Autonomous Mobile Robots

Exercise 4: Line-based Extended Kalman Filter for Robot Localization

Timo Hinzmann, Lucas Teixeira

Kalman Filter

$$\mathbf{u}_t = [\Delta s_l, \Delta s_r]$$

State Prediction (2.1)

$$\hat{\mathbf{x}} = f(\mathbf{x}_{t-1}, \mathbf{u}_t)$$

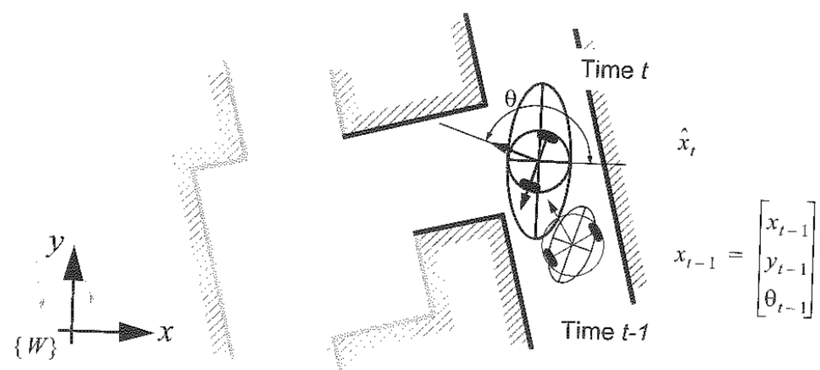
$$\mathbf{P}_t = \mathbf{F_x P}_{t-1}\mathbf{F_x}^\top + \mathbf{F_u Q}_t \mathbf{F_u}^\top$$

State Update (2.2)

- Measurement (2.2.1)
- Association (2.2.2)
- Update (2.3)

$$\mathbf{m}^i = [\alpha^i, r^i]$$

$\mathbf{P}$ $\quad$ $\mathbf{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}$ $\quad$ $\underset{2\times k}{\mathbf{M}} = \begin{bmatrix} \alpha^0 & r^0 \\ \alpha^1 & r^1 \\ \dots & \dots \end{bmatrix}$



ETH zürich

State & Covariance Prediction/Propagation

Propagation of state $\mathbf{x}$ based on $\mathbf{x}_{t-1}$ and $\mathbf{u}_t$

$$\hat{\mathbf{x}} = f(\mathbf{x}_{t-1}, \mathbf{u}_t) = \mathbf{x}_{t-1} + \begin{bmatrix} \frac{\Delta s_l + \Delta s_r}{2} \cos(\theta_{t-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_l + \Delta s_r}{2} \sin(\theta_{t-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$

Propagation of covariance $\mathbf{P}$ (Error Propagation Law)

A-Priori Cov.: $\mathbf{P}_t = \mathbf{F_x} \mathbf{P}_{t-1} \mathbf{F_x}^\top + \mathbf{F_u} \mathbf{Q}_t \mathbf{F_u}^\top$

$$\mathbf{F_x} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \theta} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \theta} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \theta} \end{bmatrix}$$

Jac. motion model wrt of state

$$\mathbf{F_u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} \end{bmatrix}$$

Jac. of motion model wrt control input

$$\mathbf{Q} = \begin{bmatrix} k|\Delta s_l| & 0 \\ 0 & k|\Delta s_r| \end{bmatrix}$$

State & Covariance Prediction/Propagation

$$\hat{\mathbf{x}} = f(\mathbf{x}_{t-1}, \mathbf{u}_t) = \mathbf{x}_{t-1} + \begin{bmatrix} \frac{\Delta s_l + \Delta s_r}{2} \cos(\theta_{t-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_l + \Delta s_r}{2} \sin(\theta_{t-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix} \qquad \mathbf{P}_t = \mathbf{F_x} \mathbf{P}_{t-1} \mathbf{F_x}^\top + \mathbf{F_u} \mathbf{Q}_t \mathbf{F_u}^\top$$

