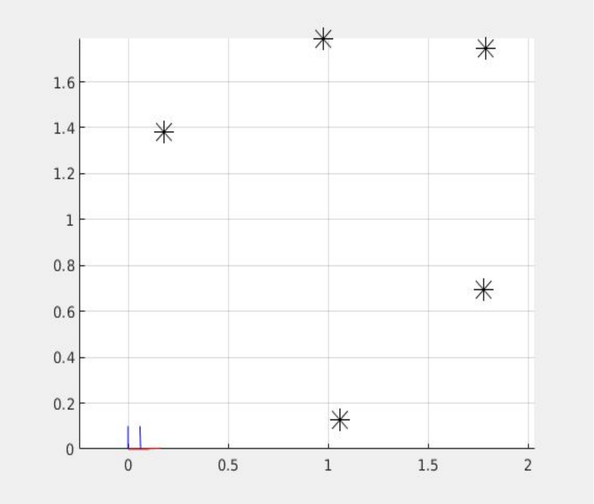
First part - Mapping
Second part - SLAM

Feras Dayoub

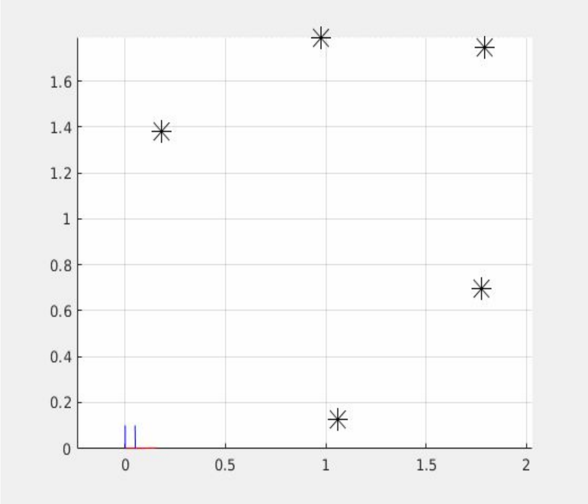
Learning objectives

- Mapping using an extended Kalman filter.
- SLAM using an extended Kalman filter.

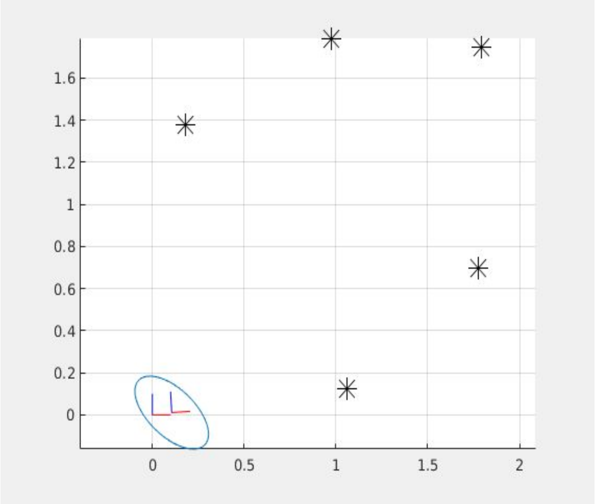
Last Lecture Recap - Localization



Ground truth (unknown)



Odometry



EKF localization

Lecture 10 Recap

Prediction step:

$$\bar{\mu}_t = f(\mu_{t-1}, \mathbf{u}_t)$$

$$\bar{\Sigma}_t = \mathbf{J}_{x_t} \mathbf{\Sigma}_{t-1} \mathbf{J}_{x_t}^T + \mathbf{J}_{u_t} \mathbf{R} \mathbf{J}_{u_t}^T$$

Update step:

For each landmark $\mathbf{z}_t^i$ do:

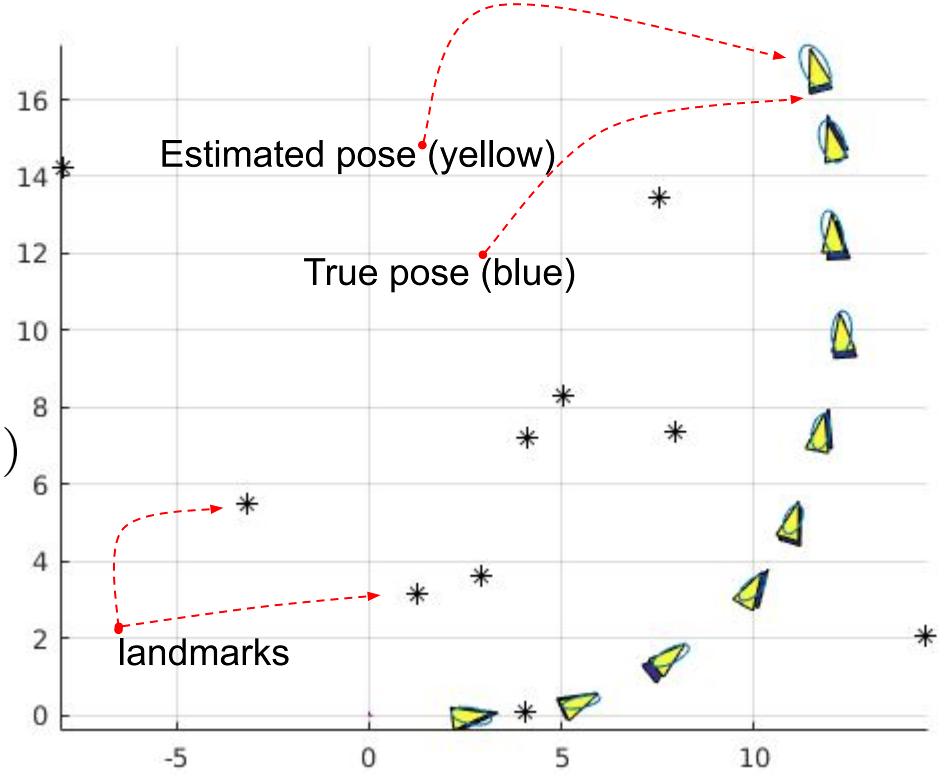
$$\bar{\mu}_t = \bar{\mu}_t + \mathbf{K}_t^i (\mathbf{z}_t^i - h(\bar{\mu}_t, i))$$

$$\bar{\mathbf{\Sigma}}_t = (\mathbf{I} - \mathbf{K}_t^i \mathbf{G}_t^i) \bar{\mathbf{\Sigma}}_t$$

