Mobile Robot
Probabilistic Robotics
Lecture 4

Jeong-Yean Yang

2020/10/22

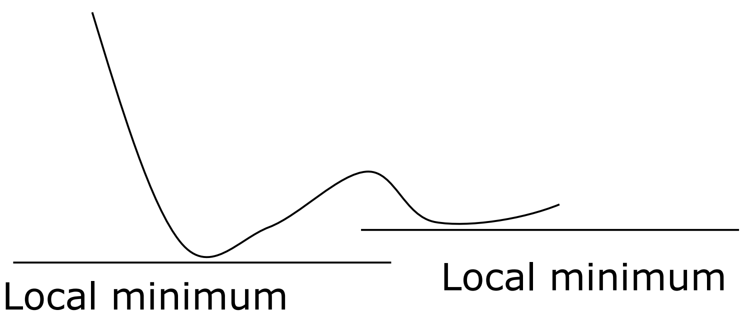
1

Question
Why Probabilistic Approach is Different?

Change your Thinking Styles from Examples

Review

Optimization
Convex hull should be exist



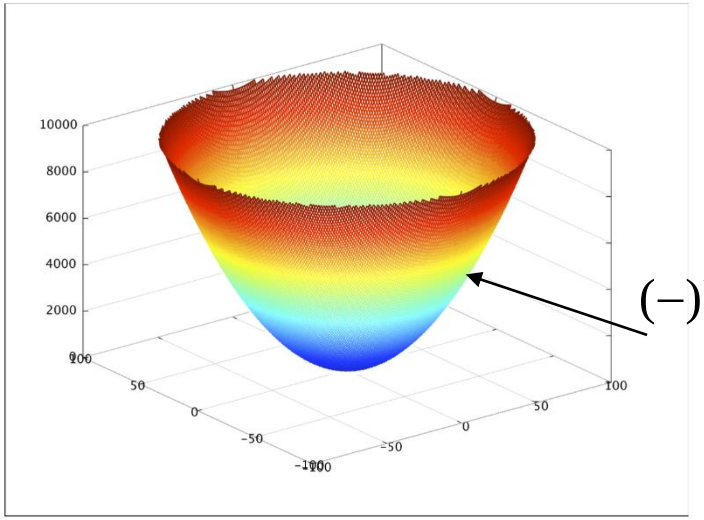
- Convex hull means that it is one of the local minima.
- Think that Y=x^2 has the minimum value at x=0

- In 2D problem. A = x^2 +y^2 has the minimum at x=0 and y=0

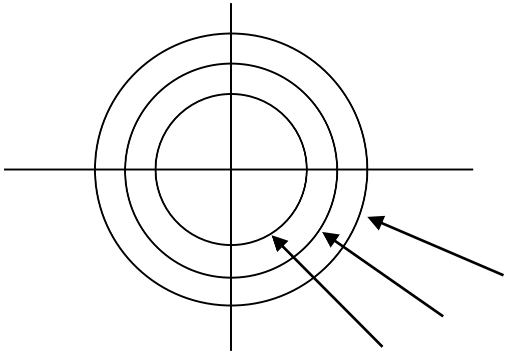
Review

Gradient Descent follows $(-)\nabla f(x,y)$

f(x,y) = x^2+y^2



$(-)\nabla f(x,y)$



- Gradient Descent stops at minimum.

$X \leftarrow guess$

Repeat

$X \leftarrow X - \alpha \nabla f(X)$

if $X - X_{old} < \varepsilon$ then stops.

Review

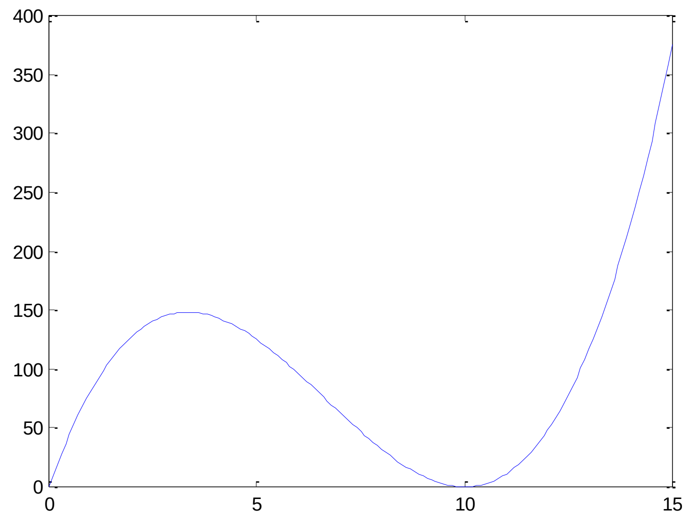
Gradient Descent Method, GDM

- You should learn GDM.
- Neural network is one of the example based on GDM.
- Over 80% of engineering method uses GDM.

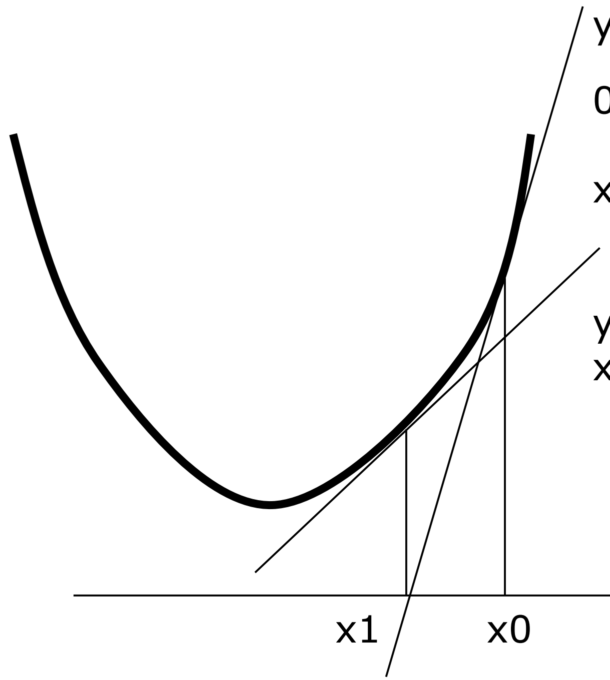
- However, before use GDM, you should think about the convex hull problem.
- If there is no convex hull, GDM will be diverged.

Differentiation-based Optimization for Complex Function

- Local Minimum can be obtained by GDM.
- Back to 2D problem.



y=x(x-10)^2



y=f′(x0)(x-x0)+f(x0)

0=f′(x0)(x-x0)+f(x0)

x1 ←- f(x0)/f′(x0)+x0

y=f′(x1)(x-x1)+f(x1)
x2←- f(x1)/f′(x1)+x1

....

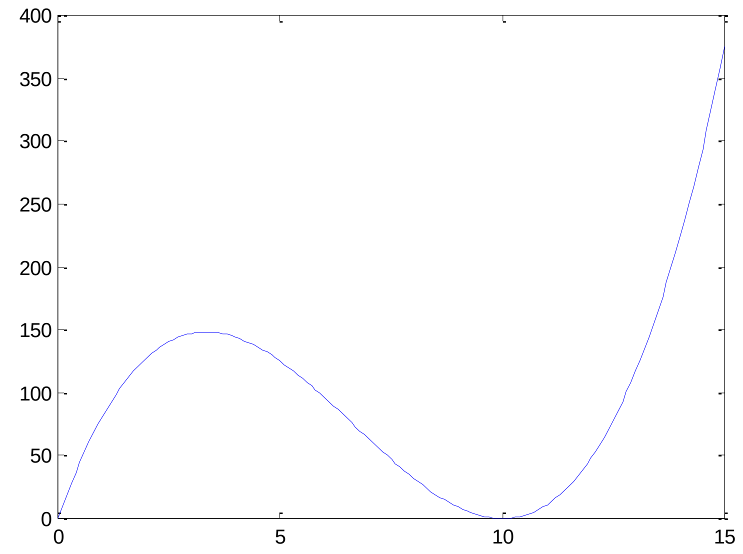
f′(x)→0

Case 1) y=x(x-10)^2

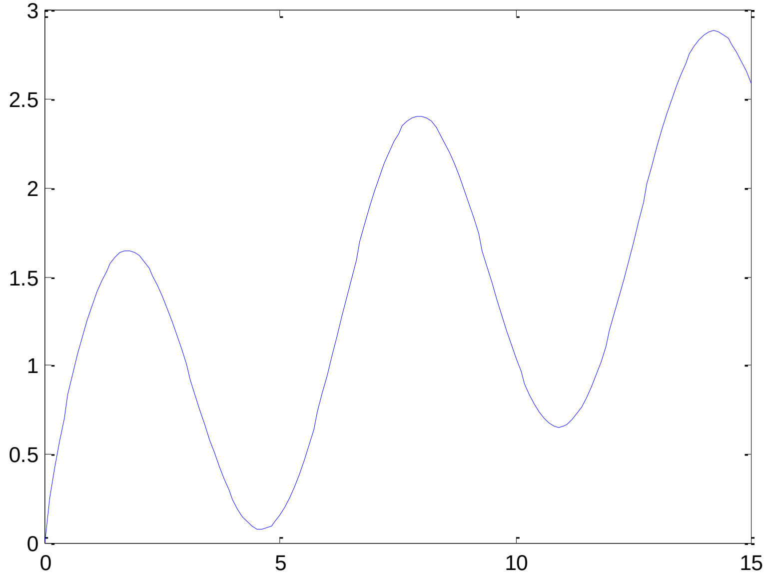
```
clear
x=0:.1:15

y = x.*(x-10).^2;
% y'= (x-10)^2 + 2*x(x-10);

figure(1);
plot(x,y);

% guess x0
x0 = 6;

x = x0;
for i=1:20
    fx = x*(x-10)^2;
    dfx= (x-10)^2 + 2*x*(x-10);
    x  = -fx/dfx+x;
     [x fx]
end
```

