Mobile Robot
Introduction to Probabilistic Methods
Lecture 6

Jeong-Yean Yang

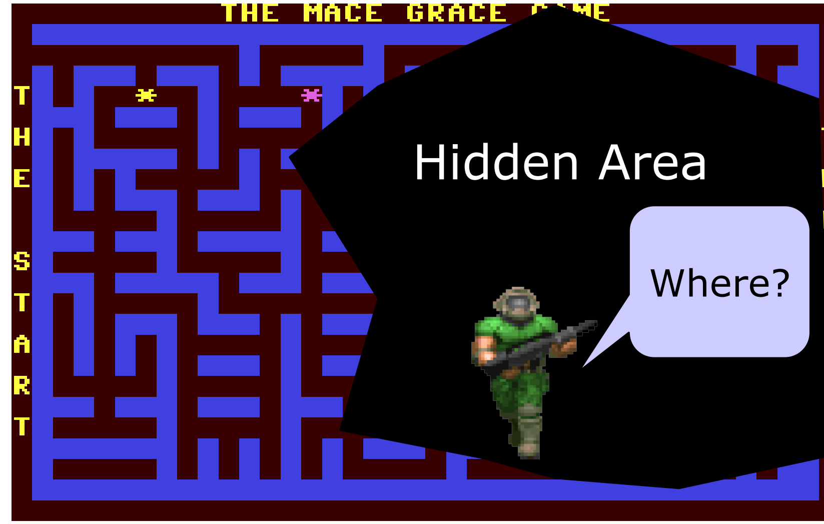
2020/12/10

1

Probabilistic Approaches for SLAM

Why we use Probability?

Why SLAM is Problematic?



Where am I?

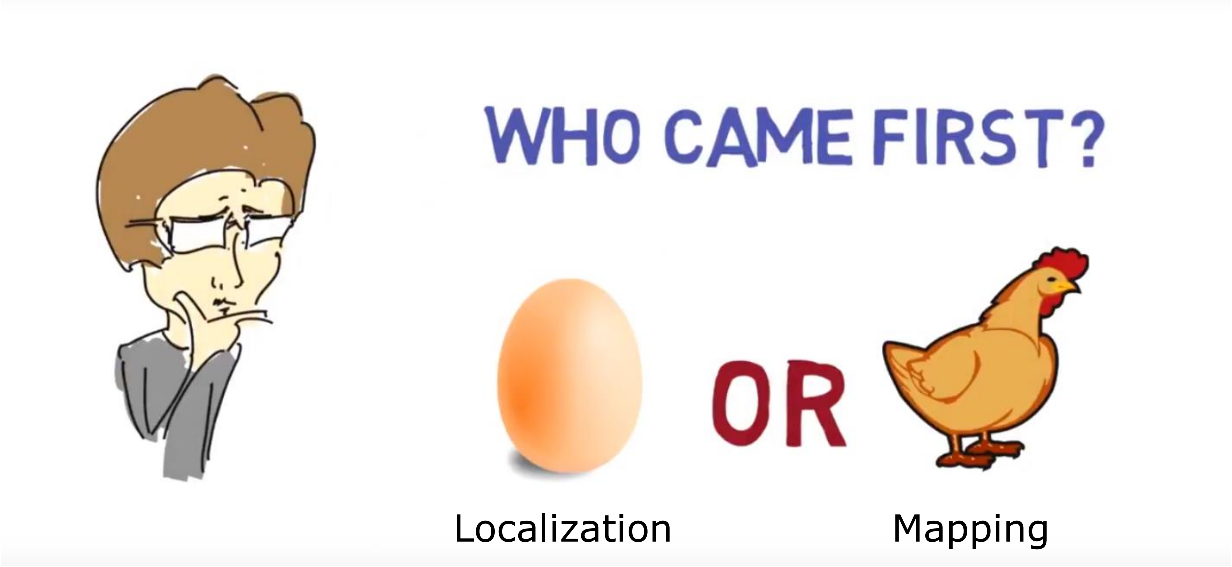
If I see a map,
I Know position.

We cannot see an entire map.
Without exploration,
We cannot get a map

- Localization vs. Mapping

Why SLAM is Problematic? Localization and Mapping occur coincidentally

- Localization requires Map
- Mapping requires position information
  - A mobile robot wants Localization and mapping at the same time



A robot must do What we did

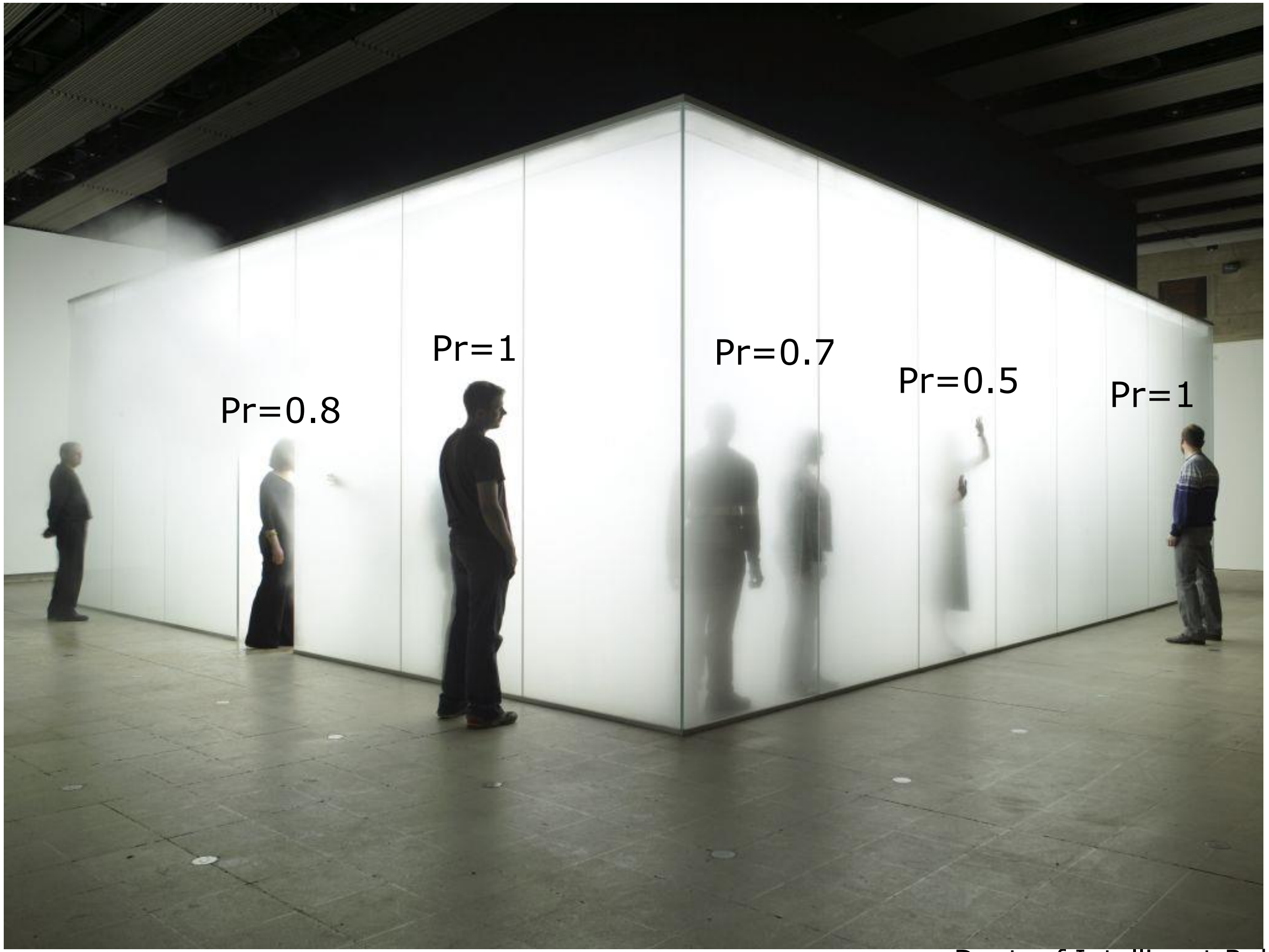
I am on (10,3),

MAYBE…TT

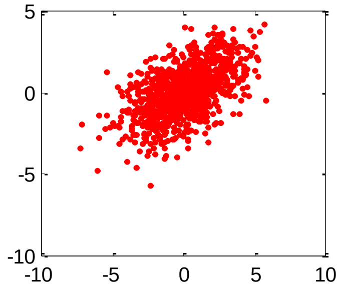
I know this road, **MAYBE…**

How to mix Two Unsure Information?

Everything in Probabilistic Robotics is NOT Sure(or Deterministic)



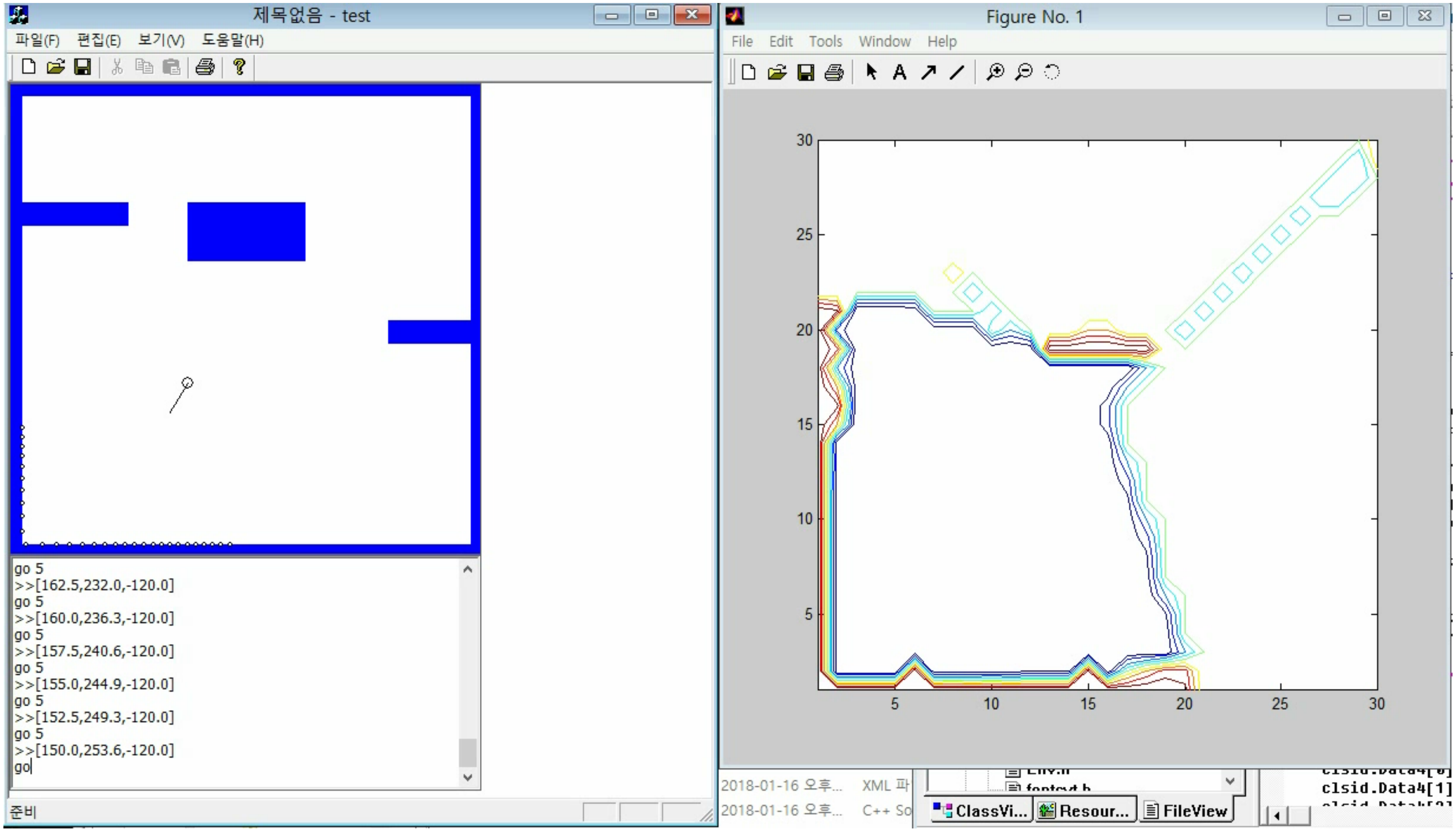
SLAM Example



- SLAM: State is NOT directly observed
  - (1) Every states are considered as Probabilistic Distribution.

Position is NOT a vector
Position is also a distribution

SLAM uses Mapping, which maps Partial information onto Final Results (2)



Wall is Not Deterministic, but a Probability

SLAM with Kalmann Filter or Particle Filter

$$x_{k+1} = F_k x_k + w_k$$
$$z_k = H_k x_k + v_k$$

Linear System

$$x_k = f_k(x_{k-1}, w_k)$$
$$z_k = h_k(x_k, v_k)$$

NonLinear System

- SLAM: Simultaneous Localization And Mapping
- Assumption:
  - Sensor information is Poor(inaccurate → but Probabilistic)

- Probabilistic Approach
  - We are familiar with accurate variable (x=3, y=2)
  - But in an actual world, x is not 3 in general.

Dept. of Intelligent Robot Eng. MU

Why Probabilistic Approach?

- Mapping (or Registration)
  - Each scene is Not Perfect. → Probabilistic distribution
  - Imperfect scene is Probabilistically merged
  - Repetitive Sampling improves accuracy
  - Sequential (Continuous) Observation(Scan) is NOT the sum of each one.
  - Sampling(Not all data) saves Computational burdens.
- Precondition:
  - Assume that Everything is Probabilistic.

