1. The extended Kalman filter

%%% Kim’s comment : Non-linear Estimation

For non-linear system, to estimate parameters is totally dependent of the considering system. The way to find “Optimal” estimator is not unique.

There are several methods

-. Non-linear least square method. This is the basic and popular in the control society. It was already proposed and to solve some problems, See Excersize 4.8, by Gauss. He solved using the batch type LSE. For the recursive, or real time type, it was introduced 1970 by Jazwinski, so called Extended kalman filter, EKF, after kalman introduction for linear system 1960.

-. Unscented Kalman filter: around 1990, English engineers Julier introduce a brilliant idea different from EKF.

-. Particle Filters, which is totally dependent of Monte Carlo simulation, was introduced around 2000. It is goof for ‘Robotic Localization’, ‘Automatic Car driving ‘ and so on.

%%%

%%% Kim’s comments : non linearity

Consider a non-linear differential equation

1. The stationary points

The points so that implies the trajectory is stationary at that point. Hence

1. At the stationary points, find the nominal solution.

Taylor series:

For a small perturbation around a, the taylor series of is at the stationary points

Hence the differential equation (1) can be approximated near the stationary points. For example, at the stationary point “a = 0” a small perturbation

And the left hand side

Combining together,

which is a linear differential equation with respect to , since in case of (1), ,

gives the solution as

In conclusion near , x(t) converges to x(t) = 0

%%%

%%% HW Week\_10

1. How about the other stationary points?
2. At the initial point draw the trajectory.
3. At the initial point , draw the trajectory. %%%

Here I want to talk about non-linear least square estimation method. Remember at

at Chapter 4.4 about “Non-linear Square Estimation”, which is the batch type. In this chapter, I will consider it with recursive Non-linear Least Square estimation.

* 1. Linearized Kalman filtering
     1. Continuous time

The EKF is

And

With initial points as

%%% kim’s comments:

1. the EKF is not the linear dynamic equation as following.
2. We need the covariance of estimators. In general the covariance of the non-linear SDE is impossible. To get the covariance we need to be linearization.
3. The optimal observer system matrix is

%%%

* 1. The extended Kalman Filter (in discrete case)

%%% Extended Kalman Filter in discrete version

<https://en.wikipedia.org/wiki/Extended_Kalman_filter#:~:text=In%20estimation%20theory%2C%20the%20extended,the%20current%20mean%20and%20covariance>.

* Modeling

%% Kim’s comment:

* Non-linearities are in the system and measurement matrices. %%%
* Predict(A priori)
* Update(Correction / Estimation)
* Innovation:
* (Near-Optimal) Kalman gain
* Update (A posteriori)

Here

1. It is suboptimal
2. The gain formula equation is different from the previous ones(last semester) but the result is same
3. The innovation may be defined as

In some references, but it may be not correct.

1. This formulae is referred as the extended discrete Kalman filter

|  |  |  |
| --- | --- | --- |
|  | Linear Model | Non Linear Model |
| Model |  |  |
| Prediction  (A posteriori) |  |  |
| (A priori) |  |  |

* **Unscented Kalman Filter – Reference [1]**

This topic is not in the textbook , however it was invented recently for non-linear filtering with gaussian additive noise.

1. Unscented Kalman

The difference between EKF and UKF(unscented Kalman Filter)

1. EKF : need to calculate the Jacobian matrix

UKF : does not need the Jacobian

1. UKF is better than EKF.
2. Simple example- one dimensional case

Consider a random variable which is a non-linear transform of a gaussian R.V.

Find the mean and variance of

* Method 1: Find the PDF : using generating function

Let

Then using the definition of the probability distribution,

<https://stats.stackexchange.com/questions/192807/pdf-of-the-square-of-a-standard-normal-random-variable>

Then **we may find the mean and the variance of y**

* Method 2 : Monte Carlo simulation

We may realize (or simulate) to solve it as many as possible using COMPUTER. Let a simulation in matlab code

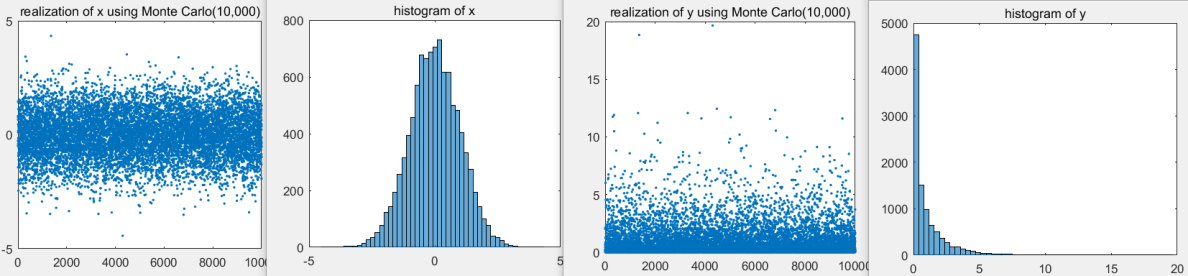
n = 10000; % the number of simulations , as Monte Carlo

x = std.\*randn(n,1); % mean=0,

histogram(x,50)

y = x.^2;

histogram(y,50)



>> mean and variance of x 0.00 and 1.00 using Monte Carlo

mean and variance of x^2 1.00 and 2.04

Hence by realization(or simulation) the mean and the variance of are similar to the analytic solution. We may call “A realization” as “A sigma point”.

