# Comparison of UKF and EKF algorithms for the Single and Double pendulum systems

**Deepak Ganesh**
(**with Magnus Tibbe and Tawsiful Islam**)

## Abstract

In this project we try to present a comparison between the methods Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). We consider two systems, namely, the Single pendulum and the Double pendulum. Most importantly, we try to highlight why one of these methods, the Unscented Kalman Filter can perform better than the Extended Kalman Filter in presence of non-linearities in the system and what conclusions we can draw from the performance of either of these algorithms in the two specified systems.

## 1 Introduction

In this project, we consider two estimation algorithms from the family of Kalman filters, namely, the Extended Kalman Filter (EKF) and the Unscented Kalman filter (UKF). The EKF has been a popular choice for state estimation in systems where the process and measurement models are nonlinear, and it is widely used in the field of control and estimation, particularly in the areas of navigation, guidance, and control of spacecraft, aircraft, and mobile robots. Whereas, the UKF is also an extension of the Kalman Filter, it differs from EKF in the choice of points for linearization. The UKF has its applications in the same areas as EKF, especially when robustness is paramount. More detailed explanation of the algorithms will be given in the subsequent sections (3.4) and (3.5).

Our main objective is to show the differences between the algorithms by making use of the Simple and Double pendulum systems. The dynamics of a simple pendulum system is fairly simple. Whereas, the double pendulum is governed by a set of nonlinear differential equations, which are more complex than those for a single pendulum.

We consider this problem because they are two very similar physical setups, while their dynamics vary significantly in complexity, and this could indeed help us bring out the differences between the EKF and UKF algorithms. The dynamics of both of these systems are nonlinear differential equations derived from Newton's second law. They are then simulated over time for several iterations using the Runge-Kutta 4 (RK4) numerical method which is a nonlinear differential equation solver. The synthesized simulation results are then used as ground truth for the angle estimation of the pendulum, and also for the measurements. This can be equivalently considered as a sensor that would instead give us the measurements in real life scenario. We have also implemented functions that perform EKF and UKF estimation for the two systems respectively. The performance of the algorithms is measured by finding the mean absolute error of the angle estimates. We expect our results to show that UKF can perform better than EKF for a highly nonlinear system like the Double pendulum, while they are more similar in performance for a fairly simpler system like the Single pendulum.

Our main contribution in this project is

- Using the numerical solver to simulate the system first, and estimating the angles for both the pendulum systems using EKF and UKF respectively, and comparing their performance in different scenarios.

## 2 Background and related Work

Kalman filter has been one of the most well-known estimation methods. They assume a Gaussian distribution for the prior, the prediction belief and the actual belief. The Extended Kalman Filter (EKF) as explained in [**1**] is a method for estimating the state of a system when the system is modeled as a set of nonlinear equations. Since non-linear models can't be directly dealt with, the algorithm uses a linear approximation of the nonlinear equations to perform the estimation, which makes it computationally efficient. The EKF is a popular choice for state estimation in systems where the process and measurement models are nonlinear, and it is widely used in the field of control and estimation, particularly in the areas of navigation, guidance, and control of spacecraft, aircraft, and mobile robots. The EKF algorithm involves a two-step process: the prediction step and the correction step. The prediction step uses the current state estimate and the system model to predict the state at the next time step. The correction step uses the new measurement to update the state estimate. It can be sensitive to the initial conditions and the choice of the linearization point, and it can also be sensitive to the presence of non-Gaussian noise.

Due to some of the disadvantages of EKF algorithm centred around its linearization, a method called Unscented

Kalman filter (UKF) has been used. The UKF algorithm, first introduced in [**2**] is an extension of the Kalman filter that can be used to estimate the state of a system when the system is modeled as a set of nonlinear equations. It uses a technique called the unscented transformation to select a set of 'sigma points' that represents the belief of the current state. These points are chosen such that they are distributed around the mean of the distribution and captures the shape of the distribution more accurately than the single linearization point in the EKF. A extended analysis of the UKF algorithm from the point of view of control theory was presented in [**3**]. It is less sensitive to the choice of the linearization point, and it can handle non-Gaussian noise more robustly. It also avoids the problem of "filter divergence" that can occur with the EKF. The UKF has applications in the same areas as EKF, especially when robustness is required. However, it should be noted that the UKF can be computationally expensive compared to the EKF, because of propagating multiple sigma points through the model for every iteration of the algorithm.

