Flight Path Reconstruction for Free Falling Units Using Extended Kalman Filter

15 January 2017

Tanay Rastogi
Masters System Control and Robotics
910721-6138
tanay@kth.se

***Abstract*—In order to know the exact position of a FFU, we need to know the flight path trajectory of the FFU. For the estimation we need a motion model which includes the dynamics of the motion and measurement from the sensor. For the localization of the units, the GPS sensor is mostly commonly used sensor. But GPS suffers from low precision and also have constrain on the functionality of the sensor at higher altitudes. The localization is hence proposed to be done using the IMU. The IMU contains a gyroscope and an accelerometer and by using the measurements from both the sensor, the position and velocity can be derived from them. The estimation of the positon using IMU is very noisy because of the approximation done during calculation. Hence, in order to minimize the effect of the noisy measurements and the motion, Kalman filter is used.**

Keywords—Free Falluing Units (FFU), filght path trajectory, Inertial Measurement Unit (IMU), GPS, Kalman filter.

I. INTRODUCTION

In order to do scientific measurement in the ionosphere region of the space, most of the time a Free Flying Unit (FFU) is used, to carry out the experiments. These FFU carry some scientific instruments relevant to the experiment. They are ejected from the rocket at certain height in the ionosphere, collect data and fall back to ground with the data stored in the memory drives. Most of the FFU are ejected follow are ballistic projectile motion and are under free fall from the moment they are ejected under the influence of miligravity.

The rocket program KTH use cylindrical disc shaped free falling unit (FFU) for conducting the experiments.[1][2]. These FFUs are then ejected from the rocket at the altitude of ~70-85kms above the earth surface. A similar experiment is going to be conducted, under the REXUS program[3] , called Wobble Control System for Spinning Free Falling Unit (WOLF). The aim of the program is to design a system to achieve a wobble free motion for the FFU, after ejected from the rocket.

In one of the previous experiment, SPIDER[4], the measurement data was corrupted because of the wobbling of the measurement instrument and was rendered useless for any analysis. Hence, the need for the wobble free motion in FFU. In order to stabilize the motion, a wobble control system is been developed by the WOLF team. For the system to work, it is important to know the local position of the FFU in its flight path.

As these FFU are freely falling towards the earth, there is no control over their flight path. The data of the experiment and the instruments are inside the FFU, hence it is important that we are able to recover them after their fall on the earth. To be able to recover them, is it important to have a good estimate of their flight trajectory so as to known their landing position on the earth.

The localization of the FFU is done via GPS, accelerometer and the gyroscope. The GPS receiver receives the location measurements and then send the coordinates to a ground station. But the GPS localization system is not efficient system and suffers with number of limitations. One of the limitation is on the operationalization of GPS system. The GPS receiver can work only for a range of altitudes. The commercially available GPS system works around the range of 6-7kms above sea level. Even for non-commercial GPS system at such high altitudes cannot transmit the position information to the ground station. Another issue here is that the localization of the FFU is completely based upon the GPS sensor alone. The commercial GPS receivers have high error rate on the position values. The main source of error is due to the inaccurate time keeping by the GPS receiver. The radio signal received by the GPS are travelling at speed of light and any small discrepancy in time can lead to wrong calculation of position.[5]

In order to reduce the dependency on the GPS and improve the estimation of the position, the localization can be done from measurements from accelerometer and gyroscope. The acceleration can be integrated in time to get the velocity and then integrate velocity in time to get the position. Measurements from the accelerometer are in body frame. The body frame axis is on the FFU and is rotating and moving with the FFU. Hence we need to convert the measurements of the accelerometer to static global frame. The gyroscope provides the angular acceleration in the body frame in all three axis. These measurements are used to get the transformation matrix to convert the accelerometer data from body frame to global frame[6]. The measurements from both the accelerometer and gyroscope are noisy. When this noisy measurements from the gyroscope is used to convert accelerometer frame, the noise from both the sensors get accumulated. Also, the integration of the position and velocity is approximated by rectangular rule, so that we can use the discrete measurements. This noisy measurements and the approximation gives a biased and noisy position and velocity values.

The estimation of position and velocity from the measurements from accelerometer and gyroscope is hence approached using Kalman filter.

For FFU, the motion model is assumed to be a linear projectile motion and the measurement model is formed as a linear model. The noise in the motion and measurement is independent white Gaussian noise, hence with these condition we can use the Kalman filter for estimation.