2

Analogy:

Sample (or Particle) is Probability

Non Parametric Method

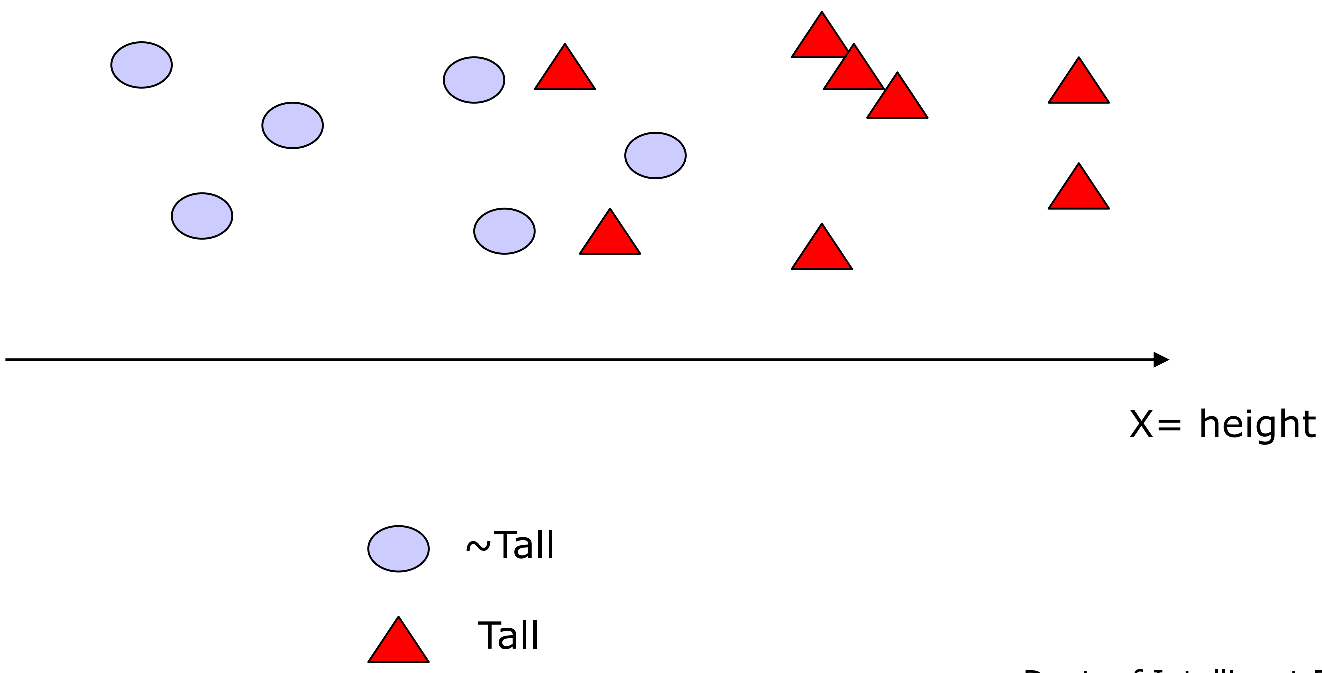
- Bayesian Classifier
  - Random data X has Probabilistic Distribution

$$x \sim N(\mu, \sigma^2) \rightarrow \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \quad (Gaussian)$$

  - Parametric estimation
- Non Parametric Method
  - No explicit distribution like Gaussian
  - Parameters: ex) Mean, sigma
  - Remind that Gaussian distribution requires ASSUMPTION.
  - Sample data generates some estimation function (Inferential type)
  - Non parametric method has amount of parameters like Neural network.

kth Nearest Neighbor

- Developed in 1960.
- Easy to understand it (Even very simple)
- Example(Tall and ~Tall)



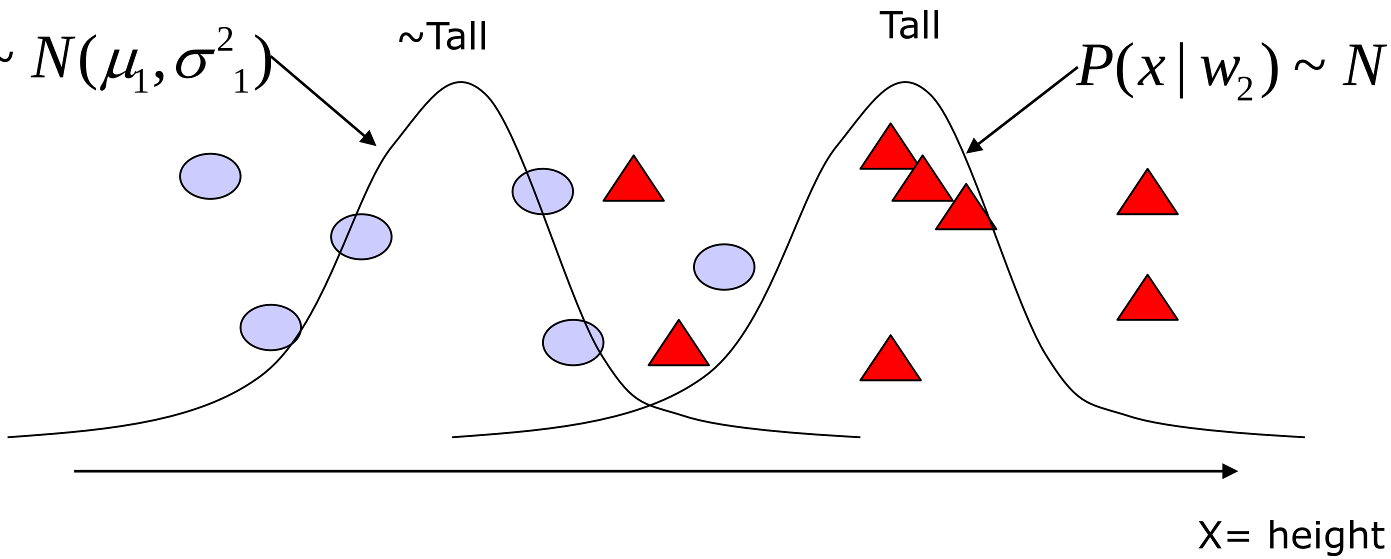
Dept. of Intelligent Robot Eng. MU

In case of Bayesian Classifier

- Find Gaussian distribution from sample data

$$P(x \mid w_1) \sim N(\mu_1, \sigma^2_1)$$

$$P(x \mid w_2) \sim N(\mu_2, \sigma^2_2)$$

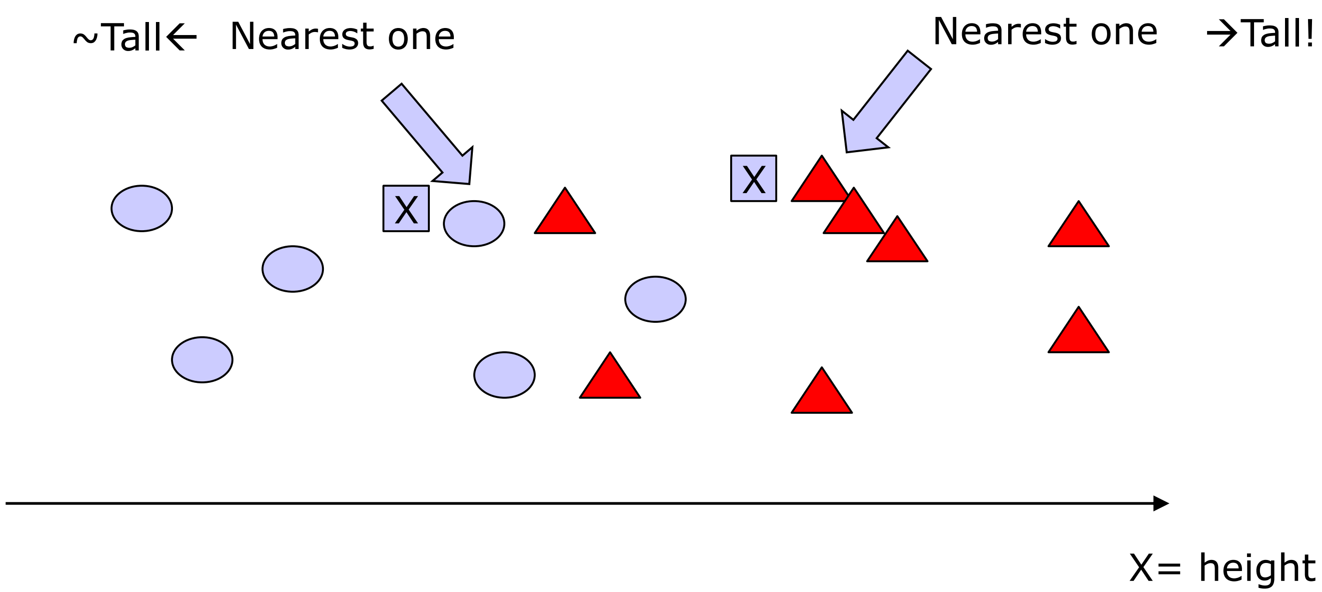


$$P(w_1 \mid x) = \frac{P(x \mid w_1)\mathrm{P}(\mathrm{w}_1)}{P(x)} \quad \begin{matrix} > \\ \text{Or} \\ < \end{matrix} \quad P(w_2 \mid x) = \frac{P(x \mid w_2)\mathrm{P}(\mathrm{w}_2)}{P(x)}$$

$$P(x) = \sum P(x \mid w_i)\mathrm{P}(\mathrm{w}_i)$$

Nearest Neighbor

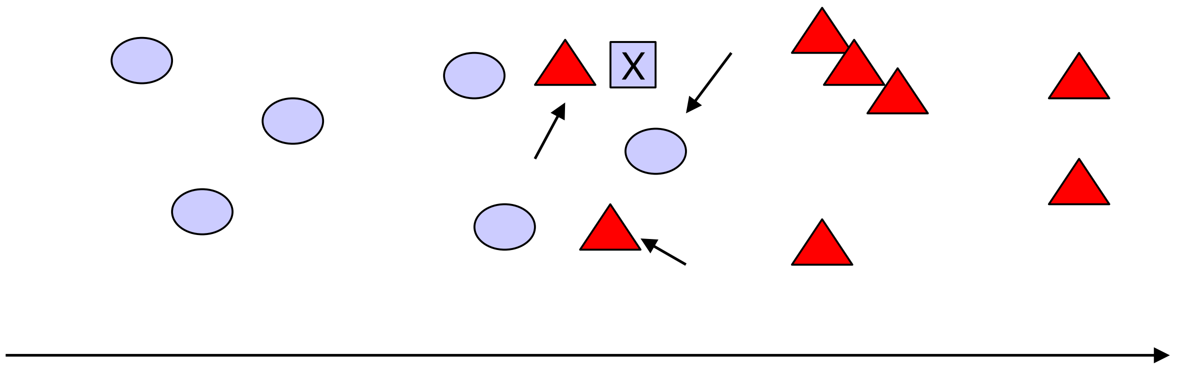
- Find the nearest sample to a given X



- If the nearest one is in a class w1, then x is w1.
- If the nearest one is in a class w2, then x is w2.
- Very simple..

Kth Nearest Neighbor

- "Kth" means that find the nearest neighbors with k number.
- More number class is the result

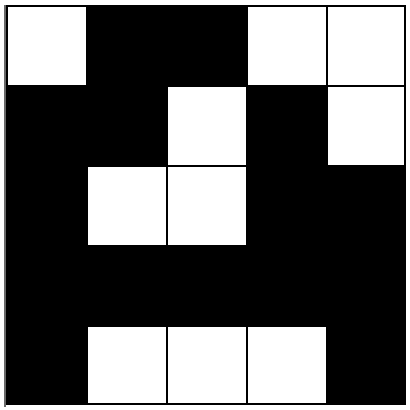


X= height

Ex) 3th Nearest neighbor
Tall cases(2) and ~Tall case (1)
2 > 1 .
therefore, it is tall.

Dept. of Intelligent Robot Eng. MU

Hand Writing Recognition



Example 5x5 writing

Instance = 25 dimension vector

X=[0,1,1,0,0, 1,1,0,1,0,
1,0,0,1,1, 1,1,1,1,1, 1,0,0,0,1]

$S \leftarrow s_i$
Distance= |x1-S1|+|x2-s2|+…|x25-s25|
Find the minimum distance.

- Example: test1
- See result.

- Oops.
- Recognition is so easy like this?
  - Generally, No.
- Why it is so good?