Both single and double pendulum systems are really generic systems widely studied in the field of mechanics and physics, and have a variety of applications such as in clock design, seismology, and in understanding physical phenomena in physics.

The simple pendulum has a simpler mathematical model and it's behavior is more predictable, while the double pendulum has a more complex mathematical model and it's behavior is chaotic, and it can be used to study more complex systems such as the motion of planets and satellites. The simple pendulum also has a special case when the angle (with respect to vertical) is really small. $\sin\theta$ can be approximated to be $\theta$ and we get a linear dynamics.

$$\begin{aligned} \frac{d\theta}{dt} &= \quad \omega \\ \frac{d\omega}{dt} &= -\frac{g\theta}{L} \end{aligned} \tag{1}$$

The motion of a double pendulum is characterized by a complex, seemingly random motion, with the secondary pendulum's motion depending heavily on the primary pendulum's motion. The double pendulum's behavior is highly sensitive to the initial conditions, meaning that even small changes in the initial conditions can lead to vastly different behaviors over time, making the system unpredictable in the long term, and this makes it chaotic. This makes it an interesting system to research and study as done by for it's nonlinearity, as done in [**4**] where they get an accurate state estimation using UKF and try to control the system.

# 3 Approach and methodology

In this section we discuss in detail the theory that is most important to understand our approach.

## 3.1 Simple pendulum

A single pendulum system consists of a mass (often referred to as the "bob") attached to a fixed point by a rod or cord. The motion of the mass is governed by the equations of motion for a simple pendulum, which are a set of nonlinear differential equations derived from the Newton's second law of motion. They can be written as,

$$\begin{aligned} \frac{d\theta}{dt} &= \quad \omega \\ \frac{d\omega}{dt} &= -\frac{g\sin\theta}{L} \end{aligned} \tag{2}$$

where $\omega$ is the angular velocity, $\theta$ is the angle made by pendulum with vertical axis, $g$ is the acceleration due to gravity, and $L$ is length of the cord.

## 3.2 Double pendulum

A double pendulum system consists of two pendulum systems connected in series. The first pendulum (or the "primary pendulum") is attached to a fixed point, and the second pendulum (or the "secondary pendulum") is attached to the first pendulum's bob. The motion of the double pendulum is given by 4 nonlinear differential state space equations and is generally more complex in comparison to the simple pendulum. These equations also have to be expressed in the state space form that can be directly fed into the Runge-Kutta numerical solver, and hence the inspiration for these equations was taken from [**6**] which is devoted to double pendulum simulations and study.

$$\begin{aligned} \frac{d\theta_1}{dt} &= \quad \omega_1 \\ \frac{d\theta_2}{dt} &= \quad \omega_2 \\ \frac{d\omega_1}{dt} &= \frac{-g(2m_1+m_2)\sin\theta_1 - m_2 g\sin(\theta_1-2\theta_2)}{L_1(2m_1+m_2-m_2\cos(2\theta_1-2\theta_2))} \\ &\quad - \frac{2\sin(\theta_1-\theta_2)m_2(\omega_2^2 L_2+\omega_1^2 L_1\cos(\theta_1-\theta_2))}{L_1(2m_1+m_2-m_2\cos(2\theta_1-2\theta_2))} \\ \frac{d\omega_2}{dt} &= \frac{2\sin(\theta_1-\theta_2)(\omega_1^2 L_1(m_1+m_2)+g(m_1+m_2)\cos\theta_1)}{L_2(2m_1+m_2-m_2\cos(2\theta_1-2\theta_2))} \\ &\quad + \frac{2\sin(\theta_1-\theta_2)(\omega_2^2 L_2 m_2\cos(\theta_1-\theta_2))}{L_2(2m_1+m_2-m_2\cos(2\theta_1-2\theta_2))} \end{aligned} \tag{3}$$

where the subscript 1 and 2 refer to the top rod and the bottom rod respectively. $m$ refers to the masses attached to the rods, and the rest of the symbols refer to the same physical quantities as in 3.1.

## 3.3 Simulating the systems

To simulate the systems and retrieve the data from them we use a numerical method called the Runge-Kutta 4 (RK4), also known as the classical Runge-Kutta method was used. It is a widely used method for solving ordinary differential equations (ODEs). The RK4 method is a fourth-order, single-step method, which means that it uses four estimates of the derivative at different points in the interval to compute the next point. The RK4 method approximates the solution of the ODE at a discrete set of time steps. The RK4 method is a one-step method, which means that it uses only the information from the current time step to compute the next time step, this makes the method easy to implement and computationally efficient. We just give a brief information on the method and won't delve deep into the mathematics of it, since it's not relevant to the discussion.

