International Journal of Science, Environment and Technology, Vol. 5, No 3, 2016, 941 – 949

ISSN 2278-3687 (O)
2277-663X (P)

MONTE CARLO BASED ANALYSIS OF TRACKING THE TRAJECTORY OF A PROJECTILE USING KALMAN FILTER

Jaydip Mukhopadhyay and Sayanka Saha
[1]Assistant Professor, [2]PG Scholar
Department of Computer Science and Engineering
RCC Institute of Information Technology
E-mail: jaydipcu@gmail.com, sayanka.saha@gmail.com

Abstract: Tracking of a moving object is always considered as an interesting problem. This problem may incorporate many complicacies due to changing environment during the flight of the object, due to inherent noise in the sensors, due to inaccurate assumption of the model. Here we are demonstrating a moving projectile, having some significant amount of mass, thrown from one point to reach another point. Later the famous Kalman filter is implemented to approximate the trajectory of the state of the moving particle.
Keywords: Moving projectile, Flight of the object, Kalman filter.

INTRODUCTION

Projectile motion is a very common article in vector mechanics. When a particle is thrown with certain initial velocity and at a particular angle is known as projectile. This particle has 2D motion and journey during its flight time. This 2D motion and displacement creates a different trajectory if compared to 1D particle motion. Projectile motion has a great deal of importance in physics as it is the framework of several practical incident like following the path of a cricket ball, path of an aero plane, path of a space craft etc.

The motion and position of a moving particle are generally measured by some measuring instruments, precisely by some sensor appropriate for respective parameters. This sensor data always contains some amount of error in their measurements because of its sensing method, outside disturbance etc. if this error let freely accommodated in the calculation or visualization of the trajectory of the projectile, the result defers from the actual trajectory. So that becomes a challenge for tracking the trajectory of a projectile. Therefore a filter is devised to refine the inherent error coming from the sensor which is embedded inside the projectile. This will improve the estimation of the trajectory of the projectile nearer to the actual phenomena. In this paper the physical model of a projectile motion is there in Section-I, the theory of motion is discussed in the said section. In Section II, the basics of Kalman

*Received April 4, 2016 * Published June 2, 2016 * www.ijset.net*

filter and its application in the proposed dynamics is mentioned. Monte Carlo simulation of Kalman filter model for projectile motion is conducted in section-III. A comparison is presented between the non-linear physical model of the projectile and Kalman formulated model to show the difference in estimation is given in section-IV. A brief conclusion followed in section-V.

Section -I

Motion theorem

Projectile is defined as, throwing a object with an initial velocity, which is then allowed to move under the action of gravity alone, without being propelled by any engine or fuel. The path followed by a projectile is called its trajectory. A projectile moves at a constant velocity in the horizontal direction while experiencing a constant acceleration of 9.8 m/s^2 downwards in the vertical direction. To be consistent, we define the upward direction to be the positive. Therefore the acceleration of gravity is, -9.8 m/s^2.

Horizontal motion of projectile: The speed in the horizontal direction is 'v_x' and this speed doesn't change. The equation which predicts the position at any time in the horizontal direction is simply, [2]

$$x = v_x t \qquad \text{-------------- (1)}$$

Vertical motion of projectile: Gravity has a downward pull, the vertical velocity changes constantly. The equation that predicts the vertical velocity at any time 'v_y' is [2]

$$v_y = v_{oy} + at \qquad \text{-------------(2)}$$

The 'V_{oy}' is simply the original velocity in the vertical or y-direction. To calculate the position in the y-direction, the full distance must be used. 'Y_o', represents the original position in the y-direction.[2]

$$y_f = y_o + v_{oy} t + \frac{1}{2} a t^2 \qquad \text{--------------(3)}$$

Section -II

Kalman filter and its basics:

The Kalman filter [2] can be defined as a recursive linear estimator which successively an estimate for a continuous valued state, over time domain, for a periodic observations of the state. It can be framed as a statistical model of the parameter of interest $x(t)$ which evolves over time. The gains employed in a Kalman filter are chosen to ensure that, with certain assumptions about the observation and process models used, the resulting estimate $\hat{x}(t)$

minimises mean-squared error and is thus the conditional mean $\hat{x}(t) = E[x(t)|Z^t]$; an average, rather than a most likely value.