end

$$\mu_t = \bar{\mu}_t$$

$$\mathbf{\Sigma}_t = \bar{\mathbf{\Sigma}}_t$$

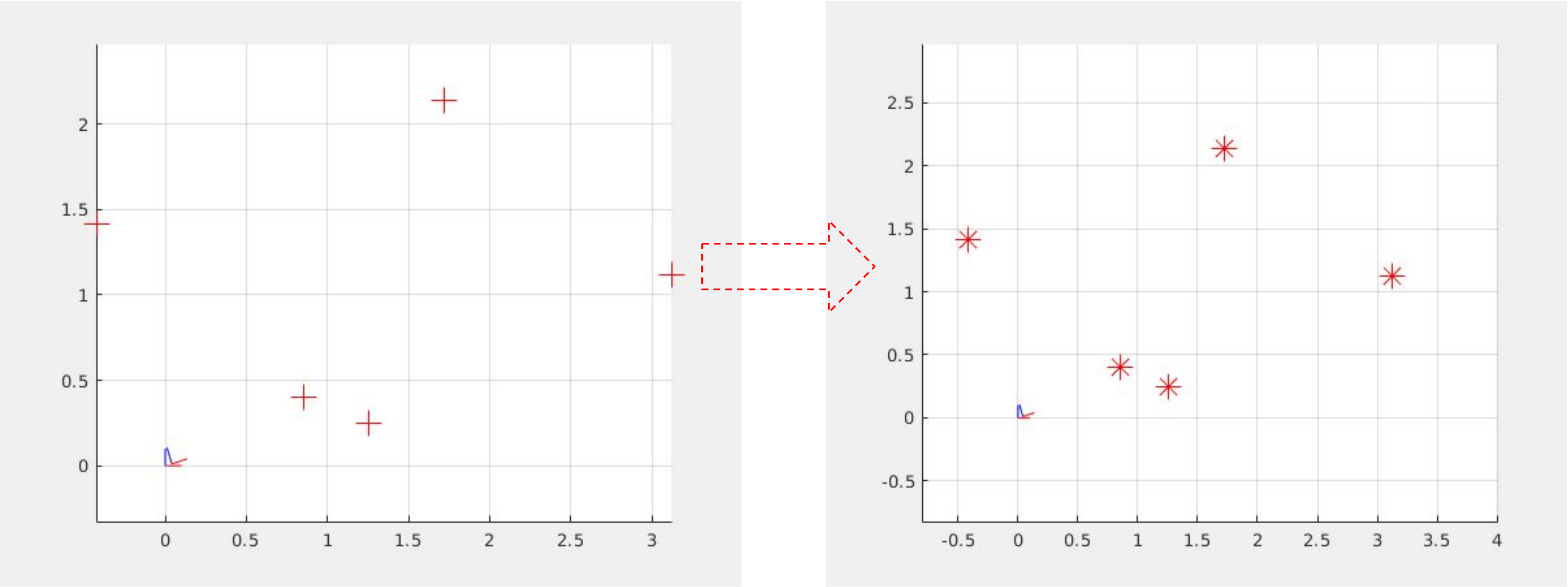


Mapping
Assumptions

- The robot knows its pose with absolute certainty.
- The robot is equipped with noisy range and bearing sensor.
- We have a way to associate the measurements with the already mapped landmarks when they appear in the view again.
- The state and the noise are Normally distributed.

The task

- Estimate the position of the landmarks in the map.

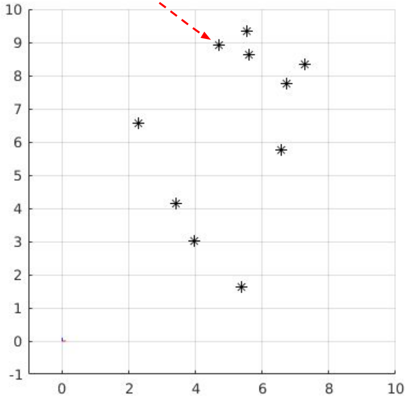


0 0.5 1 1.5 2
0 0.5 1 1.5 2 2.5 3
-0.5 0 0.5 1 1.5 2 2.5
-0.5 0 0.5 1 1.5 2 2.5 3 3.5 4

The state vector is the map

- The state vector is much larger than what we saw in the localization case.

$$\mathbf{M} = \begin{bmatrix} x_{l_1} \\ y_{l_1} \\ \vdots \\ x_{l_n} \\ y_{l_n} \end{bmatrix} \sim \mathcal{N}(\mu, \boldsymbol{\Sigma})$$



The covariance matrix

Is this matrix symmetric?

$$\Sigma_t = \begin{bmatrix} \Sigma_{l_{11}} & \Sigma_{l_{12}} & \cdots & \Sigma_{l_{1n}} \\ \vdots & \vdots & \vdots & \vdots \\ \Sigma_{l_{n1}} & \Sigma_{l_{n2}} & \cdots & \Sigma_{l_{nn}} \end{bmatrix}$$

The covariance matrix is much bigger and can be written in blocks. Each block tell us the correlation between two landmarks.

$$\Sigma_{l_{ij}} = \begin{bmatrix} \sigma_{x_i x_j} & \sigma_{x_i y_j} \\ \sigma_{y_i x_j} & \sigma_{y_i y_j} \end{bmatrix}$$

What if i == j?

Let's start from the EKF set of equations

Prediction step:

$$\bar{\mu}_t = f(\mu_{t-1}, \mathbf{u}_t)$$

$$\bar{\Sigma}_t = \mathbf{J}_{x_t} \mathbf{\Sigma}_{t-1} \mathbf{J}_{x_t}^T + \mathbf{J}_{u_t} \mathbf{R} \mathbf{J}_{u_t}^T$$

Given that the state vector only contain the positions of the landmarks, what is **f** and what are the the Jacobians matrices?

Update step:

For each landmark do:

$$\mu_t = \bar{\mu}_t + \mathbf{K}_t(\mathbf{z}_t - h(\bar{\mu}_t))$$

$$\mathbf{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{G}_t)\bar{\mathbf{\Sigma}}_t$$

The prediction step

- The landmarks are static and do not change between time steps.

$$\bar{\mu}_t = \mu_{t-1}$$

$$\bar{\boldsymbol{\Sigma}}_t = \boldsymbol{\Sigma}_{t-1}$$

The same set of equations

Prediction step:

$$\bar{\mu}_t = \mu_{t-1}$$

$$\bar{\mathbf{\Sigma}}_t = \mathbf{\Sigma}_{t-1}$$

Update step:

For each landmark do:

$$\mu_t = \bar{\mu}_t + \mathbf{K}_t(\mathbf{z}_t - h(\bar{\mu}_t))$$

$$\mathbf{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{G}_t)\bar{\mathbf{\Sigma}}_t$$

In the context of mapping, what is the function **h**?

