Simple Example of Applying Extended Kalman Filter

Keatmanee Chadaporn
Japan Advanced Institute of Science and Technology
7 PUBLICATIONS **13** CITATIONS
SEE PROFILE

Junaid Baber
University of Balochistan
30 PUBLICATIONS **126** CITATIONS
SEE PROFILE

Maheen Bakhtyar
Asian Institute of Technology
25 PUBLICATIONS **90** CITATIONS
SEE PROFILE

Simple Example of Applying Extended Kalman Filter

Chadaporn Keatmanee
Faculty of Information Technology
Sripatum University
Bangkok, Thailand
Email: chadaporn.ke@spu.ac.th

Junaid Baber
Computer Science
and Information Management
Asian Institute of Technology
Pathumtani, Thailand
Email: junaid.j.baber@ait.asia

Maheen Bakhtyar
Computer Science
and Information Management
Asian Institute of Technology
Pathumtani, Thailand
Email: maheen.bakhtyar@ait.asia

***Abstract*—Kalman Filter (KF) is a well-known algorithm for estimation and prediction especially when data has a lot of noise. KF is used for linear transition functions whereas under non-linear transition, Extended Kalman Filter(EKF) is used. The EKF is also considered to be the de-facto standard. There are tons of papers about EKF however very few of them show working of the algorithm. The purpose of this paper is to explain how to apply EKF on simple application, explaining model creation, substitution, simulation and illustration of adjusting an accuracy of the model.**

Keywords: Kalman Filter, Extended Kalman Filter, Kalman filter application

I. Introduction

KF is a recursive algorithm used for estimating a dynamic system which lacks data. The cause of lack of data is noise, or environment, example of such systems includes: autonomous and assisted navigation system. It uses prior knowledge to predict the past, present, as well as future state of system. The advantage of KF is that the data is updated in each and every iteration so that we dont need much memory to keep all data of the system for prediction. The only difference between KF and EKF is that for EKF we have to linearize, non-linear function using Jacobian matrix then we could use the rest of process of the KF algorithm.

KF is a popular recursive algorithm for motion prediction[1] as this simple application which has purpose to predict the next position of the bug movement walking in straight line. The principle of Kalman Filter is to find the probability of the hypothesis of predicted state is given by hypothesis of prior state and then using the data from measurement sensor to correct the hypothesis to get the best estimation for each time. Basically, Applying KF has to concern about model creation including state and measurement model[2] as shown in equation 1 and 2.

$$\mathbf{x}_{k+1} = A\mathbf{x} + B\mathbf{u}_k + \mathbf{w}_k \tag{1}$$

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k \tag{2}$$

The first equation is linear stochastic equation means that each $\mathbf{x}_k$ is evaluated[3]. The second equation explains that any measured value, unsure of its accuracy, is a linear combination of the measurement noise and the signal value and both of these are considered to be Gaussian. Both noise and measured noise are statistically independent in the process.

Even though the entities **A**, **B** and **H** are general form matrices, in most of our models for signal processing problems, we treat these entities to be just numeric values. In addition to further easing out, even these values may change between states; it can be assumed them to be constant, most of the times.

Upon being sure that our system fits into this model, which was difficult to be fitted in the first time, estimation of the mean and standard deviation of the noise functions $\mathbf{w}_k$ and $\mathbf{v}_k$ would be the only thing left to do. Even though in reality no signal is pure Gaussian, we may assume it with some approximation. This will not pose a problem, as we will see that the Kalman Filtering Algorithm tries to converge into correct estimations, even if the Gaussian noise parameters are poorly estimated.

KF results from the continuously updating the estimation data based on prior knowledge in each and every iteration of prediction and filtering[4] . The changing in each iteration is derived and interpreted in the term of Gaussian probability density functions. The innovation process of KF associates with the filter, that represents the novel prediction of information in noisy environment conveyed to the state estimate by using data in the last system measurement.

When either the system state (a state model) dynamics or the observation (measurement model) dynamics is nonlinear, the conditional probability density functions that provide the minimum mean-square estimate are no longer Gaussian. The optimal non-linear filter propagates these non-Gaussian functions and evaluate their mean, which represents an high computational burden. A non-optimal approach to solve the problem, in the frame of linear filters, is the Extended Kalman filter (EKF).

