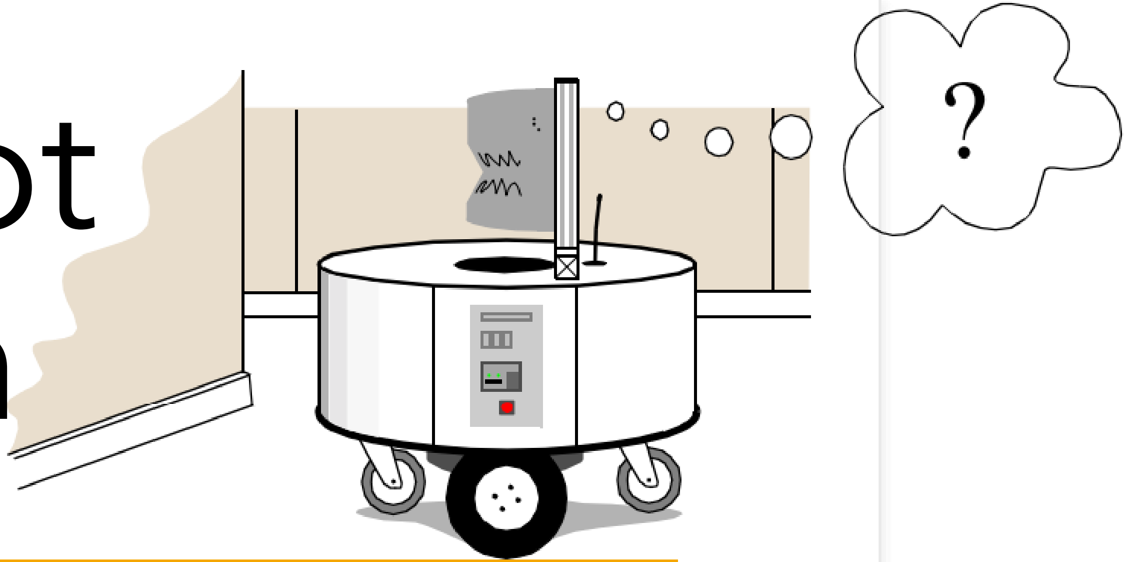
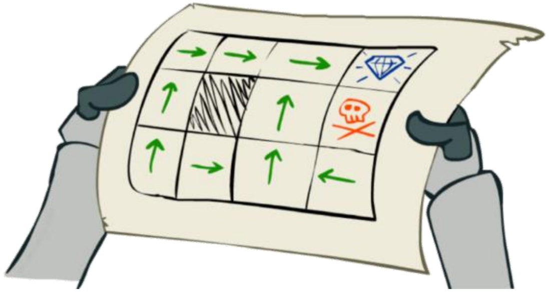
Mobile Robot Localization

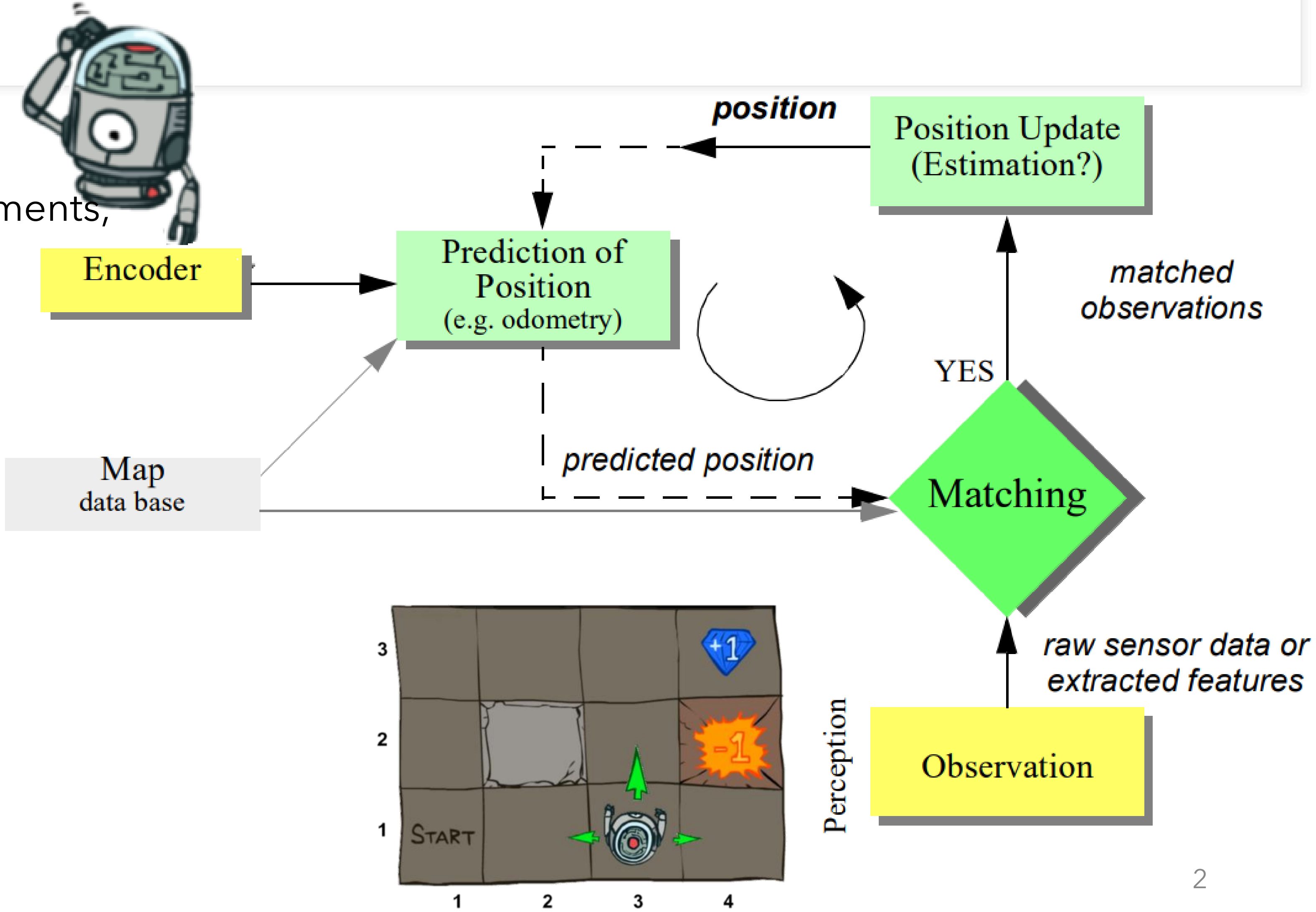


Dr. M. Shiple

Introduction

1. Sensor Noise
    1. acoustically reflective environments, multipath interference.
2. Sensor Aliasing





Introduction

1. **Sensor Noise**
    1. acoustically reflective environments, multipath interference.
    2. color CCD camera, illumination changes color saturation degree
2. **Sensor Aliasing** (the non-uniqueness of sensor readings,)
    1. Receive unique inputs in each unique local state.
    2. The amount of information is generally insufficient to identify the robot's position from a single-percept reading.
3. **Effector noise**
    1. Uncertainty about reaction from robot mobility.
    2. floor may be sloped, the wheels may slip, and a human may push the robot
    3. Uncertainty in the wheel diameter and in particular unequal wheel diameter
    4. Variation in the contact point of the wheel

Localization

- Means building a map, then identifying the robot's position relative to that map.