The same set of equations

Prediction step:

$$\bar{\mu}_t = \mu_{t-1}$$

$$\bar{\mathbf{\Sigma}}_t = \mathbf{\Sigma}_{t-1}$$

The map

$$h(\bar{\mu}_t, \mathbf{x}_t^r)$$

The true pose of the robot (known)

Update step:

<u>For each observed landmark do:</u>

$$\mu_t = \bar{\mu}_t + \mathbf{K}_t(\mathbf{z}_t - \cancel{h(\bar{\mu}_t)})$$

$$\mathbf{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{G}_t)\bar{\mathbf{\Sigma}}_t$$

The same set of equations

Prediction step:

$$\bar{\mu}_t = \mu_{t-1}$$

$$\bar{\mathbf{\Sigma}}_t = \mathbf{\Sigma}_{t-1}$$

We also assume known correspondences

Update step:

For each observed landmark do:

$$\mu_t = \bar{\mu}_t + \mathbf{K}_t^i(\mathbf{z}_t^i - h(\bar{\mu}_t, i, \mathbf{x}_t^r))$$

$$\mathbf{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t^i \mathbf{G}_t^i)\bar{\mathbf{\Sigma}}_t$$

$$\mathbf{z}_t^i = \begin{bmatrix} r^i \\ \beta^i \end{bmatrix}$$

The same measurement model we used for localization

$$h(\bar{\mu}_t, i, \mathbf{x}_t^r) = \begin{bmatrix} r^i \\ \beta^i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_r - x_{l_i})^2 + (y_r - y_{l_i})^2} \\ atan2(y_{l_i} - y_r, x_{l_i} - x_r) - \theta_r \end{bmatrix}$$

Range and bearing with respect to robot's own frame of reference at time step t.

Coordinates of landmark i

The true pose of the robot

The same set of equations

Prediction step:

$$\bar{\mu}_t = \mu_{t-1}$$

$$\bar{\boldsymbol{\Sigma}}_t = \boldsymbol{\Sigma}_{t-1}$$

Update step:

For each observed landmark do:

$$\mu_t = \bar{\mu}_t + \mathbf{K}_t^i(\mathbf{z}_t^i - h(\bar{\mu}_t, i, \mathbf{x}_t^r))$$

$$\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t^i \mathbf{G}_t^i)\bar{\boldsymbol{\Sigma}}_t$$

Given that we mapped **n** landmarks at time step **t**, what are the dimensions of these matrices?

$$\mathbf{K}_t = \bar{\boldsymbol{\Sigma}}_t \mathbf{G}_t^T (\mathbf{G}_t \bar{\boldsymbol{\Sigma}}_t \mathbf{G}_t^T + \mathbf{Q})^{-1}$$

The Jacobian matrix of the measurement function

$$\mathbf{G}_t^i = \frac{\partial h(\mu_t, i, \mathbf{x}_t^r)}{\partial \mu_t}$$

$$= \begin{bmatrix} 0 & \dots & \frac{x_{l_i} - x_r}{r} & \frac{y_{l_i} - y_r}{r} & \dots & 0 \\ 0 & \dots & -\frac{y_{l_i} - y_r}{r^2} & \frac{x_{l_i} - x_r}{r^2} & \dots & 0 \end{bmatrix}$$

The same set of equations

Prediction step:

$$\bar{\mu}_t = \mu_{t-1}$$

$$\bar{\boldsymbol{\Sigma}}_t = \boldsymbol{\Sigma}_{t-1}$$

Update step:

For each observed landmark do:

What if we observe a landmark for the first time (i.e it is not in our state vector yet).

$$\mu_t = \bar{\mu}_t + \mathbf{K}_t^i(\mathbf{z}_t^i - h(\bar{\mu}_t, i, \mathbf{x}_t^r))$$

$$\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t^i \mathbf{G}_t^i)\bar{\boldsymbol{\Sigma}}_t$$

Landmark initialization

$$\bar{\mu_t}^* = \begin{bmatrix} \bar{\mu_t} \\ l_{new} \end{bmatrix} = \begin{bmatrix} \bar{\mu_t} \\ x_{l_{new}} \\ y_{l_{new}} \end{bmatrix}$$

Given that the robot observes range and bearing to a landmark in its own frame of reference, how can we find the coordinates of the new landmark in the map frame?

Simply expand the state vector with the coordinates of the new landmark in the map.