Introduction of Kalman filter based on the linear model to approximate the trajectory of a moving projectile.

Predict step:[4]

$X_{k|k-1}=F_k x_{k-1|k-1}+B_k u_k$------------------------------ (4)

$P_{k|k-1}=F_k P_{k-1|k-1} F^T_k+Q_t$----------------------------- (5)

Update step:[4]

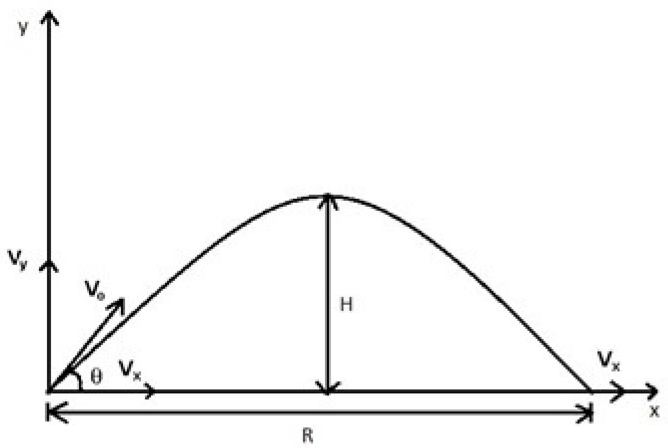
$X_{k|k}=x_{k|k-1}+K_k(y_t-H_k x_{k|k-1})$ ----------------------- (6)

$K_k=P_{k|k-1}H^T_k(H_k P_{k|k-1} H^T_k+R_k)^{-1}$-------------- (7)

$P_{k|k} = (I-K_k H_k)P_{k|k-1}$-----------------------------------(8)

Where X_k is Estimated state,Y_k is Measurement state, F_k is State transition matrix, B_k is Control matrix, $U_k$is control variables, P is State variance, Q is Process variance, K is Kalman Gain and H is Measurement matrix.

Modeling the state dynamics of a projectile through KF and we observe the following.



An object starts with velocity v_0 and at an angle Θ. As a 2D problem this velocity has two components $v_x= v_0 \cos(\Theta)$ and $v_y= v_0 \sin(\Theta)$. The notations used in the above diagram are mentioned below:

X_t: Distance from the starting point at time t

Y_t : Distance from the starting point at time t

V_x : Velocity at a time point t

V_y : Velocity at a time point t

T: time of flight

R: Range of the projectile

H: Maximum height that object can travel against gravity

Applying vector mechanics for a 2D motion,

$X_t= x_0+v_0*t$

$Y_t=y_0+v_y*t-0.5*g*t*t$

$V_x=v_0\cos(\Theta)$

$V_y=v_0\sin(\Theta)-g*t$

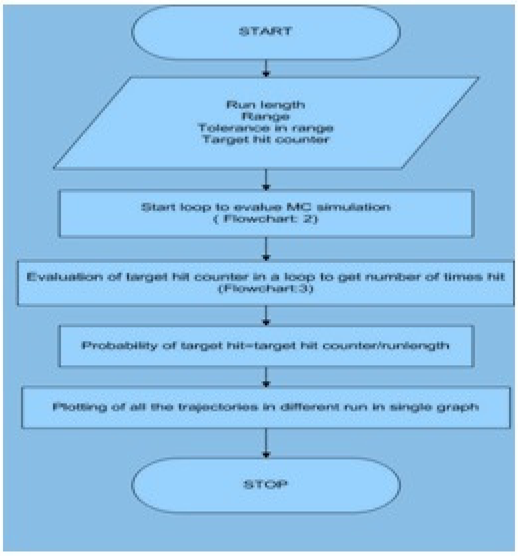
$T= (2v_0\sin (\Theta))/g$

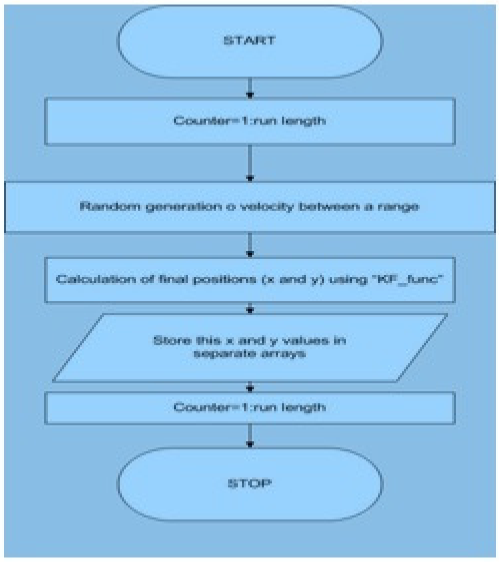
$R= (v_0*v_0*\sin (\Theta))/g$

Section: III

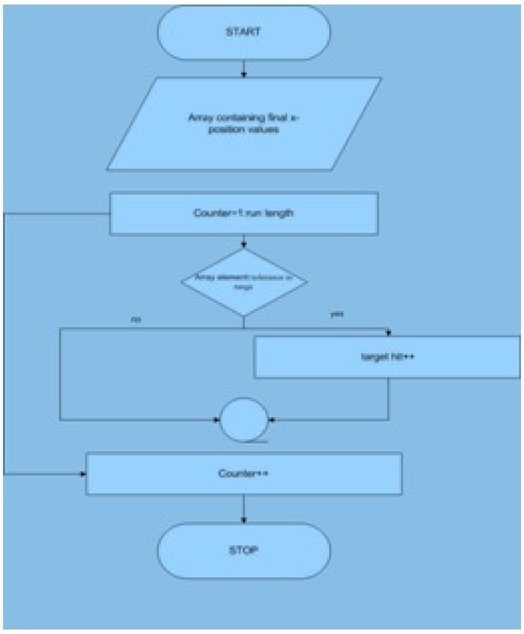
Monte Carlo simulation of Kalman filter model for projectile motion:

Monte Carlo simulation is implemented to evaluate the performance of the Kalman filter developed in this paper. The objective is to evaluate the probability of the projectile hitting a particular distance in between a specified velocity range. From the probability it will be clear that the velocity range chosen for a particular range is suitable or some correction required. [5].





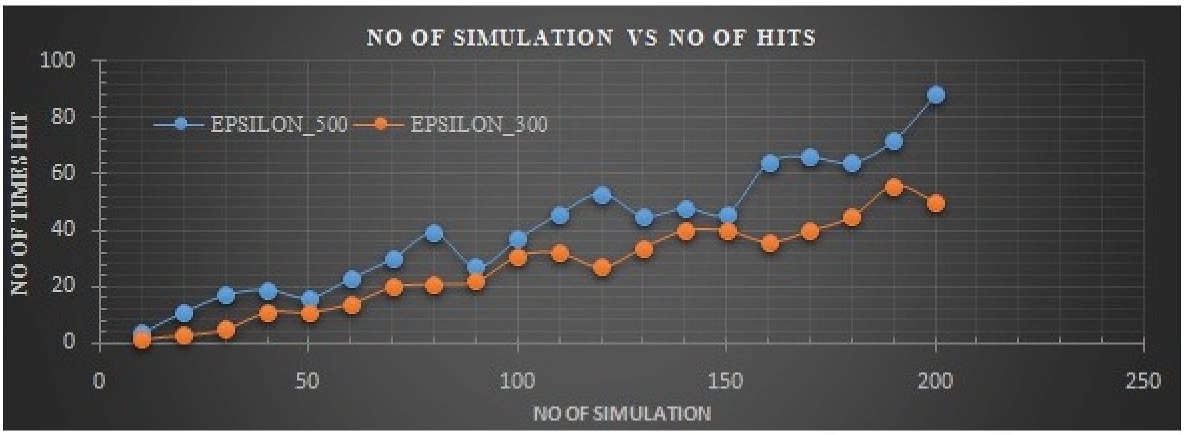
Flowchart 1: overview of entire MC simulation process Flowchart 2: Loop for MC simuation run

