



# Real-Time Radar-Based Tracking and State Estimation of Multiple Non-Conformant Aircraft

Brandon Cook<sup>1</sup>, Timothy Arnett<sup>2</sup>, Owen Macmann<sup>3</sup>, and Manish Kumar<sup>4</sup>  
*University of Cincinnati, Cincinnati, OH, 45220*

**In this study, a novel solution for automated tracking of multiple unknown aircraft is proposed. Many current methods use transponders to self-report state information and augment track identification. While conformant aircraft typically report transponder information to alert surrounding aircraft of its state, vehicles may exist in the airspace that are non-compliant and need to be accurately tracked using alternative methods. In this study, a multi-agent tracking solution is presented that solely utilizes primary surveillance radar data to estimate aircraft state information. Main research challenges include state estimation, track management, data association, and establishing persistent track validity. In an effort to realize these challenges, techniques such as Maximum a Posteriori estimation, Kalman filtering, degree of membership data association, and Nearest Neighbor Spanning Tree clustering are implemented for this application.**

## I. Introduction

AS the popularity of unmanned aerial vehicles continue to grow over the coming years, so will the potential number of aircraft lacking transponders in the national airspace. Therefore, alternative solutions for automated tracking of multiple unknown targets needs to be explored. To ensure safe separation between aircraft, it is critical for decision makers in the air traffic management system, such as air traffic controllers, to have accurate information about the vehicles they are responsible for, including: location, speed, and heading. While this information can easily be attained for conformant aircraft equipped with transponders, this feat poses a challenge for non-conformant aircraft and those lacking a transponder. As small vehicles, such as unmanned aerial vehicles, become more prevalent, many may not have the ability to report state information to relevant decision makers. Similarly, this may also be the case for vehicles that are non-conformant or have malicious intent.

For vehicles lacking the ability, or intentionally refusing to report vehicle information, it can be difficult to reliably and consistently estimate the state of the aircraft using ground-based sensors. For these unknown aircraft, often it is beneficial to resort to tracking methods that use imperfect location data provided by radars to infer information about the moving aircraft and track them over time. This process, called tracking, is an elementary problem in air traffic control - but a problem rife with opportunity for exploitation by intelligent systems. Typically, tracking is broken down into several areas of research, including, data association, sensor fusion, state estimation, track identification, and track deletion.

While many approaches and techniques<sup>1,2,3</sup> have been used to track aircraft using only radar data, many require some amount of *a priori* knowledge, such as, the number of vehicles in the airspace, vehicle model information, or the initial vehicle locations. Furthermore, many are used for post-processing or have track identification information available to aid with data association. In an effort to overcome these limitations, a real-time tracking solution that does not require any *a priori* information is presented.

In a study conducted by Reid<sup>1</sup>, a Monte Carlo method was used to simulate the tracking of vehicles over a large number of test cases. Although this study showed promising results, some assumptions limit the application of this work, including, the use of only one radar source, *a priori* knowledge of some of the vehicle states, and not accounting for maneuvering targets.

<sup>1</sup> Graduate Student, Aerospace Engineering and Engineering Mechanics, 745 Baldwin Hall, Cincinnati, OH 45221, and Research Aerospace Engineer, NASA Ames Research Center, Brandon.Cook@nasa.gov, AIAA Student Member.

<sup>2</sup> Graduate Student, Aerospace Engineering and Engineering Mechanics, AIAA Student Member.

<sup>3</sup> Graduate Student, Aerospace Engineering and Engineering Mechanics, AIAA Student Member.

<sup>4</sup> Associate Professor, Department of Mechanical and Materials Engineering.

A similar strategy for clustering data to simplify the solution space and for track identification was used in a study conducted by Park, et al.<sup>2</sup> Here, a minimum spanning tree method was proposed to cluster data without having prior knowledge of how many vehicles existed in the airspace. This solution proved effective despite lacking this information. If the number of vehicles were known *a priori*, the problem of clustering points would be drastically simplified. Once data were clustered and associated to a track, a Kalman filter was used to estimate the state of each aircraft. While the methods used in this study are useful for tracking in simulation tools, the approach could not be applied to real-time tracking systems.

In another study conducted by Chan, et al.<sup>3</sup>, a hybrid fuzzy logic and  $\alpha$ - $\beta$  gain filter was used for state estimation. The primary focus of their study was to track vehicles that are highly dynamic using an approach that would increase state estimation accuracy while minimizing the number of track losses when compared to a traditional two-stage Kalman filter. One drawback to using this two-stage Kalman filter approach is that the vehicle mode must be identified using a maneuver detector or an acceleration detector. Thus, many times the Kalman filter would have delayed transitions between each mode once a new flight mode was detected. Using the fuzzy-gain filter, the system does not make assumptions about process, system, or measurement noise; whereas, a Kalman filter relies on this information, coupled with a dynamic model to estimate the current state of each target. While the fuzzy-gain filter produced favorable results, the system requires identity information, or flight IDs, for each target to distinguish between different vehicles. Since the origin of all data points are known, clustering of data and data association was unnecessary, drastically simplifying the problem of tracking.

In this paper, the authors aim to develop an automated tracking system that addresses the following topics: track creation/clustering, track identification/establishment, sensor fusion, state estimation, data association, and track deletion. In particular, a nearest neighbor spanning tree is used for clustering, a maximum a posteriori probability (MAP) estimation is used for sensor fusion, a Kalman filter is used for state estimation, and a multi-Gaussian membership function is used for data association. As opposed to the many approaches discussed above, and in most other literature, the focus of this paper is on developing a real-time tracking solution for an airspace that contains: vehicles that are highly evasive, aircraft with unknown vehicle models, an unknown number of vehicles, multiple data sources, and no *a priori* information. Therefore, the proposed approach is a viable option for tracking non-conformant aircraft in a highly congested airspace while solely utilizing radar data. Results show a comparison of track estimation accuracy between using the sensor fusion technique alone and the application of a Kalman filter to see if the additional step provides improved estimates.

## II. Simulation Environment

### A. Data Acquisition and Simulation Assumptions

Since the main objective of this research was to allow the radar-based tracking algorithm to be performed in real time, true track data were obtained to help increase the tracking simulation speed. Once these data were gathered, it was imported into the radar tracking simulation environment where the algorithm could be developed and tested.

To obtain the aircraft data, a simulation was developed to allow several aircraft to travel throughout a designated airspace while traversing a series of waypoints. The airspace was selected to be 10 km by 10 km in size, or 100 km<sup>2</sup>. At the start of each simulation, the user could indicate how many vehicles would be active in the airspace, how many waypoints each vehicle should capture before leaving the simulation area, and the total amount of time the simulation will run. Once a vehicle captures all its assigned waypoints, it departs from the designated airspace and a new vehicle enters with random initial conditions. Therefore, throughout the entirety of the simulation, the number of aircraft within the designated airspace remains constant; however, the tracking algorithm must be able to detect when an aircraft leaves or enters the simulation area. For each time step in the simulation, that is every 0.1 seconds, the position data for each aircraft is recorded.