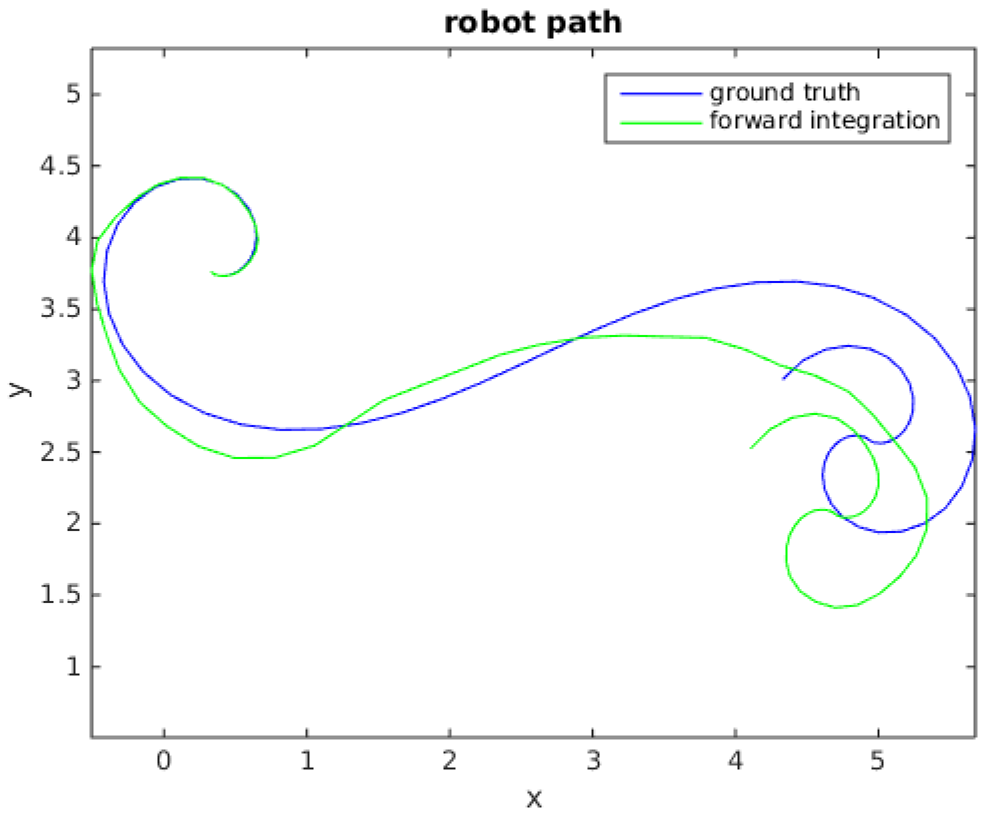
$$\mathbf{F_x} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \theta} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \theta} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \theta} \end{bmatrix} \quad \mathbf{F_u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} \end{bmatrix} \qquad \mathbf{Q} = \begin{bmatrix} k|\Delta s_l| & 0 \\ 0 & k|\Delta s_r| \end{bmatrix}$$

Tasks:

- $[\hat{\mathbf{x}}_t, \hat{\mathbf{F}}_x, \hat{\mathbf{F}}_u] = transitionFunction(\mathbf{x}_{t-1}, \mathbf{u}_t, b)$
- Derive Jacobian analytically or with MuPAD
- $validateTransitionFunction()$

```
syms x y z
sym_jac = jacobian(2*x + 3*y + 4*z, [x, y, z])
num_jac = matlabFunction(sym_jac)
```