An error model for odometric position estimation

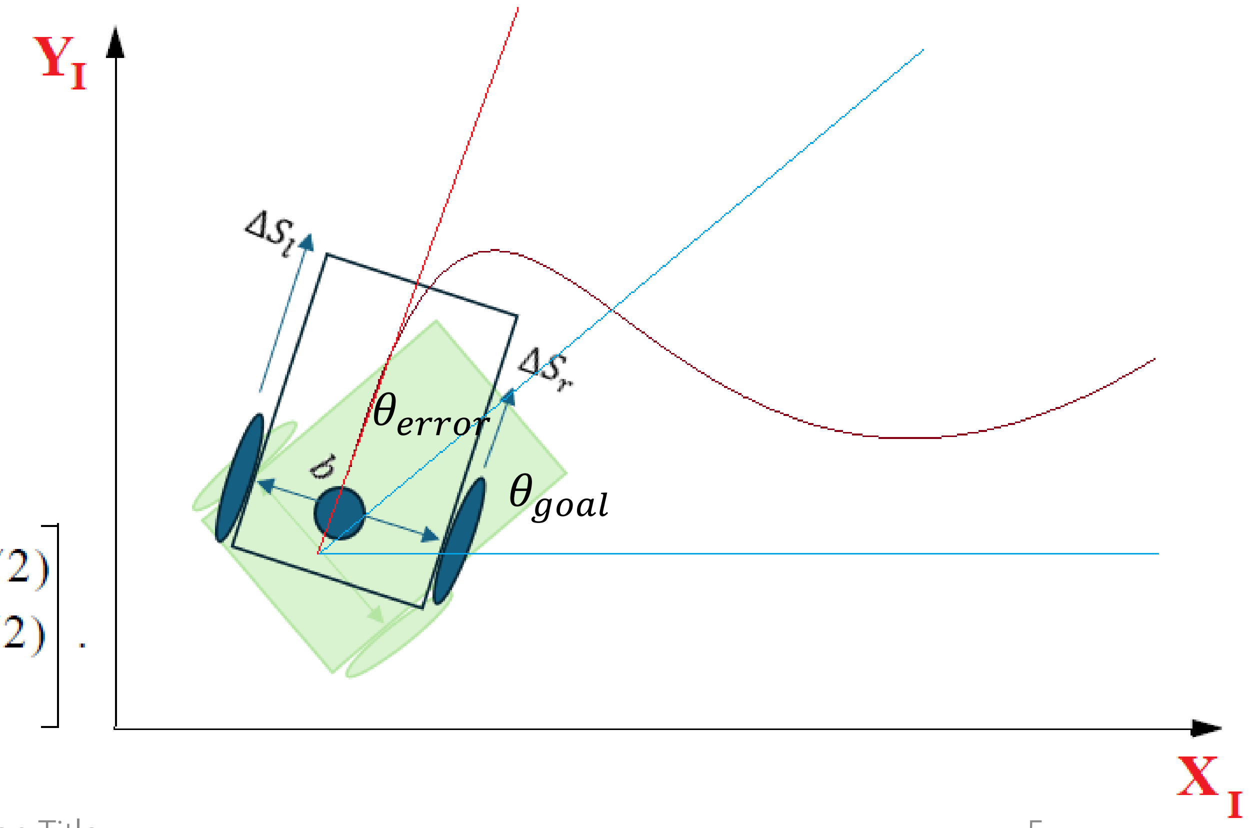
$$\Delta x = \Delta s \cos(\theta_{goal} + \frac{\theta_{error}}{2})$$

$$\Delta y = \Delta s \sin(\theta_{goal} + \frac{\theta_{error}}{2})$$

$$\Delta\theta = \frac{\Delta S_r + \Delta S_l}{b}$$

$$\Delta S = \frac{\Delta S_r + \Delta S_l}{2}$$

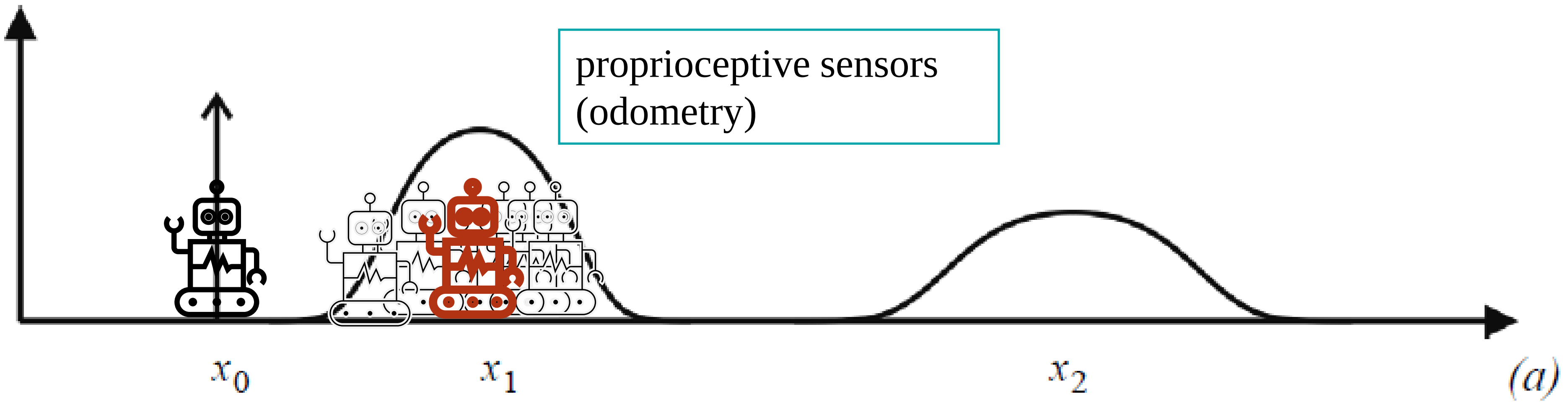
$$p' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = p + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} .$$



