Probabilistic Robotics Course

EKF SLAM [Application]

Giorgio Grisetti

grisetti@diag.uniroma1.it

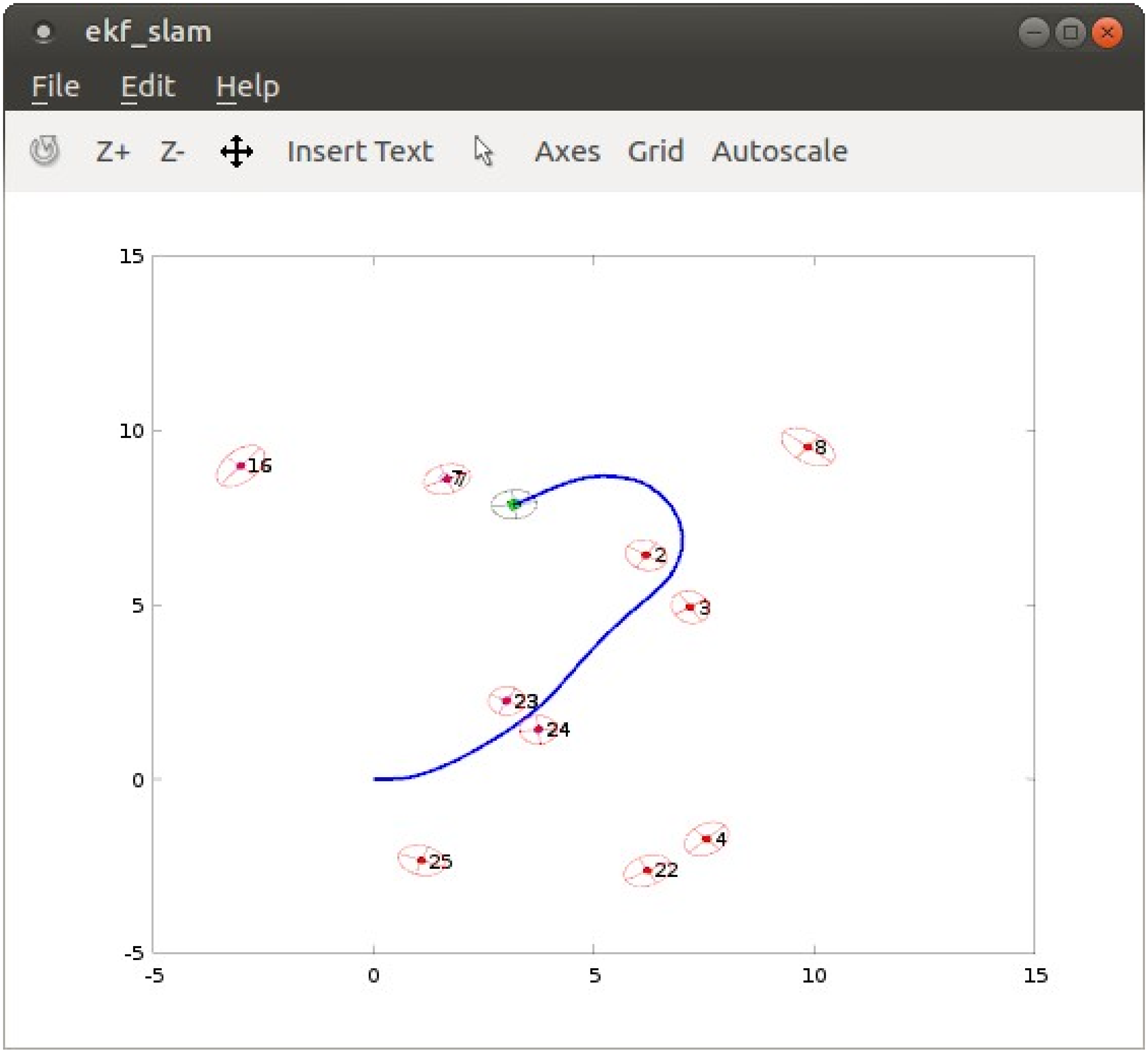
Department of Computer, Control and Management Engineering
Sapienza University of Rome

