Tracking a Cricket Ball in 3D Using Basic and Unscented Kalman Filters

Vineet Puri
Department of Electrical and Computer Engineering
Western University
London, Ontario
Email: vpuri8@uwo.ca

***Abstract*— In this project, a Basic Kalman filter and an Unscented Kalman Filter were used to track a cricket ball in three dimensions. The ball can bounce off the pitch and can also spin after the bounce. There is one camera to record the ball's position from the top and another to record the position from the side. The positions obtained from the cameras have significant measurement errors. In the last part of the project, a non-linear ball swing was also introduced to demonstrate the non-linear tracking capabilities of the Unscented Kalman Filter.**

***Keywords*— Kalman filters, Unscented Kalman filter, Estimation, target tracking, non-linear tracking**

I. INTRODUCTION

Kalman filter is an algorithm that uses a series of measurements observed over time, which contain statistical noise and other inaccuracies, to estimate unknown variables that tend to be more accurate than those based on a single measurement alone. It does so by estimating a joint probability distribution over the variables for each timeframe. It was proposed by R. E. Kalman in 1960 [1]. It has many applications like target tracking, orbit calculations, dynamic positioning, sensor data fusion, microeconomics, digital image processing and pattern recognition. But a basic Kalman Filter is limited to linear equations. When our equations are non-linear then Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) can be used. EKF has many drawbacks like Jacobians can be difficult to calculate, there can be a high computational cost and it is not optimal if the system is highly non-linear. So, in this project, first a basic Kalman filter was used and then an Unscented Kalman Filter was used to track the position of a cricket ball. After this a non-linear ball swing was introduced and then this system was again tracked using a UKF. Ball tracking can be useful in assisting the umpires or referees in taking the right decisions. It can also be useful in improving the viewer experience on television.

II. PROBLEM SETUP

In a game of Cricket, a ball is bowled by a bowler from one end of the pitch towards a batsman at the other end. In most cases, the ball bounces once before reaching the batsman as shown in fig. 1. The length of the pitch is approximately 20.12 m.

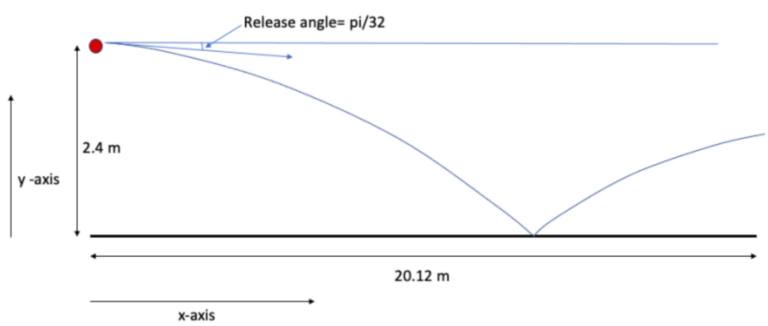

Fig. 1. Cricket pitch as seen from side

The following assumptions have been made for the Matlab simulation of the problem-

1. The bowler releases the ball at an angle along the axis parallel to the ground at an initial velocity of 125 km/h or 33.33 m/s. The release angle was fixed at pi/32.
2. The ball is released from a height of 2.4 m. (The point of release is 2.4 m above the surface)
3. The time taken by the ball to cross the pitch is not expected to be more than 2 seconds.
4. x-axis is the axis along the length of the pitch.
5. y-axis is perpendicular to the x-axis and directed towards the sky. So, gravity will act along this direction.
6. z-axis is along the width of the pitch as shown in fig. 2. So, the action of spin will be visible along this axis.
7. The bounce introduces a spin which can be observed from the top camera. The spin angle is taken to be pi/16 as shown in fig. 2.
8. It is also assumed that there is no loss of energy on bounce. Only the direction of motion is reversed along the y-axis.
9. In the last part, the ball can also swing in the air towards the right side. The amount of swing is a function of square of distance travelled by the ball from the point of release.

