

APPLICATIONS OF RADAR TRACKING USING COMPUTER SCIENCE

Jay Bhalodi, Jeff Cao, Lily Healy, Wendy Lin, Tuling Ma, Zara Mannan, Brandon Millman,
Zachary Purdy, Divya Sharma, Mimi Xu

Advisor: Randy Heuer

Assistant: Zack Vogel

ABSTRACT

Raw output from radars is often muddled by intrinsic process and measurement noise. To create an effective radar tracking system, we used linear algebra and computer science to effectively apply a Kalman filter to process raw radar data. We also applied trigonometric and other mathematical methods to implement additional features in our tracking system including collision prevention and target interception. Both our filter implementation and target tracking system proved to be more accurate and more precise than the raw input data.

INTRODUCTION

Radar tracking is a technique used to track any moving object, which becomes important in real-life applications such as guiding planes to smooth landings or preventing collisions. To track the object of interest, a radar sends out microwaves that echo off an object and send back data that a computer uses to calculate its position.

Despite its growth, the supply of data cannot accurately reflect the true flight plan by itself. Two forms of ‘noise’ disturb accuracy; process noise such as wind influences the target, while measurement noise comes with the data obtained from the radar.

To help the calculations, a mathematical system called the Kalman filter uses an iterative process to filter out the noise. The Kalman filter can estimate the state of the system, while taking into account the factor of time and inherent noise. In our project, we needed to combine the data obtained by the radar systems as well as the interpretation of an algorithm, in order to obtain estimates that would, for example, prevent collisions between two targets. We chose the Kalman filter due to its efficiency because the computations are relatively simple to implement, and it does not require that previous data be retained.

Rudolf Kalman

Rudolf E. Kalman was a control theorist who was born in Hungary. He received a bachelor’s and master’s degrees in science from the Massachusetts Institute of Technology and a doctorate of engineering science from Columbia University. Kalman was known for his work with filter systems, including the Kalman filter, which is widely used in navigational and guidance systems, radar tracking, sonar ranging, and satellite orbit determination. The Kalman filter was one of Kalman’s most significant contributions to control and filter theory [1].

Kalman filter

In his paper “A New Approach to Linear Filtering and Prediction Problems”, Dr. Kalman introduced his concept of a filter that could appropriately handle random noise and output accurate predictive data [2].

In addition to giving an accurate estimate of the target’s position and velocity, the Kalman filter also has another key advantage – it is highly efficient in terms of memory usage. The filter only retains one prediction and one set of radar measurements at any single time. Each new measurement is used to form a new prediction, and then is no longer needed. This aspect of the Kalman filter is especially useful when many measurements are taken.

Model

The foundation of the Kalman filter is the following model that shows how the state vector is updated and the measurement vector is obtained [2].

$$x_{k+1} = \Phi x_k + q_k \quad (1)$$

$$y_k = Hx_k + r_k \quad (2)$$

Equation 1 describes the actual state of the target. The current state vector, x is multiplied by the transition matrix, Φ . This product is then added to vector q , the process noise- the inherent variation of the target’s movement, to produce the state vector x for the next time step. At each time step, q is a single instance of a randomly varying state noise. Because each q is random, it cannot be known. However, the distribution of q can be understood and represented by the covariance matrix Q .

Vector y represents the measurement and is calculated by multiplying matrix x by measurement vector, which transforms the state into the form of the measurement vector. Then the measurement noise vector, r , is added to this product. Like q , r is a randomly varying vector that can never be known. Again, however, its distribution, R can be understood and modeled.

Predict

The first step in the Kalman filtering process is the prediction step. The Kalman filter will predict the state of the target by using a state estimate from a previous time step to predict its state of the current time step. The Kalman filter uses a set of matrix equations to find a prediction of the subsequent state vector [3].

$$X_{K+1|K} = \Phi \hat{X}_{K|K} \quad (3)$$

$$P_{K+1|K} = \Phi P_{K|K} \Phi^T + Q \quad (4)$$

In this equation $X_{K+1|K}$ is the state matrix at time $K+1$, based on the estimated value at time K .

The state covariance matrix (P) is an estimate of the uncertainty of the state vector. It is also estimated in the prediction step, and is based on the transition matrix and our knowledge of the random distribution of the process noise.

Update

In the update step, a new set of data is gathered, and our results from the prediction step are compared with them. First the Kalman gain matrix, a weighting factor that controls how much the filter values the measurements, is determined and updated using the state covariance matrix, the measurement matrix, and the measurement noise, as shown in Equation 5. It determines how much the filter values the data residual when calculating a new state vector in Equation 6. Lastly, the state covariance matrix is also updated in Equation 7, based again on the Kalman gain matrix. At this point, the filter returns to the predict step, and the process repeats [2].

$$K_k = P_{k|k-1} H^T [H P_{k|k-1} H^T + R]^{-1} \quad (5)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k [y_k - H \hat{x}_{k|k-1}] \quad (6)$$

$$P_{k|k} = [I - K_k H] P_{k|k-1} \quad (7)$$

IMPLEMENTATION

The position of our targets is determined by the information obtained from radar measurements, and the algorithm corrects this data to filter variant noise. Although a radar system can measure the position of an object, using an algorithm will increase accuracy and provide an estimate of a target's velocity.

To compensate for these noisy radar measurements, our implementation of the Kalman filter utilizes an efficient algorithm to accurately predict the target's true position and velocity from the position measured by radar. As each new measurement is obtained by radar at each time step, the Kalman filter updates its previous prediction, eventually formulating a more accurate model, despite the noisy radar measurements.

In our case, the dimensions we are working with are time, x-coordinate and y-coordinate. Thus, our State vector x denotes the position and velocity as follows [3]:

$$x = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}. \quad (8)$$

In this case Φ is designed as follows to update x and y coordinates by adding the product of the given constant velocities (\dot{x} and \dot{y} , respectively) and the change in time (Δt):

$$\Phi = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

If we multiply these matrices we get the following equation, which shows how the transition matrix transforms the state matrix:

$$\Phi x_{k|k} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} x \\ x + \dot{x} \Delta t \\ y \\ y + \dot{y} \Delta t \end{bmatrix} \quad (10)$$

In our case, we are only measuring the position of the target, so our measurement vector y only consists of the measured x -coordinate and y -coordinate as follows:

$$y = \begin{bmatrix} x_m \\ y_m \end{bmatrix} \quad (11)$$

Thus our measurement matrix must transform our state matrix so that only contains position elements. Thus it takes the following form:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (12)$$

To program this filter, we first needed to read the data from the text file. We took the first two lines of measurements and initialized the state vector. Then we had a loop that would break after the last line was read from the text. Before we ran the filter, we pre-filtered our measurements so that they would be in a form usable by the filter which was usually x and y coordinates centered at the origin. This pre-filtering occurred in each iteration. Then we ran the Kalman filter equations with our new data. Finally we output values in the state vector, or our post-filtered data.