The EKF implements a KF for a system dynamics that results from the linearization of the original non-linear filter, KF around the previous state estimates using Jacobian matrix sometime call only Jacobian.

II. Extended Kalman Filter

The most important part of applying EKF is model creation using knowledge of mathematics to create transition function which is non-linear for unknown parameter in each state of estimation. There are 2 models for EKF including

$$\text{State Model} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k + \mathbf{w}_k) \tag{3}$$

Where

$\mathbf{x}$ is the state model which composes of parameters using for prediction in each state

$\mathbf{x}_{k+1}$ is the next state that we should get the predicted data from using transition function which is non-linear function

$\mathbf{u}_k$ is the control data which is optional

$\mathbf{w}_k$ is Gaussian white noise

$$\text{Measurement Model} \quad \mathbf{z}_k = h(\mathbf{x}_k + \mathbf{v}_k) \tag{4}$$

Where

$\mathbf{z}_k$ is the measurement model including parameters which data come from various kinds of sensors such as thermal sensor and infrared sensor

$\mathbf{v}_k$ is Gaussian white noise

The process of the recursive algorithm, EKF is depicted in figure1. There are 2 main parts including prediction and correction state. At the correction state, using $\hat{\mathbf{x}}_k^-$ and P_k^- obtained from prediction state where $\hat{\mathbf{x}}_k^-$ is calculated by using state transition model and P_k^- is user defined, to calculate Kalman Gain or **K** which represents the trustable value of state model and measurement variable. In case, **R** approaches to 0, it means measurement variable is more trustable than state model, but if P_k^- approaches to 0, then it is vice versa. After that, estimated data is updated based on **K** values. Finally, the error covariance is updated for next iteration. Therefore, all parameters are updated in each and every iteration hence; predicted and estimated data continue to become more accurate.

The important feature of EKF is linearization of non-linear function f and h using Jacobian matrix which almost is first order derivative. The example of calculating Jacobian matrix for State transition matrix is shown as equation (5)

$$\mathbf{A}_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[i]}}(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0) \tag{5}$$

The Jacobian of functions $f_i(\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}), i = 1, 2, ..., n$ of real variables $\mathbf{x_i}$ is the determinant of the matrixwhose ith row lists all the first-order partial derivatives of the function $f_i(\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n})$.

The estimated values for state model and measurement model come from equations shown below

$$\mathbf{x}_{k+1} \approx \tilde{\mathbf{x}}_{\mathbf{k+1}} + A(x_k - \hat{x}_k) + Wwk \tag{6}$$

$$\mathbf{z}_k \approx \tilde{\mathbf{z}}_{\mathbf{k}} + H(x_k - \hat{x}_k) + Vvk \tag{7}$$

