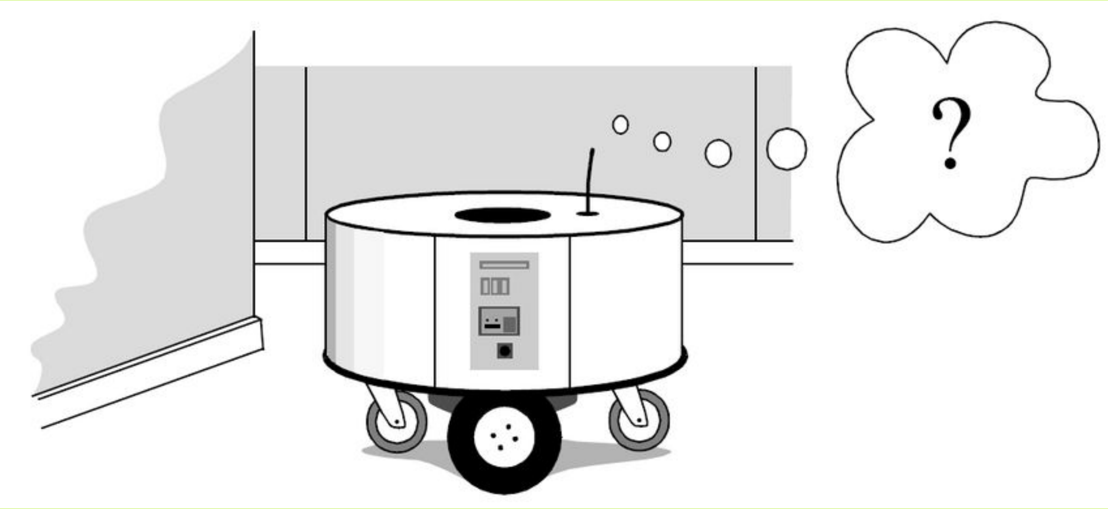


Monte Carlo Localization

Project by- Sarthak Mishra (18388) and Siddhant Sekhar (18395)

Contents

1.

Monte Carlo Method

What did we learn in the course

Monte Carlo Methods

- Monte Carlo methods, or Monte Carlo experiments, are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results.
- use randomness to solve problems that might be deterministic in principle.
- Can be used to solve any problem having a probabilistic interpretation.

2.

Robot Localization

The problem

Robot Localization

- Estimate the state of the robot at the current time-step k , given knowledge about the initial state and all measurements up to the current time.

$$Z^k = \{ z_k, i = 1 \dots k \}$$

- Three-dimensional state vector: Position and Orientation of Robot

$$\mathbf{x} = [\, x, y, \theta \,]^T$$

- Interested in constructing the posterior density $p(\, \mathbf{x}_k \mid Z^k \,)$ of the current state conditioned on all measurements.

- This PDF is taken to represent all the knowledge we possess about the state x_k, and from it we can estimate the current position.

Prediction Phase

- to localize the robot we need to recursively compute the density $p(\mathbf{x}_k \mid Z^k)$ at each time-step. This is done in two phases:
  a. **Prediction Phase** and
  b. **Update Phase**
- We use a motion model to predict the current position of the robot in the form of a predictive PDF $p(\mathbf{x}_k \mid Z^k)$, taking only motion into account.
- We assume that the current state $\mathbf{x}_k$ is only dependent on the previous state $\mathbf{x}_{k-1}$ and that the motion model is specified as a conditional density $p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ where $\mathbf{u}_{k-1}$ is a control input.
- Predictive density over x_k is obtained by integration:

$$p(\mathbf{x}_k \mid Z^{k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1} , \mathbf{u}_{k-1}) * p(\mathbf{x}_{k-1} \mid Z^{k-1}) * d\mathbf{x}_{k-1}$$

Update Phase

- We use a measurement model to incorporate information from the sensors to obtain the posterior PDF $p(\mathbf{x}_k \mid Z^k)$
- We assume that the measurement $\boldsymbol{z}_k$ is conditionally independent of earlier measurements Z^{k-1} given $\mathbf{x}_k$, and that the measurement model is given in terms of a likelihood $p(\boldsymbol{z}_k / \mathbf{x}_k)$.
- This term expresses the likelihood that the robot is at location $\mathbf{x}_k$ given that $\boldsymbol{z}_k$ was observed.
- The posterior density over $\mathbf{x}_k$ is obtained using Bayes theorem:

$$p(\mathbf{x}_k|Z^k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|Z^{k-1})}{p(\mathbf{z}_k|Z^{k-1})}$$

- After the update phase, the process is repeated recursively.