Each aircraft model was developed to be a vehicle traveling at constant speed in level two-dimensional flight. Therefore, a kinematic model was used to model the dynamics of each aircraft. Each vehicle was constrained to a maximum speed of 60 m/s and can turn with a maximum load factor of 3.5. Therefore, the maximum turn rate for each vehicle can be found by using Eq. (1), where  $\dot{\psi}_{max}$  is the maximum turn rate,  $g$  is the gravity constant,  $V$  is the aircraft speed, and  $n$  is the load factor.

$$\dot{\psi}_{max} = \frac{g}{V} \sqrt{n^2 - 1} \quad (1)$$

For more information about the various aspects of this simulation environment refer to Ref. 4. Once the data acquisition is complete, the aircraft position data will be stored in an external file so that it can be used in the radar tracking simulation environment.

In this study, five different simulation scenarios were tested. In each scenario, the number of aircraft and the total amount of simulation time was varied. For each successive scenario, the simulation time or the total number of active vehicles were increased to make tracking vehicles more difficult. Table 1 lists the parameters for all five simulation cases. Here, the ‘‘Total Vehicles’’ refers to the total number of aircraft that appear in the airspace over the entirety of the simulation.

**Table 1. Simulation Cases**

DATA SET	Max Vehicles	Total Vehicles	Simulation Time (hrs)
1	2	3	0.25
2	2	16	2
3	3	25	2
4	10	81	2
5	40	156	1

## B. Radar Sensors

In this study, two radars were used to track all vehicles in the radar tracking simulation environment. To model the radar sensors, functions were created to return noisy radar position data given the true aircraft position. This was done by first finding sources for actual radar sensor standard deviations<sup>5</sup>. The values used for the radar sensor standard deviations are shown in Table 2.

**Table 2. Radar Parameters**

Parameter	Symbol	Value
Range	$\sigma_R$	4.37 m
Angle	$\sigma_\theta$	0.002 rad

These values were then used in combination with the normal distribution pseudorandom number generation function in MATLAB to generate a model that would return normally distributed values. Thus, these normally distributed values were found using Eqs. (2) and (3).

$$R_n = R + \sigma_R \text{randn} \quad (2)$$

Where  $R$  is the actual range of the vehicle from the radar source,  $\sigma_R$  is the standard deviation in range,  $\text{randn}$  is the random number generation function, and  $R_n$  is the noisy returned range value.

$$\theta_n = \theta + \sigma_\theta \text{randn} \quad (3)$$

Similarly,  $\theta$  is the angle used to describe the actual vehicle location with respect to the radar source,  $\sigma_\theta$  is the standard deviation in angle, and  $\theta_n$  is the noisy returned angle value. Using these equations, the generated values in the data acquisition simulation can be converted into noisy radar data for use by the tracking algorithm.

The noisy data are then converted from the respective radar reference frame to the global frame with the following equations:

$$X_1 = R_{R_1} \cos(\theta_{R_1}) + X_{R_1} \quad (4)$$

$$Y_1 = R_{R_1} \sin(\theta_{R_1}) + Y_{R_1} \quad (5)$$

where  $R_{R_1}$  is the range measurement given by radar 1,  $\theta_{R_1}$  is the angle measurement given by radar 1,  $X_{R_1}$  and  $Y_{R_1}$  are the known x and y positions of radar 1 in the global frame, and  $X_1$  and  $Y_1$  are the Cartesian position measurements in the global frame. These equations are applied to every radar to find global Cartesian coordinates for all range and angle measurements (i.e.  $X_i$  and  $Y_i$  for radar  $i$ ). Note that for all results discussed, both radars swept every three seconds. It was also assumed that the radars acted more like directed ranging in that they have instantaneous measurements for all objects in the airspace at the end of the three second sweep. Increasing the fidelity of the radar models by including the differences in measurement timestamps due to sensor sweep rates would increase the realism of the measurements. However, it would not change the calculation frequency, as the measurements would be stored until the calculation was performed.

### III. Tracking Algorithm

To create the tracking algorithm that must solve many different types of problems, such as track management, data association, and state estimation, the problem was broken down into several subsystems. By doing so, the design approach can be simplified and modifications to each subsystem can be independent. The various subsystems for the tracking algorithm, including clustering techniques, data association, track establishment (i.e. track validity), sensor fusion, current state estimation, and track deletion, have been shown in Figure 1. In this figure, the overall flow diagram of the proposed solution can be found.

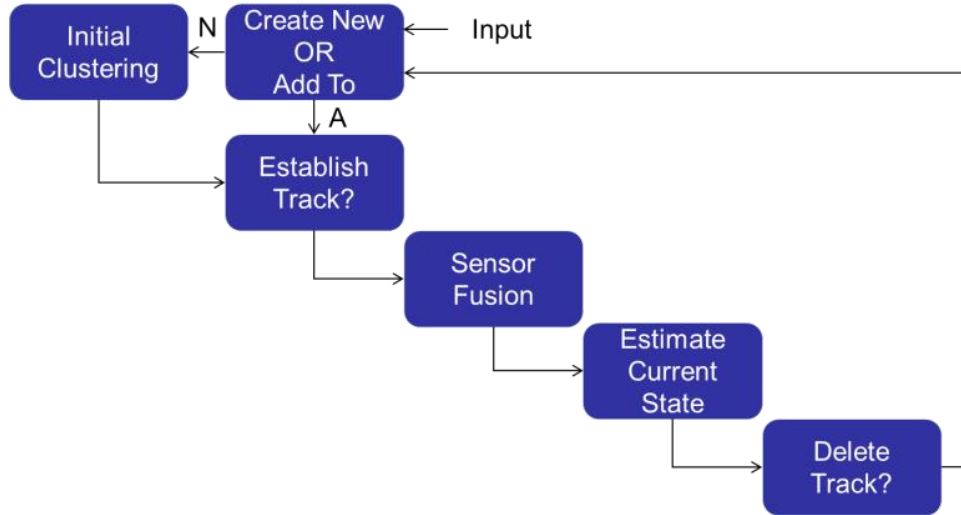


Figure 1. Tracking Algorithm Flow Diagram

In Figure 1, each block represents a different subsystem of the tracking algorithm. The arrows connecting the blocks represent the sequence of steps from when an input is received, and how the new data is processed. Here, an input is considered to be a set of position data points that were obtained from the radar returns in a given time step.