APPLICATIONS OF KALMAN FILTER

Polar Coordinate Transformations

The focus of our research was to apply the Kalman filter to more realistic situations. Since radars report range and bearing, the measurements need to be converted to Cartesian coordinates using Equations 13 through 15. In the equations, range (r) represents the distance between the source and satellite, and bearing represents the angle (α) from true north. This is because our Kalman filter program is designed to handle calculations made in the Cartesian system.

$$\theta = \frac{\pi}{2} - \alpha \quad (13)$$

$$x = r \cos \theta \quad (14)$$

$$y = r \sin \theta \quad (15)$$

In addition to these polar transformations, it is important to consider that the measurements are not centered perfectly along the coordinate axis. Due to the standard deviation of range and bearing, the overall area in which the target may fly forms a tilted ellipse. The elliptical axes are not parallel to those of the Cartesian plane. Thus, it is necessary to update the matrix containing information pertinent to measurement error, also known as the “R” matrix. The R matrix represents the variance of the X and Y coordinates of the transformed measurements, which depends on the location of the target at a point in time. Due to this correlation, R matrix is always changing for each position of the target. Fig. 1 displays a visual representation of the error calculation that the R matrix compensates for; the ellipse shows the uncertainty of the point’s true position.

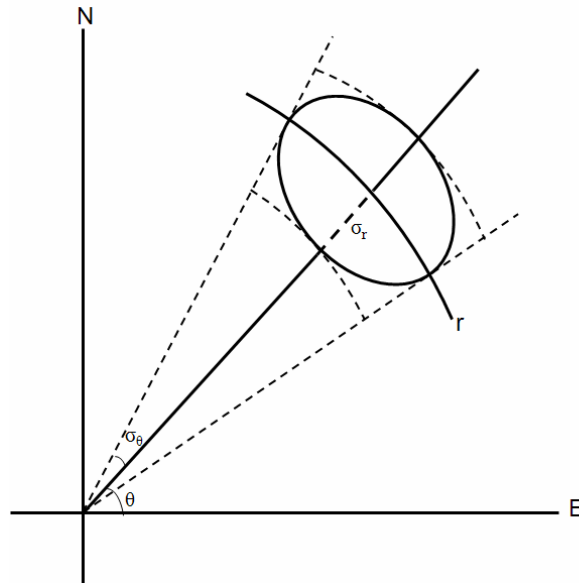


Fig. 1
Visual representation of estimation method

Since the R matrix is used in calculating the Kalman gain matrix, it must be adjusted beforehand [3]. These adjustments and transformations allow us to predict targets' given range and bearing measurements. Fig. 2 displays the residuals of our results – the residuals are the distances between our filter's predicted positions and the measurement. Our results are much more accurate than the measured values, as evidenced by the lower residual values.

$$\sigma^2_x = \sigma^2_r \cos^2 \theta + r^2 \sin^2 \theta \sigma^2_\theta \quad (16)$$

$$\sigma^2_y = \sigma^2_r \sin^2 \theta + r^2 \cos^2 \theta \sigma^2_\theta \quad (17)$$

$$\sigma^2_{xy} = \frac{1}{2} \sin 2\theta [\sigma^2_r - r^2 \sigma^2_\theta] \quad (18)$$

$$R = \begin{bmatrix} \sigma^2_x & \sigma^2_{xy} \\ \sigma^2_{xy} & \sigma^2_y \end{bmatrix} \quad (19)$$