II. State of the Art

The main focus of the flight path recovery is to accurately identify the parameters and choosing a correct motion model to estimate the position of the aircraft. The motion model can be different based upon the aircrafts and the flying conditions. Most FFU follow are ballistic projectile motion which is linear in equations[7]. For the measurement of the state, mostly a strapdown inertial measurement unit (IMU) is used with the GPS. The measurements are then fused together from the IMU and GPS to get the position estimate[8]. The IMU, containing the accelerometer and the gyroscope, can assure only the short term accuracy as the error from the sensor starting to add up[9]. The GPS sensor is more reliable in long term but are less accurate. Hence a data fusion of the IMU and GPS can give a very accurate estimation of the path.

There are many popular methods available for the path reconstruction like Output-error method[10]. For the Output-error method the motion model should be precise and it only considers the error in the measurements. In order to include the motion noise in to the estimation procedure, Kalman filter is the preferred choice. There are various methods being developed for the flight path reconstruction using different variants of Kalman filter. In most of the cases the model proposed is non-linear and have used EKF or UKF for the states estimation[11][12]. But due to the reason that the linearization is approximated and thus filter do not converge to a consistent solution. In light of above issue, a linear Kalman filter is the best approach for the state estimation. Our approach is to create an algorithm that is linear in motion and measurement with some approximation and the noise in the model will be initialized such as to adjust those approximation to give a consistent state estimation.

III. Approach

A. *Kalman Filter*

A standard Kalman filter is a set of equations that implement a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance, when some conditions are presumed.[13].The filter have two different block, one is the prediction block and another is the update block. By running these two block iteratively, the Kalman filter algorithm converges to a consistent solution. The two blocks are shown in the **Fig 1**.

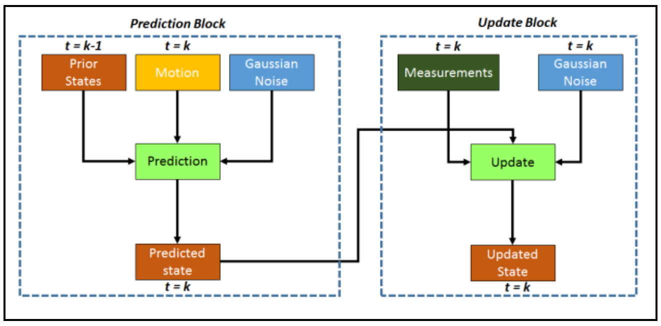

Fig 1 Kalman Filter Block Diagram

In accordance to the Bayes filter, the prior probability of the state is given by its mean and the covariance[14]. We predict the states at next transition by adding the motion to the prior with white Gaussian noise. The state transition model is given by,

$$x_k = Ax_{k-1} + Bu_k + \varepsilon \tag{1}$$

In the Eq(1), the x_k is the state at current iteration, x_{k-1} is the state at previous iteration, u_k is the motion added to the previous state to move them to next state and ε, called process noise, with zero mean and covariance *Q*. The state transition probability is,

$$p(x_k / u_k, x_{k-1}) = \det(2\pi Q)^{-1/2} \exp\{-\frac{1}{2}(x_k - Ax_{k-1} - Bu_{k-1})^T Q^{-1} (x_k - Ax_{k-1} - Bu_{k-1})\} \tag{2}$$

The predicted states are then integrated with the measurement to get the updated states. The measurement model is given by,

$$z_k = Hx_k + \vartheta \tag{3}$$

In the Eq(3), z_k is the measurement for the current state and ϑ, called the measurement noise, is white Gaussian noise with zero mean and covariance *R*. The measurement probability is given by,

$$p(z_k / x_k) = \det(2\pi R)^{-1/2} \exp\{-\frac{1}{2}(z_k - Hx_k)^T R^{-1} (z_k - Hx_k)\} \tag{4}$$

The prior and posterior beliefs are assumed to be Gaussian distributed with mean, μ and covariance Σ. The prior gets updated to posterior in accordance to the Kalman Filter algorithm given in Table 1.

TABLE I.
KALMAN FILTER ALGORITHM

Algorithm KF (μ_{k-1}, Σ_{k-1}, u_k, z_k)
Prediction: $\bar{\mu}_k = A\mu_{k-1} + Bu_k$ $\bar{\Sigma}_k = A^T \Sigma_{k-1} A + Q$

Update:

$$K = H^T \bar{\Sigma}_k \left[H \bar{\Sigma}_k H + R\right]^{-1}$$
$$\mu_k = \bar{\mu}_k + K(z_k - H\bar{\mu}_k)$$
$$\Sigma_k = (I - KH)\bar{\Sigma}_k$$

In the Table 1, $\bar{\mu}_k$ and $\bar{\Sigma}_k$ are the estimated state values, which are updated by the measurement using Kalman gain, *K*.

