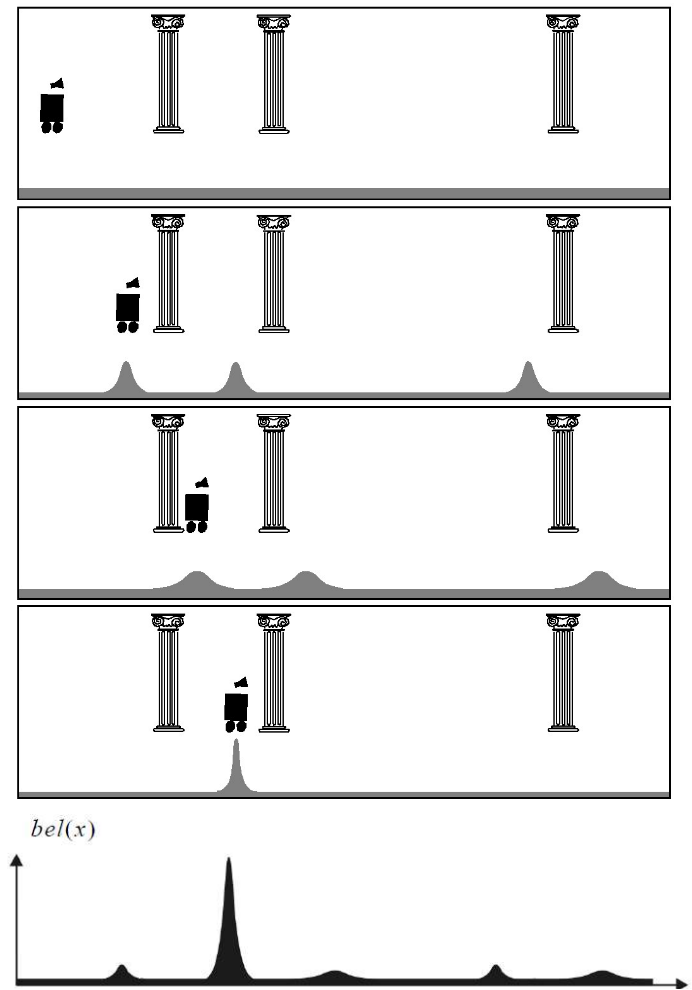
Recall: The Probabilistic Localization Example

- The robot is placed somewhere in the environment but is not told its location
- The robot queries its sensors and finds it is next to a pillar
- The robot moves one meter forward. To account for inherent noise in robot motion the new belief is smoother
- The robot queries its sensors and again it finds itself next to a pillar
- Finally, it updates its belief by combining this information with its previous belief



The two update steps in robot localization

1. ACTION (or prediction) update: the robot moves and estimates its position through its proprioceptive sensors. During this step, the robot uncertainty grows. This step uses the Total Probability formula to update belief

$$\overline{bel}(x_t) = p(x_t \mid u_t, x_{t-1}) * bel(x_{t-1})$$

$$p(x) = \int_y p(x|y)p(y)dy$$

2. PERCEPTION (or measurement) update: the robot makes an observation using its exteroceptive sensors and correct its position by opportunely combining its belief before the observation with the probability of making exactly that observation. During this step, the robot uncertainty shrinks. This step uses the Bayes Rule to update belief.

$$bel(x_t) = \eta \cdot p(z_t \mid x_t) \cdot \overline{bel}(x_t)$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

Markov versus Kalman localization

- Two approaches exist to represent the probability distribution and to compute the Total Probability and Bayes Rule during the Action and Perception phases
- Markov approach:
  - The configuration space is divided into many cells. Each cell contains the probability of the robot to be in that cell.
  - The probability distribution of the sensors model is also discrete.
  - During Action and Perception, all the cells are updated. Therefore, the computational cost is very high.
- Kalman approach:
  - The probability distribution of both the robot configuration and the sensor model is assumed to be continuous and Gaussian!
  - Need only to update mean value μ and covariance Σ. Therefore the computational cost is very low!