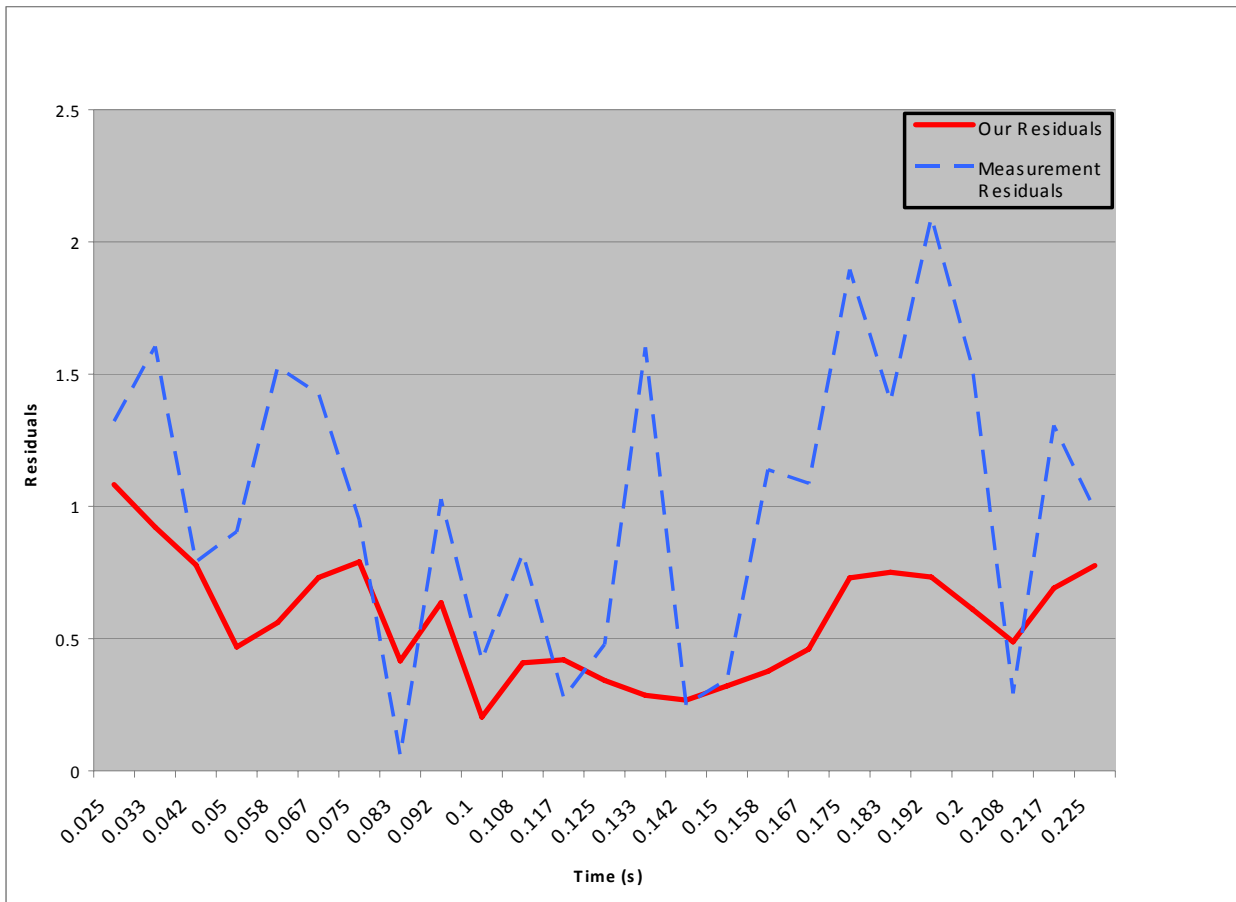


Fig. 2
Polar conversion scenario data residuals

Multiple Radars

Another adaptation of our implementation of the Kalman filter is its ability to handle input from multiple radars located in different places. With this scenario, the object was tracked with two radars – depending on which one was in range – making the incoming data more difficult to work with than previous cases. This situation also introduced variable time, since the data was no longer coming in on a regular interval. Fig. 3 shows a visual representation of the multiple radar scenario.

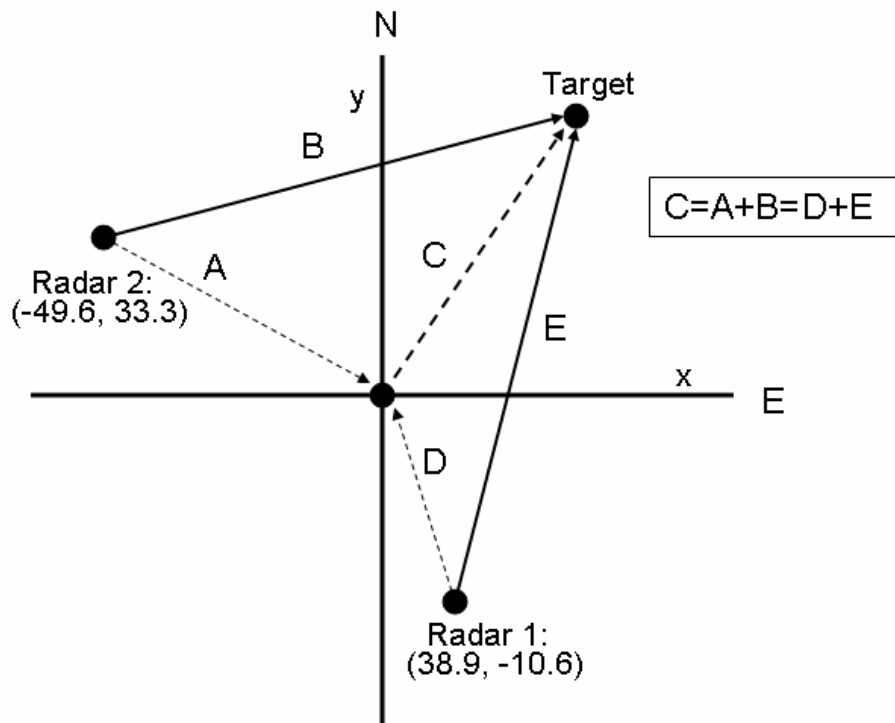


Fig. 3
Multiple radar scenario

Implementing the code for this situation required two major modifications to our filter. First, in order to calculate the newly variant time step, we had to add update the state transition (Φ) matrix, which used t each time the loop ran. In addition, since our raw data was now coming in from two radars located in different places, we had to tag the data we received to assign it to a radar, and then reconcile that data with a single frame of reference. To do this, we simply created a coordinate plane in which the two radars were placed off of the point of origin, and then offset the data we received by the distance between the origin and the radar.

Our results shown in Fig. 4 followed a different trend than past cases because of the addition of a second radar. The filter received data from only the first radar in the beginning; thus, the jump in distance residual in the middle of the graph (at 0.1 hrs time) is due to inaccurate

data from the second radar. The switch shows how quickly the filter compensated and brought the residual back down to under 0.5. This is a lower margin of error than the previous scenarios because having two radars allows us to obtain a cross-fix on the target much more accurately.

