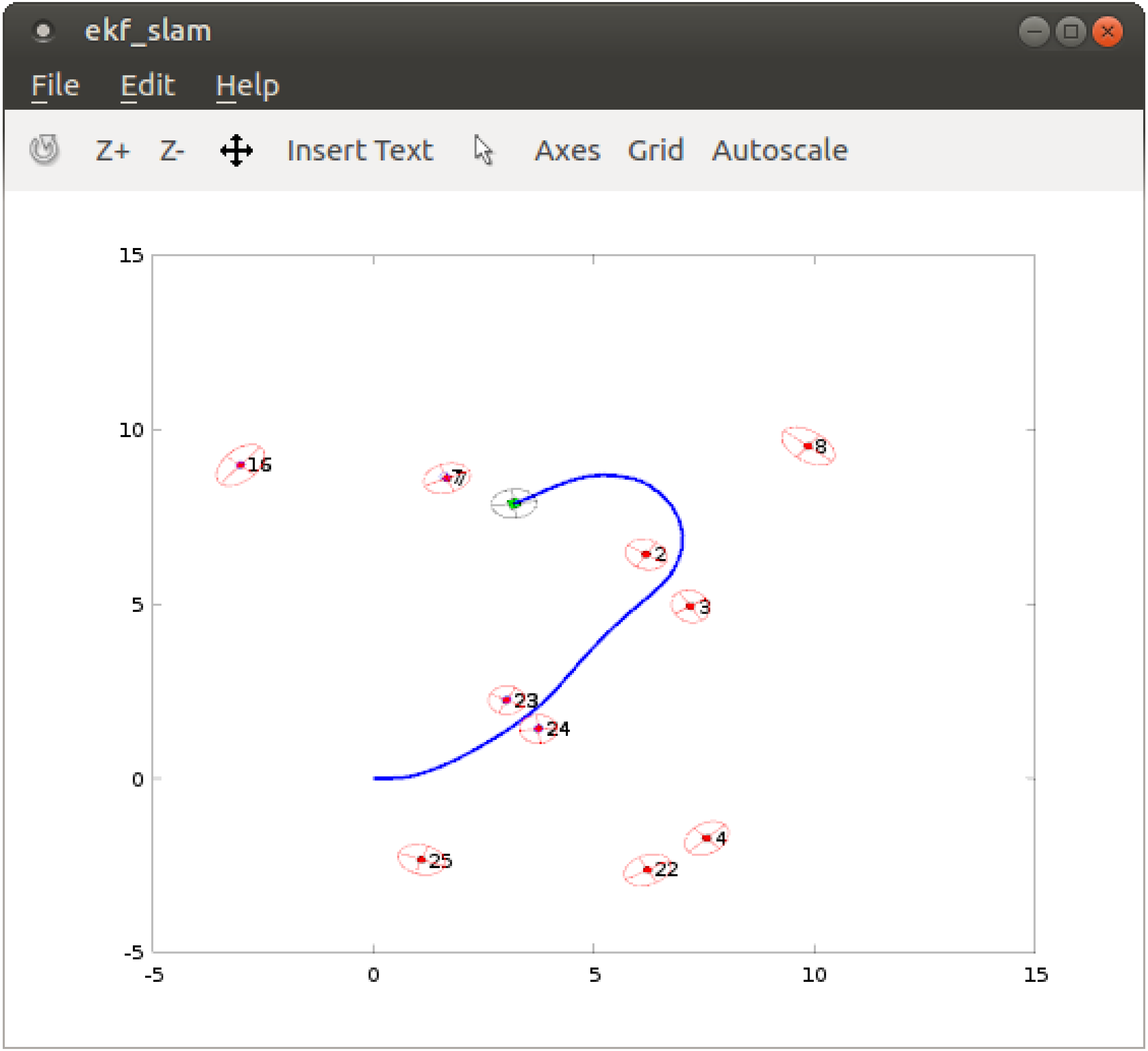
Probabilistic Robotics Course

EKF SLAM [Application]

Dominik Schlegel
schlegel@diag.uniroma1.it

Deptartment of Computer Control and Management Engineering
Sapienza University of Rome