State & Covariance Prediction/Propagation



State & Covariance Prediction/Propagation

$$\hat{\mathbf{x}} = f(\mathbf{x}_{t-1}, \mathbf{u}_t) = \mathbf{x}_{t-1} + \begin{bmatrix} \frac{\Delta s_l + \Delta s_r}{2} \cos(\theta_{t-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_l + \Delta s_r}{2} \sin(\theta_{t-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix} \qquad \mathbf{P}_t = \mathbf{F_x} \mathbf{P}_{t-1} \mathbf{F_x}^\top + \mathbf{F_u} \mathbf{Q}_t \mathbf{F_u}^\top$$

$$\mathbf{F_x} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \theta} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \theta} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \theta} \end{bmatrix} \quad \mathbf{F_u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} \end{bmatrix} \qquad \mathbf{Q} = \begin{bmatrix} k|\Delta s_l| & 0 \\ 0 & k|\Delta s_r| \end{bmatrix}$$

$$\mathbf{F_x} = \begin{bmatrix} 1 & 0 & -\frac{u_1+u_2}{2} \sin(\theta + \frac{u_2-u_1}{2b}) \\ 0 & 1 & \frac{u_1+u_2}{2} \cos(\theta + \frac{u_2-u_1}{2b}) \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{u}_t = [\Delta s_l, \Delta s_r]$$

$$\mathbf{F_u}^{11} = \frac{1}{2} \cos(\theta + \frac{u_2 - u_1}{2b}) + \frac{1}{2b} \sin(\theta + \frac{u_2 - u_1}{2b}) \frac{u_1 + u_2}{2}$$

$$\mathbf{F_u}^{12} = \frac{1}{2} \cos(\theta + \frac{u_2 - u_1}{2b}) - \frac{1}{2b} \sin(\theta + \frac{u_2 - u_1}{2b}) \frac{u_1 + u_2}{2}$$

$$\mathbf{F_u}^{21} = \frac{1}{2} \sin(\theta + \frac{u_2 - u_1}{2b}) - \frac{1}{2b} \cos(\theta + \frac{u_2 - u_1}{2b}) \frac{u_1 + u_2}{2}$$

$$\mathbf{F_u}^{22} = \frac{1}{2} \sin(\theta + \frac{u_2 - u_1}{2b}) + \frac{1}{2b} \cos(\theta + \frac{u_2 - u_1}{2b}) \frac{u_1 + u_2}{2}$$