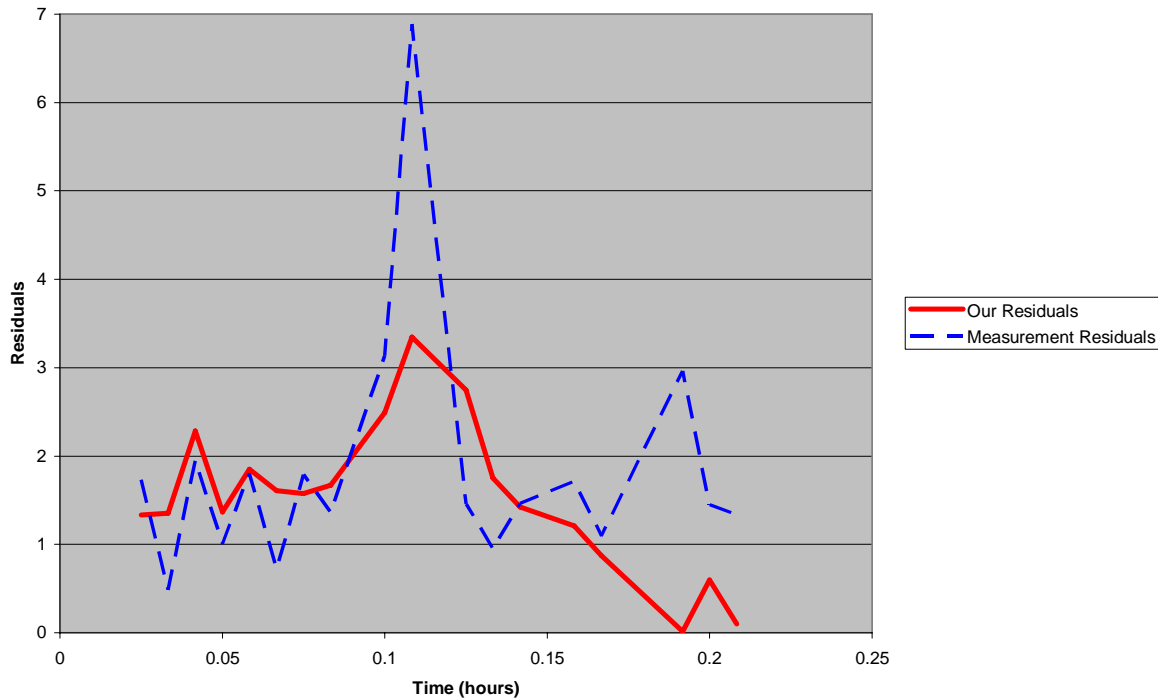


Fig. 4
Multiple radar scenario data residuals

Change in the Course of the Object

This scenario is different from the previous ones in that the target maneuvers. It begins on a linear course and maneuvers to different linear courses multiple times. Initially, the Kalman filter follows the target on its linear path. Once the our program recognizes that the radar's measurements fall outside a pre-set range of standard deviations, it will reset itself with new initial conditions matching those of the target's new course. This method is known as re-initialization.

In order to detect a change in direction, we used a form of the state covariance matrix that only includes position elements, which measured the distance between the measured point and the predicted value (in the number of standard deviations).

We were able to derive Equation 20 by using the values, highlighted below, from the state covariance matrix.

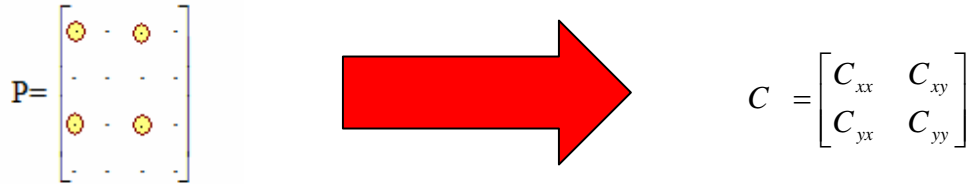


Fig. 5
Data source of inverse C matrix

Equation 20 is displayed below:

$$C^{-1} = \begin{bmatrix} C_{xx}^{-1} & C_{xy}^{-1} \\ C_{yx}^{-1} & C_{yy}^{-1} \end{bmatrix} \text{ and } C_{xy}^{-1} = C_{yx}^{-1} \quad (20)$$

In Equation 21, r is the vector between the measured point and the predicted point [4].

$$(n\sigma)^2 = r^T C^{-1} r \text{ where } r = (\vec{x} - \bar{x}) \quad (21)$$

Multiplying this matrix out yields:

$$(n\sigma)^2 = r^T C^{-1} r = (x,y) \begin{bmatrix} C_{xx}^{-1} & C_{xy}^{-1} \\ C_{yx}^{-1} & C_{yy}^{-1} \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = x^2 C_{xx}^{-1} + 2xy C_{xy}^{-1} + y^2 C_{yy}^{-1} \quad (22)$$

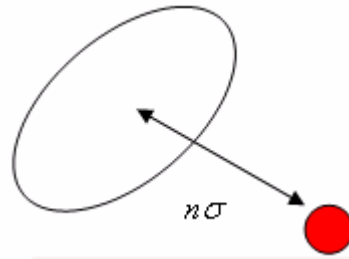


Fig. 6
Distance between a point with an error and another point measured in σ s

The distance calculated determines if the target changed direction. In this particular scenario, if the number of standard deviations is greater than two, for four radar measurements in a row, we can assume that the target maneuvered. Fig. 6 represents this distance visually. We are approximately 99.95% sure that the target has maneuvered, as determined by the two-dimensional standard Gaussian curve. As a result, the filter reinitializes and resets with the following data sets.