B. *Motion Model*

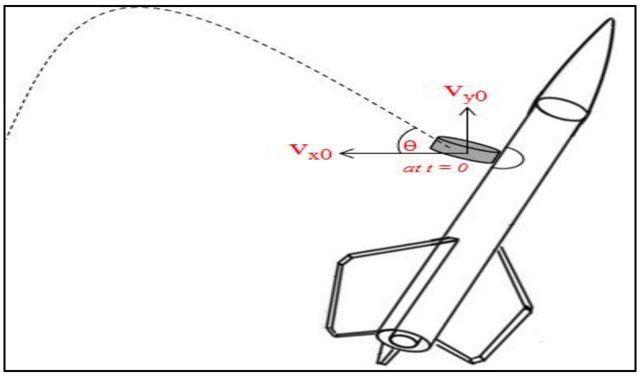

Fig 2 Projectile Motion

For the FFU, ejected from the rocket, it follows a ballistic projectile motion. In the ionosphere region, as there is no atmosphere, hence no non-linear variations in the model. The complete motion is linear following the projectile motion only under the influence of miligravity. The FFU is ejected from the rocket travelling at some velocity and acceleration in direction opposite to earth and reaching towards the apogee, as shown in the **Fig 2**. The equation of motion for the FFU in X-Y-Z direction is given by Eq(5),

$$\begin{aligned} X_k &= X_{k-1} + dt * V_{X,k-1} \\ Y_k &= Y_{k-1} + dt * V_{Y,k-1} - 0.5 * dt * g \\ Z_k &= Z_{k-1} \\ V_{X,k} &= V_{X,k-1} \\ V_{Y,k} &= V_{Y,k-1} - dt * g \\ V_{Z,k} &= V_{Z,k-1} \end{aligned} \tag{5}$$

The equations from the projectile motion can be written as the state transition model defined in Eq(1), as shown below. All the states defined in Eq(6) are in global frame.

$$\begin{bmatrix} X_k \\ Y_k \\ Z_k \\ V_{X,k} \\ V_{Y,k} \\ V_{Z,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{k-1} \\ Y_{k-1} \\ Z_{k-1} \\ V_{X,k-1} \\ V_{Y,k-1} \\ V_{Z,k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5dt^2 \\ 0 \\ 0 \\ -dt \\ 0 \end{bmatrix} [g] + Q \tag{6}$$

C. *Measurement Model*

The measurement of the localization for the FFU is taken from the accelerometer and gyroscope in X-Y-Z axis. For measurement model, the states and the measurements are in one-to-one relation, making the H matrix an identity matrix. The actual measurements is done via accelerometer and gyroscope and then they are converted to get position and velocity measurements. The equation of the measurement can be written as the measurement defined in Eq(3), as shown below.

$$\begin{bmatrix} PosX_k \\ PosY_k \\ PosZ_k \\ Vel_{X,k} \\ Vel_{Y,k} \\ Vel_{Z,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \\ Z_k \\ V_{X,k} \\ V_{Y,k} \\ V_{Z,k} \end{bmatrix} + R \tag{7}$$

The velocity of the FFU can be calculated from the accelerometer measurements, according to the equation,

$$Vel_k = Vel_{k-1} + \int_0^t (a_k - g)dt \tag{8}$$

In the Eq(8), a is the acceleration measurement on FFU in the axis for current time instance and g is the acceleration due to gravity. Similar the position of the FFU can be calculated from the velocity, according to the equation,

$$Pos_k = Pos_{k-1} + \int_0^t Vel_k dt \tag{9}$$

As the measurements are discrete samples, the integration can be approximated by rectangular rule and hence the position and velocity can be calculated as,