Test1.m

- Y’ = (x-10)^2 + 2x(x-10)
- X→ 10

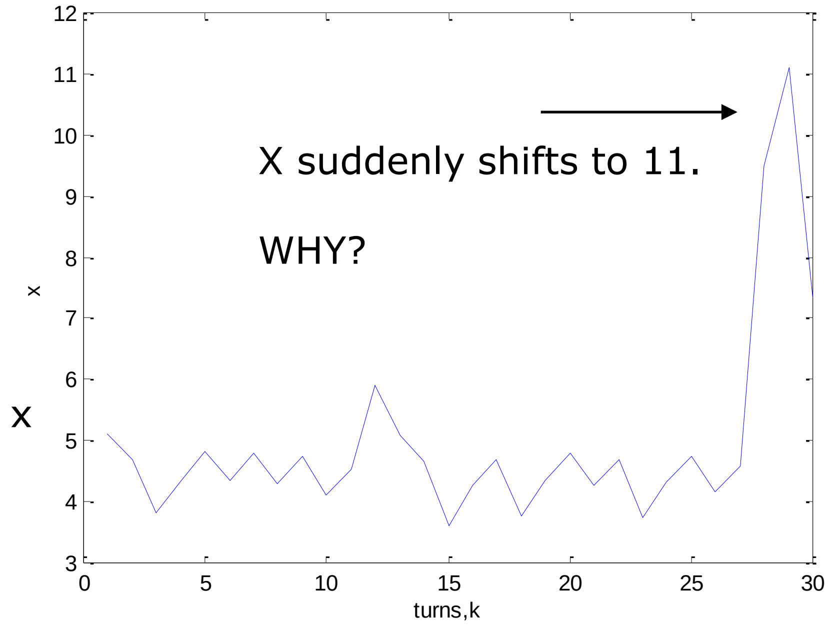


Case 2) sinX + 0.5sqrt(X)



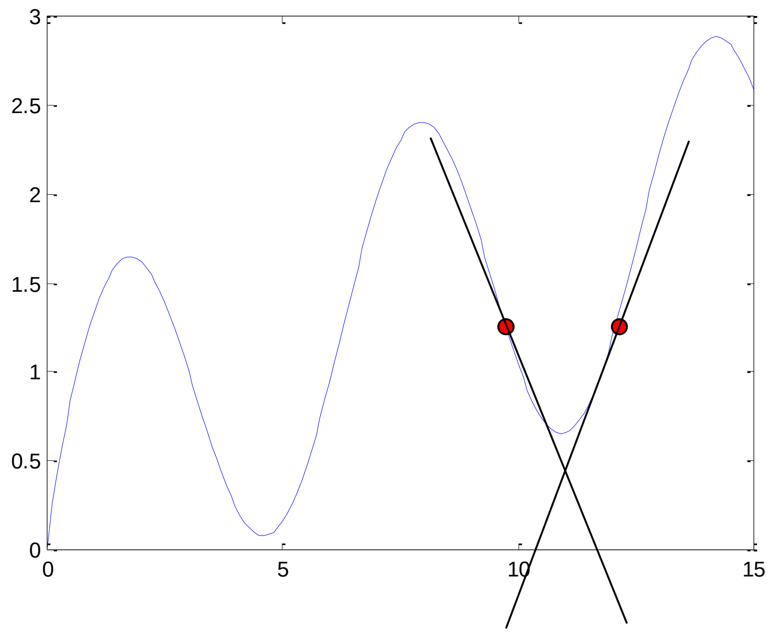

```
% guess x0
x0 = 6;

x = x0;
for i=1:50
    fx = sin(x)+0.5*sqrt(x);
    dfx= cos(x)+0.5*0.5*x^-0.5;
    x  = -fx/dfx+x;
     [x fx]
end
```

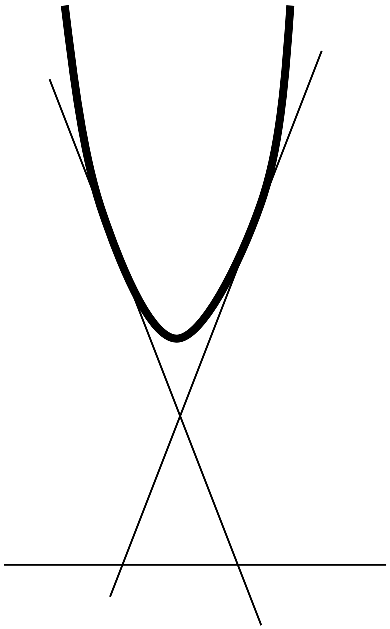
- Result



Iterative Method often OSCILLATE!

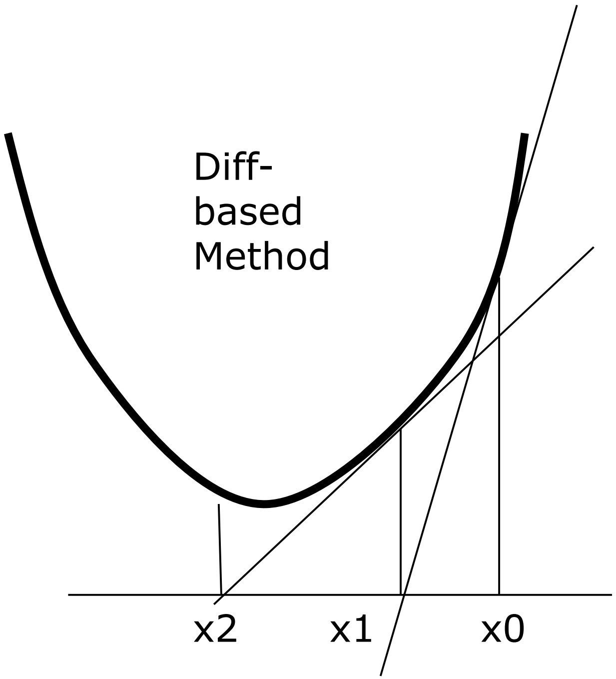
sinX + 0.5sqrt(X)

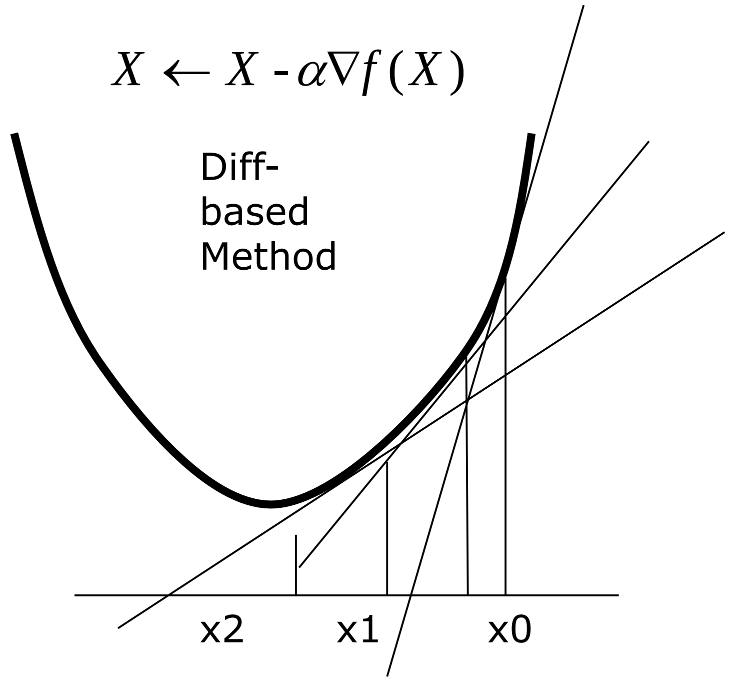




- Sin (x) is symmetric.
- Symmetric curves often generate oscillation, which does not converge to solutions.

Differentiation-based Method is Too fast to Find a Solution

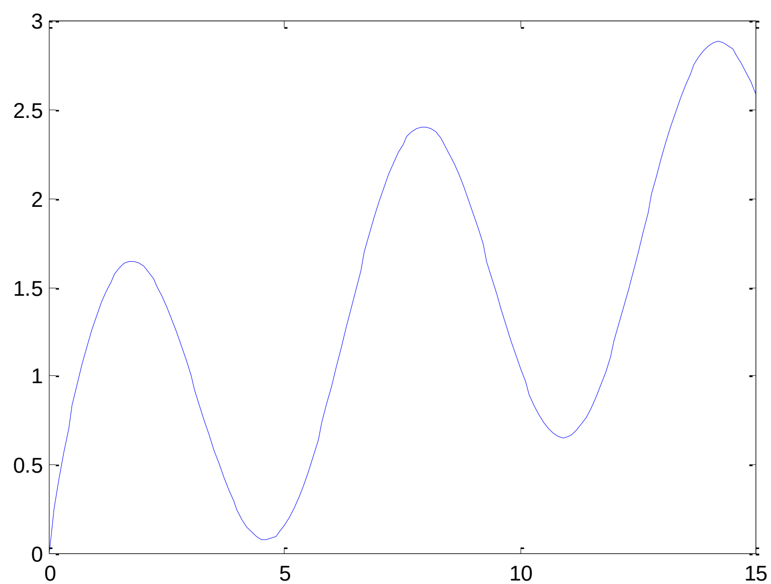




- Diff.-based Method is fast, but it becomes unstable.
- Gradient Descent Method is used alpha value, α to control convergence rate

How to Avoid Oscillation and how to find global minimum?

- Hmm.. Trial and error
  - Many trials with different initial guess, Xo.
  - Xo = 6 fails. Xo =5 fails. Xo=2 fails.
- Fortunately, a good guess can solve problem.



