University of Burgundy

Masters in Computer Vision and Robotics

PROBABILISTICS ROBOTICS

Report for LAB – 1

2D GRID LOCALIZATION

by

Vamshi Kodipaka

Supervisior:

PROF. DESIRE SIDIBE

2D GRID LOCALIZATION

Localization can be seen as a problem of coordinate transformation. Maps are described in a global coordinate system, which is independent of a robot's pose.

1. What is Localization?

Localization: Localization is the process of establishing correspondence between the map coordinate system and the robot's local coordinate system.

Environment: Environment is a physical world which is inherently unpredictable. The environment, or world, of a robot is a dynamical system that possesses internal state.

State: A state can be any random physical/abstract parameter of a robot for a given instant of time. As a consequence, the robot maintains an internal belief with regards to the state of its environment (here we consider two states sense and position: that is to measure via sense and then movement via position). Environments are characterized by state.

Sense: The robot can acquire information about its environment using its sensors. However, sensors are noisy, and there are usually many things that cannot be sensed directly. (We are expected to make our robot sense to Red and Green colors).

Movement: Movement is consider as the degree of freedom given to the robot to move in any direction possible. (Here we consider horizontal movement and vertical movement, because we are expected our robot to move in 2-dimensions as take our environment to be a 2D grid).

Uncertainties: Uncertainties are the quantifiable inaccurate measures for the state parameters. We can have uncertainties due to Environment Effects, Sensors Malfunction, Robot Actuators Malfunction, Design of Improper Model without considering all possible physical states/processes and due to Approximation of Complex Calculations with the specified time frame when signal/action call is initiated. (Here we consider only Sensor uncertainty and Movement uncertainty and assuming that robots is 100% accurate to all other state parameters).

Aim of Localization: Optimizing Robot Navigation by Sensing and Motion control.

Applications: Self Driving Cars, Mobile Robots

2. Markov localization

Markov Localization uses an explicitly specified probability distribution across all possible robots senses and positions. In this case, the robot's belief state is usually represented as separate probability assignments for every possible robot pose in its map. The action update and perception update processes must update the probability of every cell.

Note:

1. Initially, considering the probability distribution all over the 2D grid world is uniform.
2. No. of Red colored grids may or may not be equal to No. of Green colored grids in the world.

Principle:

1. In general, Markov's Localization is based on sense and move; repeat mechanism for given no. of senses and no. of movements.

2. Markov's chain uses Bayesian distribution of probabilities to calculate uncertainties. (see fig.1)

Working:

1. Here, firstly, the robot is made localized to the environment around in the world by sensor (sensing). Then the sensed measurement output is used for Motion.

2. After Motion then the robot is made to localize again by sensing from that position in the map or world.

3. Then above steps are repeated on after the other to navigate by localizing.

Constraints for degree of movement:

Here, we design the robot motion as bi-directional i.e; horizontally to right or left and vertically to up or down (either vertical or horizontal move at a time; but not both at a time). Circular movement possible.

Simple Marchov's Chain:

