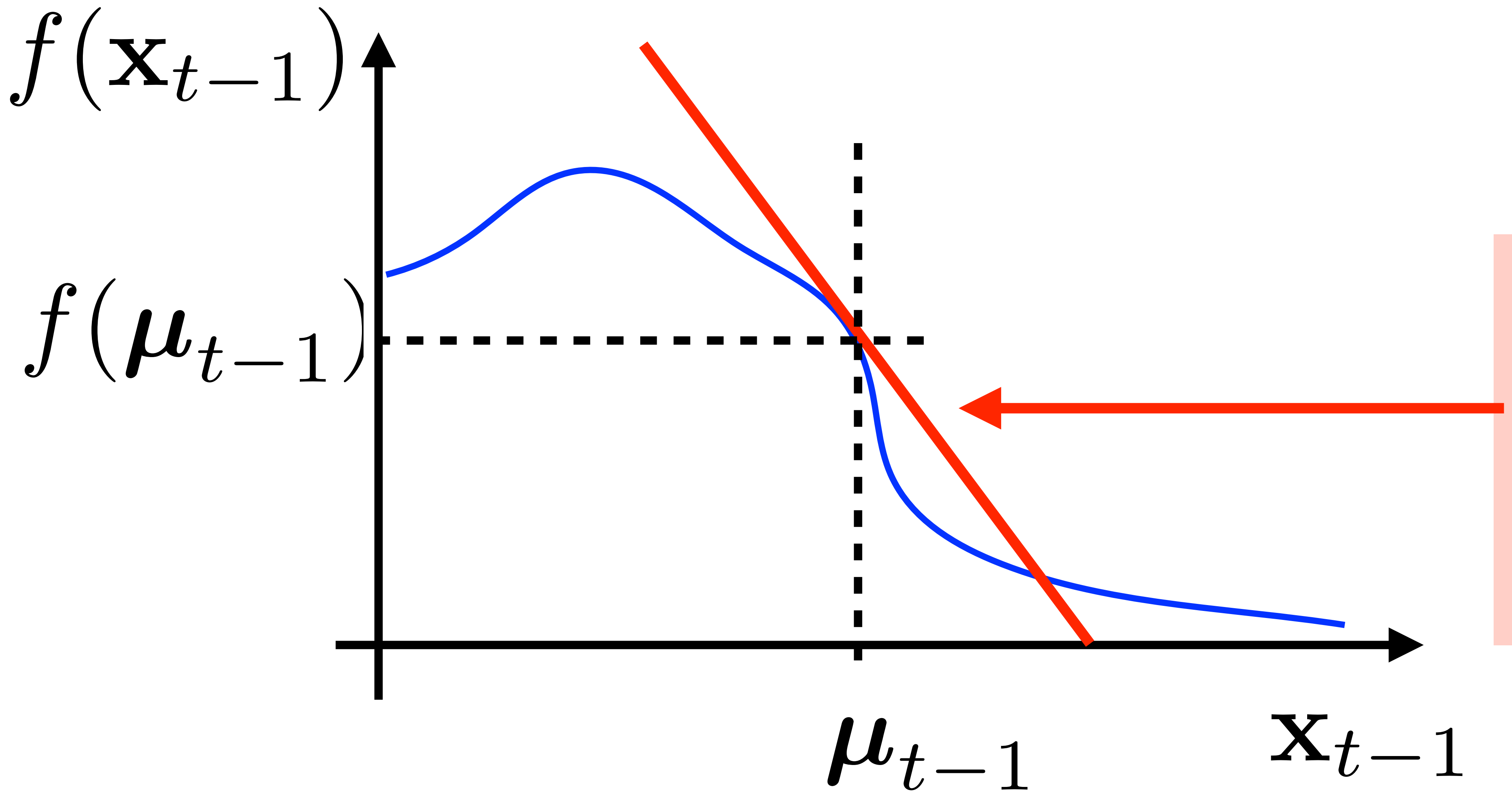
Localization - Extended Kalman filter

Karel Zimmermann

Prerequisites: Extended Kalman Filter

- First order Taylor expansion
- Jacobian



$$f(\mathbf{x}_{t-1}) \approx f(\boldsymbol{\mu}_{t-1}) + \mathbf{F}_t(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1})$$

$$\mathbf{F}_t = \frac{\partial f(\mathbf{x} = \boldsymbol{\mu}_{t-1})}{\partial \mathbf{x}}$$

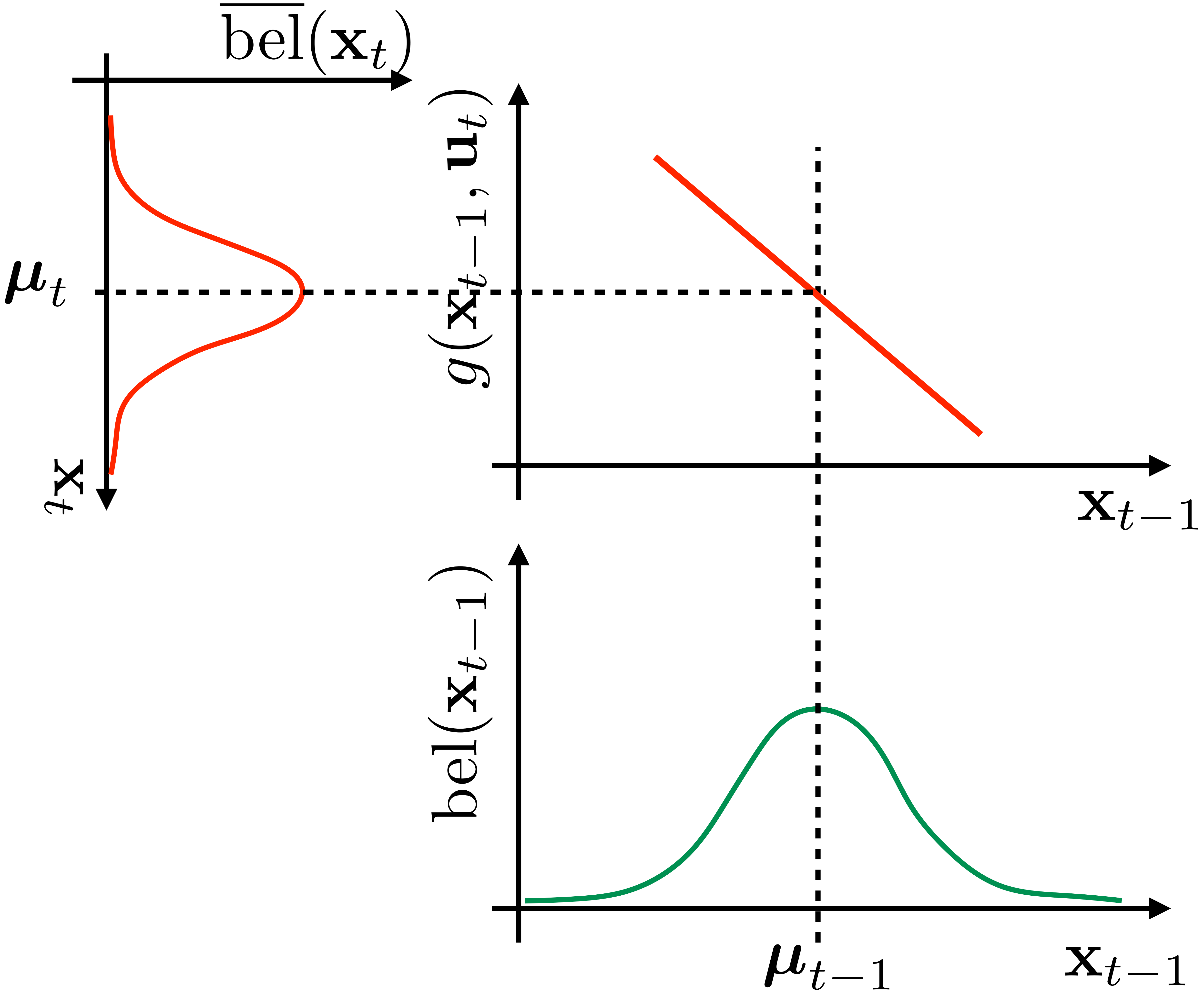
https://en.wikipedia.org/wiki/Brook_Taylor

Extended Kalman Filter

Linear system with Gaussian noise:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t) = \mathcal{N}_{\mathbf{x}_t}(\mathbf{A}_t\mathbf{x}_{t-1} + \mathbf{B}_t\mathbf{u}_t,\ \mathbf{R}_t)$$

$$p(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}_{\mathbf{z}_t}(\mathbf{C}_t\mathbf{x}_t,\ \mathbf{Q}_t)$$



Extended Kalman Filter

Non-linear system with Gaussian noise:

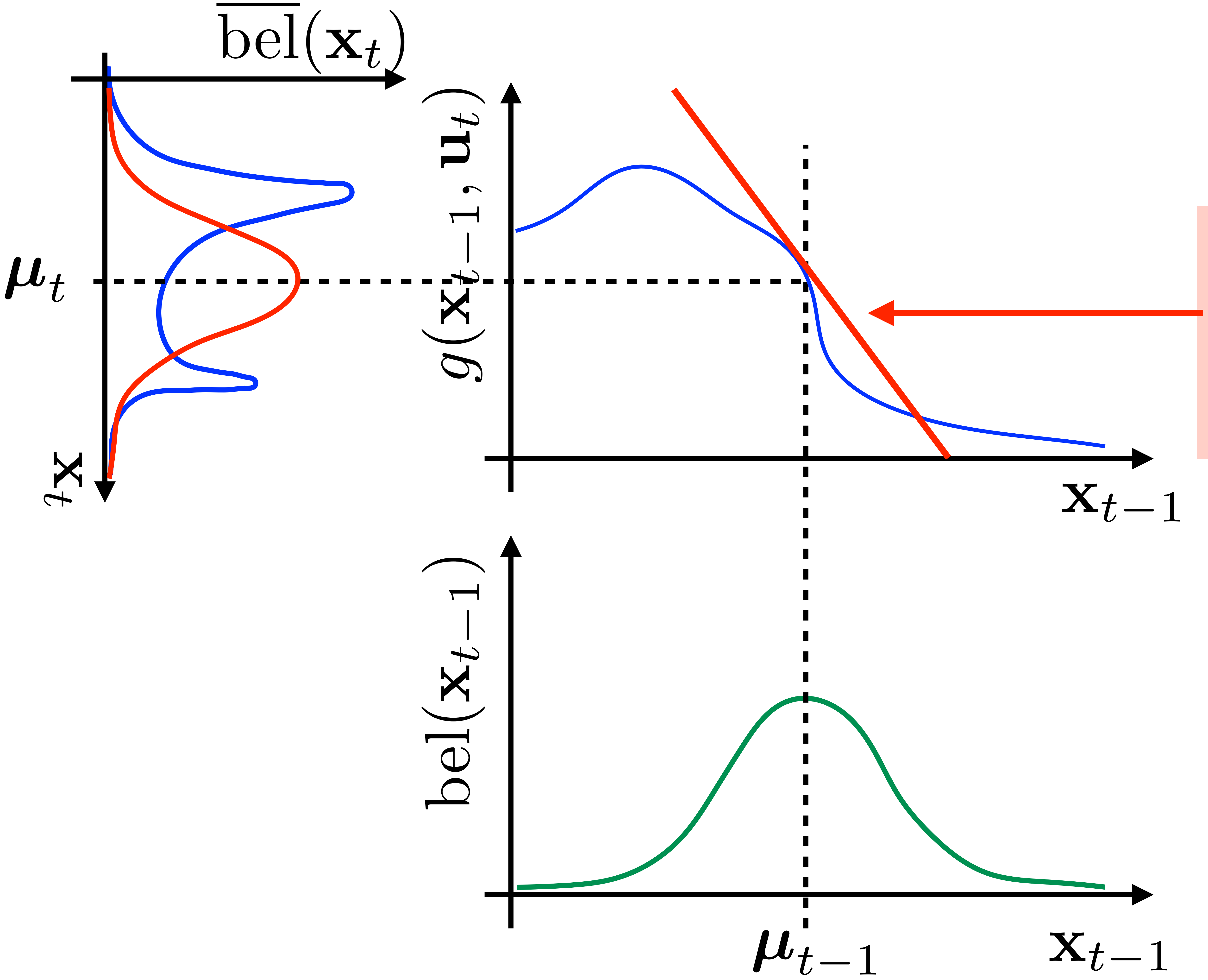
$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) = \mathcal{N}_{\mathbf{x}_t}(g(\mathbf{x}_{t-1}, \mathbf{u}_t),\ \mathbf{R}_t)$$

$$p(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}_{\mathbf{z}_t}(h(\mathbf{x}_t),\ \mathbf{Q}_t)$$

Linearized system with Gaussian noise:

$$\approx \mathcal{N}_{\mathbf{x}_t}(g(\mathbf{u}_t, \boldsymbol{\mu}_{t-1}) + \mathbf{G}_t(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}),\ \mathbf{R}_t)$$

$$\approx \mathcal{N}_{\mathbf{z}_t}(h(\overline{\boldsymbol{\mu}}_t) + \mathbf{H}_t(\mathbf{x}_t - \overline{\boldsymbol{\mu}}_t),\ \mathbf{Q}_t)$$



$$g(\mathbf{u}_t, \mathbf{x}_{t-1}) \approx g(\mathbf{u}_t, \boldsymbol{\mu}_{t-1}) + \mathbf{G}_t(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1})$$

$$\mathbf{G}_t = \frac{\partial g(\mathbf{u} = \mathbf{u}_t, \mathbf{x} = \boldsymbol{\mu}_{t-1})}{\partial \mathbf{x}}$$

Extended Kalman Filter

Linear system with Gaussian noise:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t) = \mathcal{N}_{\mathbf{x}_t}(\mathbf{A}_t\mathbf{x}_{t-1} + \mathbf{B}_t\mathbf{u}_t,\ \mathbf{R}_t)$$

$$p(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}_{\mathbf{z}_t}(\mathbf{C}_t\mathbf{x}_t,\ \mathbf{Q}_t)$$

1. Initialization: $\mathrm{bel}(\mathbf{x}_0)$
2. Prediction step:
   $$\overline{\boldsymbol{\mu}}_t = \mathbf{A}_t\boldsymbol{\mu}_{t-1} + \mathbf{B}_t\mathbf{u}_t$$
   $$\overline{\boldsymbol{\Sigma}}_t = \mathbf{A}_t\boldsymbol{\Sigma}_{t-1}\mathbf{A}_t^\top + \mathbf{R}_t$$
   $$\overline{\mathrm{bel}}(\mathbf{x}_t) = \mathcal{N}_{\mathbf{x}_t}(\overline{\boldsymbol{\mu}}_t, \overline{\boldsymbol{\Sigma}}_t)$$
3. Measurement update:
   $$\mathbf{K}_t = \overline{\boldsymbol{\Sigma}}_t\mathbf{C}_t^\top(\mathbf{C}_t\overline{\boldsymbol{\Sigma}}_t\mathbf{C}_t^\top + \mathbf{Q}_t)^{-1}$$
   $$\boldsymbol{\mu}_t = \overline{\boldsymbol{\mu}}_t + \mathbf{K}_t(\mathbf{z}_t - \mathbf{C}_t\overline{\boldsymbol{\mu}}_t)$$
   $$\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{C}_t)\overline{\boldsymbol{\Sigma}}_t$$
   $$\mathrm{bel}(\mathbf{x}_t) = \mathcal{N}_{\mathbf{x}_t}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$$
4. Repeat from 2

Linearized system with Gaussian noise:

$$\approx \mathcal{N}_{\mathbf{x}_t}(g(\mathbf{u}_t,\boldsymbol{\mu}_{t-1}) + \mathbf{G}_t(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}),\ \mathbf{R}_t)$$

$$\approx \mathcal{N}_{\mathbf{z}_t}(h(\overline{\boldsymbol{\mu}}_t) + \mathbf{H}_t(\mathbf{x}_t - \overline{\boldsymbol{\mu}}_t),\ \mathbf{Q}_t)$$

1. Initialization: $\mathrm{bel}(\mathbf{x}_0)$,
2. Prediction step:
   $$\overline{\boldsymbol{\mu}}_t = g(\mathbf{u}_t, \boldsymbol{\mu}_{t-1})$$
   $$\overline{\boldsymbol{\Sigma}}_t = \mathbf{G}_t\boldsymbol{\Sigma}_{t-1}\mathbf{G}_t^\top + \mathbf{R}_t$$
   $$\overline{\mathrm{bel}}(\mathbf{x}_t) = \mathcal{N}_{\mathbf{x}_t}(\overline{\boldsymbol{\mu}}_t, \overline{\boldsymbol{\Sigma}}_t)$$
3. Measurement update:
   $$\mathbf{K}_t = \overline{\boldsymbol{\Sigma}}_t\mathbf{H}_t^\top(\mathbf{H}_t\overline{\boldsymbol{\Sigma}}_t\mathbf{H}_t^\top + \mathbf{Q}_t)^{-1}$$
   $$\boldsymbol{\mu}_t = \overline{\boldsymbol{\mu}}_t + \mathbf{K}_t(\mathbf{z}_t - h(\overline{\boldsymbol{\mu}}_t))$$
   $$\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\overline{\boldsymbol{\Sigma}}_t$$
   $$\mathrm{bel}(\mathbf{x}_t) = \mathcal{N}_{\mathbf{x}_t}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$$
4. Repeat from 2

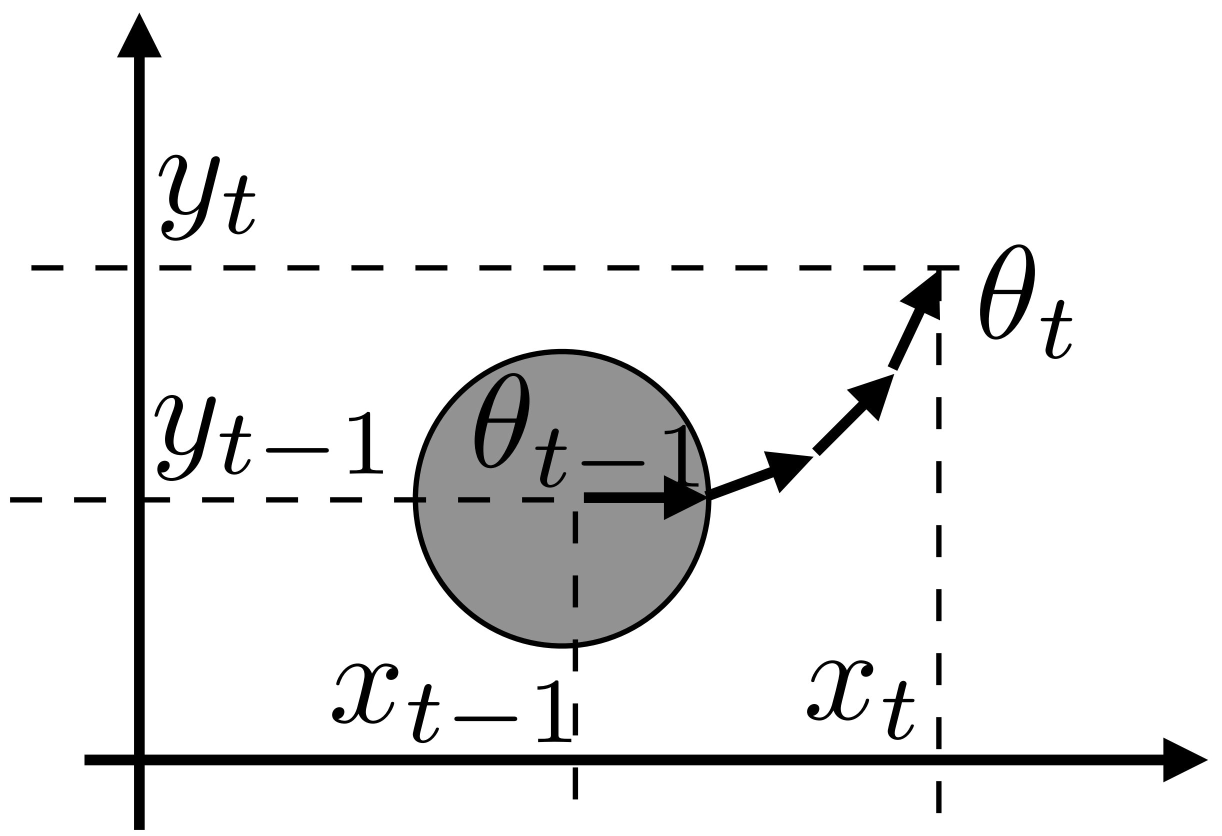
Example: Extended Kalman Filter, state=[pos_x, pos_y, heading] x_t