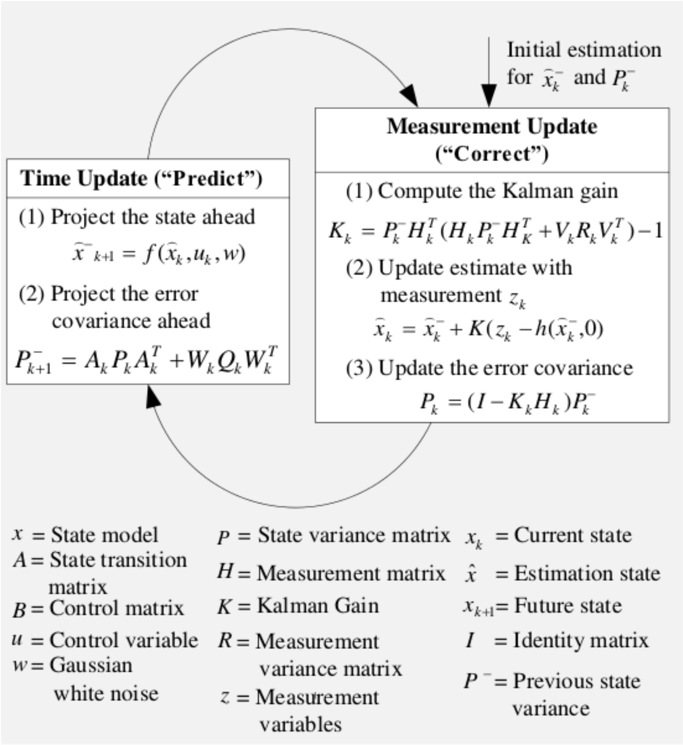
Initial estimation for $\hat{x}_k^-$ and P_k^-
Measurement Update ("Correct")
(1) Compute the Kalman gain
$K_k = P_k^- H_k^T (H_k P_k^- H_K^T + V_k R_k V_k^T) - 1$
(2) Update estimate with measurement z_k
$\hat{x}_k = \hat{x}_k^- + K(z_k - h(\hat{x}_k^-, 0)$
(3) Update the error covariance
$P_k = (I - K_k H_k) P_k^-$
Time Update ("Predict")
(1) Project the state ahead
$\hat{x}^-_{k+1} = f(\hat{x}_k, u_k, w)$
(2) Project the error covariance ahead
$P_{k+1}^- = A_k P_k A_k^T + W_k Q_k W_k^T$
x = State model
A = State transition matrix
B = Control matrix
u = Control variable
w = Gaussian white noise
P = State variance matrix
H = Measurement matrix
K = Kalman Gain
R = Measurement variance matrix
z = Measurement variables
x_k = Current state
$\hat{x}$ = Estimation state
x_{k+1} = Future state
I = Identity matrix
P^- = Previous state variance

Fig. 1. A complete process of the operation of the Extended Kalman Filter[2]

III. Applying Extended Kalman Filter

For applying both KF and EKF, knowledge of mathematics, physics, probability and statistics is required for model creation which is the most important. In this paper, we examine applying of simple applicant using bug movement in the straight line. There are 6 steps to apply EKF [5] and described as an example below:

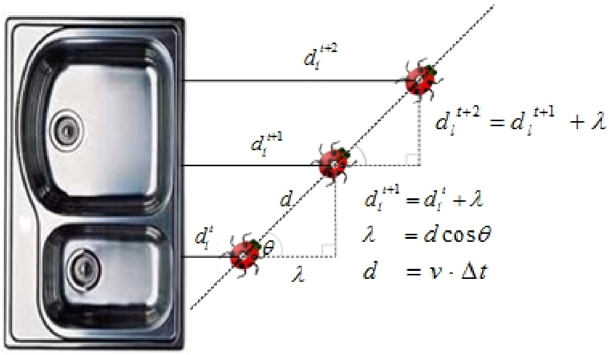

Fig. 2. A bug is walking in a stright line

1) **Understanding the current system status**
The simple application as an example is shown in figure 2 consisting of a dynamic system. Assume a bug is moving away from the sink measuring the angle and velocity of the moving bug in a straight line. The movement is linear and the only parameter changing is the distance of the bug from the sink and our objective

is to estimate the lateral distance between them.A lot of noise during the measurement is expected due to the measuring equipment such as a digital distance meter. Cosine function is used to create the state model and hence, we apply EKf and not KF, sine being a non-linear function.

2) **Modeling the state process**
The state model is shown in equation 8 where $\mathbf{x}_t$ represents the current state and $\mathbf{x}_{t+1}$ denotes the next state that needs to be predicted. The defined transition function is $d_l + \lambda$ in the model that estimates the lateral distance of the bug from the sink.

$$\mathbf{x}^t = \begin{bmatrix} {d_l}^t \\ \theta^t \\ \mathbf{v} \end{bmatrix}, \mathbf{x}^{t+1} = \begin{bmatrix} {d_l}^t + \lambda \\ \theta^t \\ \mathbf{v} \end{bmatrix} \tag{8}$$

Where ,

$\mathbf{x}$ is the state model

d_l is the lateral distance between edge of sink and the bug at the nearest point to the bug origin

θ is the tangent angle of the edge of sink at the point closest to the bug origin

$\mathbf{v}$ is velocity of the bug movement

Transition matrix A_t that is obtained from Jacobain matrix for linearization is shown in equation 9

$$A_t = \begin{bmatrix} 1 & \Delta_t * v * (-sin\theta) & \Delta_t * cos\theta \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

To keep the demonstration simple, we only discuss the lateral distance prediction (d_l) keeping the distance changing and the rest of the values are kept constant.