EKF SLAM: recap

- Estimate the current state distribution from
  - Previous state distribution
  - Sequence of observations $z_{0:t}$
  - Sequence of controls $u_{0:t-1}$
  - Transition model
  - Observation model

$$\begin{aligned}
\mu_{t|t-1} &= \boxed{\mathbf{f}(\boxed{\mu_{t-1|t-1}}, \mathbf{u}_{t-1})} \\
\mathbf{\Sigma}_{t|t-1} &= \mathbf{A}_t \boxed{\mathbf{\Sigma}_{t-1|t-1}} \mathbf{A}_t^T + \mathbf{B}_t \mathbf{\Sigma}_u \mathbf{B}_t^T
\end{aligned}$$

$$\begin{aligned}
\mu_z &= \boxed{\mathbf{h}(\mu_{t|t-1})} \\
\boxed{\mu_{t|t}} &= \mu_{t|t-1} + \mathbf{K}_t \left(\mathbf{z}_t - \mu_z\right) \\
\boxed{\mathbf{\Sigma}_{t|t}} &= \left(\mathbf{I} - \mathbf{K}_t \mathbf{C}_t\right) \mathbf{\Sigma}_{t|t-1}
\end{aligned}$$


ekf_slam
File
Edit
Help
Z+
Z-
Insert Text
Axes
Grid
Autoscale
15
10
5
0
-5
-5
0
5
10
15
16
7
8
2
3
23
24
4
25
22

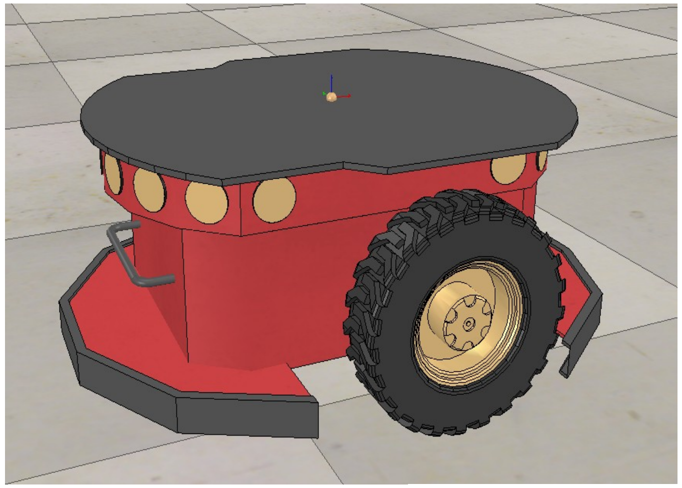
Outline

- Scenario
- Controls
- Observations
- Non-Linear Systems and Gaussian Noise
- Extended Kalman Filter SLAM

Scenario

Orazio moves on a 2D plane

- Is controlled by translational and rotational velocities
- Senses a set of uniquely distinguishable landmarks through a "2D landmark sensor"
- The location of the landmarks in the world is **not** known



Approaching the problem

We want to develop a KF based algorithm to track the position of Orazio as it moves (localization) and, at the same time, the position of the observed landmarks (mapping).

The inputs of our algorithms will be

- velocity measurements
- landmark measurements

We have no prior knowledge of the map!