Random Guess can solve it?

- If there are many local minima,
  - Random guess + GDM can solve problem.
- Random guess takes a lot of time.
  - It is inefficient in general.
  - But, random process (or perturbation) GUIDES to optimal solution

Differentiation

- Differentiation or Gradient Descent Method
  - Requires Differentiation but it becomes very Large value
  - Ex) dy/dx = 1/0.00001 → infinite.
  - Iterative method becomes UNSTABLE.

0=f'(x0)(x-x0)+f(x0)

x1 ←- f(x0)/f'(x0)+x0

$$\therefore x_{k+1} \leftarrow x_k - \frac{f(x_k)}{f'(x_k)}$$

Small f'(x) generates large noise.

→ X becomes unstable…

On behalf of Differentiation, Stochastic Searching is Stable in Every time

- Genetic algorithm.
- Think Gene for optimal agents in our environment.

- X= [x1,x2,x3,…]
- F = F(X)
- Good = sort(F)
  - Crossover = average of some Good.
  - Mutation = random value
- X← [Crossover, mutation]

Genetic Algorithm

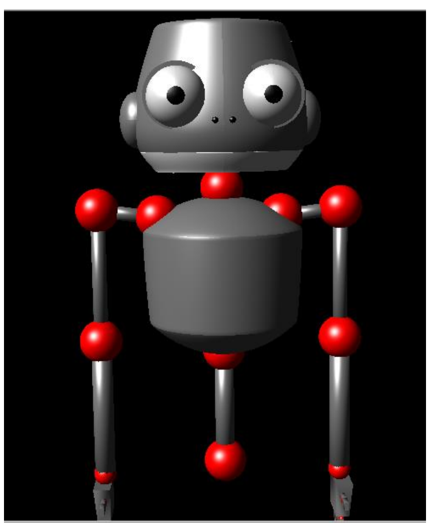
```
x=rand(1,10)-0.5;
x = 2*x+10;

for i=1:100
   fx = sin(x)+0.5*sqrt(x);

    [y I] = sort(fx);
    mx = x(I(1:4));
    crossover = [ (mx(1)+mx(2))/2,
        (mx(2)+mx(3))/2];

   x = rand(1,8)-0.5;
   x = 2*x+10;
   x = [crossover x];

   best = x(1);
   [best sin(best)+0.5*sqrt(best)]
end
```

- Ten x is used as initials.
- Find the best one
- 4 best X is averaged. (best1+best2)/2, (best3+best4)/2. → Crossover
- 8 new X are randomly chosen → Mutation

Human-like motion?

- We are experience in computer program.
- Everything is designed and there is no ERROR.
- Think different!

- Which one is better?



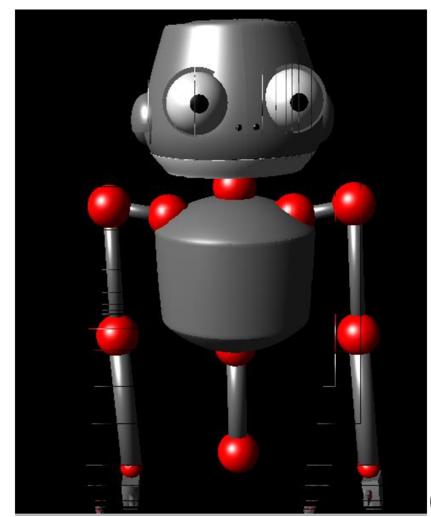
Case 1

P control

Case 2

P control
+
W(noise)

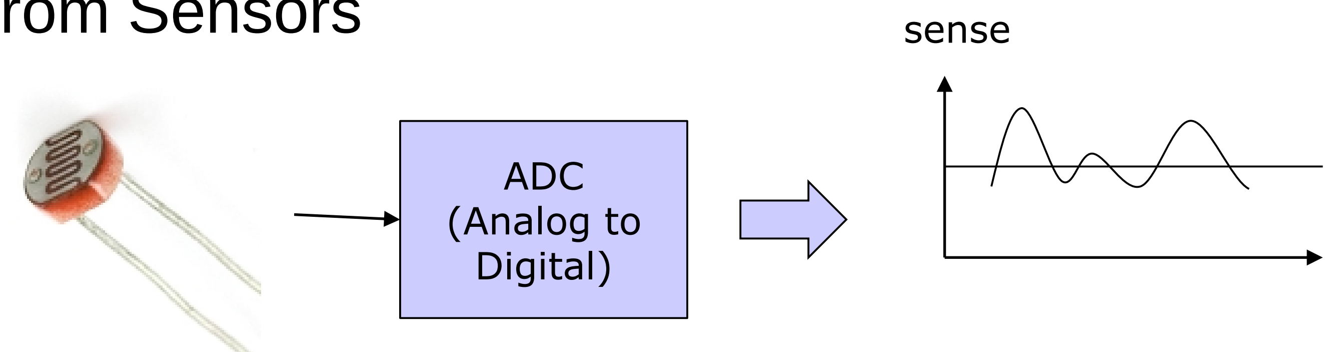
Bad
For Large Noise



2 Probabilistic Robotics

Every Sense and Action are NOT Determined.

- Senses from Sensors



- Actions with actuators

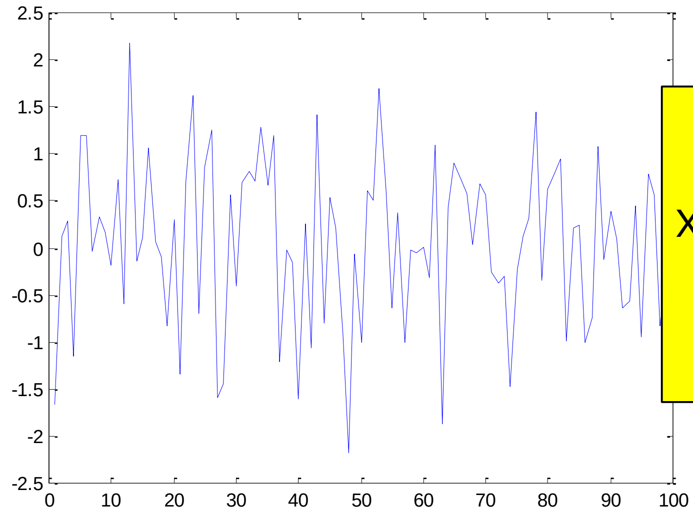
- Everything is under Probabilistic Distribution(Modeling)

Gaussian Distribution

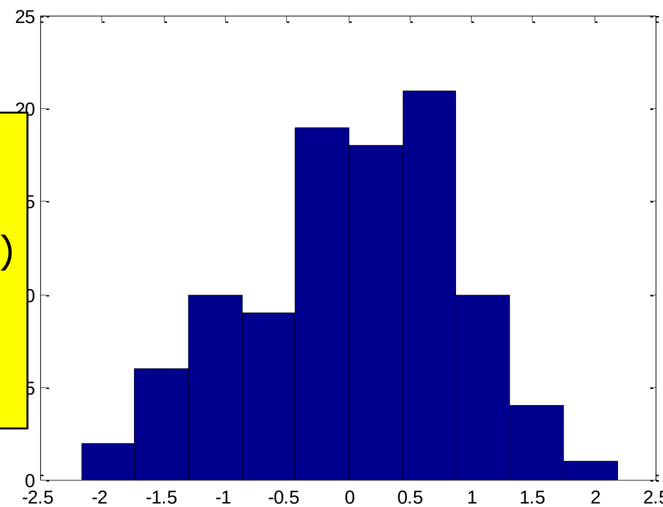
$$x \sim N(\mu,\sigma),\ \ \mathrm{PDF} = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

μ : mean σ : standard deviation

Probabilistic Density Function (PDF)



Matlab
X=normrnd(0,1,100,1)
Plot(x)
Hist(x)