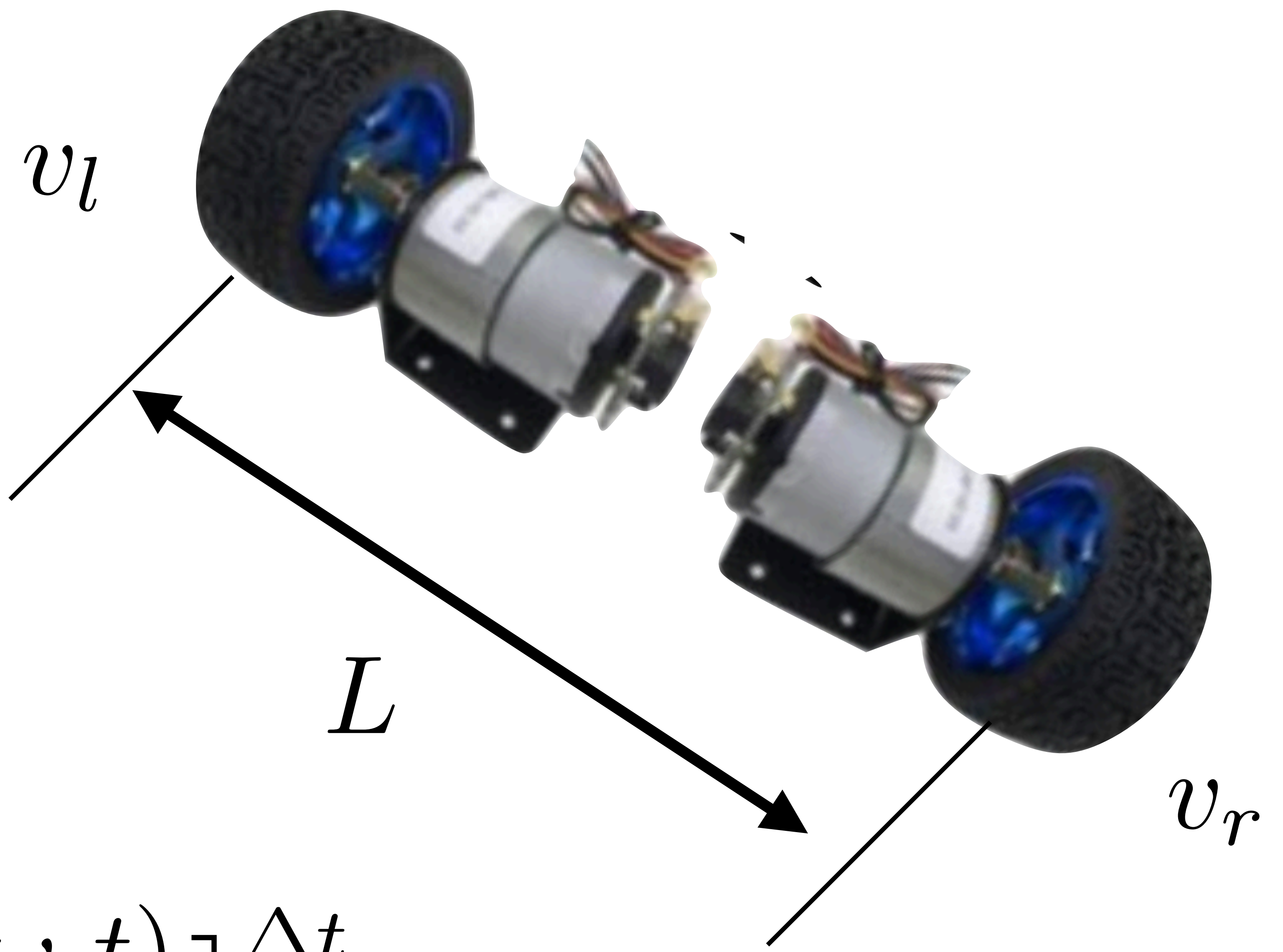
```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist
'{linear:  {x: 1.0, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 1.0}}'
```

$\mathbf{u}_t =$ [Linear velocity v , Angular velocity ω]

Control commands often replaced by wheel velocities

$$v = \frac{v_l + v_r}{2} \qquad \omega = \frac{v_l - v_r}{L}$$





$$\theta_t = \theta_{t-1} + \omega \Delta t$$

$$x_t = x_{t-1} + \int_0^{\Delta t} v_t \cos(\underbrace{\theta_{t-1} + \omega_t t}_{\theta(t)}) dt = x_{t-1} + v_t \left[\frac{\sin(\theta_{t-1} + \omega_t t)}{\omega_t} \right]_0^{\Delta t}$$

$$= x_{t-1} + \frac{v_t}{\omega_t}\big(+ \sin(\theta_{t-1} + \omega_t \Delta t) - \sin(\theta_{t-1})\big)$$

$$y_t = y_{t-1} + \frac{v_t}{\omega_t}\big(- \cos(\theta_{t-1} + \omega_t \Delta t) + \cos(\theta_{t-1})\big)$$

Example: Extended Kalman Filter

WHEEL ENCODERS

/odom

Turtlebot transition probability:

$$p\Bigg(\underbrace{\begin{bmatrix}x_t\\ y_t\\ \theta_t\end{bmatrix}}_{\mathbf{x}_t}\Bigg|\underbrace{\begin{bmatrix}x_{t-1}\\ y_{t-1}\\ \theta_{t-1}\end{bmatrix}}_{\mathbf{x}_{t-1}},\underbrace{\begin{bmatrix}v_t\\ \omega_t\end{bmatrix}}_{\mathbf{u}_t}\Bigg)=\mathcal{N}_{\mathbf{x}_t}\Bigg(\underbrace{\begin{bmatrix}x_{t-1}+\frac{v_t}{\omega_t}\big(+\sin(\theta_{t-1}+\omega_t\Delta t)-\sin(\theta_{t-1})\big)\\ y_{t-1}+\frac{v_t}{\omega_t}\big(-\cos(\theta_{t-1}+\omega_t\Delta t)+\cos(\theta_{t-1})\big)\\ \theta_{t-1}+\omega\Delta t\end{bmatrix}}_{g(\mathbf{u}_t,\mathbf{x}_{t-1})},\ \mathbf{R}_t\Bigg)$$

IMU

/imu_data

IMU measurement probability:

$$p\Bigg(\underbrace{\begin{bmatrix}\theta_t^{\text{IMU}}\end{bmatrix}}_{\mathbf{z}_t^{\text{IMU}}}\Bigg|\underbrace{\begin{bmatrix}x_t\\ y_t\\ \theta_t\end{bmatrix}}_{\mathbf{x}_t}\Bigg)=\mathcal{N}_{\mathbf{z}_t^{\text{IMU}}}\Bigg(\underbrace{\begin{bmatrix}0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_t\\ y_t\\ \theta_t\end{bmatrix}}_{h^{\text{IMU}}(\mathbf{x}_t)},\ Q_t^{\text{IMU}}\Bigg)$$

$$\mathbf{G}_t=\frac{\partial g(\mathbf{u}=\mathbf{u}_t,\mathbf{x}=\boldsymbol{\mu}_{t-1})}{\partial\mathbf{x}}=\begin{bmatrix}1 & 0 & \frac{v_t}{\omega_t}\big(+\cos(\mu_{t-1}^{\theta}+\omega_t\Delta t)-\cos(\mu_{t-1}^{\theta})\big)\\ 0 & 1 & \frac{v_t}{\omega_t}\big(+\sin(\mu_{t-1}^{\theta}+\omega_t\Delta t)-\sin(\mu_{t-1}^{\theta})\big)\\ 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{H}_t=\mathbf{C}_t=\begin{bmatrix}0 & 0 & 1\end{bmatrix}$$

ROS package for 6DOF EKF: http://wiki.ros.org/robot_pose_ekf

Example: Extended Kalman Filter

Turtlebot transition probability:

$$p\Big(\underbrace{\begin{bmatrix}x_t\\ y_t\\ \theta_t\end{bmatrix}}_{\mathbf{x}_t} \Big| \underbrace{\begin{bmatrix}x_{t-1}\\ y_{t-1}\\ \theta_{t-1}\end{bmatrix}}_{\mathbf{x}_{t-1}}, \underbrace{\begin{bmatrix}v_t\\ \omega_t\end{bmatrix}}_{\mathbf{u}_t}\Big) = \mathcal{N}_{\mathbf{x}_t}\Big(\underbrace{\begin{bmatrix}x_{t-1} + \frac{v_t}{\omega_t}\big(+ \sin(\theta_{t-1} + \omega_t \Delta t) - \sin(\theta_{t-1})\big)\\ y_{t-1} + \frac{v_t}{\omega_t}\big(- \cos(\theta_{t-1} + \omega_t \Delta t) + \cos(\theta_{t-1})\big)\\ \theta_{t-1} + \omega \Delta t\end{bmatrix}}_{g(\mathbf{u}_t, \mathbf{x}_{t-1})}, \ \mathbf{R}_t\Big)$$