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

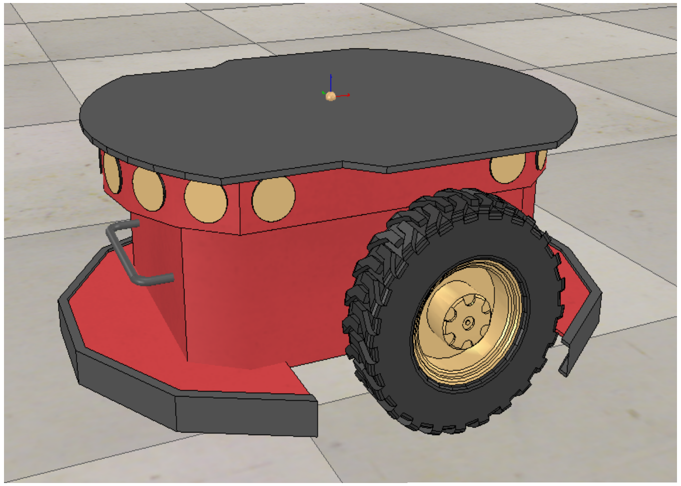
Outline

- 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: Domains

Predict: incorporate new control

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

Correct: incorporate new measurement

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

Domains

robot state

Define

- state space

$$\mathbf{x}_t^r = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} \in \Re^3$$

poses mapped to 3D vectors

landmarks in the state

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

- space of controls (inputs)

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

- space of observations (measurements)

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

EKF: Functions

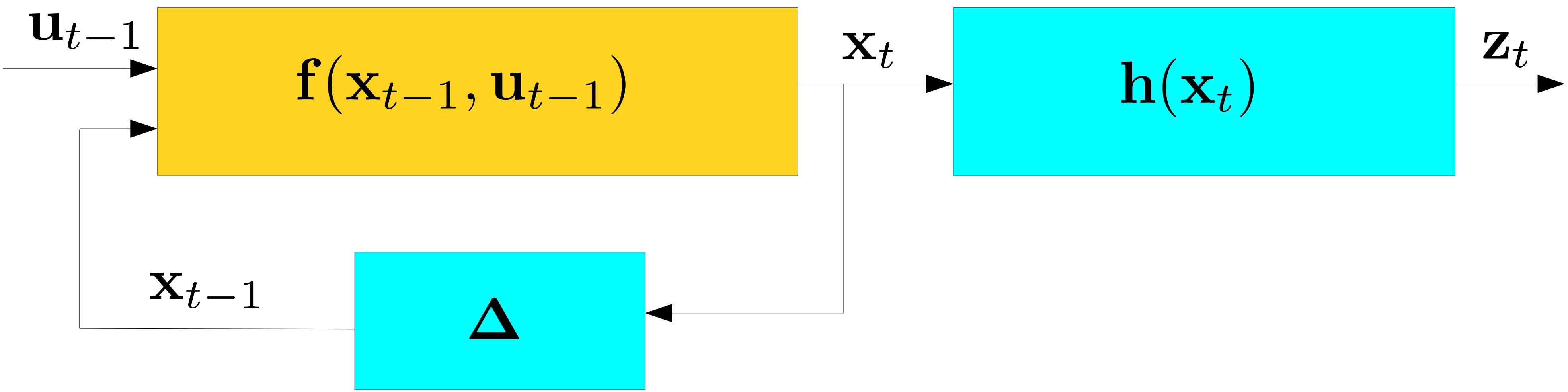
Predict: incorporate new control

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

Correct: incorporate new measurement

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

Transition Function

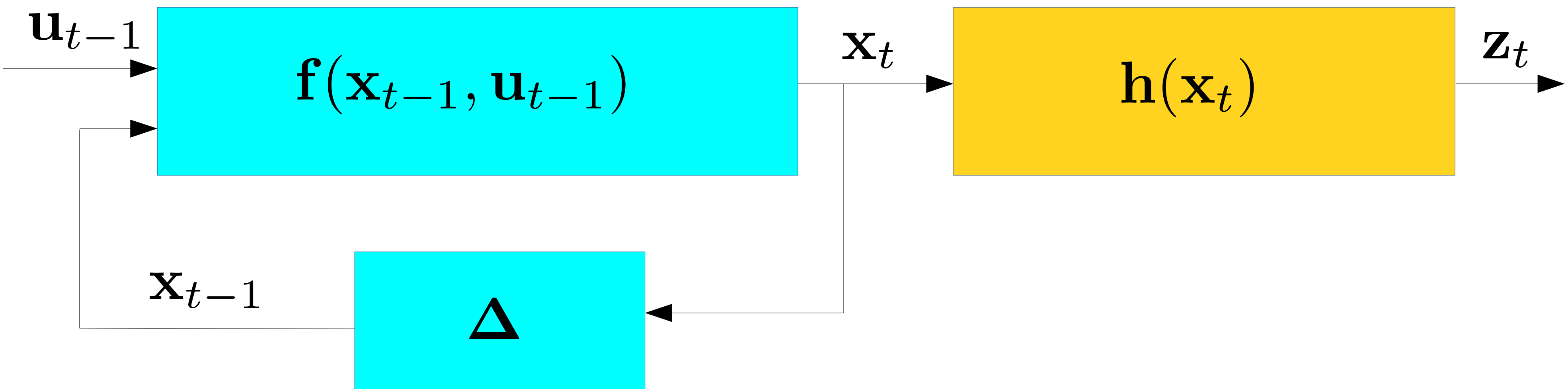


pose update (the robot moves)