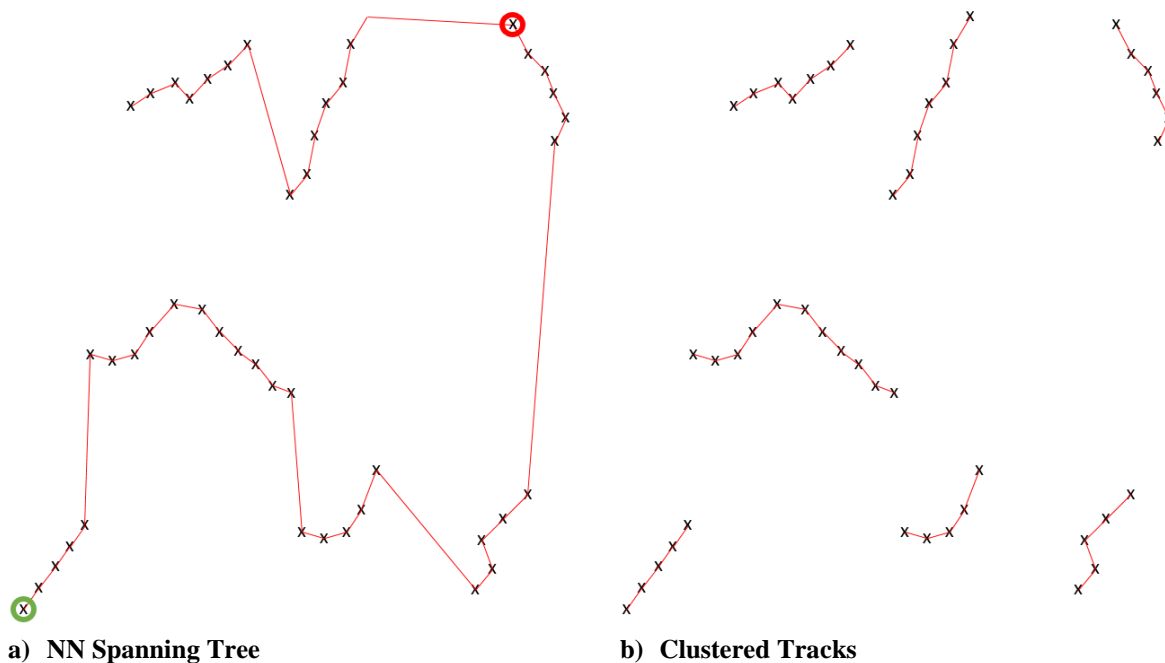
When the tracking system is initialized, no computation will occur until one or more radars have supplied an input. However, once the first input is received, the algorithm will commence and continue to run with each new set of radar returns. On the first iteration of this algorithm, it is obvious that no previous tracks have been identified or created, thus the algorithm will “Create New” clusters, denoted by the “N” arrow connecting the “Create New OR Add To” block to the “Initial Clustering” block. Therefore, the initial clustering algorithm is used to determine which data points most likely belong to the same vehicle. Once the data points have been clustered, a temporary track ID will be assigned to each potential vehicle. Next, the algorithm will use designated criteria to determine whether each cluster is a “valid” track. A valid track thus represents a true vehicle return from the radar.

Once completed, each cluster of data points will run through a sensor fusion algorithm to obtain a single fused state estimation for that time step. Given this fused value, if the algorithm is on the third iteration or higher for a particular track ID, a Kalman filter is used to estimate the current state of each vehicle. Whereas, if the algorithm is on the first or second iteration, the fused value is used as the best estimate of the current state. Finally, the algorithm will decide whether any temporary or valid tracks have gone stale. In this study a stale track means that no new data points have been associated to that particular track for a set period of time. If this is the case, the aircraft will be removed from the list of vehicles that are being tracked.

This process is then repeated for every set of new radar data returns. If one or more track ID(s) have already been assigned, it must check to see if the new data belongs to an already existing track, or if it needs to be assigned to a new track all together. To determine this, a data association platform is used. For points that are assumed to be associated to a given track, the “Initial Clustering” block will be bypassed and the algorithm will follow the “A” arrow from the “Create New OR Add To” block to the “Establish Track?” block.

#### A. Initial Clustering

When new data points are not assigned to a previously existing track, or no tracks have been created yet, a clustering algorithm is employed to determine which data points likely belong to the same vehicle. To accomplish this, a Nearest Neighbor (NN) spanning tree was used. An example of how this NN spanning tree method distinguishes which data points belong to which aircraft, and how many aircraft are present, can be seen in Figure 2.



**Figure 2. Initial Clustering Algorithm Example**

Once a set of data points are imported into this subsystem, it first calculates the distance between all pairs of points. With this information, it finds which two data points are located farthest apart, shown by the circled points in Figure 2 a). Next, one of these two data points are randomly selected and the selected point will serve as the NN spanning tree starting location, depicted as the green circle in the figure. From this selected point, the algorithm will find the closest point to this starting location, and connect the two points with a line. The system will then repeat this process of connecting the current point to the closest point that has not yet been visited until all points are included in the tree. An example of a completed NN spanning tree can be seen in Figure 2 a). Computationally, a NN spanning tree can be created in quadratic time.

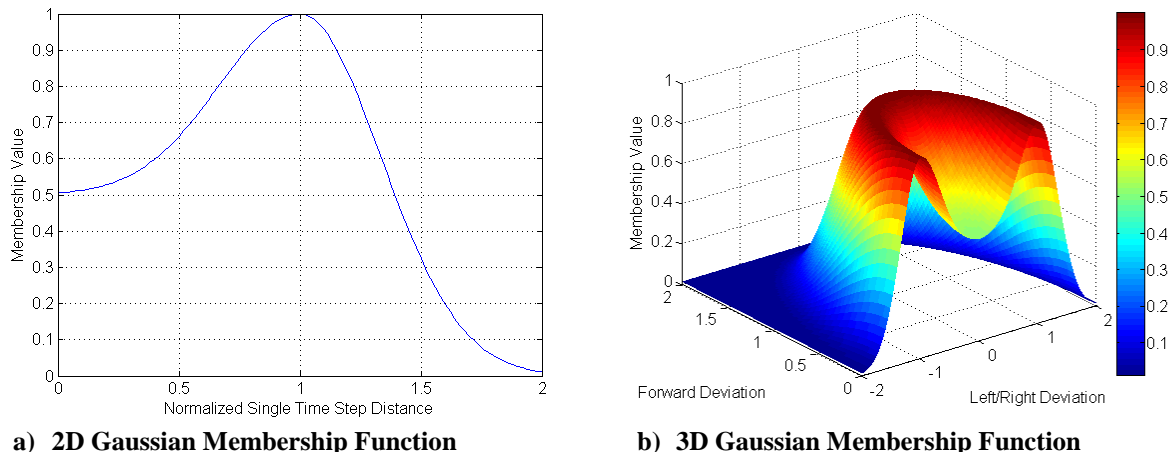
With this NN spanning tree, the algorithm will now determine which connected points belong to the same cluster. To accomplish this, the initial clustering algorithm looks to see if two connected points are too far apart from one another, or if the change in angle between edges connecting three consecutive points is too large. These criteria are defined by the user, and can be modified depending on the accuracy of the radar sources and the types of aircraft expected to be in the airspace of interest.

Once the criteria are checked for each edge on the NN spanning tree, the algorithm will delete the edges between the points that violate these thresholds. A final product of the initial clustering process can be seen in Figure 2 b). These clusters represent individual vehicle tracks. While this technique was utilized for a two-dimensional case in this study, the three-dimensional case would utilize the same approach.

## B. Data Association

If initial clusters have already been assigned, the tracking system must check to see if any new data points should be associated to an already existing vehicle track or if they should be assigned to a new vehicle track. To accomplish this, a data association system was created to determine the degree of membership for each new point to each existing track. For this study, a multi-Gaussian function was used that assigns a degree of membership to each radar return based on the last reported position and speed for each particular aircraft.

Using the last known state information of each aircraft, we can find the maximum distance it could have traveled between the last reported state and the current time step. Therefore, if an aircraft has traveled the exact distance it was expected to travel, and a new data point was found exactly at this predicted distance, it would be assigned a degree of membership of one to that particular track. However, if the point is too close to, or too far from, the predicted location of the vehicle, it would be assigned a degree of membership less than one. Therefore, zero membership signifies there is no chance that the data point could belong to that track.



**Figure 3. Gaussian Membership Function for Data Association**

In Figure 3 a), the multi-Gaussian function has been shown. As one can see from the figure, when a new data point lies at exactly one on the x-axis (i.e. exactly where it was expected to be based on the previous time step information), it is assigned a degree of membership of one. However, if it lies closer or farther than it is expected to be at that time step, the degree of membership is reduced.