WHEEL ENCODERS

/odom

IMU measurement probability:

$$p\Big(\underbrace{\begin{bmatrix}\theta_t^{\mathrm{IMU}}\end{bmatrix}}_{\mathbf{z}_t^{\mathrm{IMU}}} \Big| \underbrace{\begin{bmatrix}x_t\\ y_t\\ \theta_t\end{bmatrix}}_{\mathbf{x}_t}\Big) = \mathcal{N}_{\mathbf{z}_t^{\mathrm{IMU}}}\Big(\underbrace{\begin{bmatrix}0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_t\\ y_t\\ \theta_t\end{bmatrix}}_{h^{\mathrm{IMU}}(\mathbf{x}_t)}, \ Q_t^{\mathrm{IMU}}\Big)$$

IMU

/imu_data

$$\mathbf{G}_t = \frac{\partial g(\mathbf{u} = \mathbf{u}_t, \mathbf{x} = \boldsymbol{\mu}_{t-1})}{\partial \mathbf{x}} = \begin{bmatrix}1 & 0 & \frac{v_t}{\omega_t}\big(+ \cos(\mu_{t-1}^{\theta} + \omega_t \Delta t) - \cos(\mu_{t-1}^{\theta})\big)\\ 0 & 1 & \frac{v_t}{\omega_t}\big(+ \sin(\mu_{t-1}^{\theta} + \omega_t \Delta t) - \sin(\mu_{t-1}^{\theta})\big)\\ 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{H}_t = \mathbf{C}_t = \begin{bmatrix}0 & 0 & 1\end{bmatrix}$$

ROS package for 6DOF EKF: http://wiki.ros.org/robot_pose_ekf

Extended Kalman Filter

Non-linear system with Gaussian noise:

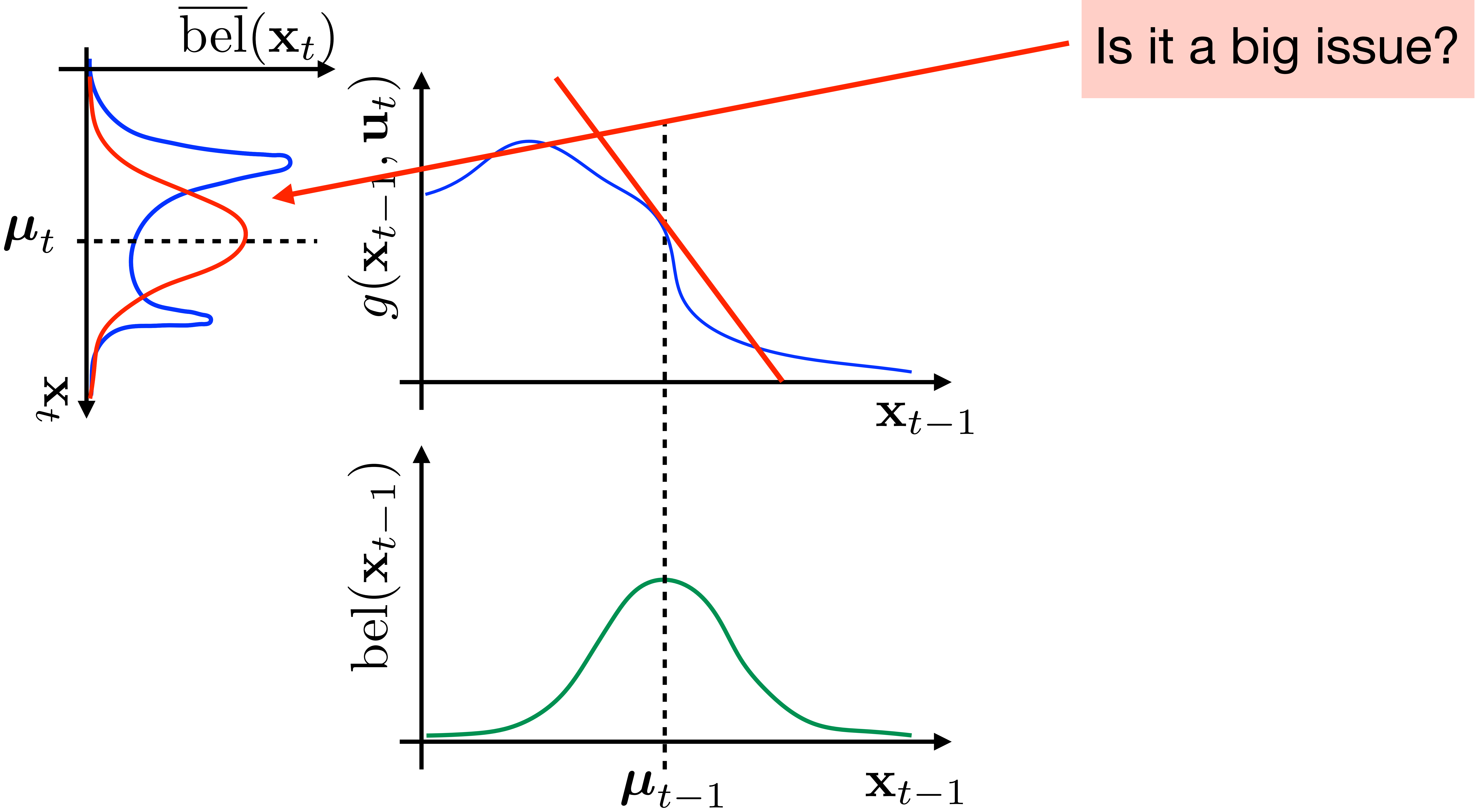
$$p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t) = \mathcal{N}_{\mathbf{x}_t}(g(\mathbf{x}_{t-1},\mathbf{u}_t),\ \mathbf{R}_t)$$

$$p(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}_{\mathbf{z}_t}(h(\mathbf{x}_t),\ \mathbf{Q}_t)$$

Linearized system with Gaussian noise:

$$\approx \mathcal{N}_{\mathbf{x}_t}(g(\mathbf{u}_t,\boldsymbol{\mu}_{t-1}) + \mathbf{G}_t(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}),\ \mathbf{R}_t)$$

$$\approx \mathcal{N}_{\mathbf{z}_t}(h(\overline{\boldsymbol{\mu}}_t) + \mathbf{H}_t(\mathbf{x}_t - \overline{\boldsymbol{\mu}}_t),\ \mathbf{Q}_t)$$



Extended Kalman Filter

Non-linear system with Gaussian noise:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t) = \mathcal{N}_{\mathbf{x}_t}(g(\mathbf{x}_{t-1},\mathbf{u}_t),\ \mathbf{R}_t)$$

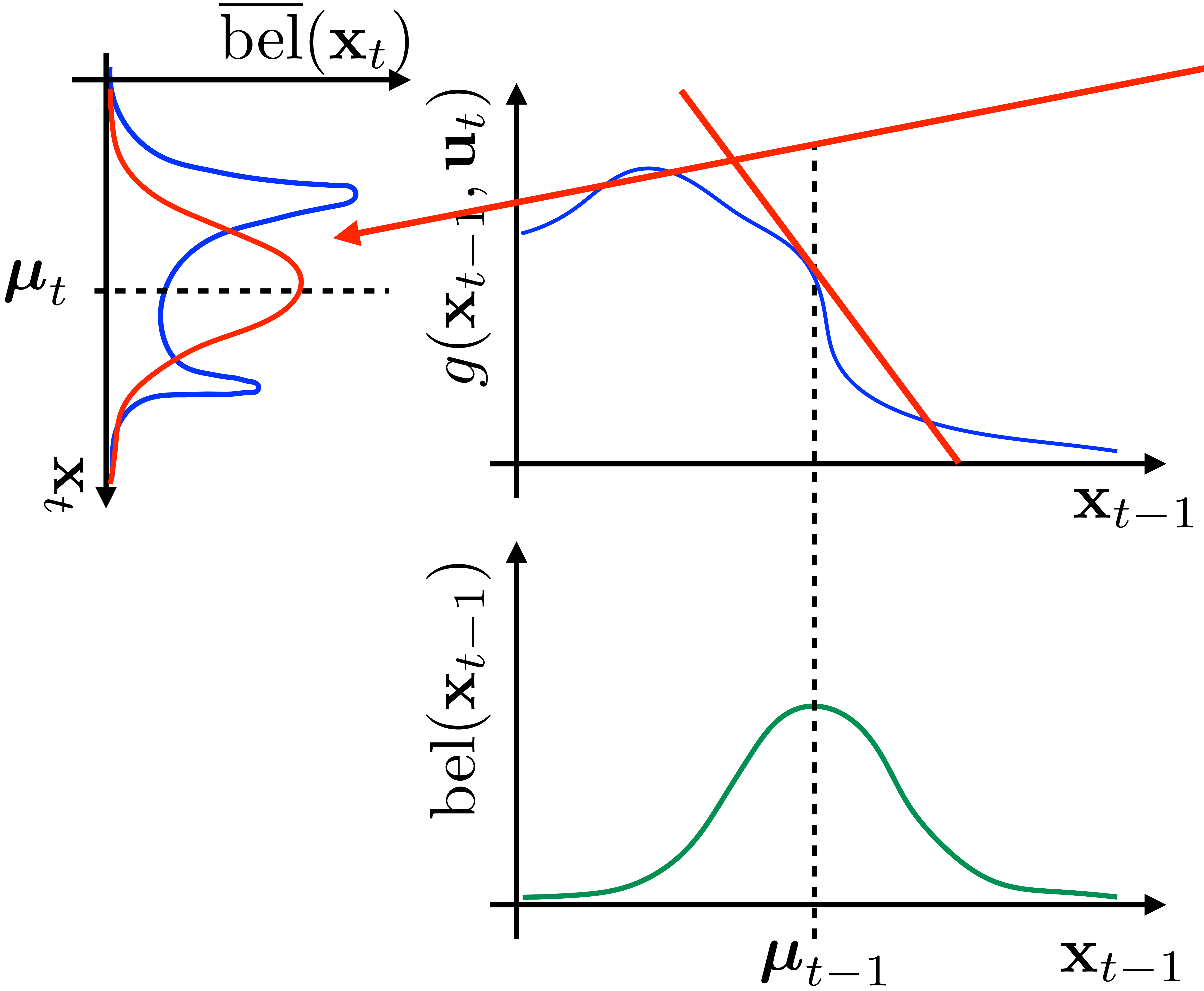
$$p(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}_{\mathbf{z}_t}(h(\mathbf{x}_t),\ \mathbf{Q}_t)$$