3.

Monte Carlo Localization

The solution

Monte Carlo Localization

- In sampling-based methods one represents the density $p(\mathbf{x}_k \mid Z^k)$ by a set of **N** random samples or particles $S_k = \{ s^i_k ; i = 1 \ldots N \}$ drawn from it.
- The goal is then to recursively compute at each time step ***k*** the set of samples S_k that is drawn from $p(\mathbf{x}_k \mid Z^k)$.
- These class of algorithms known as **Particle Filters** or **Monte Carlo Filter** or **Monte Carlo Localization.** The algorithm proceeds as follows:
  - **Prediction Phase:-**
    - we start from the set of particles S_{k-1} computed in the previous iteration
    - apply the motion model to each particle s^i_{k-1}, by sampling from the density $p(\mathbf{x}_k \mid s^i_{k-1}, \mathbf{u}_{k-1})$:

MCL (Prediction and Update Phase)

- Monte Carlo Prediction Phase:
  (I) for each particle s^i_{k-1}:
  draw one sample s'^i_k from $p(x_k \mid s^i_{k-1}, u_{k-1})$
- The prime in s'^i_k indicates that we have not yet incorporated any sensor measurement at time *k*.

- **Update Phase:-**
  - we take into account the measurement $\mathbf{z}_k$
  - Weight each sample S'_k by the weight $m^i_k = p(\mathbf{z}_k / s'^i_k)$, i.e. the likelihood of s'^i_k given $\mathbf{z}_k$.
  - We then obtain S_k by resampling from this weighted set:
    (II) for j = 1..N:
    draw one S_k sample s^j_k from $\{ s'^i_k, m^i_k \}$

MCL (Prediction and Update Phase)

- The resampling selects higher probability samples s'^{i}_{k} that have a high likelihood associated with them.
- In doing so a new set S_k is obtained that approximates a random sample from $p(\mathbf{x}_k \mid Z^k)$.

After the update phase, the steps **(I)** and **(II)** are repeated recursively. To initialize the filter, we start at time k = 0 with a random sample $S_0 = \{ s^{i}_{0} \}$.