Due to the fact that aircraft can have a large change in heading between consecutive time steps, the degree of membership for the case where a point is closer than expected has not been punished as much as the case where it lies too far. If, for example, an aircraft is traveling straight on the previous time step, but then immediately turns hard to its right, when the next radar return is obtained the distance between the two consecutive locations of the aircraft could be quite close to one another. Therefore, we do not want to disassociate data points when an aircraft has turned unexpectedly. Thus, it assigns a higher membership for points closer than expected, and assigns a comparatively lower membership for points located farther than expected. One possible drawback to this approach is for cases where new vehicles pop up in the airspace. This scenario may cause the data association metric to breakdown and miss-assign popup vehicle points to already existing nearby tracks.

In Figure 3 b), a three-dimensional plot of the degree of membership has been shown. This represents the three-dimensional case that will associate points lying in front of the previous aircraft location to its degree of membership. While the true function associates points in all directions, the points lying behind the aircraft have been excluded to help the reader visualize the plot trends. Here, the axes on the horizontal plane represent the location of the new data point with respect to the previous point. Therefore, the point (0, 0) on the horizontal plane represents the exact location of the previous data point. Similar to the 2D representation, the height of the plot represents the membership value.

To determine which raw radar points belong to which track IDs, if any, each new data point is assigned a degree of membership to each existing cluster. Once all memberships have been assigned, an assignment algorithm is used to allocate points to existing clusters. First, the assignment algorithm will step through each point and see if it has membership to only one possible existing cluster. If membership exists to only one cluster, and the membership is higher than 0.5, the point will be assigned to that particular cluster. With the remaining points, the algorithm will then step through each existing cluster and see if exactly two points have a degree of membership greater than 0.1 for that cluster. Since only two radar sources were used throughout the simulation, if a cluster has only two remaining possible points, the points would be assigned to that cluster. Next, the algorithm will step through each remaining radar point to see if it has a degree of membership much higher for one particular cluster than all remaining clusters. If this membership difference is above 0.5, the point will be assigned. Lastly, the algorithm finds the maximum value of all remaining membership values, if this maximum is higher than 0.1 it will be assigned accordingly.

Points that did not meet the threshold criteria are assigned to existing tracks. Thus, these points will be clustered using the “initial clustering” techniques described in the previous section immediately following the data association step. These new clusters are thus assigned new track IDs.

### C. Establish Track

After the tracking algorithm has assigned new data points to existing tracks, or has created new tracks, it will now determine if a track is “valid.” A track is valid when it is believed to represent an actual aircraft. Therefore, in a real-world scenario this aircraft shall be reported to all other aircraft in its vicinity, or to an air traffic controller, to ensure no aircraft collide with one another.

To determine if a temporary track ID should be reassigned to a valid track ID, two criteria are checked and must be met. First, the algorithm checks to see how many data points have been assigned to that particular track. Next, it will calculate the lifetime of the track. This is done by looking at the most recent time stamp for a set of data points that have been assigned to that track and the first ever recorded time stamp for that track. If the lifetime is greater than a user-defined threshold, 15 seconds, and has more than enough data points assigned to it, the track will become valid.

#### D. Sensor Fusion

Once the raw measurements have been associated to a track, the values need to be fused in order to provide an overall best estimate for the measured vehicle location. To fuse the measurements, the posterior probability distributions of the measurements as given by the normal distribution parameters in Table 2 were maximized to yield the best overall estimation of the true location. The maximized posterior (MAP) estimate for the range using two radar sources is given in the following equation:

$$R_{MAP} = \frac{\sigma_{R_2}^2}{\sigma_{R_1}^2 + \sigma_{R_2}^2} R_1 + \frac{\sigma_{R_1}^2}{\sigma_{R_1}^2 + \sigma_{R_2}^2} R_2 \quad (6)$$

where  $R_{MAP}$  is the overall fused range value in the global frame,  $R_1$  and  $R_2$  are the range measurements in the global frame for radars 1 and 2 respectively, and  $\sigma_{R_1}$  and  $\sigma_{R_2}$  are the standard deviations in range for radars 1 and 2 respectively. For the general case where  $n$  radars exist, the MAP estimate can be found using Eq. (7).

$$R_{MAP} = \sum_{i=1}^n \left( \frac{\prod_{j=1}^n (\sigma_j^2)}{\sum_{p=1}^n \left[ \prod_{q=1}^n (\sigma_q^2) \right]} R_i \right), j \neq i \text{ and } q \neq p \quad (7)$$

To find  $R_i$ , the range from each respective radar in the global frame,  $X_i$  and  $Y_i$  (Eqs. (4) and (5)) were used to create the relationship shown in Eq. (8).

$$R_i = \frac{X_i}{\cos\left(\tan^{-1}\left(\frac{Y_i}{X_i}\right)\right)} \quad (8)$$

After finding the MAP estimate for range in the global frame, the same must be done for angle. This is shown in the following equation:

$$\theta_{MAP} = \frac{\sigma_{\theta_2}^2}{\sigma_{\theta_1}^2 + \sigma_{\theta_2}^2} \theta_1 + \frac{\sigma_{\theta_1}^2}{\sigma_{\theta_1}^2 + \sigma_{\theta_2}^2} \theta_2 \quad (9)$$

where  $\theta_{MAP}$  is the overall fused angle value in the global frame,  $\theta_1$  and  $\theta_2$  are the angle measurements in the global frame for radars 1 and 2 respectively, and  $\sigma_{\theta_1}$  and  $\sigma_{\theta_2}$  are the standard deviations in angle for radars 1 and 2 respectively. Again, this can be represented for  $n$  radars using the general form shown in Eq. (10).

$$\theta_{MAP} = \sum_{i=1}^n \left( \frac{\prod_{j=1}^n (\sigma_j^2)}{\sum_{p=1}^n \left[ \prod_{q=1}^n (\sigma_q^2) \right]} \theta_i \right), j \neq i \text{ and } q \neq p \quad (10)$$

Similar to  $R_i$ ,  $\theta_i$  was found using  $X_i$  and  $Y_i$ . This is shown in Eq. (11).