$$x \sim N(\mu,\sigma)$$

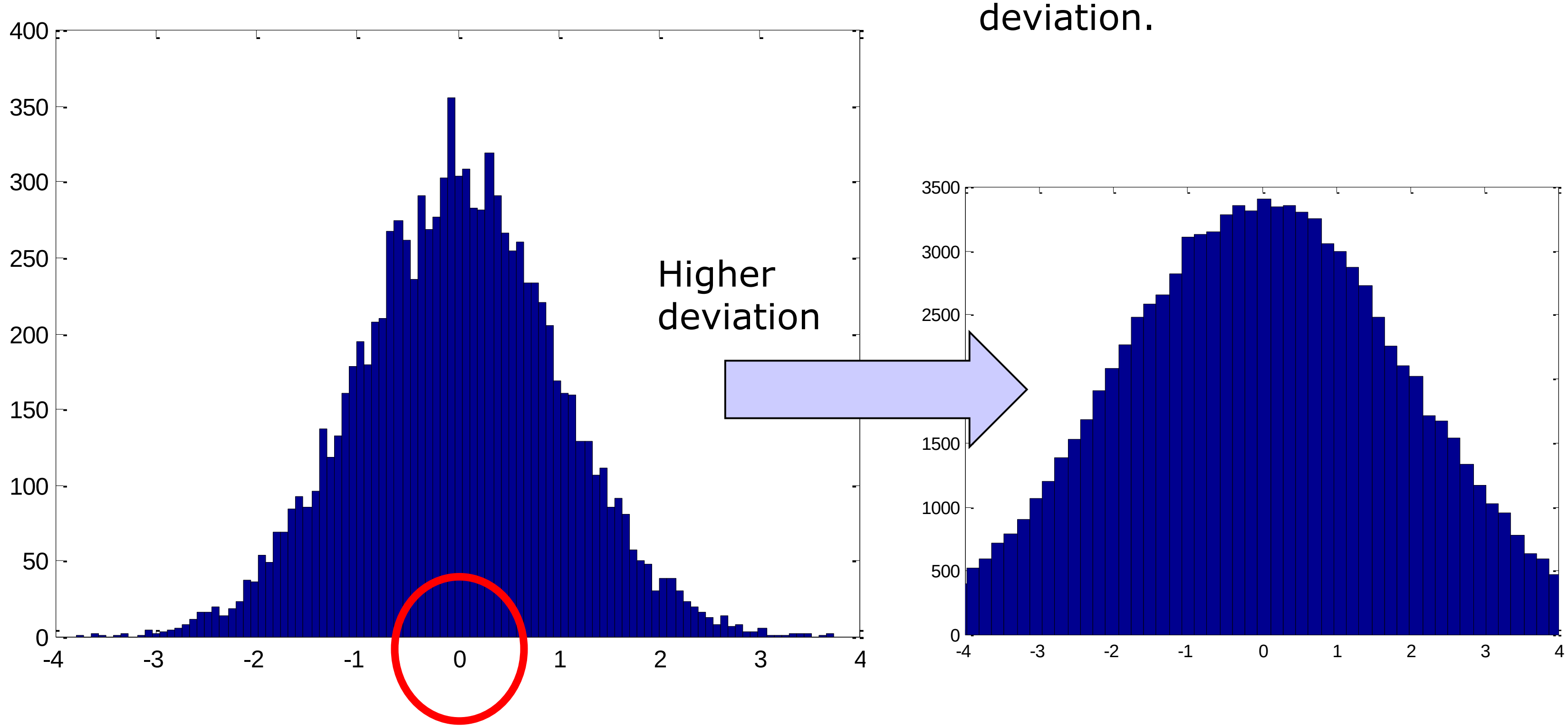
- X has Gaussian distribution with mean and deviation

With More Samples,

- 10000 samples,

Test1.m

See the changes of mean and deviation.



Mean value

Stochastic Process in Robotics

- 1. **Noise is unavoidable phenomenon**
  - Random process should be under consideration
  - Our world is filled with Noise, but nothing in mathematics and simulation world

    → Noise modeling is required
- 2. Thus, Random process is favorable in many cases.
  - Random process (or perturbation) increases the possibility of finding solution.
- 3. Random Process is bad for mathematic modeling?
  - No, Prob. Modeling is possible.
  - Yes, because you CANNOT use “If – then” method.
  - Therefore, OPTIMIZATION is required for stochastic process

Deterministic Model Vs. Stochastic Model.

- In control or some methods,
  - Modeling is required.
  - Something beyond modeling is curse for designers.

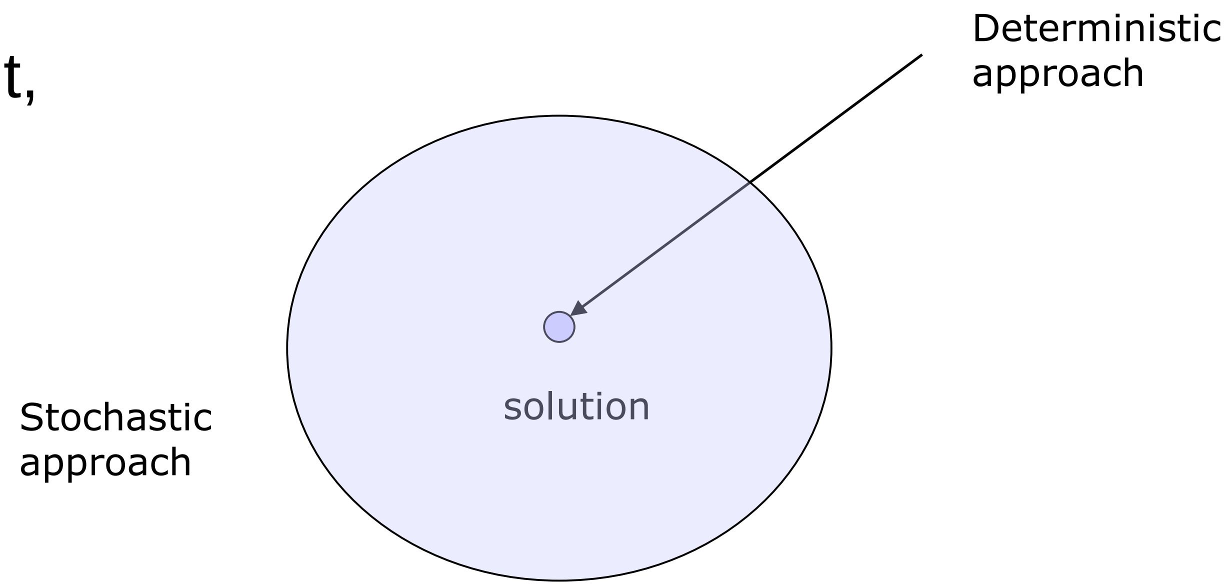
- Random property could be GOOD, but it is NOT accurate.
  - Optimization is required as in the case of GA problem.

- When GA is proposed, some peoples said that,
  - It is NOT good method. We cannot estimate when GA will be over.
  - But, think different…….

Robotics Trend is Rapidly Shifting from Deterministic to Stochastic

- We cannot determine when GA will stop.
- Even we cannot say that it is an optimal value.
- GA has been perished.

- But,



the SOLUTION lies around probabilistic condition

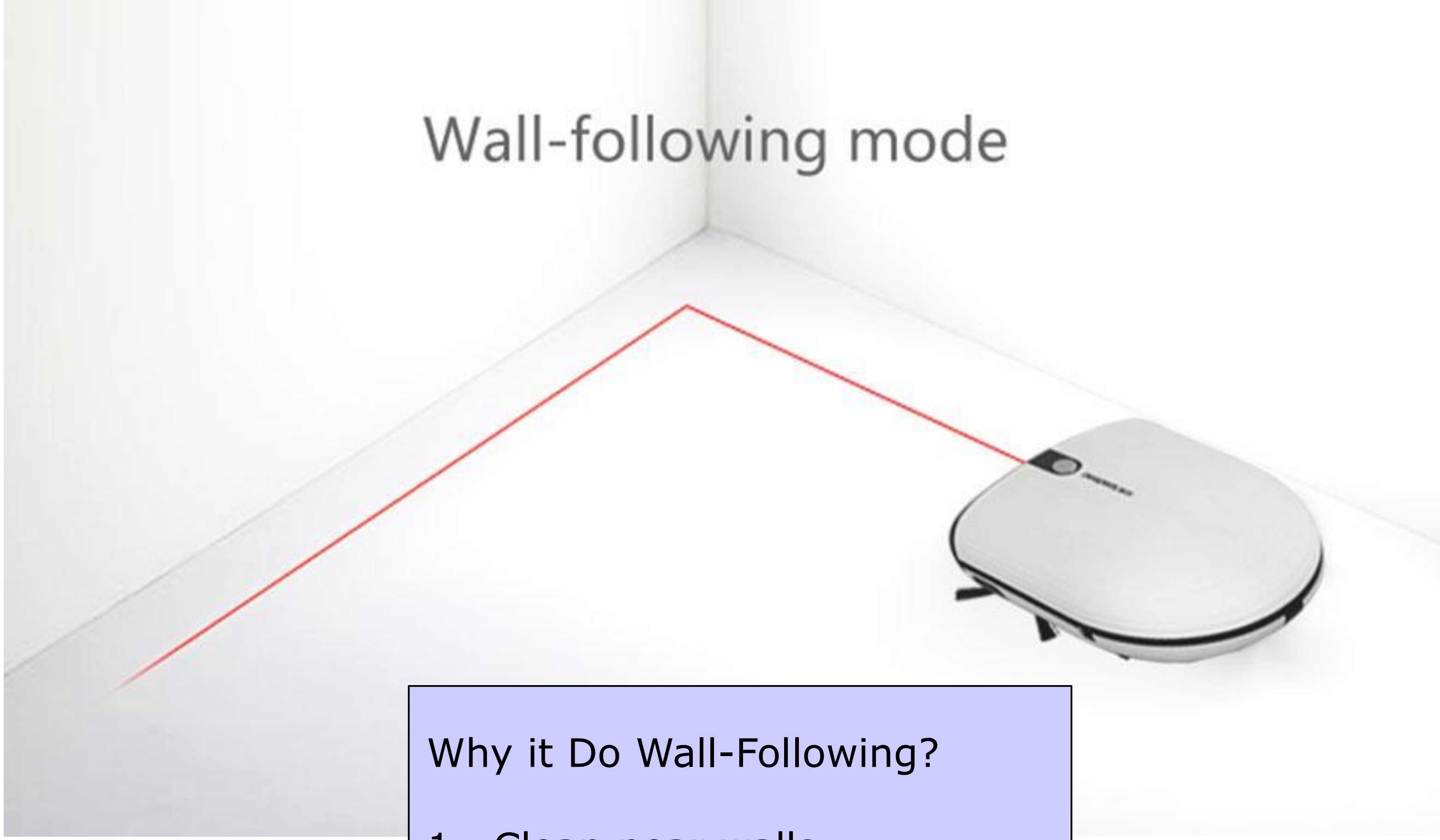
What is Stochastic Method?

- Rule 1. Stochastic or Probabilistic method does NOT follow the exact solution.
  - Maybe there will be a solution…I think so.
- Rule 2. Stochastic or Probabilistic method has some kinds of random values.
  - Thus, Stochastic method requires Optimization.
- AI has been based on deterministic method.
- Nowadays, Learning is based on Stochastic Method.

