Particle Filter (just intuitions)

Model the location of the object with a particle at that location.

Since we know the object’s location probabilistically, let’s scatter \( N \) particles at ... and then compute a weight for each particle based on its agreement with motion models & measurement.

At any time, the particles capture how the object location is distributed over space.

1. \( N \) particles

2. Move

3. Obtain \( P(m|s) \) = measurement error

4. Based on measurement error distribution

5. Redistribute low weight particles at the locations of...
Observe:

Always in the system

Scattering of \( N \) particles captures my

Particles are like magnifying lens (or clones) \( \rightarrow \) you allocate

more particles to places/trajectories

1. P.F. operating on space.
2. Kind of motion model can be incorporated
3. Noise can be modeled
4. \( S_k = f(S_{k-1}) + n \) \( \exists \) any possible
   \( m = g(S_k) + n \) \( \text{even if} \)
5. CPU load is function of \( N \), so you can
   based on system resources.

Kalman Filter: When \( f \) and \( g \) are

AND noise is \( \} \) simple algorithm

in that case.
SLAM: Simultaneous localization & mapping

Main goal:
Can I compute both the sensor location and the visible objects' locations:

\[ \langle L_1, L_2, L_3, L_4 \rangle \]
\[ \langle L_{10}, L_{11} \rangle \]

Basically, I want: \[ P( \) }