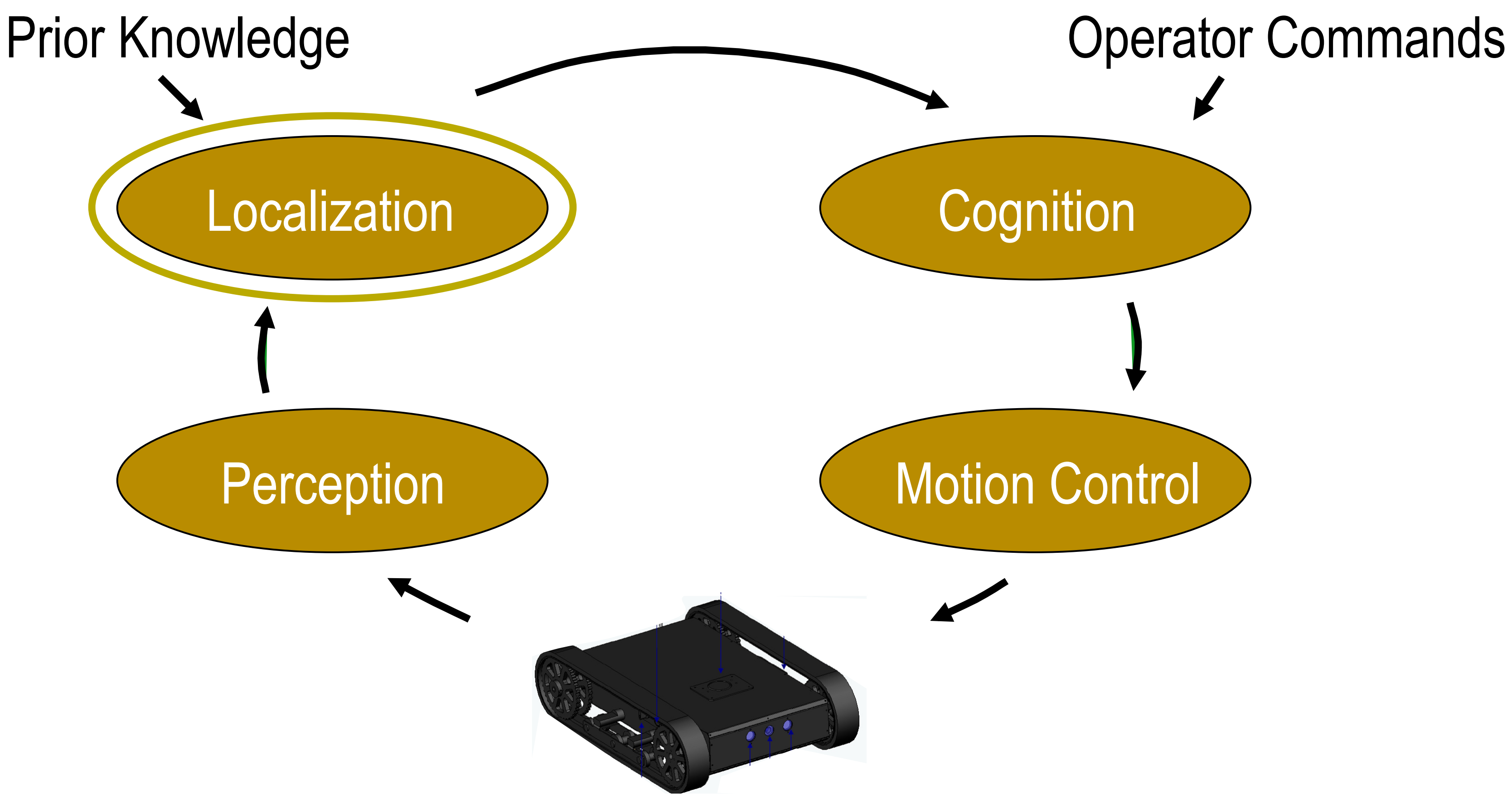
E190Q – Lecture 12
Autonomous Robot Navigation

Instructor: Chris Clark

Semester: Spring 2014

Figures courtesy of Probabilistic Robotics (Thrun et. Al.)

Control Structures
Planning Based Control



SLAM

- Introduction to SLAM
- Landmark based SLAM
- Occupancy Grid based SLAM

Methods

- Mapping Problem
  - Determine the state of the environment given a known robot state.
- Localization Problem
  - Determine the state of a robot given a known environment state.
- SLAM – Simultaneous Localization and Mapping
  - Simultaneously determine the state of a robot and state of the environment.

SLAM

- Full SLAM
  - Estimates entire path of robot and across all time.

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

- On Line SLAM
  - Estimates current pose of the robot and map.
  - Integrations typically done one at a time

$$p(x_t, m \mid z_{1:t}, u_{1:t})$$

SLAM

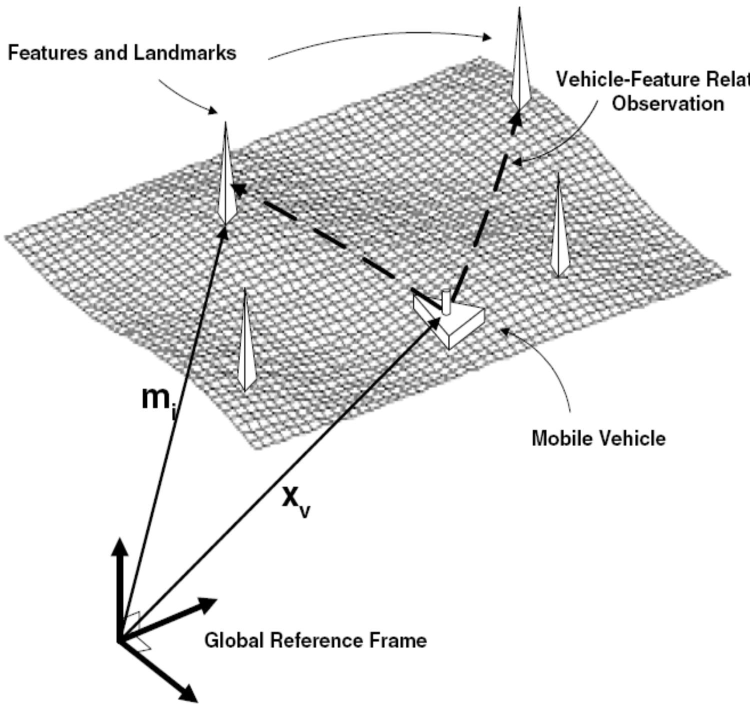
- Introduction to SLAM
- Landmark based SLAM
- Occupancy Grid based SLAM

SLAM

- Landmark based SLAM
  - Features
    - **Observable** parts or characteristics of objects in the environment.
    - E.g. corners, colors, walls, etc.
  - Landmarks
    - **Static** and **easily recognizable** features**.**
    - E.g. Orange cones

SLAM

- Landmark based SLAM
  - Given:
    - The robot's odometry **u**
    - Observations of nearby features **z**
  - Estimate:
    - Robot States **x**
    - Landmark States **M**



EKF SLAM

- To start, lets recall our EKF Localization…

EKF Localization

- In our example, the state vector to be estimated, $\mathbf{x}$, was a 3x1 vector

  e.g.

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- Associated Covariance, $\mathbf{P}$

$$\mathbf{P} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta\theta} \end{bmatrix}$$

EKF Localization

Prediction

1. $\mathbf{x'}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t)$
2. $\mathbf{P'}_t = \mathbf{F}_{x,t-1} \mathbf{P}_{t-1} \mathbf{F}_{x,t-1}^T + \mathbf{F}_{u,t} \mathbf{Q}_t \mathbf{F}_{u,t}^T$

Correction

3. $\mathbf{z}^i_{exp,t} = h^i(\mathbf{x'}_t, \mathbf{M})$
4. $\mathbf{v}_t = \mathbf{z}_t - \mathbf{z}_{exp,t}$
5. $\Sigma_{IN,t} = \mathbf{H}^i_{x',t} \mathbf{P'}_t \mathbf{H}^i_{x',t}{}^T + \mathbf{R}^i_t$
6. $\mathbf{K}_t = \mathbf{P'}_t \mathbf{H}_{x',t}^T (\Sigma_{IN,t})^{-1}$
7. $\mathbf{x}_t = \mathbf{x'}_t + \mathbf{K}_t \mathbf{v}_t$
8. $\mathbf{P}_t = \mathbf{P'}_t - \mathbf{K}_t \Sigma_{IN,t} \mathbf{K}_t^T$

