Detection and Tracking of Multiple Intruders Using the Recursive-RANSAC Algorithm

Jared Wikle, Timothy W. McLain
Brigham Young University

Introduction

Unmanned air systems (UAS) have been growing in popularity during the past 10 to 15 years. In 2001 the Pentagon had less than 50 unmanned aircraft and by 2012 had over 7,500 [1]. The Department of Defense has also shown more interest in unmanned aircraft by providing $6.0 billion in 2011 [2, 3]. UAS technology has not been growing within the United States territory as quickly due to regulations of integrating these aircraft within the National Airspace System (NAS). Because of these regulatory challenges the Department of Defense (DoD) has stated that, “integrating UAS into the NAS is essential to fulfill our airborne mission requirements” [4].

In order for unmanned aircraft systems (UAS) to be integrated into the National Air Space the FAA has stated that unmanned aircraft must provide an, “equivalent level of safety,” to the see and avoid requirements for manned aircraft [5]. This technology known as Sense and Avoid (SAA) needs to be developed so that the ownship will not have a mid-air collision with other aircraft in the sky known as intruders. Since the FAA does not allow unmanned aircraft into the NAS without a SAA system they have created a temporary way to gain access to this space through what is known as the Certificate of Authorization (COA) process [6].

Another problem arises in creating a SAA system for unmanned aircraft because these aircraft have a much larger range of the sizes and performance capabilities than manned aircraft [7]. These constraints of size, weight and power (SWaP) make it almost impossible to use existing sensor technologies [8].

The class G airspace seems to be the first place that small unmanned aircraft will likely be allowed to fly. The Aviation Rule-making Committee (ARC) has been given the responsibility to create regulations for sUAS [9]. A statement made by the ARC is that these aircraft, “will likely experience the greatest near-term growth in civil and commercial operations because of their versatility and relatively low initial cost and operation expenses” [10]. The only thing that prevents these sUAS from gaining access to the NAS is creating an acceptable SAA system.

Within BYU's MAGICC lab research is being done to develop a SAA system using radar as the primary measurement sensor which will provide both range and bearing.

State of Practice

As mentioned in the introduction there currently doesn’t exist a SAA system for sUAS; however, BYU’s MAGICC lab has been actively trying to create such a system. Past research has involved creating a Matlab and Simulink simulation environment that uses an EKF to detect a single

intruder that flies at constant altitude, velocity and heading. This simulation environment has also included a model of a radar that results in a 120^o field of view in the horizontal plane and 30^o in the vertical plane. This radar model is formed by pointing three radars in the forward direction as shown in **figure 1**.

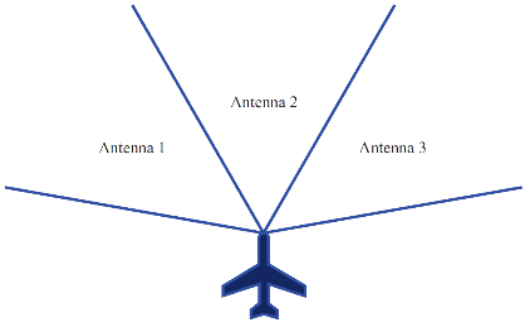

Figure 1: Radar antenna setup for Matlab/Simulink simulation

This existing simulation also includes a simple tree branching guidance algorithm to plan a new path. The research being done in this paper focuses on improving the current target detection and tracking algorithm and leaves the guidance algorithm improvements for other researchers in BYU's MAGICC lab.

Research Challenges

As stated in the previous section the target detection and tracking algorithm that has been developed in BYU's MAGICC lab uses an EKF to detect a single intruder. The EKF in this simulation requires previous knowledge of the number of intruders being sensed by the radar. The existing model also doesn't handle spurious measurements, missed measurements, or multiple measurements. Some other limitations of the existing simulation are that it doesn't handle multiple intruders, overtaking intruders and intruders that don't have straight and level flight. Overcoming these shortcomings has been the aim of this research. The first steps that have been taken have included extending the target detection and tracking algorithm to handle multiple intruders. It has also been extended to handle spurious measurements, missed measurements, and multiple measurements at the same time step. Steps that have not been taken yet include overtaking intruders and intruders that don't have straight and level flight; however, this is starting to be researched.

Technical Approach

Recursive RANSAC

A new algorithm know as Recursive RANSAC was recently developed here in BYU's MAGICC lab. The main purpose of this algorithm is to dynamically estimate the states of multiple targets even when the measurements are noisy, include spurious measurements, and when multiple targets are being measured.

