Simultaneous Localization And Mapping using Extended Kalman Filtering

Florian Breut

SLAM using EKF

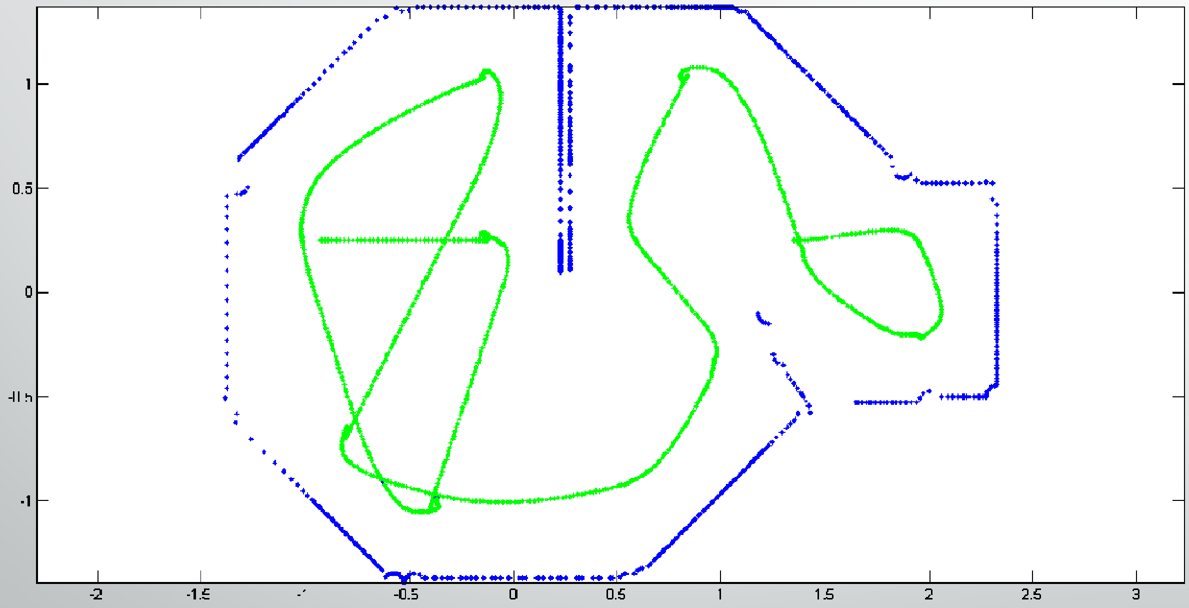
1. SLAM
   1. What is SLAM ?
   2. An easy task ?
2. EKF
   1. The Bayesian Approach
   2. Kalman Filter
   3. Extended KF
   4. SLAM EKF
3. Implementation & Results
   1. Matlab + V-rep
   2. Other Approaches

Simultaneous Localization and Mapping

PART 1

SLAM

What is SLAM ?



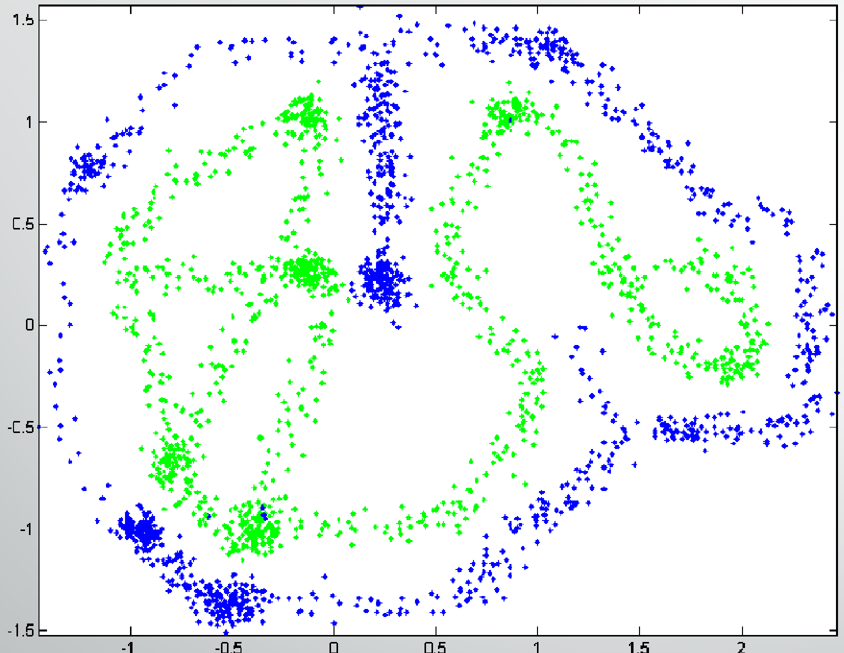
Mapping from the robot pose
OK
Robot pose given the map
OK
Both from noisy sensors
Not easy

