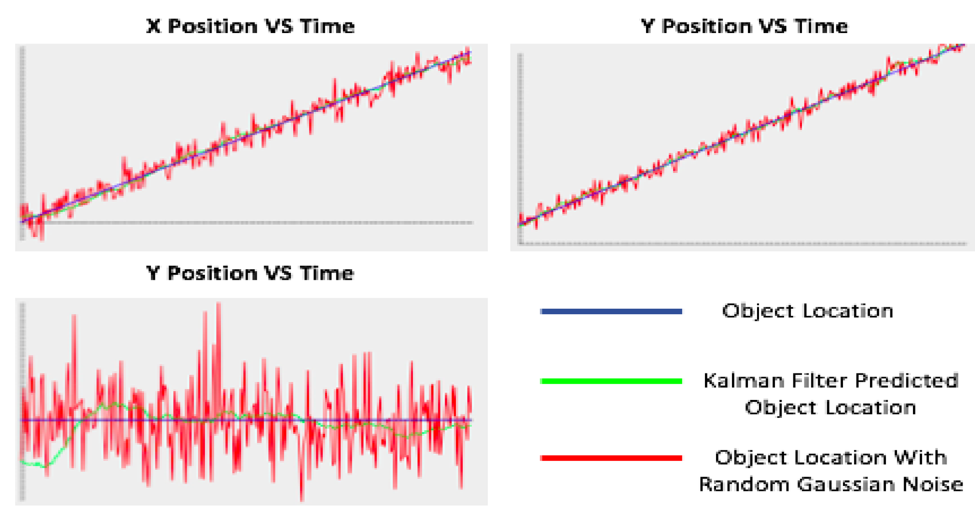

Figure 3: Simulation of Extended Kalman Filter.

As seen in Figure 3, our Extended Kalman Filter(EKF) is able to quickly converge on a highly accurate estimate of an object's location. In addition, when a set of EKF's are used with our detection to track association algorithm, multiple objects can be tracked in a cluttered environment as seen in Figure 4.

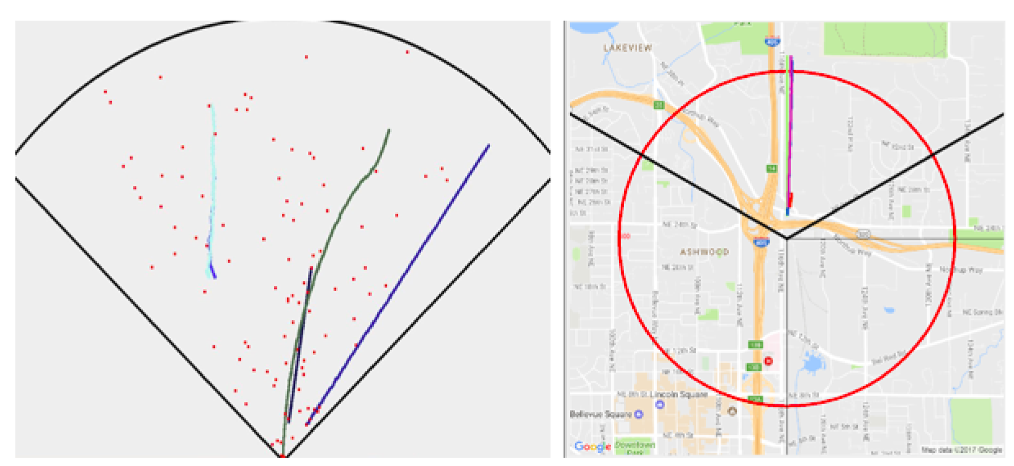

Figure 4: Simulation of Multi-Detection Tracking System on left and track of drone to 500 meters on right.

Graphical User Interface & Results

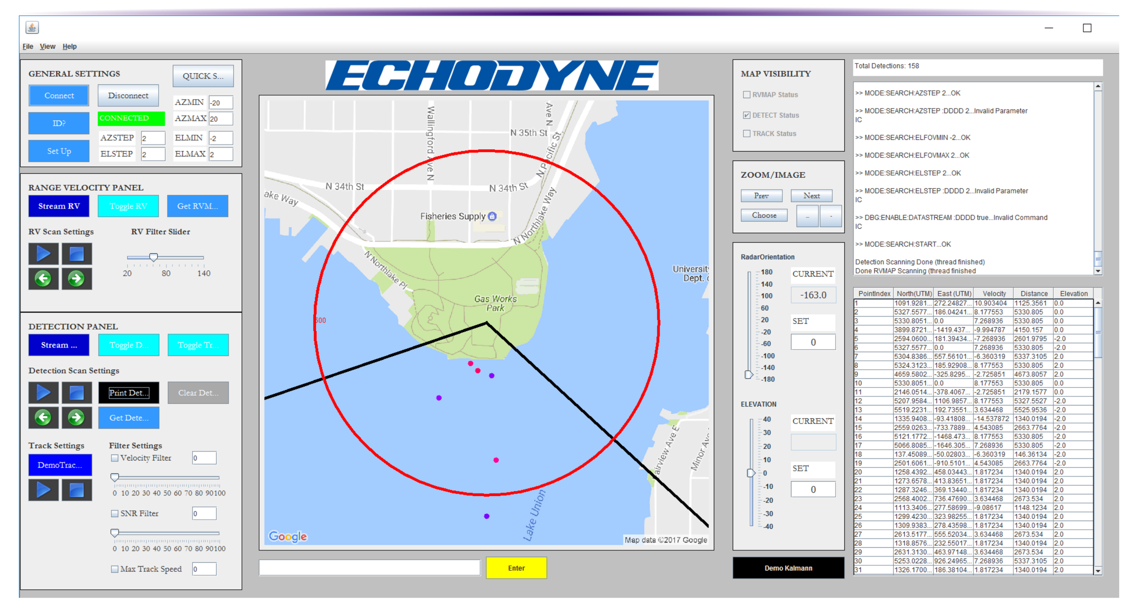

Figure 5: Graphical User Interface for Drone Detection System

Key Features of User Interface:

- Ability to correctly plot range-velocity, detection, and track data regardless of geographic position.
- Ability to adjust key radar functions such as scanning resolution and field of view.
- View key information about each detection and range velocity data from the radar.

Future Improvements & Research

While our team was able to produce a GUI with much of the features necessary to effectively detect and track drones, there are still many improvements that can be made. In particular, our team believes the following improvements will greatly enhance the usability and effectiveness of our product:

- Optimize the multi-detection tracking system to work with a radar that takes 15-30 seconds to gather detections from its full field of view.
- Expand our user interface and tracking system to be able to network with multiple radars.
- Implement a 3-D viewing module to visually incorporate elevation into the current 2-D plots.

Acknowledgements

Our team would like to thank our faculty advisor, Professor John Sahr, and our industry advisors: Tom Driscoll, Jack Hunt, Arlen Schmidt, Robert Worl, and Bryan Womack. Each one of these individuals played a vital role in assisting our team while conducting research and developing our drone detection and tracking system.

Contact Information

- Skyler Martens: skylem3@uw.edu
- Huy Nguyen: nguyeh89@uw.edu
- Daniel Park: dswpark@uw.edu
- Jianfeng Huang: hjf007@uw.edu