Probabilistic Map-Based Localization

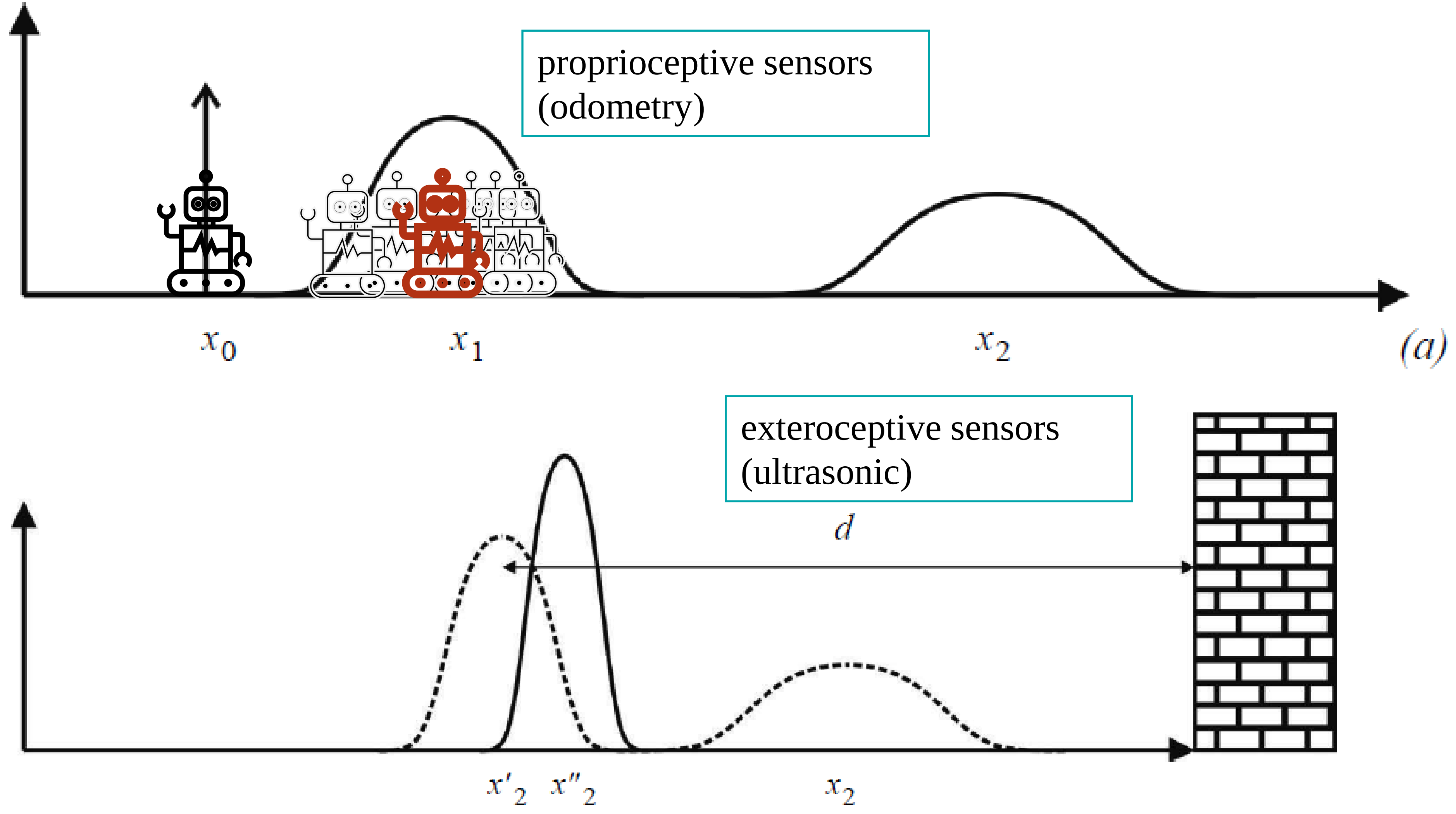
probabilistic robotics

The data coming from the robot sensors are affected by measurement errors, and therefore we can only compute the probability that the robot is in a given configuration



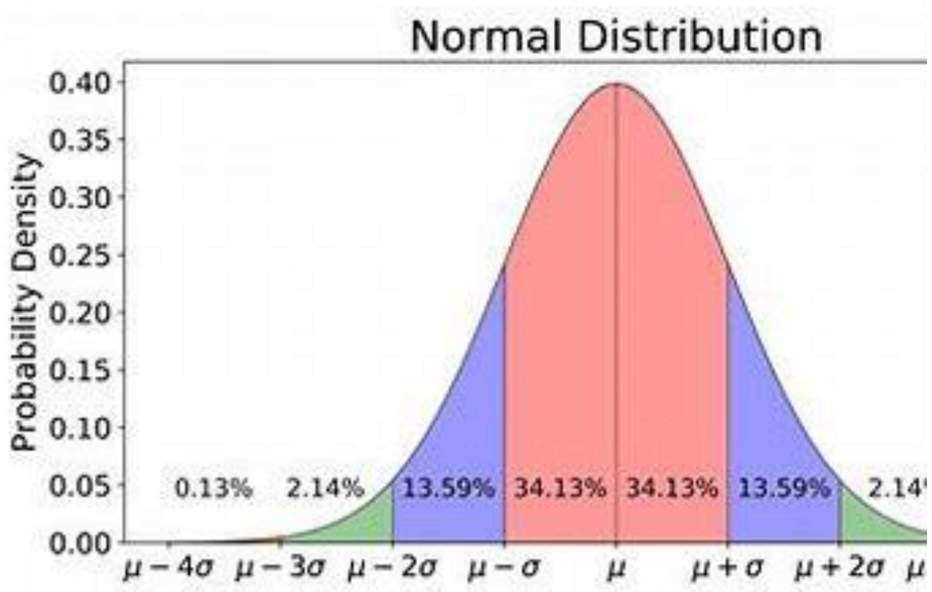
1. *Markov localization*
2. *Kalman filter localization*

probabilistic robotics



Markov Technique

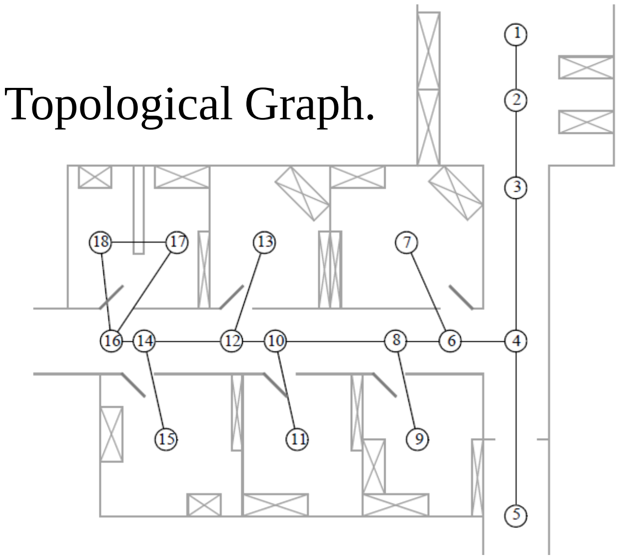
- Retains a and parameterization of the robot's belief about position with respect to the map.
- Starting from any *unknown* position.
- Robot can track multiple, completely disparate possible positions.