3

Stochastic Approach
Simple Example with Case Study

Example: Cleaning Robot

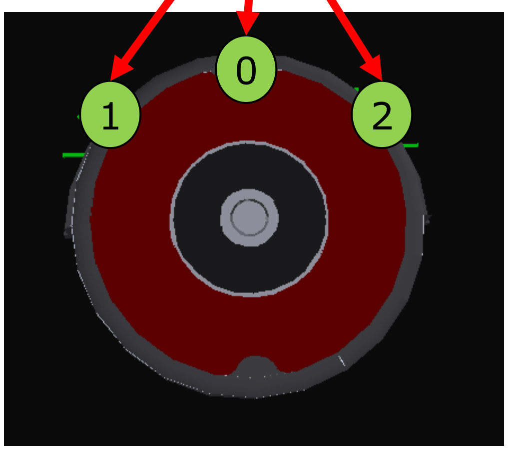


Why it Do Wall-Following?

1. Clean near walls
2. No sensor for map building

Example: Cleaning Robot

Three distance
sensors 0~2



$$\hat{\dot{x}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \frac{r}{2} \begin{pmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ -1/a & 1/a \end{pmatrix} \begin{pmatrix} w_L \\ w_R \end{pmatrix}$$

Random
Noise
For an actual
Environment

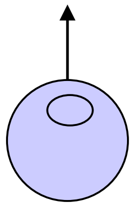
Only two inputs

Strategy
How to move it

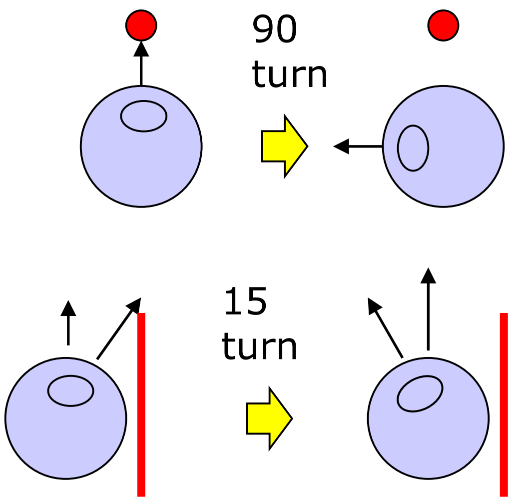
Deterministic Strategy

- X = [x,y,q]
- Xd=[xd,yd,qd] desired value.

- Case 0) No avoidance.
  - wL=wR= w0



- Case 1) Center Sensor < threshold
  - Frontal obstacles.
  - New target = qd = 90.
- Case 2) Right Sensor < threshold
  - Right obstacles.
  - New target = qd = 15.



"Curling" with Deterministic Method

Slip and Wall Obstacles are Not Determined

- Why it fails to do wall following?

- 1.Wheel slip is Not determined
  - Small Errors from Wheel slips generates Bigger Errors
  - Rotation with two wheel Encoders CANNOT be accurate.

- 2. Wall is the sources of Uncertainties(Noises)
  - Wall is not fixed.
  - Even in fixed walls, measurement errors occur

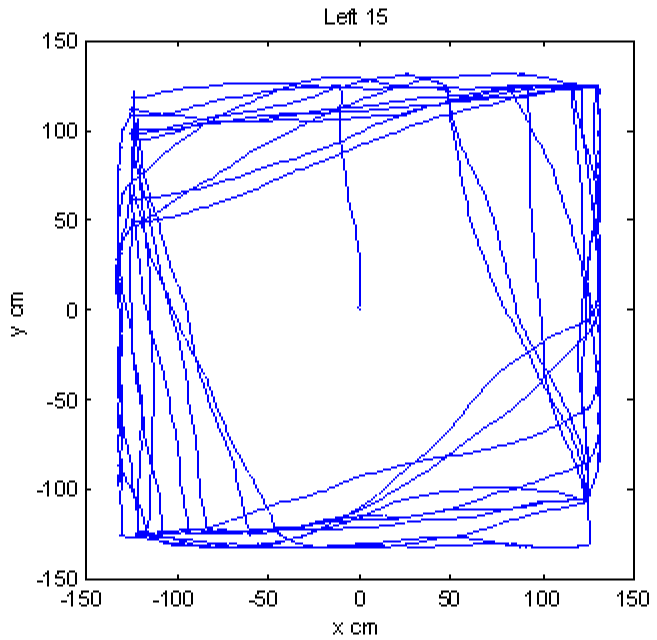
- Result: Environment is a Stochastic World

Stochastic Environment for Cleaning Robots: What We Have to Change?

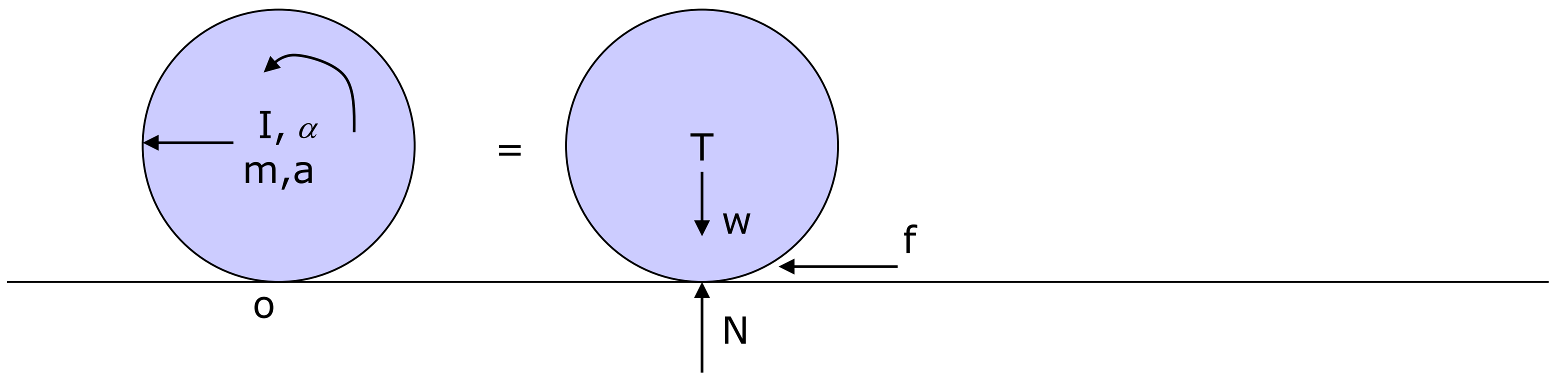
- Stochastic Environment
  - Everything is probabilistically determined.
  - Think the distribution : X~N(mean, sigma)

- Simple rules from Deterministic ways
  - 1. Go straight
  - 2. If no obstacles then jump to 1.
  - 3. Turn to left.
  - 4. Jump to step 2.

- Even with Noise on Wheels..
  - Speed of Left, Right wheel ~ N(0,1)*5



Wheel Dynamics



$$1)\ ma_x = f$$

$$2)\ ma_y = 0 = -W + N$$

$$3)\ I_c\alpha = \left(\frac{1}{2}mr^2\right)\alpha = T - rf$$

$$\therefore \frac{1}{2}mr^2\alpha = T - rf$$

$$4)\ a_x = r\alpha$$

$$mr\alpha = f$$

$$\frac{1}{2}mr^2\alpha = T - mr^2\alpha$$

$$\therefore \alpha = T\frac{2}{3mr^2}$$

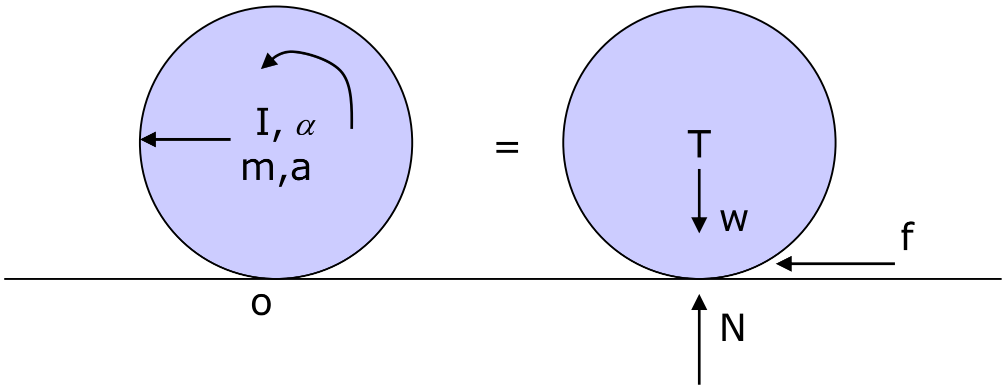
vs

When T=const., if r ↓ then α ↑ but w ↓

Dept. of Intelligent Robot Eng. MU

Probabilistic Process is required for Mobile Robot rather than any other Robotics fields.

- Mobile robot depends on wheel movement.