The simulation results for each time step got from the RK4 solver is then used as the ground-truth data to measure the error of our two estimation algorithms. We also introduce some noise in these simulated values to incorporate them as our synthesized measurements to use for the measurement model.

### 3.4 Extended Kalman Filter (EKF)

It is an extension of the the Kalman filter for nonlinear state estimation. It also uses the linearizations of the dynamics and the measurement models using taylor series approximation and jacobian computations. After the linearization, it has a prediction and an update step. It estimates the mean and covariance $(\mu_t, \Sigma_t)$ for the given time step, given the mean and covariance $(\mu_{t-1}, \Sigma_{t-1})$ for the previous time step. We also inject a noise $\epsilon$ in the prediction step to inject some nonlinearity into the system.

### Linearization

$$\begin{aligned} g(u_t, x_{t-1}) &\approx g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1}) \\ h(x_t) &\approx h(\bar{\mu_t}) + H_t(x_t - \bar{\mu_t}) \end{aligned} \tag{4}$$

where $g(u_t, x_t)$ is the nonlinear state transition model given by the pendulum models in our case and $h(x_t)$ is the measurement model given directly by the simulating the system. $G_t$ is the jacobian for the nonlinear function $g(u_t, x_{t-1})$, $H_t$ is the jacobian for the nonlinear function $h(x_t)$ and $R_t$ is the process model noise's covariance.

**Prediction Step**

$$\begin{aligned} \bar{\mu_t} &= g(u_t, \mu_{t-1}) + \epsilon, \qquad \epsilon \sim N(0, 0.1) \\ \bar{\Sigma_t} &= G_t \Sigma_{t-1} G_t^T + R_t \end{aligned} \tag{5}$$

**Kalman Gain**

$$K_t = \bar{\Sigma_t} H_t^T (H_t \bar{\Sigma_t} H_t^T + Q_t)^{-1} \tag{6}$$

where $H_t$ is the jacobian for the nonlinear function $h(x_t)$ and $Q_t$ is the measurement model noise's covariance.

**Update Step**

$$\begin{aligned} \mu_t &= \bar{\mu_t} + K_t(z_t - h(\bar{\mu_t})) \\ \Sigma_t &= (I - K_t H_t)\bar{\Sigma_t} \end{aligned} \tag{7}$$

### 3.5 Unscented Kalman Filter (UKF)

UKF is another estimation method that doesn't require jacobian calculations and linearization approximations as in EKF. Instead, it uses an Unscented transformation to perform state estimation. The current belief state's mean and covariance are approximated by $(2n + 1)$ sigma points that are weighted appropriately. Thus, this algorithms involves 3

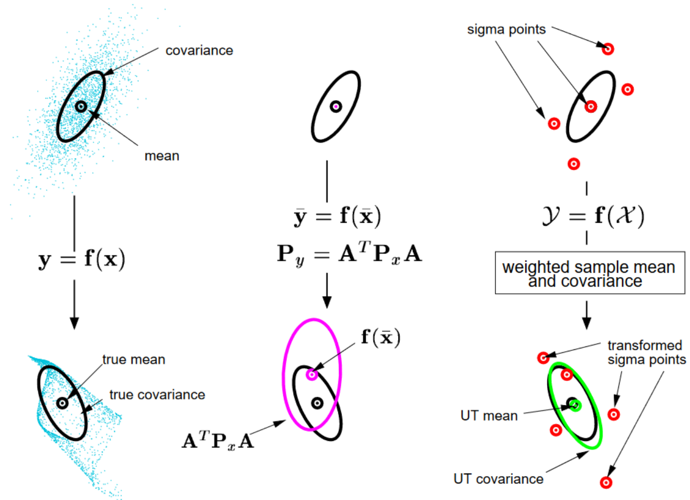


Figure 1: Illustration of the Unscented transformation for mean and covariance propagation **(a)** actual **(b)** through linearization (EKF) **(c)** Unscented transformation

steps given the mean and covariance from the previous time step. For more detailed explanation of the algorithm one could refer to [**2**]. We also inject a noise $\epsilon$ to add more nonlinearity to the prediction of the state as we did for EKF. ***Figure 1*** was used for illustration and was taken from [**5**].