The landmark initialization function

$$\mathbf{z} = \begin{bmatrix} r \\ \beta \end{bmatrix}$$

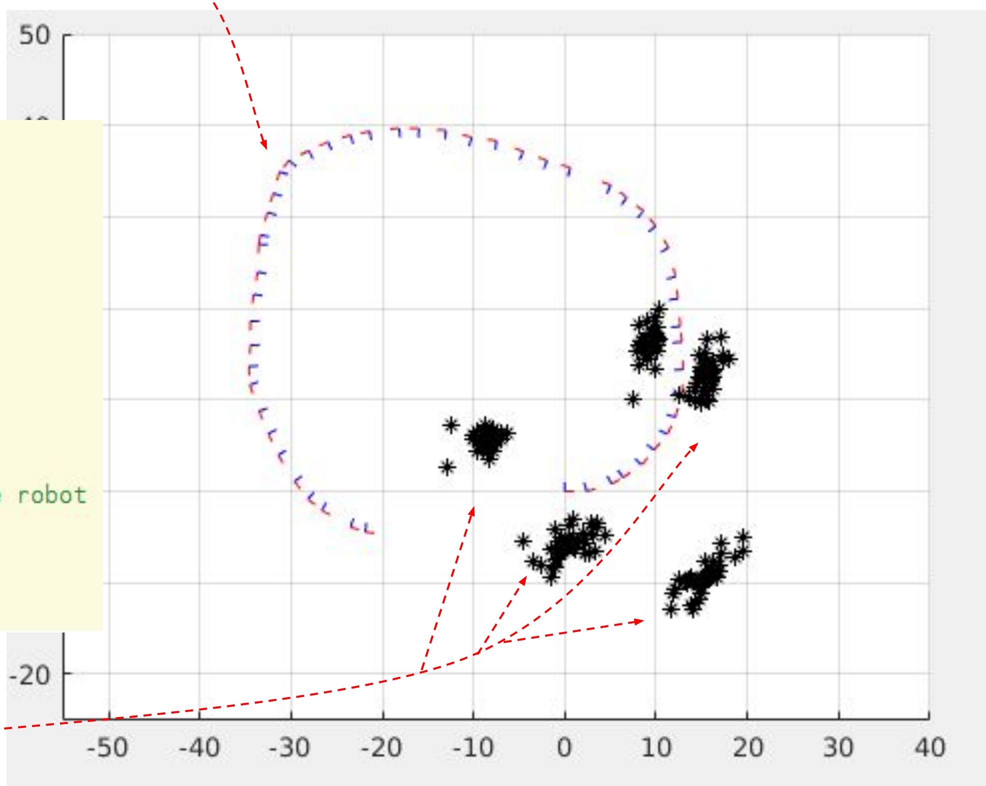
$$l_{new} = q(\mathbf{z}_t^{new}, \mathbf{x}_t^r)$$

Range and bearing to a never seen before landmark.

$$l_{new} = \begin{bmatrix} x_r + r \times cos(\theta_r + \beta) \\ y_r + r \times sin(\theta_r + \beta) \end{bmatrix}$$

True pose of the robot

```
%%
load_data()
% this simulator runs for 50 steps
nsteps = 50;
for k = 1:nsteps
    % the true pose of the robot is known
    xr          = get_pose(k);
    plot_robot(xr)
    % set of ranges and bearings to the landmarks
    z          = sense(k);
    for i=1:length(z)
       zi  = z(i,:);
       % plot the (x,y) of each landmark based on the pose of the robot
       l   = initL(zi,xr);
       scatter (l(1),l(2),'k*');
    end
end
```



Running the initialisation function after each time step. No filtering

What about the covariance matrix?

What is this matrix?

$$\bar{\Sigma}_t^* = \begin{bmatrix} \bar{\Sigma}_t & 0 \\ 0 & \mathbf{L_z Q L_z^T} \end{bmatrix}$$

The covariance matrix expands as well!

Zero matrices!