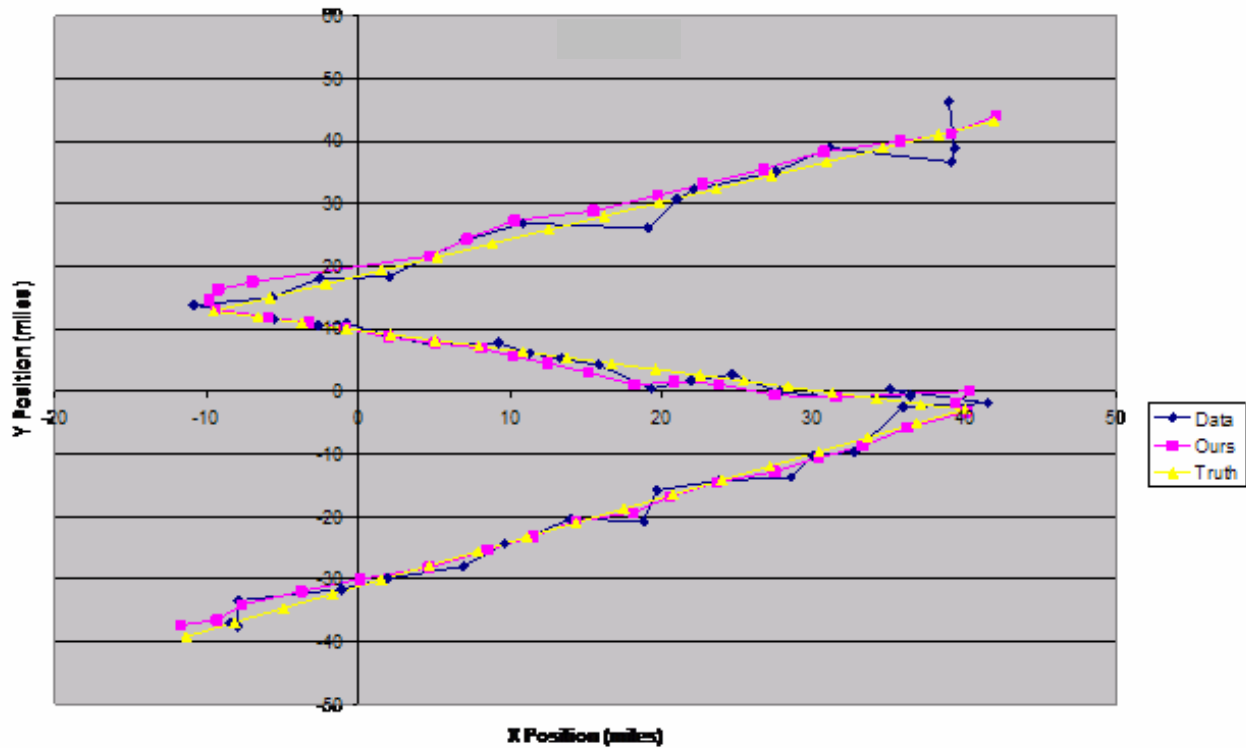


Fig. 7
Position of Target given by Data, Our Calculations, and True Values

Fig. 7 depicts the position given by data, our calculations, and the actual position. Although the measurements obtained were noisy, the Kalman filter provided accurate estimates that are comparable to the actual position. The darkest line, which is the position given by raw data, is irregular and deviates from the actual path many times; our results demonstrate that the Kalman filter proved to be quite effective in providing an accurate measure of the target's location.

The residuals, displayed below in Fig. 8, show that the filter can recover quickly after each maneuver is made.

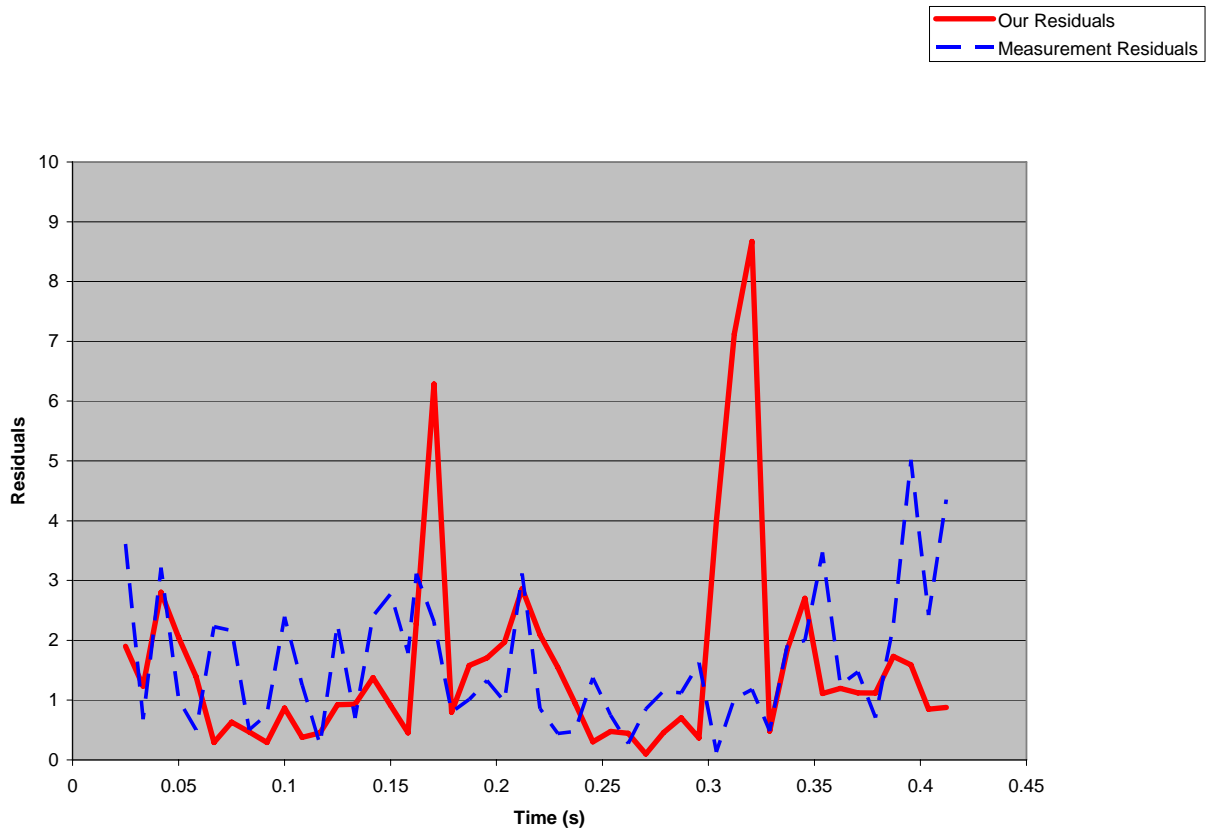


Fig. 8
Maneuvering target scenario residuals

Intercepting a Target

Another useful application of the Kalman filter is for intercepting targets. This involves projecting the path of the target and directing the interceptor to travel on the shortest possible interception course at each time step. In order to do this, we used the Kalman filter estimates to predict the target's course and velocity, and then calculated the bearing angle the interceptor would have to travel at to intercept the target.

The scenario is illustrated below in Fig. 9. Point I represents the interceptor, while Point T represents the target. Point P is the projected point of interception at time t . To calculate the bearing angle that the interceptor would have to travel at, α , we first calculated the value of τ using the Law of Sines, as shown in Equations 23 and 24.