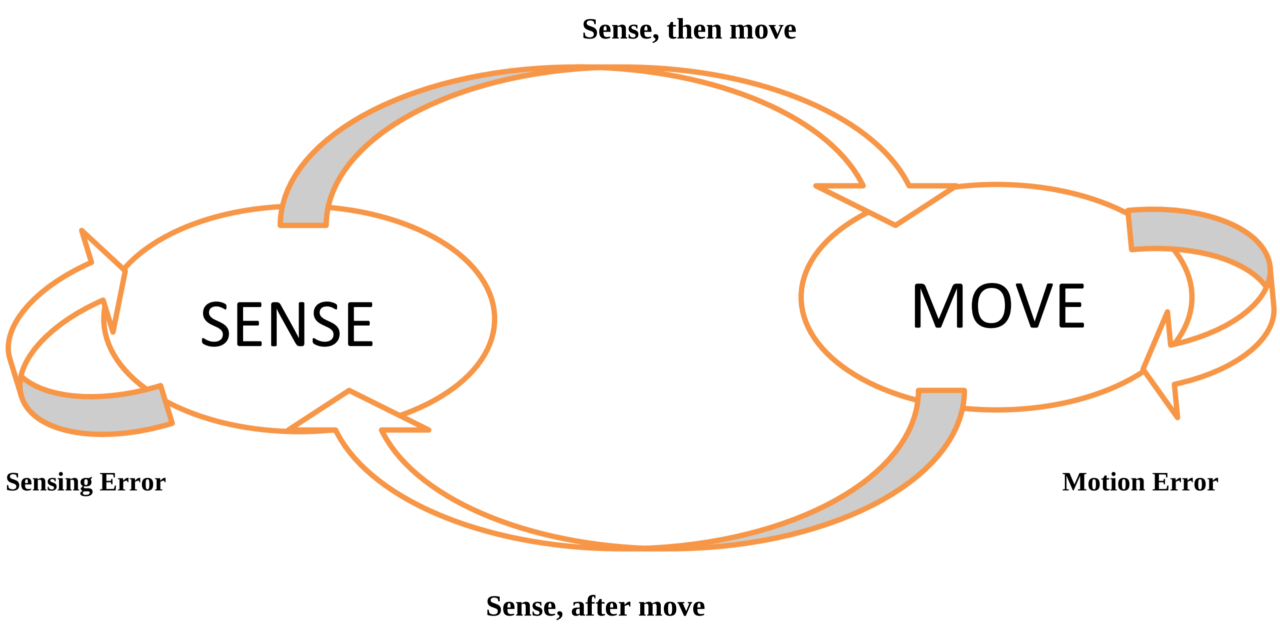


Fig.1 : General Marchov's Localization chain for Real World Application

From above fig. we have 4 transitions:

1. Sense then move.

2. Multiple Senses (to minimize the sensing error) and Move

   Sensing Error: Multiple measurements can be done for avoiding this uncertainties (considering PHit or PMiss for every sense execution)

   Considering for every sense execution,

   PHit(Probability of Sensor making correct measurement t)=0.6
   PMiss(Probability of Sensor making wrong measurement)=0.4

3. **Motion Error:** This is due to several parameters of actuators, world etc. So, we chose the appropriate actuators and environment to make this error negligible

   Considering motion error, for every move:

   PCorrect(Probability of robot making correct move/to the correct position)=0.8
   POvershoot(Probability of robot making wrong move due to actuator lead) = 0.1
   PUndershoot(Probability of robot making wrong move due to actuator lag) = 0.1

4. Move Multiple times and Sense (again).

Observation: Markov's chain uses Bayesian distribution for uncertainties considerations.

Note: For now, we consider one single sense and then after single move and repeating this steps for given set of senses and movements. Here either vertical or horizontal movement corresponds to one single movement. So, Marchov's chain restricts as below shown fig.

Simplified Marchov's Localization:

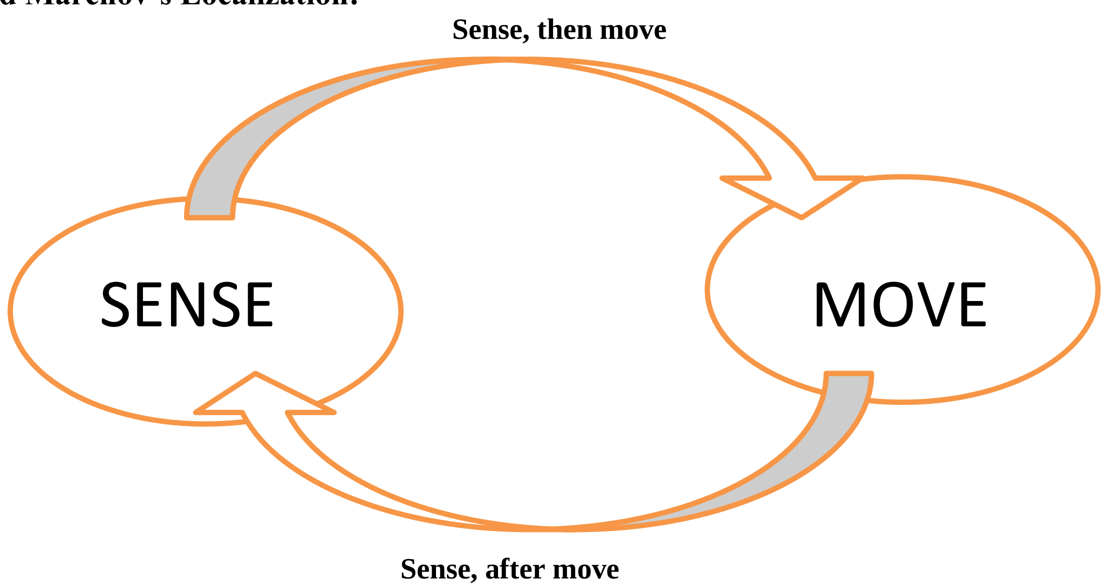


Fig.2 : Marchov's Localization chain we used is/as a trade-off for sensing and motion errors

Maths behind the Localization:

For First Sense and First Motion:

for initial movement, $$P(\text{motion1/sense0}) = \frac{P(\text{sense0/motion1}) \times P(\text{motion1})}{P(\text{sense0})}$$

for second movement, $$P(\text{motion2/sense1}) = \frac{P(\text{sense1/motion2}) \times P(\text{motion2})}{P(\text{sense1})}$$

So, here sense posterior probability will be the prior probability to the movement and likewisely, posterior probability of move will be the prior probability to the sense.