$$\mathbf{F_u}^{31} = -1/b$$

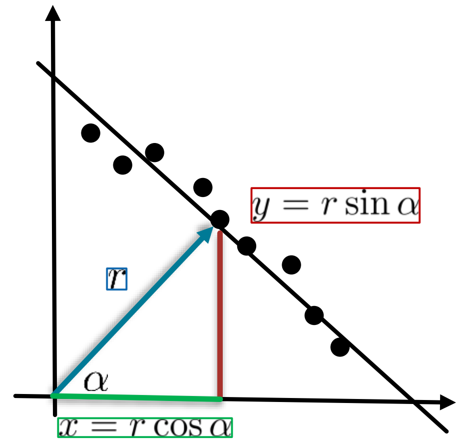
$$\mathbf{F_u}^{32} = 1/b$$

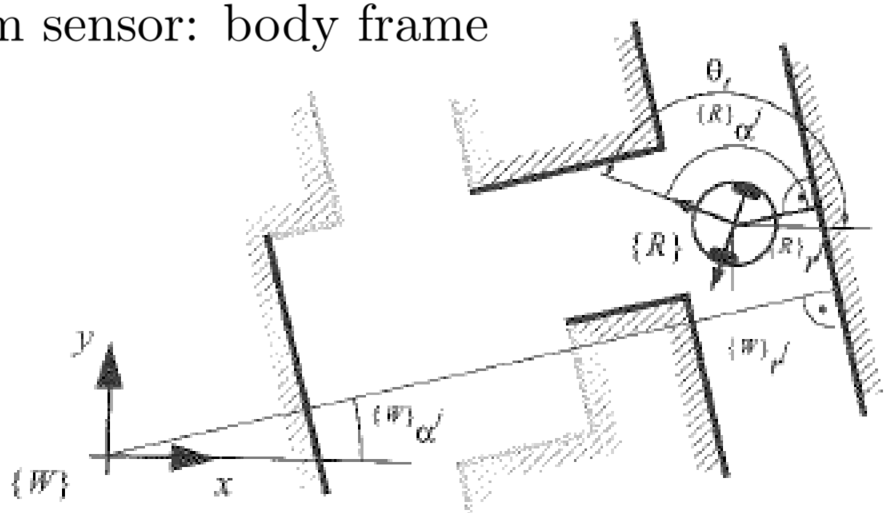
State & Covariance Prediction/Propagation

```
function [f, F_x, F_u] = transitionFunction(x,u, b)

syms x1 x2 x3
syms u1 u2
syms b_

f1 = x1 + (u1+u2)/2 * cos(x3 + (u2-u1)/(2*b_));
f2 = x2 + (u1+u2)/2 * sin(x3 + (u2-u1)/(2*b_));
f3 = x3 + (u2-u1)/b_;

f_handle = matlabFunction([f1 f2 f3]','Vars',{x1 x2 x3 u1 u2 b_});
f = f_handle(x(1),x(2),x(3),u(1),u(2),b);

df=jacobian([f1 f2 f3],[x1,x2,x3]);
Fx_handle=matlabFunction(df,'Vars',{x1 x2 x3 u1 u2 b_});
F_x = Fx_handle(x(1),x(2),x(3),u(1),u(2),b);

du=jacobian([f1 f2 f3],[u1,u2]);
Fu_handle=matlabFunction(du,'Vars',{x1 x2 x3 u1 u2 b_});
F_u = Fu_handle(x(1),x(2),x(3),u(1),u(2),b);
```

State Update: Measurement Function

- Line parametrization: $\mathbf{m}^i = \begin{bmatrix} \alpha^i & r^i \end{bmatrix}^\top$
- lines in map: world frame
- lines from sensor: body frame

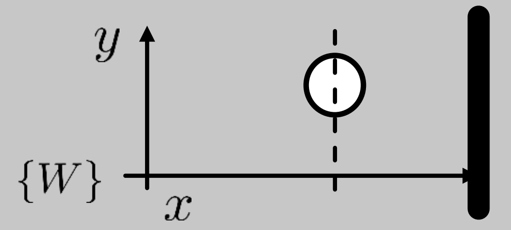




$$\hat{z}^j = \begin{bmatrix} \hat{\alpha}^j \\ \hat{r}_j \end{bmatrix} = h^j(\hat{x}, m^j) = \begin{bmatrix} {}^W\alpha^j - \hat{\theta} \\ {}^W r^j - (\hat{x}\cos({}^W\alpha^j) + \hat{y}\sin({}^W\alpha^j)) \end{bmatrix}$$

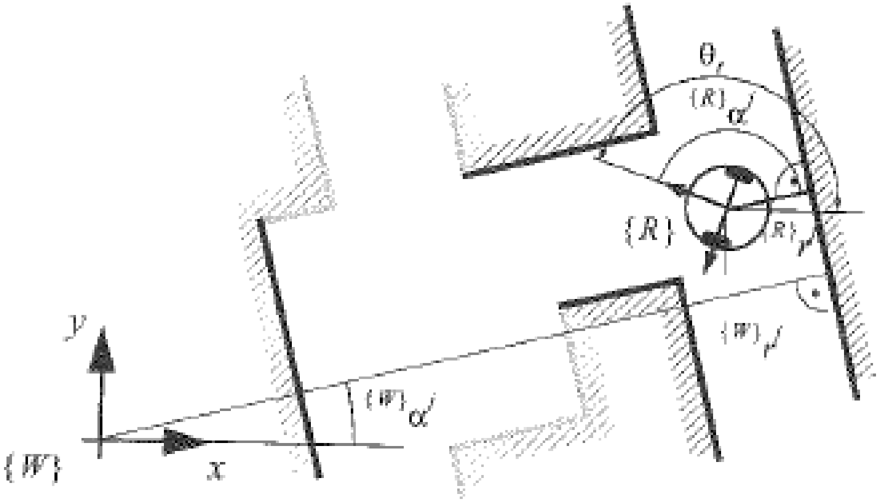
$${}^W\alpha^j = 0$$

$$\mathbf{H} = \begin{bmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} & \frac{\partial h_1}{\partial \theta} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} & \frac{\partial h_2}{\partial \theta} \end{bmatrix}$$



$$\hat{z}^j = \begin{bmatrix} \hat{\alpha}^j \\ \hat{r}_j \end{bmatrix} = \begin{bmatrix} -\hat{\theta} \\ {}^W r^j - \hat{x} \end{bmatrix}$$