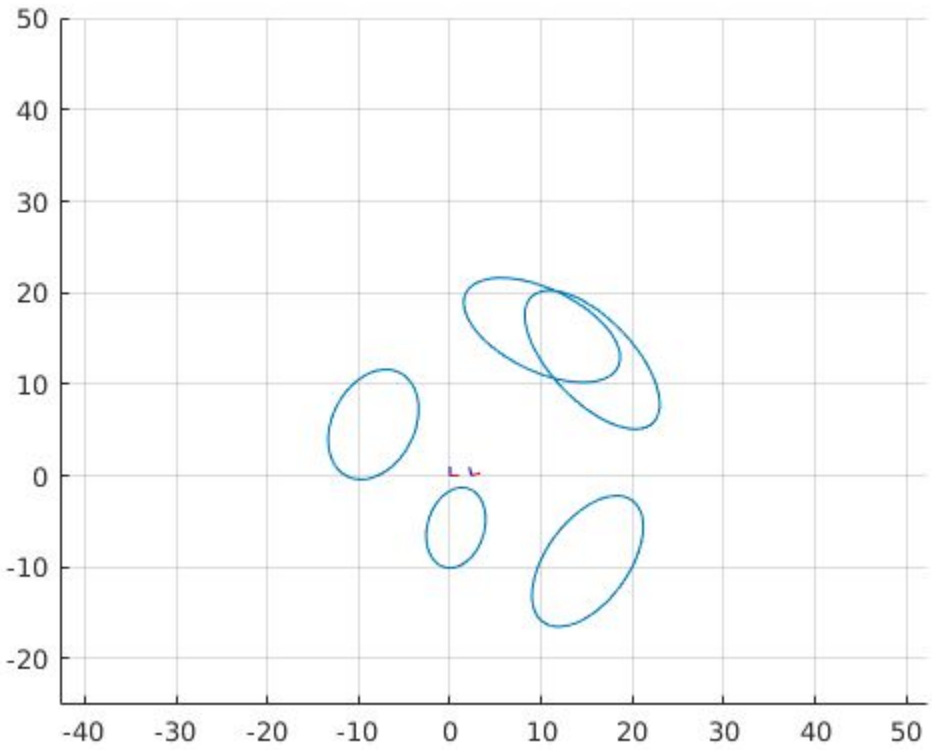
The covariance of the sensor noise.

The Jacobian of the landmark initialisation function

$$\mathbf{L}_z = \frac{\partial q(\mathbf{z}, \bar{\mu}_t)}{\partial \mathbf{z}}$$

$$= \boxed{\quad ? \quad}$$

At time step t=1 in the case where we have observed all the landmarks for the first time