Features of KNN

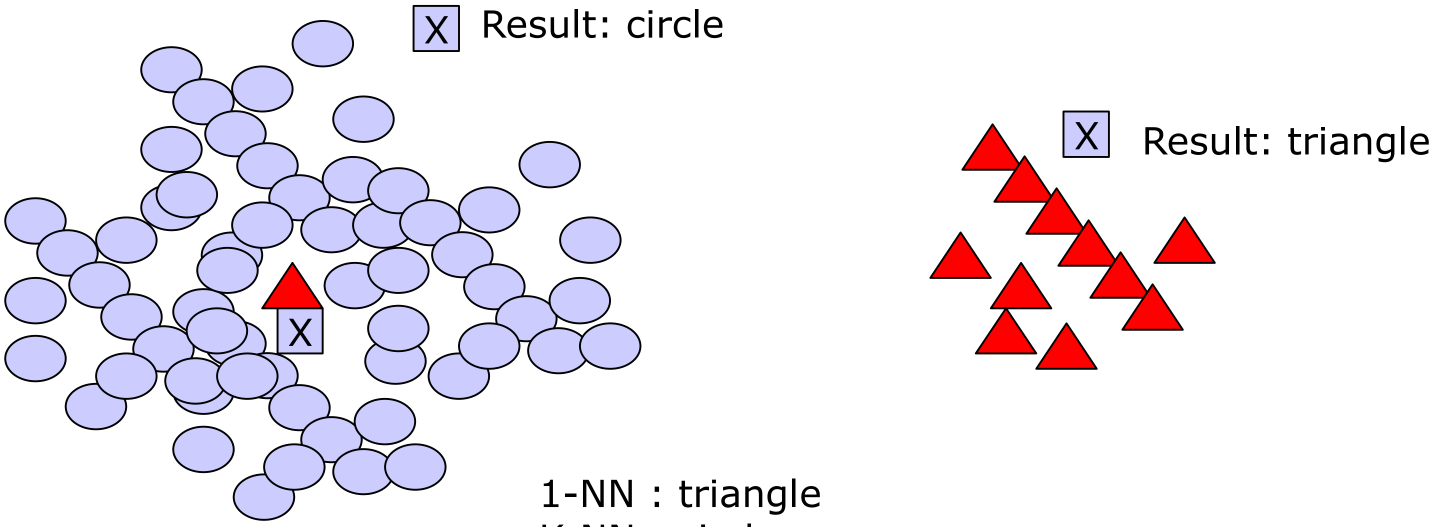
- Even Image is possible
  - Ex) Instance x is 640x480 dimension.

- Most learning method do learning after sampling.
- When kNN is learned?
  - When Sample is added, there is NO learning.

- However, kNN does not do learning procedure.
- Learning occurs, when we find nearest neighbors.
  - → Lazy learning.
- Problem
  - With more sample, comparison is painful process.

Why kNN rather than 1-NN ?

- 1-NN finds the nearest neighbor.
  - Very specific solution (Over fitting)
- K-NN finds the k nearest neighbors.
  - Less specific solution → Not Sensitive to Noisy sample.



1-NN : triangle
K-NN : circle

Distance of KNN

- When x is an instance vector,
- Generally, 2 norm is used for distance measurement
  - 1-Norm: absolute value , |x|
  - 2-Norm: vector distance

$$\| \mathrm{X} \| = \sqrt{x^2 + y^2 + z^2}$$

- Distance of NN
  - $S=\{s_1, s_2, s_3, \ldots s_N\}$
  - If the current X is added, $S=\{ s_1, s_2, s_3, \ldots s_N, s_{N+1}\}$ where $s_{N+1} \leftarrow X$.

Distance = $\| \mathrm{X} - s_i \| = \| \mathrm{e}_i \| = \sqrt{e_1^2 + e_2^2 + e_3^2 + \ldots + e_{Dim}^2}$

Discrete Problems for KNN

- Continuous data
- X=(0.01, 0.1, 0.4)
- Si=(1.2, 0.4, 0.2)

- Distance = 1.2434

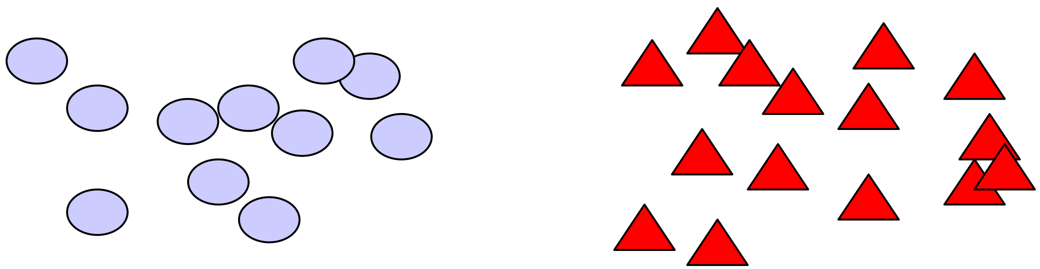
- Hand writing example

- X=(0,1,0,1,0,0)
- Si=(1,0,0,0,0,0)
- Sj=(0,1,0,0,1,1)
- Distance
- ||X-Si|| = sqrt(3)
- ||X-Sj|| = sqrt(3)

- Discrimination is not so accurate!

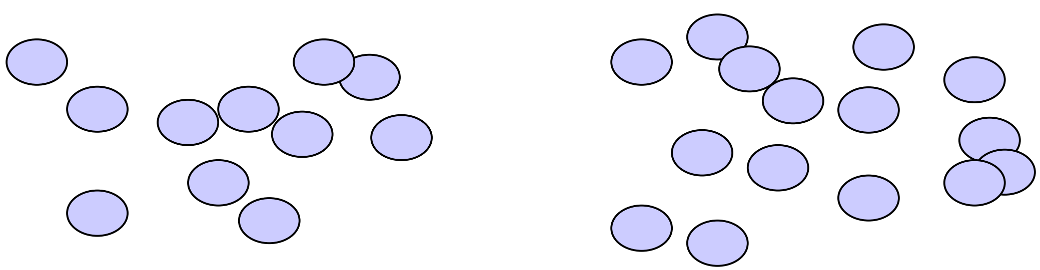
Clustering Method
Important Tools for Intelligent Robotics

- Pattern recognition requires Class definition



2 classes

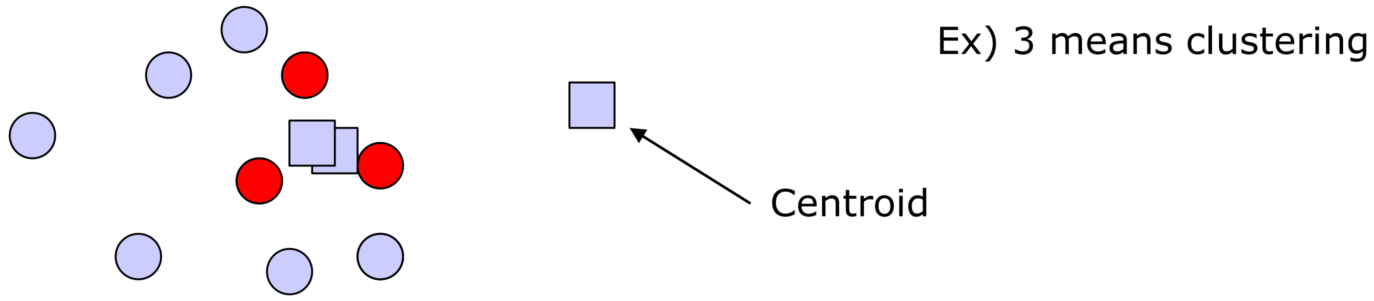
- How many classes here?



- There are only two lumps → Two clusters.

Clustering Method find how many clusters are there

- Many clustering methods.
- Example) K Means Clustering Method.
- 1. Assume there are K clusters.
- 2. Guess each centroid of cluster.
- 3. Find k points to closest centroid
- 4. Recompute the centroid of each cluster.

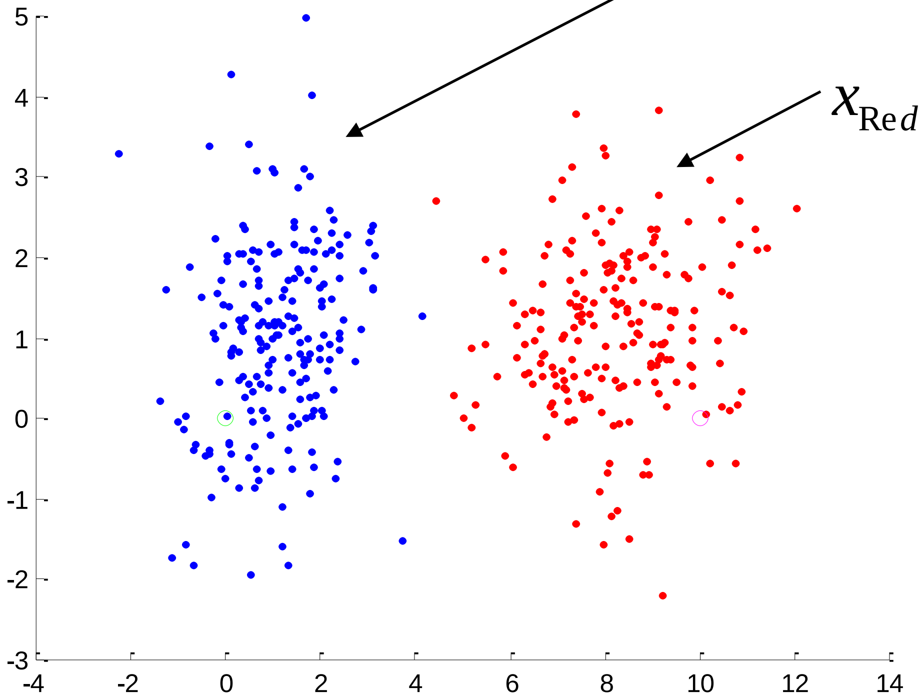


Example) K means Clustering

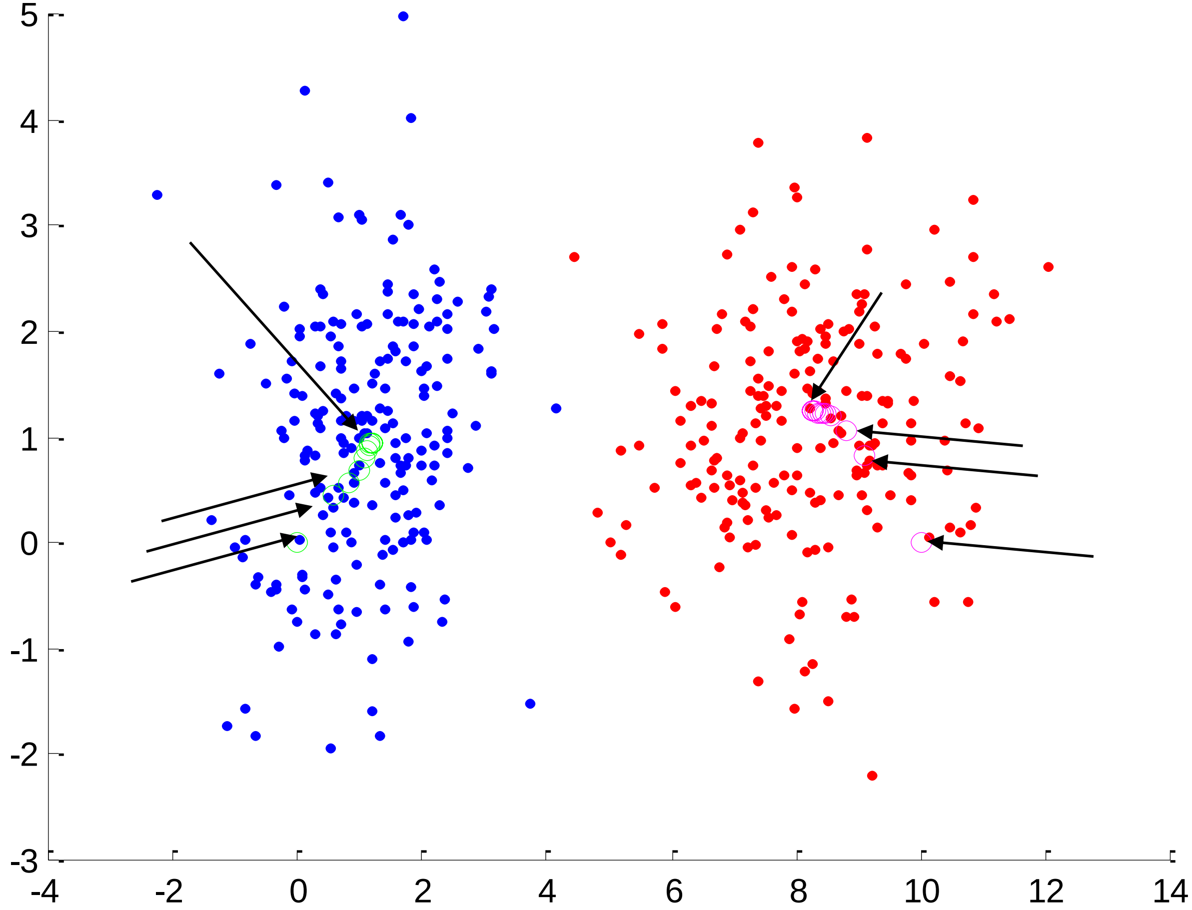
- Test 2.m

$$x_{Blue} \sim N(\mu,\sigma) = N\left(\begin{bmatrix}1\\1\end{bmatrix},\begin{bmatrix}1 & 0.1\\0.1 & 1.5\end{bmatrix}\right)$$

$$x_{Red} \sim N(\mu,\sigma) = N\left(\begin{bmatrix}8\\1\end{bmatrix},\begin{bmatrix}2 & 0.1\\0.1 & 1\end{bmatrix}\right)$$



Example) K means Clustering
Centroid comes close to mean value



C1=(1.1796, 0.9455)

C2=(8.2737, 1.2528)

Centroid of Cluster
What is it?

- In k means cluster,
  - centroid approaches mean value of the test distribution.
  - But it is not on mean value.
  - Why?

- Think the role of K mean cluster.
  - K closest points are Not whole data. Just Sample.

    → In each turn, K mean clustering method find the centroid of K closest points.
  - If Initial centroid is biased, centroid is sometimes biased.
- If we guess wrong number of centroid, how it works?

3

Why we need Observer?

Probabilistic Approach toward Kalman Filter

You Measure Everything?

- Your graduation is on Prof. Y's decision

Graduation
Is OK?