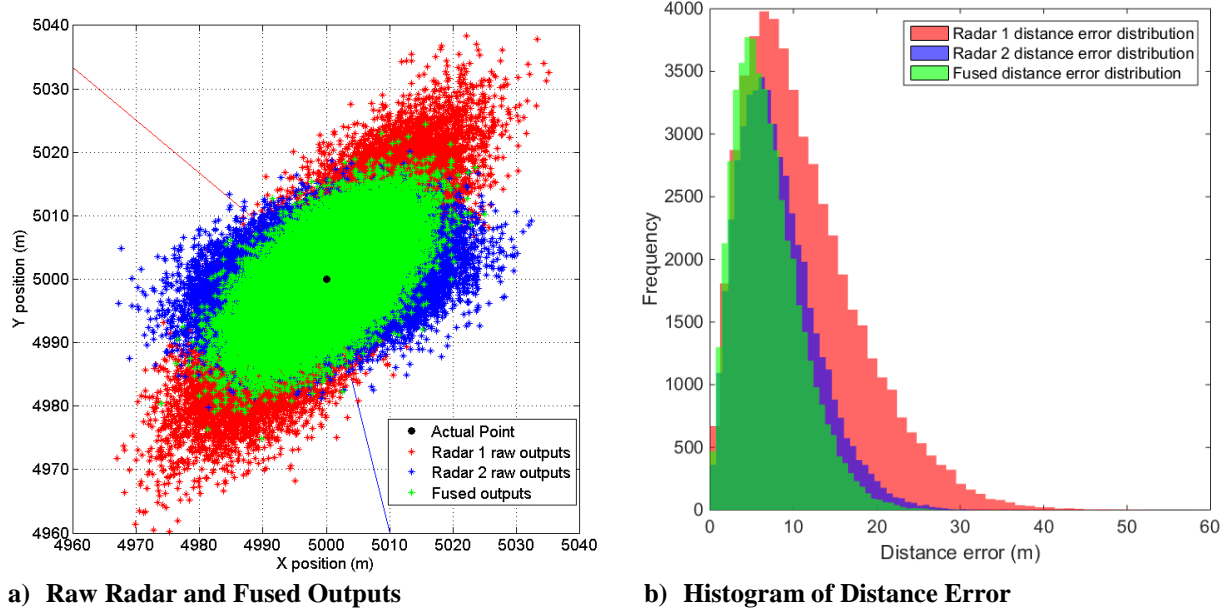
$$\theta_i = \tan^{-1}\left(\frac{Y_i}{X_i}\right) \quad (11)$$

Now that the measurements are fused into global values for  $R$  and  $\theta$ , they can be converted back to Cartesian with the following equations:

$$X_F = R_{MAP} \cos(\theta_{MAP}) \quad (12)$$

$$Y_F = R_{MAP} \sin(\theta_{MAP}) \quad (13)$$

The above data fusion equations were tested in a scenario where radar 1 was placed at the location (200, 9000) [m] and radar 2 was placed at (6000, 1000) [m]. The standard deviations in range and angle for both radars were the same as in Table 2. Using these values and a normal distribution, each radar measured the location of an object placed at (5000, 5000) [m] 50,000 times (Eqs. (2) through (5)). The results of the raw measurements and the fused values and are shown in Figure 4.



a) Raw Radar and Fused Outputs

b) Histogram of Distance Error

Figure 4. Distribution Plots for 50,000 Runs of the Same Point

As can be seen in Figure 4 a), the fused value appears to have a much smaller distribution about the actual value. With these data points, the distance from each point, raw and fused, to the actual value was calculated. These distance values were plotted in a histogram to compare the accuracy of the raw values versus the fused values. This histogram demonstrating the distribution of results can be seen in Figure 4 b). Here, it is clear that the distribution of the fused values is more representative of the actual location of the object. Both the mean error and the standard deviation for the fused distribution is less than either single radar distribution.

### E. Current State Estimation

In order to provide the best estimate for the current vehicle position, a Kalman estimation algorithm was developed. The Kalman estimator uses a general aircraft model combined with fused sensor measurements to give an overall best estimate for the new aircraft position. The aircraft model used is a kinematic model given by Eqs. (14), (15), and (16), and assumes each aircraft is constrained to 2D level flight.

$$x_i = x_{i-1} + v_{i-1} \cos(\psi_{i-1}) dt \quad (14)$$

$$y_i = y_{i-1} + v_{i-1} \sin(\psi_{i-1}) dt \quad (15)$$

$$\psi_i = \psi_{i-1} + \dot{\psi}_{i-1} dt \quad (16)$$

In the above equations,  $i$  represents the current time step,  $i - 1$  represents the previous time step,  $dt$  is the time step increment,  $v$  is speed,  $\dot{\psi}$  is angular velocity,  $\psi$  is heading angle, and  $x$  and  $y$  are Cartesian positions. From these equations and using an Euler method integration step, the state space representation was created and is shown with the following equations.

$$\begin{bmatrix} x_i \\ y_i \\ \psi_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i-1} \\ y_{i-1} \\ \psi_{i-1} \end{bmatrix} + \begin{bmatrix} \cos(\psi_{i-1}) dt & 0 \\ \sin(\psi_{i-1}) dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} v_{i-1} \\ \dot{\psi}_{i-1} \end{bmatrix} \quad (17)$$

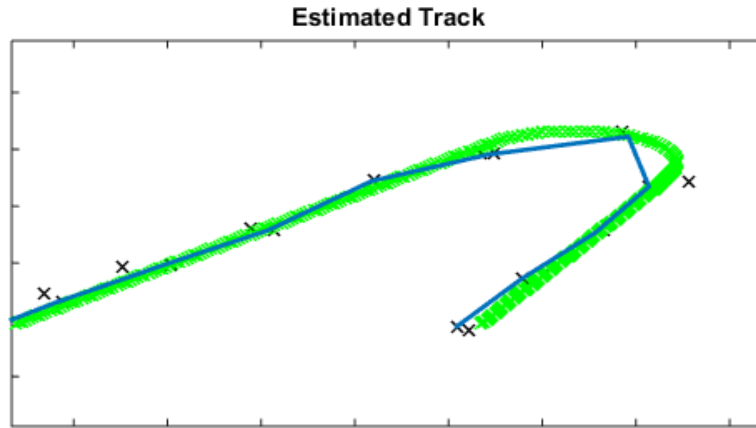


The measurement equation was then created and is shown in Eq. (18).

$$z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{i-1} \\ y_{i-1} \\ \psi_{i-1} \end{bmatrix} \quad (18)$$

Where  $z$  is a vector of the fused measurements,  $X_F$  and  $Y_F$ .

The state space representation was then used to create a Kalman estimator using the common algorithm<sup>6</sup>. This estimator was used by the algorithm on each iteration, for each existing track, after the third time step. This is due to the fact that the first value for  $\dot{\psi}$  is only available on the third time step. Before the third time step, only the fused sensor values are used for state estimation.



**Figure 5. Vehicle Track Showing Actual Path (green), Radar Returns (black), Kalman Estimates (blue)**

In Figure 5, an example of the Kalman estimation for a track based on the associated radar values and model predictions is shown. It should be noted that since the measurements are taken every 3 seconds and the vehicle can turn more than 90 degrees in that time, less weight was put on the model predictions than the measurement values in the filter. This resulted in the Kalman estimated path more closely following the fused measurements while still taking the assumed model into account. This could be improved by adding more radars that sweep at different times or increasing the measuring frequency of the current radars.

Computing the Kalman estimated value is not computationally expensive. The Kalman estimation can be computed in linear time, proportional to the number of fused data inputs, and each point can be processed on the order of  $10^{-5}$  seconds.