**Calculation of sigma points**

Given the mean and covariance estimates $\mu$ and $\Sigma$ respectively, we can calculate the $(2n + 1)$ sigma points around the mean estimate with their weights, as follows:

$$\begin{aligned} \chi_0 &= \mu \\ \chi_i &= \mu + \left(\sqrt{(n+\lambda)\Sigma}\right)_i & i &= 1, \ldots, n \\ \chi_i &= \mu - \left(\sqrt{(n+\lambda)\Sigma}\right)_i & i &= n+1, \ldots, 2n \\ W_0^{(m)} &= \frac{\lambda}{(n+\lambda)} \\ W_0^{(c)} &= \frac{\lambda}{(n+\lambda)} + (1 - \alpha^2 + \beta) \\ W_i^{(m)} &= W_i^{(c)} = \frac{1}{(2(n+\lambda))} & i &= 1, \ldots, 2n \end{aligned} \tag{8}$$

where $W^{(m)}$ gives the weights for predicting the mean, and $W^{(c)}$ gives the weights for predicting the covariance. Also, $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter where $\alpha$ determines the spread of the $2n$ other points around the mean, $\kappa$ is the secondary scaling parameter usually set to 0, and $beta$ is to incorporate prior and is optimally set to 2.

We also have a matrix square root in $\left(\sqrt{(n+\lambda)P_x}\right)_i$ for which Cholesky factorization is used. It states that for a matrix $P$ that can be factorized as $SS^T = P$, the square root of full covariance matrix $P$ can be given by the matrix $S$ according to [**5**] where they use the concept of Square-Root UKF state estimation.

**Prediction**

$$\bar{\mu}_t = \sum_{i=0}^{2n} W_i^{(m)} g(\chi_{t-1}^i, u_t) + \epsilon, \qquad \epsilon \sim N(0, 0.1)$$
$$\chi_t^0 = \mu_{t-1}$$
$$\chi_t^i = \mu_{t-1} \pm \left(\sqrt{(n+\lambda)\Sigma_{t-1}}\right)_i$$
$$\bar{\Sigma}_t = \sum_{i=0}^{2n} W_i^{(c)} \left(g(\chi_{t-1}^i, u_t) - \bar{\mu}_t\right) \left(g(\chi_{t-1}^i, u_t) - \bar{\mu}_t\right)^T + R_t \tag{9}$$

where $g(x_t, u_t)$ is the nonlinear state transition model and $R_t$ is the process noise covariance.

**Update**

$$\begin{aligned} Z_t &= h(\chi_t, 0) \\ \hat{z}_t &= \sum_{i=0}^{2n} W_i^{(m)} Z_t^i \\ S_t &= \sum_{i=0}^{2n} W_i^{(c)} \left(Z_i^t - \hat{z}_t\right) \left(Z_i^t - \hat{z}_t\right)^T + Q_t \\ \bar{\Sigma}_t^{xz} &= \sum_{i=0}^{2n} W_i^{(c)} \left(\chi_t^i - \bar{\mu}_t\right) \left(\chi_t^i - \bar{\mu}_t\right)^T \\ K_t &= \bar{\Sigma}_t^{xz} S_t^{-1} \\ \mu_t &= \bar{\mu}_t + K_t(z_t - \hat{z}_t) \\ \Sigma_t &= \bar{\Sigma}_t - K_t S_t K_t^T \end{aligned} \tag{10}$$

where $Q_t$ is the measurement model noise, $(z_t - \hat{z}_t)$ gives us the innovation, $K_t$ refers to the Kalman gain and $\bar{\Sigma}_t^{xz}$ is the covariance of the sigma points before propagating them through the state transition model.

## 3.6 Estimation error calculation

We make use of the Mean absolute error (MAE), to calculate the error between the angles estimated by EKF and UKF, and the ground truth generated by simulations. The error can be given as follows:

$$(\text{MAE})_i = \sum_{n=1}^{i} \frac{|\theta_n^{est} - \theta_n^{act}|}{i} \tag{11}$$

where $\theta_n^{est}$ is the theta estimated using EKF or UKF, $\theta_n^{act}$ is the ground truth theta, and $i$ stands for the current iteration. So, this error value considers the sum of absolute estimation errors from the start to the current iteration, divided by the number of current iteration.