EKF

1. Predict: incorporate new control

$$\begin{aligned}
\mu_{t|t-1} &= \mathbf{f}(\mu_{t-1|t-1}, \mu_{u,t-1}) \\
\mathbf{A}_t &= \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mu_{t-1|t-1}} \\
\mathbf{B}_t &= \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mu_{u,t-1}} \\
\mathbf{\Sigma}_{t|t-1} &= \mathbf{A}_t \mathbf{\Sigma}_{t-1|t-1} \mathbf{A}_t^T + \mathbf{B}_t \mathbf{\Sigma}_u \mathbf{B}_t^T
\end{aligned}$$

2. Correct: incorporate new measurement

$$\begin{aligned}
\mathbf{C}_t &= \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mu_{t|t-1}} \\
\mathbf{K}_t &= \mathbf{\Sigma}_{t|t-1} \mathbf{C}_t^T \left(\mathbf{\Sigma}_z + \mathbf{C}_t \mathbf{\Sigma}_{t|t-1} \mathbf{C}_t^T\right)^{-1} \\
\mu_{t|t} &= \mu_{t|t-1} + \mathbf{K}_t \left(\boxed{\mathbf{z}_t - \mathbf{h}(\mu_{t|t-1})}\right) \\
\mathbf{\Sigma}_{t|t} &= (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t)\, \mathbf{\Sigma}_{t|t-1}
\end{aligned}$$

innovation

3. SLAM: add new landmarks to state

Domains: State Space

Robot:

$$\mathbf{X}_t^{[r]} = [\mathbf{R}_t | \mathbf{t}_t] \in SE(2) \Longrightarrow \mathbf{x}_t^{[r]} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} \in \Re^3$$

Landmarks:

$$\mathbf{x}_t^{[n]} = \begin{pmatrix} x_t^{[n]} \\ y_t^{[n]} \end{pmatrix} \in \Re^2 \qquad {}_{n=1,2,\cdots,N}$$

Full state vector:

$$\mathbf{x}_t = \begin{pmatrix} \mathbf{x}_t^{[r]} \\ \mathbf{x}_t^{[1]} \\ \mathbf{x}_t^{[2]} \\ \vdots \\ \mathbf{x}_t^{[N]} \end{pmatrix} \in \Re^{(3+2N)}$$

Domains: Ctrl and Meas.

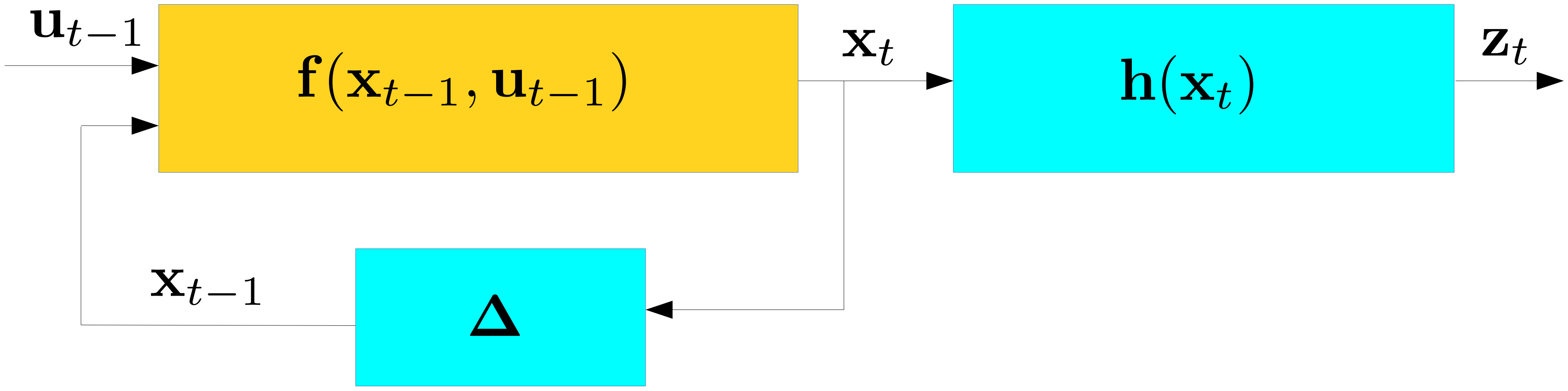
Controls: (same as in EKF Localization)

$$\mathbf{u}_t = \begin{pmatrix} u_t^{[1]} \\ u_t^{[2]} \end{pmatrix} \in \Re^2$$