EKF SLAM

- In SLAM, the state vector to be estimated

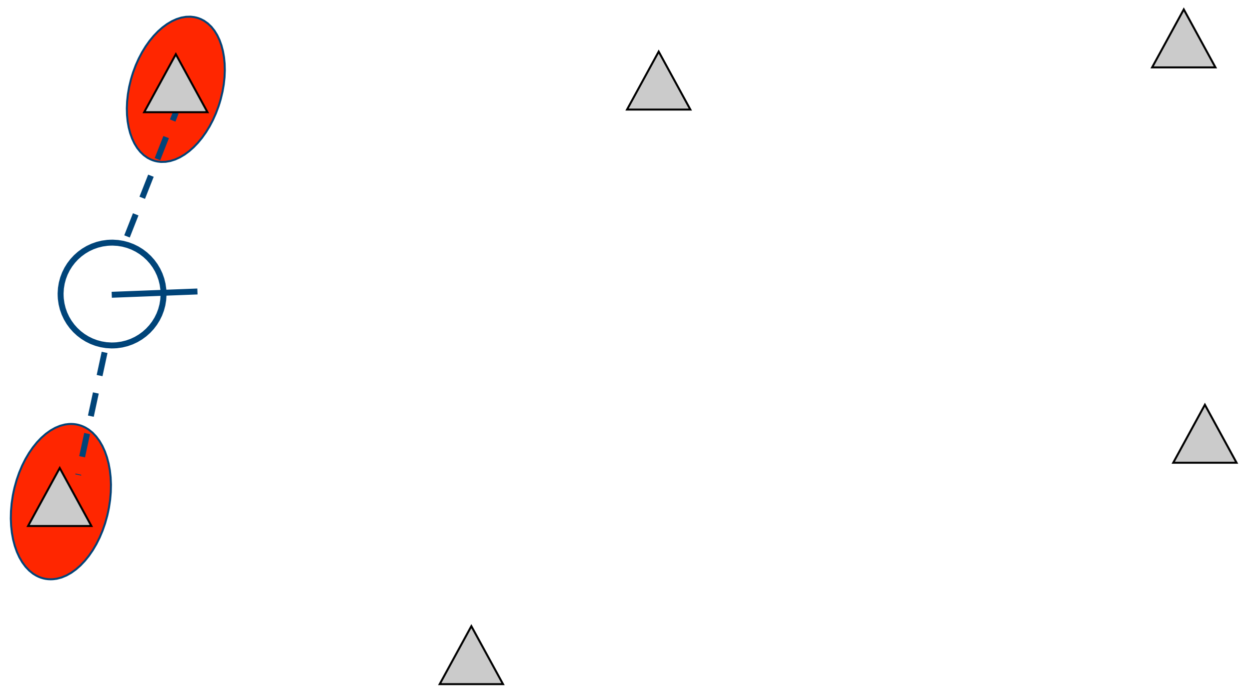
$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \\ x_{f1} \\ y_{f1} \\ \ldots \\ x_{fN} \\ y_{fN} \end{bmatrix}$$

EKF SLAM

- The covariance Matrix $\mathbf{P}$

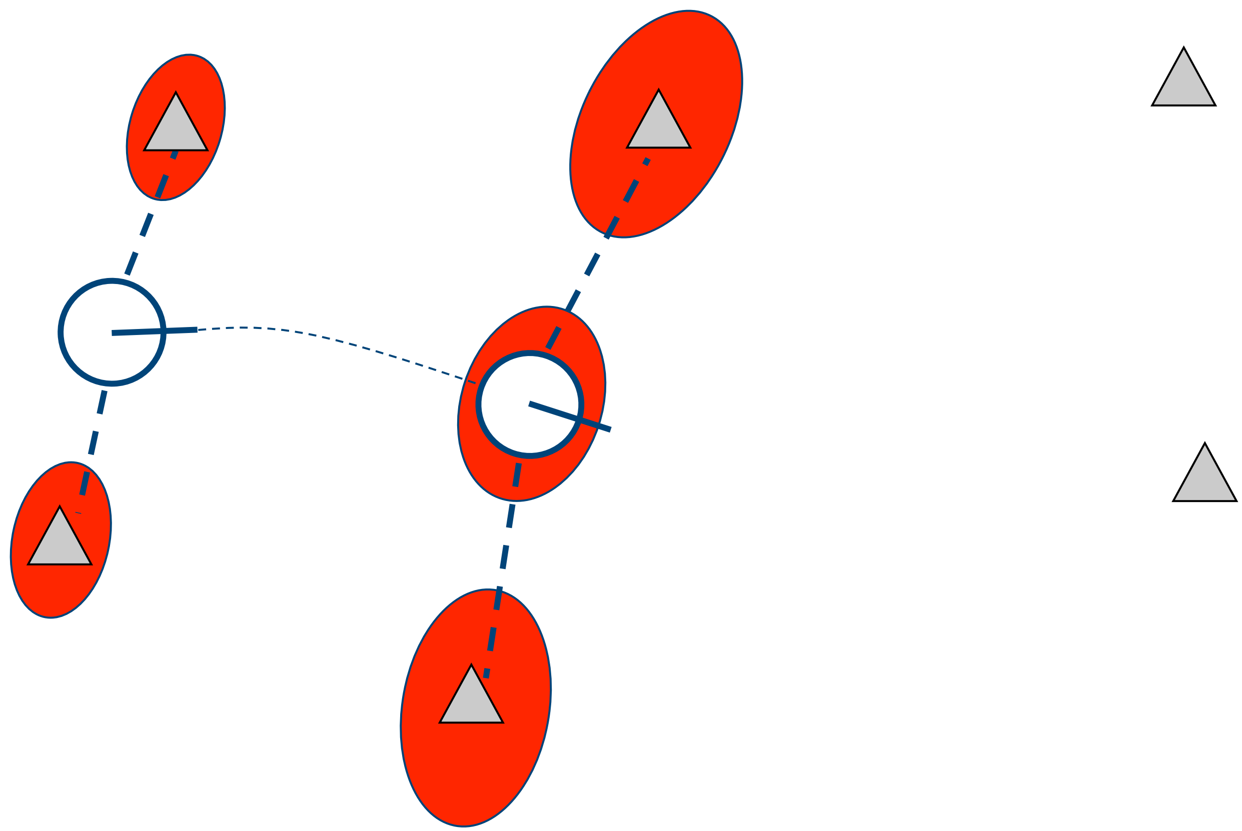
$$\mathbf{P} =$$

Landmark Based Example



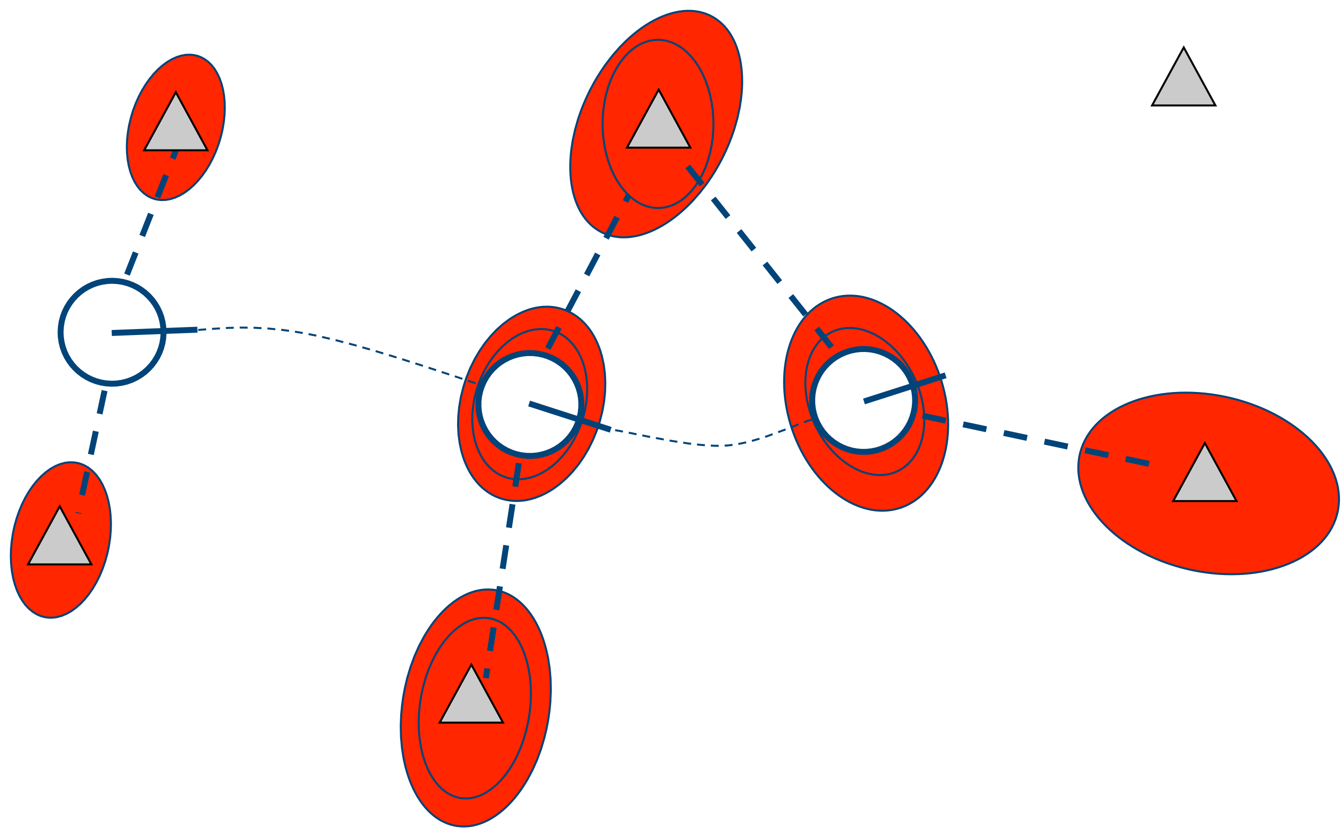
Robot path error correlates errors in the map