* + Method 3: Analytic way without PDF

**Let as**

Then

The non-linear transform is

Therefore taking expectations

For the variance of

Taking expectations

Since

The variance of is

Now if then

* + Method 4: Linearize method

Since

If we take the first order term as linearization procedure in general,

Taking expectations

For the variance

Now if then

* + Summarize

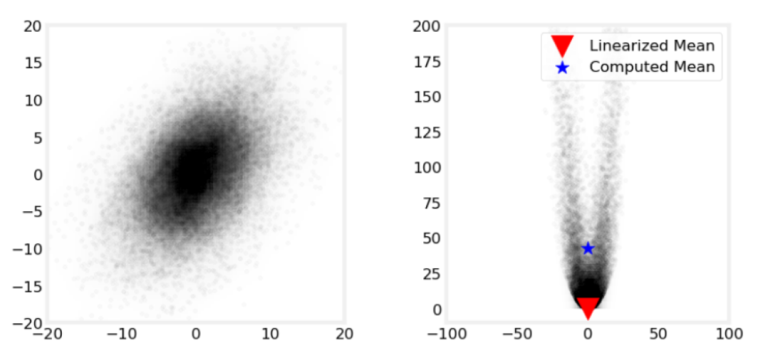
|  |  |  |  |
| --- | --- | --- | --- |
|  | Mean | Variance |  |
| PDF | 1 | 2 | difficult |
| Monte Carlo | 1 | 2.04 | Simple but costly and not realtime |
| Without PDF | 1 | 2 | Comparatively easy |
| Linearized | 0 | 0 | Too large of error |

1. Another example for multivariate case

Consider two gaussian random variables as

And its non-linear transformed system as

Then Monte Carlo simulation 10,000 times,



Here, the mean of the using Monte Carlo is (0.1064, 43.3120).

Now for linearization method,

In this figure, the errors between the Monte and the linearized mean are

%%% **HW.3.1**

3.1 prove pdf of as (1) and find the mean and the variance if

%% Hint : to get the mean and variance, using symbolic math in matlab

3.2 To get the mean and the variance of y, use Monte Carlo with different number of simulation to see the result, i.e., n = 1000,5000, 50000.

3.3 Prove (2), --- we may call this is ‘kurtosis” in the case of

%%%

1. The Unscented transformation

%%% Kim’s comment

The anther method, which is a brilliant idea, is developed in the statistics, where the Monte Carlo method is popular to get probabilistic characters. One of the deficiencies of Monte Carlo simulation is that the cost is very high and it is difficult to be a real time. See [3] %%

4.1 Unscented Transform basic idea

Let’s non-linear transformed function

To implement a Kalman filter, it is needed to get the mean and variance of .

In the previous examples, to get PDF, Monte Carlo , or without PDF, still it is difficult. Now we may introduce “Unscented Transform” to solve(or get approximate) to this problem.

1. The basic idea : Revisit Taylor series

A continuous function which is differentiable at any order, then it can be represented by polynomial expansion as

This polynomial expansion is satisfied for any “ ”. For example let

And the polynomial expansion at ,

Is it simple? How about this at

hence , then

Divide at both side by gives

Good… **Now the unscented kalman is “taylor expansion when the non linear function has a random variable !!** %%%

1. The problem

Let is a Gaussian random variable, i.e.,   
 A second random variable is related to through the nonlinear function

Find

* 1. Taylor series in case of random variables..

Consider a nonlinear transform of a Gaussian random vectors )

Let , then using taylor expansion at , here

* 1. Mean of

Now we consider a small variation around , , here . If is substituted to the above equation,

Now take ‘expectation’ at both sides ,

Since a Gaussian pdf is symmetric with respect to the mean, the odd terms as

the equation will be

If is small enough so that the forth moment and the higher orders will be neglected so that the mean will be approximated to

Here one of important facts. The mean of the nonlinear transform function is dependent of not just **the mean** of but **the variance** of the perturbation near the mean.

%%% Kim’s comment

In general, the first approximation of the nonlinear function is

so that the first derivative, we may call it a gradient, matters. But here the second derivative matters. In case of EKF, the approximation is done as

%%%

* 1. Covariance of

It will be shown as

As approximation,

* 1. Unscented Transform[2]

In order to get the mean and covariance of the transformed nonlinear function, if it is difficult to get it analytically, one method is a Monte Carlo simulation, which is needed a lot of sigma points. This is too heavy. Another way is introduced as a Unscented Transform which is sampling from the distribution function not by random.

* + 1. **Define sigma points**

Define sigma points with weighting function

where

%%% Kim’s comment

1. So how many sigma points? If the numbers is
2. Here will be chosen later, as 1,2,3…, as a tuning parameter. In general, is a Gaussian, then
3. The weighting factor has a constraint as
4. what is , the square root of a symmetric square matrix?

It is defined as Let then there exists N such that

Then

It is unique. Be careful as

which may not be unique.

How to get N? It is called “Cholesky Factorization” . And in matlab

>> N = chol(M)

%%%

* + 1. **Initiate the nonlinear transformed points** as

Then the mean is

And the covariance of

%%% kim’s comment:

In order to calculate the mean and the variance of the non-linear transformed function is that

1. No need the pdf of the nonlinear function
2. From (e) and (f), the mean and the variance is determined just the weighted sum of the sigma points.
3. Of course (e) and (f) are approximated values of the real, however, in UKF, (e) and (f) will be used.
4. The mean of the state is
5. The covariance of the state is

%%

* + 1. **Example**
* Example of UT in

Consider a non-linear system

1. Define sigma points

Since n=1,

Where

1. Projection these points thru
2. Calculating to get

which is independent of the tuning parameter , , which is the same of ““TRUE mean of y in 2.1 and 2.3”.

1. Calculating the variance of

Which is the same of “TRUE variance of y in 2.1 and 2.3” if

[1]” Kalman and Bayesian Filters in Python”, ch.10

[2]” A New Extension of the Kalman Filter to Non-linear System”

[3] “A General Method for Approximating Nonlinear Nonlinear Transformations of Probability Distributions