RoboLab - A Search and Rescue Mission

Niklas Kaspareit, Luci Fumagalli Jana de Wiljes

SFB 1294 Spring School 2023

Motivation

Motivation

Figure 1: Earthquake in Turkey (February 2023) - Rescuers and helpers are searching for survivors

Source: https://www.politico.eu/article/death-toll-in-turkey-and-syria-passes-5000-as-rescuers-race-against-time-earthquake/

Motivation

Motivation

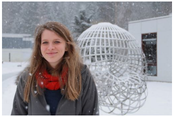
Figure 2: Football team got trapped in cave in Indonesia (July 2018) - Rescue team with their equipment at cave entrance

Source: https://en.wikipedia.org/wiki/Tham_Luang_cave_rescue

Agenda

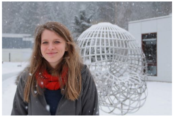
The RoboTeam

The RoboTeam



- Dr. Jana de Wiljes

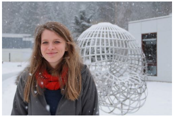
The RoboTeam



- Dr. Jana de Wiljes

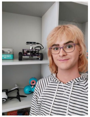
- Niklas Kaspareit

The RoboTeam



- Dr. Jana de Wiljes

- Niklas Kaspareit



- Luci Fumagalli

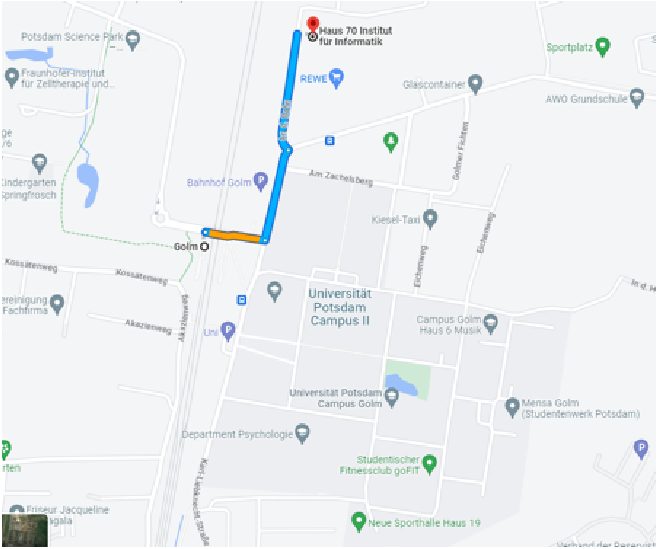
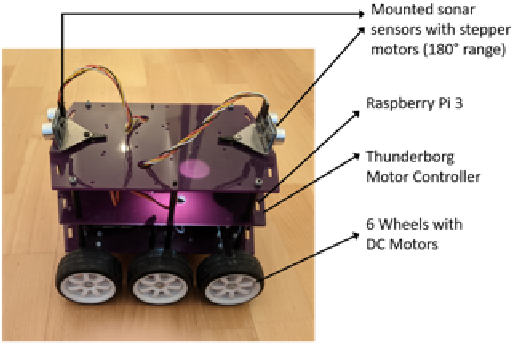


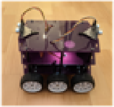
Figure 3: Our RoboLab is easily accessible from the Golm train station.