the landmarks don't move

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

Measurement Function



robot pose

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

Landmark position

$n = 1, .., N$

We have N measurement functions, one for each landmark

relative position of the n^th landmark w.r.t the robot at time t

Noise considerations

Are our functions modeling reality 100%?

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_v
- velocity dependent terms whose standard deviation grows with the speed (i.d.)
  - translational noise $\sigma_{u_t^1}$
  - rotational noise $\sigma_{u_t^2}$

$$\mathbf{n}_{u,t} \sim \mathcal{N}(\mathbf{n}_{u,t}; \mathbf{0}, \underbrace{\begin{pmatrix} \sigma_v^2 + \sigma_{u_t^1}^2 & 0 \\ 0 & \sigma_v^2 + \sigma_{u_t^2}^2 \end{pmatrix}}_{\Sigma_u})$$

Measurement Noise

We assume it is zero mean and constant

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

EKF: Jacobians

Predict: incorporate new control

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

Correct: incorporate new measurement

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

Jacobian 1: State

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

$$\mathbf{A}_t = \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}^r} & \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}^{[1]}} & \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}^{[2]}} & \cdots & \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}^N} \end{pmatrix}$$

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

pose block

landmark 1 block

landmark **n** block

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

Jacobian 2: Controls

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

$$\mathbf{B}_t = \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{u}} = \begin{pmatrix} \cos(\theta_{t-1}) & 0 \\ \sin(\theta_{t-1}) & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix} \begin{array}{l} \\ \leftarrow \text{pose block} \\ \\ \leftarrow \text{landmark block 1} \\ \\ \\ \leftarrow \text{landmark block N} \end{array}$$

Jacobian 3: Measurements

For the measurement function, we need to derive w.r.t **all** state variables

Prediction of the n^{th} landmark observation:

n^{th} landmark location

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

pose

The n^{th} measurement Jacobian will have a block structure

$$\mathbf{C}_t^{[n]} = \frac{\partial \mathbf{h}^{[n]}(\cdot)}{\partial \mathbf{x}_t} = \begin{pmatrix} \frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}_t^r} & \mathbf{0} & \cdots & \frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}_t^{[n]}} & \cdots & \mathbf{0} \end{pmatrix}$$

pose block

landmark block

Jacobian 3: Measurements

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

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

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

$$\mathbf{C}_t^{[n]} = \frac{\partial \mathbf{h}^{[n]}(\cdot)}{\partial \mathbf{x}_t} = \begin{pmatrix} \underbrace{\frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}_t^r}}_{\text{pose block}} & \mathbf{0} & \cdots & \underbrace{\frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}_t^{[n]}}}_{\text{landmark block}} & \cdots & \mathbf{0} \end{pmatrix}$$

Data Association

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

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

Let $j(m)$ be the index of the landmark that generates the m^th observation block.

$\mathrm{j}(m) \in [1, 2, \ldots N]$ — assignment function j(3)=5 means measurement 3 originates from landmark 5

$\mathbf{x} = \mathbf{x}^r, \mathbf{x}^{[1]}, \ldots, \mathbf{x}^{[N]}$ — state (robot, N landmarks)

$\mathbf{z} = \mathbf{z}^{[1]}, \ldots, \mathbf{z}^{[M]}$ — measurements

$\hat{\mathbf{z}} = \hat{\mathbf{z}}^{[1]}, \ldots, \hat{\mathbf{z}}^{[N]}$ — predictions

Data Association

For now we assume to know the assignment j.