...

It is a Black box

When X>?,
Graduation
is OK.

Fail, T_T…

Graduation!!

We cannot read his mind.

But we can estimate "X>?"

So, We guess X from Observation Z

Your Estimation
Your Measurement
Your Guess

Z: observation

We only know Z

His standard
His mind
His viewpoint

X: Actual State

We don’t know X

I think

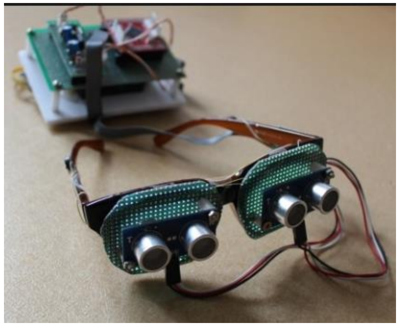
Can you Read his Mind?

Z= 0.9
Excellent

Fail

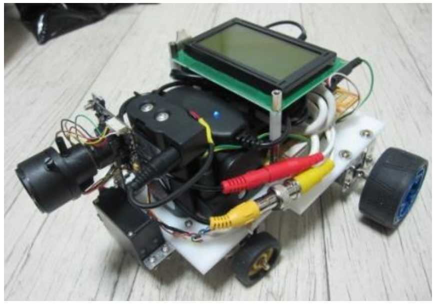
Fail

Z= 0.6
Not bad



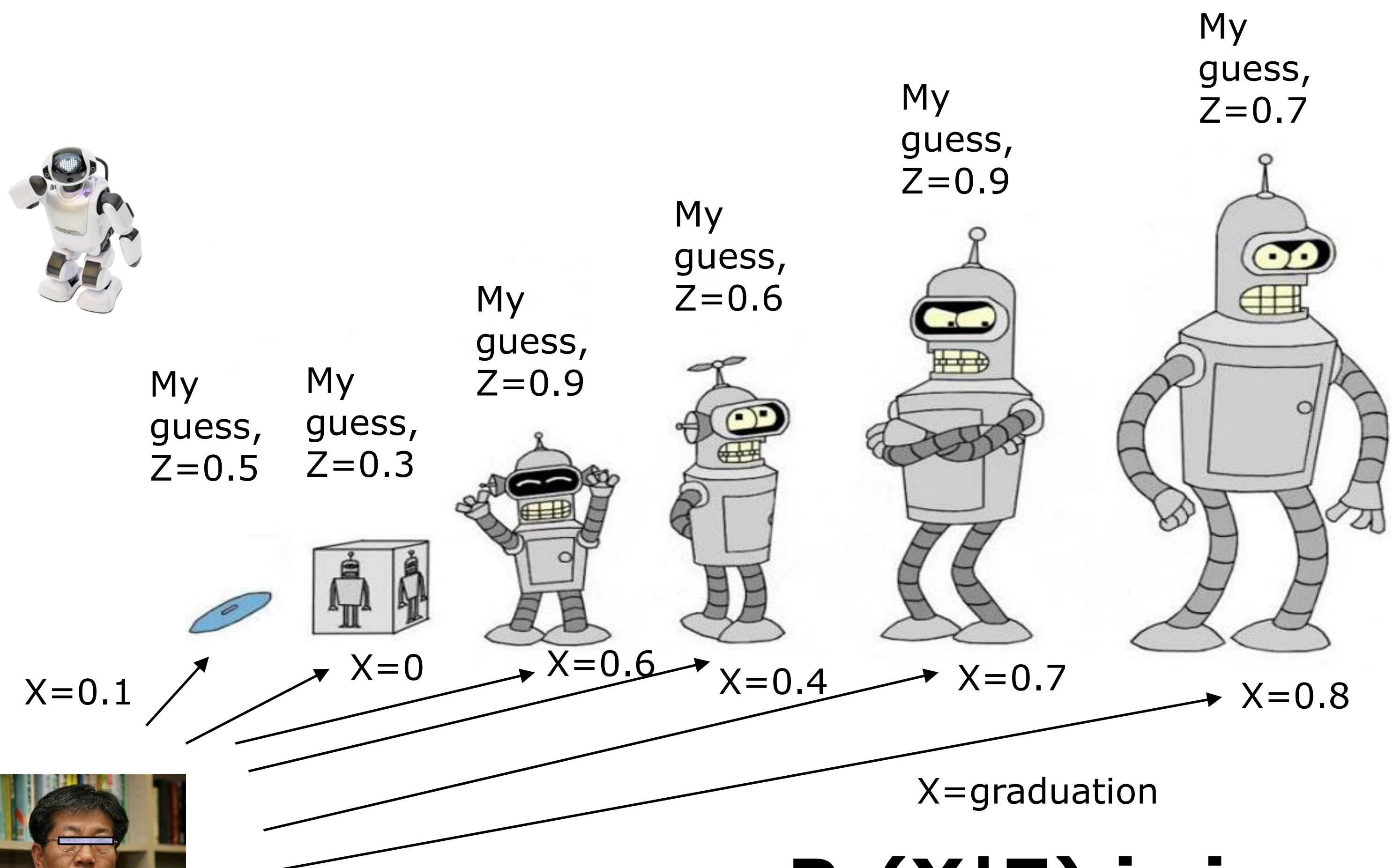
Graduation

Z= 0.8
good



Pr(X|Z) is about 1/3

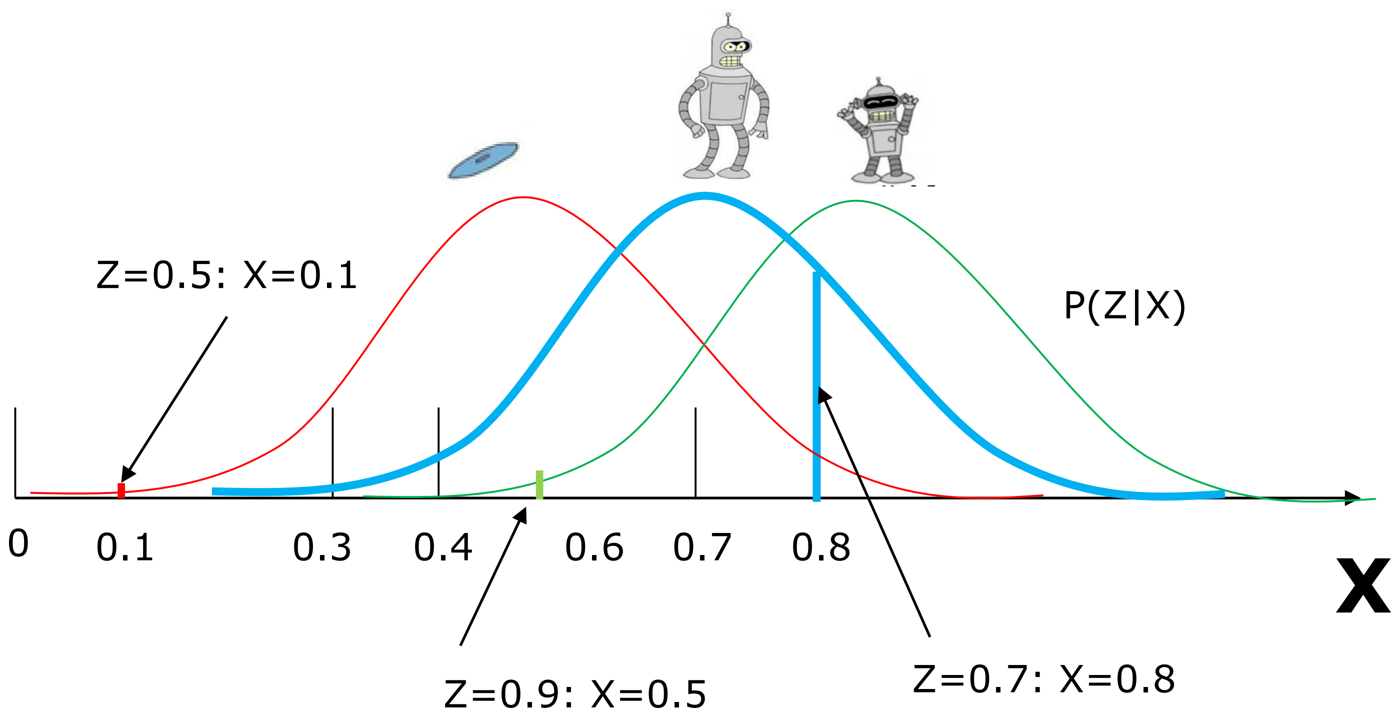
How we Improve P(x|z) ?
The best way is Repetitive Confirmation



Pr(X|Z) is improved

You Focused on
Why P(X|Z) is improved?

We Do Not know X but, P(X|Z) becomes increase to 1

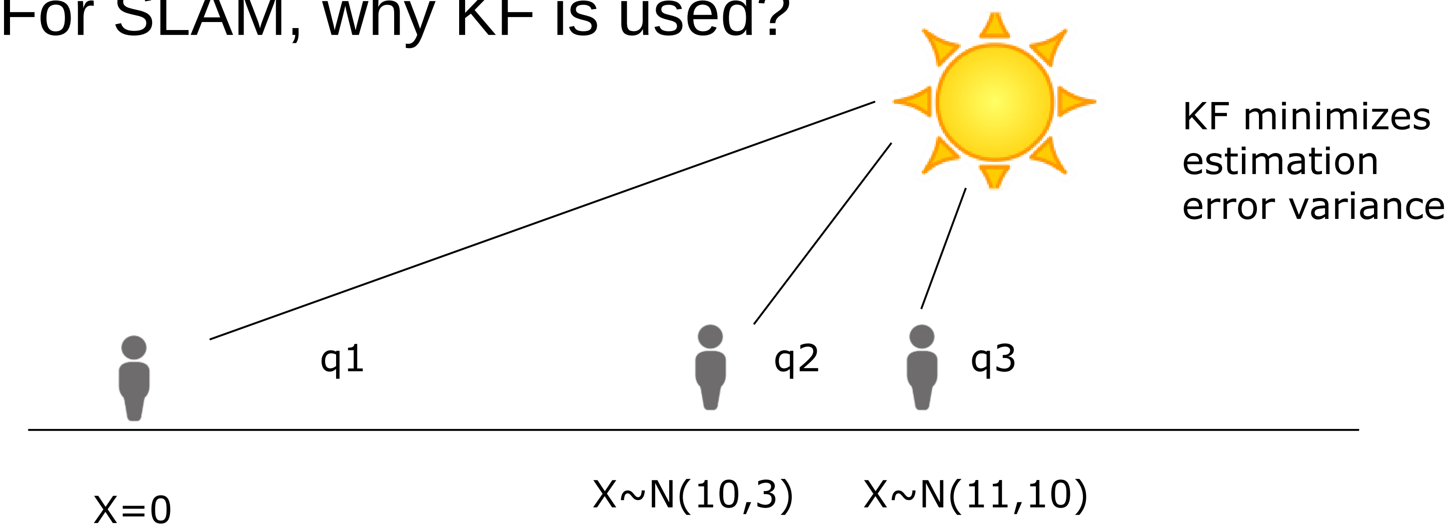


4 Basics of Control for Kalman Filter

Kalman filter

- Kalman Filter(KF)
  - Estimates current state with observed state.
  - Estimation error is minimized by using Gaussian concept
  - Prediction + Update process.

- For SLAM, why KF is used?



KF model

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$

$$z_k = H_k x_k + v_k$$

Test3.m

Noisy data(blue) is filtered out as Green data

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^{\ T} + Q_{k-1}$$

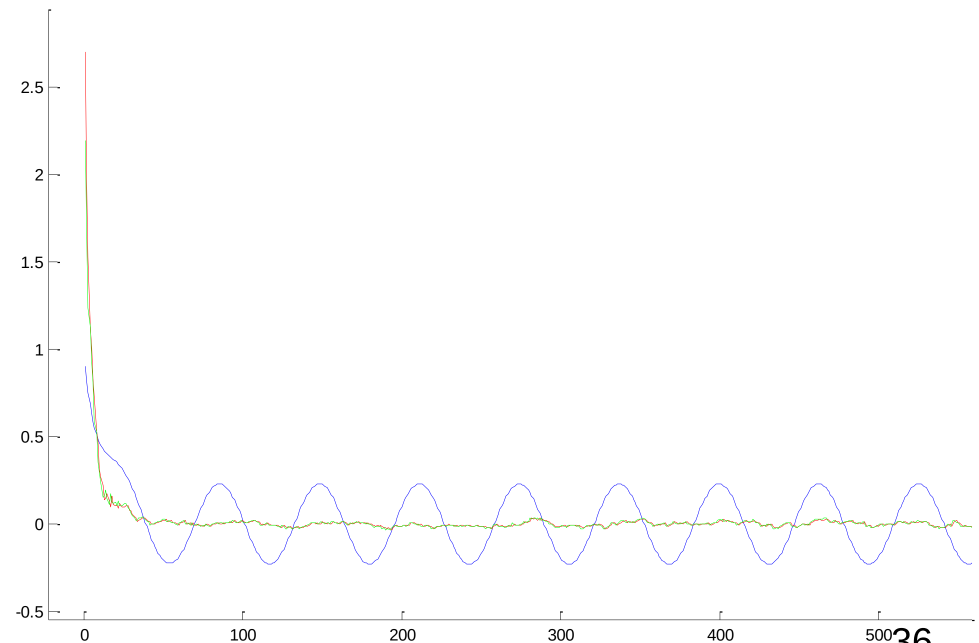
$$\hat{y}_k = z_k - H_k \hat{x}_{k|k-1}$$

$$S_k = H_k P_{k|k-1} H_k^{\ T} + R_k$$

$$K_k = P_{k|k-1} H_k^{\ T} S_k^{\ -1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \hat{y}_k$$

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



Pre Knowledge for KF.