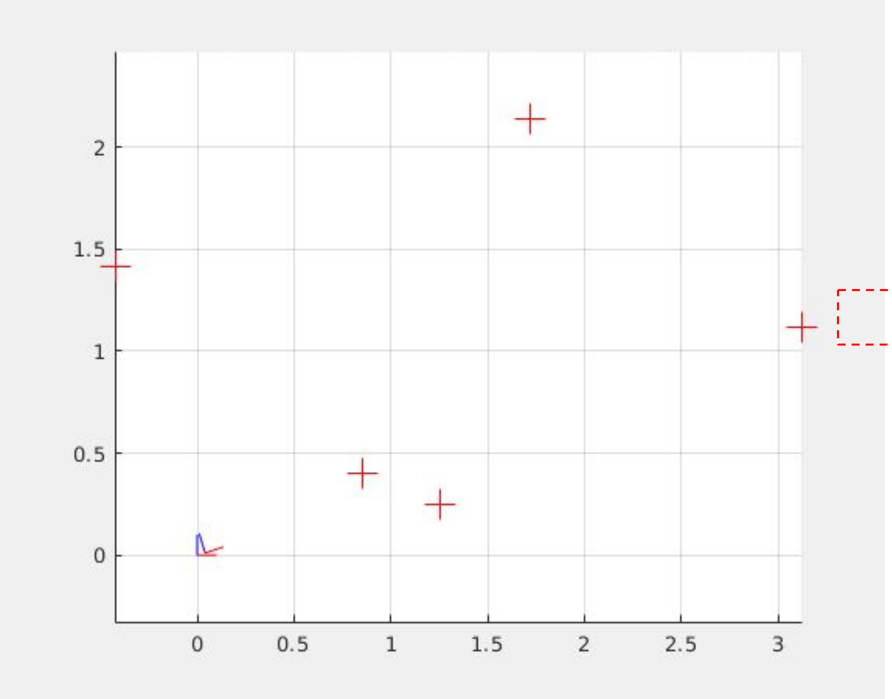


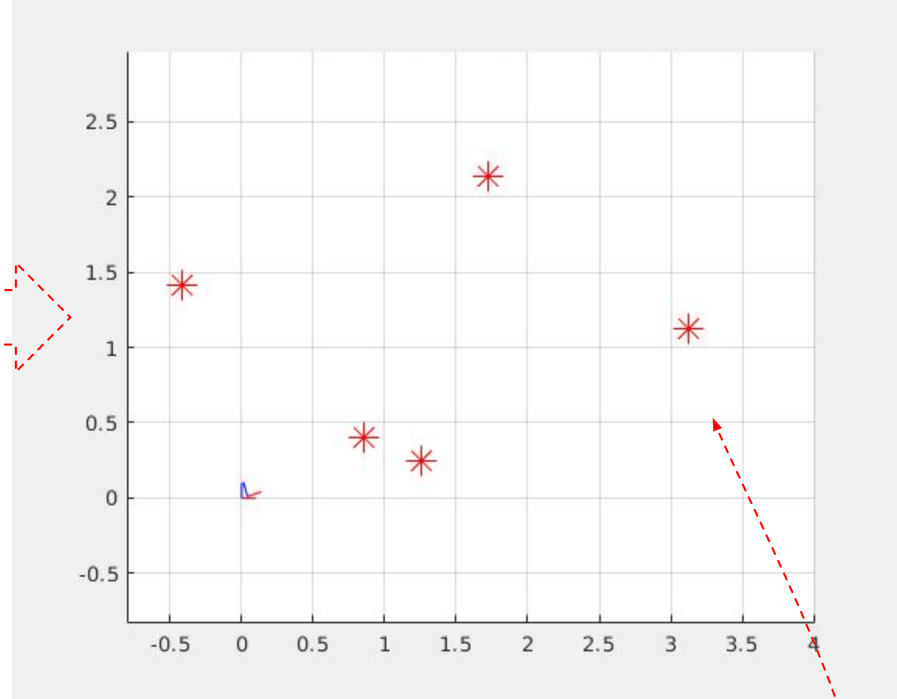
Putting it all together

1. Make a new Measurement (range and bearing to a landmark).
2. if we have not seen the landmark before:
    - Do landmark initialization based on the robot current pose.

    else

    - Predict the landmark position based on the robot current pose.
3. Update the state vector and the covariance.
4. Move.
5. Go to 1.

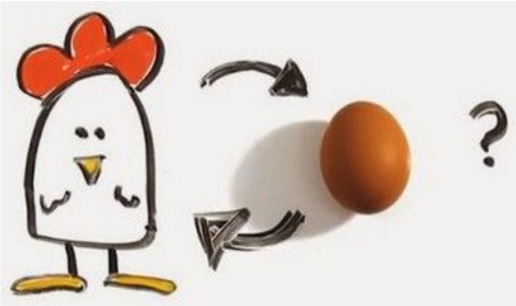




The ellipses are our 3-sigma bounds confidence of the position of the landmarks.
The red stars are the true (unknown) position of the landmarks and the black stars are our estimate.

Simultaneous Localization And Mapping

Feras Dayoub



Learning objectives

- SLAM using an extended Kalman filter.

Lecture 9 - 10 recap

Prediction step:

$$\bar{\mu}_t = f(\mu_{t-1}, \mathbf{u}_t)$$

$$\bar{\Sigma}_t = \mathbf{J}_{x_t} \mathbf{\Sigma}_{t-1} \mathbf{J}_{x_t}^T + \mathbf{J}_{u_t} \mathbf{R} \mathbf{J}_{u_t}^T$$

Update step:

For each observed landmark do:

$$\bar{\mu}_t = \bar{\mu}_t + \mathbf{K}_t^i (\mathbf{z}_t^i - h(\bar{\mu}_t, i))$$

$$\bar{\mathbf{\Sigma}}_t = (\mathbf{I} - \mathbf{K}_t^i \mathbf{G}_t^i) \bar{\mathbf{\Sigma}}_t$$

Prediction step:

$$\bar{\mu}_t = \mu_{t-1}$$

$$\bar{\mathbf{\Sigma}}_t = \mathbf{\Sigma}_{t-1}$$

Update step:

For each observed landmark do:

$$\bar{\mu}_t = \bar{\mu}_t + \mathbf{K}_t^i (\mathbf{z}_t^i - h(\bar{\mu}_t, i, \mathbf{x}_t^r))$$

$$\bar{\mathbf{\Sigma}}_t = (\mathbf{I} - \mathbf{K}_t^i \mathbf{G}_t^i) \bar{\mathbf{\Sigma}}_t$$

localization **mapping**

Assumptions

- The robot does not know its pose in the map.
- The wheel encoders are noisy.
- The robot does not know the position of the landmarks in the map.
- The sensor onboard the robot is noisy.
- The robot can associate the measurements with the landmarks.

The task

- The robot should **localize itself** inside a map using a set of landmarks and at the same time use its pose and sensor **to map** the positions of the landmarks.