Landmark Based Example



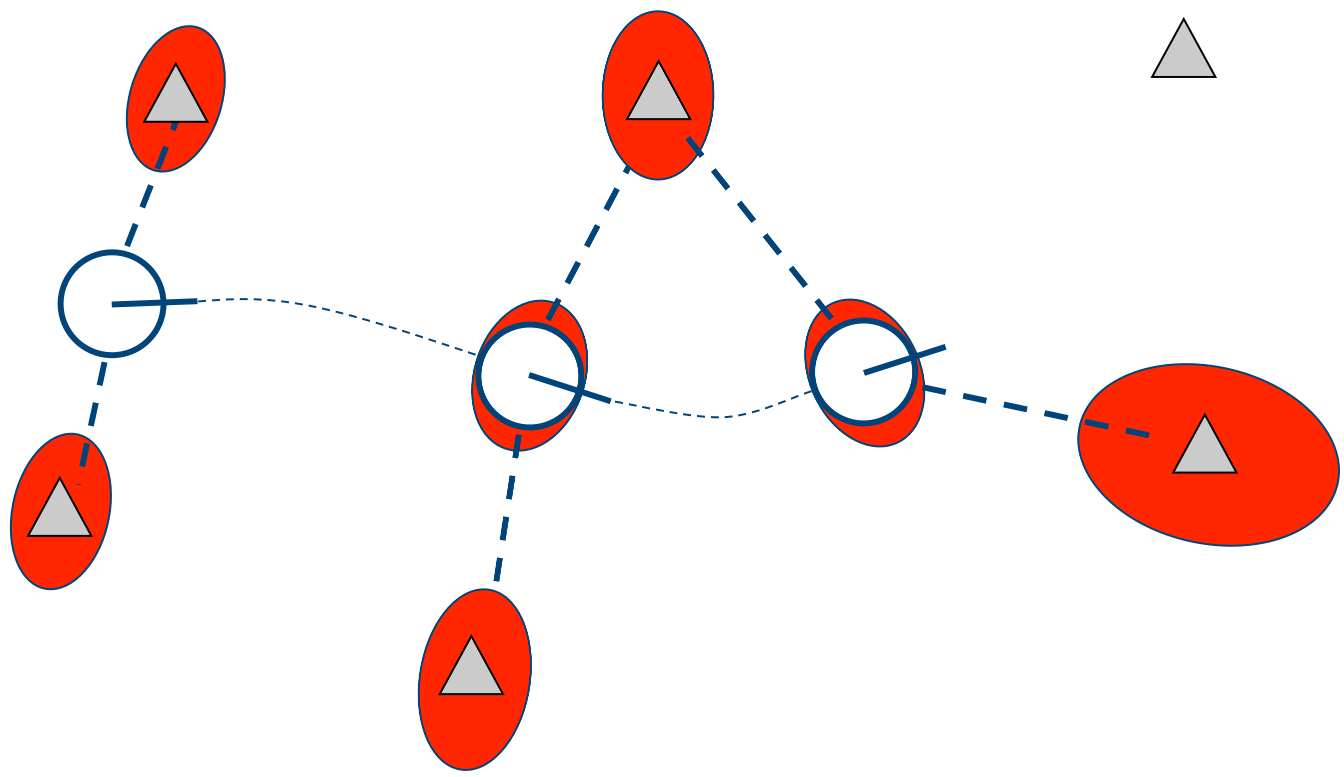
Robot path error correlates errors in the map

Landmark Based Example



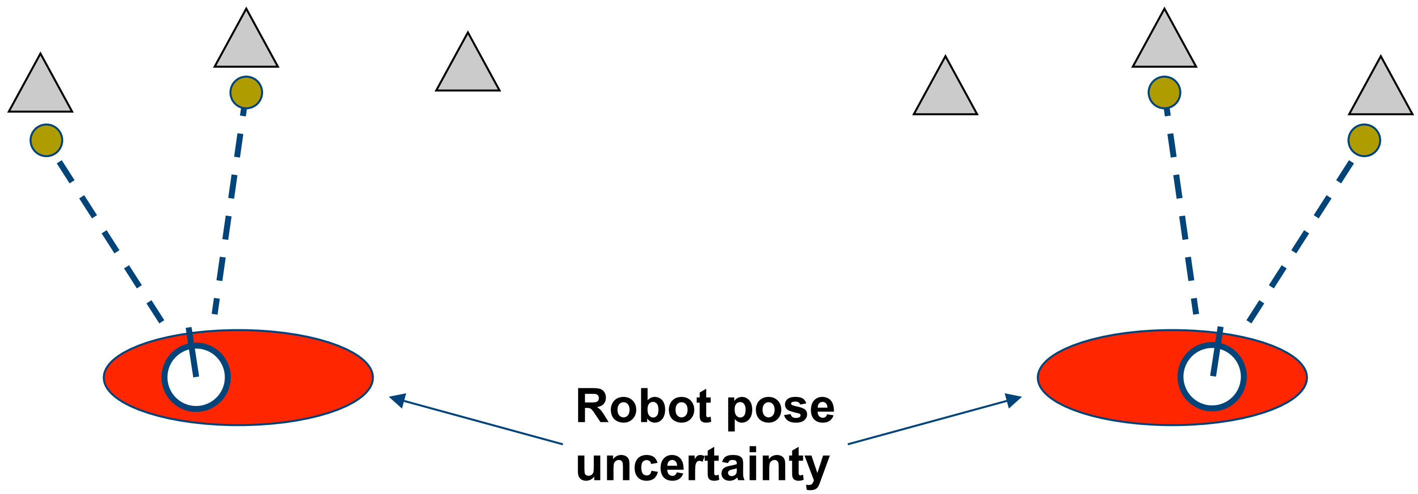
Robot path error correlates errors in the map

Landmark Based Example



Robot path error correlates errors in the map

Why is SLAM a hard problem?



- The matching between observations and landmarks is unknown
- Wrong data associations can have catastrophic consequences

EKF SLAM

Prediction

1. $\mathbf{x}'_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t)$
2. $\mathbf{P}'_t = \mathbf{F}_{x,t-1} \mathbf{P}_{t-1} \mathbf{F}_{x,t-1}^T + \mathbf{F}_{u,t} \mathbf{Q}_t \mathbf{F}_{u,t}^T$

Correction

3. $\mathbf{z}^i_{exp,t} = h^i(\mathbf{x}'_t)$
4. $\mathbf{v}_t = \mathbf{z}_t - \mathbf{z}_{exp,t}$
5. $\Sigma_{IN,t} = \mathbf{H}^i_{x',t} \mathbf{P}'_t \mathbf{H}^i_{x',t}{}^T + \mathbf{R}^i_t$
6. $\mathbf{K}_t = \mathbf{P}'_t \mathbf{H}_{x',t}{}^T (\Sigma_{IN,t})^{-1}$
7. $\mathbf{x}_t = \mathbf{x}'_t + \mathbf{K}_t \mathbf{v}_t$
8. $\mathbf{P}_t = \mathbf{P}'_t - \mathbf{K}_t \Sigma_{IN,t} \mathbf{K}_t^T$

Prediction Step

- Localization Motion model

$$\mathbf{x'}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \Delta s_t \cos(\theta_{t-1} + \Delta\theta_t/2) \\ \Delta s_t \sin(\theta_{t-1} + \Delta\theta_t/2) \\ \Delta\theta_t \end{bmatrix}$$

Prediction Step

- SLAM Motion Model

$$\mathbf{x'}_t = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \\ x_{f1t-1} \\ y_{f1t-1} \\ \ldots \\ x_{fNt-1} \\ y_{fNt-1} \end{bmatrix} + \begin{bmatrix} \Delta s_t \cos(\theta_{t-1} + \Delta\theta_t/2) \\ \Delta s_t \sin(\theta_{t-1} + \Delta\theta_t/2) \\ \Delta\theta_t \\ 0 \\ 0 \\ \ldots \\ 0 \\ 0 \end{bmatrix}$$

Prediction Step

- Covariance
  - Recall, we linearize the motion model f to obtain

$$\mathbf{P'}_t = \mathbf{F}_{x,t-1}\,\mathbf{P}_{t-1}\mathbf{F}_{x,t-1}^{\,T} + \mathbf{F}_{u,t}\,\mathbf{Q}_t\,\mathbf{F}_{u,t}^{\,T}$$

where

$\mathbf{Q}_t$ = *Motion Error Covariance Matrix*

$\mathbf{F}_{x,t-1}$ = *Derivative of f with respect to state* $\mathbf{x}_{t-1}$

$\mathbf{F}_{u,t}$ = *Derivative of f with respect to control* $\mathbf{u}_t$

Prediction Step

- Covariance

$$P'_t = F_{x,t-1} P_{t-1} F_{x,t-1}^T + F_{u,t} Q_t F_{u,t}^T$$

Prediction Step

- Covariance

$$F_{x,t-1} = \begin{bmatrix} dx_t/dx_{t-1} & dx_t/dy_{t-1} & dx_t/d\theta_{t-1} & dx_t/dx_{f1t-1} & \ldots & dx_t/dy_{fNt-1} \\ dy_t/dx_{t-1} & dy_t/dy_{t-1} & dy_t/d\theta_{t-1} & dy_t/dx_{f1t-1} & \ldots & dy_t/dy_{fNt-1} \\ d\theta_t/dx_{t-1} & d\theta_t/dy_{t-1} & d\theta_t/d\theta_{t-1} & d\theta_t/dx_{f1t-1} & \ldots & d\theta_t/dy_{fNt-1} \\ dx_{f1t}/dx_{t-1} & dx_{f1t}/dy_{t-1} & dx_{f1t}/d\theta_{t-1} & dx_{f1t}/dx_{f1t-1} & \ldots & dx_{f1t}/dy_{fNt-1} \\ dy_{f1t}/dx_{t-1} & dy_{f1t}/dy_{t-1} & dy_{f1t}/d\theta_{t-1} & dy_{f1t}/dx_{f1t-1} & \ldots & dy_{f1t}/dy_{fNt-1} \\ & \ldots & & \ldots & & \\ dy_{fNt}/dx_{t-1} & dy_{fNt}/dy_{t-1} & dy_{fNt}/d\theta_{t-1} & dy_{fNt}/dx_{f1t-1} & \ldots & dy_{fNt}/dy_{fNt-1} \end{bmatrix}$$