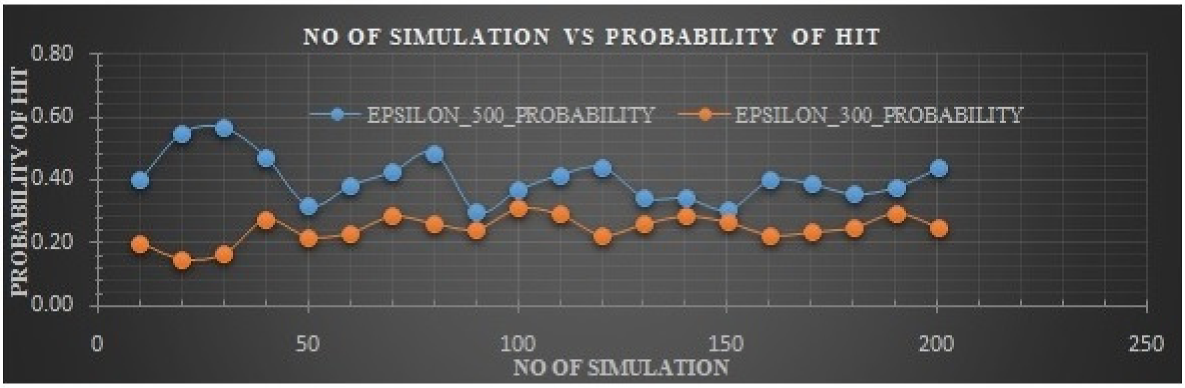


Flowchart 3: Evaluation of the number of target hit

- It receives the initial velocity as input arguments. It will utilize the Kalman filter modeled inside this file. All the parameters of this filter can be changed from this file.
- "output screenshot paste of velocity matrix"
- After evaluation of x_predict vector its element wise value is used to evaluate the element wise value of measurement vector. The x_predict vectors elements are used to create measurement elements randomly within a certain range. This range can be changed in this file. If the measurement data is available outside user, then this part can be modified to take input from user. The x_update result is sent to the parent program as an output.

DISCUSSION ON MC SIMULATION ON KF BASED PROJECTILE TRACKING PROBLEM:



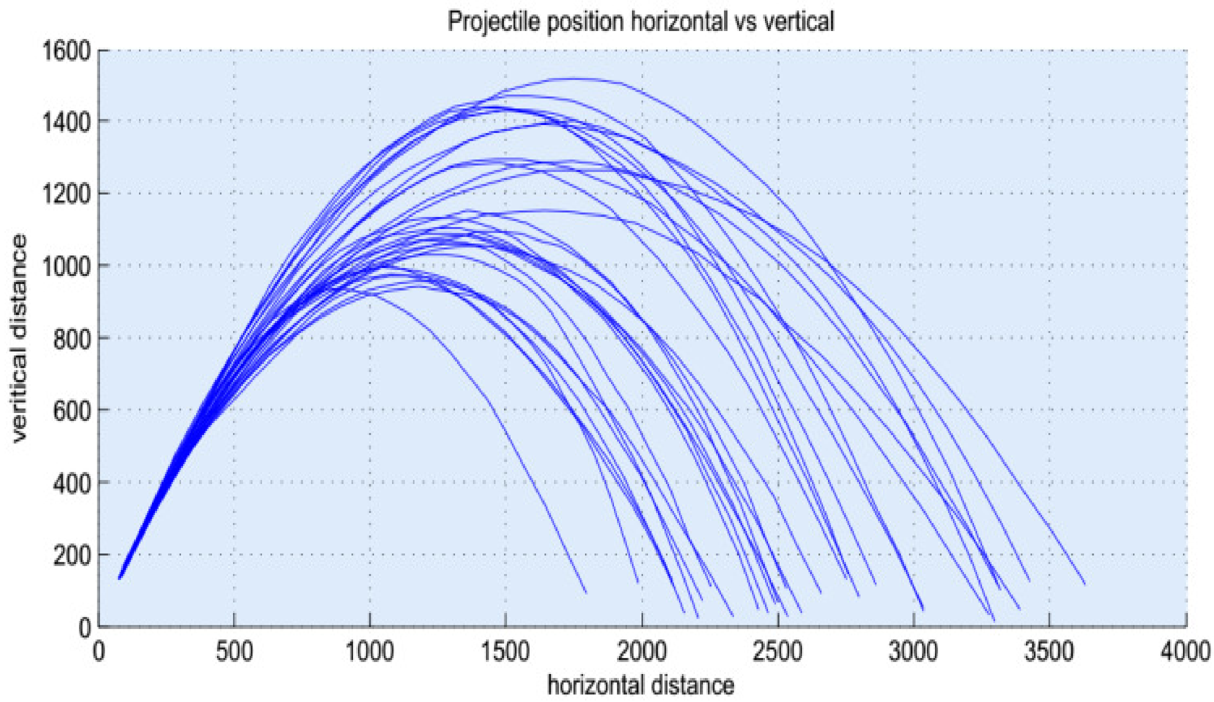


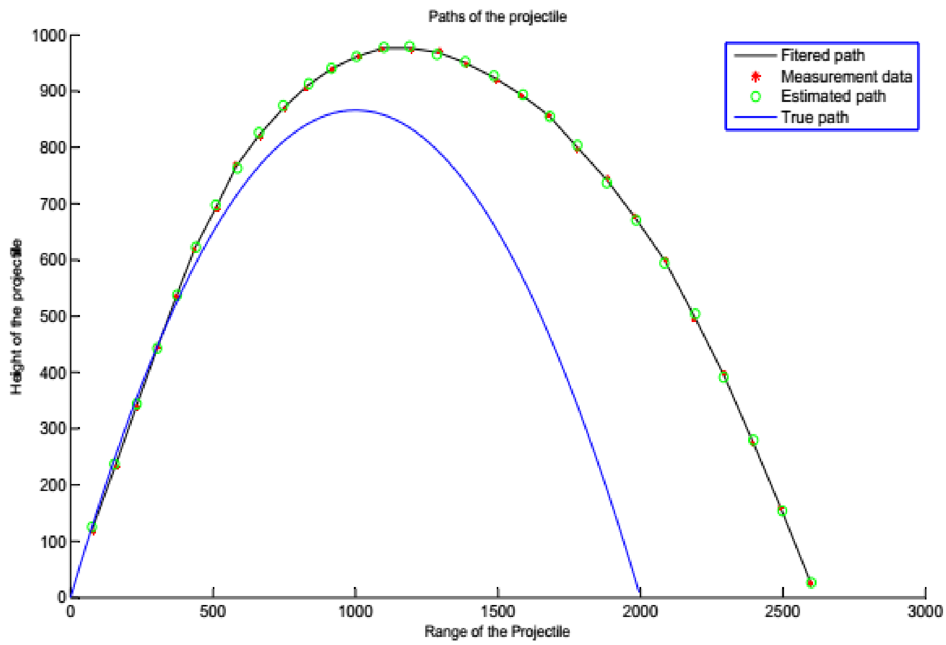
From the above plot it is clear that the filter model formulated in this work handles randomness in velocity well. At some runs the probability is correspondingly high compared to other runs. That is because of the wider span of tolerance, i.e. “epsilon” is considered. At the same run when the tolerance is less, i.e. 300 compared to previous run, the probability is nearer to values from other runs. It is also clear the number of hit decreases as the tolerance decreases and result are more consistent compared to larger tolerance. The choice of tolerance is completely based on the situation under experiment. This tolerance is filter parameter independent. So any change in this value will not affect filter’s performance. To increase further fineness in the result, the filter parameters may be tuned as per the system under consideration. From our earlier discussion it was shown that there is a large difference in the result considering the non-linear physical model as actual model. This is because the air resistance which is acting as an external source of control input is not present in the linear model of the Kalman filter developed here. That non-linear model requires linearization using Taylor series or any other linearization method like finite difference etc. but if we use the 1st order liberalized expression of that non-linear model the error will be more and the filter mostly drifts from the actual as the truncation error piling up exponentially.