The Recursive RANSAC algorithm creates and stores multiple hypothesis models and when new measurements are received they update existing models to which they are inliers to using a Kalman Filter (KF). If a new measurement is not an inlier to an existing model then RANSAC is called and a new model is formed with the current measurement. Models that have a sufficient number of inliers are labeled as good models and are used as tracks to existing intruders.

One of the primary goals of the RANSAC algorithm is to find an initial condition for the measurement that has been received that has the most number of inliers. The Recursive RANSAC algorithm that has been developed here at BYU is only able to find an initial condition if the state space equation and output equation are linear.

Implementation of Recursive RANSAC Algorithm

The state space equations and output equation that are being used in my research are all nonlinear equations. This presents a problem in

using the Recursive RANSAC algorithm for my research because, as was stated in the previous section, the Recursive RANSAC algorithm requires linear state space equations and output equations. This has required me to do some modifications to the Recursive RANSAC algorithm in order for it to work. The states that are being used in this modified Recursive RANSAC algorithm are shown in **Equation 1**.

Equation 1

$$x = \begin{pmatrix} p_{n,int} \\ p_{e,int} \\ V_{a,int} \\ \psi_{int} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

One of the main changes to the Recursive RANSAC algorithm is the use of and Extended Kalman Filter (EKF) instead of a KF to account for the nonlinear state space equation and output equation. The state-space equation that is being used by the EKF is shown in **Equation 2**.

Equation 2

$$f(x) = \begin{pmatrix} x_3 * cos(x_4) \\ x_3 * sin(x_4) \\ 0 \\ 0 \end{pmatrix}$$

Finally the output equation that is being used by the EKF is shown in **Equation 3** where the two outputs are range and bearing.

Equation 3

$$h(x) = \begin{pmatrix} \sqrt{(x_1 - p_n)^2 + (x_2 - p_e)^2} \\ \tan^{-1}\left(\frac{x_2 - p_e}{x_1 - p_n}\right) - \psi \end{pmatrix}$$

As discussed in the previous section one of the primary rolls of the RANSAC algorithm is to identify an initial condition that has the most number of inliers with the current measurement. Since the Recursive RANSAC algorithm that has been developed requires linear system in order to identify the initial condition, I had to create a new method of determining the initial condition.

Solving the Nonlinear Equation Backwards in Time

The new method implemented to find the initial conditions is to solve the nonlinear state equation backwards in time. The general form of the equation is show in **Equation 4** below.

Equation 4

$$x[k-1] = x[k] - dt * f(x)$$

This equation is placed inside a for loop that iterates the length of the measurement window. In order to solve this equation backwards in time we need a set of states to start with which represent the current time or ending conditions of the aircraft. The method for determining the ending conditions will be discussed in the following section.

Finding the Ending Conditions

The north and east positions of the ending condition can be determined directly from range and bearing and are shown in **Equations 5 & 6** below.

Equations 5 & 6

$$pn_0 = pn_{own} + r * cos(\phi + \chi_{own})$$

$$pe_0 = pe_{own} + r * sin(\phi + \chi_{own})$$

The other two states, velocity and heading, are not as easily determined. In order to determine velocity and heading we need to use the current measurement and one other randomly selected measurement. The north and east positions are calculated for both these measurements and then a straight line is drawn between the two

positions. The heading can be determined by the direction that this line points. The velocity can be determined because we know the distance between these two points and we also know at what time the measurements were taken. The equations for both of these states are shown below in **Equations 7 & 8**.

Equations 7 & 8

$$Va_0 = \frac{sqrt((pn_1 - pn_2)^2 + (pe_1 - pe_2)^2)}{t_1 - t_2}$$

$$\psi_0 = tan^{-1}\left(\frac{pe_1 - pe_2}{pn_1 - pn_2}\right)$$

Now that we have found all four states, north position, east position, velocity, and heading, we can now solve the nonlinear state equation backwards in time. After we solve the nonlinear equation backwards in time we see which of all the measurements are inliers to this trajectory. **Figure 2** below shows what a single iteration of the RANSAC algorithm might produce.