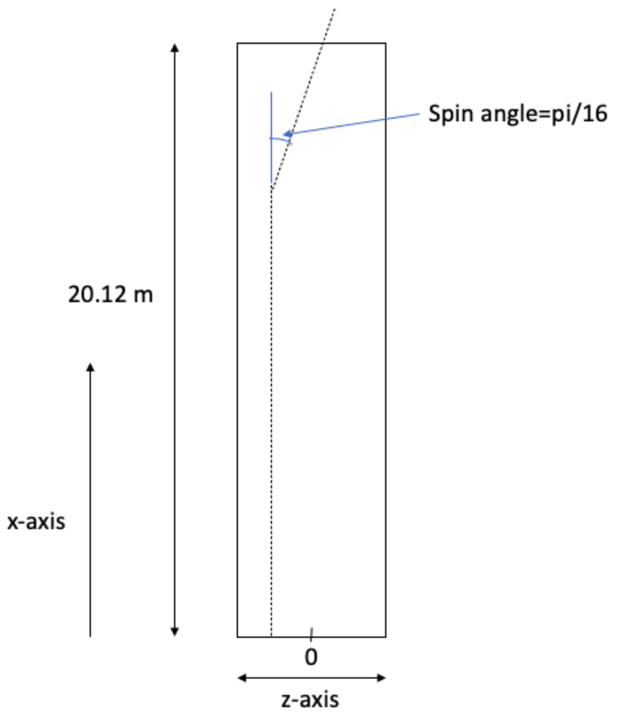

Fig. 2. Cricket pitch as seen from top

Our target is to track the ball along the three axes as it moves from one end of the pitch to the other. The process equations will be based on the following relations-

$x(t)=x_0 +v_{0x}t$
$v_x(t)= v_{0x}$
$y(t)= y_0 +v_{0y}t-(½)gt^2$
$v_y(t)= v_{0y}-gt$
$z(t)=z_0 +v_{0z}t$
$v_z(t)= v_{0z}$

Here x(t), y(t) and z(t) are positions along the x,y and z axis respectively. $v_x(t)$, $v_y(t)$ and $v_z(t)$ are velocities along the x, y and z axis respectively. g is the acceleration due to gravity. x_0, y_0 and z_0 are the initial positions.

Converting these equations into recurrent relations, with Δt representing a fixed time step we get-

$x_n=x_{n-1} +v_{x(n-1)}\Delta t$
$v_{xn}= v_{x(n-1)}$
$y_n= y_{n-1} +v_{y(n-1)}\ \Delta t -(½)g\ \Delta t^2$
$v_{y\ n}= v_{y\ (n-1)}-g\Delta t$
$z_n=z_{n-1} +v_{z(n-1)}\Delta t$
$v_{zn}= v_{z(n-1)}$

The state vector can be represented as

$[x_n\ v_{xn}\ \ y_n\ v_{yn}\ z_n\ v_{zn}]^T$

The state transition matrix is given by-

$$\begin{pmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The control vector will be $[0\ \ 0\ \ (1/2)g(\Delta\ t)^2\ \ g^*\ \Delta\ t\ \ 0\ \ 0]^T$. And the measurement matrix is simply an identity matrix of dimensions 6 by 6.

III. BASIC KALMAN FILTER

Basic Kalman Filter or just 'Kalman filter' is a linear optimal state estimation method. A dynamic model is presented with a state equation and a measurement equation through the reliable estimation corrected by measurements [2]. For a Kalman filter, the state equation can be represented as follows-

$x_k=Ax_{k-1}+Bu_{k-1}+Q_{1k}$

The measurement equation can be represented as-

$z_k=Hx_k+Q_{2k}$

here x_k is the state vector, z_k is the measurement vector, A is the state transition matrix, H is the measurement matrix, Q_{1k} is the process noise vector, B is the system control vector, Q_{2k} is the measurement noise vector.

Prediction equations for Kalman filter will be-

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

$$P_k^- = AP_{k-1}A^T + Q$$

And the update equations are defined as follows:

$$K_k = P_k^- H^T(HP_k^- H^T + R)^{-1}$$

$$\hat{x}_{k-1} = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

$$P_k = (I - K_k H)P_k^-$$

where K_k is the Kalman gain matrix
$\hat{x}_k$ is the optimum filter value,
P_k is the filter deviation matrix.