Robot Pose

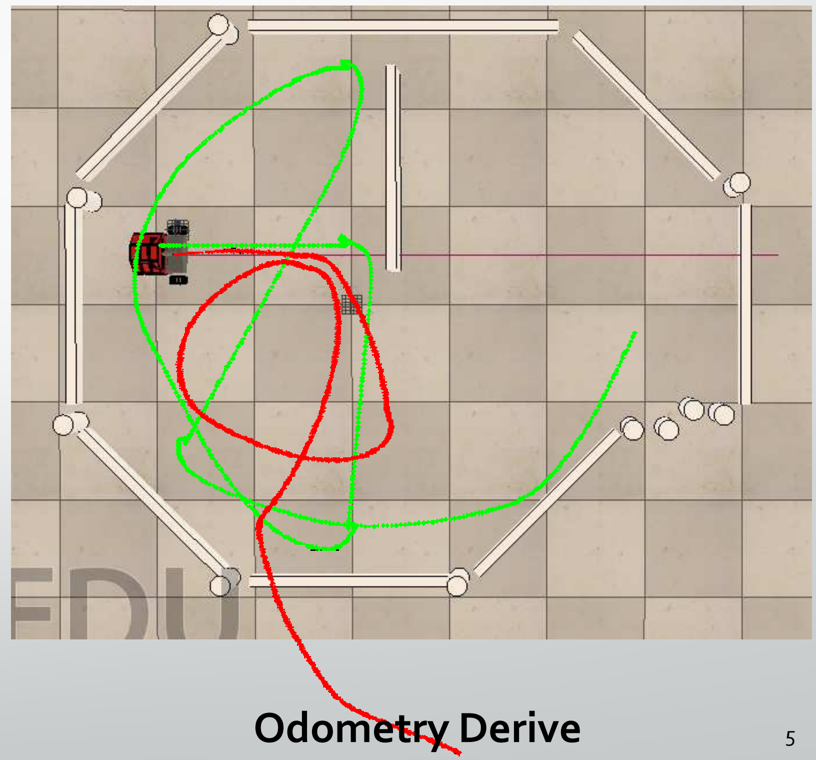
$$x_t = (x, y, \theta)$$

SLAM

An easy task ?



Sensor Noise



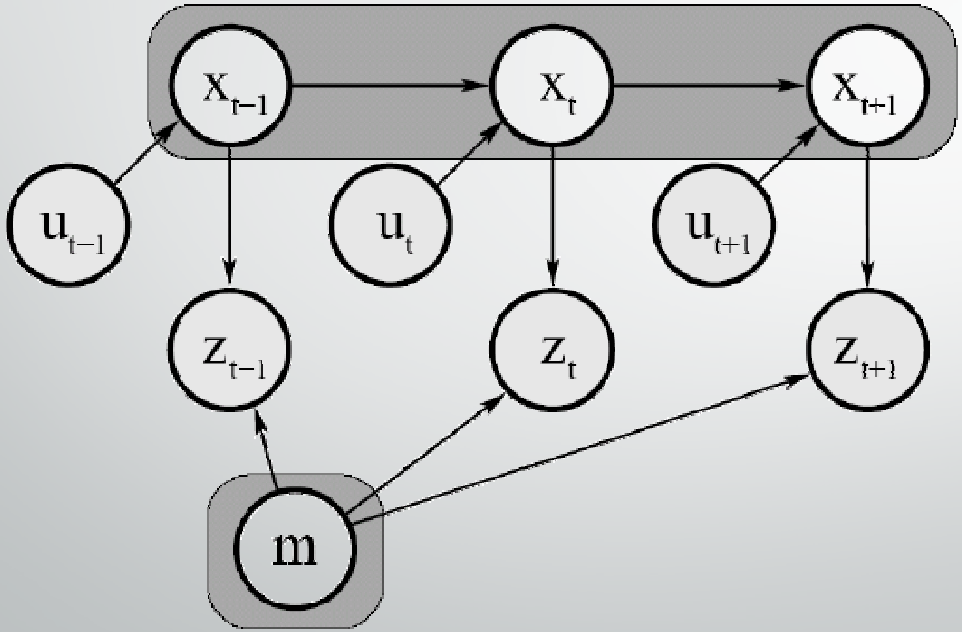
Odometry Derive

Extended Kalman Filter

PART 2

SLAM

Full SLAM Bayesian Network



Given

- The robot's Controls :