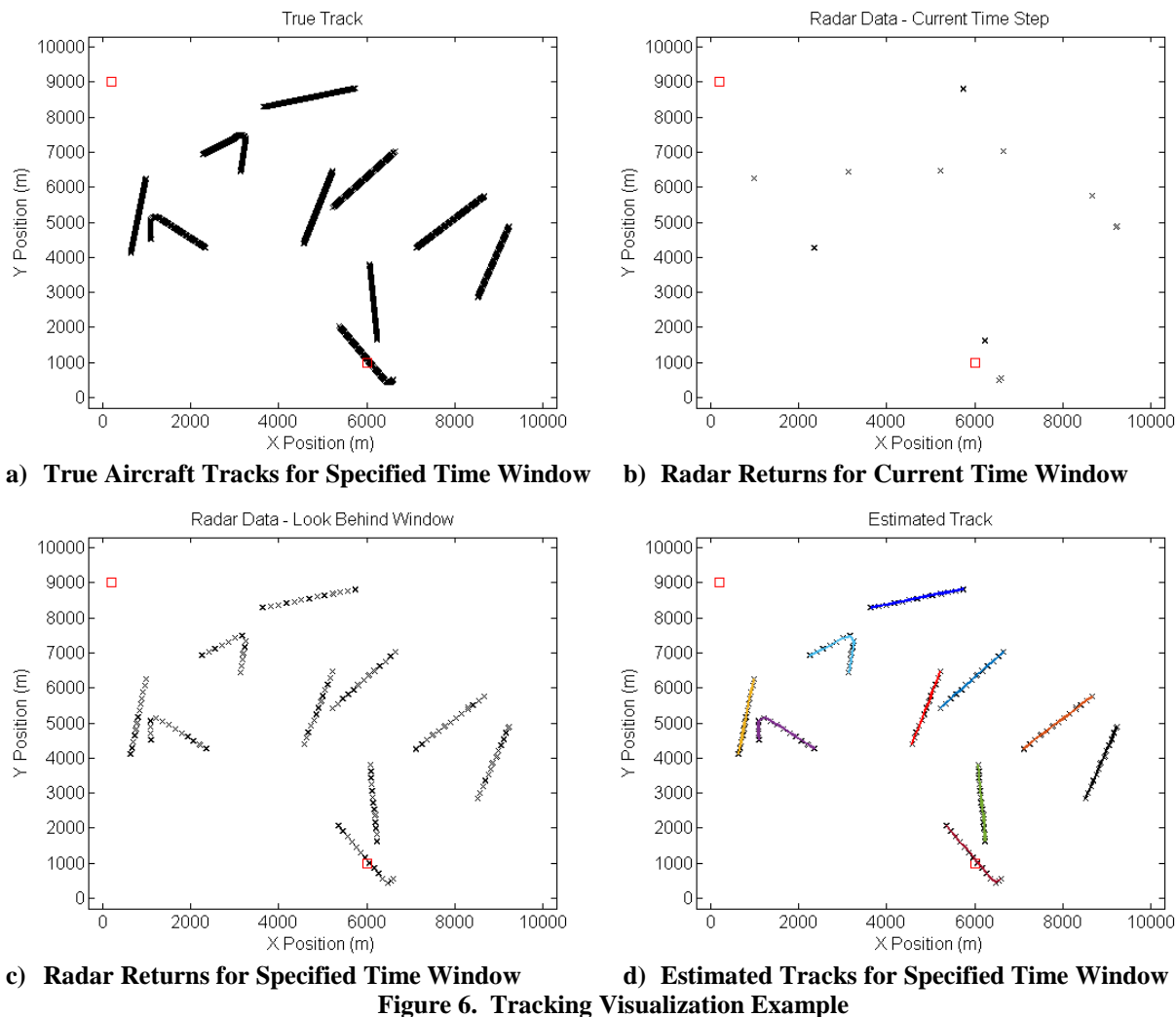
#### F. Delete Track

Similar to the criteria used when determining when a track should be deemed valid, a time threshold is used to determine when a track has gone “stale.” A track becomes stale when the vehicle is no longer detected by radar for an extended period of time. In practice, this could happen if a vehicle were to crash, land, or leave the designated airspace. In addition, a stale track can be one that never became valid. Therefore, this method can also be used to throw out data points that should have belonged to a particular aircraft, but did not meet the data association criteria due to a bad radar data return; that is, too much uncertainty existed in the radar return to be assigned to the correct track.

To determine if an aircraft track has gone stale and should be deleted from the list of candidate clusters to which new points can be assigned, the current time step minus the last recorded data return time stamp must meet a user-defined threshold. For this study, a threshold of 15 seconds was used. Therefore, if a track has not been assigned a new data point for 15 seconds, the track will be deleted and deemed stale. While this threshold was sufficient for all cases tested in this study, this threshold would need to be modified to account for different vehicle platforms, the amount of time between consecutive data returns, and the probability of detection for each radar source.

## IV. Results

To briefly depict the tracking simulation environment, a snapshot from data set 4 can be seen in Figure 6.



**Figure 6. Tracking Visualization Example**

In Figure 6 a), the true aircraft position data are shown. To help visualize where the vehicle has been in the past, a trail of each vehicle position has been depicted. In Figure 6 b), a snapshot of the current time step radar returns is shown. This is the data that would be imported into the tracking algorithm for this particular time step. In Figure 6 c), a trail of radar returns is shown. Lastly, Figure 6 d) displays the tracking algorithm output. Here, each track ID has been plotted with a different track color, and has been superimposed on top of the raw radar data returns. As one can see, all data points have been successfully associated to their correct tracks, and all tracks are distinguishable by the tracking algorithm.

To analyze the performance of the tracker, one can examine the comparison between the number of true vehicles in the airspace versus the number of tracks assigned by the tracking algorithm. For this study, the ratio between the assigned tracks and the true number of vehicles was used as a figure of merit. To compute this ratio, the total number of assigned tracks was calculated at the conclusion of the simulation. This total number of tracks was then divided by the true number of vehicles that existed in the airspace throughout the entirety of the simulation. Therefore, a value less than 1 implies an under-assignment of tracks; that is, the tracking algorithm did not identify some new data points as new vehicles. If a value of greater than 1 was found, this means that an over-assignment of tracks has occurred. An example for when this may occur is when the tracking algorithm loses an aircraft, deeming it as a stale track. If at a later time the algorithm recognizes this aircraft, it will be assigned a new flight ID; thus, the number of tracked vehicles is now more than the true number of vehicles in the airspace. This ratio is referred to as the Correctly Identified Tracks (CIT) ratio. For each simulation case, the CIT ratio was recorded and can be found in Table 3.

In addition to using the CIT ratio as a performance measure for the tracking algorithm, four additional parameters were considered. As the traffic density of the airspace increases, so does the likelihood of two aircraft coming within close proximity of one another. Many times tracking algorithms will miss-assign data points associated with these

aircraft when a radar return that resonated from one aircraft, is thought to have come from another nearby aircraft. Throughout all simulations, the tracker will tally the total number of instances where this occurs. The results of this “Miss-Association” metric can be found in Table 3.

As previously mentioned in the simulation environment description, when a vehicle exits the airspace and a new vehicle appears with random initial conditions, no restriction is placed on where the new vehicle will popup. Therefore, in many instances the new vehicle would appear very close to where the exiting vehicle was expected to be. When this would occur, the data association algorithm would sometimes assign the popup vehicle data points to the exiting vehicle’s track ID. Thus, resulting in an under-assignment of track IDs. Instances of this nature could occur in physical systems where an aircraft lands or exits the sensor range, while another vehicle takes off or is picked up by radar. Due to this possible occurrence in practice, the simulation was not restricted to omit these cases. The total number of instances where a popup aircraft was not correctly identified is shown in Table 3 as the “Missed Popups” metric.

There were also instances where a popup vehicle appeared near an exiting aircraft and was correctly assigned a new track ID. However, due to being close in proximity, on the next iteration the algorithm miss-assigned the popup vehicle’s points to the exiting vehicle’s track ID. These instances are known as the “Lost Popups” in Table 3. In these cases, the CIT Ratio is not affected since the new point was initially identified and assigned a new track ID, thus yielding the same total number of tracks.