State Update: Measurement Function



$$\hat{z}^j = \begin{bmatrix} \hat{\alpha}^j \\ \hat{r}_j \end{bmatrix} = h^j(\hat{x}, m^j)$$

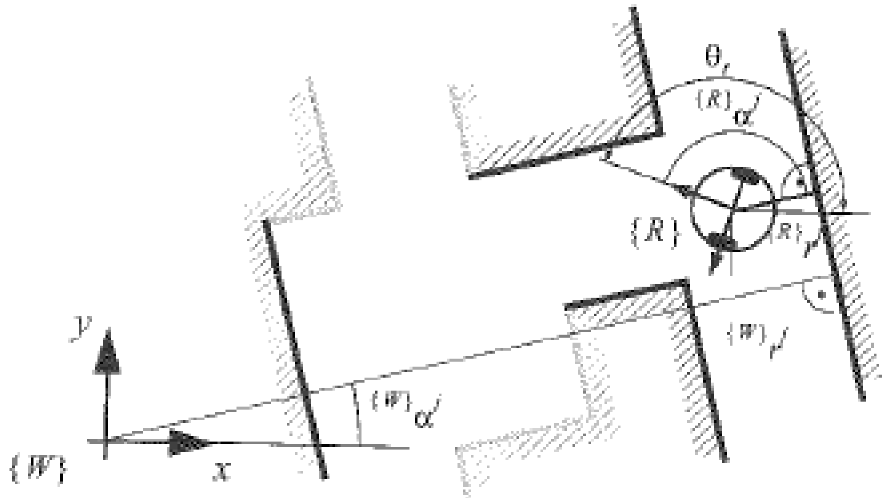
$$= \begin{bmatrix} {}^W\alpha^j - \hat{\theta} \\ {}^W r^j - (\hat{x}\cos({}^W\alpha^j) + \hat{y}\sin({}^W\alpha^j)) \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} & \frac{\partial h_1}{\partial \theta} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} & \frac{\partial h_2}{\partial \theta} \end{bmatrix}$$

Tasks:

- $[\hat{\mathbf{z}}_t, \hat{\mathbf{H}}_\mathbf{t}] = measurementFunction(\hat{\mathbf{x}}_t, \mathbf{m}^i)$
- $\hat{\mathbf{z}}_t$: Predicted observation
- $\hat{\mathbf{H}}_\mathbf{t}$: Jacobian of measurement model with respect to state
- $validateMeasurementFunction()$

State Update: Measurement Function



$$\hat{z}^j = \begin{bmatrix} \hat{\alpha}^j \\ \hat{r}_j \end{bmatrix} = h^j(\hat{x}, m^j) = \begin{bmatrix} {}^W\alpha^j - \hat{\theta} \\ {}^W r^j - (\hat{x}\cos({}^W\alpha^j) + \hat{y}\sin({}^W\alpha^j)) \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} & \frac{\partial h_1}{\partial \theta} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} & \frac{\partial h_2}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ -\cos({}^W\alpha^j) & -\sin({}^W\alpha^j) & 0 \end{bmatrix}$$

ETH zürich State Update: Measurement Association

$$\mathbf{v}_t^{ij} = \mathbf{z}_t^j - \hat{\mathbf{z}}_t^i$$

Innovation

Predicted Measurement

Observed Measurement

$$\mathbf{\Sigma}_{IN_t}^{ij} = \hat{\mathbf{H}}_t^i \, \hat{\mathbf{P}}_t (\hat{\mathbf{H}}_t^i)^\top + \mathbf{R}_t^j$$