Markov localization example: Probability After Sense

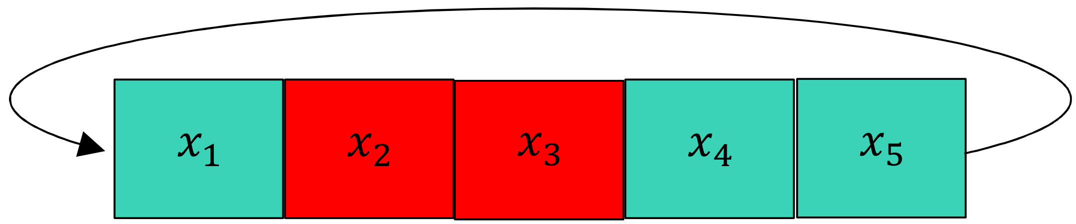
- Consider a map with five cells along a line, cells x_2 and x_3 are red (R) and cells x_1, x_4, and x_5 are green (G).

x_1	x_2	x_3	x_4	x_5

- Initial belief distribution: 0.2 0.2 0.2 0.2 0.2
- Sense: Z = red
  - P(Z|R) = 0.8 and
  - P(Z|G) = 0.2
- Belief after sense: .04 .16 .16 .04 .04 $\Sigma = .44$
- Posterior distribution: 1/11 4/11 4/11 1/11 1/11 $\Sigma = 1$

Exact Motion

- Robot moves along the cells cyclically and **accurately**



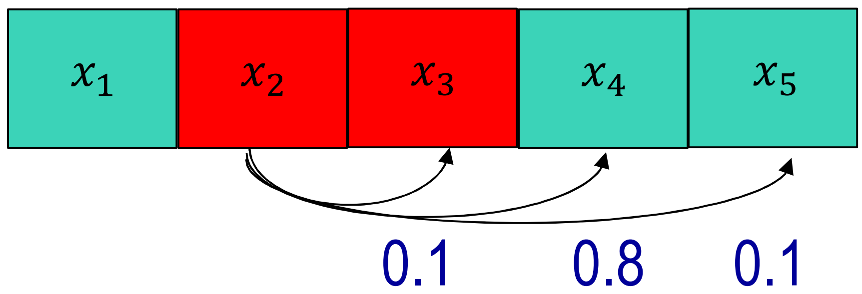
- Belief distribution: 1/11 4/11 4/11 1/11 1/11

- Belief after move: 1/11 4/11 4/11 1/11 1/11

Inexact Motion

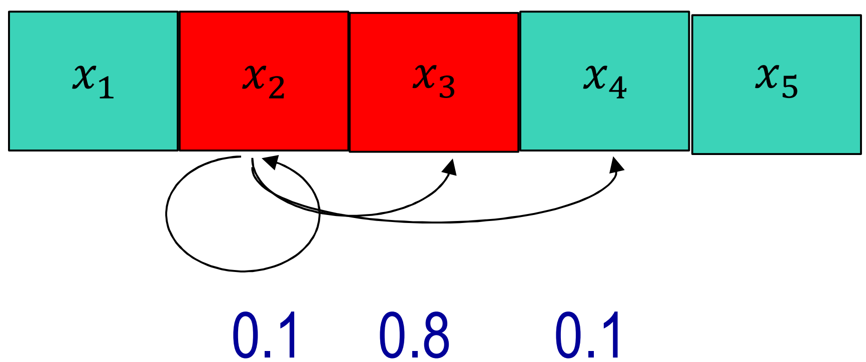
- Robot moves along the cells cyclically and **inaccurately**.

$$P(x_{i+2}|x_i) = 0.8$$
$$P(x_{i+1}|x_i) = 0.1$$
$$P(x_{i+3}|x_i) = 0.1$$



- U = 2

$$P(x_{i+1}|x_i) = 0.8$$
$$P(x_i|x_i) = 0.1$$
$$P(x_{i+2}|x_i) = 0.1$$

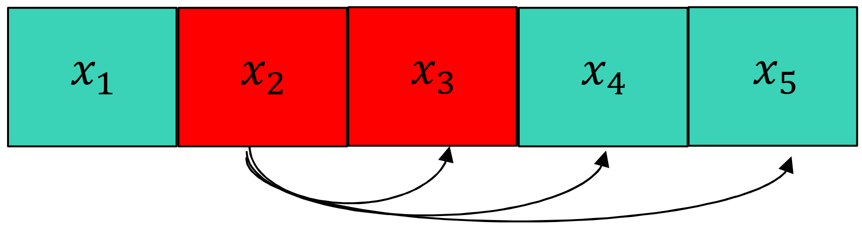


- U = 1

Inexact Motion I

- Robot moves along the cells cyclically and **inaccurately**.