Measurements:

$$\mathbf{z}_t^{[m]} = \begin{pmatrix} x_t^{[m]} \\ y_t^{[m]} \end{pmatrix} \in \Re^2 \quad {}_{m=1..M}$$

Transition Function

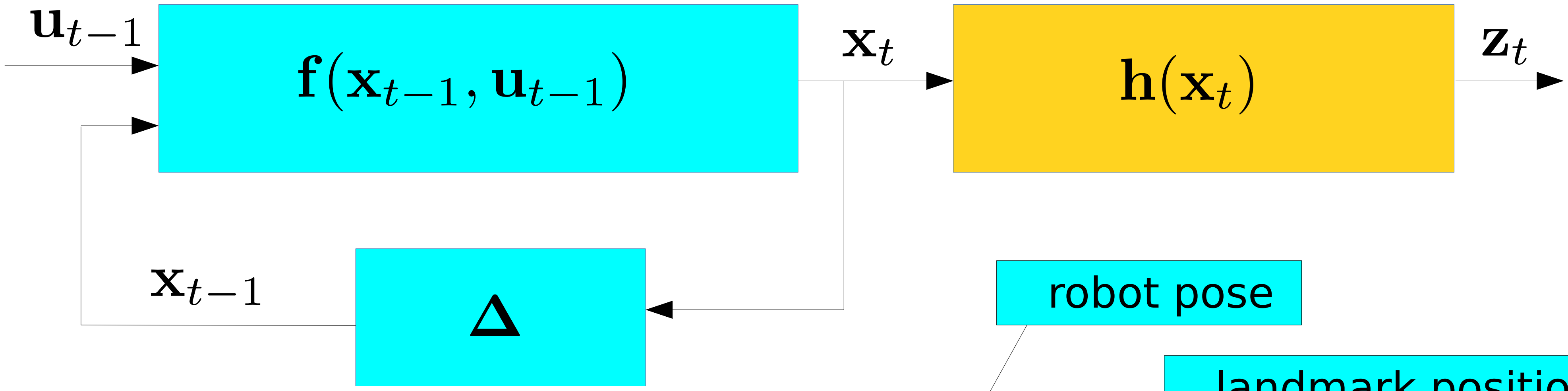


pose update (the robot moves)

the landmarks don't move

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) = \begin{pmatrix} x_{t-1} + u_{t-1}^{[1]} \cdot \cos(\theta_{t-1}) \\ y_{t-1} + u_{t-1}^{[1]} \cdot \sin(\theta_{t-1}) \\ \theta_{t-1} + u_{t-1}^{[2]} \\ \mathbf{x}_{t-1}^{[1]} \\ \mathbf{x}_{t-1}^{[2]} \\ \dots \\ \mathbf{x}_{t-1}^{[N]} \end{pmatrix}$$

Measurement Function



robot pose

landmark position

$$\begin{aligned}
\mathbf{z}_t^{[n]} &= \boxed{\mathbf{h}^{[n]}(\mathbf{x}_t)} = \mathbf{h}(\mathbf{x}_t^{[r]}, \mathbf{x}_t^{[n]}) \\
&= \mathbf{R}_t^T(\mathbf{x}_t^{[n]} - \mathbf{t}_t) \\
&= \begin{pmatrix} \cos\theta_t(x_t^{[n]} - x_t) + \sin\theta_t(y_t^{[n]} - y_t) \\ -\sin\theta_t(x_t^{[n]} - x_t) + \cos\theta_t(y_t^{[n]} - y_t) \end{pmatrix}
\end{aligned}$$

$n = 1, .., N$

We have N potential measurement functions, one for each landmark

relative position of the nth landmark w.r.t. the robot at time t

Control Noise

We assume the control inputs are effected by a zero-mean Gaussian noise resulting from the sum of two aspects:

- a constant noise σ_u
- velocity dependent terms whose standard deviation grows with the speed (i.d.)
  - translational noise standard deviation: $\sigma_T = u_t^{[1]}$
  - rotational noise standard deviation: $\sigma_R = u_t^{[2]}$

$$\mathbf{n}_{u,t} \sim \mathcal{N}(\mathbf{n}_{u,t}; \mathbf{0}, \underbrace{\begin{pmatrix} \sigma_u^2 + \sigma_T^2 & 0 \\ 0 & \sigma_u^2 + \sigma_R^2 \end{pmatrix}}_{\Sigma_u})$$

Measurement Noise

For each landmark we measure, we consider a Gaussian noise (in x and y):

$$\mathbf{n}_z \sim \mathcal{N}(\mathbf{n}_z; \mathbf{0}, \underbrace{\begin{pmatrix} \sigma_z^2 & 0 \\ 0 & \sigma_z^2 \end{pmatrix}}_{\Sigma_z})$$

We assume it is zero mean and constant with standard deviation σ_z .

Jacobian 1: State

$$\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) = \begin{pmatrix} x_{t-1} + u_{t-1}^{[1]} \cdot \cos(\theta_{t-1}) \\ y_{t-1} + u_{t-1}^{[1]} \cdot \sin(\theta_{t-1}) \\ \theta_{t-1} + u_{t-1}^{[2]} \\ \mathbf{x}_{t-1}^{[1]} \\ \mathbf{x}_{t-1}^{[2]} \\ \dots \\ \mathbf{x}_{t-1}^{[N]} \end{pmatrix}$$

$$\mathbf{A}_t = \left.\frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mu_{t-1|t-1}} = \begin{pmatrix} \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}^{[r]}} & \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}^{[1]}} & \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}^{[2]}} & \dots & \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}^{[N]}} \end{pmatrix}$$

$$\frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}^{[r]}} = \begin{pmatrix} 1 & 0 & -u_{t-1}^{[1]} \cdot \sin(\theta_{t-1}) \\ 0 & 1 & +u_{t-1}^{[1]} \cdot \cos(\theta_{t-1}) \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{pmatrix}$$

$$\frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}^{[2]}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix}$$

Jacobian 2: Controls

$$\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) = \begin{pmatrix} x_{t-1} + u_{t-1}^{[1]} \cdot \cos(\theta_{t-1}) \\ y_{t-1} + u_{t-1}^{[1]} \cdot \sin(\theta_{t-1}) \\ \theta_{t-1} + u_{t-1}^{[2]} \\ \mathbf{x}_{t-1}^{[1]} \\ \mathbf{x}_{t-1}^{[2]} \\ \cdots \\ \mathbf{x}_{t-1}^{[N]} \end{pmatrix}$$

$$\mathbf{B}_t = \left.\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}_{t-1}} = \begin{array}{c} \left(\begin{array}{cc} \cos(\theta_{t-1}) & 0 \\ \sin(\theta_{t-1}) & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{array}\right) \end{array} \begin{array}{l} \\ \leftarrow \text{pose block} \\ \\ \leftarrow \text{landmark block 1} \\ \\ \\ \leftarrow \text{landmark block N} \end{array}$$

Jacobian 3: Measurements

For the measurement function, we need to derive w.r.t. **all** state variables (N+3)

Prediction of the nth landmark observation:

nth landmark location

$$\mathbf{h}^{[n]}(\mathbf{x}_t) = \mathbf{R}_t^T(\mathbf{x}_t^{[n]} - \mathbf{t}_t)$$

pose

The nth measurement Jacobian will have a block structure

$$\mathbf{C}_t^{[n]} = \left.\frac{\partial \mathbf{h}^{[n]}(\cdot)}{\partial \mathbf{x}}\right|_{x=\mu_{t|t-1}} = \begin{pmatrix} \frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}^{[r]}} & \mathbf{0} & \cdots & \frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}^{[n]}} & \cdots & \mathbf{0} \end{pmatrix}$$

pose block

landmark block

Jacobian 3: Measurements

We need to calculate the derivatives only for the non zero blocks

$$\mathbf{h}^{[n]}(\mathbf{x}_t) = \mathbf{R}_t^T(\mathbf{x}_t^{[n]} - \mathbf{t}_t)$$