Result & Discussion

Projectile motion horizontal VS vertical:

The plot of trajectories at different simulation run shows one thing that the range of the runs varies widely and few of them are only inside the tolerance limit set inside the code. This is backing up by the small probability of target hit. From this one thing is clear that the randomness in simulation can’t be set large. Using these results generated for different run a close range of randomness can be determined in which the probability of hitting the target will be somewhat healthy.

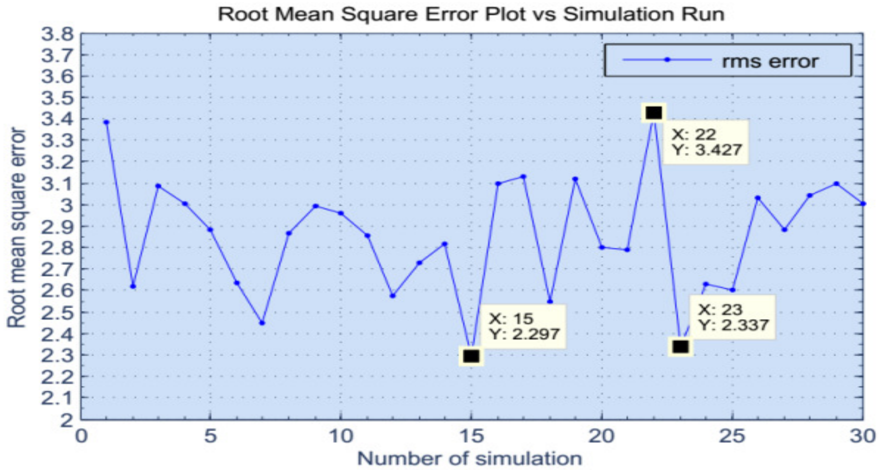


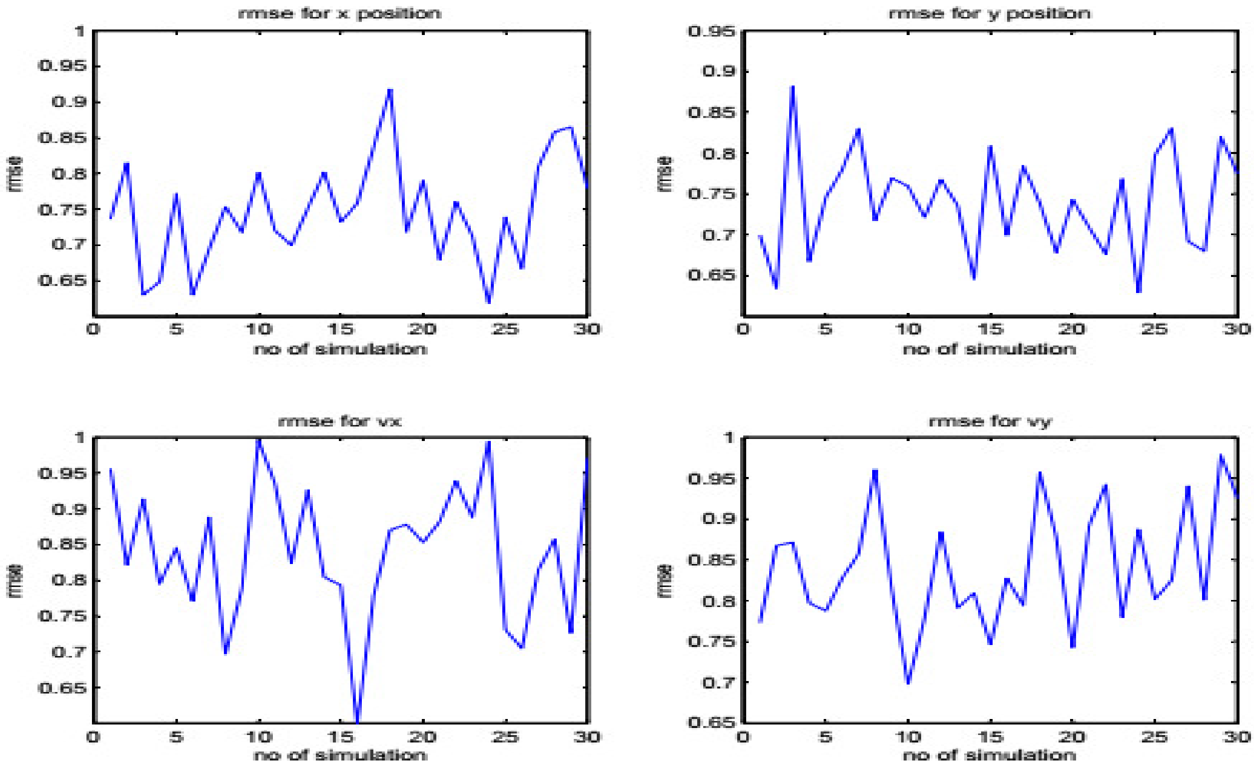


The rms error calculated using the following expression-

$$rmse = \sqrt{mean(y - x\_predict)^2}$$

From the rms error the filter performance can be evaluated. The rms error value decreases with increasing accuracy in the filter performance to evaluate x_predict. So, rms error is an estimator of the filter performance quantitatively. From the bottom plot we can see that for n=50 the max rmse is 3.427 and min rmse is 2.297. This much wide variation in rmse indicates that the filter is unable to handle all the random situation with same degree of accuracy. This instability will increase as the simulation number will increase.