Prediction Step

- Covariance

$$P'_t = F_{x,t-1} P_{t-1} F_{x,t-1}^T + \mathbf{F}_{u,t} \mathbf{Q}_t \mathbf{F}_{u,t}^T$$

$$\mathbf{Q}_t = \begin{bmatrix} k\,|\Delta s_{r,t}| & 0 \\ 0 & k\,|\Delta s_{l,t}| \end{bmatrix}$$

$$\mathbf{F}_{u,t} = \begin{bmatrix} df/d\Delta s_{r,t} & df/d\Delta s_{l,t} \end{bmatrix}$$

Prediction Step

- Covariance

$$
\mathbf{F}_{u,t} = \begin{bmatrix} dx_t/d\Delta s_{r,t} & dx_t/d\Delta s_{l,t} \\ dy_t/d\Delta s_{r,t} & dy_t/d\Delta s_{l,t} \\ d\theta_t/d\Delta s_{r,t} & d\theta_t/d\Delta s_{l,t} \\ dx_{f1t}/d\Delta s_{r,t} & dx_{f1t}/d\Delta s_{l,t} \\ dy_{f1t}/d\Delta s_{r,t} & dy_{f1t}/d\Delta s_{l,t} \\ \multicolumn{2}{c}{\cdots} \\ dy_{fNt}/d\Delta s_{r,t} & dy_{fNt}/d\Delta s_{l,t} \end{bmatrix}
$$

EKF SLAM

Prediction

1. $\mathbf{x'}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t)$
2. $\mathbf{P'}_t = \mathbf{F}_{x,t-1} \mathbf{P}_{t-1} \mathbf{F}_{x,t-1}^T + \mathbf{F}_{u,t} \mathbf{Q}_t \mathbf{F}_{u,t}^T$

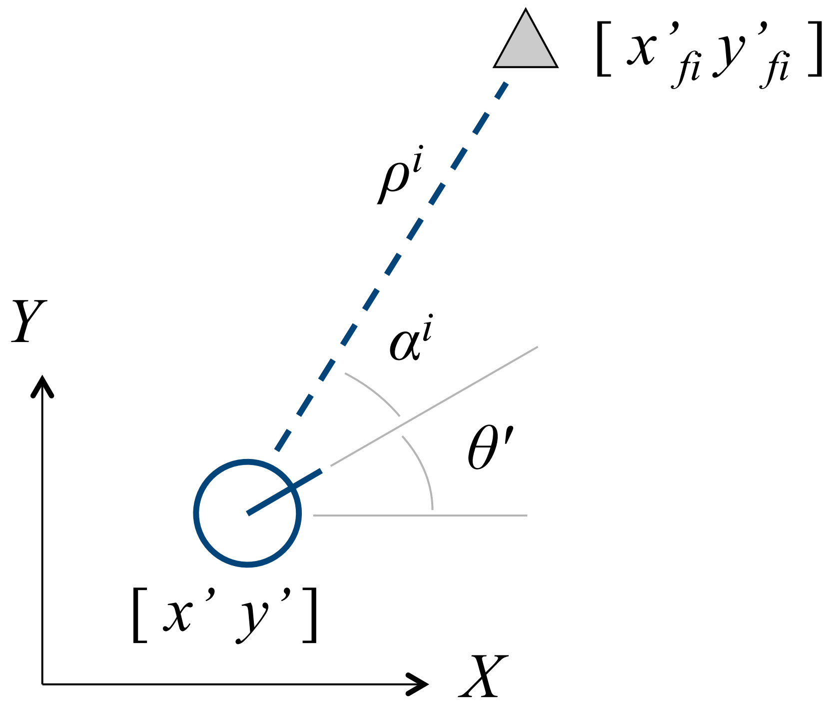
Correction

3. $\mathbf{z}^i_{exp,t} = h^i(\mathbf{x'}_t)$
4. $\mathbf{v}_t = \mathbf{z}_t - \mathbf{z}_{exp,t}$
5. $\Sigma^i_{IN,t} = \mathbf{H}^i_{x',t} \mathbf{P'}_t \mathbf{H}^i_{x',t}{}^T + \mathbf{R}^i_t$
6. $\mathbf{x}_t = \mathbf{x'}_t + \mathbf{K}_t \mathbf{v}_t$
7. $\mathbf{P}_t = \mathbf{P'}_t - \mathbf{K}_t \Sigma_{IN,t} \mathbf{K}_t^T$
8. $\mathbf{K}_t = \mathbf{P'}_t \mathbf{H}_{x',t}^T (\Sigma_{IN,t})^{-1}$

Correction Step

- Measurement of i^{th} landmark

$$\mathbf{z}^{i}_{t} = \begin{bmatrix} \alpha^{i}_{t} \\ \rho^{i}_{t} \end{bmatrix}$$



Correction Step

- Expected Measurement calculation

$$\mathbf{z}^{i}_{exp,t} = \begin{bmatrix} \alpha^{i}_{exp,t} \\ \rho^{i}_{exp,t} \end{bmatrix}$$

$$= h^{i}(\mathbf{x}'_{t})$$

$$= \begin{bmatrix} atan2(y_{fi} - y'_{t}, x_{fi} - x'_{t}) - \theta'_{t} \\ ((y_{fi} - y'_{t})^{2} + (x_{fi} - x'_{t})^{2})^{0.5} \end{bmatrix}$$

Correction Step

- Innovation calculation

$$\mathbf{v}^i_t = \mathbf{z}^i_t - \mathbf{z}^i_{exp,t}$$

$$= \begin{bmatrix} \alpha^i_t - \alpha^i_{exp,t} \\ \rho^i_t - \rho^i_{exp,t} \end{bmatrix}$$

Correction Step

- Innovation covariance calculation

$$\Sigma^{i}_{IN,t} = \mathbf{H}^{i}_{x',t}\,\mathbf{P}'_{t}\,\mathbf{H}^{i\,T}_{x',t} + \mathbf{R}^{i}_{t}$$

where

$\mathbf{R}^{i}_{t}$ = *Feature Measurement Error Covariance Matrix*

$\mathbf{H}^{i}_{x',t}$ = *Derivative of h with respect to state* $\mathbf{x}'_{t}$

Correction Step

- Innovation covariance calculation

$$\Sigma^{i}_{IN,t} = \mathbf{H}^{i}_{x',t}\,\mathbf{P'}_{t}\,\mathbf{H}^{i\,T}_{x',t} + \mathbf{R}^{i}_{t}$$

$$\mathbf{H}^{i}_{x',t} = \begin{bmatrix} d\alpha^{i}_{t}/dx'_{t} & d\alpha^{i}_{t}/dy'_{t} & d\alpha^{i}_{t}/d\theta'_{t} & d\alpha^{i}_{t}/dx'_{f1t} \ldots d\alpha^{i}_{t}/dy'_{fNt} \\ d\rho^{i}_{t}/dx'_{t} & d\rho^{i}_{t}/dy'_{t} & d\rho^{i}_{t}/d\theta'_{t} & d\rho^{i}_{t}/dx'_{f1t} \ldots d\rho^{i}_{t}/dy'_{fNt} \end{bmatrix}$$

Correction Step

- Innovation covariance calculation

$$\Sigma^{i}_{IN,t} = \mathbf{H}^{i}_{x',t}\,\mathbf{P}'_{t}\,\mathbf{H}^{i\;T}_{x',t} + \mathbf{R}^{i}_{t}$$

$$\mathbf{R}^{i}_{t} = \begin{bmatrix} {\sigma^{i}_{\alpha}}^{2} & 0 \\ 0 & {\sigma^{i}_{\rho}}^{2} \end{bmatrix}$$

Correction Step

- For N features …

$$\mathbf{z}_t = [\mathbf{z}^1_t \quad \mathbf{z}^2_t \ldots \quad \mathbf{z}^N_t]^T$$

$$\mathbf{z}_{exp,t} = [\mathbf{z}^1_{exp,t} \quad \mathbf{z}^2_{exp,t} \ldots \quad \mathbf{z}^N_{exp,t}]^T$$

Correction Step

- For N features…

$$\mathbf{v}_t = \mathbf{z}_t - \mathbf{z}_{exp,t}$$

$$= [\mathbf{v}^1_t \quad \mathbf{v}^2_t \dots \quad \mathbf{v}^N_t]^T$$

Correction Step

- For N features …

$$\mathbf{H}_{x',t} = \begin{bmatrix} \mathbf{H}^{1}_{x',t} \\ \mathbf{H}^{2}_{x',t} \\ \vdots \\ \mathbf{H}^{N}_{x',t} \end{bmatrix}$$

Correction Step

- For N features …

$$\Sigma_{IN,t} = \mathbf{H}_{x',t}\,\mathbf{P}_t\,\mathbf{H}_{x',t}^T + \mathbf{R}_t$$