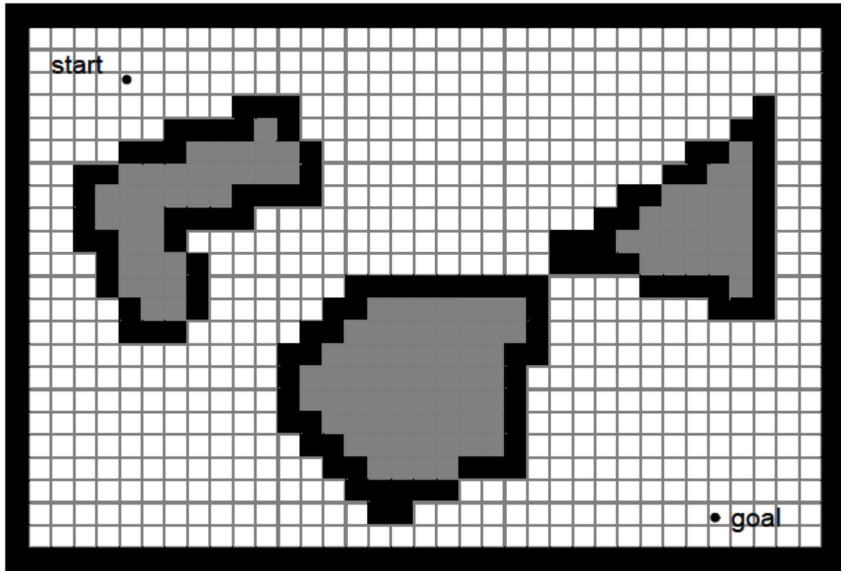
Markov Technique

- Requires a discrete representation of the space.
- There is a trade off between required memory and discrete map resolution.

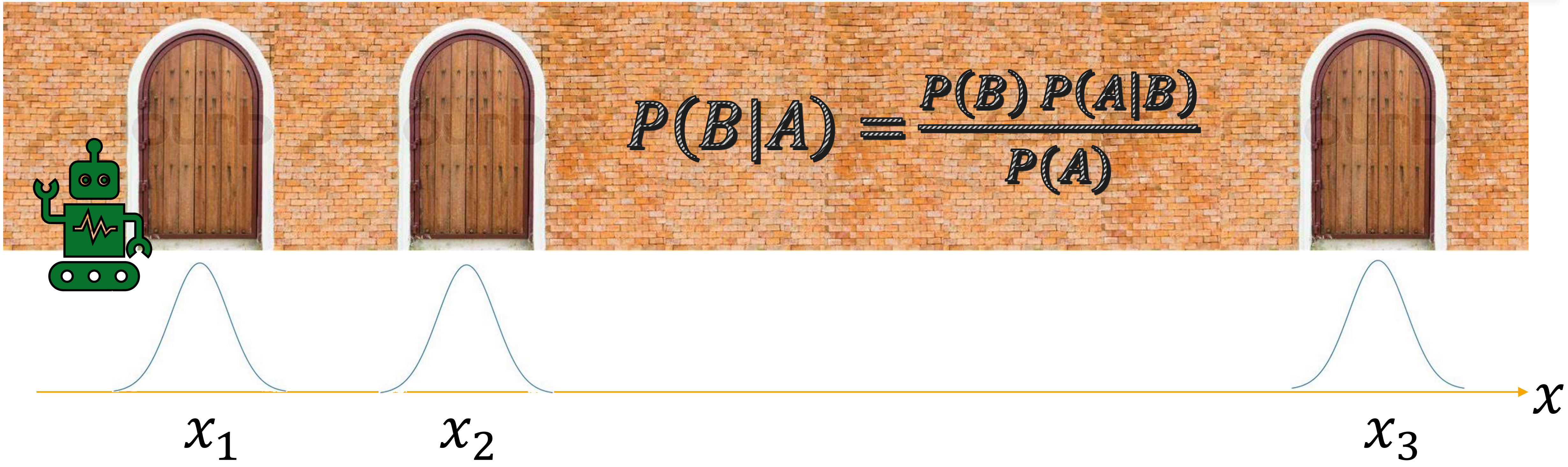
Topological Graph.