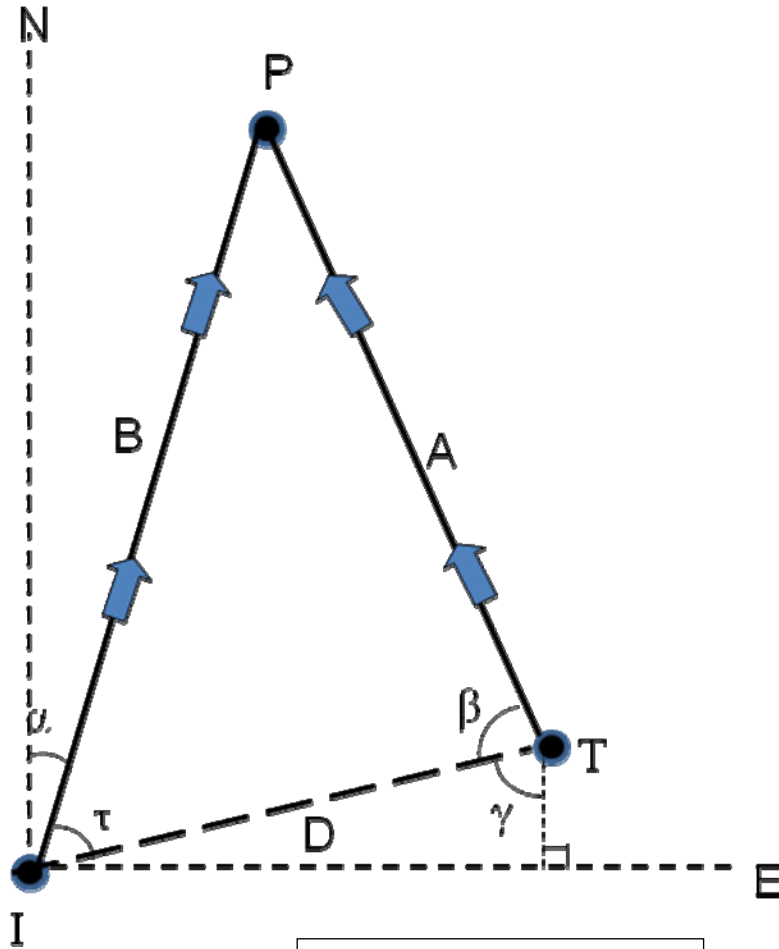


Fig. 9
Intercepting targets scenario

$$\frac{\sin \tau}{A} = \frac{\sin \beta}{B} \quad (23)$$

$$\tau = \sin^{-1}\left(\frac{A}{B}\right)\sin \beta \quad (24)$$

Then we substituted in values for $\frac{A}{B}$ and β using Equations 25 and 26, and were able to solve for τ using Equation 24.

$$\frac{A}{B} = \frac{v_T t}{v_I t} = \frac{v_T}{v_I} \quad (25)$$

$$\beta = \cos^{-1}\left(\frac{\langle x_I - x_T, y_I - y_T \rangle \bullet \langle v_{T_x}, v_{T_y} \rangle}{D \|\langle v_{T_x}, v_{T_y} \rangle\|}\right) \quad (26)$$

In Equation 27, D , the distance between the interceptor and the target, can be calculated through the use of the distance formula.

$$D = \sqrt{(x_T - x_I)^2 + (y_T - y_I)^2} \quad (27)$$

Also, γ is the sum of α and τ , due to the parallel vertical dashed lines. As a result, the bearing angle, α , can be calculated in Equations 28 and 29.

$$\gamma = \alpha + \tau = \cos^{-1}\left(\frac{y_T - y_I}{D}\right) \quad (28)$$

$$\alpha = \cos^{-1}\left(\frac{y_T - y_I}{D}\right) - \tau \quad (29)$$

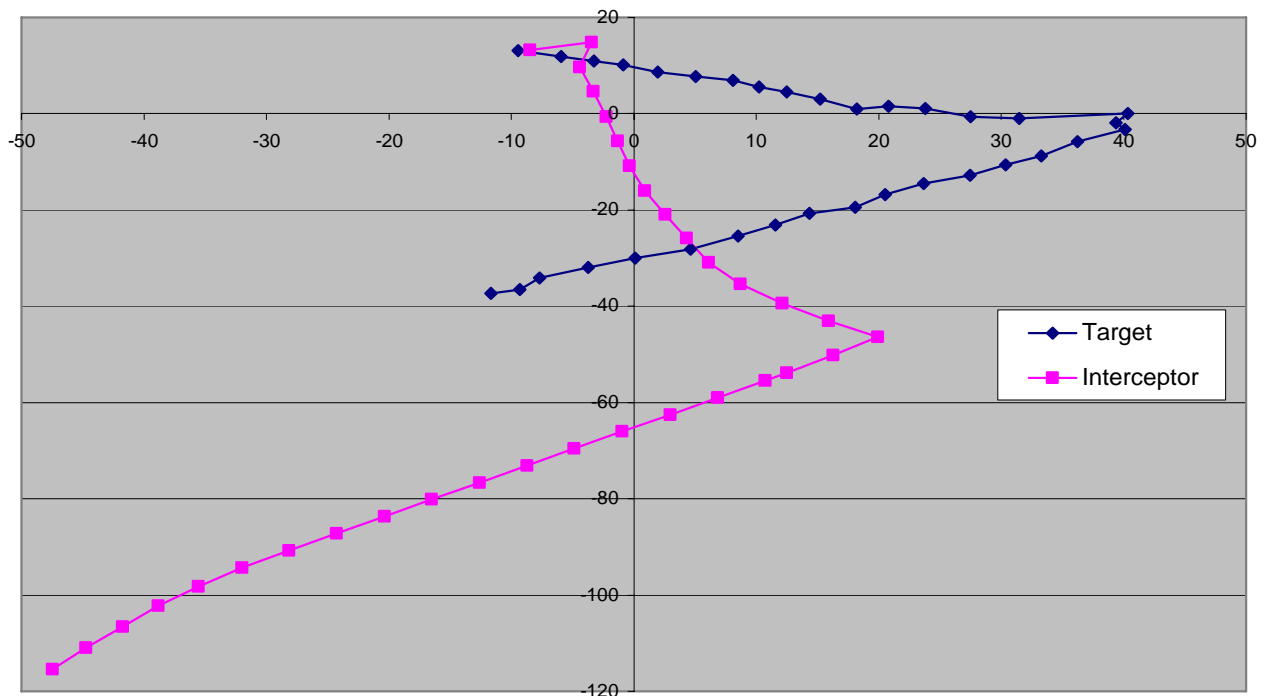
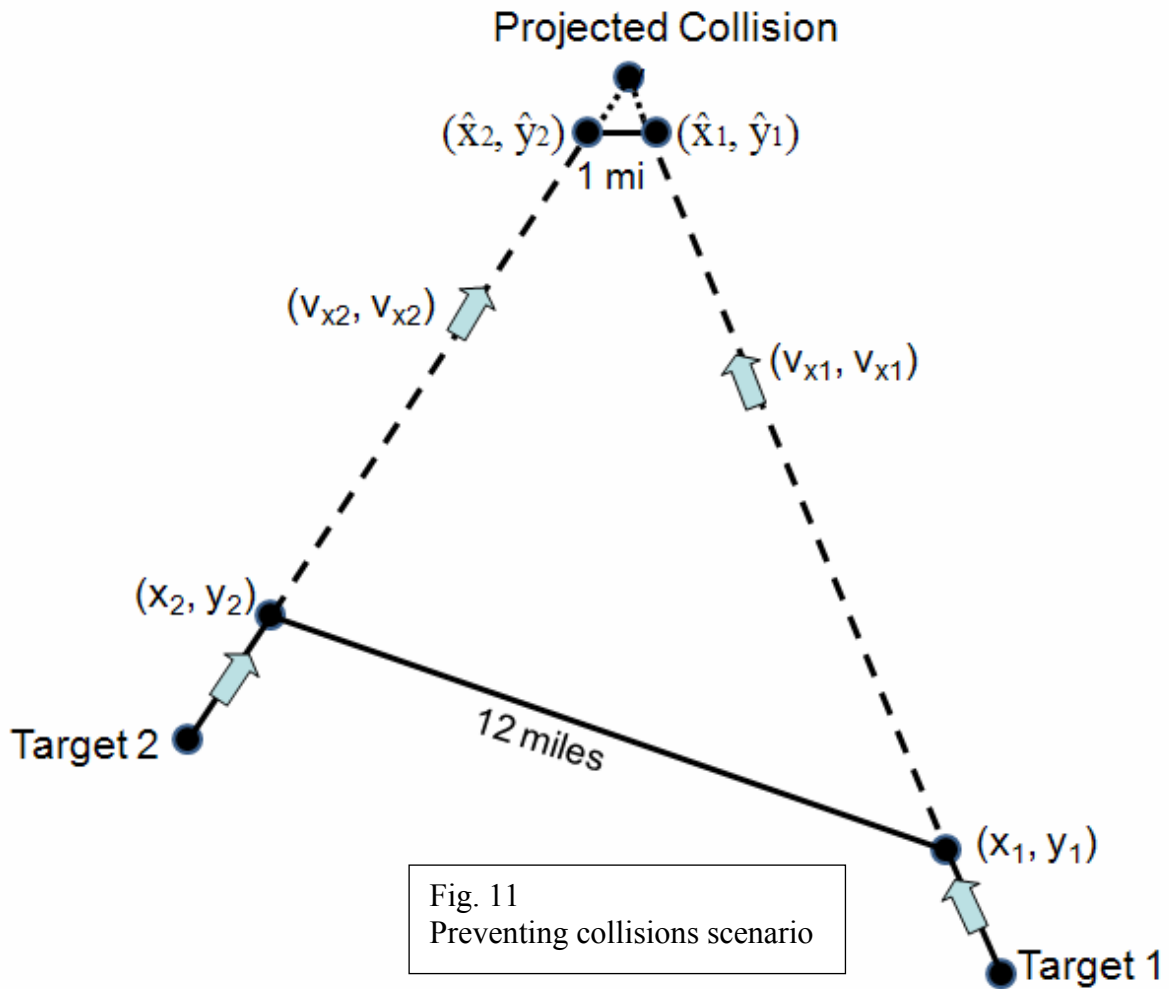