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

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

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:

```
land =

  1x25 struct array containing the fields:

    id
    x_pose
    y_pose
```

```
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
```

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
```

EKF SLAM

```
#load your own dataset dataset, without landmarks (first entry remains empty)
[_, poses, transitions, observations] = loadG2o("../datasets/dataset_point.g2o");

#set initial pose at the origin - we don't know the map and neither our location
mu = [0;  #x coordinate
      0;  #y coordinate
      0]; #orientation theta (yaw angle)

#initialize covariance: high value means high uncertainty
sigma = eye(3);

#bookkeeping: to and from mapping between robot pose (x,y, theta) and landmark indices (i)
#all mappings are initialized with invalid value -1 (meaning that the index is not mapped)
#since we do not know how many landmarks we will observe, we allocate a large enough buffer
id_to_state_map = ones(10000, 1)*-1;
state_to_id_map = ones(10000, 1)*-1;

#simulation cycle: for the number of transitions recorded in the dataset
for t = 1:length(transitions)

  #obtain current transition
  transition = transitions(t);

  #obtain current observation
  observation = observations(t);

  #EKF predict
  [mu, sigma] = prediction(mu, sigma, transition);

  #EKF correct
  [mu, sigma, id_to_state_map, state_to_id_map] = correction(mu,
                                                              sigma, observation,
                                                              id_to_state_map,
                                                              state_to_id_map);