EKF SLAM

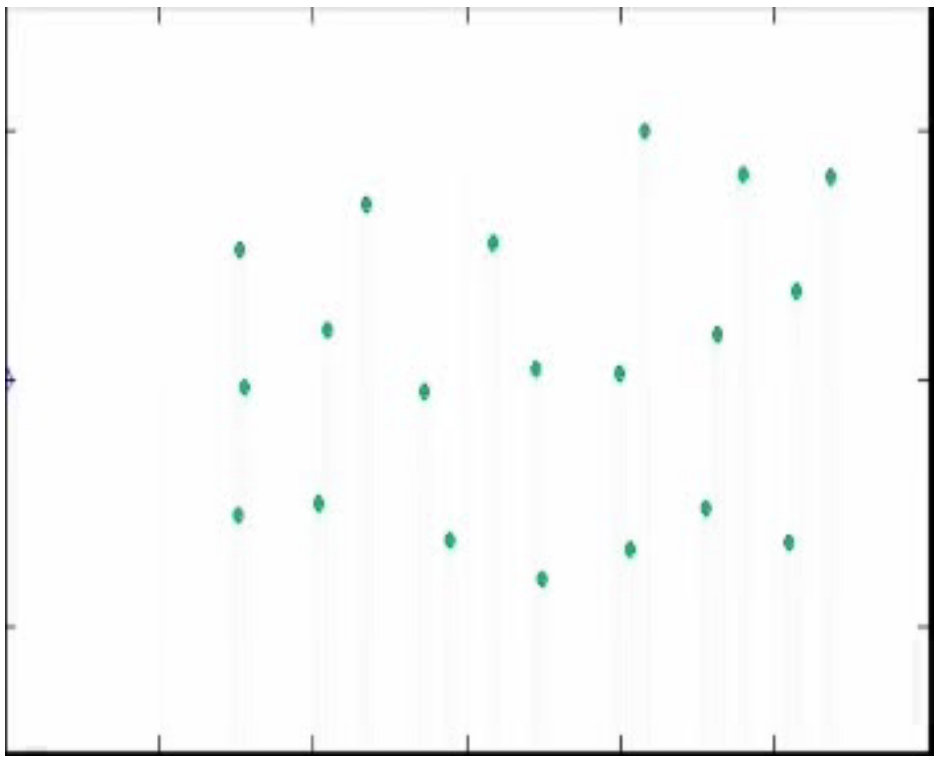
Prediction

1. $\mathbf{x}'_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t)$
2. $\mathbf{P}'_t = \mathbf{F}_{x,t-1} \mathbf{P}_{t-1} \mathbf{F}_{x,t-1}^T + \mathbf{F}_{u,t} \mathbf{Q}_t \mathbf{F}_{u,t}^T$

Correction

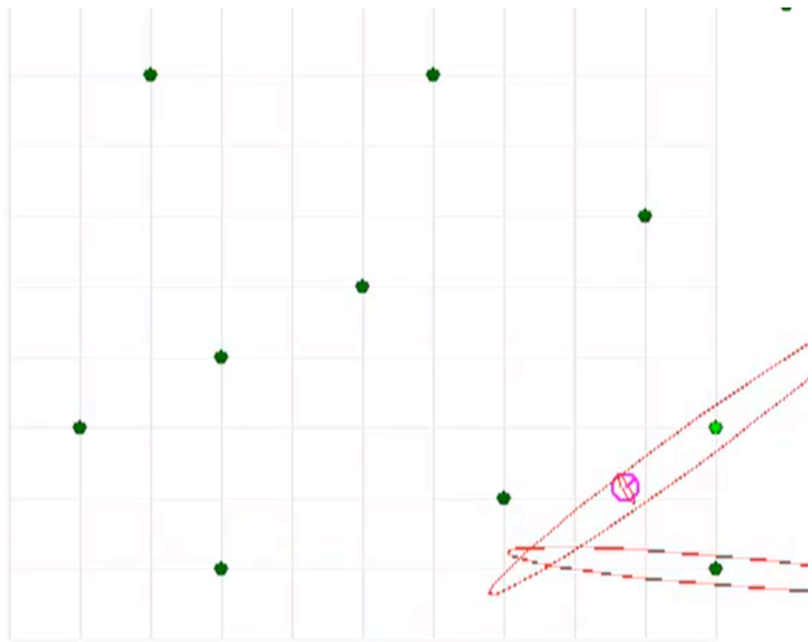
3. $\mathbf{z}^i_{exp,t} = h^i(\mathbf{x}'_t)$
4. $\mathbf{v}_t = \mathbf{z}_t - \mathbf{z}_{exp,t}$
5. $\Sigma^i_{IN,t} = \mathbf{H}^i_{x',t} \mathbf{P}'_t \mathbf{H}^i_{x',t}{}^T + \mathbf{R}^i_t$
6. $\mathbf{K}_t = \mathbf{P}'_t \mathbf{H}_{x',t}^T (\Sigma_{IN,t})^{-1}$
7. $\mathbf{x}_t = \mathbf{x}'_t + \mathbf{K}_t \mathbf{v}_t$
8. $\mathbf{P}_t = \mathbf{P}'_t - \mathbf{K}_t \Sigma_{IN,t} \mathbf{K}_t^T$

EKF SLAM



J. Langalaan – Penn State

EKF SLAM



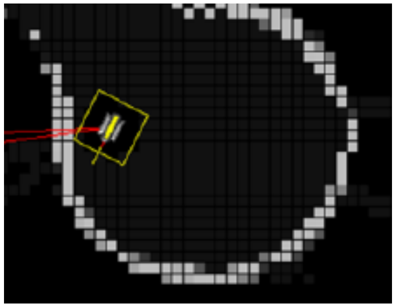
http://www.youtube.com/watch?v=vCVS9WAffi4

SLAM

- Introduction to SLAM
- Landmark based SLAM
- Occupancy Grid based SLAM

Localization & Mapping

- Occupancy Grid Mapping
  - Doesn't require knowledge of features!
  - The environment is discretized into a grid of equal sized cells, $\mathbf{M} = \{c_{ij}\}$
  - Each cell (i, j) is assigned a likelihood $P(c_{ij}) \in [0,1]$ of being occupied



- FastSLAM- *[Thrun et al., 2005]*

Localization & Mapping

- What is a Particle?
  - A particle is an individual state estimate.
  - In our SLAM, a particle i has three components

$$\{\ \underbrace{\mathbf{x}^i}_{\text{State}}\ \underbrace{\mathbf{M}^i}_{\text{Map}}\ \underbrace{w^i}_{\text{Weight}}\ \}$$

1. The state is $\mathbf{x} = [\, x\ y\ z\ \theta\ u\ v\ r\ w \,]$
2. The map is an occupancy grid $\mathbf{M}$
3. The weight w that indicates it's likelihood of being the correct state.

FastSLAM for Occupancy Grids

- Algorithm (Loop over time step t):

1. For $i = 1 \ldots N$
2. Pick $\mathbf{x}_{t-1}^{[i]}$ from $\mathbf{X}_{t-1}$
3. Draw $\mathbf{x}_t^{[i]}$ with probability $P(\mathbf{x}_t^{[i]} \mid \mathbf{x}_{t-1}^{[i]}, o_t)$
4. Calculate $w_t^{[i]} = P(z_t \mid \mathbf{x}_t^{[i]}, \mathbf{M}_t^{[i]})$
5. Update $\mathbf{M}_t^{[i]}$
6. Add $\mathbf{x}_t^{[i]}$ to $\mathbf{X}_t^{Predict}$

Prediction

7. For $j = 1 \ldots N$
8. Draw $\mathbf{x}_t^{[j]}$ from $\mathbf{X}_t^{Predict}$ with probability $w_t^{[j]}$
9. Add $\mathbf{x}_t^{[j]}$ to $\mathbf{X}_t$

Correction

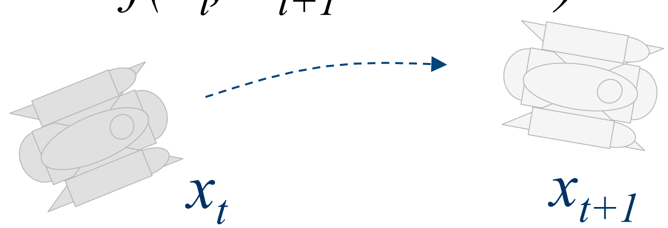
FastSLAM for Occupancy Grids

- Step 3: Draw $\mathbf{x}_t^{[i]}$ from $P(\mathbf{x}_t^{[i]} \mid \mathbf{x}_{t-1}^{[i]}, o_t)$
- The state vector is propagated forward in time to reflect the ROV motion based on control inputs and uncertainty
- The dynamic model is used to propagate particle states

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_{t+1} + \underbrace{\boldsymbol{randn}(0,\sigma_u)}_{\text{Experimentally Determined Process Noise}})$$

Experimentally Determined
Process Noise

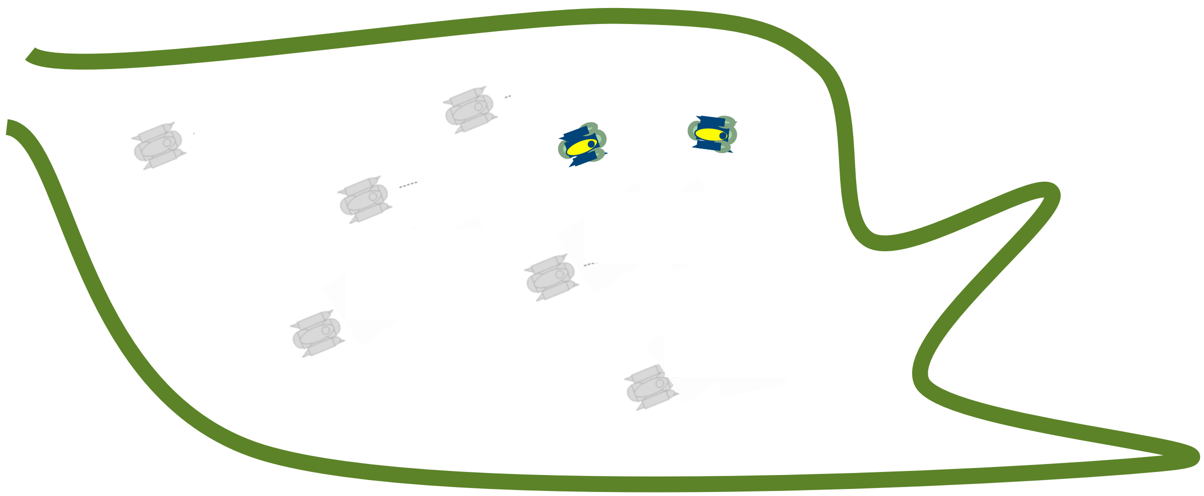
$$f(\boldsymbol{x}_t, \boldsymbol{u}_{t+1} + \boldsymbol{randn})$$