RoboLab history

RoboLab history

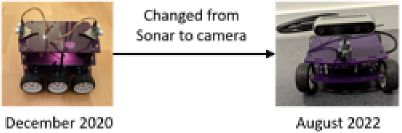


RoboLab history

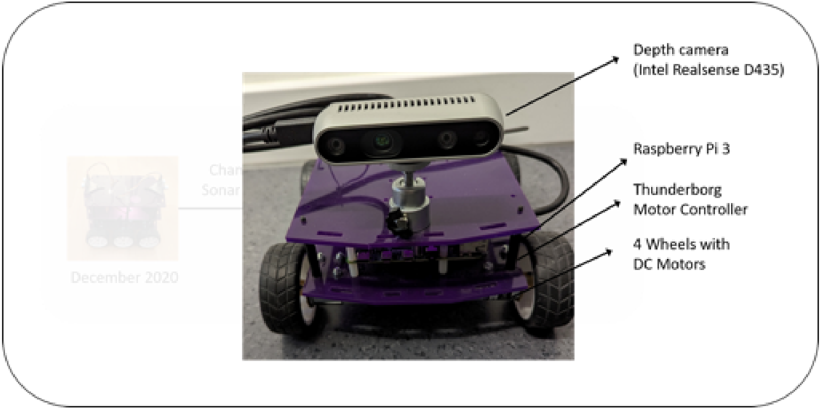


December 2020

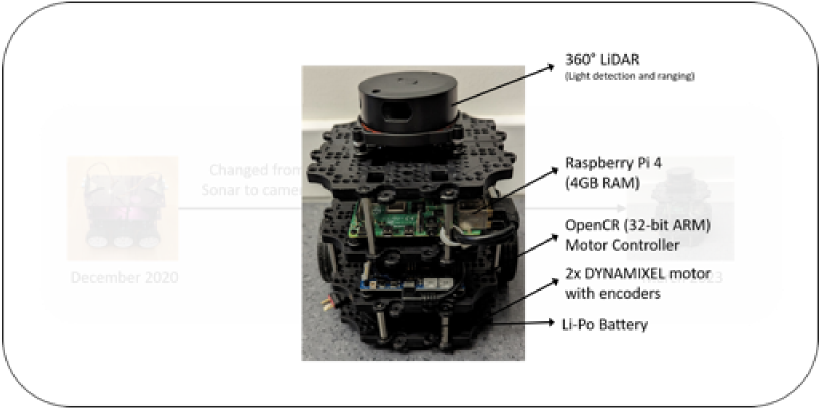
RoboLab history



RoboLab history



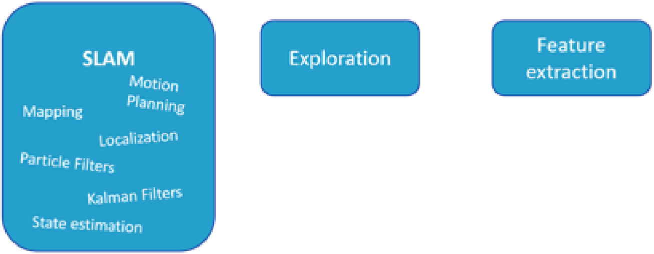
RoboLab history


360° LiDAR
(Light detection and ranging)
Raspberry Pi 4
(4GB RAM)
OpenCR (32-bit ARM)
Motor Controller
2x DYNAMIXEL motor
with encoders
Li-Po Battery
December 2020