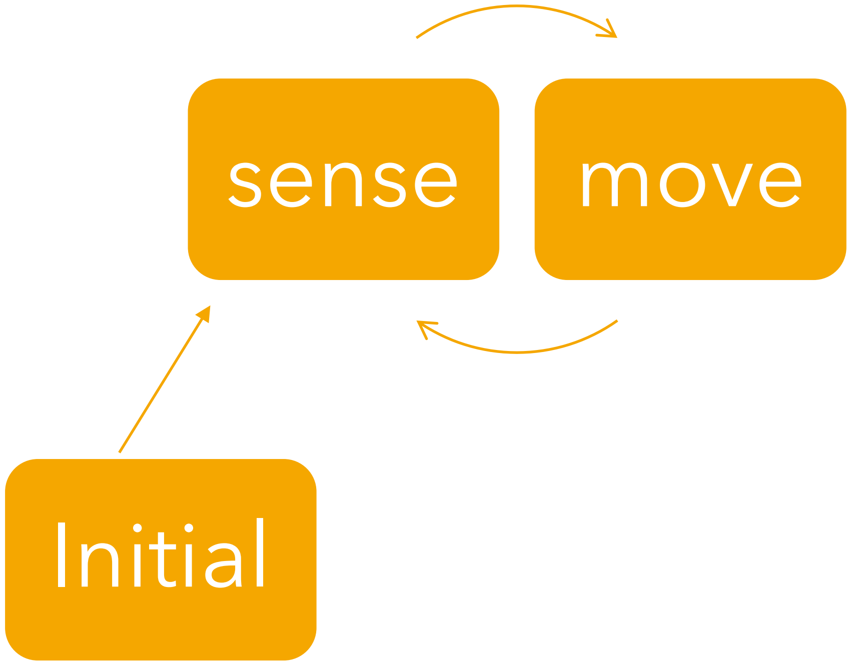
Geometric Grid

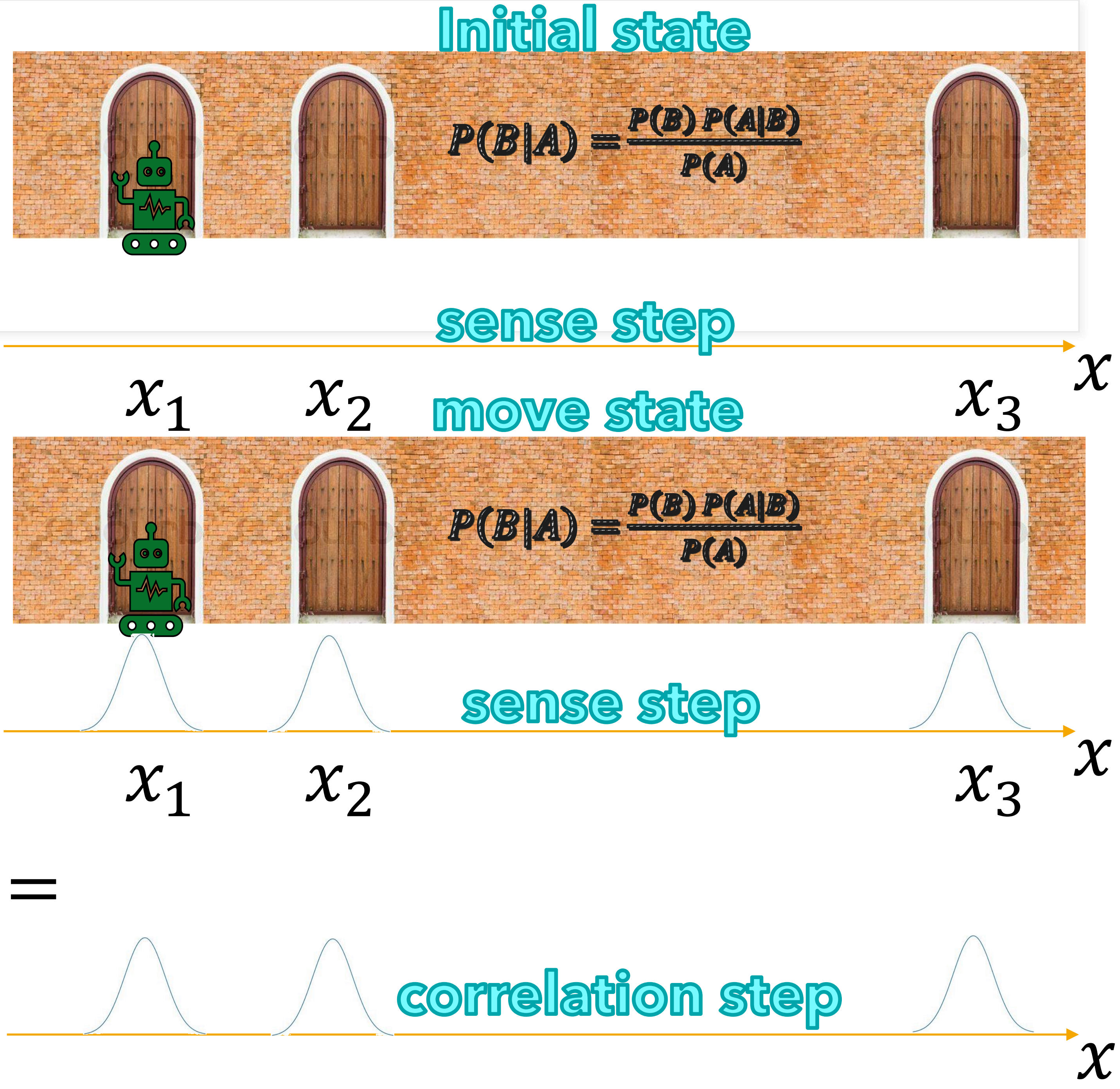


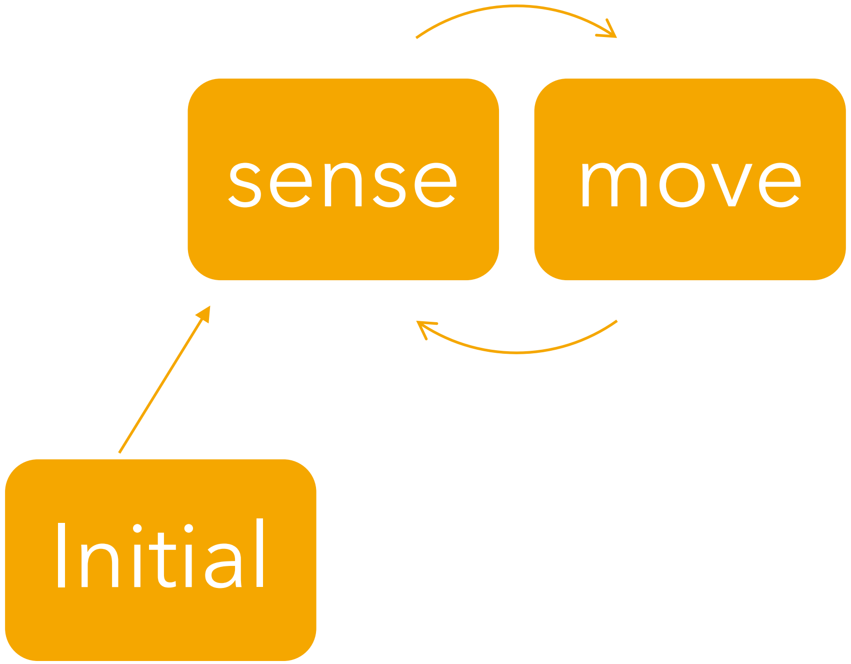
Perception rule



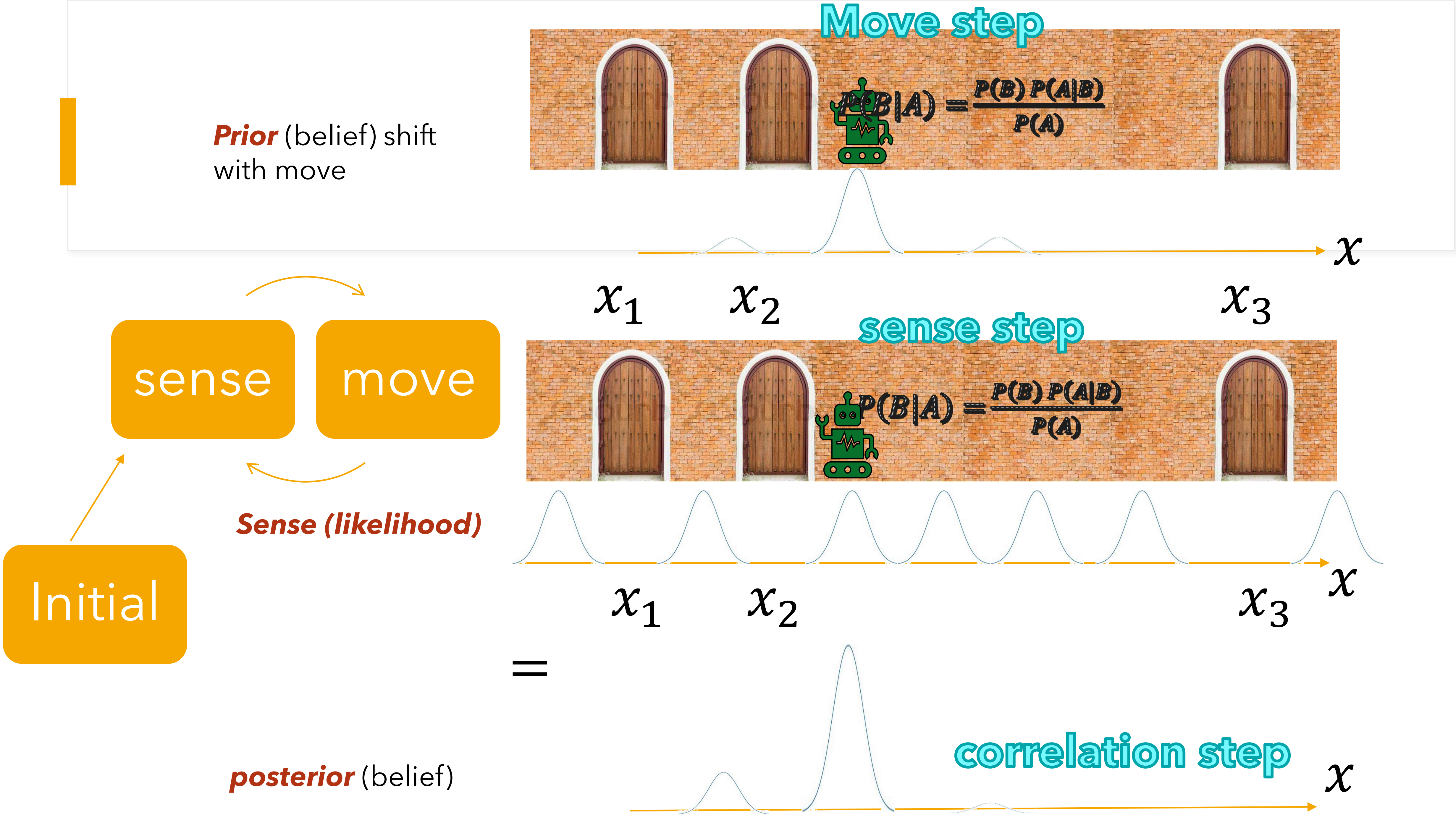
Perception rule



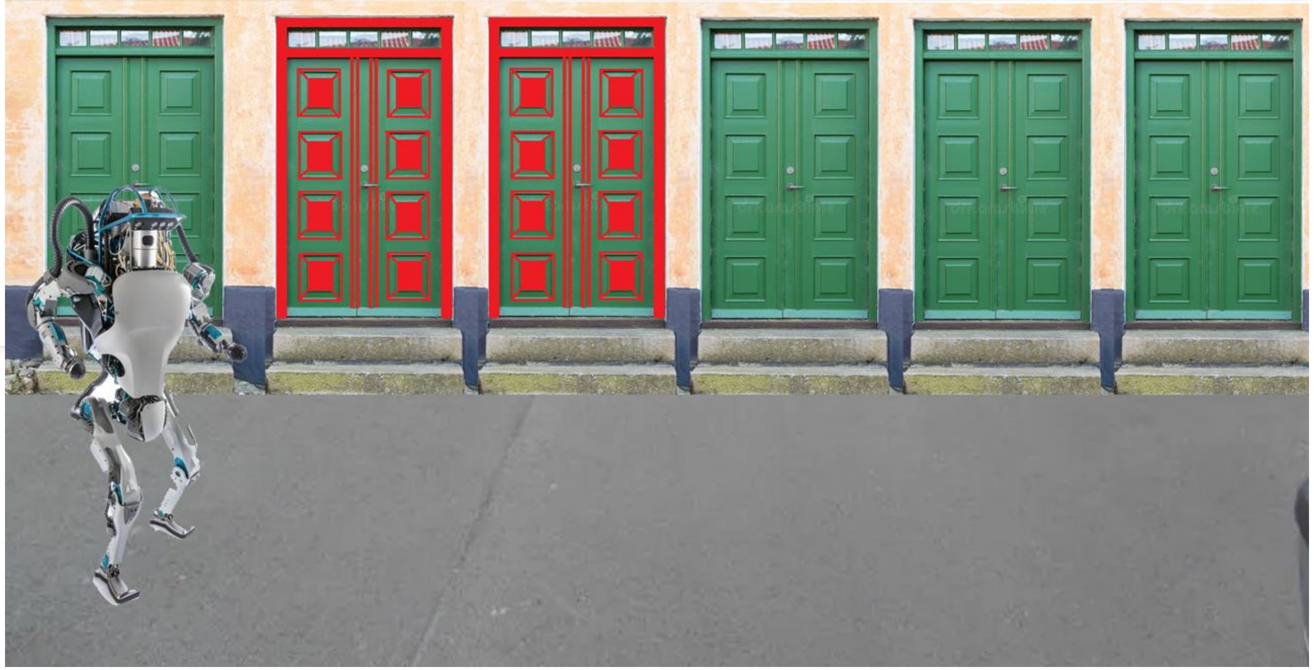



sense
move
Initial


Move step
$P(B|A) = \frac{P(B)\,P(A|B)}{P(A)}$
x_1 x_2 x_3 x
$P(B|A) = \frac{P(B)\,P(A|B)}{P(A)}$
sense step
x_1 x_2 x_3 x
=
correlation step