Fig. 10
Intercepting targets' positions

In Fig. 10 we are looking at a graph of the position of the target and the interceptor dispatched to catch it.



Let (x_1, y_1) represent the current position of target1 and (x_2, y_2) represent the current position of target2. Also, let (v_{x1}, v_{y1}) and (v_{x2}, v_{y2}) represent the component velocities of target1 and target2 respectively.

Then use the distance formula with the two positions to find if the following expression holds true:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \leq 12 \quad (30)$$

If the above is true, calculate future positions of each target to find whether,

$$\sqrt{(\hat{x}_2 - \hat{x}_1)^2 + (\hat{y}_2 - \hat{y}_1)^2} < 1 \quad (31)$$

Where \hat{x} represents the future x-position and \hat{y} the future y-position

So let:

$$\hat{x}_1 = x_1 + v_{x1}\Delta t \quad (32)$$

$$\hat{y}_1 = y_1 + v_{y1}\Delta t \quad (33)$$

$$\hat{x}_2 = x_2 + v_{x2}\Delta t \quad (34)$$

$$\hat{y}_2 = y_2 + v_{y2}\Delta t \quad (35)$$

$$\Delta x = x_2 - x_1 \quad (36)$$

$$\Delta y = y_2 - y_1 \quad (37)$$

$$\Delta v_x = v_{x2} - v_{x1} \quad (38)$$

$$\Delta v_y = v_{y2} - v_{y1} \quad (39)$$

Where:

$$\Delta t = \text{time}$$

Substitute the variables in Inequality 31 with the variables in Equations 32-39 as shown below:

$$\sqrt{(\Delta x + \Delta v_x t)^2 + (\Delta y + \Delta v_y t)^2} < 1 \quad (40)$$

Expand:

$$\Delta t^2 (\Delta v_x + \Delta v_y) + \Delta t (2\Delta v_y + 2\Delta v_x) + (\Delta x^2 + \Delta y^2) - 1 < 0 \quad (41)$$

Solve for Δt using quadratic formula:

$$\Delta t = (-b \pm \sqrt{b^2 - 4ac}) / 2a \quad (42)$$

Where:

$$a = \Delta v_x + \Delta v_y \quad (43)$$

$$b = 2\Delta v_y + 2\Delta v_x \quad (44)$$

$$c = \Delta x^2 + \Delta y^2 - 1 \quad (45)$$

Use the determinant to find the number of solutions. If:

$b^2 - 4ac < 0$	targets never travel within a mile of each other
$b^2 - 4ac = 0$	targets get within a mile of each other at one point
$b^2 - 4ac > 0$	targets get within a mile of each other at one point

Table 1
Using determinant to find proximity of targets to each other

Thus, if the determinant yields a positive answer, a warning should be issued to pilots of both targets to avoid collision. Since, the warning would occur as soon as they are 12 miles apart, it would give sufficient time for evasive action. As shown in Fig. 12, the two targets come within 12 miles of each other. At this point, it is projected that they would come within one mile of each other, so the lines stop, indicating that the targets need to maneuver to avoid a collision.