$$\frac{\partial \mathbf{h}^{[n]}(\cdot)}{\partial \mathbf{x}^{[r]}} = \begin{pmatrix} -\mathbf{R}_t^T & \frac{\partial \mathbf{R}_t^T}{\partial \theta_t}\left(\mathbf{x}_t^{[n]} - \mathbf{t}_t\right) \end{pmatrix} \qquad \frac{\partial \mathbf{h}^{[n]}(\cdot)}{\partial \mathbf{x}^{[n]}} = \mathbf{R}_t^T$$

$$\mathbf{C}_t^{[n]} = \left.\frac{\partial \mathbf{h}^{[n]}(\cdot)}{\partial \mathbf{x}}\right|_{x=\mu_{t|t-1}} = \begin{pmatrix} \frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}^{[r]}} & \mathbf{0} & \cdots & \frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}^{[n]}} & \cdots & \mathbf{0} \end{pmatrix}$$

pose block

landmark block

Data Association

The observation z_t will originate from a subset of landmarks in the state.

The order of these measurements does not necessarily match the order of the landmarks in the state.

Let $j(m)$ be the index of the landmark that generates the mth measurement block.

$$\mathbf{x} = \mathbf{x}^{[r]}, \mathbf{x}^{[1]}, \dots, \mathbf{x}^{[N]}$$

state (robot, N landmarks)

$$\mathbf{z} = \mathbf{z}^{[1]}, \dots, \mathbf{z}^{[M]}$$

measurements (M <= N)

$$\mathbf{h}(\mathbf{x}) = (\mathbf{h}^{[1]}(\mathbf{x}), \dots, \mathbf{h}^{[N]}(\mathbf{x}))^\top$$

measurement predictions (N)

$$\mathrm{j(m)} \in [1, 2, \dots N]$$

data association function
j(3) = 5 means
measurement 3 originates
from landmark 5

Data Association

For now we assume to know the assignment $j(m)$.

With this assignment we can build a prediction based on M measured landmarks, and its Jacobian!

$$\mathbf{h}(\mathbf{x}_t) = \begin{pmatrix} \mathbf{h}^{[j(1)]} \\ \mathbf{h}^{[j(2)]} \\ \vdots \\ \mathbf{h}^{[j(M)]} \end{pmatrix} \qquad \mathbf{C}_t = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial \mathbf{h}^{[j(1)]}}{\partial \mathbf{x}} \\ \frac{\partial \mathbf{h}^{[j(2)]}}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial \mathbf{h}^{[j(M)]}}{\partial \mathbf{x}} \end{pmatrix}$$

Populating the Map

Time: t

current state

$$\mu_t = \left(\; \mathbf{x}_t^{[r]} \; \right)$$

$$\mathbf{x}_t^{[r]} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix}$$

robot state

Id: 1

Id: 7

Id: 9

$$\textbf{id\_to\_state\_map} = \begin{pmatrix} -1 & -1 & \ldots & \ldots & -1 \end{pmatrix}$$

$$\textbf{state\_to\_id\_map} = \begin{pmatrix} -1 & -1 & \ldots & \ldots & -1 \end{pmatrix}$$