A final metric used to evaluate the algorithm performance was the number of vehicles that were thought to have gone stale prior to actually leaving the airspace. If for example, an aircraft is traveling along and the data-association system does not assign new data points to its track (i.e. assigned to another existing track or to a new track altogether), the aircraft track has the potential of going stale if no future data points are assigned. Thus, the track has been lost. Another example of how a track can be prematurely lost is if a vehicle exiting the airspace is near another active vehicle. Not to be confused with the lost popup metric, the exiting vehicle is near another already active vehicle, as opposed to an exiting vehicle being near a popup vehicle. In these cases, the data association may assign the vehicle that has left the airspace the data points from the other nearby aircraft. Therefore, the nearby aircraft would not be assigned any new data points and would go stale; whereas, the exiting vehicle would remain an active track, following the other vehicle’s path. These cases are referred to as the “Lost Track” cases below.

**Table 3. Data Association Metrics**

<b>DATA SET</b>	<b>CIT Ratio</b>	<b>Miss-Associations</b>	<b>Missed Popups</b>	<b>Lost Popups</b>	<b>Lost Tracks</b>
1	1.00	0	0	0	0
2	1.00	0	0	1	0
3	1.00	0	0	1	0
4	1.00	0	0	1	0
5	0.98	2	4	5	3

By referencing the CIT Ratio column in Table 3, it can be seen that throughout all simulations there were no over-assignments of flight IDs. Thus, the algorithm did not miss-assign new points that truly belonged to an already existing track, to a new track ID. However, in data set 5 the tracking algorithm miss-assigned points to an existing track ID when they truly belonged to a new track ID, resulting in a CIT Ratio less than one. In all instances, this under-assignment of tracks was a result of a missed popup scenario. While this result is undesirable with respect to system performance and reliability, this was expected to occur.

In data sets 1-4, the tracking algorithm correctly identified all tracks, never had any miss-associations, and had no tracks go stale prior to their mission completion. However, in data sets 2-4, each simulation had one instance where a popup vehicle was originally correctly identified, but was then assigned to the exiting vehicle’s ID.

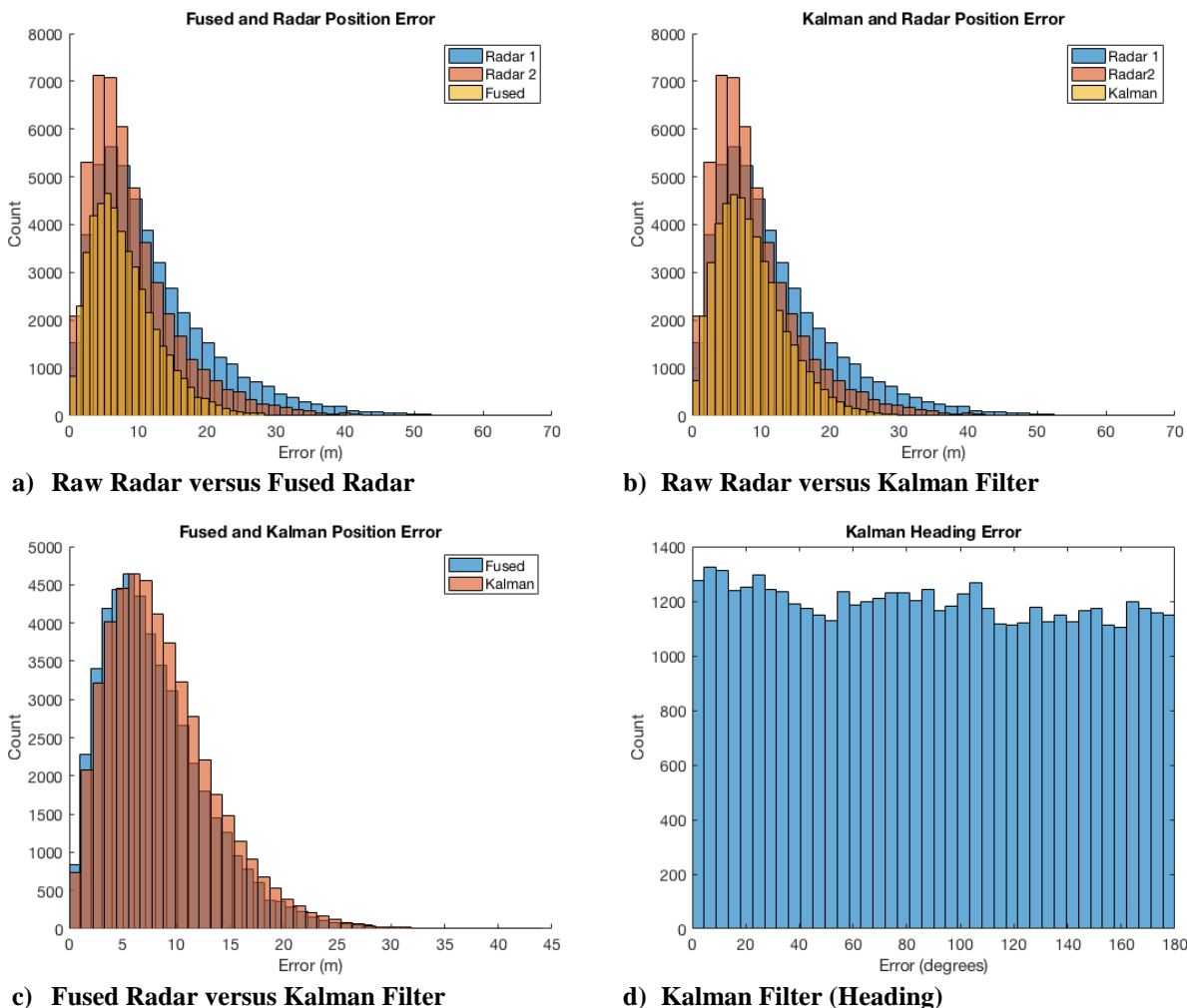
When the total number of aircraft was drastically increased in data set 5, the tracking algorithm began to have more difficulty. Throughout the simulation, the tracking algorithm missed four popups in total. Due to these missed popups, the total number of identified tracks was four less the total number of true tracks, thus yielding a CIT ratio of 0.98. Due to the increased density of the airspace, missing popups or losing popups after being identified was expected to occur. When a vehicle exits the airspace, or a new one pops up, the probability of having another nearby aircraft is increased.

As seen in Table 3, on three instances a track was lost prior to completing its mission. As previously described, there were two options for how a track could be lost. First, if data points that belong to a track are not assigned correctly, but are instead assigned to a new cluster, an over-assignment of vehicles would occur. The second scenario is when an exiting vehicle is near another active track. If the exiting vehicle cluster steals the nearby aircraft radar points, the true vehicle will no longer have future data points to be assigned. Thus, the track will go stale. In this second scenario, the total number of tracks are retained, thus, the CIT ratio is not affected. Throughout testing, all lost tracks were due to the second scenario described above.

On one occasion two vehicles in close proximity caused a miss-association of data points in data set 5. When two active vehicles were crossing one another's paths, the two of the four radar returns were miss-assigned. Although this only happened on one occasion, this was marked as two missed data associations. As a result, both the fused radar values and the Kalman filter values had a larger amount of error than all other estimations.