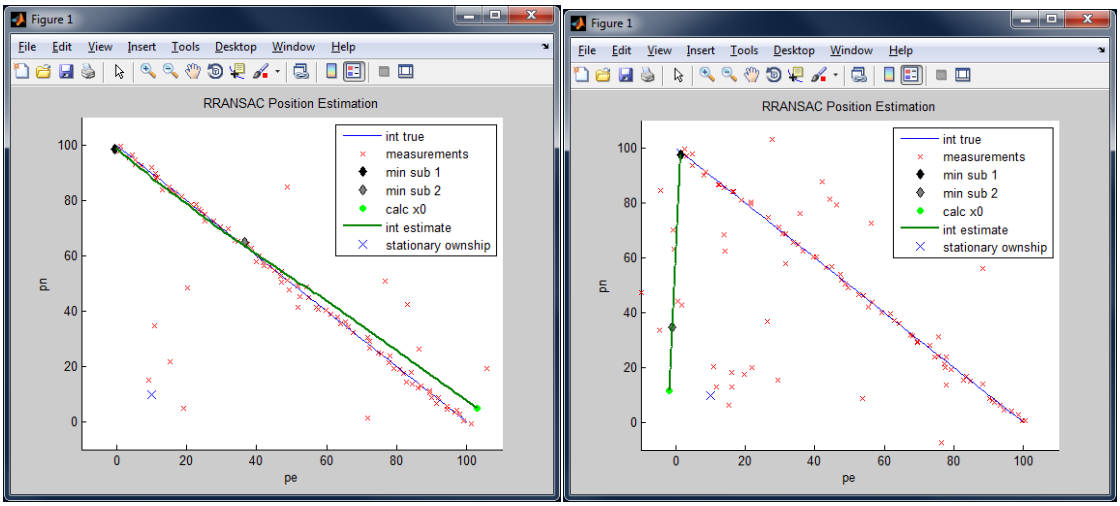

Figure 2: Both images on the left and the right show the results for 1 iteration of the RANSAC algorithm. The line in blue is the true track of the intruder and the green line is the best estimate of the intruder. The red x's are the measurements of the intruders position. The light green circle is the initial conditions that the RANSAC algorithm calculates.

The above steps are repeated several times and the iteration with the most inliers is used to choose the initial condition of the intruder. **Figure 3** that follows shows the best track out of 100 iterations of the RANSAC algorithm. The image on the left shows the north and east positions. The image on the upper right shows the velocity and the image on the bottom right shows the heading. Notice that the estimated track, shown in green follows very closely to the true track shown in blue.

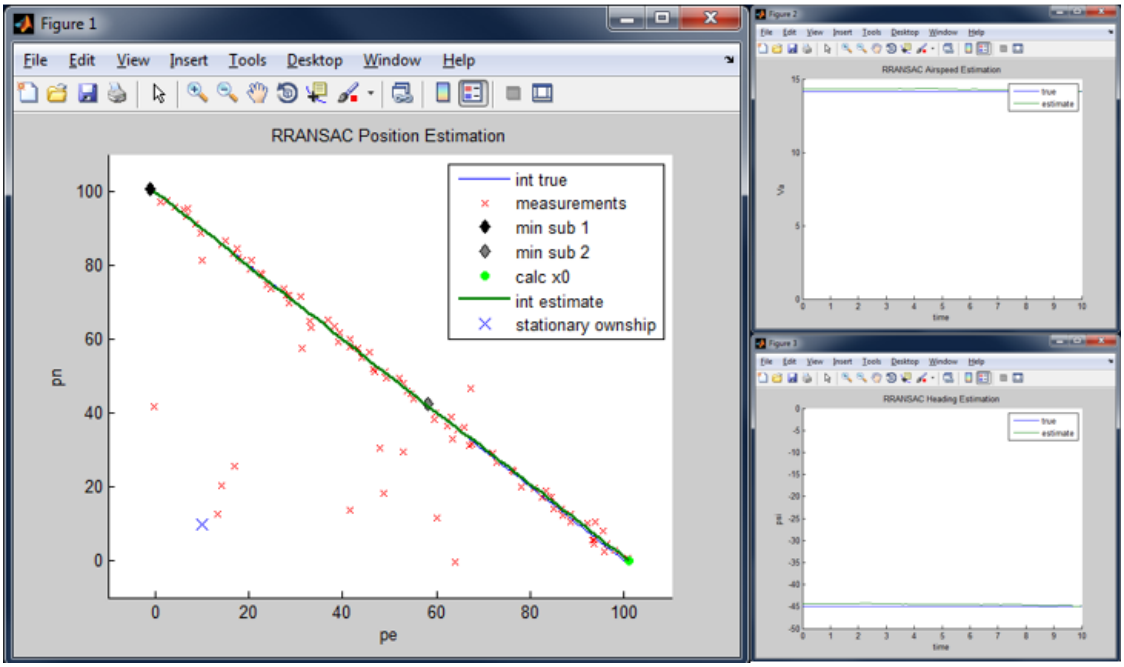

Figure 3: All three images are the result of 100 iterations of RANSAC and the iteration with the most inliers is selected. The line in blue is the true track of the intruder and the green line is the best estimate of the intruder. The red x's are the measurements of the intruders position. The light green circle is the initial conditions that the RANSAC algorithm calculates. The image on the left shows the north and east positions. The image in the upper right is the velocity of the intruder. The image on the bottom right is the heading of the intruder.