$$Vel_k = Vel_{k-1} + dt * (a_k - g) \tag{10}$$

$$Pos_k = Pos_{k-1} + dt * Vel_k \tag{11}$$

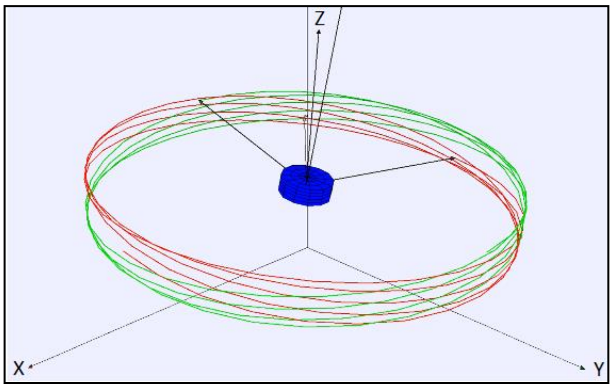

Fig 3 Wobbling Motion of FFU

D. Frame Conversion – Body to Global

The measurement for the localization of the FFU is taken from accelerometer and the gyroscope are in body frame. The FFU is spinning and wobbling as shown in the Fig 3.

In the Fig 3, the X- axis and the Y-axis of the body frame is rotating with the FFU, the Z- axis is normal to the FFU plane around which conning is happening. Hence the measurements from the accelerometer cannot be used directly to determine the position and velocity. They have to be converted from body to global frame.

As shown in the **Fig 2**, global frame for all the parameters is from the point of ejection of the FFU.

The conversion is done using a 3x3 rotation matrix, *C* relative to the global frame. Each of the column vector is a unit vector along one of the body axes specified in terms of global axes[6]. Hence a vector in global frame can be defined as,

$$a_g = C * a_b \tag{12}$$

In Eq(12), the a_g is the vector in global frame and a_b in body frame. The rotation matrix is given by,

$$C_k = C_{k-1} * \exp(\int_0^t \Omega dt) \tag{13}$$

In the Eq(13), the C_{k-1} is the rotation matrix at previous time instance. The Ω is a skew symmetric matrix of angular velocities of the FFU, taken from the measurements from gyroscope. The matrix is given by,

$$\Omega = \begin{bmatrix} 0 & -\omega_{bz,k} & \omega_{by,k} \\ \omega_{bz,k} & 0 & -\omega_{bx,k} \\ -\omega_{by,k} & \omega_{bx,k} & 0 \end{bmatrix} \tag{14}$$

As the values of the gyroscope are discrete samples, the integration is approximated using the rectangular rule and taylor series until 2nd order, as given below[6],

$$C_k = C_{k-1} * (I + \frac{\sin\sigma}{\sigma} B + \frac{1-\cos\sigma}{\sigma^2} B^2) \tag{15}$$

In the Eq(15), the parameters are given by

$$\sigma = \left| \omega_b dt \right|$$

$$\omega_b = \left[\omega_{bx}, \omega_{by}, \omega_{bz} \right]^T$$

$$B = \begin{bmatrix} 0 & -dt * \omega_{bz,k} & dt * \omega_{by,k} \\ dt * \omega_{bz,k} & 0 & -dt * \omega_{bx,k} \\ -dt * \omega_{by,k} & dt * \omega_{bx,k} & 0 \end{bmatrix}$$

Hence the acceleration measurements from the accelerometer can be written in global frame as,

$$a_{k,g} = C_k * a_{k,b} \tag{16}$$

IV. Implementation

The motion and measurement model are defined based upon the Eq(1) and Eq(3), and the prediction and the update equation are based upon the algorithm in Table 1.

The algorithm is been implemented in Matlab using different functions as described.

A. main_entry.m

This is the entry point for the algorithm. The file does the initialization, generates the measurements, and run the Kalman filter and display results.

B. generateData.m

Inputs: ~

Outputs: Angular velocity and Acceleration measurements in X-Y-Z axis with and without noise.

For the project, we are testing the algorithm by simulating the measurements for the accelerometer and gyroscope based upon the dynamics equations, modeled for actual FFU. The function generates 600 samples, for each measurement generated per second. Hence a dataset for 10min runtime. A independent white gaussian noise is added to both accelerometer and gyroscope data to mimic the noisy measurements.

C. initialization.m

Inputs: Noiseless angular velocity and acceleration data sample for time k = 1

Outputs: Initial values for state(X), covariance(P), measurements(Z), control (u), process matrix(A,B), measurement matrix(H), kalman gain(K), measurement noise(R), process noise(Q), rotation matrix(C) and sampling time (dt)

The function does the initialization of all the variables to be used in the algorithm for time instance k = 1.

The measurement at time t = 1 is used for initialization of the mean of the states. The measurement is derived from a noiseless acceleration and angular velocity.