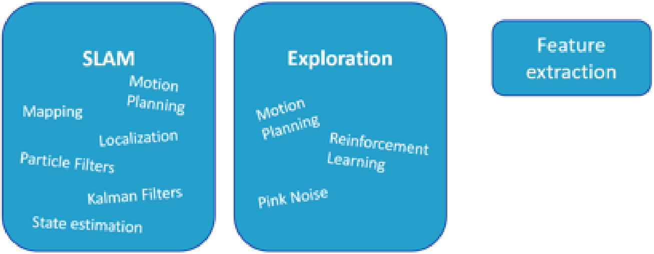
Robotics in Search and Rescue

Robotics in Search and Rescue

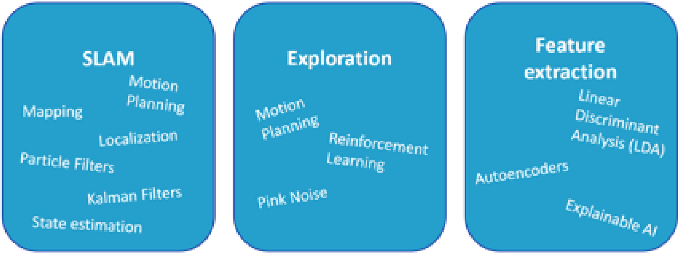
Robotics in Search and Rescue



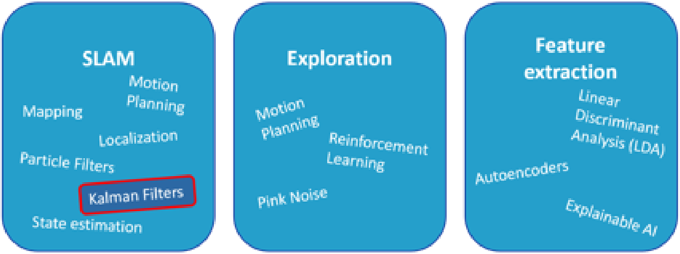
Robotics in Search and Rescue



Robotics in Search and Rescue



Robotics in Search and Rescue



ROS2 - the robotics operating system

ROS2 - the robotics operating system

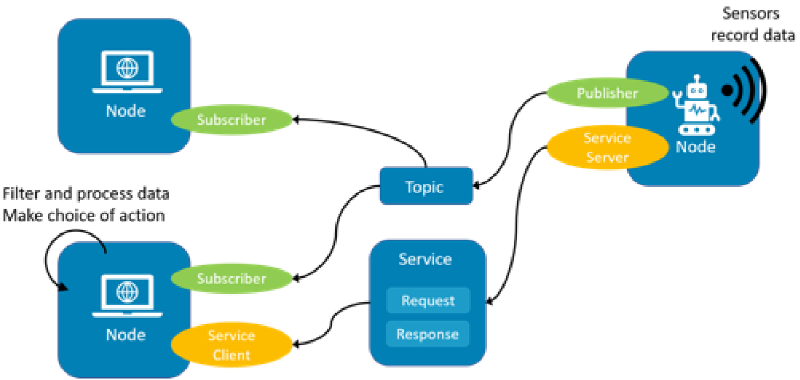


Figure 4: Ros2 Communication model

Figure 5: Map and Pose visualised in rviz on the client using the data published by the robot via a topic

Motion models

Velocity motion model

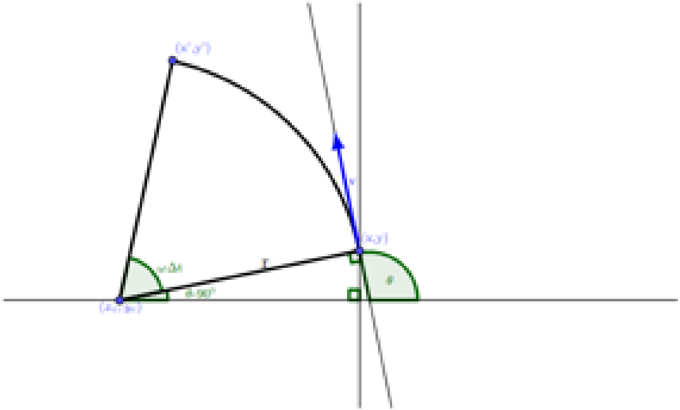


Figure 6: Representation of Velocity motion model

source: Seckler, 2D Robotic Mapping Using a Grid-Based Fast-SLAM Algorithm with Applications of Reinforcement Learning, 2020.

Velocity Motion Model

$$\text{Robot pose; } \boldsymbol{x}_t = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix}$$

$$\text{Robot control : } \boldsymbol{u}_t = \begin{pmatrix} \hat{v}_t \\ \hat{\omega}_t \end{pmatrix}$$

$$\text{measurements : } \boldsymbol{z}_t$$

Velocity Motion Model

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} + \begin{pmatrix} -\frac{\hat{v}_t}{\hat{\omega}_t} \sin\theta_t + \frac{\hat{v}}{\hat{\omega}} \sin\left(\theta_t + \omega_t \Delta t\right) \\ \frac{\hat{v}}{\hat{\omega}_t} \cos\theta_t - \frac{\hat{v}}{\hat{\omega}} \sin\left(\theta_t + \omega \Delta t\right) \\ \hat{\omega}\Delta t + \hat{\gamma}\Delta t \end{pmatrix}$$

Standard Odometry Model

- Movement from $\begin{pmatrix} x_t & y_t & \theta_t \end{pmatrix}$ to $\begin{pmatrix} x_{t+1} & y_{t+1} & \theta_{t+1} \end{pmatrix}$

Standard Odometry Model

- Movement from $\begin{pmatrix} x_t & y_t & \theta_t \end{pmatrix}$ to $\begin{pmatrix} x_{t+1} & y_{t+1} & \theta_{t+1} \end{pmatrix}$
- Odometry information: $u = \begin{pmatrix} \delta_{rot1} & \delta_{trans} & \delta_{rot2} \end{pmatrix}$