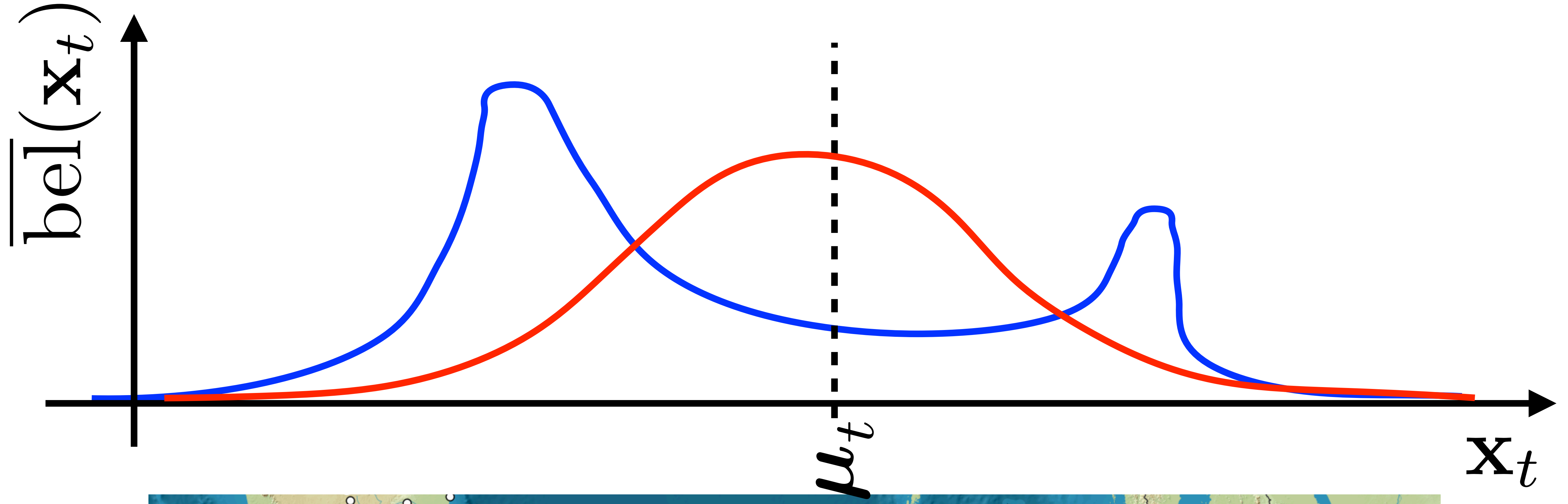
Linearized system with Gaussian noise:

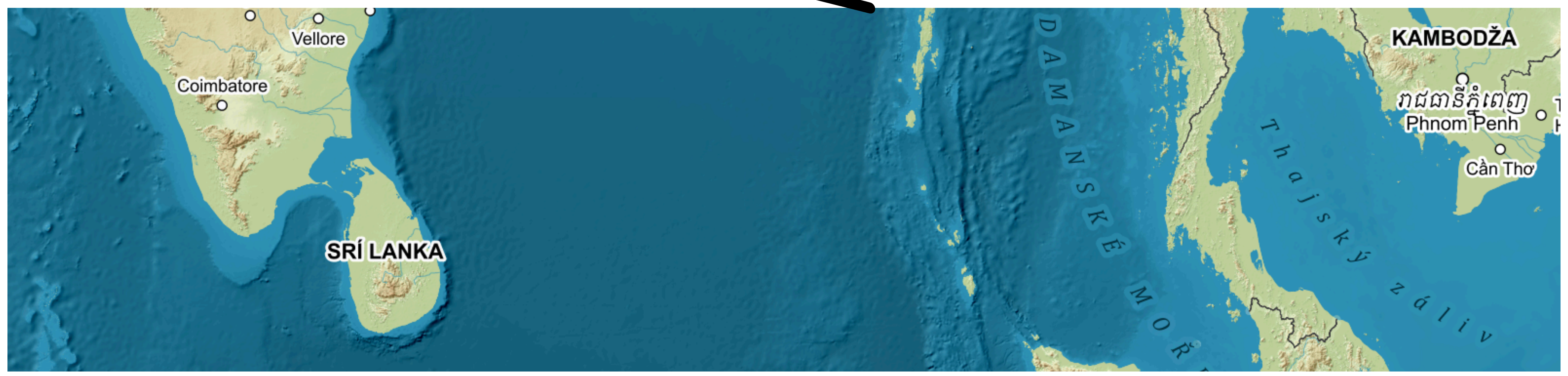
$$\approx \mathcal{N}_{\mathbf{x}_t}(g(\mathbf{u}_t,\boldsymbol{\mu}_{t-1}) + \mathbf{G}_t(\mathbf{x}_{t-1}-\boldsymbol{\mu}_{t-1}),\ \mathbf{R}_t)$$

$$\approx \mathcal{N}_{\mathbf{z}_t}(h(\overline{\boldsymbol{\mu}}_t) + \mathbf{H}_t(\mathbf{x}_t-\overline{\boldsymbol{\mu}}_t),\ \mathbf{Q}_t)$$

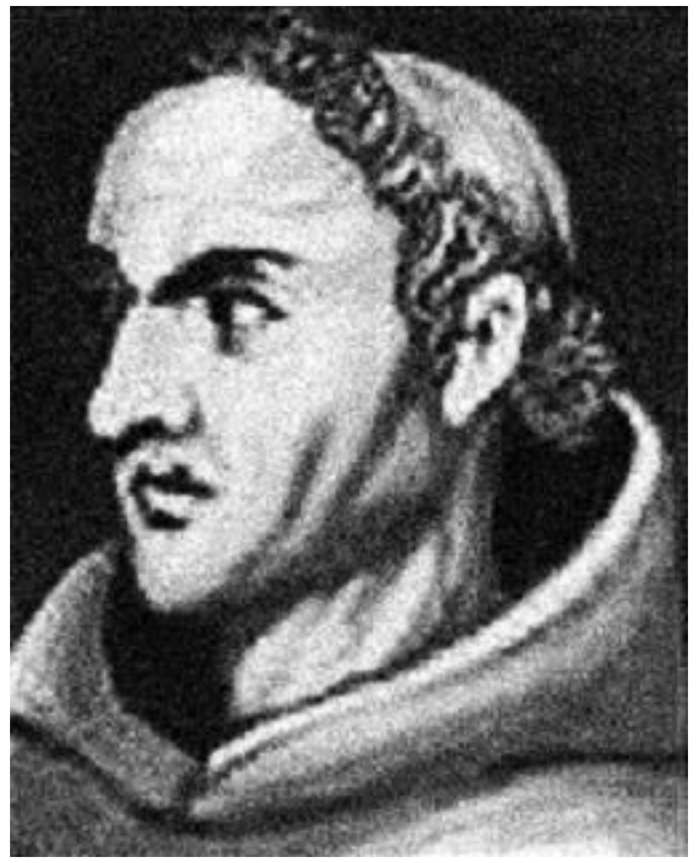
Is it a big issue?