$\boldsymbol{u_{1:T}} = (\boldsymbol{u_1}, \boldsymbol{u_2}, \dots, \boldsymbol{u_T})$

- Observations :

$\boldsymbol{z_{1:T}} = (\boldsymbol{z_1}, \boldsymbol{z_2}, \dots, \boldsymbol{z_T})$

Wanted

- Map : $\boldsymbol{M}$
- Robot pose :

$\boldsymbol{x_{1:T}} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_T})$

EKF
The Bayesian Model

- **Dynamic Model**

  $$x_t = A_t x_{t-1} + B_t u_t + e_t$$

  Observation Model

  $$z_t = H_t x_{t-1} + v_t$$

  Assumptions

  Gaussian Noise

  Linear Model

Full SLAM

$$P(x_{1:t}, M | z_{1:t}, u_{1:t})$$

Online SLAM

$$P(x_t, M | z_{1:t}, u_{1:t})$$

EKF

The Bayesian Model

$$bel(x_t) = P(x_t|z_{1:t}, u_{1:t})$$

$$bel(x_t) \propto P(z_t|x_t, z_{1:t-1}, u_{1:t})P(x_t|z_{1:t-1}, u_{1:t})$$

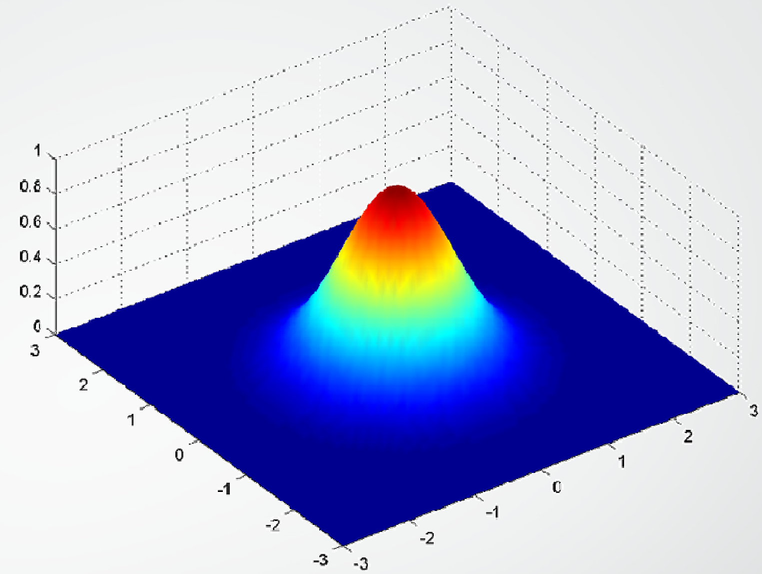
$$bel(x_t) \propto P(z_t|x_t)P(x_t|z_{1:t-1}, u_{1:t})$$

$$bel(x_t) \propto P(z_t|x_t) \int P(x_t|x_{t-1}, z_{1:t-1}, u_{1:t})P(x_{t-1}|z_{1:t-1}, u_{1:t})dx_{t-1}$$

$$bel(x_t) \propto P(z_t|x_t) \int P(x_t|x_{t-1}, u_t)P(x_{t-1}|z_{1:t-1}, u_{1:t-1})dx_{t-1}$$

$$\boxed{bel(x_t)} \propto \boxed{P(z_t|x_t)} \int \boxed{P(x_t|x_{t-1}, u_t)}\boxed{bel(x_{t-1})}dx_{t-1}$$

EKF
Gaussian Hypothesis



Observation Model

$$P(z_t|x_t) \Rightarrow \frac{1}{\sqrt{\det(2\pi Q_t)}} \exp\left(-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t)\right)$$