Standard Odometry Model

- Movement from $\begin{pmatrix} x_t & y_t & \theta_t \end{pmatrix}$ to $\begin{pmatrix} x_{t+1} & y_{t+1} & \theta_{t+1} \end{pmatrix}$
- Odometry information: $u = \begin{pmatrix} \delta_{rot1} & \delta_{trans} & \delta_{rot2} \end{pmatrix}$

$$\delta_{trans} = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2}$$
$$\delta_{rot1} = \text{atan2}(y_{t+1} - y_t, x_{t+1} - x_t) - \theta_t$$
$$\delta_{rot2} = \theta_{t+1} - \theta_t - \delta_{rot1}$$

Standard Odometry Model

- Movement from $\begin{pmatrix} x_t & y_t & \theta_t \end{pmatrix}$ to $\begin{pmatrix} x_{t+1} & y_{t+1} & \theta_{t+1} \end{pmatrix}$
- Odometry information: $u = \begin{pmatrix} \delta_{rot1} & \delta_{trans} & \delta_{rot2} \end{pmatrix}$

$$\delta_{trans} = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2}$$
$$\delta_{rot1} = \text{atan2}(y_{t+1} - y_t, x_{t+1} - x_t) - \theta_t$$
$$\delta_{rot2} = \theta_{t+1} - \theta_t - \delta_{rot1}$$

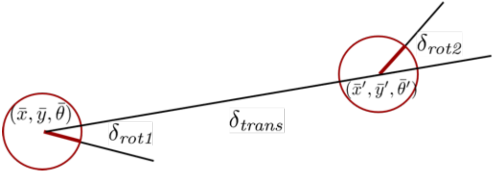
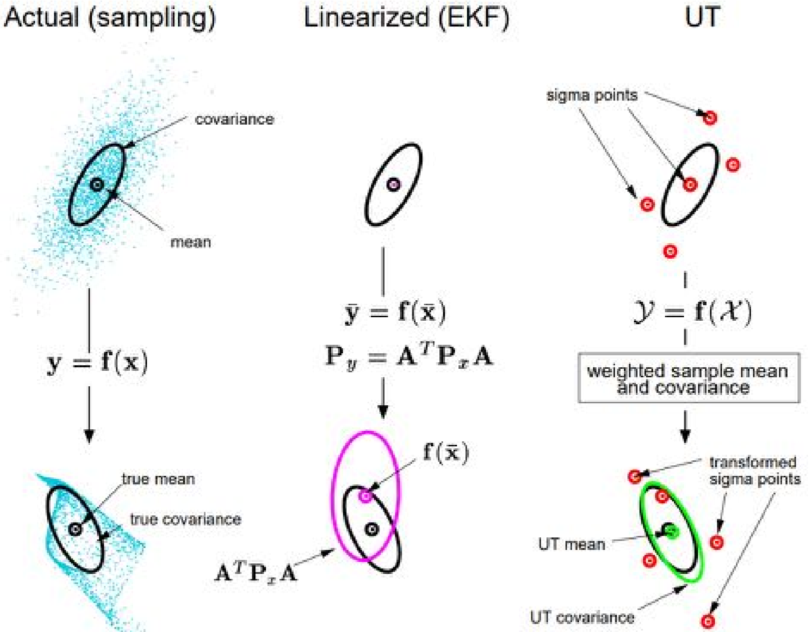


Figure 7: Odometry Motion Model representation

source: Stachniss (TU Bonn), Course: Robot Mapping, 2014

Unscented Kalman Filter

Comparison of Filtering approaches



Extended Kalman Filter

Algorithm 1 Extended Kalman Filter

$\bar{\mu}_t = g(u_t, \mu_{t-1})$

$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$

$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$

return μ_t, Σ_t

EKF to UKF (1)

~~$\bar{\mu}_t = g(u_t, \mu_{t-1})$~~

~~$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$~~

$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$

return μ_t, Σ_t

EKF to UKF (1)

1: $\bar{\mu}_t =$ calculate $\bar{\mu}$ by computing Sigma Points

2: $\bar{\Sigma}_t =$ propagation of motion of Sigma Points

3: $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$

4: $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

5: $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$

6: return μ_t, Σ_t

EKF to UKF(1) - Unscented Transform

- Sigma Points

$$\chi^{[0]} = \mu$$
$$\chi^{[i]} = \mu + (\sqrt{(n+\lambda)\Sigma})_i \quad \text{for } i = 1, \cdots, n$$
$$\chi^{[i]} = \mu - (\sqrt{(n+\lambda)\Sigma})_{i-n} \quad \text{for } i = n+1, \cdots, 2n$$

- Weights

$$w_m^{[0]} = \frac{\lambda}{n+\lambda}$$
$$w_c^{[0]} = w_m^{[0]} + (1 + \alpha^2 + \beta)$$
$$w_m^{[i]} = w_c^{[i]} = \frac{1}{2(n+\lambda)} \quad \text{for } i = 1, \cdots, 2n$$

source: Julier et al., The Scaled Unscented Transformation, 2002.