endfor
```

Populating the Map

Time: t

current mean

$$\mu_t = \begin{pmatrix} x_t^r \\ y_t^r \\ \theta_t^r \end{pmatrix}$$

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

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

Populating the Map

Time: t+1

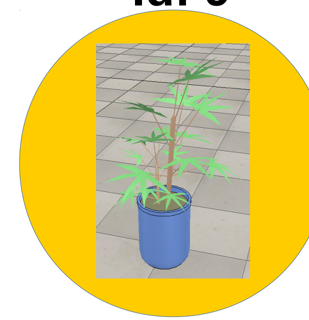
current mean

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

$$\mathbf{x}^{[n]}_t = \begin{pmatrix} x^{[n]}_t \\ y^{[n]}_t \end{pmatrix}$$

Id: 7

Id: 9



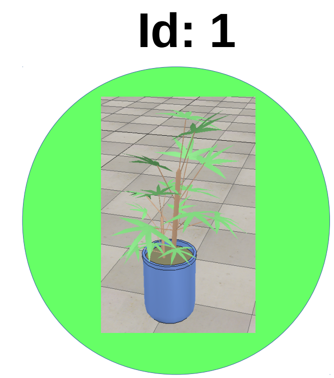
Position 7

Position 9

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

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

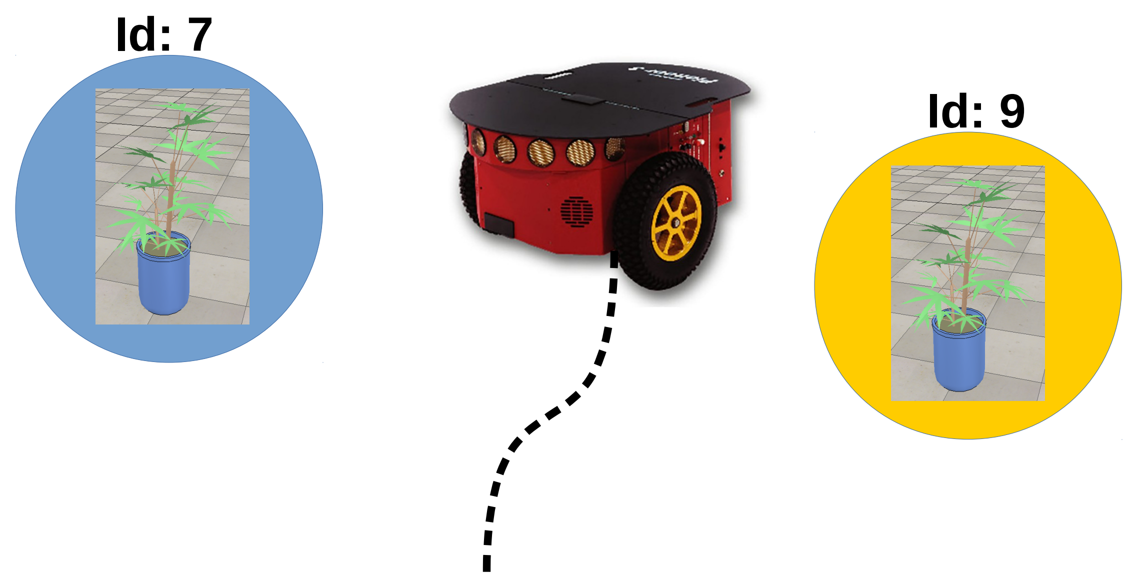
Populating the Map



Time: t+2

current mean

$$\mu_{t+2} = \begin{pmatrix} x^r_{t+2} \\ y^r_{t+2} \\ \theta^r_{t+2} \\ \mathbf{x}^{[7]}_{t+2} \\ \mathbf{x}^{[9]}_{t+2} \\ \mathbf{x}^{[1]}_{t+2} \end{pmatrix}$$



$$\textbf{id\_to\_state\_map} = \begin{pmatrix} 3 & \ldots & 1 & -1 & 2 & \ldots & -1 \end{pmatrix}$$

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

Correction

Two cases may arise:

- I have already seen the current landmark
- The current one is a new landmark

example:

Time: t+3

current observations : { 7 , 9 , 2}

$$\mathbf{id\_to\_state\_map} = \begin{pmatrix} 3 & -1 & \dots & 1 & -1 & 2 & \dots & -1 \end{pmatrix}$$

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

Correction

Two cases may arise:

- I have already seen the current landmark
- The current one is a new landmark

Trick

First correct with the **reobserved** landmarks, then add the **new** ones

example:

Time: t+3

current observations : { 7 , 9 , $\boxed{2}$}

2 is a new landmark

$$\mathbf{id\_to\_state\_map} = \begin{pmatrix} 3 & \boxed{-1} & \ldots & 1 & -1 & 2 & \ldots & -1 \end{pmatrix}$$

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

Correction

First consider only the reobserved landmarks

Time: t+3

observations : { 7 , 9 , _ }

I need to know what is the position of landmark 7 in the mu vector

current landmark: **7**

$$\mathbf{C}_t^{[n]} = \frac{\partial \mathbf{h}^{[n]}(\cdot)}{\partial \mathbf{x}_t} = \begin{pmatrix} \frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}_t^r} & \mathbf{0} & \cdots & \boxed{\frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}_t^{[n]}}} & \cdots & \mathbf{0} \end{pmatrix}$$

$$\textbf{id\_to\_state\_map} = \begin{pmatrix} 3 & -1 & \dots & 1 & -1 & 2 & \dots & -1 \end{pmatrix}$$

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

Correction

First consider only the reobserved landmarks

Time: t+3

observations : { 7 , 9 , _ }

current landmark: **7**

$$\mathbf{C}_t^{[n]} = \frac{\partial \mathbf{h}^{[n]}(\cdot)}{\partial \mathbf{x}_t} = \begin{pmatrix} \frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}_t^r} & \mathbf{0} & \cdots & \frac{\partial \mathbf{h}^{[n]}}{\partial \mathbf{x}_t^{[n]}} & \cdots & \mathbf{0} \end{pmatrix}$$

It's the first landmark we have seen

$$\mathbf{id\_to\_state\_map} = \begin{pmatrix} 3 & -1 & \ldots & \boxed{1} & -1 & 2 & \ldots & -1 \end{pmatrix}$$

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

Same reasoning holds for landmark 9

Correction

Correct/Update the filter with the reobserved landmarks, then add the **new** landmark **2**

Add initial mean and covariance of the new landmark

$$\mu_{t+3} = \begin{pmatrix} x^r_{t+3} \\ y^r_{t+3} \\ \theta^r_{t+3} \\ \mathbf{x}^{[7]}_{t+3} \\ \mathbf{x}^{[9]}_{t+3} \\ \mathbf{x}^{[1]}_{t+3} \\ \mathbf{x}^{[2]}_{t+3} \end{pmatrix} \quad \mathbf{\Sigma}_{t+3} = \begin{pmatrix} \mathbf{\Sigma}_{t+3} & \\ & \mathbf{\Sigma}^{[2]}_{t+3} \end{pmatrix}$$

$$\mathbf{id\_to\_state\_map} = \begin{pmatrix} 3 & -1 & \ldots & 1 & -1 & 2 & \ldots & -1 \end{pmatrix}$$

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

Correction

Correct/Update the filter with the reobserved landmarks, then add the **new** landmark 2

$$\mu_{t+3} = \begin{pmatrix} x^r_{t+3} \\ y^r_{t+3} \\ \theta^r_{t+3} \\ \mathbf{x}^{[7]}_{t+3} \\ \mathbf{x}^{[9]}_{t+3} \\ \mathbf{x}^{[1]}_{t+3} \\ \mathbf{x}^{[2]}_{t+3} \end{pmatrix} \quad \boldsymbol{\Sigma}_{t+3} = \begin{pmatrix} \boldsymbol{\Sigma}_{t+3} & \\ & \boldsymbol{\Sigma}^{[2]}_{t+3} \end{pmatrix}$$

$$\textbf{id\_to\_state\_map} = \begin{pmatrix} 3 & 4 & \ldots & 1 & -1 & 2 & \ldots & -1 \end{pmatrix}$$

$$\textbf{state\_to\_id\_map} = \begin{pmatrix} 7 & 9 & 1 & 2 & \ldots & -1 \end{pmatrix}$$

Update vectors