The variance is given by Table 2. The measurements in X-Y-Z are independent to the axis. Hence the covariance between the position and velocity between different axes will be zero. As the states are derived from the measurement which are independent of each other, the variance in position and velocity is defined by the measurement noise. Similarly the covariance between velocity and position is defined as variance in position divided by sampling time. The Table 3 gives the variance matrix at time k = 1

TABLE II.

VARIANCE MATRIX

Var(pos-x)	Cov(pos-x, pos-y)	Cov(pos-x, pos-z)	Cov(pos-x, vel-x)	Cov(pos-x, vel-y)	Cov(pos-x, vel-z)
Cov(pos-y, pos-x)	**Var(pos-y)**	Cov(pos-y, pos-z)	Cov(pos-y, vel-x)	Cov(pos-y, vel-y)	Cov(pos-y, vel-z)
Cov(pos-z, pos-x)	Cov(pos-z, pos-y)	**Var(pos-z)**	Cov(pos-z, vel-x)	Cov(pos-z, vel-y)	Cov(pos-z, vel-z)
Cov(vel-x, pos-x)	Cov(vel-x, pos-y)	Cov(vel-x, pos-z)	**Var(vel-x)**	Cov(vel-x, vel-y)	Cov(vel-x, vel-z)
Cov(vel-y, pos-x)	Cov(vel-y, pos-y)	Cov(vel-y, pos-z)	Cov(vel-y, vel-x)	**Var(vel-y)**	Cov(vel-y, vel-z)
Cov(vel-z, pos-x)	Cov(vel-z, pos-y)	Cov(vel-z, pos-z)	Cov(vel-z, vel-x)	Cov(vel-x, vel-y)	**Var(vel-z)**

TABLE III.

VARIANCE MATRIX AT K = 1

R_x	0	0	R_x/dt	0	0
0	**R_y**	0	0	R_y/dt	0
0	0	**R_y**	0	0	R_z/dt
R_x/dt	0	0	**R_vel_x**	0	0
0	R_y/dt	0	0	**R_vel_y**	0
0	0	R_z/dt	0	0	**R_vel_z**

The process noise, Q, is initialized with a small values, as it is assumed that the motion model is very precise. The motion model includes all the dynamic parameters for the motion in space. In ionosphere, there is no force acting on the FFU except gravity. The only unsure motion is due to the ejection of the FFU from the rocket, and the noise will

The measurement noise is initialized to a higher value as we don't know the amount of noise present in the sensor. The noise from the gyroscope will be accumulated while calculating the rotation matrix. This will again be added to the accelerometer, which itself is noisy. Also, the position and velocity are approximated by rectangular rule and taylor series, hence resultant measurement is noisy. The measurement noise tries to simulate all the noises mentioned above.

D. *measurements.m*

Inputs: Mean of the states, previous rotation matrix, angular velocity and acceleration values at current time instance, sampling time.

Outputs: Measurements for the current time instance and the rotation matrix.

The function gives the measurements at current time instance. It takes the angular velocity and acceleration in body frame, convert the acceleration values into global frame and outputs the measured position and velocity values, with the rotation matrix at that time instance. These are the measurements used to update the estimated states in Kalman filter.

E. *body2global.m*

Inputs: Acceleration and angular velocity in body frame, the previous states values, previous rotation matrix and sampling time

Outputs: Position, velocity and updated rotation matrix.

The function is the major part of the measurement function. It converts the accelerometer data to global frame, integrate it to get velocity and again to get position values.

F. *kf.m*

Input: Previous states, current measurement, control, process and measurement noise, state and measurement matrices.

Output: Updated state

In this function we run the Kalman filter from time instance t=2 until 600.

G. *trueData.m*

Input: Mean value of state at k = 1

Output: The actual states we should get with the projectile motion.

The function that generates the ideal projectile motion for the static, non-rotating body, under free fall. This is just as reference to see how good the Kalman filter works for the actual case.

V. RESULTS AND CONCLUSION

The Matlab program has been run for different values of process and measurement noise, and the response been captured.

For each case of Q and R value, 4 trial were done to make sure that the values selected give consistent result for all situations.

The noises are initialized as Q = 1 and R = 10. The **Fig 4**, shows the response of the filter for the above initialized values, for one of the trials done. The x-axis is the displacement of FFU in x direction, and y-axis is the displacement of FFU in y direction. The '*blue*' dotted line is the actual motion by integrating the measurements and the '*yellow*' dotted line is the true projectile motion. The **Fig 4**, shows that the Kalman Filter diverged and the states estimation is not consistent at all. There can be two reason behind the divergence. Firstly, the process noise is too high, secondly, the measurement noise is too less.