3. Implementation with Test Inputs

Markov Localization uses an explicitly specified probability distribution across all possible robot's senses and positions.

1. Considering a 5X5 world with random colored 2D-grid world. (not necessarily the world below)

```
world = {  'R','G','G','R','R';
            'R','G','G','R','G';
            'R','G','R','R','G';
            'R','G','R','G','R';
            'G','R','G','G','R'};
```

2. As Firstly, prior probabilities of senses are set to 0.04; i.e; P(sense).
   As, total probabilities do sum to one, each color grid gets probability: p= 1/25 = 0.04

```
p = ones(5,5)/25;    % uniform p.d.
```

3. Always consider that robot makes at least one sense (not necessarily Red)

```
z={'R','R'};   % Here we can take multiple measurements
```

4. Always consider that robot makes at least one move per sense.

   Notations:

   [0 , 1] → move horizontally by one unit right
   [0 , -1] → move horizontally by one unit left
   [1 , 0] → move vertically by one unit down
   [-1 , 0] → move vertically by one unit down

```
We take,
      u = 2;  %   horizontal move by 2 units
      v = 0;  %   no vertical move
```

5. Iterate :: one moves per one sense

```
for i = 1:length(z)
        q = twoDsense(p,z(i), world)       %sensing
        q = twoDmove(q ,u , v)             %move
end
```

---- To run the matlab code → Read instructions in README.TXT

4. Observation

(Tradeoffs for sensing accuracy and motion error)

Here, we have to compromise either with sensor accuracy or motion error.

If one wants to achieve high sensitive model with repetitive sensed measurements by sensor (to avoid uncertainties in sensing measurement) then move for every single movement, then motion of the robot is delayed in time. So time delay can be compromised for sensing accuracy.

If one wants to achieve accurate motion (to avoid uncertainties in overshoot or undershoot of motion uncertainties), then repetitive sense measurements should be controlled by using sensor i.e; sensing has to be compromised. So either the case is a problematic issue in real world application.

5. Conclusion

A trade-off is maintained between sensor measurement and motion error inorder to make robot more optimized to the existing environment.

Advantages:

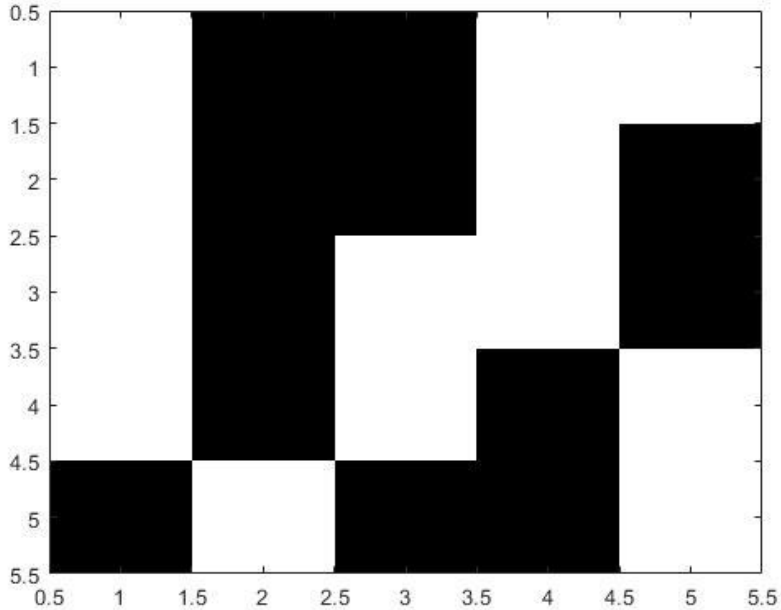
Markov localization allows for localization starting from any unknown position and can thus recover from ambiguous situations because the robot can track multiple, completely disparate possible positions.