- State space expression
- 2nd order mass-spring-damper system.

$$m\ddot{x}+c\dot{x}+kx=F(t)$$

- Second order differential equation has two solutions

$$m\ddot{x}_h+c\dot{x}_h+kx_h=0$$

Homogeneous Solution

$$m\ddot{x}_p+c\dot{x}_p+kx_p=F(t)$$

Particular solution

Easier than 1st order,

$$ay'' + by' + cy = 0$$

$$Define\ Dy = \frac{dy}{dx}$$

$$\rightarrow aD^2 y + bDy + cy = 0$$

$$\rightarrow (aD^2 + bD + c)y = 0$$

$$D = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- D operator for simplifying 2nd order Differential Eq.

$$ex)\ y''+5y'+4y=0$$

$$D^2y+5Dy+4y=0$$

$$(D^2+5D+4)y=0$$

$$(D+1)(D+4)y=0$$

$$(D+1)y=0 \quad or$$

$$(D+4)y=0$$

Remind 1st order Equation

$$(D+1)y=0 \quad or \quad (D+4)y=0$$

$$y'+y=0 \quad or \quad y'+4y=0$$

$$y=C_1e^{-x} \quad or \quad y=C_2e^{-4x}$$

$$\therefore y=C_1e^{-x}+C_2e^{-4x}$$

Mass-Spring System

$$m\,y'' + ky = 0$$

$$mD^2 y + ky = 0$$

$$(D^2 + \frac{k}{m})y = 0$$

$$\left(D - \sqrt{\frac{k}{m}}i\right)\left(D + \sqrt{\frac{k}{m}}i\right)y = 0$$

Remind

$$(D - z_1)(D - z_2)y = 0$$

$$\Rightarrow y = C_1 e^{z_1 x} + C_2 e^{z_2 x}$$

$$y = c_1 e^{\sqrt{\frac{k}{m}}ix} + c_2 e^{-\sqrt{\frac{k}{m}}ix}$$

$$= c_1 e^{wix} + c_2 e^{-wix}$$

$$= c_1(\cos(wx) + i\sin(wx))$$

$$+ c_2(\cos(wx) - i\sin(wx))$$

$$= A\cos(wx) + B\sin(wx)$$

$$= C\sin(wx + \varphi)$$

Hyperbolic function

Particular Solution of 2nd order Diff. Eq.

1

$$ex)\, y''+5y'+4y=\cos 2x$$

$$(D^2+5D+4)\,\mathrm{y}=\cos 2\mathrm{x}$$

$$(D^2+5D+4)\,\mathrm{y}_h=0$$

$$\mathrm{y}_h=c_1e^{-x}+c_2e^{-4x}$$

2

$$(D^2+5D+4)\,\mathrm{y}_p=\cos 2\mathrm{x}$$

$$if\ \ \mathrm{y}_p=\alpha\sin 2x$$

$$y''=-4\alpha\sin 2x,\quad y'=2\alpha\cos 2x$$

$$\Rightarrow$$

$$-4\alpha\sin 2x+$$

$$5*2\alpha\cos 2x+$$

$$4*\alpha\sin 2x=\cos 2x$$

$$10\alpha\cos 2x=\cos 2x$$

$$\therefore \alpha=1/10$$

3

$$y=y_p+y_h$$

$$=\frac{1}{10}\sin 2x+c_1e^{-x}+c_2e^{-4x}$$

Particular Solution is the Controller

- U : controller

$$m\ddot{y} + c\dot{y} + ky = F(t) = \mathrm{u(t)}$$

- U=0 → Homogenous solution → System dynamics

$$m\ddot{y} + c\dot{y} + ky = 0$$

- If we define $y = x_1, \dot{\mathrm{y}} = x_2$, every dynamics is expressed as state variable x.
- → State space

State Space Notation

$$m\ddot{y} + c\dot{y} + ky = 0$$

$$x_1 = y$$

$$x_2 = \dot{y}$$

$$\Rightarrow \quad m\dot{x}_2 + cx_2 + kx_1 = 0 \quad \Rightarrow \quad \dot{x}_2 + \frac{c}{m}x_2 + \frac{k}{m}x_1 = 0$$

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\dfrac{k}{m} & -\dfrac{c}{m} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\Rightarrow \quad \dot{x} = Ax$$

State Space Notation with Control Input

$$m\ddot{y} + c\dot{y} + ky = u$$
$$x_1 = y$$
$$x_2 = \dot{y}$$

$$\Rightarrow \quad m\dot{x}_2 + cx_2 + kx_1 = u \quad \Rightarrow \quad \dot{x}_2 + \frac{c}{m}x_2 + \frac{k}{m}x_1 = u$$

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

$$\Rightarrow \quad \dot{x} = Ax + Bu$$

Laplace Transform and Eigenvalue of A

$$m\ddot{y} + c\dot{y} + ky = 0$$

$$(mD^2 + cD + k)\,y = 0$$

$$(D - w_d i)(D + w_d i)\,y = 0$$

$$w_d = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m} = -\frac{c}{2m} \pm w_d i$$

$$y = e^{-\frac{c}{2m}t}(c_1 e^{w_d ix} + c_2 e^{-w_d ix})$$

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{pmatrix} x$$

$$Av = \lambda v$$

$$(A - \lambda I)v = 0$$

$$Det(A - \lambda I) = 0$$

$$Det\begin{pmatrix} -\lambda & 1 \\ -\frac{k}{m} & -\frac{c}{m} - \lambda \end{pmatrix} = \lambda\left(\frac{c}{m} + \lambda\right) + \frac{k}{m} = 0$$

$$\lambda^2 + \frac{c}{m}\lambda + \frac{k}{m} = 0$$

State Space Model

- Exactly, Linear Model is expressed as,

$$\dot{x} = Ax + Bu$$

Eig(A) are root(Poles).

- Non linear system dynamics

$$\dot{x} = f(x) + u$$

- Generally, Non Linear system dynamics

$$\dot{x} = f(x, u)$$

Feedback Observation

- Assume that state variable x can be observed.
- But it is sometimes impossible or corrupted with noise

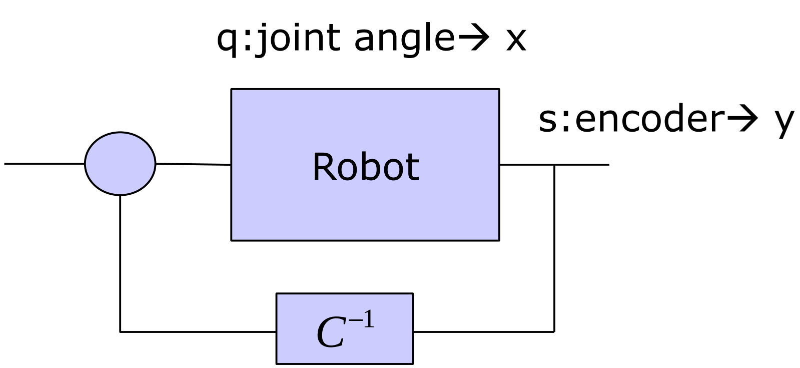
$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Special case

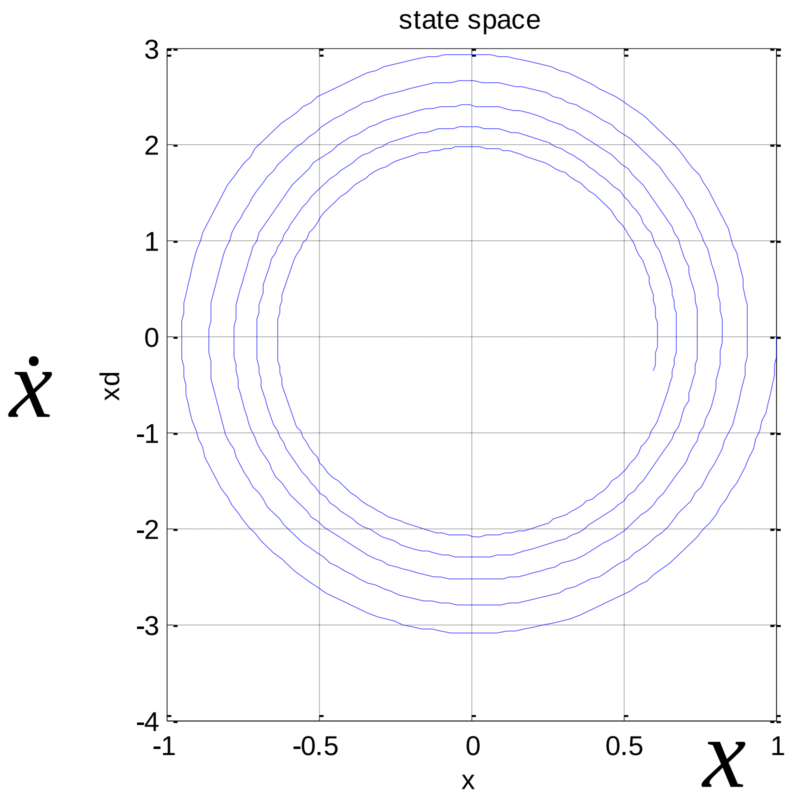
$$C = 1, D = 0$$

$$y = x$$

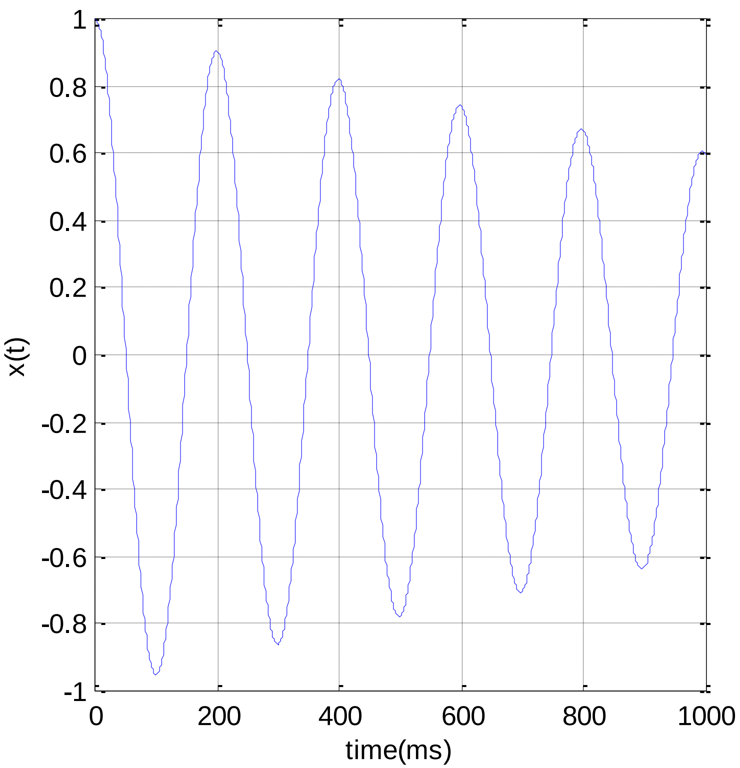


2nd order Mass-Spring-Damper

- Example) exms.m



State space



Time domain

Our Model is Perfect?

No, it has uncertainties and Model is Imperfect

2nd order mass-spring-damper system

$$\dot{X} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{pmatrix} X + Bu +$$

$$= AX + Bu$$

$$Y = CX + Du$$

$$= [1 \quad 0]\begin{bmatrix} x \\ \dot{x} \end{bmatrix} + 0u$$

$$= x$$

$$\dot{X} = AX + Bu + N$$

N : Process Noise

$$Y = CX + Du + N'$$

N' : Measurement Noise

Process Noise

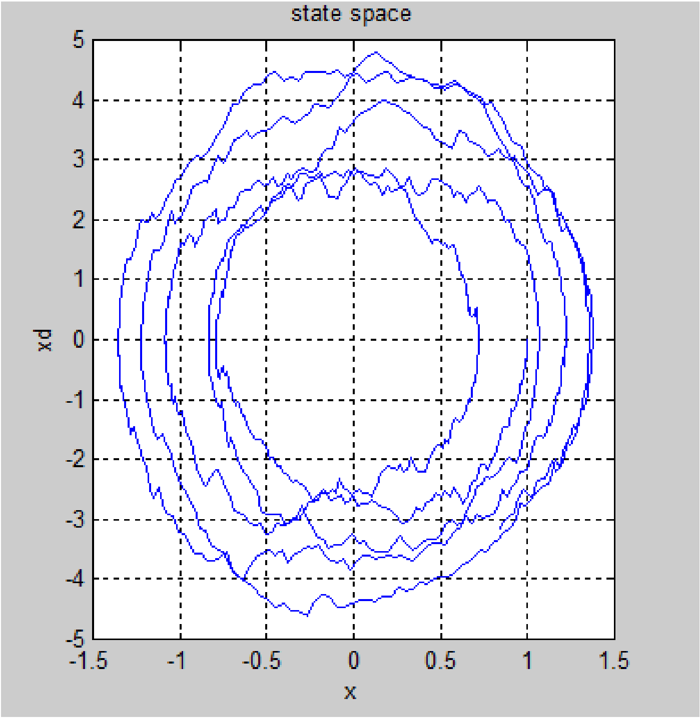
$$\dot{X} = AX + Bu + N$$

$$N : \text{Process Noise}$$

$$\dot{X} = \begin{pmatrix} 0 & 1 \\ -\dfrac{k}{m} & -\dfrac{c}{m} \end{pmatrix} X + Bu + \begin{pmatrix} N_v \\ N_a \end{pmatrix}$$