The Q values in the filter is now increased to see its effect on the states estimation. The noises are initialized as Q = 10 and R = 10. The **Fig 5**, shows the response of the filter for the above initialized values. The figure shows that increase in the process noise does not help in the state estimation. The **Fig 6** and **Fig 7** confirms the above observation for Q=50 and Q=100.

The Q values in the filter is now decreased to see its effect on the states estimation. The noises are initialized as Q = 0.10 and R = 10. The **Fig 8**, shows the response of the filter for the above initialized values. The estimation is better but still not consistent.

The above proves that the motion model is precise and has very less noise in the system to affect the state estimation.

The R values in the filter is now increased to see its effect on the states estimation. The noises are initialized as R = 100, R=200 and R=500 for Q=1. The **Fig 9**, **Fig 10** and **Fig 11**, shows the response of the filter for the above initialized values respectively. The filter converges more often and is consistent.

The above proves that the measurement model is noisy and measurements have to be adjusted with very big noise to make the filter more consistent.

Because of the small process noise and big measurement noise, the Kalman Filter is more biased toward the measurement, in sense the update will be heavily dependent on the estimated mean and variance and not so much on the measurements.

Also, even with small process noise and big measurement noise, the filter often diverges. The **Fig 12** shows the response for R=500 and Q=1. As it can be seen in the figure, the estimate are far away from the true values. The filter is still not consistent for all cases. Hence, there is still some issue that makes the filter diverge. One of the reason of divergence is the measurements itself. Currently we are using all of the measurements for the update of the states. As the measurements are very noisy, a small series of wrong measurements can diverge the states estimation completely. To make the filter more consistent, we should remove those measurements which does not represent the states and consider only which does.

VI. Future Work

The next step will be to develop an outlier detection algorithm to remove those measurements which does not represent the states. This will make filter more consistent. Another step is to use the measurements from the actual accelerometer and gyroscope and develop the measurement model in much more realistic manner. This way we will have more accurate value of the measurement noise. Further to do data fusion with the GPS measurement with the accelerometer and gyroscope to have much better state estimation of the FFU. Also to update the process model to be more precise to the dynamics of the not just the FFU motion, but covers a range of motion and attempt to use the kalman filtering techniques for non-linear systems.

VII. References

[1] L. Klicker, A. David, A. Grima, A. Peitroniro Gabriele, G. Vieira Andreas, and R. Bregstrom, "Student Experiment Document - Team SLED," no. November, 2016.

[2] H. Lindberg, M. Crimella, M. Montecchia, and S. Westerlund, "Student Experiment Document - Team SLED," no. September, 2015.

[3] S. Mawn and A. Schmidt, "REXUS User Manual," 2014.

[4] C. Duca, "Development, integration, testing and launch support of the electronic equipment of the SPIDER/Leewaves electromagnetic fields sounding rocket," *Eur. J. Eng. Educ.*, vol. 8, no. 4, pp. 389–389, 1984.

[5] A1RONZO, "GPS Basic," *Sparkfun*. [Online]. Available: https://learn.sparkfun.com/tutorials/gps-basics.

[6] O. J. Woodman, "An Introduction to Inertial Navigation," *Univ. Cambridge*, no. 696, pp. 1–37, 2007.

[7] G. P. Bevan, "Maximum distance attained by a ballistic projectile launched from a free swinging rigid List of Figures," pp. 1–15, 2003.

[8] L. Zhao, H. Qiu, and Y. Feng, "Analysis of a robust Kalman filter in loosely coupled GPS/INS navigation system," *Meas. J. Int. Meas. Confed.*, vol. 80, pp. 138–147, 2016.

[9] J. Wendel, C. Schlaile, and G. Trommer, "Direct Kalman Filtering of GPS/INS for Aerospace Applications," *Int. Symp. Kinematic Syst. Geod. Geomatics Navig.*, pp. 144–149, 2001.

[10] L. Góes, E. Hemerly, and B. Maciel, "Parameter Estimation and Flight Path Reconstruction using Output-Error Method," *24th Int. Congr. Aeronaut. Sci.*, pp. 1–10, 2004.

[11] Z. Liu and J. Chen, "An Improved Square Root Unscented Kalman Filter for Projectile ' s Attitude Determination," pp. 1747–1751, 2010.