3) **Measurement process modeling**
The measurement parameters are constant keeping the example simple. They are the constant values obtained from the digital distance meter.

$\mathbf{z}$ is value measured using digital distance meter

$\mathbf{H}$ is 1 because no unit transformation

4) **Noise modeling** Kalman and extended Kalman filter use Gaussian white noise that makes the variance and covariance meaningful. Noise is introduced in the filter as in the matlab code[1].

5) **Filter Testing** Matlab is used to simulate the execution of EKF algorithm. The results of our simulation are shown in figure 3. The model successfully works and the estimated value is very close to the actual true value even when a lot of noise is introduced.

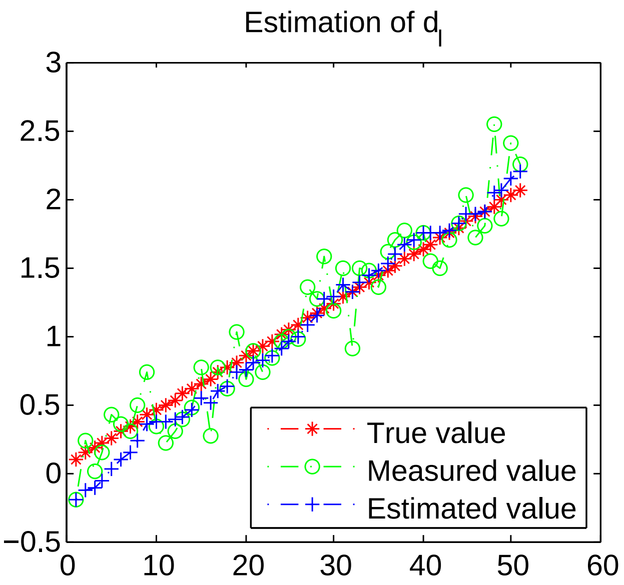

Fig. 3. A successfully working model simulation with R=0.05

6) **Filter Refinement** The EKF is refined by changing the value of the noise parameters such as P (state variance variable) or R (measurement variance variable). In our experiment we change the values of R, being a trustworthy candidate of measurement variable, and analyze its effect.
The results of our simulation(figure 3, figure 4 and figure 5) depicts the effect of varying the values of R In figure 3, R is assigned the value 0.05 that means medium level of trust in the measured value by the sensor, moreover it is suitable for successfully model.
In figure 4, R is given the value 0.001 depicting a trust in the measured variables obtained from the sensor because it is approached to 0 so the estimated value is following the measured value and not the actual true value.
In figure 5, R is assigned the value 0.1, quite different from 0, depicting a no trust state on the measured variable. Therefore, the estimated value is based on the results obtained from the transition function in the state model.
Defining the noise parameters show that every parameter in EKF is important for adjusting an accuracy level of the model.

IV. Conclusion

Our results show that the novel EKF algorithm is working quite satisfactory even in the presence of high noise from the sensor. The estimated values are obtained from EKF using state and transition model. Both the models are extremely important for the prediction. If any of these models is incorrect then it is not useful and meaningful for the EKF. Accuracy of the system can be increased if the model is

[1] can be downloaded from http://chadaphone.wordpress.com/2012/10/19/simple-example-of-applying-extended-kalman-implementation-with-matlab/