Disadvantages:

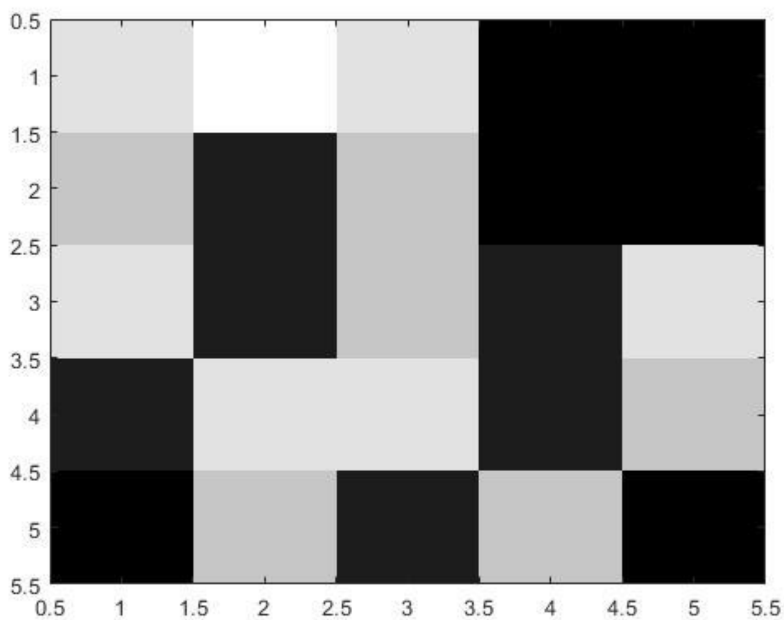
To update the probability of all positions within the whole state space at any time requires a discrete representation of the space (grid). The required memory and computational power can thus limit precision and map size.

6. Outputs

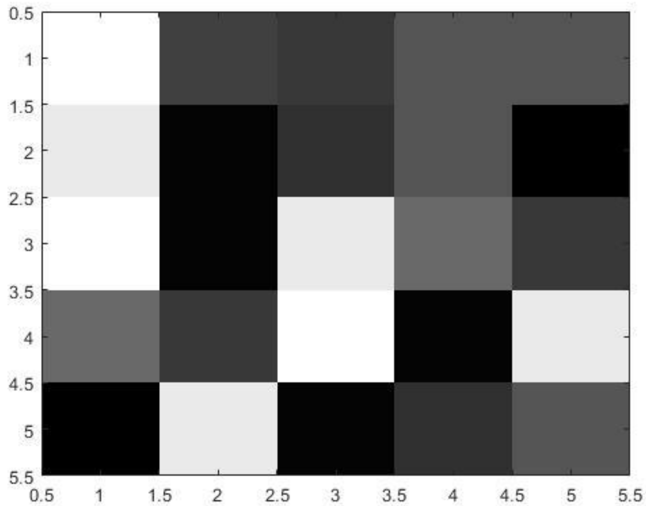
Sense-1: Sense with Inaccuracy



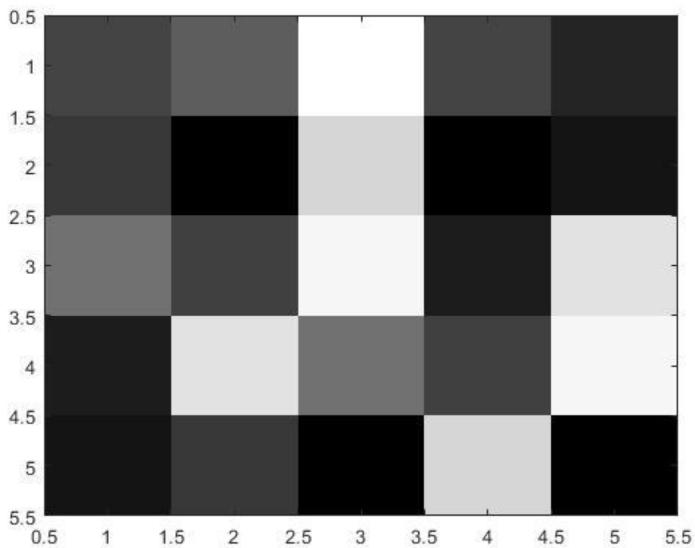
Move-1: Move with Motion uncertainty



Sense-2: Sense with Inaccuracy
(Taking Motion-1 as Sense-2 Input)



Move-2: Move with Motion uncertainty



You can observe the sense and move probabilities variations in gray scale

We can also try this with various combinations of initial probabilities, world, sense measurements, motion parameters.

7. References

[1] “PROBABILISTIC ROBOTICS” by Sebastian THRUN and Wolfram BURGARD
[2] “Mobile Robot Localization” :: http://www.cs.cmu.edu/~rasc/Download/AMRobots5.pdf
[3] Localization Program - AI for Robotics :: https://www.youtube.com/watch?v=9a42_zEeeA0
[4] Introduction to Mobile Robotics:: http://ais.informatik.uni-freiburg.de/teaching/ss18/robotics/
[5] Udacity’s online course:: https://www.udacity.com/course/cs271