EKF to UKF(1) - Unscented Transform

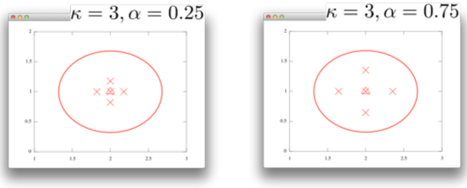


Figure 9: source: Stachniss (TU Bonn), Course: Robot Mapping, 2014

EKF to UKF(1) - Unscented Transform

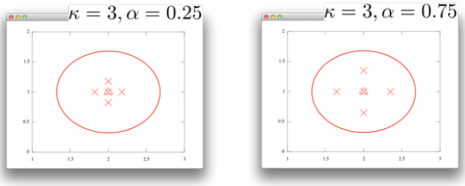


Figure 9: source: Stachniss (TU Bonn), Course: Robot Mapping, 2014

- Choice of Parameters

$$\kappa \geq 0$$
$$\alpha \in (0, 1]$$
$$\lambda = \alpha^2(n + \kappa) - n$$
$$\beta = 2 \quad \text{for Gaussians}$$

source: Julier et al., The Scaled Unscented Transformation, 2002.

EKF to UKF (1)

$\chi_{t-1} = (\mu_{t-1} \quad \mu_{t-1} + (\sqrt{(n+\lambda)\Sigma_{t-1}})_i$ for $i = 1, \cdots, n$
$\mu_{t-1} - (\sqrt{(n+\lambda)\Sigma_{t-1}})_{i-n}$ for $i = n+1, \cdots, 2n\ )$

$\bar{\chi}_t^* = g(u_t, \chi_{t-1})$

$\bar{\mu}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\chi}_t^{*[i]}$

$\bar{\Sigma}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\chi}_t^{*[i]} - \bar{\mu}_t)(\bar{\chi}_t^{*[i]} - \bar{\mu}_t)^T + R_t$

$\bar{\chi}_t = (\bar{\mu}_t \quad \bar{\mu}_t + (\sqrt{(n+\lambda)\bar{\Sigma}_t})_i$ for $i = 1, \cdots, n$
$\bar{\mu}_t - (\sqrt{(n+\lambda)\bar{\Sigma}_t})_{i-n}$ for $i = n+1, \cdots, 2n\ )$

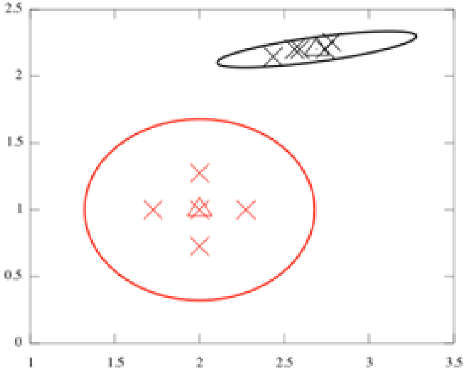
$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$

return μ_t, Σ_t

EKF to UKF (1)



$$g((x,y)^T) = \begin{pmatrix} 1 + x + \sin(2x) + \cos(y) \\ 2 + 0.2y \end{pmatrix}^T$$

source: Stachniss (TU Bonn), Course: Robot Mapping, 2014

EKF to UKF (2)

1: $\chi_{t-1} = (\mu_{t-1} \qquad \mu_{t-1} + (\sqrt{(n+\lambda)\Sigma_{t-1}})_i$ for $i = 1, \cdots, n$
$\mu_{t-1} - (\sqrt{(n+\lambda)\Sigma_{t-1}})_{i-n}$ for $i = n+1, \cdots, 2n$)

2: $\bar{\chi}_t^* = g(u_t, \chi_{t-1})$

3: $\bar{\mu}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\chi}_t^{*[i]}$

4: $\bar{\Sigma}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\chi}_t^{*[i]} - \bar{\mu}_t)(\bar{\chi}_t^{*[i]} - \bar{\mu}_t)^T + R_t$