Innovation covariance

Measurement Covariance

$$d_t^{ij} = (\mathbf{v}_t^{ij})^\top (\mathbf{\Sigma}_{IN_t}^{ij})^{-1} \mathbf{v}_t^{ij}$$

Mahalanobis distance

$$d_t^{ij} < g^2$$

Validation gate

Tasks:

- $[\hat{\mathbf{v}}_t, \hat{\mathbf{H}}_\mathbf{t}, \mathbf{R}_t] = associateMeasurements(\hat{\mathbf{x}}_t, \hat{\mathbf{P}}_t, \mathbf{Z}_t, \mathbf{R}_t, \mathbf{M}, g)$
- $validateAssociations()$

State Update

$$\mathbf{S} = \mathbf{H}\mathbf{P}_{t-1}\mathbf{H}^{\top} + \mathbf{R}$$

$$\boxed{\mathbf{K}} = \mathbf{P}_t\mathbf{H}^{\top}\mathbf{S}^{-1}$$

Kalman Gain

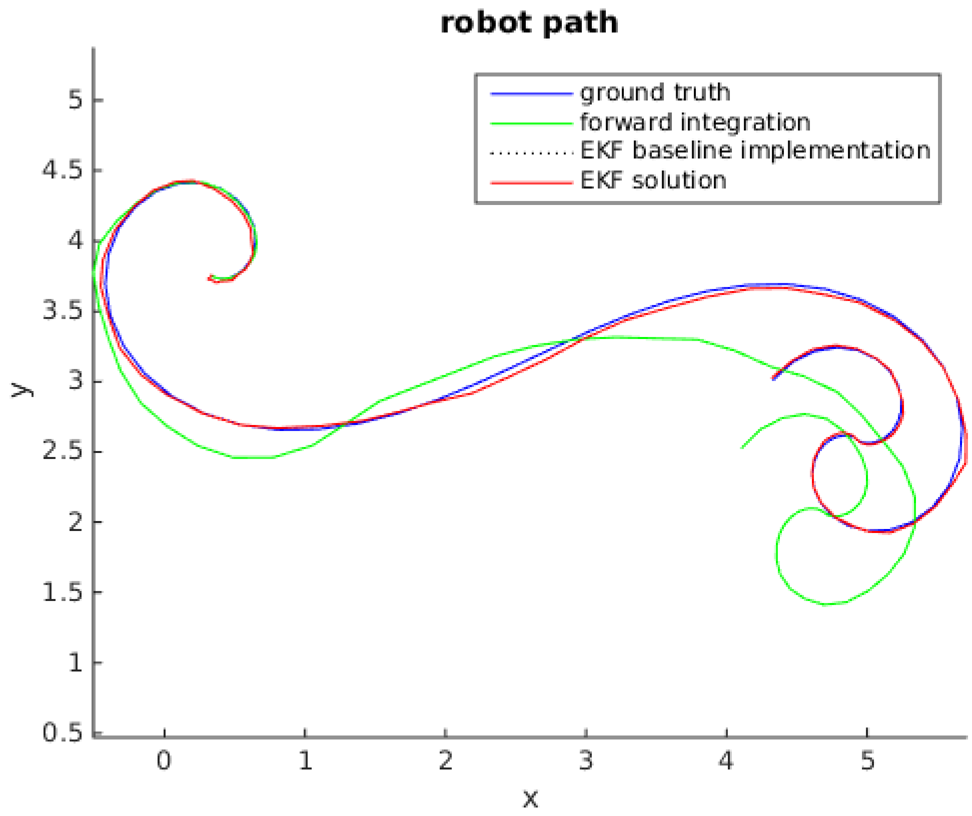
$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_{t-1}$$

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{K}\mathbf{v}$$

Tasks:

- $[\mathbf{x}_t, \mathbf{P_t}] = filterStep(\mathbf{x}_{t-1}, \mathbf{P}_{t-1}, \mathbf{u}_t, \mathbf{Z}_t, \mathbf{R}_t, \mathbf{M}, g, b)$
- $validateFilter()$
- $incrementalLocalization()$

Final EKF solution



V-Rep Simulation