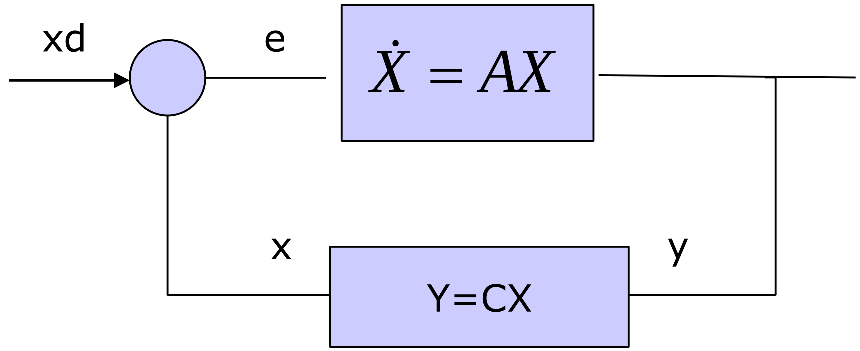
Ex) exmsprocess.m

```
% m*xdd + c*xd + kx = F = 0;
for i=1:1000
    s=[s; x xd];

    Na = 0.1*randn;
    Nv = 0.1*randn;
    N=[Na,Nv]';


    xdd = (0-k*x-c*xd)/m+Na;
    xd = xdd*dt+xd+Nv;
    x = xd*dt+x;
end
```



Measurement Noise

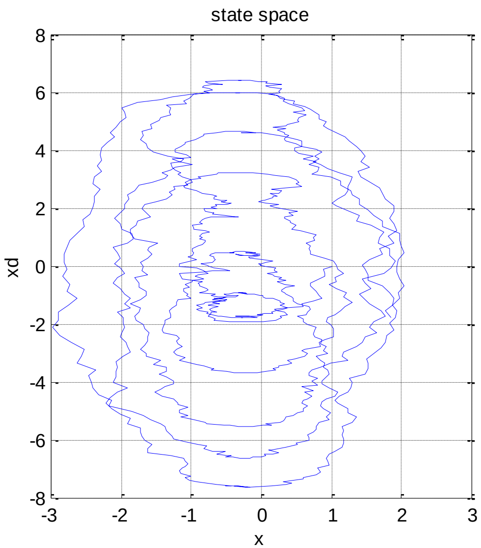
$$Y = CX + Du + N'$$
$$= [1 \quad 0]\begin{bmatrix} x \\ \dot{x} \end{bmatrix} + 0u + N'$$
$$= x + N$$

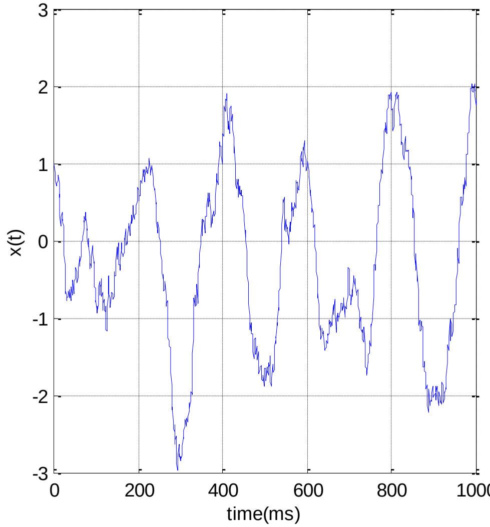


Remind that
All measurements x for 'e=xd-x' are actually Y.

Ex) exmsm.m

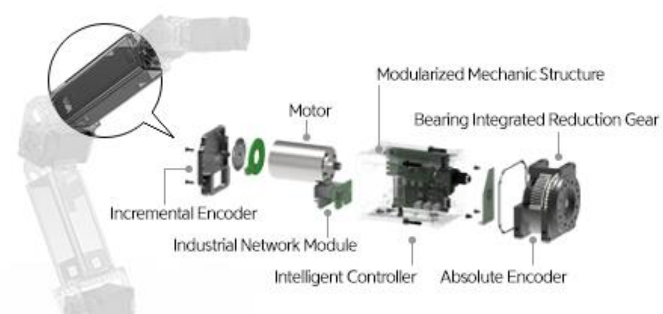
```
% m*xdd + c*xd + kx =
for i=1:1000
   s=[s; x xd];

   y=x+0.1*randn

   x=y;

   xdd = (0-k*x-c*xd).
   xd = xdd*dt+xd+Nv;
   x = xd*dt+x;
end
```



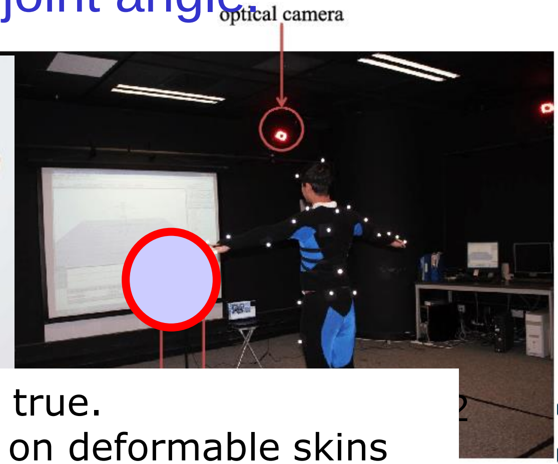


Process and Measurement Noises are UNAVOIDABLE Problems

- In spite of all, why we did not add noise model?
  - We neglect noises in many cases
  - Noise model is somewhat complex and unpredictable
  - Also, you are undergraduate student..
- Intuitive Example
  - Encoder signal is an actual value?
  - Encoder is perfect but Joint angle is NOT perfect
  - Assumption of that encoder is same with joint angle.



Encoder is perfect



Marker signal is not true.
Why? It is attached on deformable skins

5 Kalman Filter

Expectation in Probability

- **E{x}: Expectation, What a value occurs with Prob.**

$$\mathrm{E}\{\mathrm{x}\} = \int x p(x)\,\mathrm{dx} \quad \text{or} \quad = \sum_k x_k p_k$$

- Remind Probabilistic density function
  - Ex) Gaussian

$$\mathrm{PDF}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \right)$$

- Expectation of “1” of Dice throwing

$$\mathrm{E}\{\mathrm{x}\} = \sum_k x_k p_k = 1\frac{1}{6} + 2\frac{1}{6} + 3\frac{1}{6} + 4\frac{1}{6} + 5\frac{1}{6} + 6\frac{1}{6}$$

$$= \frac{\sum_k x_k}{6} = \mu$$

- Expectation converges into Mean value.

Goal of Kalman Filter

3. Update P' by P

$$e' = \hat{X}' - \hat{X}$$

$$P' = E\{e' \bullet e'\}$$

4.minimize

Estimation Error

$\rightarrow$ *Kalman Gain*

$\min P = E\{e \bullet e\}$

$\rightarrow K$

2. $\hat{X}'_{k+1}$:

Prediction by

model,

$A\hat{X}_k + BU_k$

$$\hat{X}'$$

$$\dot{X} = AX + BU$$

Continuous form

$$X_{k+1} = A' X_k + B' U_{k+1}$$

Discrete form

5.*measurement*, z

$\rightarrow$ *Correction of*

$\hat{X}, P$ *with* $\hat{X}', K, z, P'$

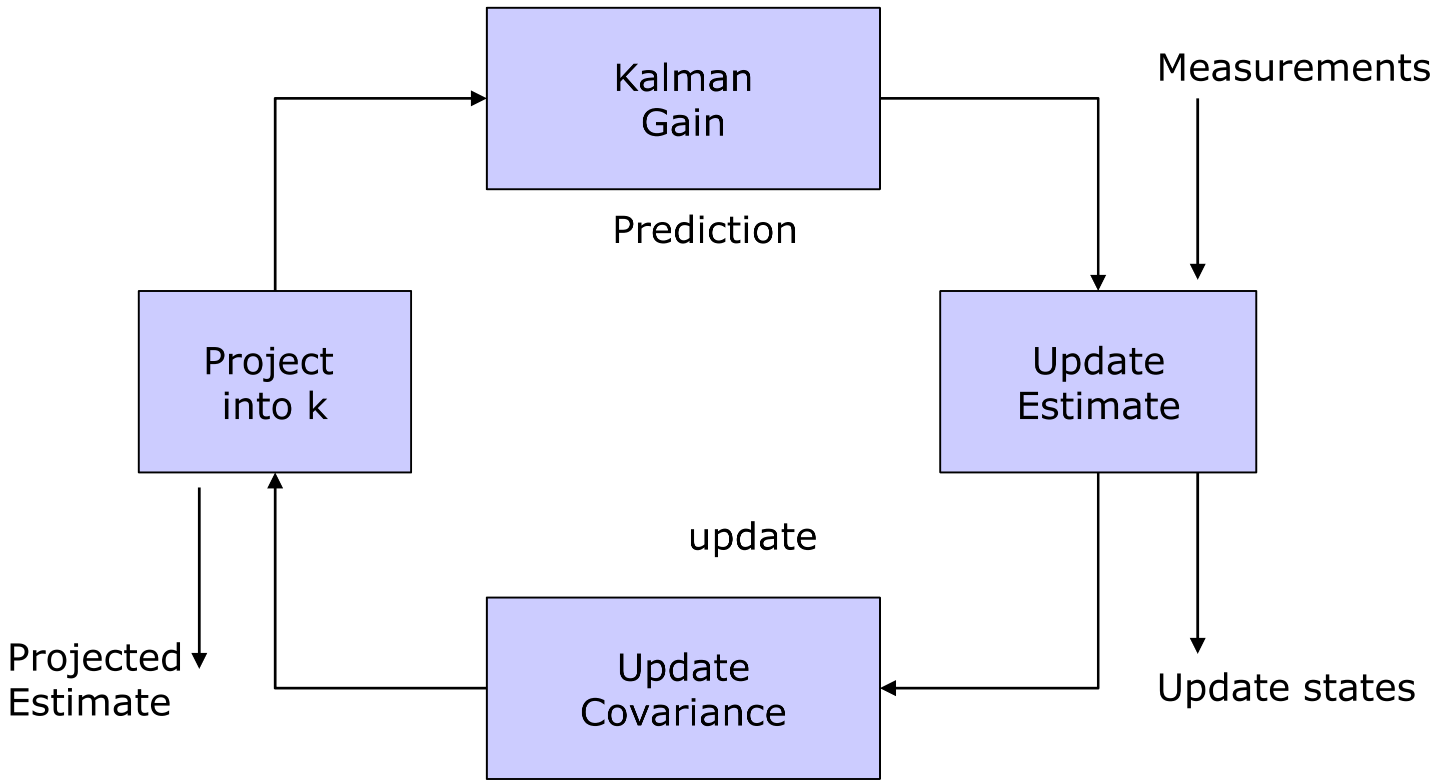
$$\hat{X}$$

1. Estimation of X

X is an actual value,
but we don't know it.

Kalman Filter

Initial Estimates $\hat{x}_{k-1|k-1} = \hat{x}$



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

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$
$$z_k = H_k x_k + v_k$$

$$\dot{x} = Ax + Bu$$

W, V= noise

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^{T} + Q_{k-1}$$

$$S_k = H_k P_{k|k-1} H_k^{T} + R_k$$
$$K_k = P_{k|k-1} H_k^{T} S_k^{-1}$$

$$\hat{y}_k = z_k - H_k \hat{x}_{k|k-1}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \hat{y}_k$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

Kalman Filter Definitions

State Equation

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$

$$z_k = H_k x_k + v_k$$

Prediction

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^{\,T} + Q_{k-1}$$

Kalman Gain

$$S_k = H_k P_{k|k-1} H_k^{\,T} + R_k$$

$$K_k = P_{k|k-1} H_k^{\,T} S_k^{\,-1}$$

Correction

$$\hat{y}_k = z_k - H_k \hat{x}_{k|k-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \hat{y}_k$$

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

state vector x cannot be directly measured

x : *state vector*

$\hat{x}$: *state vector estimate*

z : *observation vector*

u : *control* vector

F : *state* transition

B:control

P:covariance of state vector estimate

Q:process noise covariance

R:measurement noise covariance

H: observation matrix

$$w \sim N(0, Q)$$

$$x \sim N(\hat{x}, P)$$

$$v \sim N(0, R)$$

$\hat{x}_{k|k-1}$: Prediction (xp)

$\hat{x}_{k-1|k-1}$: Estimate (xe)

Derivation of K.F.

$$x_{k+1} = F_k x_k + w_k$$
$$z_k = H_k x_k + v_k$$

Simply, we think time invariant system

$$x_{k+1} = F x_k + w$$
$$z_k = H x_k + v$$

$x : actual\ value$ → **Unknowns**

$\hat{x} : estimated\ value$ → **Estimation from Model**
But, How we update it?
Update $\hat{x}$ with y

$Our\ Goal : \hat{x} \rightarrow x$

Before update (Prediction)

$$\mathrm{A}' = A_{k|k-1}$$

After Update

$$\mathrm{A} = A_{k|k}$$

1. Estimation of X

$$\dot{X} = AX + BU$$

Continuous form

$$X_{k+1} = AX_k + BU_{k+1}$$

Discrete form

$$\dot{X} = \frac{X_{k+1} - X_k}{\Delta t} = AX_k + BU_{k+1}$$

$$\therefore X_{k+1} = (1 + \Delta t A) X_k + \Delta t B U_{k+1}$$

$$= A_{k+1} X_k + B_{k+1} U_{k+1}$$

- We want to know an actual value, X
- X is not measured directly
- Thus, instead of X, we use estimation, $\hat{X}$