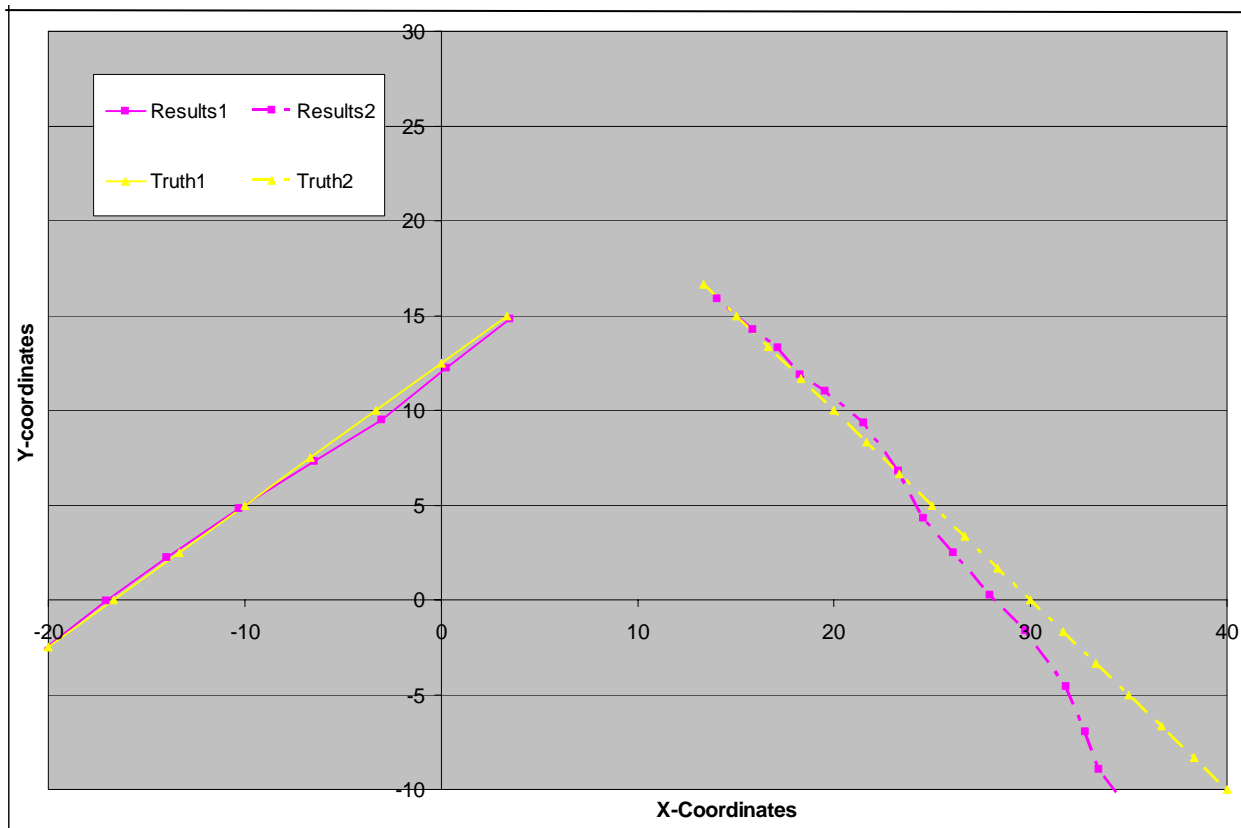


Fig. 12
Position of two targets given by our results and true values

SUMMARY

The Kalman filter is an effective method of improving the accuracy and precision of measured coordinates. Even when presented with advanced situations, such as tracking multiple or maneuvering targets, intercepting the targets, and implementing collision avoidance systems; the Kalman filter was able to adapt and remain a highly effective tool.

Effectiveness of Kalman filter

From our data, we can see that the Kalman filter is a very effective tool, even more so when given noisy measurements. In all of our residual graphs we can see that after the first few data points, the residuals of the values we have gotten from the Kalman filter are less than residuals of the data. In the following chart, the average residual for each case is listed. The first 4 residuals are omitted in each case, because the Kalman filter is most effective when it has run for a longer period of time:

	Our Results' Average Residual	Average Data Residual
Polar Coordinate Transformations	0.523578	0.995015
Multiple Radars	1.518172	2.008087
Change in the Course of the Object	1.033573	1.57557
Intercepting Targets (Target 1)	1.143326	1.482508
Intercepting Targets (Target 2)	1.076956	1.420316
Preventing Collisions (Target 1)	2.263782	2.348858
Preventing Collisions (Target 2)	4.051644	4.264486

Table 2
Average residuals for each case

Clearly, the average residual of our results is smaller than the average residual of the data. This shows that a Kalman filter out performs a radar tracking system only based on measurements.

This accuracy is especially important for the very practical scenarios like intercepting targets and preventing collisions. In these cases, results exclusively based on the last two measurements would give the same level of accuracy for each step, an accuracy that would be no better than the measurement noise and state noise. This lack of accuracy could lead to a failure to intercept a target or to prevent a collision of two planes. The Kalman filter, on the other hand, becomes more accurate and puts less value in the measurements the longer it runs. Thus the Kalman filter is an extremely helpful tool in any radar tracking endeavor.

Possible Improvements

Despite the minor percentage error of our data, measurements should continue to be improved. In a real-world situation, objects do not always fly at the same velocity and thus acceleration must be accounted for. Also, our equations only account for linear flight. These little details are important, especially since a tiny mistake can be multiplied to a grand scale in the real world.

REFERENCES

- [1] [IEEE] Institute of Electrical and Electronics Engineers. 2003 Jan 23. Rudolf E. Kalman, 1930-. IEEE History Center.
<http://www.ieee.org/web/aboutus/history_center/biography/kalman.html>. Accessed 2008 July 21.
- [2] Kalman, R. E. 1960. A New Approach to Linear Filtering and Prediction Problems. ASME Journal of Basic Engineering 1960 March.
- [3] Blackman SS. 1986. Multiple-Target Tracking with Radar Applications. Artech House, Inc.
- [4] Atwood B. 2003. Covariance and GLAST.
<<http://www-glast.slac.stanford.edu/software/AnaGroup/WBA072003-Covariance.pdf>>. Accessed 2008 July 21.