5: $\bar{\chi}_t = (\bar{\mu}_t \qquad \bar{\mu}_t + (\sqrt{(n+\lambda)\bar{\Sigma}_t})_i$ for $i = 1, \cdots, n$
$\bar{\mu}_t - (\sqrt{(n+\lambda)\bar{\Sigma}_t})_{i-n}$ for $i = n+1, \cdots, 2n$)

6: $K_t =$ Compute expected observations and Kalman Gain using Sigma Points

7: $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

8: $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$

9: return μ_t, Σ_t

EKF to UKF (2)

6: $\bar{\mathcal{Z}} = h(\bar{\chi}_t)$

7: $\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$

8: $S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + Q_t$

9: $\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\chi}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T$

10: $K_t = \bar{\Sigma}_t^{x,z} S_t^{-1}$

11 :$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

12: $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$

13: return μ_t, Σ_t

EKF to UKF (3)

6: $\bar{\mathcal{Z}} = h(\bar{\chi}_t)$

7: $\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$

8: $S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + Q_t$

9: $\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\chi}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T$

10: $K_t = \bar{\Sigma}_t^{x,z} S_t^{-1}$

11 :$\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t))$

12: $\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$

13: return μ_t, Σ_t

1: $\chi_{t-1} = (\mu_{t-1} \quad \mu_{t-1} + (\sqrt{(n+\lambda)\Sigma_{t-1}})_i$ for $i = 1, \cdots, n$
$\mu_{t-1} - (\sqrt{(n+\lambda)\Sigma_{t-1}})_{i-n}$ for $i = n+1, \cdots, 2n$)

2: $\bar{\chi}_t^* = g(u_t, \chi_{t-1})$

3: $\bar{\mu} = \sum_{i=0}^{2n} w_m^{[i]} \bar{\chi}_t^{*[i]}$

4: $\bar{\Sigma}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\chi}_t^{*[i]} - \bar{\mu}_t)(\bar{\chi}_t^{*[i]} - \bar{\mu}_t)^T + R_t$

5: $\bar{\chi}_t = (\bar{\mu}_t \quad \bar{\mu}_t + (\sqrt{(n+\lambda)\bar{\Sigma}_t})_i$ for $i = 1, \cdots, n$
$\bar{\mu}_t - (\sqrt{(n+\lambda)\bar{\Sigma}_t})_{i-n}$ for $i = n+1, \cdots, 2n$)

6: $\bar{\mathcal{Z}} = h(\bar{\chi}_t)$

7: $\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$

8: $S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + Q_t$

9: $\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\chi}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T$

10: $K_t = \bar{\Sigma}_t^{x,z} S_t^{-1}$

Unscented Kalman Filter ii

$\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$

$\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$

return μ, Σ_t

Comparison of EKF and UKF

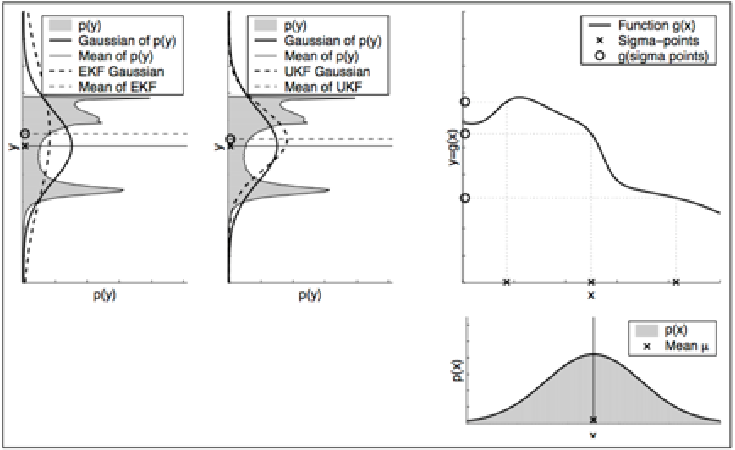


Figure 10: EKF and UKF in a normal setting of a non linear function

source: Thrun et al.: "Probabilistic Robotics", Chapter 3.4, 2005.