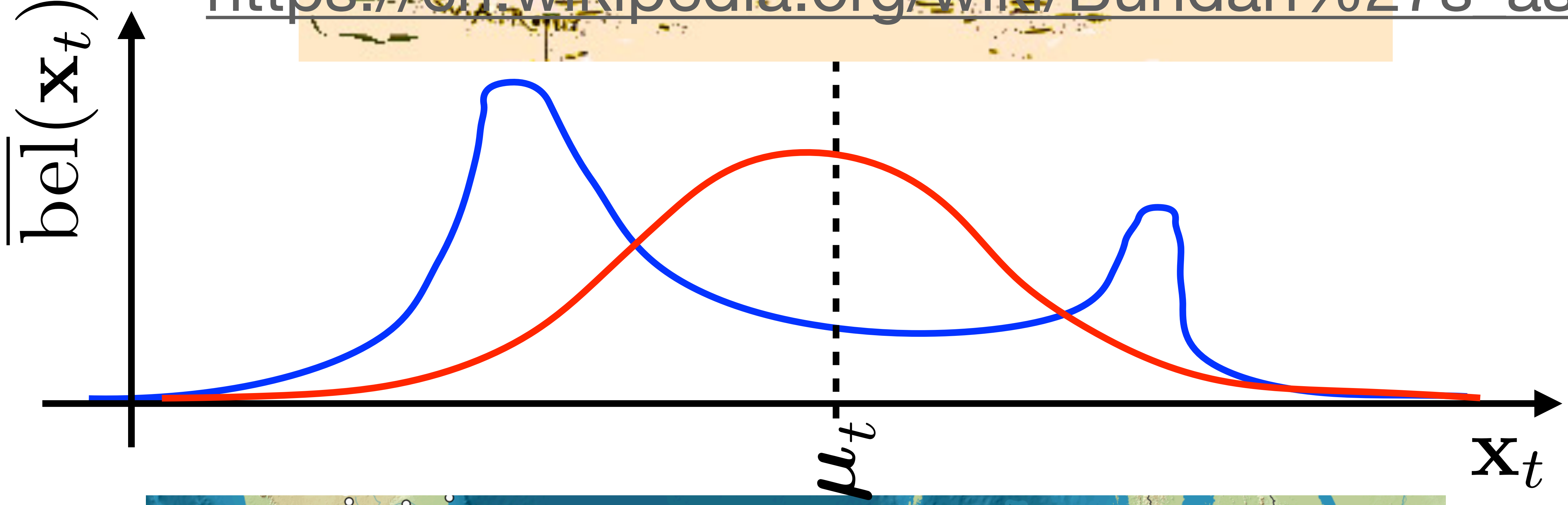
Extended Kalman Filter

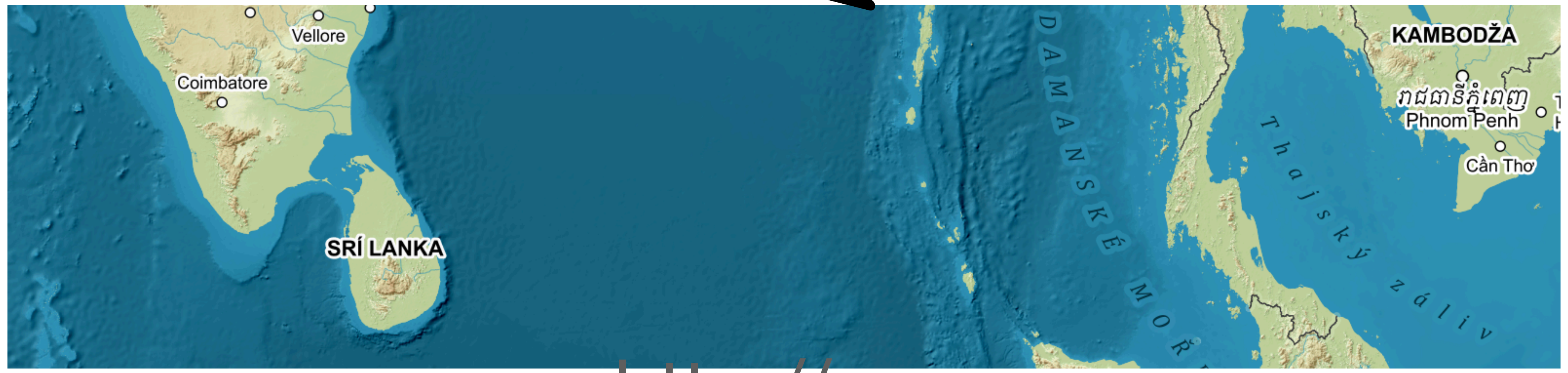


Jean Buridan

https://en.wikipedia.org/wiki/Buridan%27s_ass

$\mathbf{x}_{t-1}$





http://mapy.cz

Example: Extended Kalman Filter

Turtlebot transition probability:

$$p\Bigg(\underbrace{\begin{bmatrix}x_t\\ y_t\\ \theta_t\end{bmatrix}}_{\mathbf{x}_t}\,\Bigg|\,\underbrace{\begin{bmatrix}x_{t-1}\\ y_{t-1}\\ \theta_{t-1}\end{bmatrix}}_{\mathbf{x}_{t-1}},\underbrace{\begin{bmatrix}v_t\\ \omega_t\end{bmatrix}}_{\mathbf{u}_t}\Bigg) = \mathcal{N}_{\mathbf{x}_t}\Bigg(\underbrace{\begin{bmatrix}x_{t-1}+\frac{v_t}{\omega_t}\big(+\sin(\theta_{t-1}+\omega_t\Delta t)-\sin(\theta_{t-1})\big)\\ y_{t-1}+\frac{v_t}{\omega_t}\big(-\cos(\theta_{t-1}+\omega_t\Delta t)+\cos(\theta_{t-1})\big)\\ \theta_{t-1}+\omega\Delta t\end{bmatrix}}_{g(\mathbf{u}_t,\mathbf{x}_{t-1})},\ \mathbf{R}_t\Bigg)$$

WHEEL ENCODERS

/odom

IMU measurement probability:

$$p\Bigg(\underbrace{[\theta_t^{\mathrm{IMU}}]}_{\mathbf{z}_t^{\mathrm{IMU}}}\,\Bigg|\,\underbrace{\begin{bmatrix}x_t\\ y_t\\ \theta_t\end{bmatrix}}_{\mathbf{x}_t}\Bigg) = \mathcal{N}_{\mathbf{z}_t^{\mathrm{IMU}}}\Bigg(\underbrace{\begin{bmatrix}0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_t\\ y_t\\ \theta_t\end{bmatrix}}_{h^{\mathrm{IMU}}(\mathbf{x}_t)},\ Q_t^{\mathrm{IMU}}\Bigg)$$

IMU

/imu_data

LIDAR measurement probability

???

Example: Extended Kalman Filter

Turtlebot transition probability:

$$p\Big(\underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t} \Big| \underbrace{\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}}_{\mathbf{x}_{t-1}}, \underbrace{\begin{bmatrix} v_t \\ \omega_t \end{bmatrix}}_{\mathbf{u}_t}\Big) = \mathcal{N}_{\mathbf{x}_t}\Big(\underbrace{\begin{bmatrix} x_{t-1} + \frac{v_t}{\omega_t}\big(+ \sin(\theta_{t-1} + \omega_t \Delta t) - \sin(\theta_{t-1})\big) \\ y_{t-1} + \frac{v_t}{\omega_t}\big(- \cos(\theta_{t-1} + \omega_t \Delta t) + \cos(\theta_{t-1})\big) \\ \theta_{t-1} + \omega \Delta t \end{bmatrix}}_{g(\mathbf{u}_t, \mathbf{x}_{t-1})}, \;\; \mathbf{R}_t \Big)$$

WHEEL ENCODERS

/odom

IMU measurement probability:

$$p\Big(\underbrace{\begin{bmatrix} \theta_t^{\mathrm{IMU}} \end{bmatrix}}_{\mathbf{z}_t^{\mathrm{IMU}}} \Big| \underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t}\Big) = \mathcal{N}_{\mathbf{z}_t^{\mathrm{IMU}}}\Big(\underbrace{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{h^{\mathrm{IMU}}(\mathbf{x}_t)}, \;\; Q_t^{\mathrm{IMU}} \Big)$$

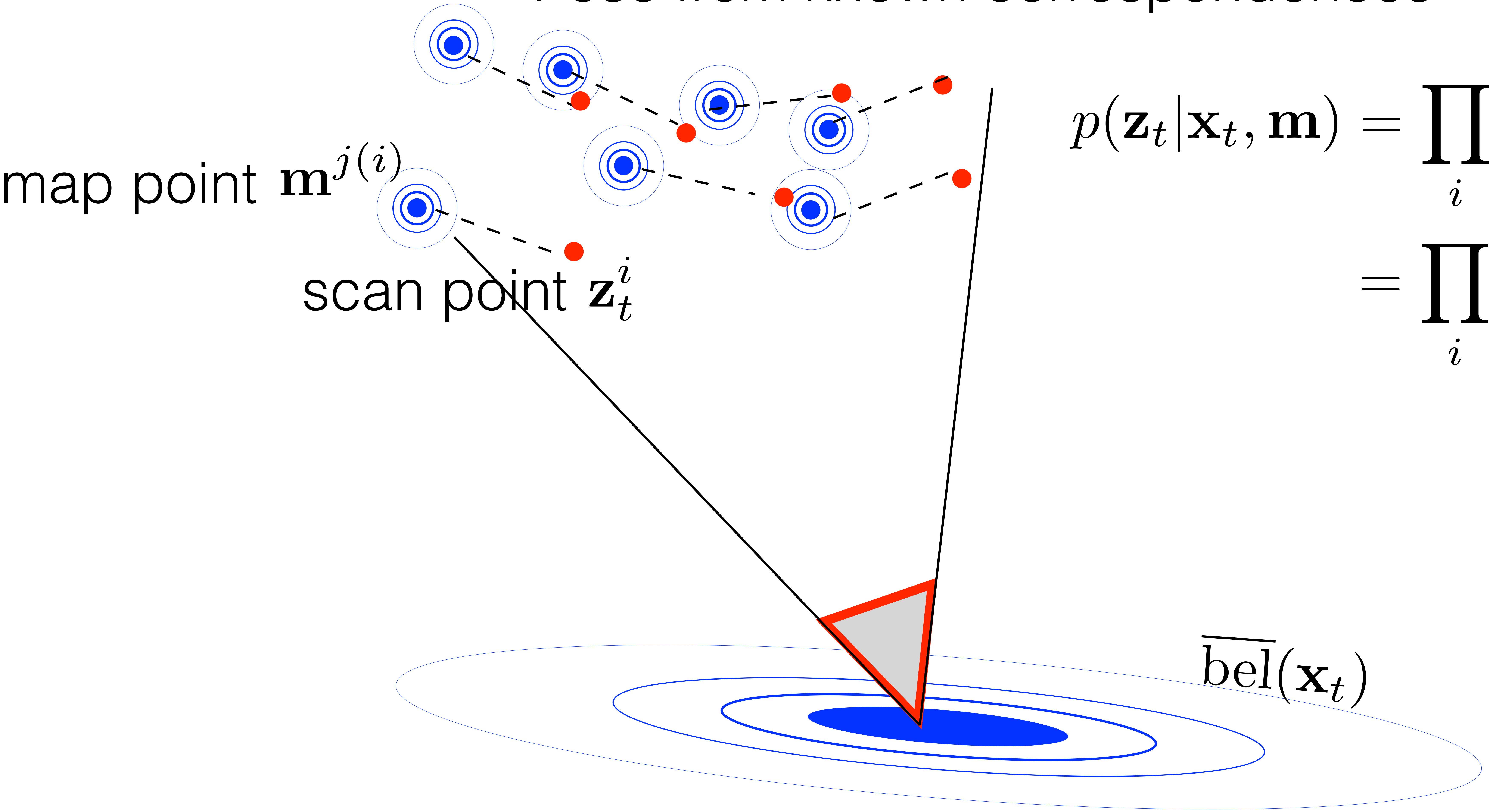
IMU

/imu_data

LIDAR measurement probability:

$$p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m})$$

Pose from known correspondences



$$p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t,\mathbf{m}^{j(i)})$$

$$= \prod_i \mathcal{N}_{T(\mathbf{z}_t^i,\mathbf{x}_t)}(\mathbf{m}^{j(i)},\mathbf{Q}_t^i)$$