Rolling condition $a_x = r\alpha$

$$ma_x = mr\alpha = f$$

$$I_c\alpha = \left(\frac{1}{2}mr^2\right)\alpha = T - rf = T - mr^2\alpha$$

$$\therefore \frac{3}{2}mr^2\alpha = T$$

In case of slip

$$ma_x = f$$

$$I_c\alpha = \left(\frac{1}{2}mr^2\right)\alpha = T - rf = T - rma_x$$

$$\therefore T = \left(\frac{1}{2}mr^2\right)\alpha + rma_x$$

Rolling Vs. Slip

Rolling condition	Slip Condition
$a_x = r\alpha$ $\frac{3}{2}mr^2\alpha = T$	$ma_x = f$ $T = \left(\frac{1}{2}mr^2\right)\alpha + rma_x$

- Slip: Wheel cannot move forward ($a_x < r\alpha$) even though α is NOT zero.
- Assume that $a_x = 0$

$$\frac{3}{2}mr^2\alpha_{Roll} = T \qquad T = \left(\frac{1}{2}mr^2\right)\alpha_{slip} + 0$$

$$\alpha_{Roll} = \frac{2T}{3mr^2} \qquad \alpha_{slip} = \frac{2T}{mr^2}$$

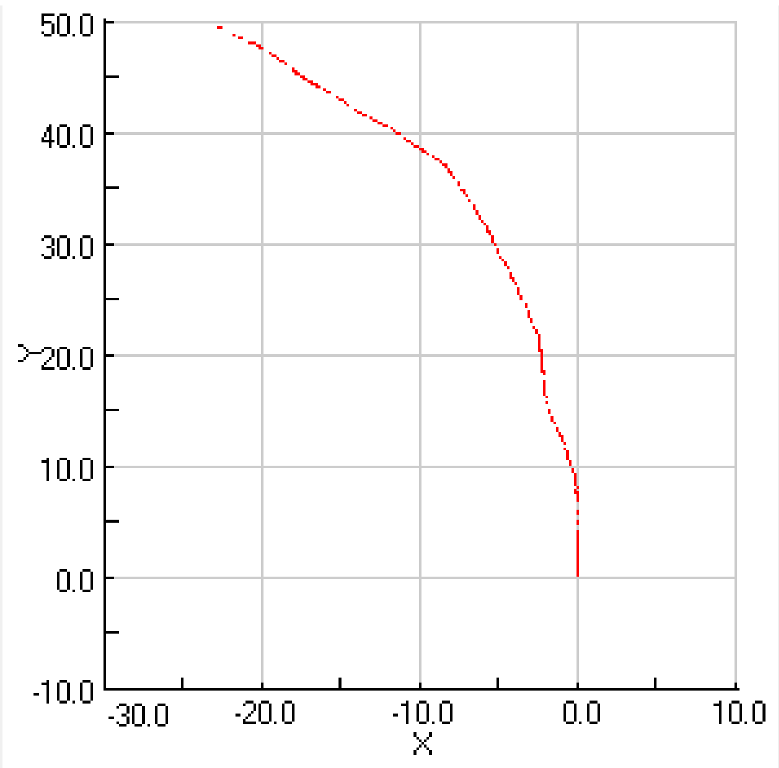
$$\alpha_{Roll} < \alpha_{Slip}$$

Slip is one of the STRONG Noises

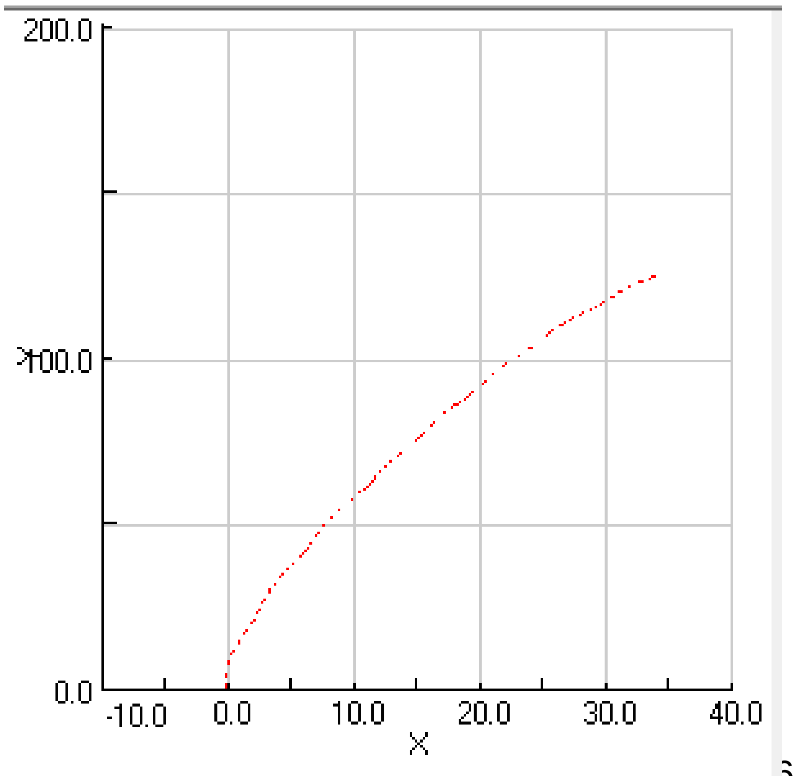
- Slip depends on road status, which is hardly observed in general.
- Mobile robot moves with two wheels
  - Jacobian $dX = Jd\Theta$ determines the movement in X-Y-q space.
  - This assumption is also from Non-Slip(Rolling) condition.
- Therefore, mobile robot is always being biased by Slip noise.
- How can we solve this?
  - Localization(SLAM) is an alternative way.
  - On the other hand, is there a good approach?....
    → Stochastic approach.

Moving in a straight line is IMPOSSIBLE in a Stochastic World

- Is it possible?
  - Actually, **0.2* N(0,1)** noise is applied



W=(wl,wr)=(1,1)



W=(wl,wr)=(5,5)

Example: Roomba

Question 1
Why does it do spiral motion?

Question 2
After bumping, why it starts to turn to left?

It does not look so useful.

Spiral motion has been known that
It is efficient way for a cleaning robot can cover larger area.

Yes, it is. However, I think that a Roomba took a very Tactical method in a stochastic world.

Mobile Robot CANNOT follow Straight line without SLAM.

- iRobot takes **excellent strategy** from my observation.
- If it surely cannot move on straight line,
  - Take the strategy of rotation → Turn to Right (Spiral motion)
  - Also, Spiral motion can find a space.

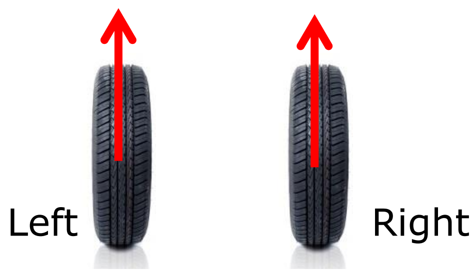
    : Answer to 1st question.
- How to search walls or avoid obstacles?
  - Turn to Right will find a space.
  - Then, turn to left will avoid obstacles.
  - Walls will be on right side.

    : Answer to 2nd question.
- Basically, two approaches are thought in stochastic thinking.

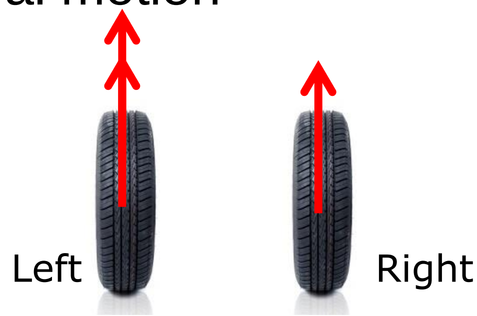
Modeling for Two Approaches in Stochastic World.

Proportional to Error

- 1.Go Straight → Turning Right for Spiral motion



Deterministic

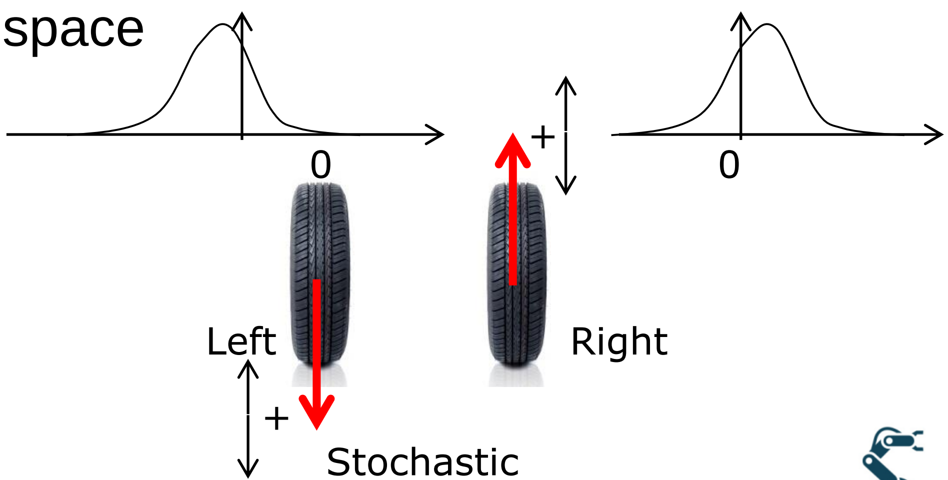


Stochastic

VL~N(m+e,1), VR~N(m,1)