$$P(x_{i+2}|x_i) = 0.8$$
$$P(x_{i+1}|x_i) = 0.1$$
$$P(x_{i+3}|x_i) = 0.1$$



- U = 2   0.1   0.8   0.1

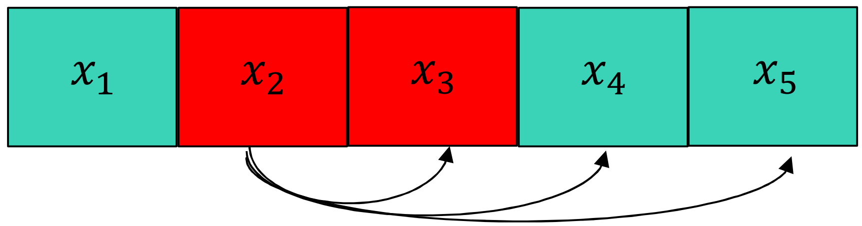
- Belief before move: 0   1   0   0   0

- Belief after move: 0   0   0.1   0.8   0.1

Inexact Motion II

- Robot moves along the cells cyclically and **inaccurately**.

$$P(x_{i+2}|x_i) = 0.8$$
$$P(x_{i+1}|x_i) = 0.1$$
$$P(x_{i+3}|x_i) = 0.1$$



- U = 2   0.1   0.8   0.1

- Belief before move: 0   .5   0   .5   0

- Belief after move:

.5x.8   5x0.1   .5x.1   5x8   5x.1+.5x1

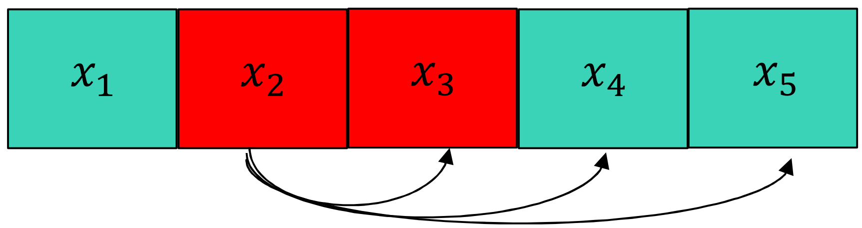
.4   .05   .05   .4   .1

$\Sigma = 1$

Inexact Motion III

- Robot moves along the cells cyclically and **inaccurately**.

$$P(x_{i+2}|x_i) = 0.8$$
$$P(x_{i+1}|x_i) = 0.1$$
$$P(x_{i+3}|x_i) = 0.1$$



- U = 2   0.1   0.8   0.1

- Belief before move: 0.2   0.2   0.2   0.2   0.2

- Belief after move:

.2x.1+0.2x.8+0.2x0.1

= .2(.1+.8+.1)

= .2

Localization Summary

- Sense and Move
  - Sense: Gain information (belief)
  - Move: lose information (belief)
- Belief: Probability
- Belief update
  - Sense: Product followed by normalization
  - Move: Convolution (addition)

Bayes' Rule

- Measurement update: x_i = *ith* grid cell, Z = measurement

Measured probability

Posterior probability

Priori probability

- Bayes' Rule

$$P(x_i|Z) = \frac{P(Z|x_i)P(x_i)}{P(Z)}$$

$$P(Z) = \sum_i P(Z|x_i)P(x_i)$$

Bayes' Rule: Steps to compute $P(x_i|Z)$

- Measurements: x_i = *ith* grid cell, Z = measurement

- Bayes' Rule
$$P(x_i|Z) = \frac{P(Z|x_i)P(x_i)}{P(Z)}$$
$$P(Z) = \sum_i P(Z|x_i)P(x_i)$$