Populating the Map

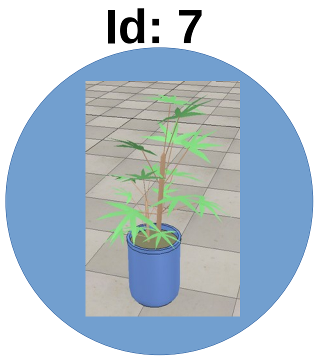
Time: t+1

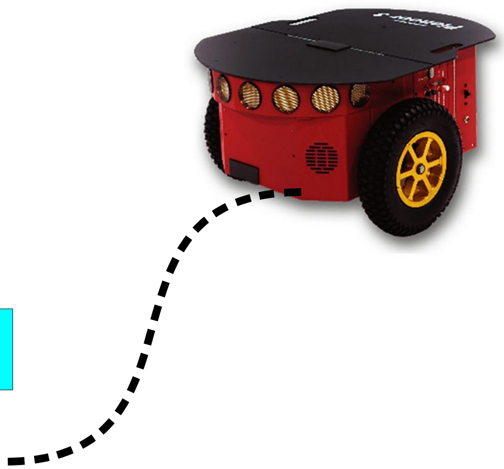
current state

$$\mu_{t+1} = \begin{pmatrix} \mathbf{x}_{t+1}^{[r]} \\ \boxed{\mathbf{x}_{t+1}^{[7]}} \end{pmatrix}$$

$$\mathbf{x}_t^{[7]} = \begin{pmatrix} x_t^{[7]} \\ y_t^{[7]} \end{pmatrix}$$

landmark state





Position 7

$$\mathbf{id\_to\_state\_map} = \begin{pmatrix} -1 & \dots & \boxed{1} & -1 & \dots & -1 \end{pmatrix}$$

$$\mathbf{state\_to\_id\_map} = \begin{pmatrix} \boxed{7} & -1 & \dots & \dots & -1 \end{pmatrix}$$

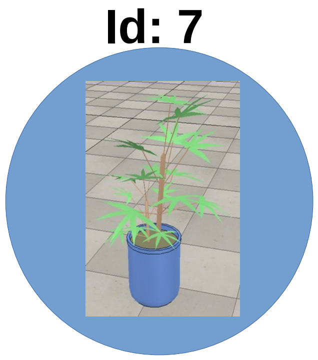
Populating the Map

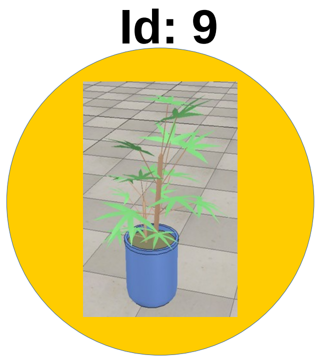
Time: t+2

current state

$$\mu_{t+2} = \begin{pmatrix} \mathbf{x}_{t+2}^{[r]} \\ \mathbf{x}_{t+2}^{[7]} \\ \mathbf{x}_{t+2}^{[9]} \end{pmatrix}$$

$$\mathbf{x}_t^{[9]} = \begin{pmatrix} x_t^{[9]} \\ y_t^{[9]} \end{pmatrix}$$





Position 9

$$\textbf{id\_to\_state\_map} = \begin{pmatrix} -1 & \dots & 1 & -1 & 2 & \dots & -1 \end{pmatrix}$$