The Kalman filter algorithm was used in Matlab to track the cricket ball problem discussed in section II. The estimation results obtained are as shown in fig. 3 and fig. 4. The standard deviation for process noise was taken as 0.01 and measurement noise was taken as 0.1. It was found that the movement of the ball could be tracked with some error.

But a basic Kalman Filter is limited to linear equations and EKF has many drawbacks as already discussed. In the next part, the Unscented Kalman Filter was implemented for the same system.

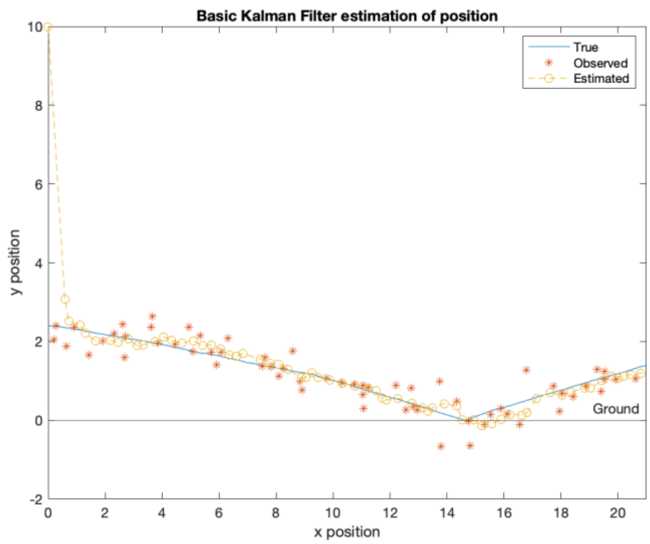

Fig. 3. Kalman filter estimation along x and y axis (side view)

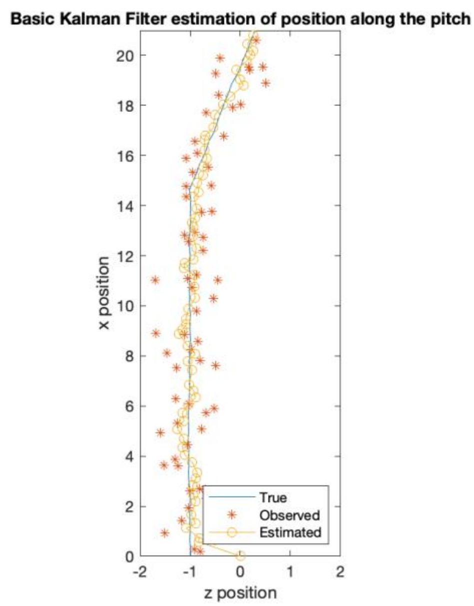

Fig. 4. Kalman filter estimation along x and z axis (top view)

III. Unscented Kalman filter

UKF uses linear Kalman filter framework based on Unscented Transform (UT), and has a definite sampling strategy instead of a random sampling strategy [3]. The UT is founded on the intuition that it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation [4]. A set of points called sigma points are chosen and a non-linear transformation is applied to each point to get transformed points. The statistics of the transformed points can then be calculated to form an estimate of the nonlinearly transformed mean and covariance [5]. Approach is shown in fig. 5.

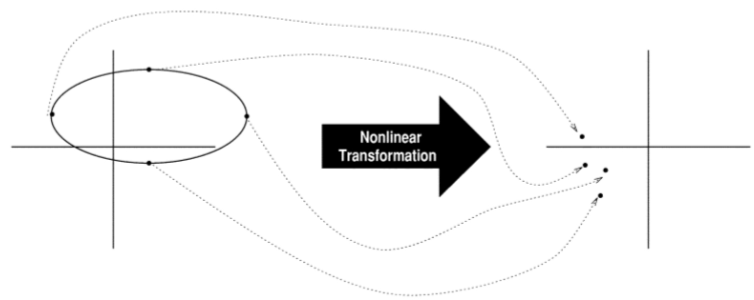

Fig. 5. The principle of UT. This is fig. 2 in [5].