The previous **Figure 3** shows what the Recursive RANSAC algorithm does when a measurement is received that is not an inlier to an existing model. As can be imagined, there will be many models being formed. New models will be created and old models will be replaced. **Figure 4** that follows shows how the Recursive RANSAC algorithm works continuously in a simulation. Notice that the estimates are not as good at the beginning of the track, but soon the error becomes small. In each of the figures, measurements are not being received until about 20 seconds into the simulation, because the intruders are outside the range of the radar sensors.

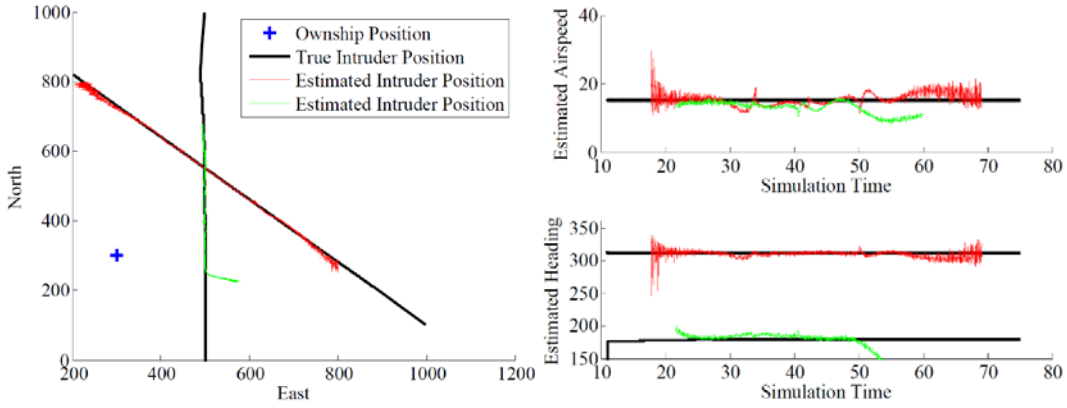

Figure 4: The three images show the true states of two intruders in black and the estimated states in red and green. The left image shows the north and east positions. The upper right image shows the airspeed and the bottom right image shows the heading.

Conclusion

In conclusion the Recursive RANSAC algorithm has been a successful method in estimating the states of multiple intruders. New techniques were required to alter the RANSAC algorithm so that the initial conditions of a measurement could be found. These alterations were required because Recursive RANSAC was developed specifically for linear state space models. Further research will be done to address overtaking intruders and intruders that don't fly in straight and level flight.

Acknowledgements

First I want to thank Utah NASA EPSCoR for sponsoring my research. I also want to thank the industrial companies who have contributed to my funding via the C-UAS. Finally I want to acknowledge my mentor Dr. Timothy McLain for helping me identify this research topic.

References

[1] P. Bergen, “A dangerous new world of drones,” October 2012, http://www.cnn.com/2012/10/01/opinion/bergen-world-of-drones/index.html?hpt=hp c2. 1

[2] Office of the Secretary of Defense, “Unmanned systems integrated roadmap FY2007-2032,” Department of Defense, Tech. Rep., December 2007. 1

[3] ——, “Unmanned systems integrated roadmap FY2011-2036,” Department of Defense, Tech. Rep., December 2011. 1

[4] UAS Task Force: Airspace Integration Integrated Product Team, “OSD airspace integration plan for unmanned aviation,” Department of Defense, Tech. Rep., March 2011. 1, 3, 4

[5] FAA Order 7610.4. Federal Aviation Administration, 2004. 2

[6] R. E. Weibel and R. J. Hansman, “Safety considerations for operation of unmanned aerial vehicles in the national airspace system,” MIT International Center for Air Transportation, Tech. Rep. ICAT-2005-1, March 2005. 2, 3

[7] K. Dalamagkidis, K. Valavanis, and L. Piegl, “Current status and future perspectives for unmanned aircraft system operations in the US,” Journal of Intelligent and Robotic Systems, vol. 52, pp. 313–329, 2008. 2

[8] C. Geyer, S. Singh, and L. Chamberlain, “Avoiding collisions between aircraft: State of the art and requirements for UAVs operating in civilian airspace,” Carnegie Mellon University, Jan 2008. 2, 7

[9] FAA Order 1110.150. Federal Aviation Administration, 2008. 3

[10] Federal Aviation Administration, “FAA makes progress with UAS integration,” October 2012, http://www.faa.gov/news/updates/?newsId=68004. 3