Motion Model

$$P(x_t|x_{t-1}, u_t) \Rightarrow \frac{1}{\sqrt{\det(2\pi R_t)}} \exp\left(-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1}(x_t - A_t x_{t-1} - B_t u_t)\right)$$

EKF
Motion Model

- **Odometry**

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \delta_{trans}\cos(\theta + \delta_{rot1}) \\ \delta_{trans}\sin(\theta + \delta_{rot1}) \\ \theta + \delta_{rot1} + \delta_{rot2} \end{pmatrix}$$

- **Velocity Model**

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{\hat{v}}{\widehat{w}}\sin\theta + \frac{\hat{v}}{\widehat{w}}\sin(\theta + w\Delta t) \\ -\frac{\hat{v}}{\widehat{w}}\cos\theta + \frac{\hat{v}}{\widehat{w}}cos(\theta + w\Delta t) \\ \widehat{w}\Delta t \end{pmatrix}$$

EKF
Observation Model

Range bearing observation

- Sonar
- Laser scan
- Triangulation with camera

$$\begin{pmatrix} \mu_{j,x} \\ \mu_{j,y} \end{pmatrix} = \begin{pmatrix} \mu_{t,x} \\ \mu_{t,y} \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\varphi_t^i + \mu_{t,\theta}) \\ r_t^i \sin(\varphi_t^i + \mu_{t,\theta}) \end{pmatrix}$$

EKF

Kalman Filter Algorithm

$$\textbf{Kalman\_filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t):$$

Prediction

$$\bar{\mu}_t = A_t\, \mu_{t-1} + B_t\, u_t$$
$$\bar{\Sigma}_t = A_t\, \Sigma_{t-1}\, A_t^T + R_t$$

Correction

$$K_t = \bar{\Sigma}_t\, C_t^T (C_t\, \bar{\Sigma}_t\, C_t^T + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t\, \bar{\mu}_t)$$
$$\Sigma_t = (I - K_t\, C_t)\, \bar{\Sigma}_t$$

$$return\ \mu_t, \Sigma_t$$

EKF

Extended Kalman Filter

Observation Model

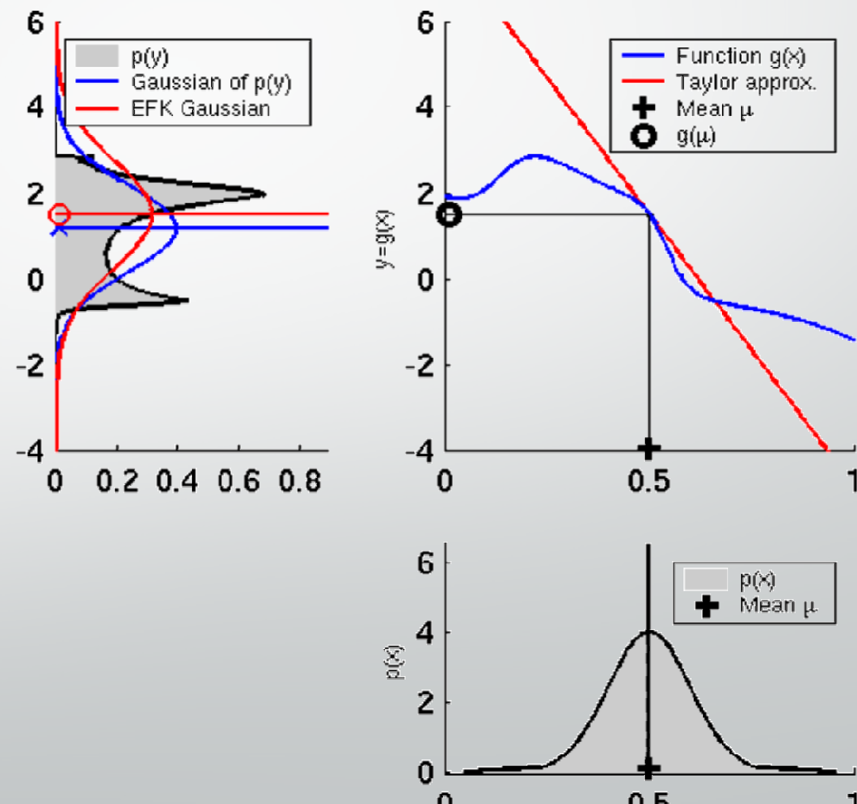
$$z_t = h(x_t) + v_t$$

$$h(x_t) \approx h(\mu_t) + \mathrm{H_t}(\mathrm{x_t} - \mu_t)$$

Dynamic Model

$$x_t = g(x_{t-1}, u_t) + e_t$$

- $g(x_{t-1}, u_t) \approx g(\mu_{t-1}, u_t) + \mathrm{G_t}(\mathrm{x_{t-1}} - \mu_{t-1})$