A general formula for the number of sigma points is 2N+1 where N is the dimensions of the state vector x. In the Matlab simulation, 13 sigma points were chosen as there are 6 states in the state vector of the problem. Sigma points can be chosen as per the following equations-

$$
\begin{aligned}
\mathcal{X}^{[0]} &= \mu \\
\mathcal{X}^{[i]} &= \mu + \left(\sqrt{(n+\lambda)\,\Sigma}\right)_i \qquad \text{for } i = 1,\ldots,n \\
\mathcal{X}^{[i]} &= \mu - \left(\sqrt{(n+\lambda)\,\Sigma}\right)_{i-n} \qquad \text{for } i = n+1,\ldots,2n
\end{aligned}
$$

χ(Caligraphic X) -> Sigma Points Matrix
μ -> mean of the Gaussian
n-> dimentionality of system
λ-> Scaling Factor
Σ-> Covariance Matrix

λ is the scaling factor which tells us how far the sigma points should be chosen from the mean. λ was chosen to be 3-n in simulation as has been suggested by some sources [6]. The weights for sigma points can be calculated through the following equations-

$$
\begin{aligned}
w^{[0]} &= \frac{\lambda}{n+\lambda} \\
w^{[i]} &= \frac{1}{2(n+\lambda)} \qquad \text{for } i = 1,\ldots,2n
\end{aligned}
$$

The UKF algorithm [7] can be presented as follows-

Initialize with:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$$
$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]$$
$$\hat{\mathbf{x}}_0^a = E[\mathbf{x}^a] = [\hat{\mathbf{x}}_0^T\ 0\ 0]^T$$

$$\mathbf{P}_0^a = E[(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)^T] = \begin{bmatrix} \mathbf{P}_0 & 0 & 0 \\ 0 & \mathbf{P}_v & 0 \\ 0 & 0 & \mathbf{P}_n \end{bmatrix}$$

For $k \in \{1, \dots, \infty\}$,

Calculate sigma points:

$$\mathcal{X}_{k-1}^a = \left[\hat{\mathbf{x}}_{k-1}^a \ \ \hat{\mathbf{x}}_{k-1}^a \pm \sqrt{(L+\lambda)\mathbf{P}_{k-1}^a}\right]$$

Time update:

$$\mathcal{X}_{k|k-1}^x = \mathbf{F}[\mathcal{X}_{k-1}^x, \mathcal{X}_{k-1}^v]$$
$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}_{i,k|k-1}^x$$
$$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1}^x - \hat{\mathbf{x}}_k^-][\mathcal{X}_{i,k|k-1}^x - \hat{\mathbf{x}}_k^-]^T$$
$$\mathcal{Y}_{k|k-1} = \mathbf{H}[\mathcal{X}_{k|k-1}^x, \mathcal{X}_{k-1}^n]$$
$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{i,k|k-1}$$

Measurement update equations:

$$\mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-][\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-]^T$$
$$\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-][\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-]^T$$
$$\mathcal{K} = \mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} \mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^{-1}$$
$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathcal{K}(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$$
$$\mathbf{P}_k = \mathbf{P}_k^- - \mathcal{K} \mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} \mathcal{K}^T$$

where, $\mathbf{x}^a = [\mathbf{x}^T\ \mathbf{v}^T\ \mathbf{n}^T]^T$, $\mathcal{X}^a = [(\mathcal{X}^x)^T\ (\mathcal{X}^v)^T\ (\mathcal{X}^n)^T]^T$, λ=composite scaling parameter, L=dimension of augmented state, $\mathbf{P}_v$=process noise cov., $\mathbf{P}_v$=measurement noise cov., W_i=weights

Algorithm 3.1 Unscented Kalman Filter equations [7]

UKF algorithm was used to track the cricket ball in our problem in Matlab. The tracking estimates are as shown in fig. 6 and fig. 7. It can be seen that the cricket ball could be tracked with some degree of accuracy.