## 3.7 Sanity check on pendulum models

It's really easy to go wrong with the pendulum equations and their simulation through the numerical solver especially because of the complex dynamics of the double pendulum. Hence, we do a sanity check on the simulations by plotting the potential energy, kinetic energy and the total energy of the system. The Kinetic energy should always be non-negative, and zero at the position of release. On the other hand, the potential energy is always maximum at the position of release and becomes zero at the bottom when the angular velocity reaches its maximum value. But the total energy of the system is expected remain constant (with maybe small approximation errors from numerical solvers) once the pendulum is released, and the simulation begins. This is because for an system without external interactions, the total energy doesn't get dissipated and is expected to remain constant, but the constant value depends on the initial angle that we choose to release the pendulum from. In this way we can easily verify if the pendulum models that we implemented are right.

# 4 Results and Discussion

The UKF and EKF estimation methods described in the previous section were tested and compared with the numerical simulation through the two systems, namely, the Simple and Double pendulum. As mentioned before in section (3.3) the ground truth and measurement values were created by simulating the systems. Also, for comparison we refer to the following process ($R$) and measurement noise ($Q$) values which become appropriate covariance matrices according to the system we use:

- Low model covariance values $R, Q \sim N(0, 0.0001)$
- High model covariance values $R, Q \sim N(0, 0.01)$

## 4.1 Simple Pendulum

The simple pendulum as discussed in (3.1), is a fairly simpler system even though it has nonlinear dynamics. In this case UKF doesn't add anything new to the estimation than the EKF and both are expected to have a similar estimation trend, because the nonlinearity is easier to estimate.

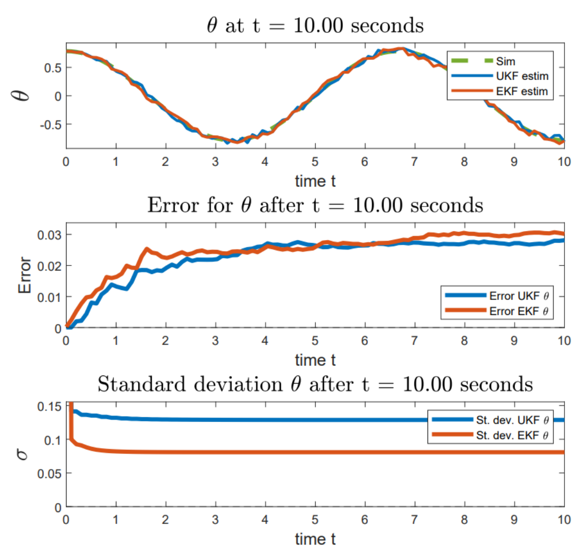


Figure 2: Plots of the estimated angle, mean absolute error and the standard deviation respectively considering a smaller initial angle and higher model noises

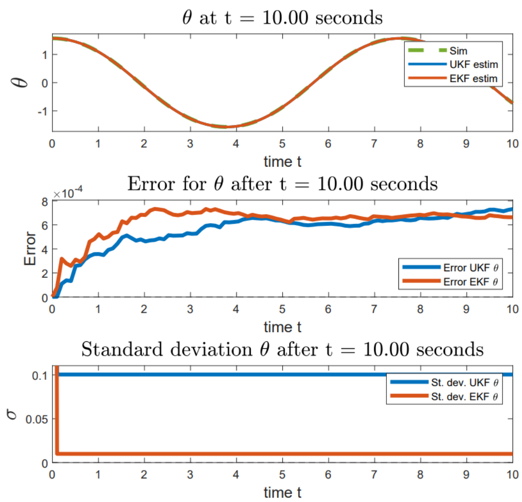


Figure 3: Plots of the estimated angle, mean absolute error and the standard deviation respectively considering a smaller initial angle, and process noise higher than measurement noise

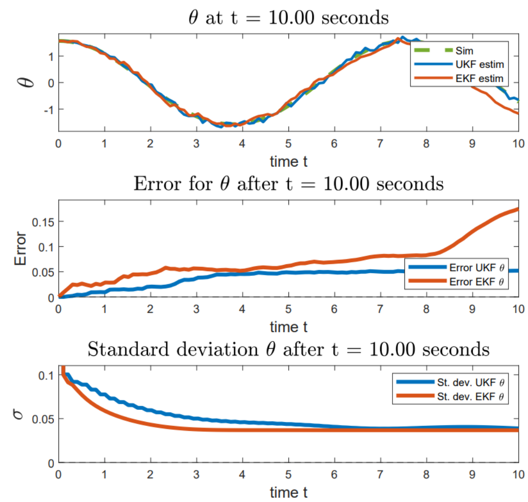


Figure 4: Plots of the estimated angle, mean absolute error and the standard deviation respectively considering a higher initial angle, and measurement noise higher than process noise