FastSLAM for Occupancy Grids

- Step 3: Draw $\mathbf{x}_t^{[i]}$ from $P(\, \mathbf{x}_t^{[i]} \mid \mathbf{x}_{t-1}^{[i]} \,, o_t)$

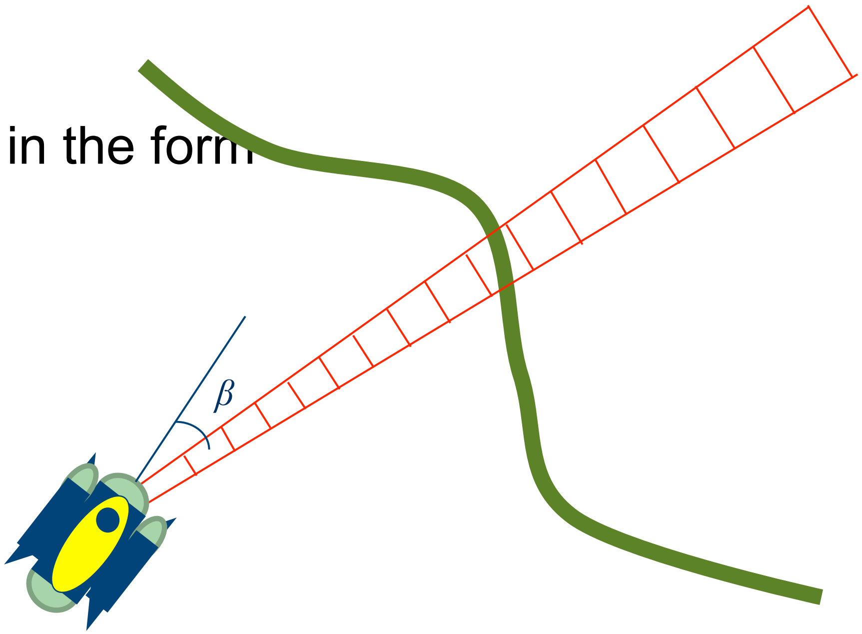
$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_{t+1} + \boldsymbol{randn})$$



FastSLAM for Occupancy Grids

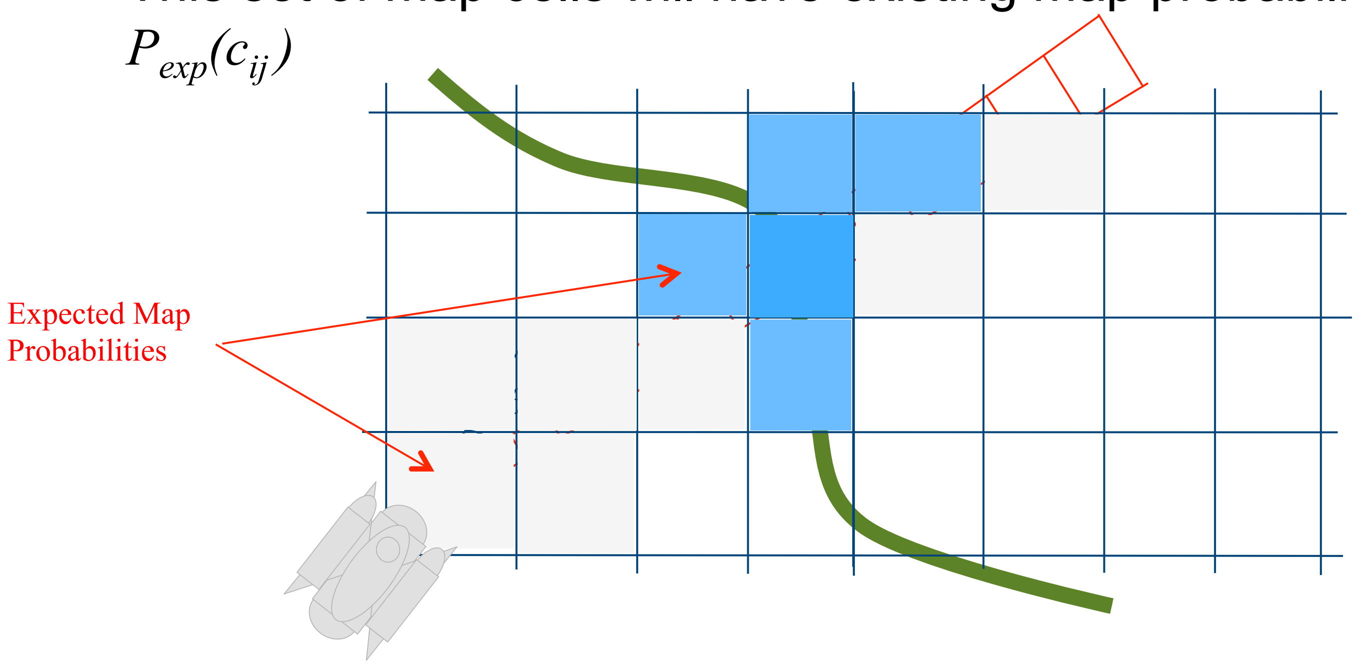
- Step 4: Calculate weights $w_t^{[i]} = P(z_t \mid \mathbf{x}_t^{[i]}, \mathbf{M}_t^{[i]})$
  - Particle weights are calculated by comparing probabilities of cell occupation from actual sonar measurements with current map cell probabilities
  - Sonar measurements come in the form

$$z = [\underbrace{\beta}_{\text{sonar angle}}\ \underbrace{s^0\ s^1\ \ldots\ s^B}_{\text{Strength of returns for increasing range}}]$$



FastSLAM for Occupancy Grids

- Step 4: Calculate weights $w_t^{[i]} = P(\, z_t \mid \mathbf{x}_t^{[i]}, \mathbf{M}_t^{[i]} \,)$
  - Given the state of the particle within a map, we can project which map cells the sonar would overlap
  - This set of map cells will have existing map probabilities $P_{exp}(c_{ij})$



FastSLAM for Occupancy Grids

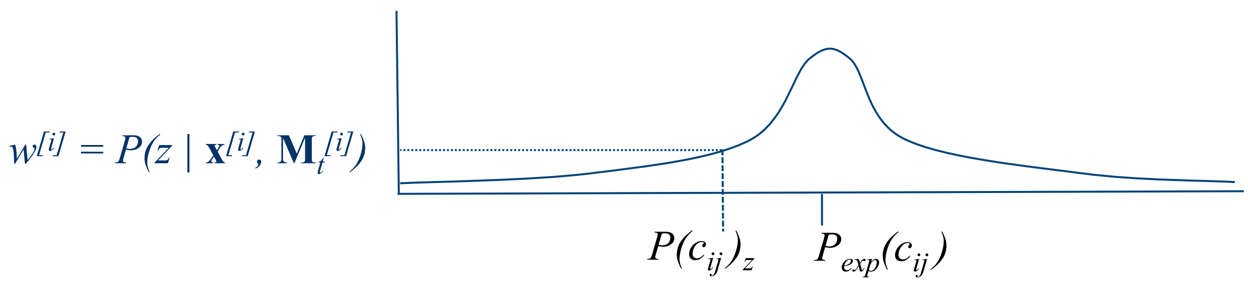
- Step 4: Calculate weights $w_t^{[i]} = P(z_t \mid \mathbf{x}_t^{[i]}, \mathbf{M}_t^{[i]})$
  - Given the **actual sensor signal strengths** s corresponding to each map cell, one can calculate a probability of a cell being occupied.

$$P(c_{ij})_z = K_s\, s$$

Where K_s is a scalar that maps signal strength to probability.
(e.g. = $1/s_{max}$)

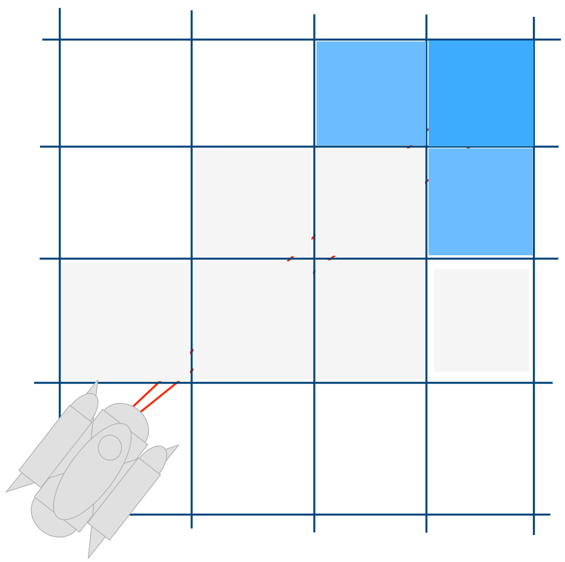
FastSLAM for Occupancy Grids

- Step 4: Calculate weights $w_t^{[i]} = P(z_t \mid \mathbf{x}_t^{[i]},\mathbf{M}_t^{[i]})$
  - To calculate the particles weight $w^{[i]}$, we compare the expected map probabilities $P_{exp}(c_{ij})$ based on the current map, with the sensor based probabilities $P(c_{ij})_z$



FastSLAM for Occupancy Grids

- Step 5: Update $\mathbf{M}_t^{[i]}$
  - Modify the occupancy likelihood of each cell $P(c_{ij})$ using sonar measurement z. We convert signal strength to a probability, and then add with the log odds!!!