- 2.Turn Left → Look for empty space for wall following

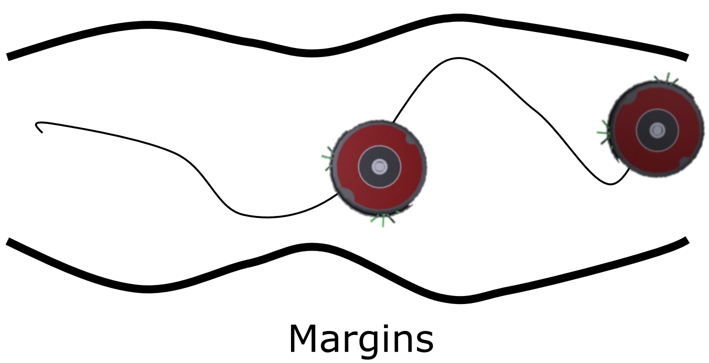
Deterministic

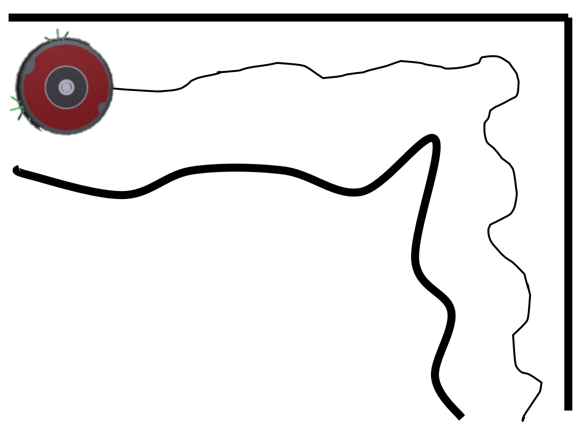


Stochastic

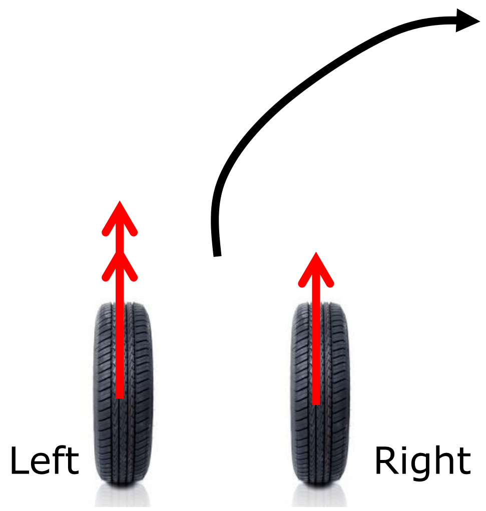
Stochastic Strategy for Cleaning Robot

- 1. Easy to Programming
  - Everything is considered under Distribution
  - Mean, Variance is the Parameter for everything.
- 2. Keeping a robot in Desired Margin
  - In the case of Trajectories, a robot should tracks within a specified marginal area





Moving in a Straight Line is Not Possible. Then, Do a Circular Motion



Stochastic

VL~N(m+e,1), VR~N(m,1)

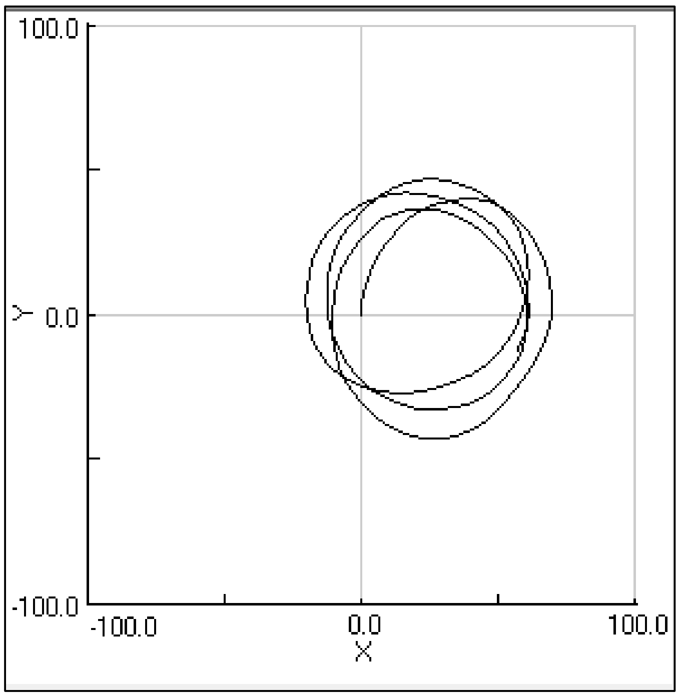
- Strategy
  - WL > WR
  - It will turn to right direction

- Environment has probabilistic noises, too.

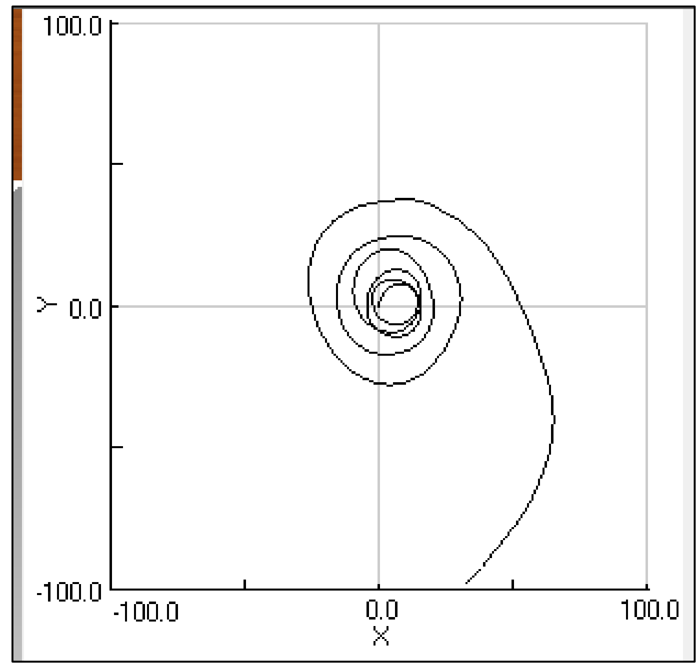
- WL~N(m+offset, s)
- WR~N(m,s)

Example: Circular or Spiral Motion

- Left: WL > WR failed for spiral motion
- Right: We need another parameterization for spiral motion
  - Exponential function is added



Only VL > VR for circular motion



VL = v0 + exp(-at)
VR = v0 – exp(-at)

Parameterization: Generally, complex problem is simplified with some parameters.

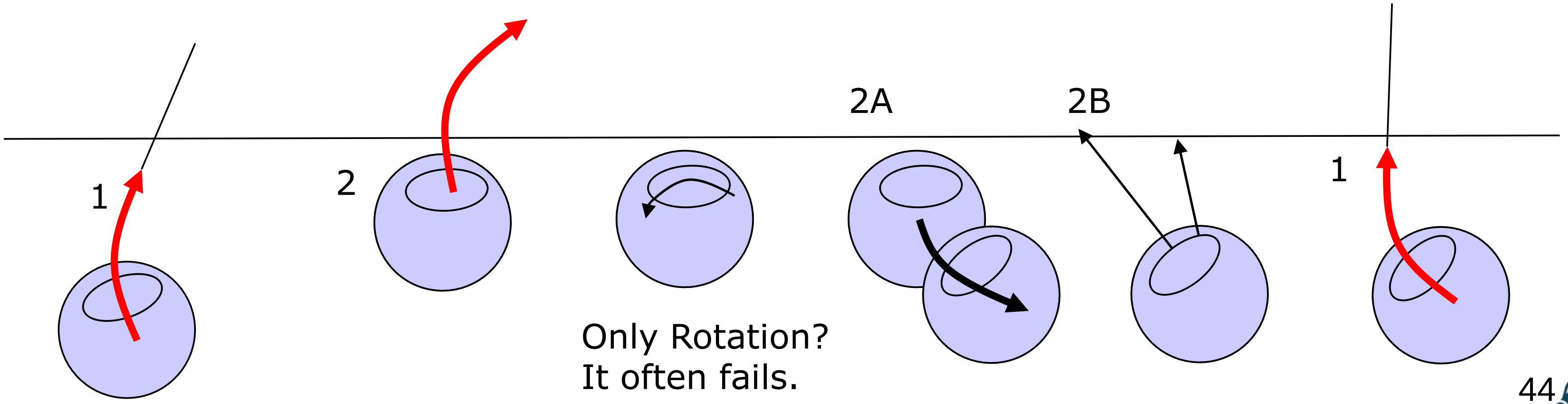
- New strategy

$$w_L = w_0 + \exp(-\alpha t)$$
$$w_R = w_0 - \exp(-\alpha t)$$

- t increases, WL =WR
- So, a mobile robot starts to "**go forward"** motion as in the case of Roomba
- Thus, 1st generation Roomba set the room size.
  - Think that room size is proportional to α
- Parameter, α can control spiral trajectory

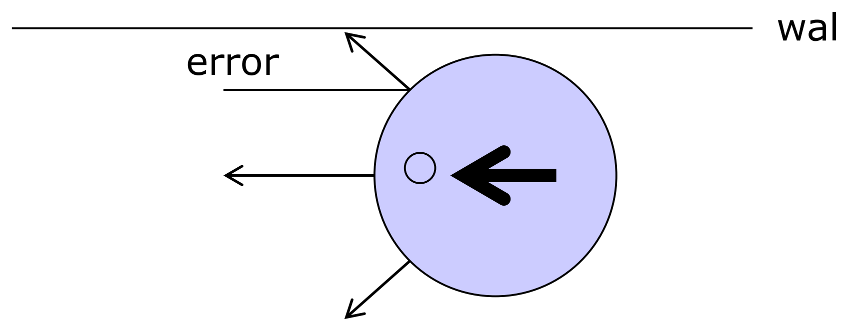
Wall following

- 1. Spiral motion should be Wall Detection
  - Movement for a Right Open Space
- 2. If a robot detect a wall, make a Left Open Space
  - A.First, Left Backward moving
  - B.Then, a robot find a Left Open Space (by Backward + Left turn)
  - C. Do Spiral motion.