MCL - Graphical Interpretation

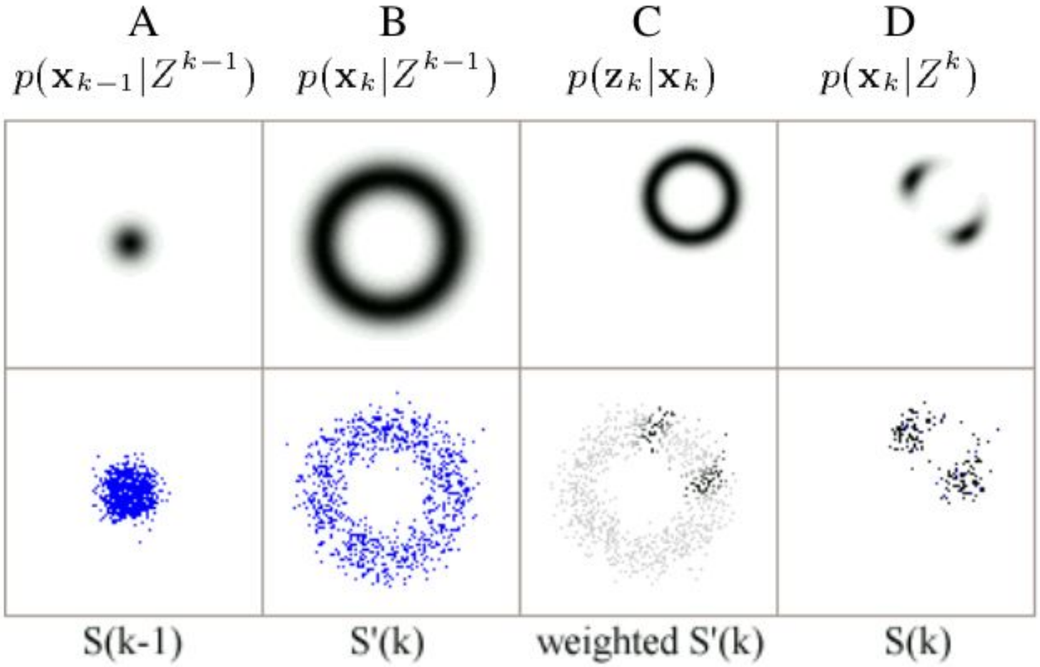
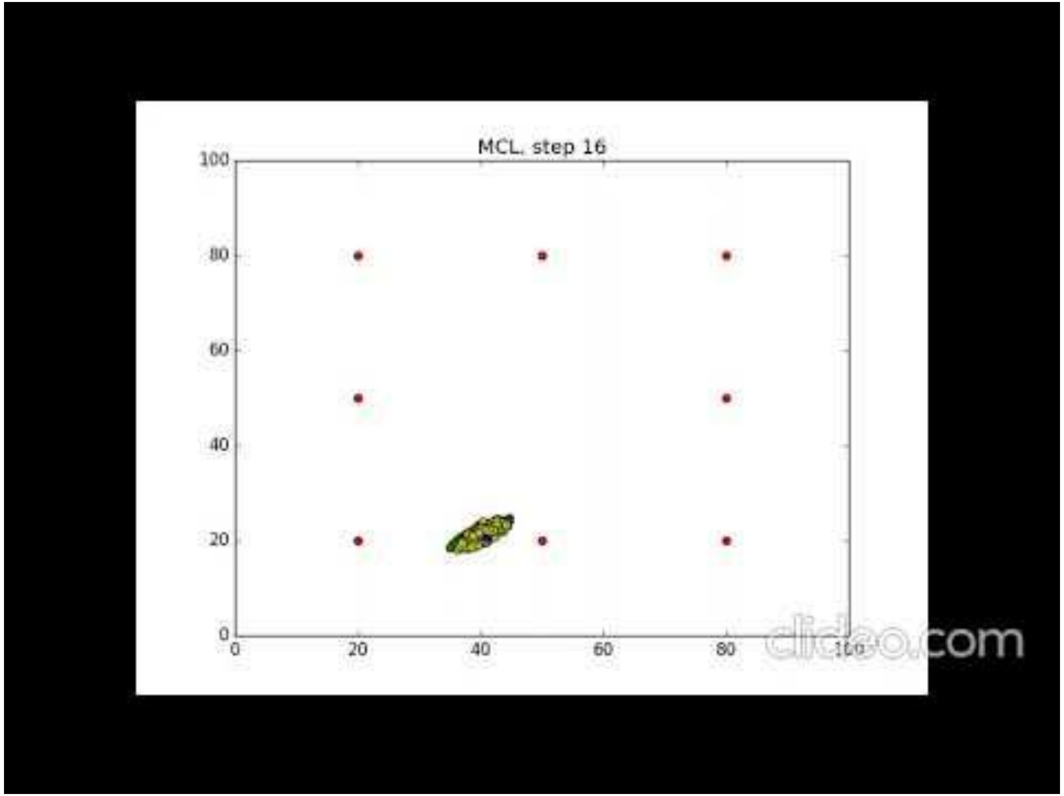


Fig. 1: The probability densities and particle sets for one iteration of the algorithm.

4.

Results and Discussions

What we obtained after hours of tedious effort :P



Result output from our code.

Limitations of MCL

works best for 10% to 20% perpetual noise

performs poorly when the noise level is too small

MCL with accurate sensors may perform worse than MCL with inaccurate sensor!!

Bibliography

- Monte Carlo Localization for Mobile Robots
- Monte Carlo Localization (Wiki)
- Particle Filter (Wiki)
- Particle Filter and Monte Carlo Localization (Lecture by Prof. Cyrill Stachniss)
- Robust Monte Carlo Localization for Mobile Robots