- But, remind that there is Process Error

1. Estimation of X with $\hat{X}$ and covariance, P

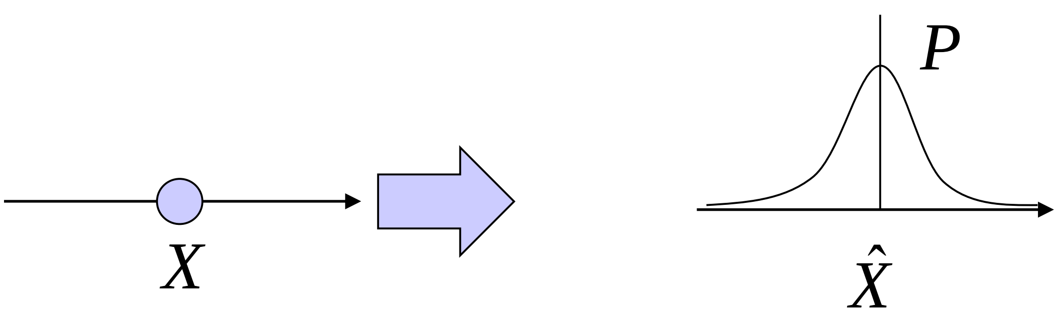
- Perfect Model

$$X_{k+1} = AX_k + BU_{k+1}$$

- Process error

$$X_{k+1} = AX_k + BU_{k+1} + w_{k+1}$$

- Actual value X is divided into two factor



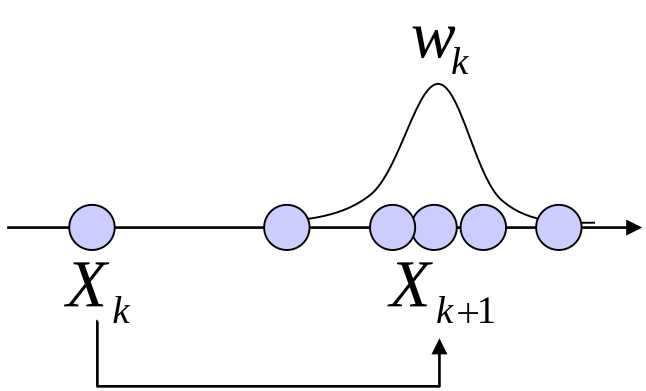
2. Prediction of $\hat{X}$ by system model

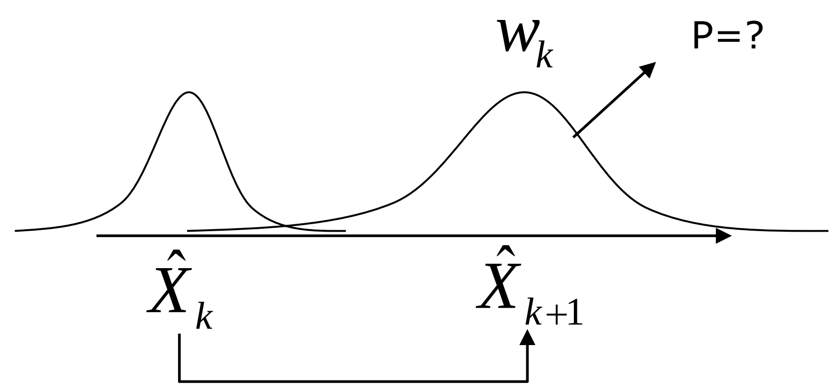
- Estimation $\hat{X}$ changes under system dynamics

$$X_{k+1} = AX_k + BU_{k+1} + w_{k+1}$$

$$\hat{X}_{k+1} = A\hat{X}_k + BU_{k+1} ??$$

Actual value, X



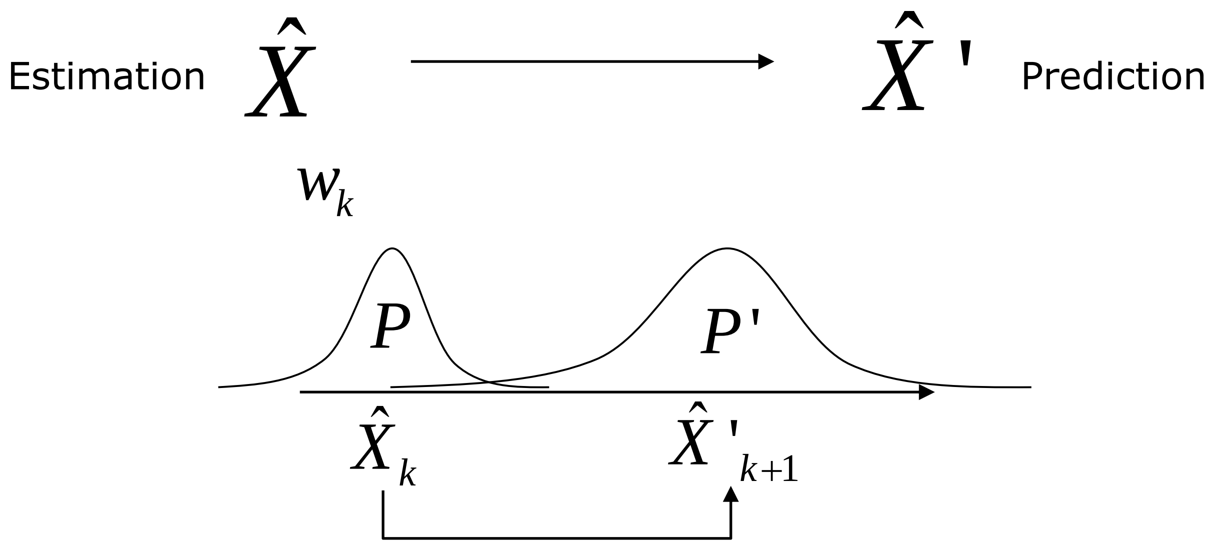


$$X_{k+1} = AX_k + BU_{k+1} + w_{k+1}$$

$$\hat{X}_{k+1} = A\hat{X}_k + BU_{k+1}$$

2. Prediction of $\hat{X}$ by system model

- Estimation $\hat{X}_k$ is given, but $\hat{X}_{k+1}$ is Not clear.
- Thus, we use a new concept, Prediction.
  - Prediction is temporarily used by system dynamics
  - Prediction will be updated by measurement



$$\hat{X}'_{k+1} = A\hat{X}_k + BU_k$$

Dept. of Intelligent Robot Eng. MU

3. Prediction of Covariance, P

- Definition of Covariance, P

$$P = E\{e^2\}, \quad e = \hat{x} - x$$

P means how much estimation is biased from an actual value

- Prediction of Covariance, P'

$$P' = ? \qquad e' = \hat{x}' - x$$

$$P' = E\{e_{k+1}{}' e_{k+1}{}'\}$$

$$= E\{(\hat{x}'_{k+1} - x_{k+1})^2\}$$

$$= E\{(\hat{x}' - Fx - w)^2\}$$

$$E\{w^2\} = Q$$

$$= E\{(F\hat{x} - Fx - w)^2\}$$

$$= E\{(Fe - w)^2\} = F^2 E\{e^2\} + E\{w^2\}$$

$$= F^2 P + Q + 0$$

4. Definition of Kalman Gain Measurement Updates Estimation

Actual System
But we don't know it

$$x_{k+1} = Fx_k + w$$

$$z_k = Hx_k + v$$

w, v cannot be directly measured,

But, F and H can be modeled.

Estimation $x \rightarrow \hat{x}$

Prediction

$$\hat{x}'_{k+1} = F\hat{x}_k$$

$$\hat{z}'_{k+1} = H\hat{x}'_{k+1}$$

$$Update:$$

$$z_{k+1} - \hat{z}'_{k+1} \rightarrow K\ gain \rightarrow \hat{x}_{k+1} - \hat{x}'_{k+1}$$

Measurement, z

Estimation is updated

$$\therefore \hat{x} - \hat{x}' \triangleq K(z - \hat{z}')$$

State Estimation $\hat{x}$

Estimation Error $e = \hat{x} - x$

$x_{k+1} = Fx_k + w$ **w, v : Noise**

$z_k = Hx_k + v$

Additional Reference

Every values have Prob. distribution

Covariance P $P = E\{ee^T\} = E\{e^2\}$

Objective Prob. Error Becomes zero. →Minimization

$$E\left\{\begin{bmatrix} e_1e_1 & e_1e_2 \\ e_1e_2 & e_2e_2 \end{bmatrix}\right\} \Rightarrow E\left\{\begin{bmatrix} e_1e_1 \to 0 & 0 \\ 0 & e_2e_2 \to 0 \end{bmatrix}\right\}$$

Prediction $\hat{x}_{k+1}' = F\hat{x}_k$ We know only F, but don't know w and v

$$Generally,\ \hat{x}_{k+1}' \neq \hat{x}_{k+1}$$

Kalman Gain definition, K (Brilliant idea)

$$\hat{x} = \hat{x}' + K(z - H\hat{x}')$$

$z = Hx + v : Actual$

$z : measurement$

$z - H\hat{x}' : Observation\ error$

$$\hat{x} - \hat{x}' = K(z_k - \hat{z}'_k) \text{ and } \hat{z}'_k = H\hat{x}'_{k+1}$$

$$\hat{x} - \hat{x}' = K(z_k - H\hat{x}'_{k+1})$$

4. Kalman Gain
How to minimize Estimation Error

- Estimation Error

$$e = \hat{x} - x$$

- Covariance = cost function of estimation error

$$P = E\{e^2\} = E\{(\hat{x} - x)^2\}$$

$$P = E\{e^2\} = E\{(\hat{x} - x)^2\}$$

$$\hat{x} = \hat{x}' + K(z - H\hat{x}')$$

$$e = \hat{x} - x$$

$$e' = \hat{x}' - x$$

$$P = E\{(\hat{x}' + K(z - H\hat{x}') - x)^2\}$$

$$= E\{(\hat{x}' + K(Hx + v - H\hat{x}') - x)^2\}$$

$$= E\{(\hat{x}' - x + Kv - KH(\hat{x}' - x))^2\}$$

$$= E\{((\hat{x}' - x)(I - KH) + Kv)^2\}$$

5. P update
4. Kalman Gain
1 Dim example of Minimum Estimation Error

$$P = E\{(e'(I-KH)+Kv)^2\}$$
$$= (I-KH)^2 E\{e'^2\} + K^2 E\{v^2\} + 2(I-KH)KE\{ev\}$$
$$= (I-KH)^2 P' + K^2 E\{v^2\} + 0$$
$$= (I-KH)^2 P' + K^2 R$$

$$R = E\{v^2\}$$

Independent
Event
Covariance=0

Our goal is Covariance P becomes smaller

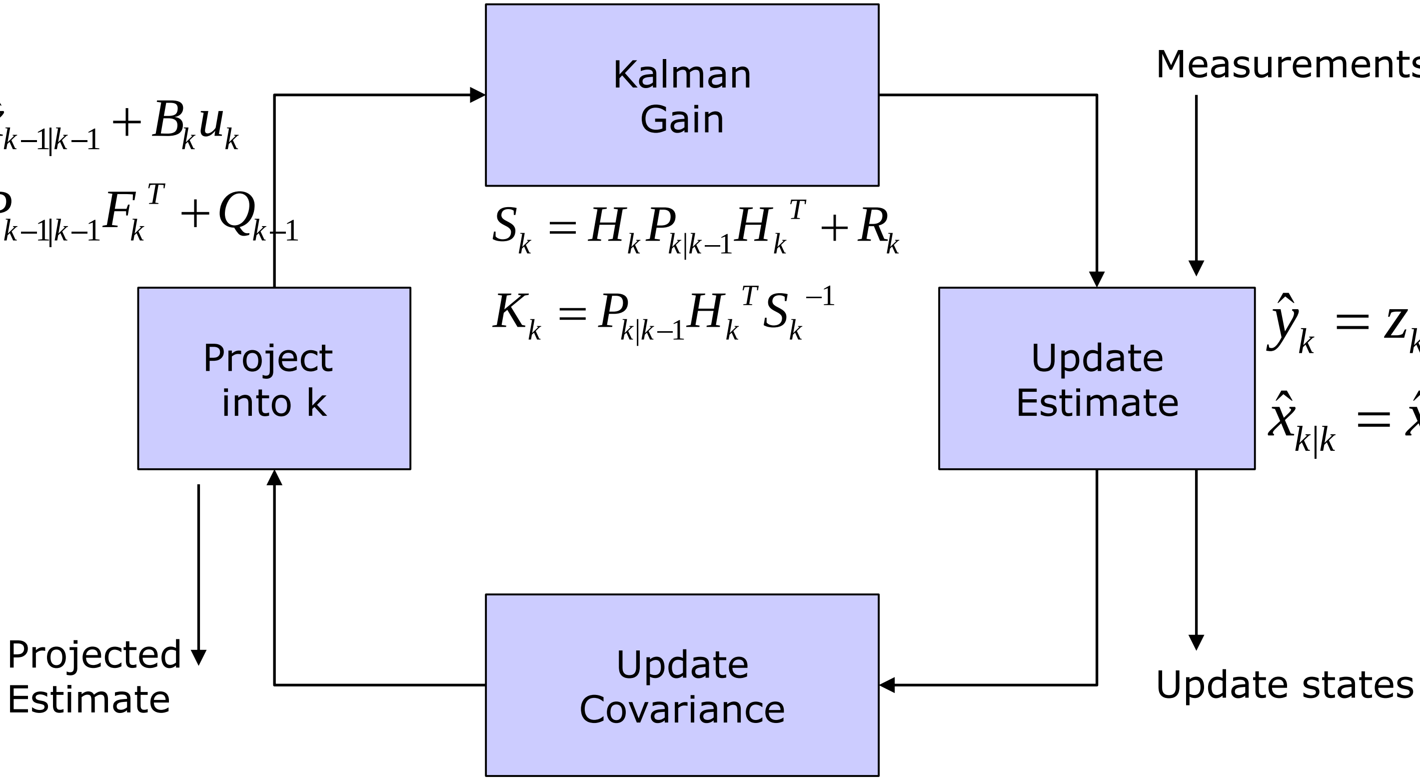
$$\frac{dP}{dK} = -2H(I-KH)P' + 2KR = 0$$

$$\therefore K = \frac{HP'}{H^2 P' + R} = \frac{HP'}{S}$$

Remind Kalman Filter

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$
$$z_k = H_k x_k + v_k$$

Initial Estimates $\hat{x}_{k-1|k-1}$



$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^{\,T} + Q_{k-1}$$

$$S_k = H_k P_{k|k-1} H_k^{\,T} + R_k$$
$$K_k = P_{k|k-1} H_k^{\,T} S_k^{\,-1}$$

$$\hat{y}_k = z_k - H_k \hat{x}_{k|k-1}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \hat{y}_k$$

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

Understanding Matlab Code with K.F.