- For the first case, we simulate the simple pendulum from a smaller initial angle and estimate the angle $\theta$. From the first plot of ***Figure 2*** we can observe that the estimation accuracy of both the methods are good compared to the ground truth. From the second plot of ***Figure 2*** we can compare the performance of UKF against EKF, and can say that they have similar mean absolute error values which is expected when the simple pendulum is released from a smaller initial angle, because the model then tends to be more linear. The standard deviation (or the covariances in belief distribution of $\theta$) converge really faster and tends to be higher for UKF, because EKF generally tends to have a higher confidence in its estimates.

- For the second case, we consider a higher initial angle and release the pendulum from that position. The angle estimation is almost perfect if not perfect, because the errors are of the order of $10^{-4}$ as it can be seen in ***Figure 3***. This is primarily because we have a much lower measurement model noise ($Q$) as compared to the process noise ($R$), and the estimation of the angle is more reliant on the measurement model, because the measurements that stem from the simulated values match, and are accurate enough with the ground truth values. Also, UKF performs similarly to EKF, while being slightly better as we would expect in the case of a Simple pendulum.

- In the third scenario we consider a higher initial angle as well. The angle estimation has some errors as seen from ***Figure 4***, but the algorithms perform closer to each other for a fewer time steps, then the error from EKF starts to diverge and increase over UKF. In this case we switch the ratios of the model noises and have a higher measurement model noise this time. The state transition model has a lesser process noise (better model) and hence the linearization approximation in EKF comes into play to make it a bit worse than UKF, even if the simple pendulum model itself isn't highly nonlinear.

The other set of parameter choices seem to be conclusive of just the fact that the UKF is closer in performance but better than EKF for a simple pendulum, and that EKF tends to diverge in the case of having a lesser process noise compared to the measurement noise.

### 4.2 Double Pendulum

The Double pendulum is a more complex system than the simple pendulum, with a set of highly nonlinear equations describing its dynamics along with the complex jacobians required for the EKF algorithm.

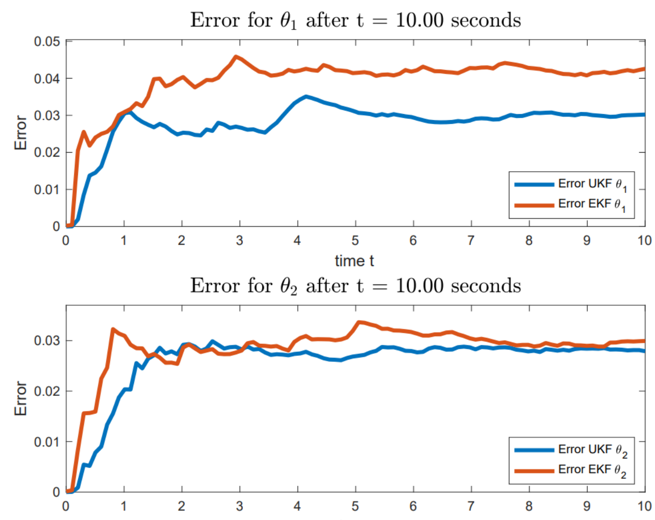


Figure 5: Plots of the mean absolute errors in the case of low process and measurement noises

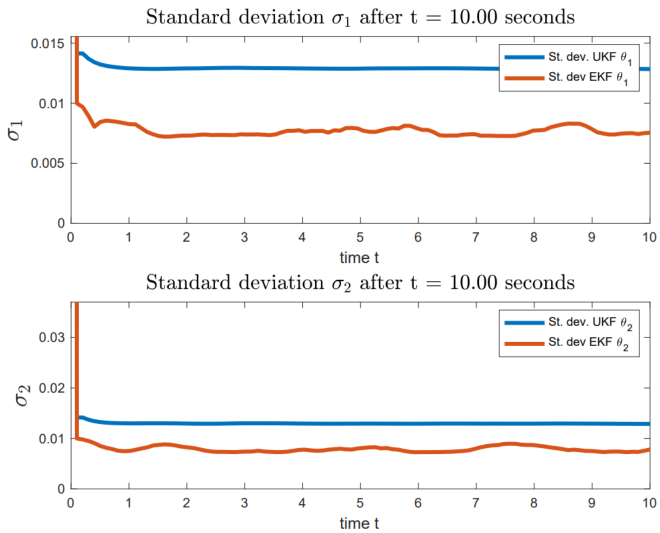


Figure 6: Plots of the standard deviations in the case of low process and measurement noises