EKF
Extended Kalman Filter

$$\textbf{Extended\_Kalman\_filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\textbf{:}$$

$$
\begin{aligned}
&\bar{\mu}_t = g(u_t, \mu_{t-1}) \\
&\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t
\end{aligned}
$$

$$
\begin{aligned}
&K_t = \bar{\Sigma}_t \, H_t^T (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1} \\
&\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)) \\
&\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t \\
&return \ \mu_t, \Sigma_t
\end{aligned}
$$

SLAM with EKF

EKF SLAM algorithm

- N landmarks
- New landmark proposal at each t
- ML function for landmark rejection
- Extension of the state representation

SLAM with EKF

EKF SLAM algorithm

- **Extension of the state vector**

$$x_t = (x, y, \theta) \quad => \quad x_t = (\boxed{x, y, \theta}, m_{x1}, m_{y1}, \dots, m_{xn}, m_{yn})$$

- **Extension of the belief representation**

$$\begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} & \Sigma_{x\theta} \\ \Sigma_{yx} & \Sigma_{yy} & \Sigma_{y\theta} \\ \Sigma_{\theta x} & \Sigma_{\theta y} & \Sigma_{\theta\theta} \end{pmatrix} => \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} & \Sigma_{x\theta} & \Sigma_{xmx1} & \Sigma_{xmy1} & \cdots & \Sigma_{xmxn} & \Sigma_{xmyn} \\ \Sigma_{yx} & \Sigma_{yy} & \Sigma_{y\theta} & \Sigma_{ymx1} & \Sigma_{ymy1} & \cdots & \Sigma_{ymxn} & \Sigma_{ymyn} \\ \Sigma_{\theta x} & \Sigma_{\theta y} & \Sigma_{\theta\theta} & \Sigma_{\theta xm1} & \Sigma_{\theta my1} & \cdots & \Sigma_{\theta mxn} & \Sigma_{\theta myn} \\ \Sigma_{mx1x} & \Sigma_{mx1y} & \Sigma_{mx1\theta} & \Sigma_{mx1mx1} & \Sigma_{mx1my1} & \cdots & \Sigma_{mx1mxn} & \Sigma_{mx1myn} \\ \Sigma_{my1x} & \Sigma_{my1y} & \Sigma_{my1\theta} & \Sigma_{my1mx1} & \Sigma_{my1my1} & \cdots & \Sigma_{my1mxn} & \Sigma_{my1myn} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \Sigma_{mxnx} & \Sigma_{mxny} & \Sigma_{mxn\theta} & \Sigma_{mxnmx1} & \Sigma_{mxnmy1} & \cdots & \Sigma_{mxnmxn} & \Sigma_{mxnmyn} \\ \Sigma_{mynx} & \Sigma_{myny} & \Sigma_{myn\theta} & \Sigma_{mynmx1} & \Sigma_{mynmy1} & \cdots & \Sigma_{mynmxn} & \Sigma_{mynmyn} \end{pmatrix}$$

SLAM with EKF

EKF SLAM algorithm

- **Motion Model** $x_t = x_{t-1} + F_x^T \begin{pmatrix} -\frac{\hat{v}}{\hat{w}}\sin\theta + \frac{\hat{v}}{\hat{w}}\sin(\theta + w\Delta t) \\ -\frac{\hat{v}}{\hat{w}}\cos\theta + \frac{\hat{v}}{\hat{w}} cos(\theta + w\Delta t) \\ \hat{w}\Delta t \end{pmatrix} + N(0, F_x^T R_t F_x)$

With $F_x = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \end{pmatrix}$

- **Observation Model** $z_t = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ atan2(m_{j,y} - y, m_{j,x} - x) \\ m_{j,s} \end{pmatrix} + \mathrm{N}(0, \begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_\varphi & 0 \\ 0 & 0 & \sigma_s \end{pmatrix}))$