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$
$$z_k = H_k x_k + v_k$$

- We don't know exact w, v
- But, we know Covariance from Gaussian Distribution of w,v.

Initial Estimates $\hat{x}_{k-1|k-1}(xp = 3)$

$$\hat{x}_{k|k-1} : \text{Prediction (xp)}$$

$$\hat{x}_{k-1|k-1} : \text{Estimate (xe)}$$

Understanding Matlab Code with K.F.
1. Prediction

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$
$$z_k = H_k x_k + v_k$$

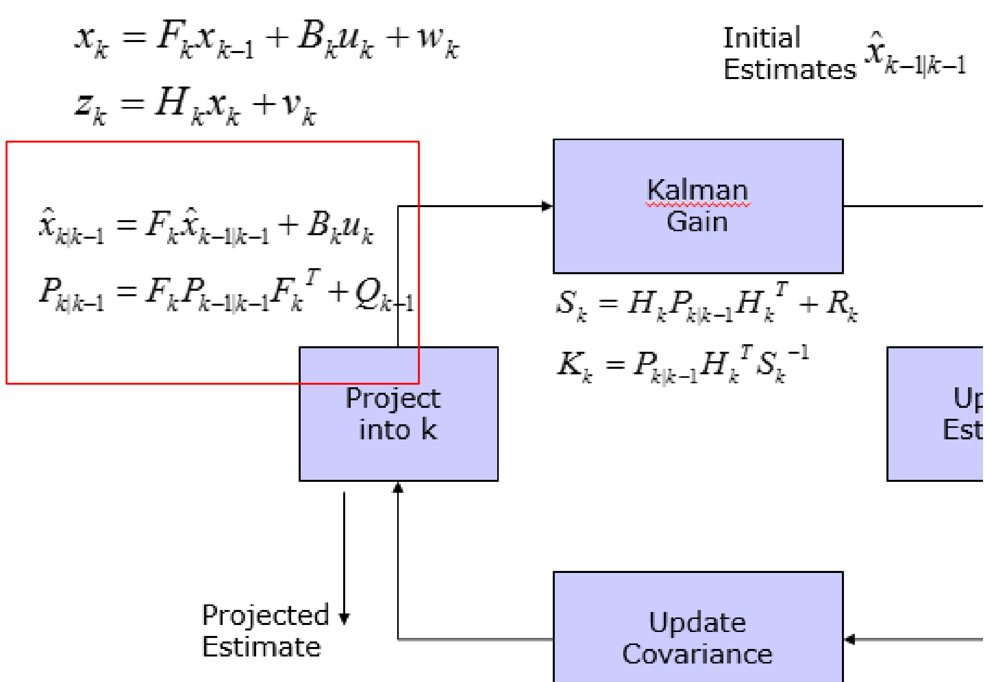
$\hat{x}_{k|k-1}$: Prediction (xp)

$\hat{x}_{k-1|k-1}$: Estimate (xe)

Prediction

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_{k-1}$$

```
w = sqrt(Q)*sin(0.1*i);
v = randn(1)*sqrt(R);

% system dynamics
x = (1-dt)*x + w;
z = h*x +v;

F = (1-dt);

% prediction
xp = F*xe;
Pp = F*Pe+Q;
```



State variable cannot be Directly Measured We should estimate state variable by Prob.

$$x_k = F_k x_{k-1} + B_k u_k + w_k \qquad w \sim N(0, Q)$$

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$$

$$\hat{x}_{prediction} = F_k \hat{x}_{estimate} + B_k u_k$$

$$P : \text{covariance of state variable estimate}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^{\ T} + Q_{k-1}$$

Understanding Matlab Code with K.F.
2. Kalman Gain

$$z_k\,(\text{observation}) = H_k x_k + v_k, \quad \text{v\textasciitilde N(0,R)}$$

```
% prediction
xp = F*xe;
Pp = F*Pe+Q;

% Kalman gain
S = h*Pp*h + R;
k = Pp*h/S;

% Correction or Update
y  = (z- h*xp);
xe = xp+k*y;
Pe = (1-k*h)*Pp;
```

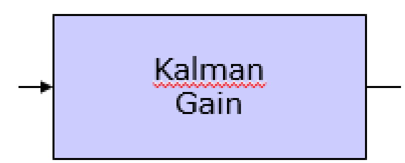
- New Estimate

= Prediction + Measure * Kalman Gain

Kalman Gain

$$S_k = H_k P_{k|k-1} H_k{}^T + R_k$$

$$K_k = P_{k|k-1} H_k{}^T S_k{}^{-1}$$



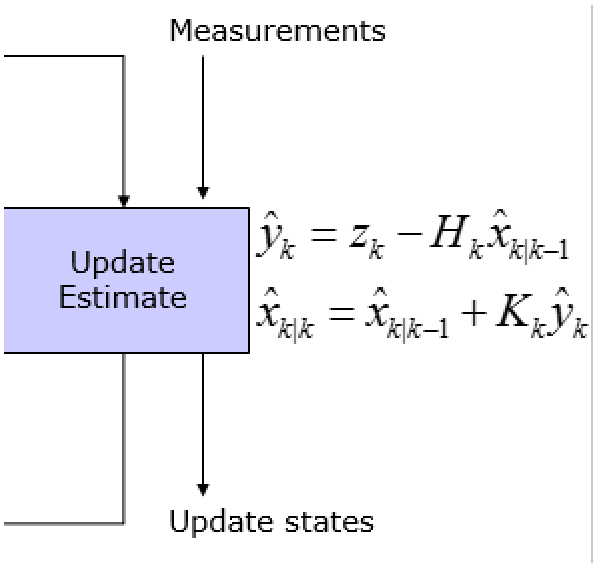
$$S_k = H_k P_{k|k-1} H_k{}^T + R_k$$

$$K_k = P_{k|k-1} H_k{}^T S_k{}^{-1}$$

Understanding Matlab Code with K.F.
3. Correction(Update Estimate)

- We want to know x_{k-1} but only know estimate $\hat{x}_{k-1|k-1}$
- We also know prediction $\hat{x}_{k|k-1}$

```matlab
% Correction or Update
y  = (z- h*xp);
xe = xp+k*y;
Pe = (1-k*h)*Pp;
```



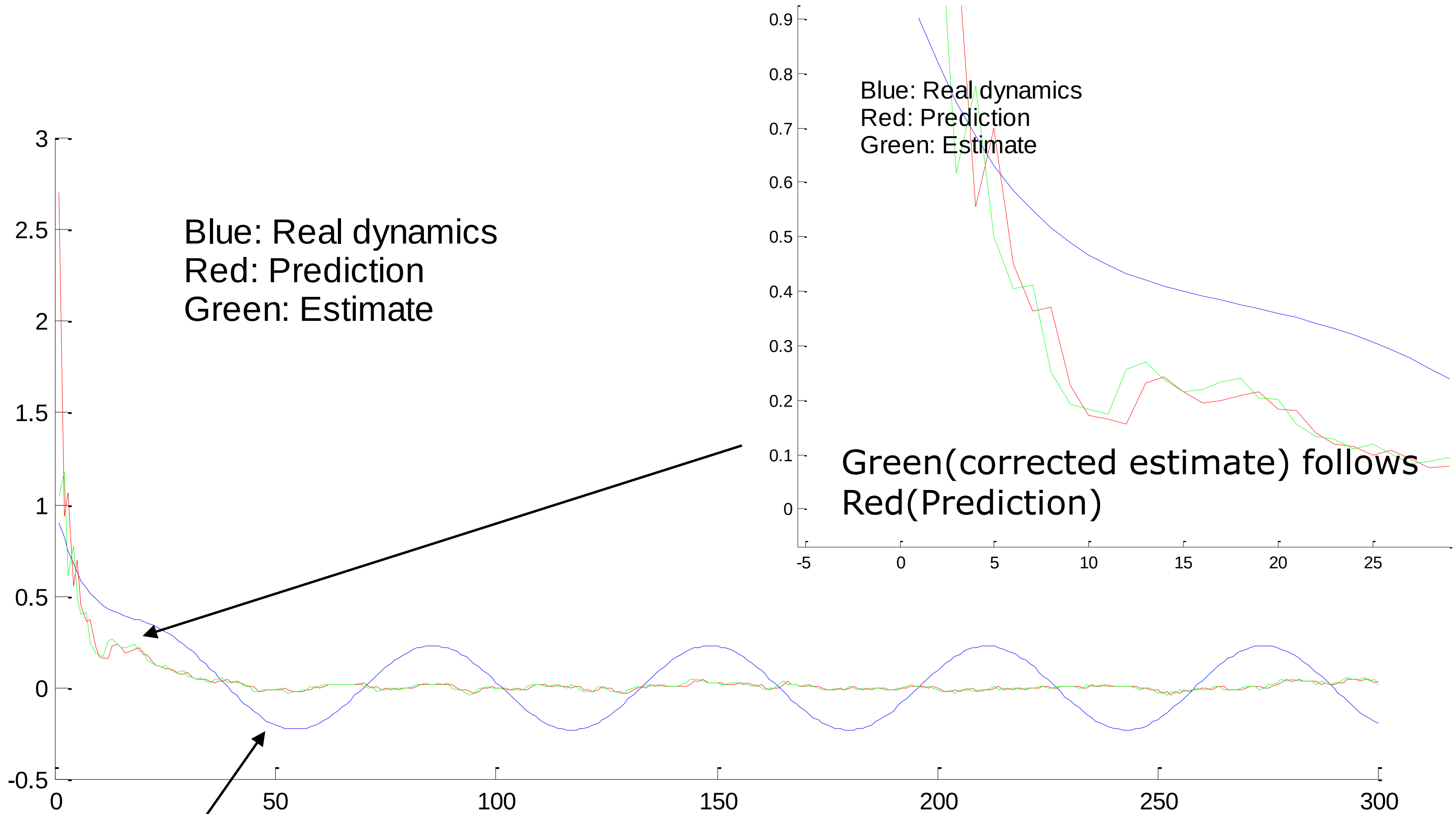
Correction

$$\hat{y}_k = z_k - H_k \hat{x}_{k|k-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \hat{y}_k$$

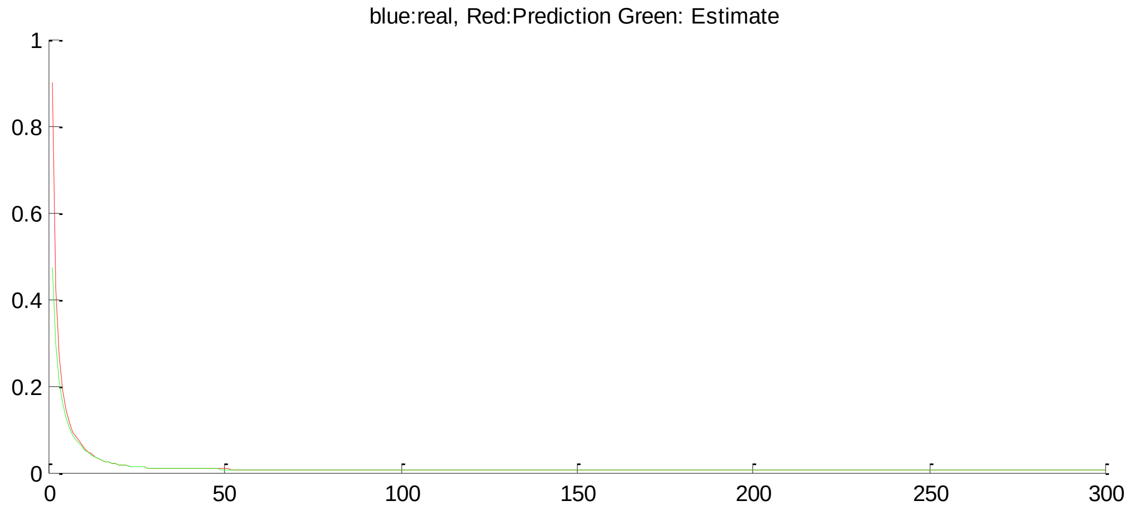
$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

Example Test3



Process Error

Covariance P becomes very Smaller.



- Process Noise Covariance Q makes system to be oscillatory.
- P(Covariance of state variable estimate) becomes smaller → State variable estimate is believable.