[12] H. A. Mohamed and P. K. Schwarz, "Adaptive Kalman Filtering for INS/GPS," *J. Geod.*, vol. 73, no. 4, pp. 193–203, 1999.

[13] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *In Pract.*, vol. 7, no. 1, pp. 1–16, 2006.

[14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, vol. 53, no. 9. 2013.

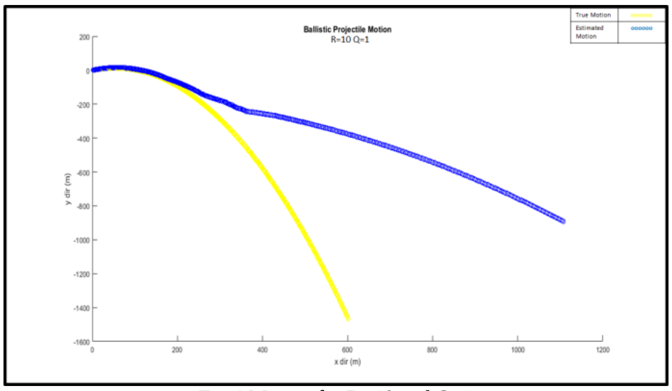

Fig 4 Motion for R=10 and Q=1

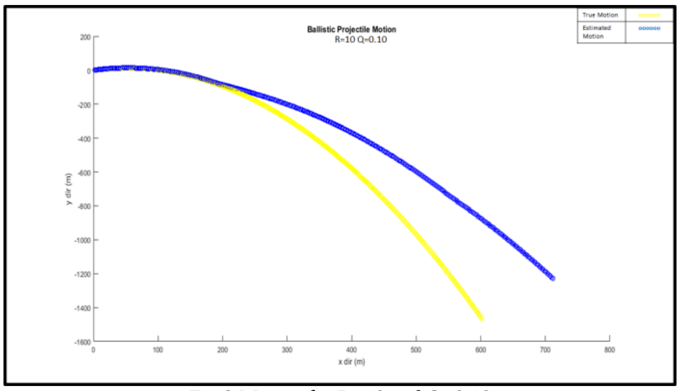

Fig 8 Motion for R=10 and Q=0.10

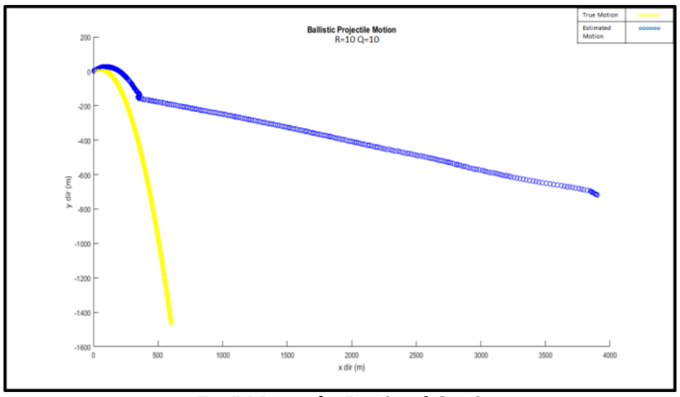