- Steps to compute $P(x_i|Z)$

$$\bar{P}(x_i|Z) = P(Z|x_i)P(x_i)$$

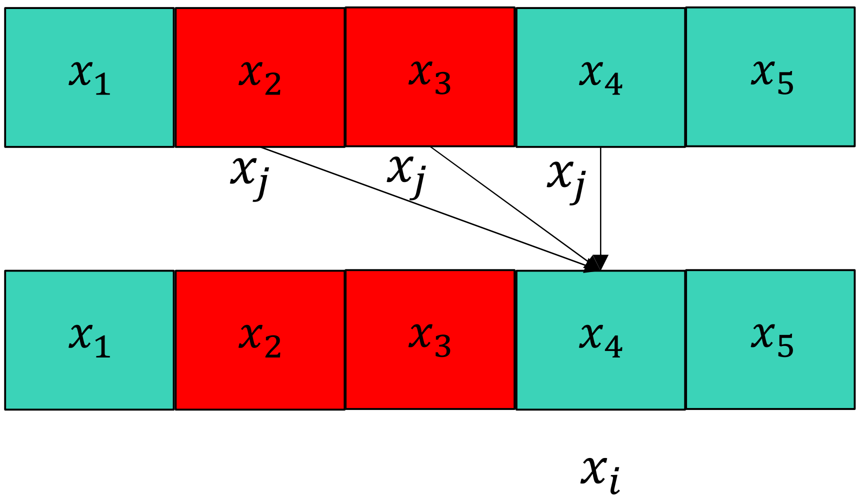
$$\alpha = \sum \bar{P}(x_i|Z)$$

$$P(x_i|Z) = \frac{\bar{P}(x_i|Z)}{\alpha}$$

Theorem of Total Probability

- Motion Update: *i* = *ith* grid cell, t = time,

$$P(x_i^t) = \sum_j P(x_j^{t-1}) P(x_i | x_j)$$



- Theorem of Total Probability

$$P(A) = \sum_B P(A|B) P(B)$$

Summary

- Localization
- Markov Localization
- Probabilities
- Bayes rule for measurement update

$$\bar{P}(x_i|Z) = P(Z|x_i)P(x_i)$$

$$\alpha = \sum \bar{P}(x_i|Z)$$

$$P(x_i|Z) = \frac{\bar{P}(x_i|Z)}{\alpha}$$

- Total probability for motion update

$$P\left(x_i^t\right) = \sum_j P\left(x_j^{t-1}\right)P(x_i|x_j)$$

Probability Examples I

- Let P(X) = 0.2.

  P(¬X) = $1 - P(X) = 1 - 0.2 = 0.8$

- Let P(X) = 0.2, P(Y) = 0.2, and X and Y are independent

  P(X,Y) = $P(X)P(Y) = 0.2 \times 0.2 = 0.04$

- Let P(X) = 0.2, P(Y|X) = 0.6, and P(Y| ¬X) = 0.6

  P(Y) = $P(Y|X)P(X) + P(Y|\neg X)P(\neg X)$

  $= 0.6 \times 0.2 + 0.6 \times (1 - 0.2)$

  $= 0.12 + 0.48 = 0.6$

Probability Examples II

- Cancer test
  - P(C) = 0.001
  - P(¬C) = 0.999
  - P(POS | C) = 0.8
  - P(POS | ¬C) = 0.1

$$\bar{P}(x_i|Z) = P(Z|x_i)P(x_i)$$

$$\alpha = \sum \bar{P}(x_i|Z)$$

$$P(x_i|Z) = \frac{\bar{P}(x_i|Z)}{\alpha}$$

- Probability of Cancer when having a positive cancer test
  - P(C | POS) =

$$P(C|POS) = \frac{P(POS|C)P(C)}{P(POS)} = \frac{0.0008}{0.1007} \simeq 0.008$$

$$P(POS) = P(POS|C)P(C) + P(POS|\neg C)P(\neg C)$$

$0.8 \times 0.001 \quad 0.1 \times 0.999$

$0.0008 + 0.0999 = 0.1007$

Probability Examples III

- Two Coins
  - Fair coin P(H | F) = 0.5
  - Unfair coin P(H | ¬F) = 0.1
  - P(F) = 0.5
- Flit a coin and observe H
  - P(F | H) =

$$\bar{P}(x_i|Z) = P(Z|x_i)P(x_i)$$

$$\alpha = \sum \bar{P}(x_i|Z)$$

$$P(x_i|Z) = \frac{\bar{P}(x_i|Z)}{\alpha}$$

$$P(F|H) = \frac{P(H|F)P(F)}{P(H)} = \frac{0.25}{0.3} = \ldots$$

$$P(H) = P(H|F)P(F) + P(H|\neg F)P(\neg F)$$

$0.5 \times 0.5 \quad 0.1 \times 0.5$

$0.25 + 0.05 = 0.3$

Localization with using Python: Measurement Update

- Python code

Sense: Z = red
P(Z|R) = 0.8 and
P(Z|G) = 0.2

```
p=[0.2, 0.2, 0.2, 0.2, 0.2]
world=['green', 'red', 'red', 'green', 'green']
Z = 'red'
pHit = 0.8
pMiss = 0.2

def sense(p, Z):
    q=[]
    sum = 0.0
    for i in range(len(p)):
        hit = (Z == world[i])
        q.append(p[i] * (hit * pHit + (1-hit) * pMiss))
        sum += q[i]

    for i in range(len(q)):
        q[i] = q[i]/sum

    return q

print (sense(p,Z))
```

- Result

```
[0.0909090909090909, 0.36363636363636365, 0.36363636363636365, 0.0909090909090909,
 0.0909090909090909]
>>>
```