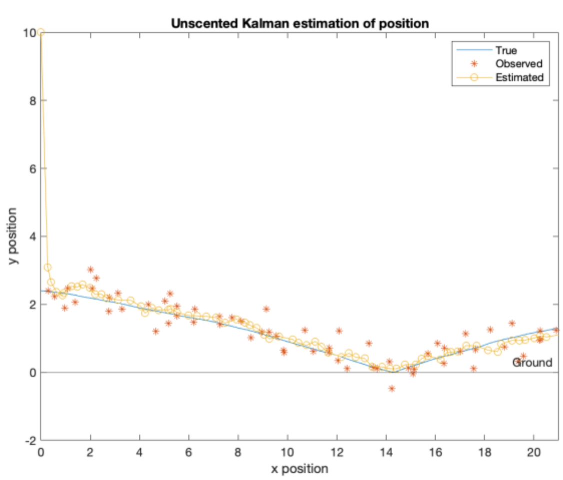

Fig. 6. Unscented Kalman filter estimation along x and y axis (side view)

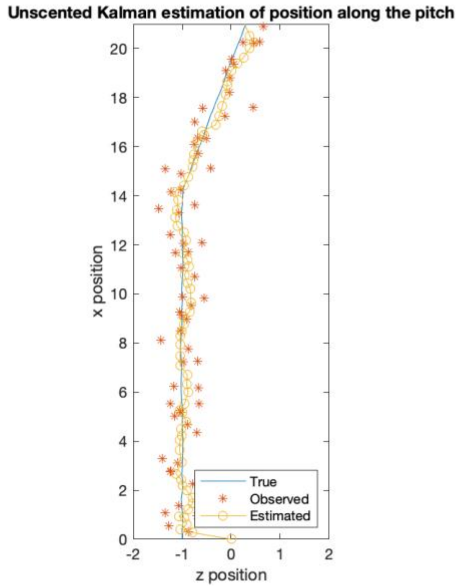

Fig. 7. Unscented Kalman filter estimation along x and z axis (top view)

But the real strength of UKF lies in its ability to track the non-linear systems. In the next part a non-linear component was introduced in the problem.

IV. Unscented Kalman filter with Non Linear Swing

UKF can be particularly useful if there are non-linearities in the system as these cannot be handled well by a Basic Kalman Filter. A non-linear swing of the ball was introduced in the system such that the amount of swing is a function of square of distance from the point of release. So, a ball swings more rapidly as it gets farther away along the pitch. The effect of swing will be visible along the z-axis. It was assumed that this swing deviates the ball in the air to the right side. This changed the process and measurement equations a bit. The new state equations are-

$$x_n = x_{n-1} + v_{x(n-1)}\Delta t$$
$$v_{xn} = v_{x(n-1)}$$
$$y_n = y_{n-1} + v_{y(n-1)}\,\Delta t - (½) g\, \Delta t^2$$
$$v_{y\,n} = v_{y\,(n-1)} - g\Delta t$$
$$z_n = z_{n-1} + v_{z(n-1)}\Delta t + s\,(x_n)^2$$
$$v_{zn} = v_{z(n-1)}$$

where s is a constant that controls the amount of swing.

The new state vector can be represented as

$$[x_n\ v_{xn}\ \ y_n\ v_{y\,n}\ z_n\ v_{zn}\ (x_n)^2]^T$$

The new state transition matrix is given by-

$$\begin{pmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Now UKF was used to track this updated problem. The number of sigma points were chosen as 15 this time as there are now 7 states in the state vector. The tracking estimates are as shown in fig 8 and fig. 9. It can be seen that the UKF algorithm was able to track the ball with a decent level of accuracy.

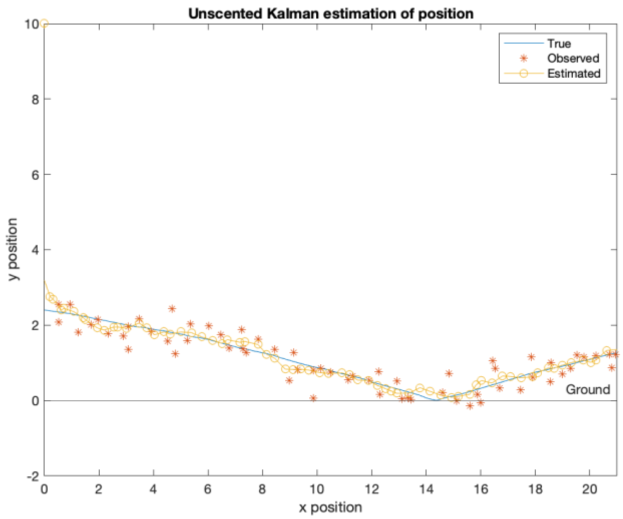

Fig. 8. Unscented Kalman filter estimation along x and y axis with non-linear swing (side view)