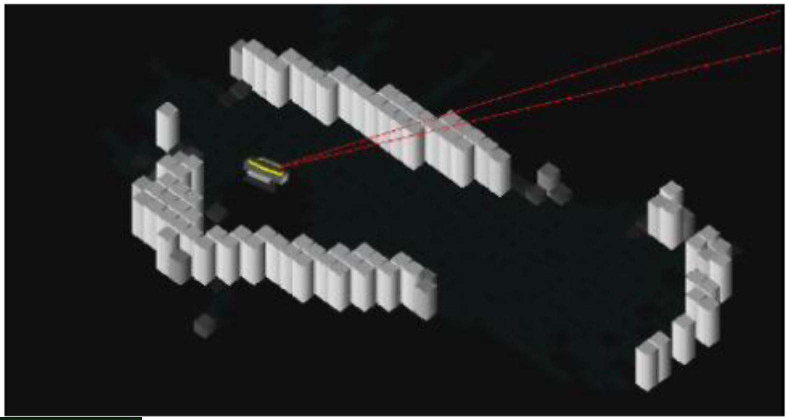


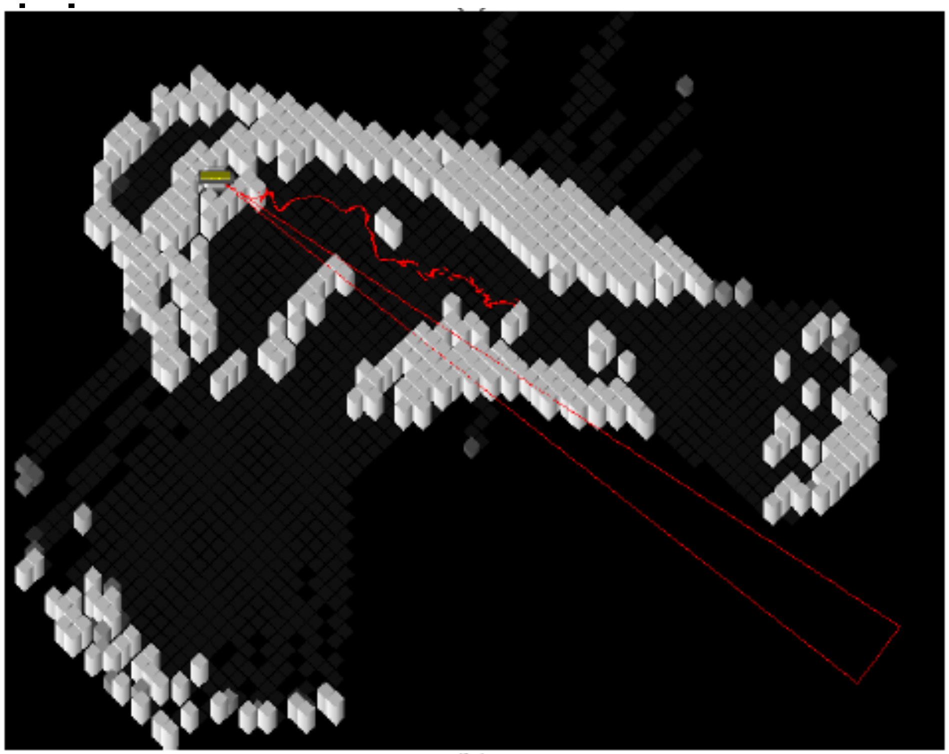
FastSLAM for Occupancy Grids

- Swimming Pool Trial

Malta Cistern Deployment

- Results II: SLAM while moving
  - SLAM with no tether m





Malta Cistern Deployment

- Results II: SLAM while moving
  - Original model

$$x_t^k = f(x_{t-1}^k, u_t)$$

  - New model

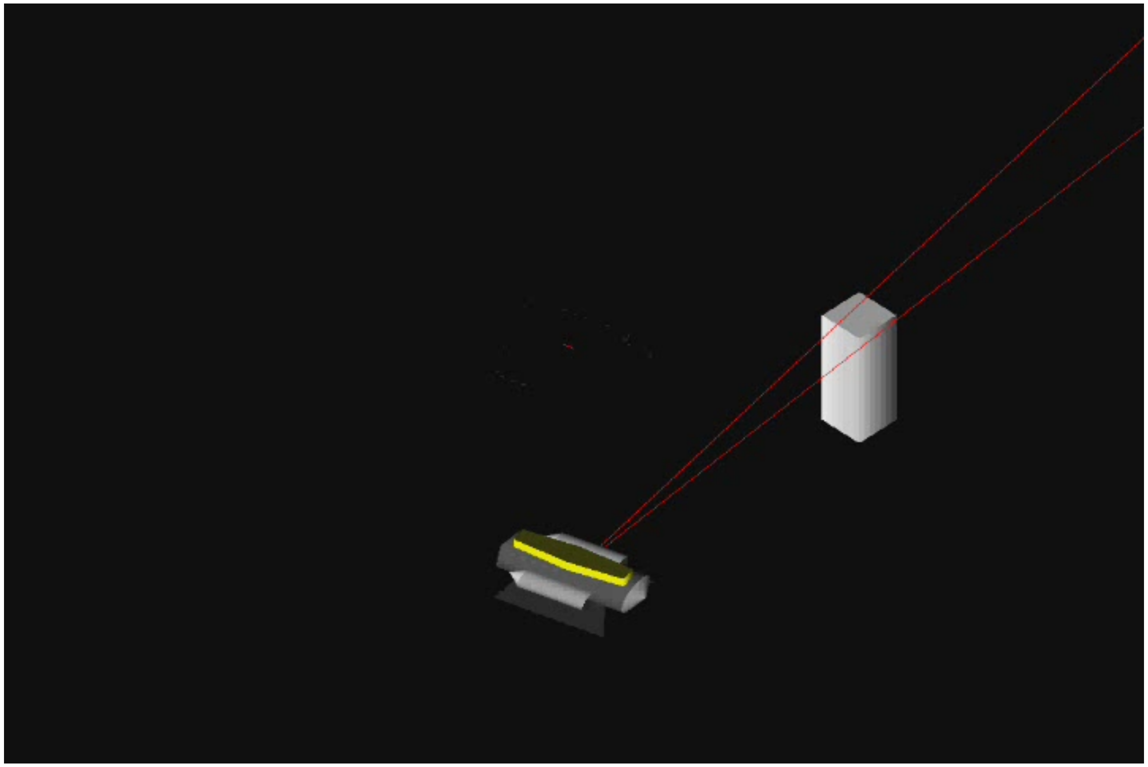
$$x_t^k = f\left(x_{t-1}^k,\ u_t(1+r_1) - \varepsilon u_t(1+r_2)\right)$$

$$\varepsilon = \begin{cases} 0 & if\ r_3 < \lambda \\ 1 & else \end{cases}$$

    - Where r_1 and r_2 are normally distributed random variables and r_3 is a uniformly distributed random variable