x


Move step
$P(B|A) = \frac{P(B)\,P(A|B)}{P(A)}$
Prior (belief) shift with move
x
x_1
x_2
x_3
sense step
sense
move
Initial
Sense (likelihood)
x
x_1
x_2
x_3
=
correlation step
x
posterior (belief)

Numerical example:



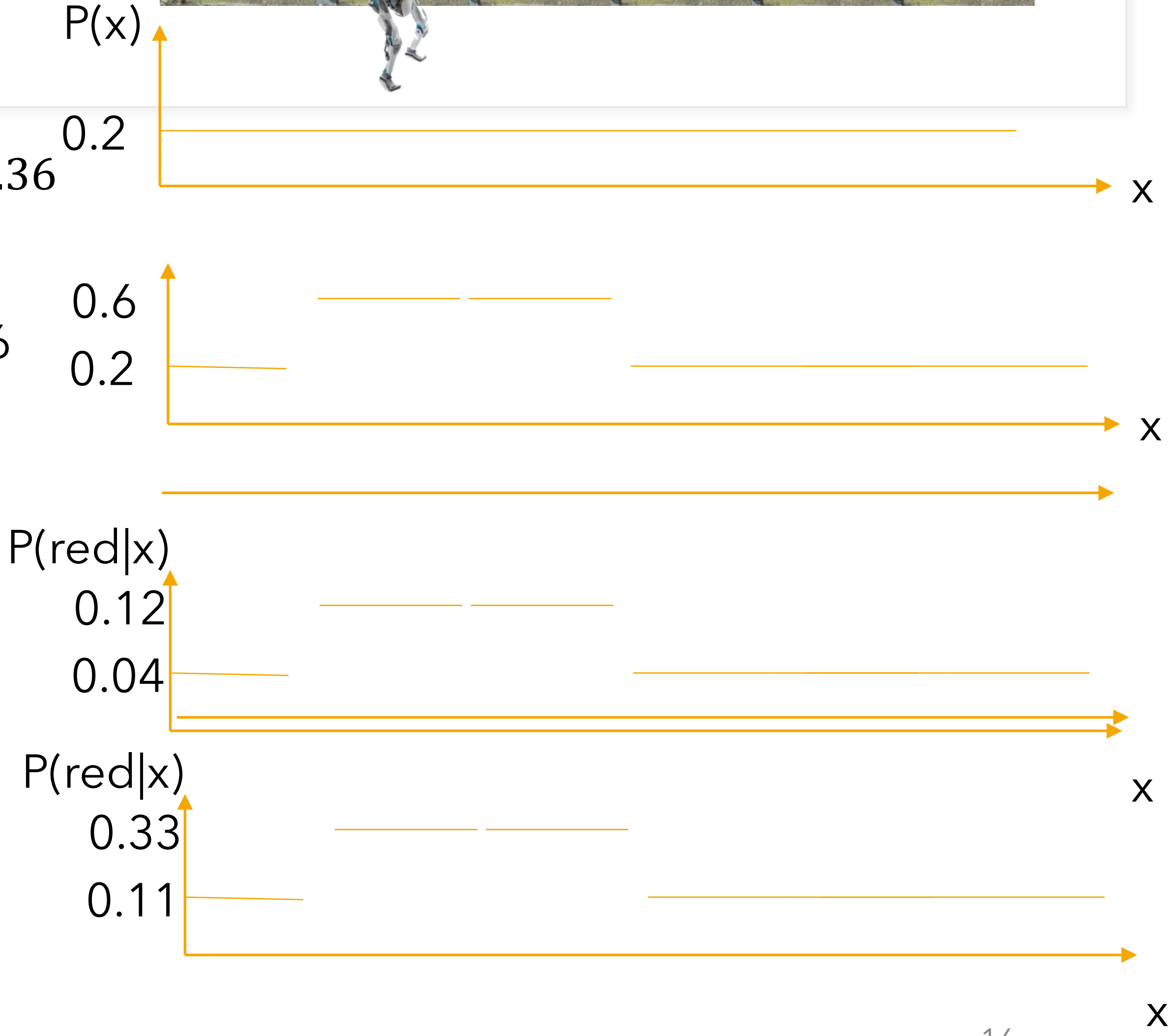
Assume that a robot is equipped with an uncertain color sensor that measures the color correctly with probability 0.6 and the other color with probability 0.2