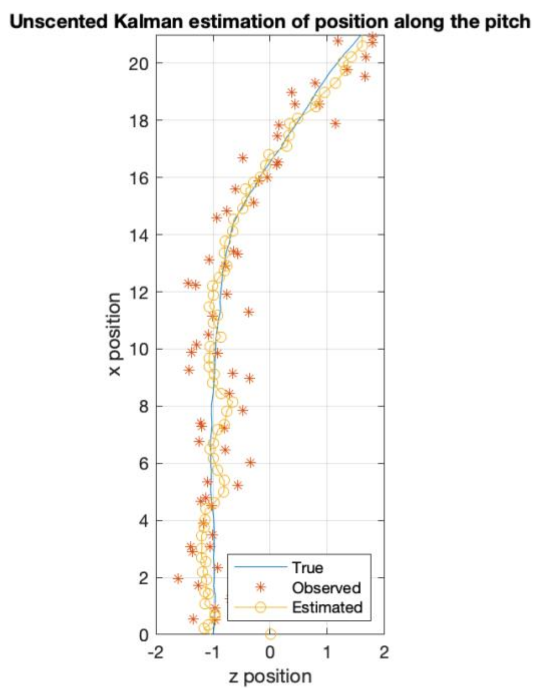

Fig. 9. Unscented Kalman filter estimation along x and z axis ith non-linear swing (top view)

V. RESULTS

The results in the form of prediction estimates have been shown through figures for each of the three cases discussed. The standard deviation for process noise was taken as 0.01 and measurement noise was taken as 0.1 for each case. It was also observed that on increasing these noise values, the prediction estimates became less accurate. In the last case with non-linear swing, the ball curved towards the right side in the air and this deviation increased with the distance travelled by the ball. But UKF algorithm was still able to track the movement of the ball with a decent accuracy.

VI. CONCLUSION

In this project, Kalman Filter and Unscented Kalman Filter were discussed and applied to track a cricket ball in three dimensions. First, a basic Kalman filter was used and then an Unscented Kalman filter was used to track the ball. Both filters were able to track the ball with some degree of accuracy. In a third case, UKF was also used to track the ball with an extra non-linear swing component. UKF was found to be very effective in tracking the non-linear system. UKF is a powerful nonlinear estimation method and has been shown to be a superior alternative to EKF in a variety of applications including state estimation [8]. The researches have shown that the UKF significantly outperforms the EKF for certain problems such as the attitude estimation and bearing-only tracking which are highly nonlinear systems [9].

In the future, more non-linearities involved in this tracking problem can be included in the system. Additional algorithms like particle filters can also be implemented and compared with the results obtained through UKF.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems", *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35-45, 1960.

[2] D. Salmond, "Target tracking: introduction and Kalman tracking filters," IEE Target Tracking: Algorithms and Applications (Ref. No. 2001/174), 2001, pp. 1/1-1/16 vol.2, doi: 10.1049/ic:20010245.

[3] Q. Li, R. Li, K. Ji and W. Dai, "Kalman Filter and Its Application," 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), 2015, pp. 74-77, doi: 10.1109/ICINIS.2015.35.

[4] J. K. Uhlmann, Simultaneous map building and localization for real time applications, 1994.

[5] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in Proceedings of the IEEE, vol. 92, no. 3, pp. 401-422, March 2004, doi: 10.1109/JPROC.2003.823141.

[6] Harveen Singh Chadha, "The Unscented Kalman Filter: Anything EKF can do I can do it better! Available online : "https://towardsdatascience.com/the-unscented-kalman-filter-anything-ekf-can-do-i-can-do-it-better-ce7c773cf88d

[7] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373), 2000, pp. 153-158, doi: 10.1109/ASSPCC.2000.882463.

[8] L. Bingrong, Q. Changwen, Z. Zheng and G. Linpeng, "Application of UKF Alogrithm in Passive Localization and Tracking," 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2018, pp. 2530-2534, doi: 10.1109/IAEAC.2018.8577916.

[9] S. Yang and H. Li, "Application of EKF and UKF in Target Tracking Problem," 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016, pp. 116-120, doi: 10.1109/IHMSC.2016.25.