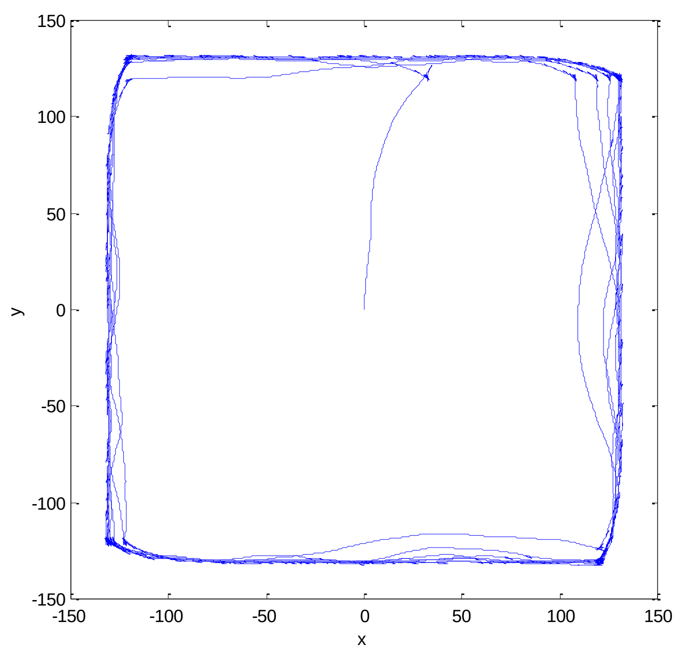


Wall Following

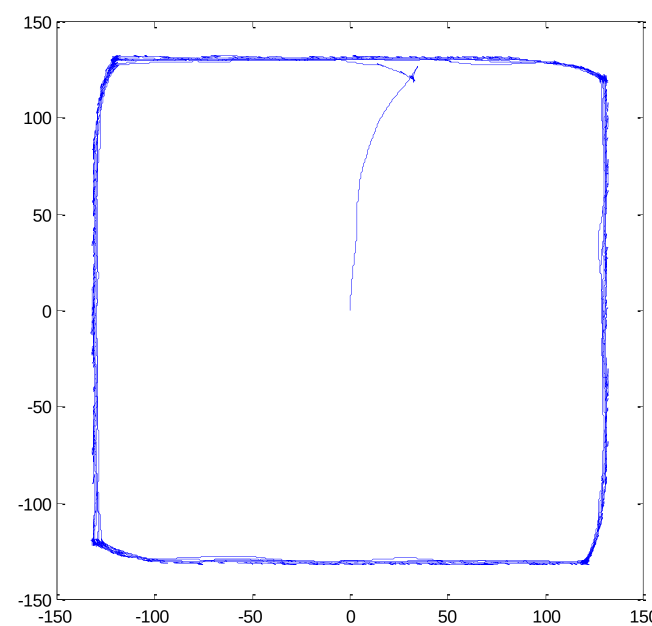
1. Go straight with R=V0, L = V0+e
   if Distance of Right > e then
   R = V0, L = Vo+e*K. (K>>1)
2. If no obstacles then jump to 1.
3. Turn to left with Perturbation
4. Jump to step 2.



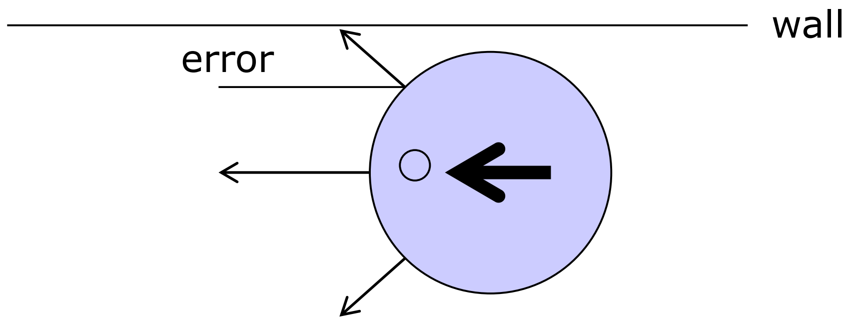
Only Unbalanced Perturbations



Following Walls



Over Bumping …
Double Threshold Instead of Steering Control

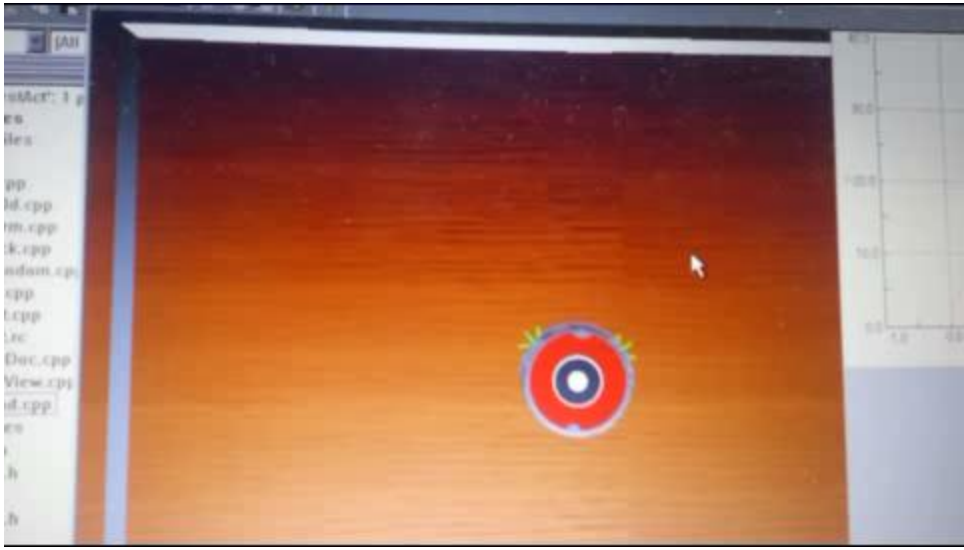


- Steering Control for Keeping Distance Error
  - Sensor Noise is too High and Expensive method.
- Double Threshold
  - if Distance of Right > e then
  - R = V0, L = Vo+e*K. (K>>1)
  - if Distance of Right < e2 then
  - R = V0+e' , L = V0

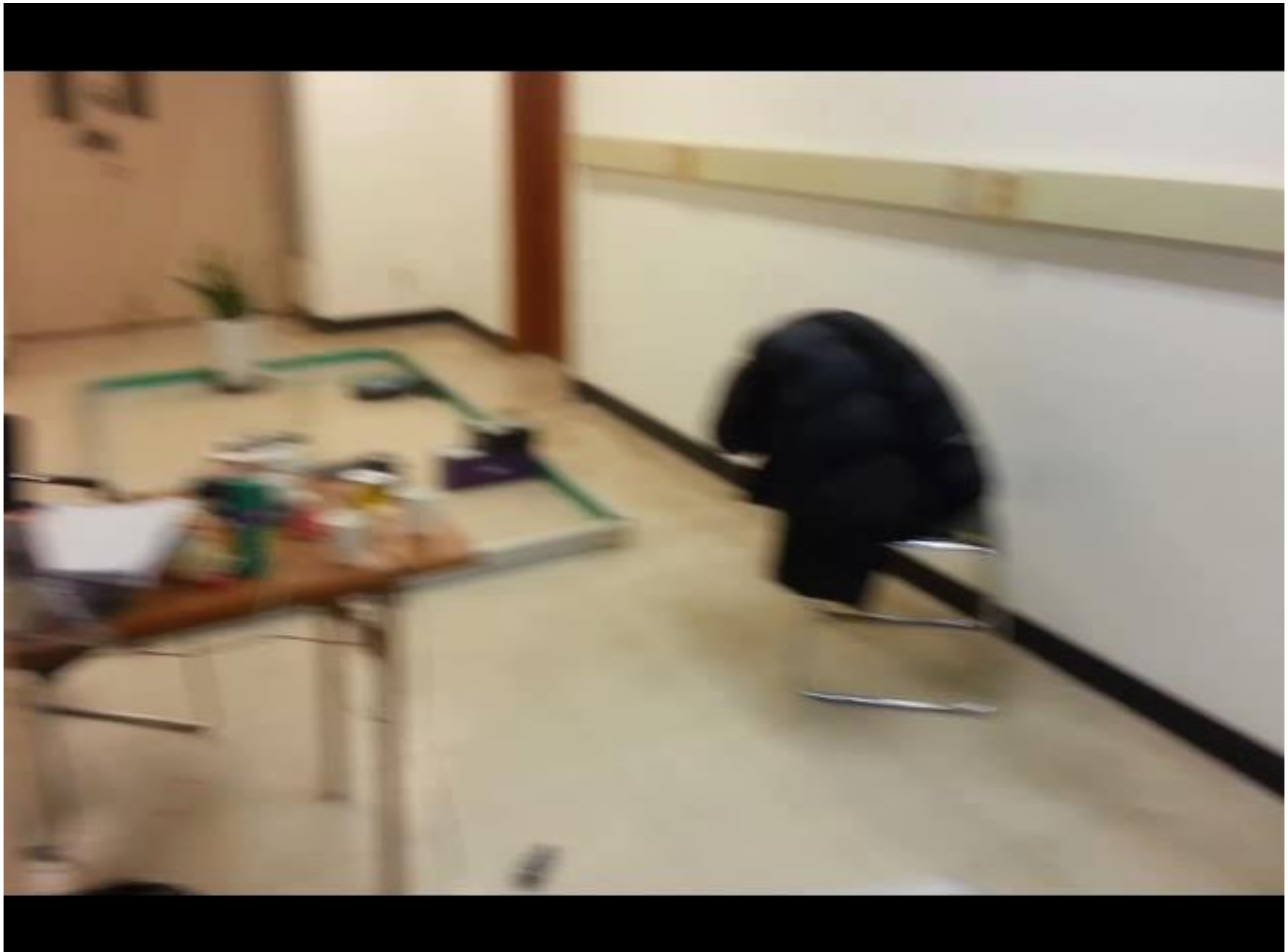
Double Threshold Strategy

Following Walls
: Much bumping

Double Threshold



Experimental Results



Can you Feel the Difference between Deterministic and Stochastic Approaches?

- Deterministic method tries to design a model
  - World will work as we thought
  - If World does not work as you think, you have to change your model
- Stochastic method think a world as non deterministic
  - A world will work PROBABLY.
  - A world will work as one of my thinking, **which depends on probabilistic distribution.**