If sensor read red = 0.6

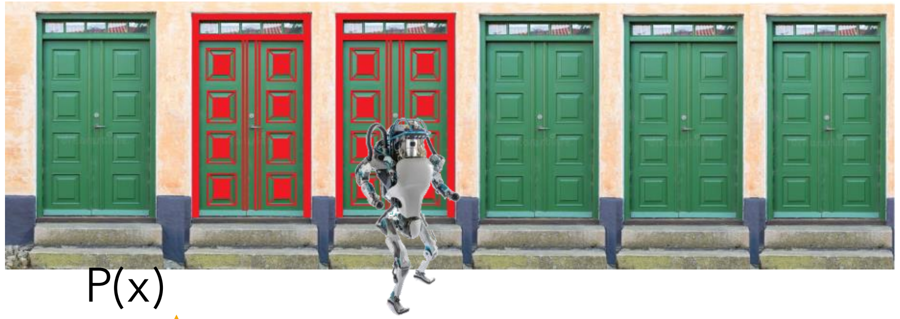
Numerical example:

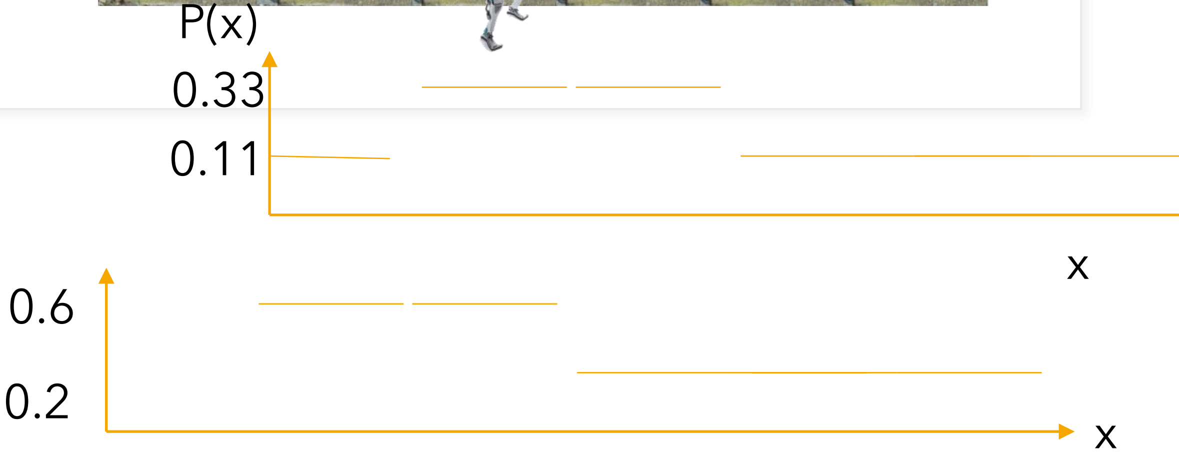
1. $\sum_1^5 p(x|red) = 0.04 + 0.12 + 0.12 + 0.04 + 0.04 + 0.04 = 0.36$
2. Normalization step : divide all P(x)/ $\sum_1^5 p(x|red)$

sensor read red = 0.6



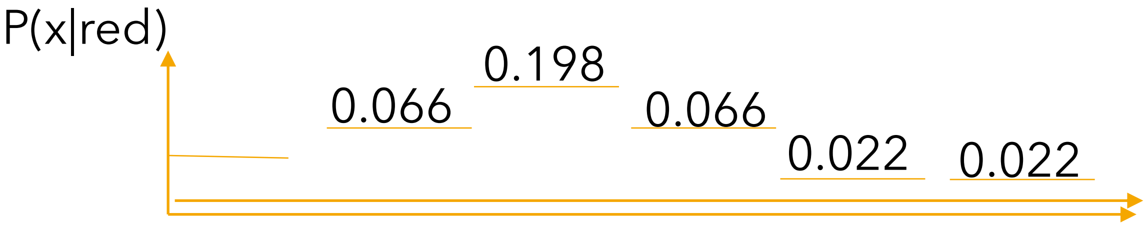
If move 1 cell:



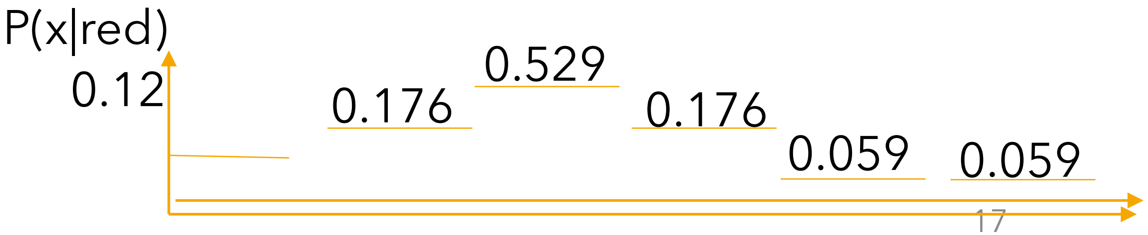


sensor read red = 0.6

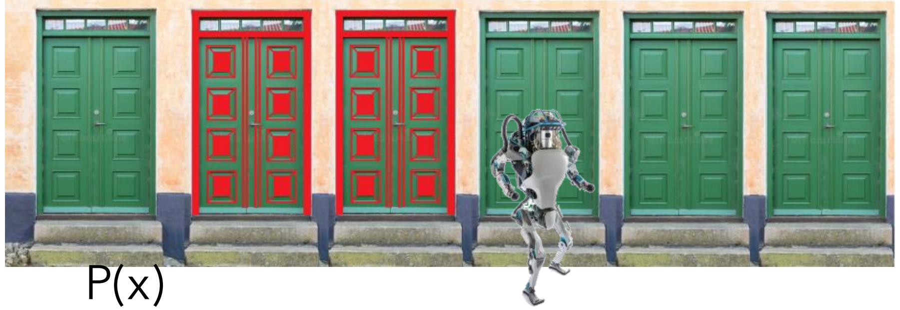
1. $\sum_1^5 p(x|red) = 0.04 + 0.12 + 0.12 + 0.04 + 0.04 + 0.04 = 0.36$
2. Normalization step : divide all P(x)/ $\sum_1^5 p(x|red)$