Comparison of EKF and UKF - Small Covariance

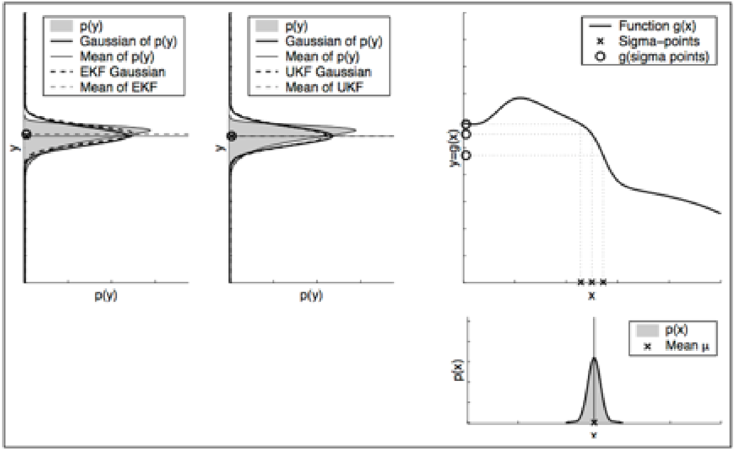


Figure 11: EKF and UKF in a setting of a non linear function with small covariance

source: Thrun et al.: "Probabilistic Robotics", Chapter 3.4, 2005.

Outlook

Outlook

- Pink Noise for exploration

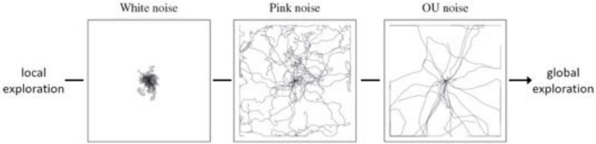


Figure 12: Pink Noise performs better in exploration tasks

source: Eberhard et al.. Pink Noise Is All You Need. 2023.

Outlook

- Pink Noise for exploration

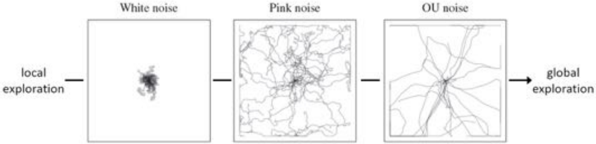


Figure 12: Pink Noise performs better in exploration tasks
source: Eberhard et al.. Pink Noise Is All You Need. 2023.

- MultiAgent Reinforcement Learning

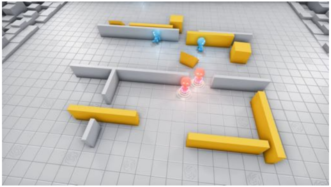


Figure 13: Multi-Agent Reinforcement Learning with Hide and Seek (OpenAI)
OpenAI. Emergent tool use from multi-agent interaction

Interested in Robotics? - Come visit us!

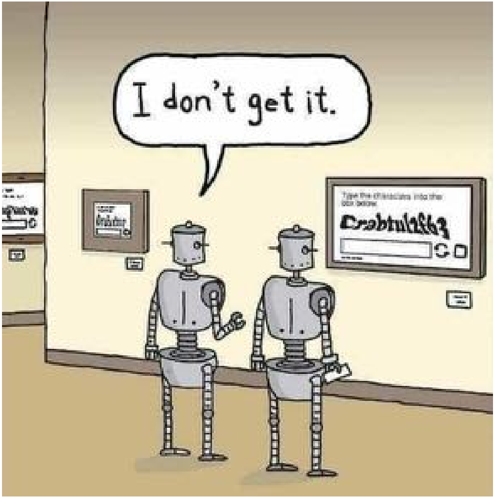


Figure 14: source: https://www.memedroid.com/memes/detail/3796469/captcha?refGallery=tags&page=1&tag=robot

EKF to UKF (3) - BACKUP

$$
\begin{aligned}
\Sigma_t &= (I - K_t H_t)\bar{\Sigma}_t \\
&= \bar{\Sigma}_t - K_t H_t \bar{\Sigma}_t \\
&= \bar{\Sigma}_t - K_t (\bar{\Sigma}^{x,z})^T \\
&= \bar{\Sigma}_t - K_t (\bar{\Sigma}^{x,z} S_t^{-1} S_t)^T \\
&= \bar{\Sigma}_t - K_t (K_t S_t)^T \\
&= \bar{\Sigma}_t - K_t S_t^T K_t^T \\
&= \bar{\Sigma}_t - K_t S_t K_t^T
\end{aligned}
$$