- In the first case we have low process noise and measurement model noises, we get the mean absolute errors of the estimate to be better for UKF than the EKF convincingly for $\theta_1$ but the errors are more similar for the estimation of $\theta_2$ while UKF is still slightly better according to the ***Figure 5***. The standard deviations in ***Figure 6*** follow the trend and converge really soon to a really low value, and EKF has a lesser covariance when compared to estimation through UKF as expected because EKF tends to be more confident about its belief state distribution. While the difference in standard deviation is higher for the estimated $\theta_1$, it is lower for $\theta_2$.

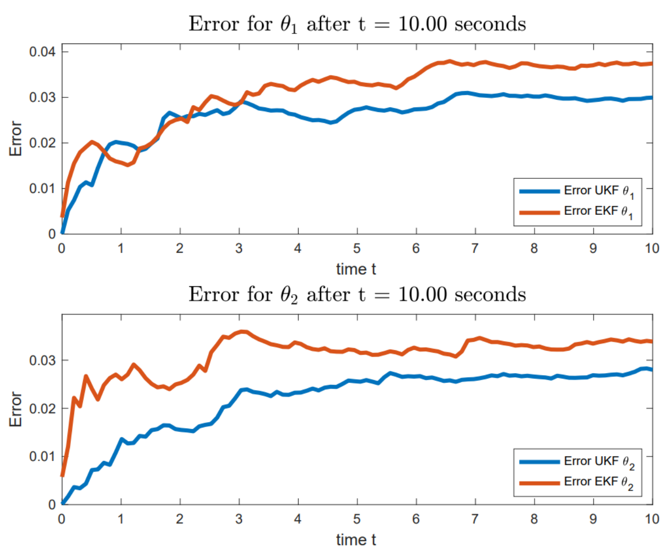


Figure 7: Plots of the mean absolute errors in the case of high process and measurement noises

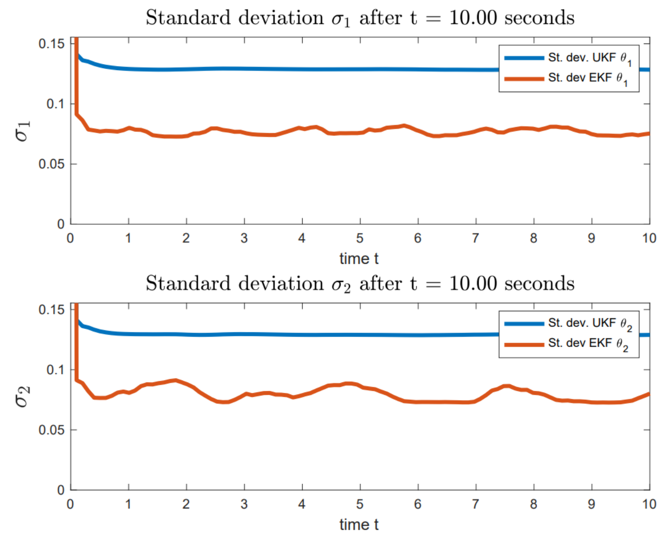


Figure 8: Plots of the standard deviations in the case of high process and measurement noises

- For the second case of having much higher process and measurement model noises, the mean absolute errors of EKF and UKF in ***Figure 7*** are more similar for the estimation of $\theta_1$ and has a significant difference in the estimation of $\theta_2$. The standard deviations follow the expected trend, while the differences between them remain more or less the same for estimation of both the states $\theta_1$ and $\theta_2$. We can also observe that the standard deviation values in ***Figure 8*** are relatively 10 times higher than the case where we have low noises. This is because the process and measurement noises add up to the estimated covariance values over each iteration.