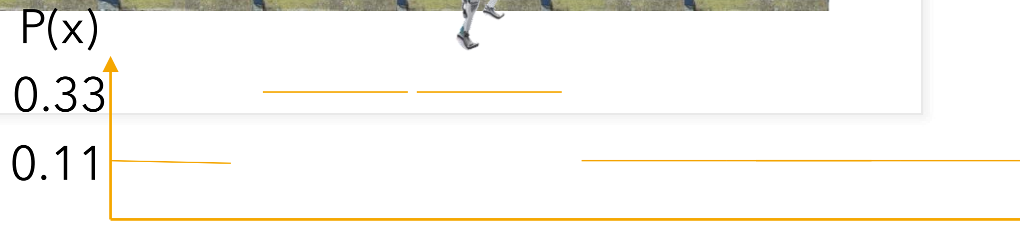


1. $\sum_1^5 p(x|red) = 2 * 0.066 + 0.198 + 2 * 0.022 = 0.374$
2. Normalization step : divide all P(x)/ $\sum_1^5 p(x|red)$



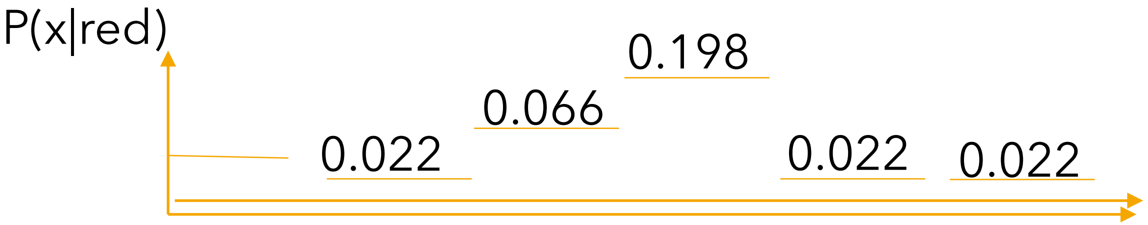
Move suddenly 2 cells:



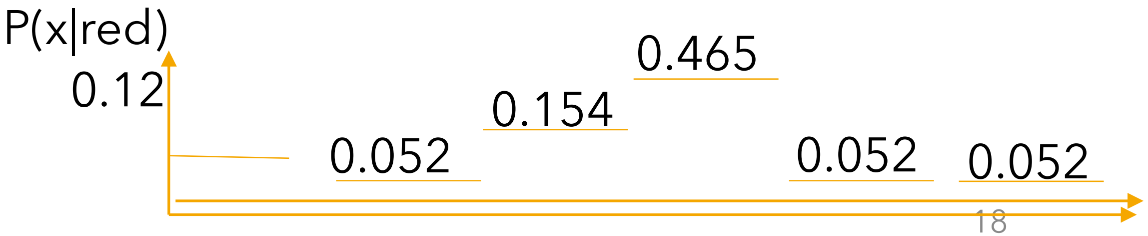


1. $\sum_1^5 p(x|red) = 0.04 + 0.12 + 0.12 + 0.04 + 0.04 + 0.04 = 0.36$
2. Normalization step : divide all P(x)/ $\sum_1^5 p(x|red)$

sensor read green = 0.6



1. $\sum_1^5 p(x|red) = 1 * 0.066 + 0.198 + 3 * 0.022 = 0.426$
2. Normalization step : divide all P(x)/ $\sum_1^5 p(x|red)$



Algorithm Pseudo Code

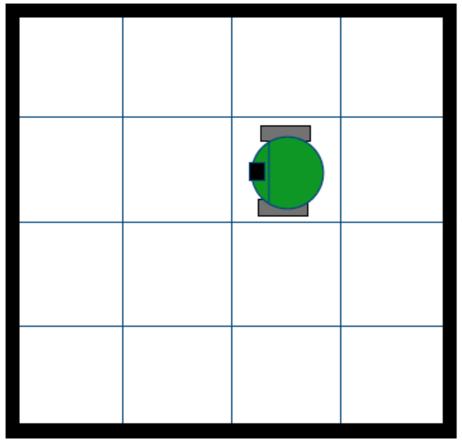
```
for i = 1:n
    P(x_i) = 1/n

while (true)
    o = getOdometryMeasurements
    z = getRangeMeasurements
    for i = 1:n
        P(x_i') = predictionStep(P(x_j), o)
    for i = 1:n
        P(x_i) = correctionStep(P(x_j'), z)
```

summary

- PREDICTION Step (belief) : Updating the belief state P(x) (proprioceptive sensors)
- Correction step : $p(x|red)$ (exteroceptive sensors)
- Update step: $P(x)\ P(x|red)$.
- Normalization step = P(x)/ $\sum_1^5 p(x|red)$

2D Grid map example

- *Consider a robot equipped with encoders and a perfect compass moving in a square room that is discretized into a map of 16 cells:*



.02	.05	.05	.05
.02	.05	.18	.05
.05	.05	.18	.05
.05	.05	.05	.05

Initial state

1/16	1/16	1/16	1/16
1/16	1/16	1/16	1/16
1/16	1/16	1/16	1/16
1/16	1/16	1/16	1/16

Odometry model

.00	.00	.00
.00	.00	1.0
.00	.00	.00

$(\Delta x, \Delta y)$

$(-1.0, 0.0)$

0.0	0.2	0.0
0.0	0.5	0.1
0.0	0.2	0.0

Case 1:

Initial state

0.02	0.05	0.05	0.05
0.02	0.05	0.18	0.05
0.05	0.05	0.18	0.05
0.05	0.05	0.05	0.05

→

Move step
Sense step

0.0 0.05	0.2 0.05	0.0 0.05	0.02
0.0 0.05	0.5 0.18	0.1 0.05	0.02
0.0 0.05	0.2 0.18	0.0 0.05	0.05
0.05	0.05	0.05	0.05

→

Correlation step

0.0	0.01	0.0	0.0
0.0	0.09	0.005	0.0
0.0	0.036	0.0	0.0
0.0	0.0	0.0	0.0

$$\sum_{1}^{5} p(x|red) = 0.01 + 0.09 + 0.36 + 0.005 = 0.141$$

Case 2:

Initial state

0.02	0.05	0.05	0.05
0.02	0.05	0.18	0.05
0.05	0.05	0.18	0.05
0.05	0.05	0.05	0.05

→

Move step
Sense step

0.05	0.05	0.05	0.02
0.0 0.05	0.2 0.18	0.0 0.05	0.02
0.0 0.05	0.5 0.18	0.1 0.05	0.05
0.0 0.05	0.2 0.05	0.0 0.05	0.05

→

Correlation step

0.0	0.0	0.0	0.0
0.0	0.036	0.0	0.0
0.0	0.09	0.005	0.0
0.0	0.01	0.0	0.0

$$\sum_{1}^{5} p(x|red) = 0.01 + 0.09 + 0.36 + 0.005 = 0.141$$

overall

0.0	0.01	0.0	0.0
0.0	0.09	0.005	0.0
0.0	0.036	0.0	0.0
0.0	0.0	0.0	0.0

+

0.0	0.0	0.0	0.0
0.0	0.036	0.0	0.0
0.0	0.09	0.005	0.0
0.0	0.01	0.0	0.0

=

0.0	0.01	0.0	0.0
0.0	0.045	0.005	0.0
0.0	0.045	0.005	0.0
0.0	0.01	0.0	0.0

Thanks