Example: Extended Kalman Filter

Turtlebot transition probability:

$$p\Big(\underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t} \Big| \underbrace{\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}}_{\mathbf{x}_{t-1}}, \underbrace{\begin{bmatrix} v_t \\ \omega_t \end{bmatrix}}_{\mathbf{u}_t}\Big) = \mathcal{N}_{\mathbf{x}_t}\Big(\underbrace{\begin{bmatrix} x_{t-1} + \frac{v_t}{\omega_t}\big(+\sin(\theta_{t-1} + \omega_t \Delta t) - \sin(\theta_{t-1})\big) \\ y_{t-1} + \frac{v_t}{\omega_t}\big(-\cos(\theta_{t-1} + \omega_t \Delta t) + \cos(\theta_{t-1})\big) \\ \theta_{t-1} + \omega \Delta t \end{bmatrix}}_{g(\mathbf{u}_t, \mathbf{x}_{t-1})}, \ \mathbf{R}_t \Big)$$

IMU measurement probability:

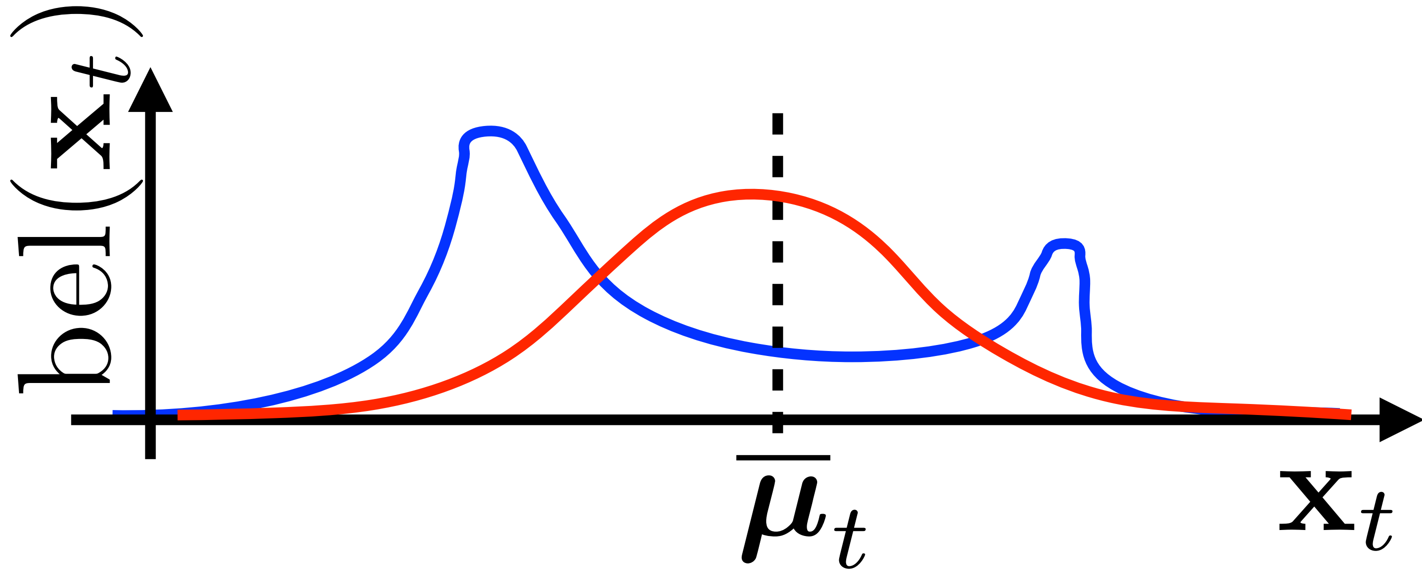
$$p\Big(\underbrace{[\theta_t^{\mathrm{IMU}}]}_{\mathbf{z}_t^{\mathrm{IMU}}} \Big| \underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t}\Big) = \mathcal{N}_{\mathbf{z}_t^{\mathrm{IMU}}}\Big(\underbrace{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{h^{\mathrm{IMU}}(\mathbf{x}_t)}, \ Q_t^{\mathrm{IMU}} \Big)$$

LIDAR measurement probability:

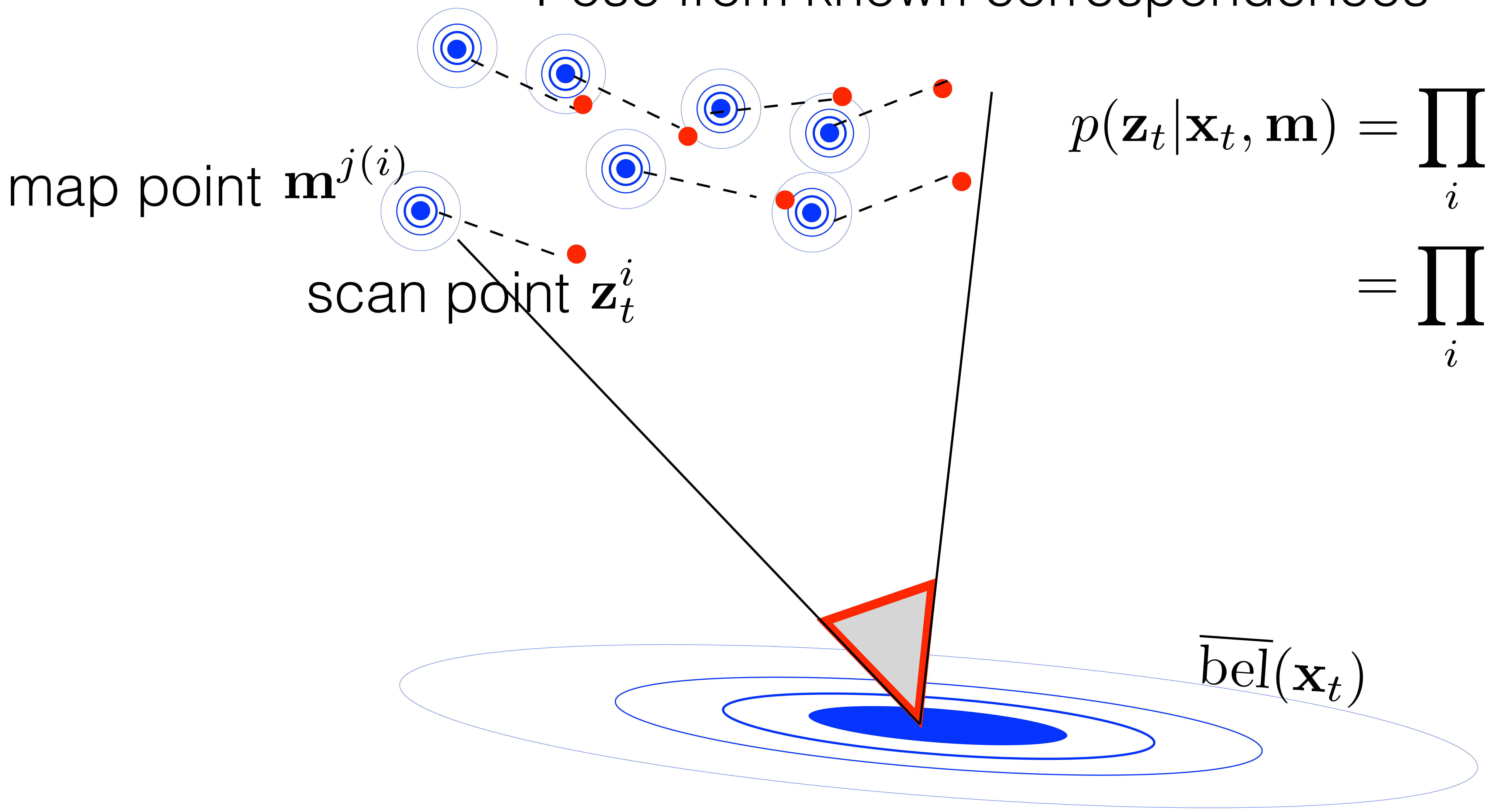
$$p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) = \prod_i \mathcal{N}_{T(\mathbf{z}_t^i, \mathbf{x}_t)}(\mathbf{m}^{j(i)}, \mathbf{Q}_t^i)$$

$$\approx \prod_i \mathcal{N}_{\mathbf{z}_t}\big(h(\overline{\boldsymbol{\mu}}_t, \mathbf{m}^{j(i)}) + \mathbf{H}_t^i(\mathbf{x}_t - \overline{\boldsymbol{\mu}}_t), \ \mathbf{Q}_t^i\big)$$