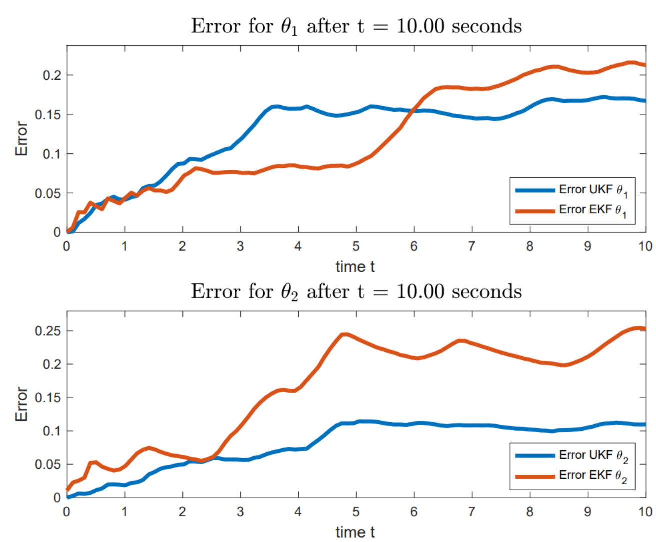


Figure 9: Plots of the mean absolute errors in the case of high process and low measurement noises

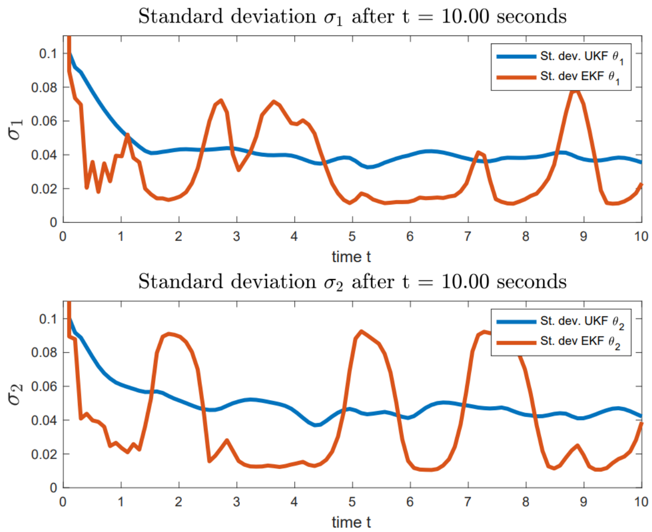


Figure 10: Plots of the standard deviations in the case of high process and low measurement noises

- In the third case, when we change the ratio of noises and have a high process noise and a low measurement noise in ***Figure 10***, we get an erroneous result for the standard deviations of either angles estimated through the EKF algorithm, whereas it slowly converges for the UKF algorithm. As we can see in ***Figure 9***, EKF starts to diverge more with the iterations, whereas we can see that the UKF doesn't seem to do so. We can also observe that the errors due to UKF remain significantly lower than EKF while estimating $\theta_2$.

From the results of our plots for the double pendulum, we observe in general that the estimation trend doesn't depend much on the initial angle of release of the double pendulum (assuming that both the rods are collinear before their release, i.e. $\theta_1 = \theta_2$). In this system, the dynamics of the second rod depends more on the movement of the first rod because it is constrained to a fixed point.

## 5 Conclusion

Through an analysis of both the systems we attempt to present a comparison between the estimation methods of UKF and EKF. The theory and background of EKF, UKF and the pendulum systems were discussed in detail. The algorithms were then tested through several simulations and using simulated ground truth and measurement values. In one of the cases, the comparison between the algorithms was ambiguous for the double pendulum, and this didn't give us a convincing result to reach a conclusion. But in general, from the rest of the comparisons, UKF seems to be the better choice for a highly nonlinear system such as the double pendulum because the error converges irrespective of the initial estimates, whereas their performance is more similar for a much simpler nonlinear system such as the single pendulum.

## References

[1] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. (2005) *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press.

[2] E. A. Wan and R. Van Der Merwe (2000) *The unscented Kalman filter for nonlinear estimation*, Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium

[3] Rambabu Kandepu, Bjarne Foss, Lars Imsland (2008) *Applying the unscented Kalman filter for nonlinear state estimation*, Journal of Process Control.

[4] Xu, Bowen; Lyu, You and Gadsden, Stephe (2018). Estimation And Control Of A Double-Inverted Pendulum. 10.25071/10315/35250.

[5] R. Van der Merwe and E. A. Wan (2001) *The square-root unscented Kalman filter for state and parameter-estimation*, IEEE International Conference on Acoustics, Speech, and Signal Processing.

[6] *https://www.myphysicslab.com/pendulum/double-pendulum-en.html*, Erik Neumann (2000)