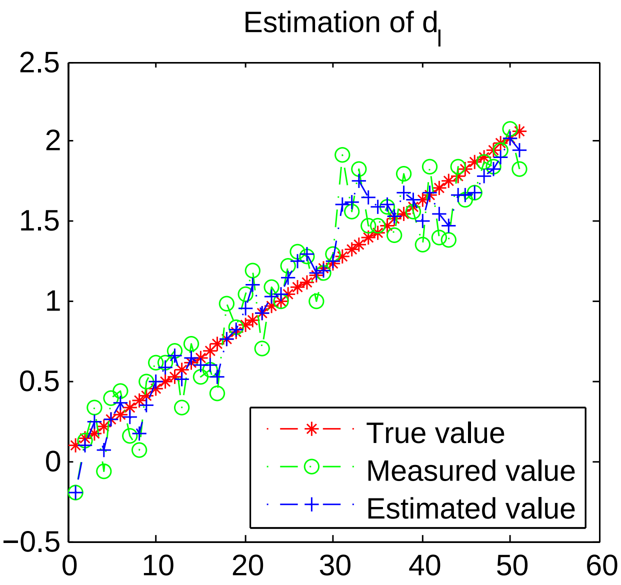

Fig. 4. The simulation with R=0.001

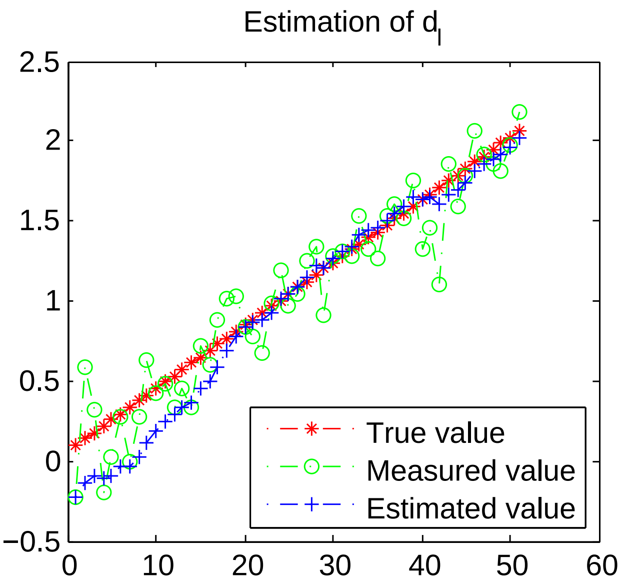

Fig. 5. The simulation with R=0.1

kept simple at first and then tested for the complex cases. Moreover, the noise parameters also play a very important role and the model can be defined by adjusting the parameters such as R and P to make the model more accurate.

Acknowledgment

I would like to thank EEAAT Conference Publishing for providing their guidelines and opportunity of sharing the knowledge. I sincerely thank Associate Professor Matthew N. Dailey for his valuable comments, suggestions, and motivation.

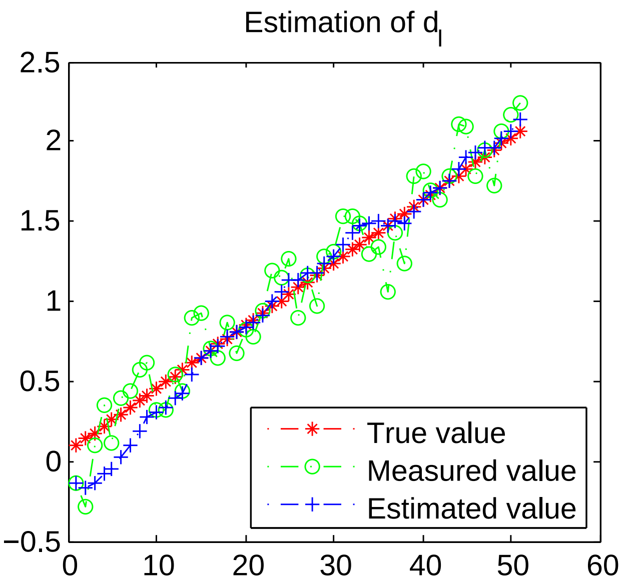

Fig. 6. The simulation with R=0.08

References

[1] R. Azuma and G. Bishop, “Improving static and dynamic registration in an optical see-through hmd,” in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, 1994.

[2] G. Welch and G. Bishop, “An introduction to the kalman filter,” 1995.

[3] B. Esme, “Kalman filter for dummies.” [Online]. Available: http://bilgin.esme.org/BitsBytes/KalmanFilterforDummies.aspx

[4] C. Derivation, , and M. I. Ribeiro, “Kalman and extended kalman filters:,” 2004. [Online]. Available: http://www.isr.ist.utl.pt/~mir

[5] “Kalman filter applications.” [Online]. Available: www.cs.unc.edu/~welch/kalman/media/pdf/kftool_models.pdf