EKF localization / SLAM



Pose from known correspondences



$$p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t,\mathbf{m}^{j(i)})$$

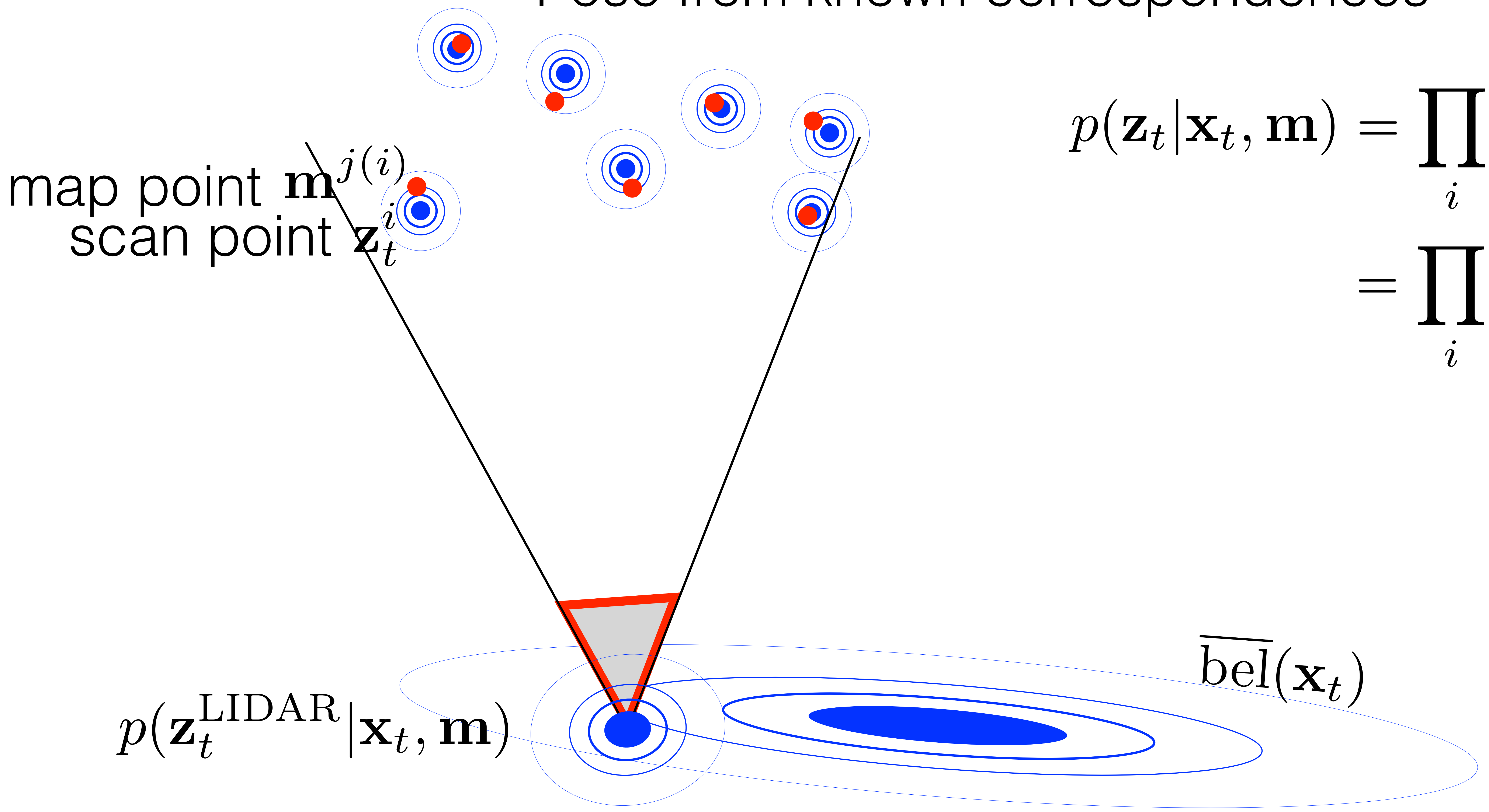
$$= \prod_i \mathcal{N}_{T(\mathbf{z}_t^i,\mathbf{x}_t)}(\mathbf{m}^{j(i)},\mathbf{Q}_t^i)$$

$$\mathbf{z}_t^{\text{LIDAR}} = \arg\max_{\mathbf{x}_t} \; p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})$$

Pose from known correspondences

$$p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \prod_i p(\mathbf{z}_t^i|\mathbf{x}_t,\mathbf{m}^{j(i)})$$

$$= \prod_i \mathcal{N}_{T(\mathbf{z}_t^i,\mathbf{x}_t)}(\mathbf{m}^{j(i)},\mathbf{Q}_t^i)$$



$$\mathbf{z}_t^{\text{LIDAR}} = \arg\max_{\mathbf{x}_t} \; p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})$$

Example: Extended Kalman Filter

Turtlebot transition probability:

$$p\Big(\underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t} \,\Big|\, \underbrace{\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}}_{\mathbf{x}_{t-1}}, \underbrace{\begin{bmatrix} v_t \\ \omega_t \end{bmatrix}}_{\mathbf{u}_t}\Big) = \mathcal{N}_{\mathbf{x}_t}\Big(\underbrace{\begin{bmatrix} x_{t-1} + \frac{v_t}{\omega_t}\big(+\sin(\theta_{t-1}+\omega_t\Delta t) - \sin(\theta_{t-1})\big) \\ y_{t-1} + \frac{v_t}{\omega_t}\big(-\cos(\theta_{t-1}+\omega_t\Delta t) + \cos(\theta_{t-1})\big) \\ \theta_{t-1} + \omega\Delta t \end{bmatrix}}_{g(\mathbf{u}_t,\mathbf{x}_{t-1})}, \;\; \mathbf{R}_t\Big)$$

WHEEL ENCODERS

/odom

IMU measurement probability:

$$p\Big(\underbrace{[\theta_t^{\text{IMU}}]}_{\mathbf{z}_t^{\text{IMU}}} \,\Big|\, \underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t}\Big) = \mathcal{N}_{\mathbf{z}_t^{\text{IMU}}}\Big(\underbrace{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{h^{\text{IMU}}(\mathbf{x}_t)}, \;\; Q_t^{\text{IMU}}\Big)$$

IMU

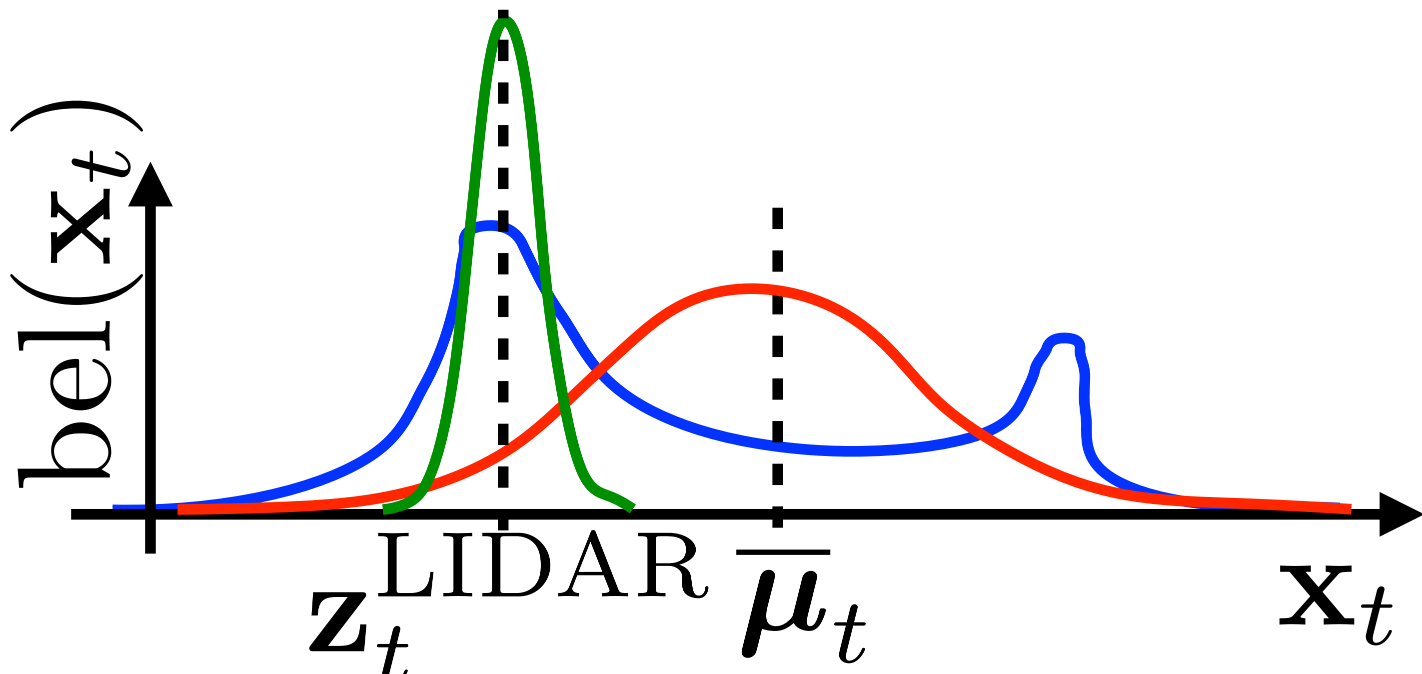
/imu_data

LIDAR measurement probability:

$$\mathbf{z}_t^{\text{LIDAR}} = \arg\max_{\mathbf{x}_t} \; p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m})$$

$$\underbrace{\begin{bmatrix} x_t^{\text{LIDAR}} \\ y_t^{\text{LIDAR}} \\ \theta_t^{\text{LIDAR}} \end{bmatrix}}_{\mathbf{z}_t^{\text{LIDAR}}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \mathcal{N}_{\mathbf{z}_t}(\mathbf{0}, \mathbf{Q}_t^{\text{LIDAR}})$$

generalized ICP
localization / SLAM



Summary Extended Kalman Filter (EKF)

- EKF is suboptimal observer of the state for non-linear systems under Gaussian noise

- EKF is KF with transition and measurement probabilities iteratively approximated by the first order Taylor expansion.

- It nicely scales to higher dimension and tackles the non-linearity well for smooth functions.

- It has been used for onboard guidance and navigation system for the Apollo Spacecraft Mission https://en.wikipedia.org/wiki/Apollo_(spacecraft)

- There are other ways of non-linearity approximation such as Assumed Density Filter (ADF) or Unscented Kalman Filter (UKF).

- Next: Lidar and corresponding measurements probability models