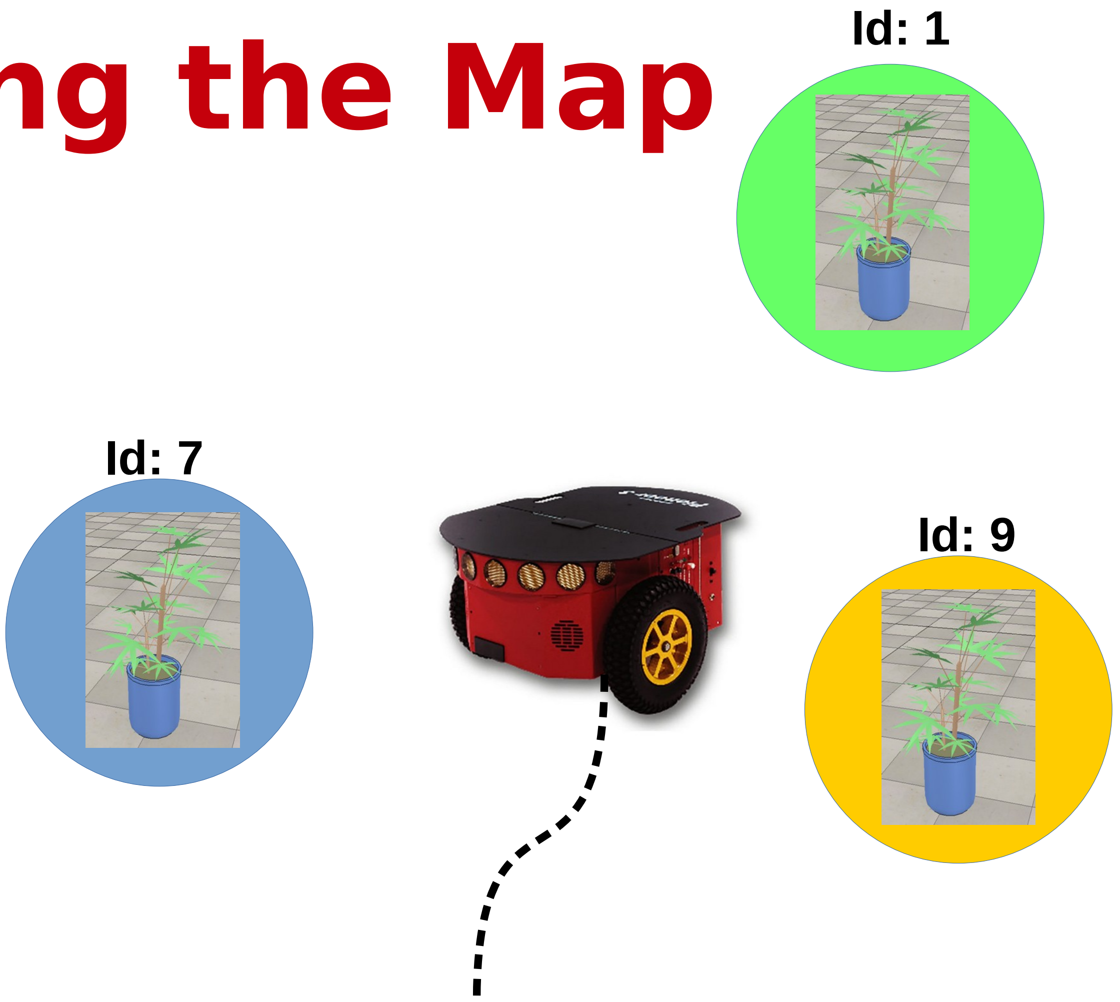
$$\textbf{state\_to\_id\_map} = \begin{pmatrix} 7 & 9 & -1 & \dots & \dots & -1 \end{pmatrix}$$

Populating the Map

Time: t+3

current state

$$\mu_{t+3} = \begin{pmatrix} \mathbf{x}_{t+3}^{[r]} \\ \mathbf{x}_{t+3}^{[7]} \\ \mathbf{x}_{t+3}^{[9]} \\ \mathbf{x}_{t+3}^{[1]} \end{pmatrix}$$



$$\mathbf{id\_to\_state\_map} = (\boxed{3} \quad \dots \quad \boxed{1} \quad -1 \quad \boxed{2} \quad \dots \quad -1 \quad)$$

$$\mathbf{state\_to\_id\_map} = (\boxed{7}\ \boxed{9}\ \boxed{1} \quad \dots \quad \dots \quad -1 \quad)$$

Updating the Map

Partition the sensed landmarks in two classes:

- Already known landmarks (part of the state)
  - Use them to perform an EKF correction
- New landmarks
  - Add them to the state **after** the EKF correction

Hands On!

g2o Wrapper

Load your V-REP acquired dataset

```
[land, pose, transition, obs] = loadG2o('my_dataset.g2o');
```

This call returns an array of objects with the fields:

This time we don’t know the landmarks!

```
pose =

  1x137 struct array containing the fields:

    id
    x
    y
    theta
```

```
transition =

  1x136 struct array containing the
fields:

    id_from
    id_to
    v
```

```
obs =

  1x136 struct array containing the fields:

    pose_id
    observation
```