Fig 5 Motion for R=10 and Q=10

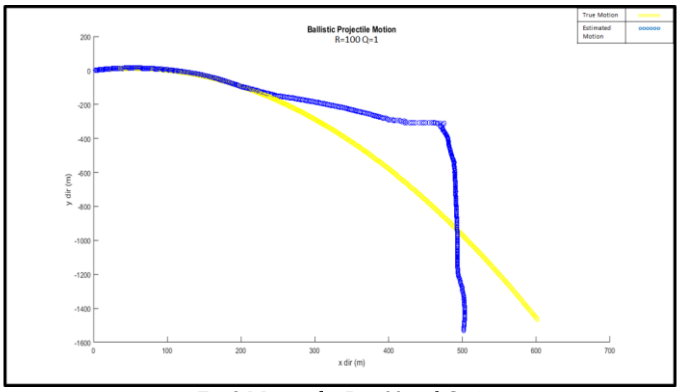

Fig 9 Motion for R=100 and Q=1

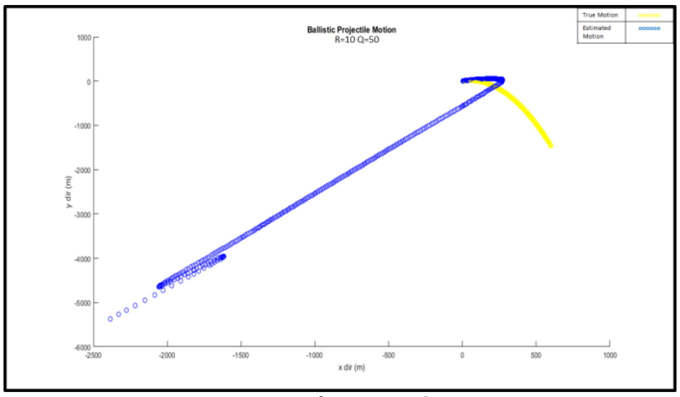

Fig 6 Motion for R=10 and Q=50

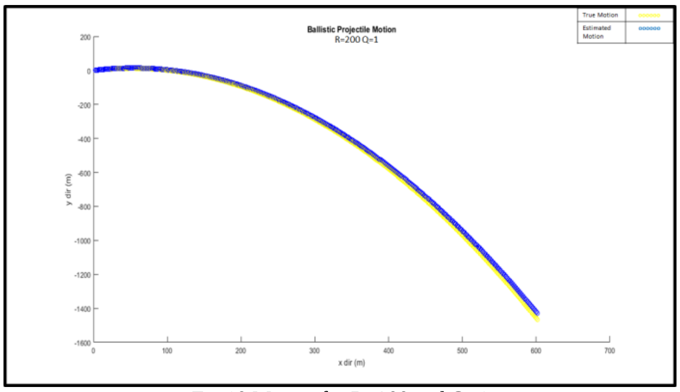

Fig 10 Motion for R=200 and Q=1

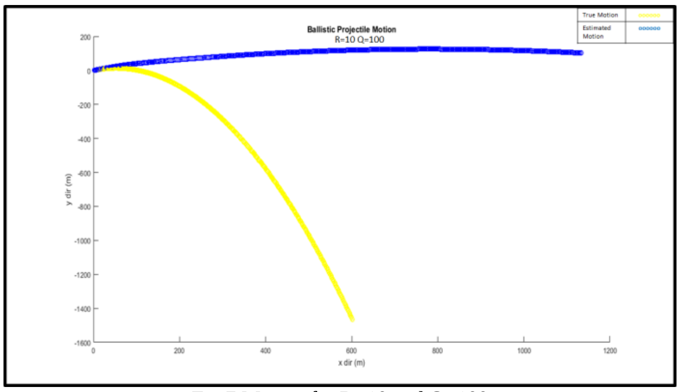

Fig 7 Motion for R=10 and Q=100

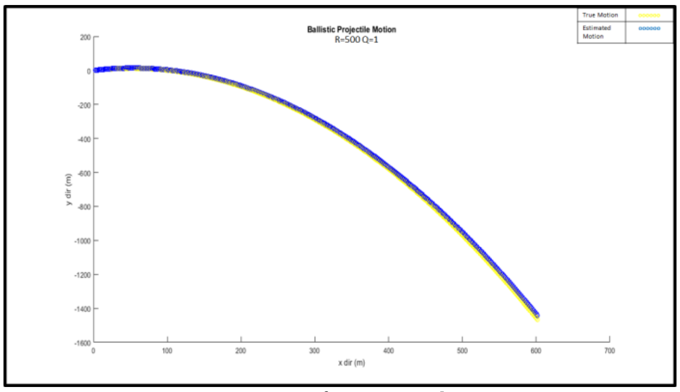

Fig 11 Motion for R=500 and Q=1

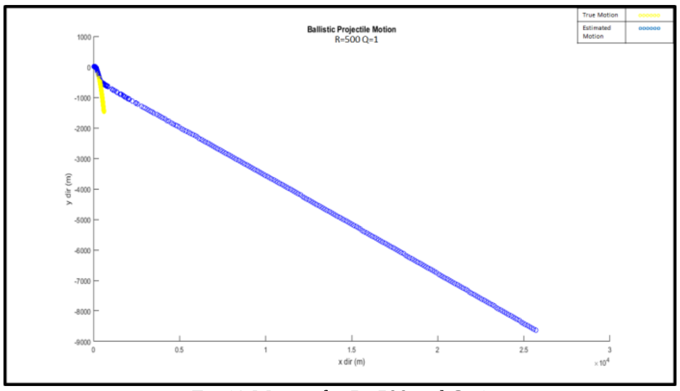

Fig 12 Motion for R=500 and Q=1