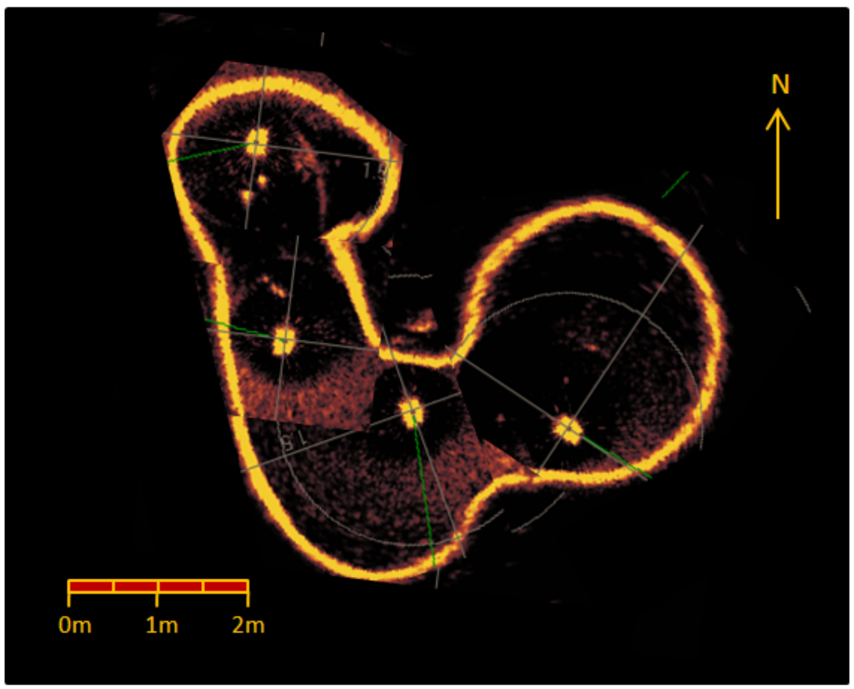
Malta Cistern Deployment

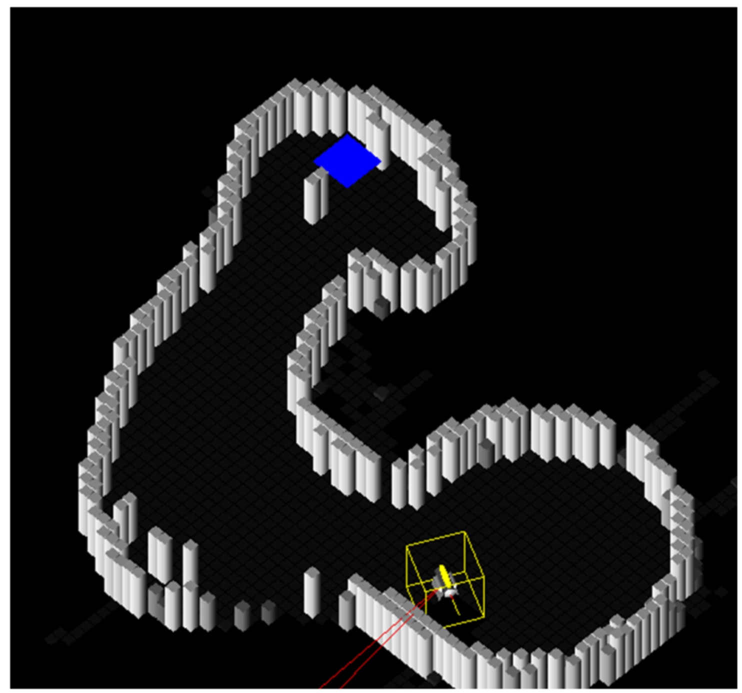
- Results II: SLAM while moving



Malta Cistern Deployment

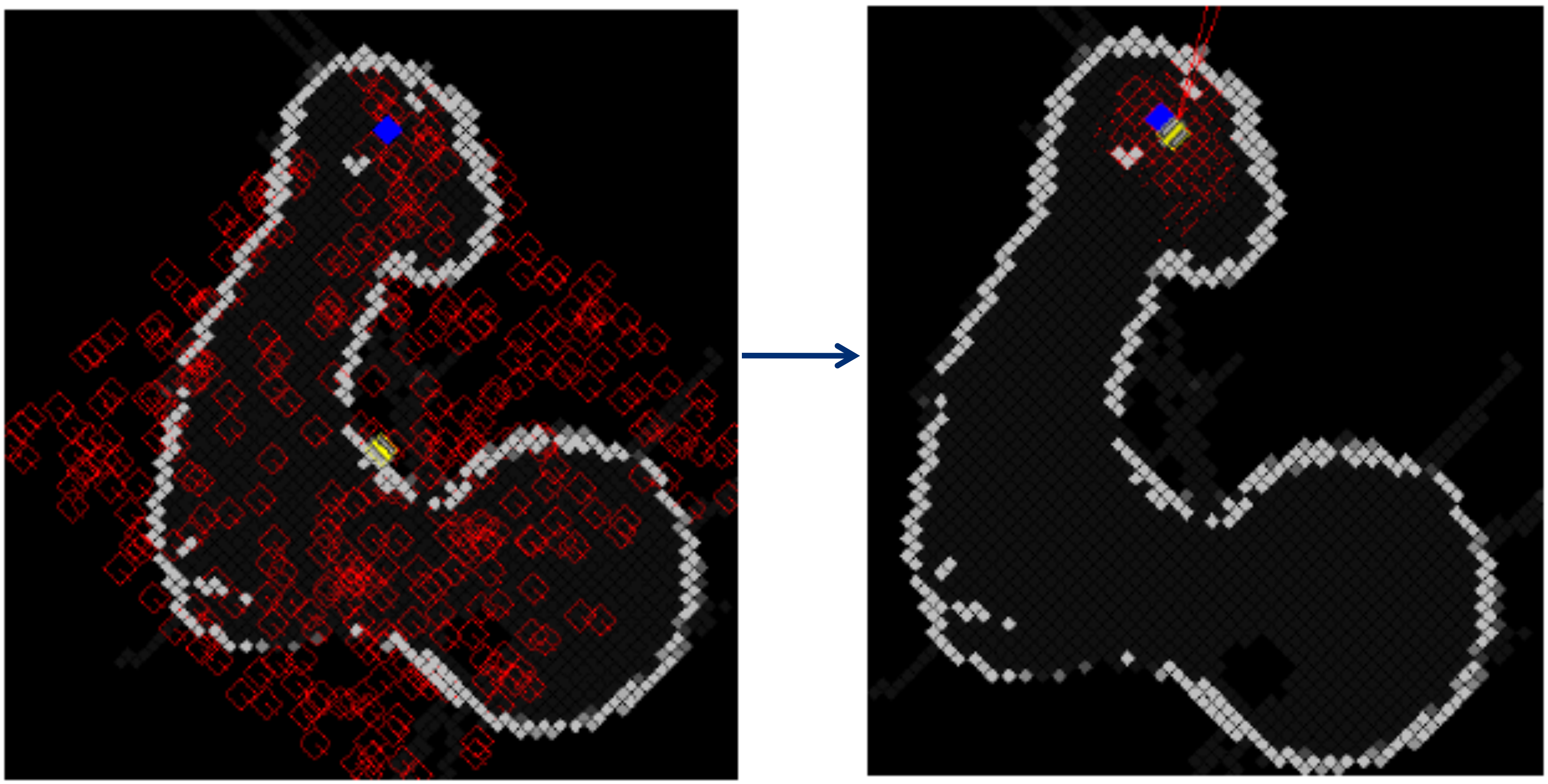
- Results III: SLAM with stationary scans





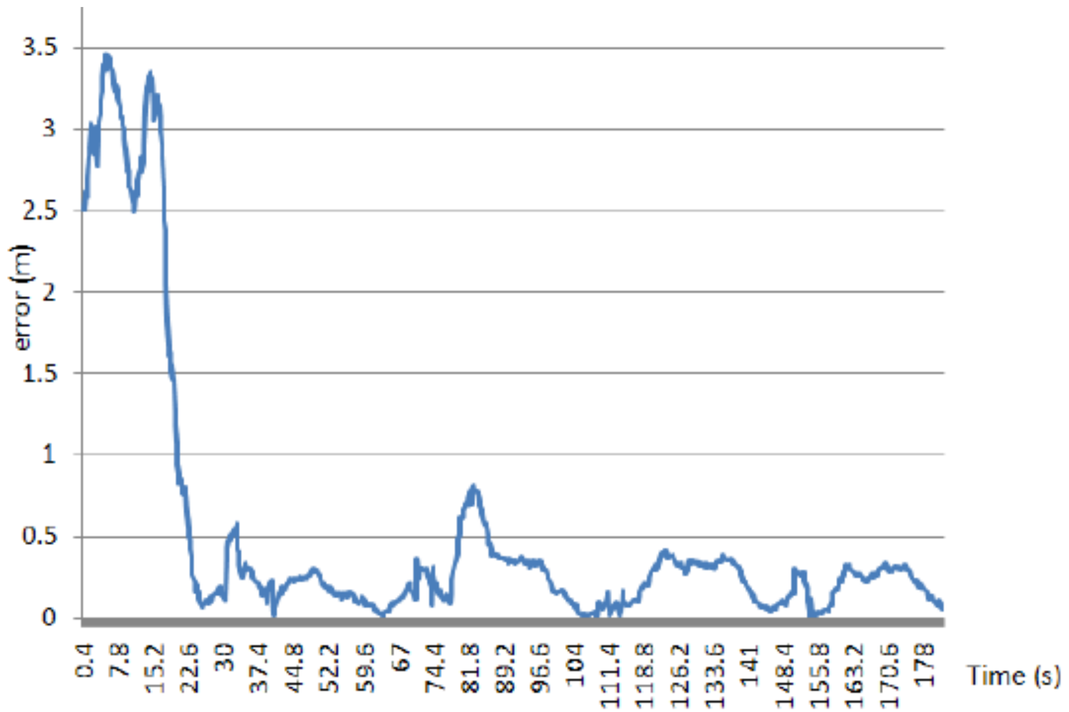
Malta Cistern Deployment

- Results IV: Localization with unknown start



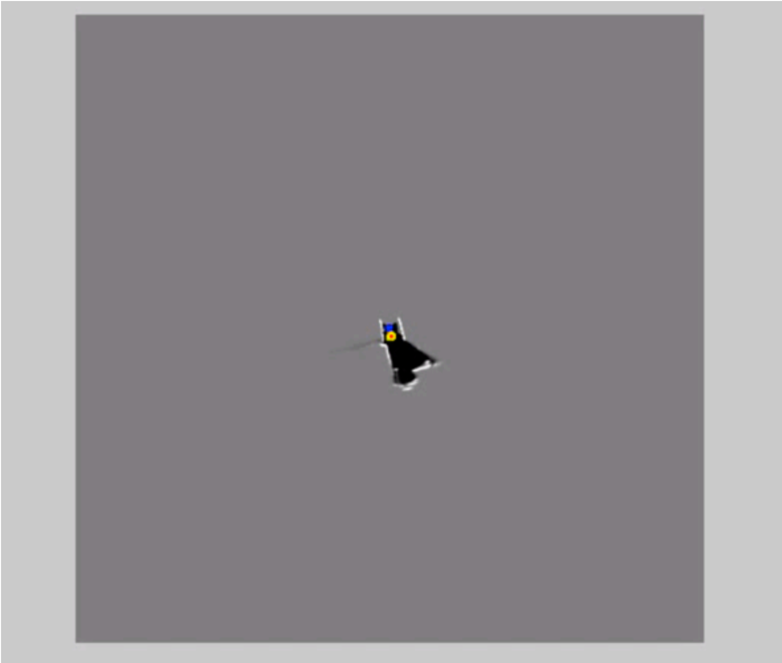
Malta Cistern Deployment

- Results IV: Localization with unknown start location



FastSLAM for Occupancy Grids

- Results IV: Localization with unknown start location



http://www.youtube.com/watch?v=1ENQtQ8nP3A