In the following graph the root mean square errors for every parameter is plotted. Here errors are not unbiased because at some run in Monte-Carlo simulation the errors are more than other runs. The higher error rate indicates the chance of reaching the prescribed target for the projectile is very small as the deviation is more. From this random error analysis a close range of velocity can be determined in which the probability of reaching the target for the projectile is more.

Conclusion

From this project the projectile can be tracked by using Kalman filter so that the Estimation error will come down. Moreover simple projectile tracking for three dimensional system can be implemented in future.

References

[1] R.D.H. Warburton, J. Wang, "J. Burgdörfer, Analytic Approximations of Projectile Motion with Quadratic Air Resistance" doi:10.4236/jssm.2010.31012 Published Online March 2010

[2] Kalman, R.E. (1960). "A New Approach to Linear Filtering and Prediction Problems". *Journal of Basic Engineering* **82**: 35. doi:10.1115/1.3662552.

[3] University at Stuttgart, "Kalman filtering Implimentation with matlab", Institute of Geodesy.

[4] Kalman, R.E. (1960). "A New Approach to Linear Filtering and Prediction Problems". Journal of Basic Engineering **82**: 35. Doi:10.1115/1.3662552

[5] Anderson, Herbert L. (1986). "Metropolis, Monte Carlo and the MANIAC" *(PDF). Los Alamos Science* ***14****: 96–108.*