In all cases where this "Miss-Association" occurred, the algorithm did not indefinitely mix up the track IDs. Once they crossed each other's path, the data points were again correctly assigned. One way to fix this miss-assignment of data points would be to allow the algorithm to consider which points came from which radar source. If such information was available, the tracking algorithm could be sure not to assign two points to a vehicle that resonated from the same radar source.

To evaluate the performance of the sensor fusion and state estimation systems, the position and heading error was calculated for all aircraft and all data sets. In Figure 7, the results for data set 5 are displayed as histograms.



**Figure 7. Histograms of Position and Heading Error**

In Figure 7 a), the raw radar position error for each source and the fused radar position error is shown. It can be seen that the fused radar error is consistently less when compared to using either radar source individually. The shape of the histogram is skewed to the left, insinuating that lesser errors are found more often than larger errors. This same trend holds for the Kalman state estimation error, as seen in Figure 7 b). In Figure 7 c), the fused radar error and the Kalman error has been shown on the same histogram plot. It can be seen that the fused values consistently have slightly less error than the Kalman values. Lastly, in Figure 7 d), a histogram of the heading error calculated by the Kalman state estimation is shown. This distribution implies that the heading cannot be predicted with high accuracy. This inaccuracy is due to the vehicles having a relatively high turn rate when coupled with the slow sensor sweep rate. These factors make the reachable solution space quite large, making it difficult for the Kalman filter to accurately predict the true heading value.

The results for all other data sets were similar in nature. The mean and standard deviation for all errors and all data sets can be seen in Table 4.

**Table 4. Aircraft Position and Heading Errors**

DATA SET	Statistic	Position Error (m)				Heading Error (degrees)
		Radar 1	Radar 2	Fused	Kalman	Kalman
1	Mean	11.5	9.2	7.7	8.4	78.4
	Std. Dev.	8.9	6.5	4.9	4.9	49.2
2	Mean	11.5	9.4	7.9	8.5	84.5
	Std. Dev.	8.5	6.9	4.7	4.8	51.3
3	Mean	11.7	9.3	7.9	8.5	86.1
	Std. Dev.	8.5	6.8	4.7	4.8	53.4
4	Mean	12.0	9.2	8.0	8.6	87.5
	Std. Dev.	8.8	6.8	4.9	5.0	53.2
5	Mean	12.0	9.1	8.0	8.6	88.3
	Std. Dev.	8.8	6.6	4.9	5.1	52.2

As seen from the above table, the results for each data set were quite consistent. In all cases, Radar 1 and Radar 2 had the highest amount of position error. Once passed through the sensor fusion system, the fused radar values had less total mean error and had a lower standard deviation. Thus, the aircraft true location could be more accurately represented. After the fused values were found, the algorithm used a Kalman filter to predict the aircraft location and heading. In all cases, the Kalman filter had slightly more average error and a slightly higher standard deviation.

The increased error in the Kalman values is due to the way the Kalman filter was constructed. Specifically, more reliance was placed on the fused radar values because there was little knowledge about the aircraft model. By increasing the weight on the fused values, all Kalman values were close to the original fused point location. However, by not having placed much trust on the vehicle model, errors were added to the system. In all scenarios tested, the radar sources were located relatively close to all vehicles in the operational airspace, thus, the raw radar errors were fairly small. As aircraft begin traveling farther away from the radar sources, we would expect the fused radar values to have a higher amount of error. Thus, in these instances we would expect the combined sensor fusion and Kalman filter system to outperform the sensor fusion system if used by itself.

The sensor fusion values could estimate the position of the aircraft, but not the heading of the vehicle. However, adding the Kalman filter allowed the heading to be estimated. When aircraft were traveling in a fairly linear fashion the state estimation technique was highly effective. However, as aircraft began to turn, due to the aggressiveness of their navigation controllers, having a large maximum turn rate, and a low radar sweep rate, the heading estimation begins to break down. If data were available at a faster rate or if the vehicles were limited in turn rate, we would expect the Kalman filter to have greater performance, especially in the area of heading estimation.

## V. Conclusion

The tracking method proposed in this study was complete and robust, capable of tracking every vehicle with a mean error less than 8.7 m. Although data points were occasionally associated to incorrect tracks, the methods proposed in this study were sufficient to show this approach could be used if some optimization and tuning were imposed. Throughout testing, the sensor fusion system was marginally superior to using the fusion and Kalman estimation hybrid system. For all cases, the difference between the two system errors was less than 10%. This slight increase in error for the hybrid system was likely due to the low confidence placed on the vehicle model used for the Kalman filter, and the low sensor sweep rate. Nonetheless, both architectures minimized the true distance errors and imposed better confidence for data association.

This work serves as a good baseline for future studies, and would require some additional research and refinement prior to implementation. One area for future work would be in the data association algorithm. At this time, the user-defined thresholds have not been optimized to account for various vehicle models and/or sensor sweep rates. In addition, further research would need to be conducted to deal with aircraft that can hover or are highly evasive. These types of models pose a great challenge and would need to be overcome using an adaptive membership association based on the detected vehicle dynamics. Lastly, in this study the radar sensors do not incorporate a non-unity probability of detection. Integrating this element may require additional tuning for the track identification/deletion and data association algorithms.

Once more intelligent algorithms for data association, track validity, and track deletion are created, more complex scenarios with fewer assumptions can be tested. By increasing the traffic density in a given area, the radar returns

become more difficult to associate to existing or new tracks. Thus, in areas of an airspace expecting higher traffic flow densities, such as those found near airports, higher precision radars should be used to identify vehicles. In doing so, a weighted average system can be imposed to put more weight on data returns from radars that are closer to the identified object versus returns from farther radars with higher uncertainty.

### References

- <sup>1</sup>Reid, D., "An Algorithm for Tracking Multiple Aircraft," *IEE Transactions on Automatic Control*, Vol. 24, Issue 6, Dec. 1979, pp. 843-854. doi: 10.1109/TAC.1979.1102177.
- <sup>2</sup>Park, C., Lee, H.T., Musaffar, B., "Radar Data Tracking Using Minimum Spanning Tree-Based Clustering Algorithm," *11<sup>th</sup> AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Sept. 2011. doi:10.2514/6.2011-6825.
- <sup>3</sup>Chan, C. C. K., Lee, V., Leung, H., "Radar Tracking for Air Surveillance in a Stressful Environment Using a Fuzzy-Gain Filter," *IEEE Transactions on Fuzzy Systems*, Vol. 5, Issue 1, Feb. 1997, pp. 80-89. doi:10.1109/91.554452.
- <sup>4</sup>Cook, B., Arnett, T., Rich, B., Kivelevitch, E., "UAS Collision Avoidance, Navigation, and Target Assignment in a Congested Airspace Using Fuzzy Logic," *AIAA SciTech 2015 – AIAA Infotech @ Aerospace*, Jan. 2015. doi:10.2514/6.2015-2031.
- <sup>5</sup>Curry, G. R., "RADAR Measurement and Tracking", *RADAR System Performance Modeling*, 2<sup>nd</sup> ed., Artech House, MA, 2005, pp. 169-171.
- <sup>6</sup>Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Vol. 82, Issue 1, March 1, 1960, pp. 35-45. doi:10.1115/1.3662552.