Implementation and Results

PART 3

Implementation & Simulation tools

Matlab SLAM toolbox

V-rep

Robot model simulation

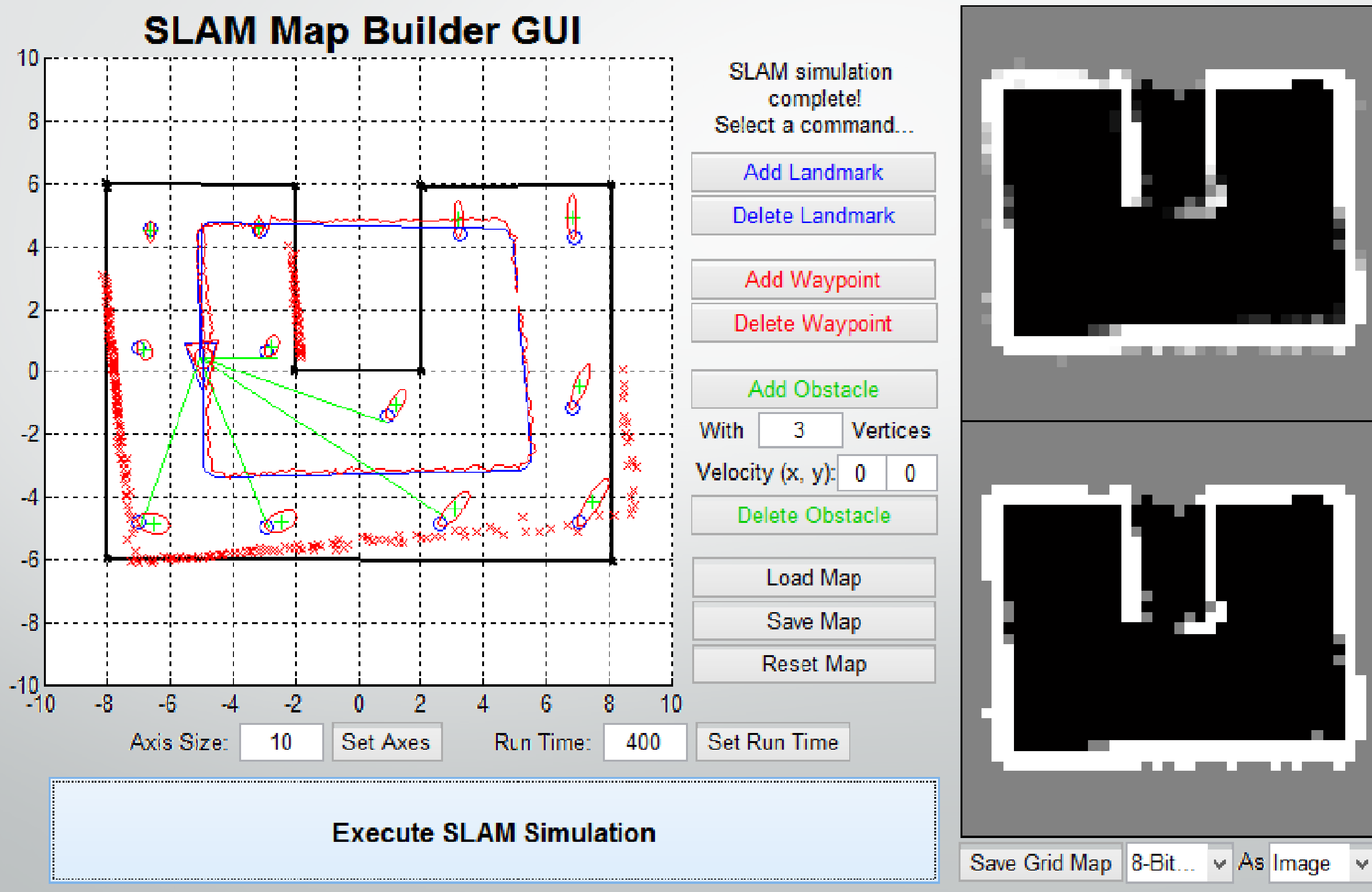
Matlab

EKF toolbox

CV Toolbox

SLAM with EKF

EKF SLAM with correspondences



Bayesian Approaches
KF family & Others

Kalman Filters

- Kalman Filter
- Extended KF
- Unscendent KF

Others

- GraphSLAM
- FastSLAM
- Sparse Extended Information Filter
- Particle Filter
- Grid localization
- Monte Carlo Localization
- SLAM 3D

Thank you