SLAM: the chicken or egg problem

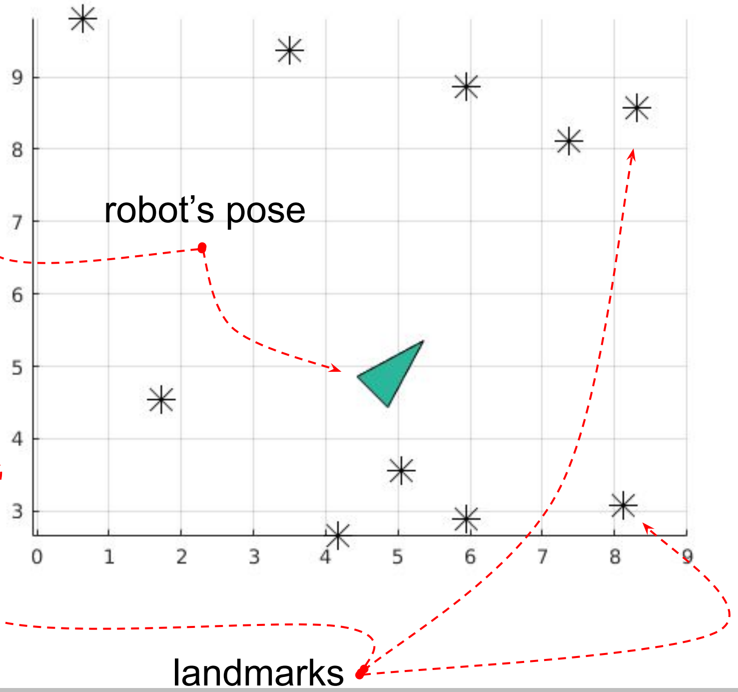
- As we saw in lecture 9, we need the position of the landmarks (i.e the map) to estimate the pose of the robot.
- And we saw in lecture 10 that in order to estimate the position of the landmarks in the map we need the true pose of the robot.
- In this lecture we are going to do the two above processes at the same time. This is called simultaneous localisation and mapping (SLAM).

Localize yourself in a map that you are building using the estimation of your pose in it!

The state vector

- The state vector contains both the pose of the robot and the positions of the landmarks in the map.

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}^r \\ M \end{bmatrix} = \begin{bmatrix} x_r \\ y_r \\ \theta_r \\ x_{l_1} \\ y_{l_1} \\ \vdots \\ x_{l_n} \\ y_{l_n} \end{bmatrix}$$



We still live in a Gaussian world!

$$\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$$

You used this in the localization case

You used this in the mapping case

$$\mu = \begin{bmatrix} \mu^r \\ \mu_{l1} \\ \vdots \\ \mu_{ln} \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \boxed{\Sigma_{\mathbf{x}^r}} & \Sigma_{\mathbf{x}^r l_1} & \cdots & \Sigma_{\mathbf{x}^r l_n} \\ \Sigma_{l_1 \mathbf{x}^r} & \Sigma_{l_{11}} & \cdots & \Sigma_{l_{1n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{l_n \mathbf{x}^r} & \Sigma_{l_{n1}} & \cdots & \Sigma_{l_{nn}} \end{bmatrix}$$

The mean vector
of the robot pose

The same set of equations

Prediction step:

$$\bar{\mu}_t = f(\mu_{t-1}, \mathbf{u}_t)$$

$$\bar{\Sigma}_t = \mathbf{J}_{x_t}\mathbf{\Sigma}_{t-1}\mathbf{J}_{x_t}^T + \mathbf{J}_{u_t}\mathbf{R}\mathbf{J}_{u_t}^T$$

- We predict both the pose of the robot and the positions of the landmarks.
- In the prediction step the robot moves and the landmarks stay static.

Update step:

For each landmark $\mathbf{z}_t^i$ do:

$$\bar{\mu}_t = \bar{\mu}_t + \mathbf{K}_t^i(\mathbf{z}_t^i - h(\bar{\mu}_t, i))$$

$$\bar{\mathbf{\Sigma}}_t = (\mathbf{I} - \mathbf{K}_t^i\mathbf{G}_t^i)\bar{\mathbf{\Sigma}}_t$$

end

$$\mu_t = \bar{\mu}_t$$

$$\mathbf{\Sigma}_t = \bar{\mathbf{\Sigma}}_t$$

$$\mathbf{K}_t = \bar{\mathbf{\Sigma}}_t\mathbf{G}_t^T(\mathbf{G}_t\bar{\mathbf{\Sigma}}_t\mathbf{G}_t^T + \mathbf{Q})^{-1}$$

Prediction step:

$$\bar{\mu}_t = \begin{bmatrix} f_r(\mu^r_{t-1}, \mathbf{u}_t) \\ \mu_{l1_{t-1}} \\ \vdots \\ \mu_{ln_{t-1}} \end{bmatrix}$$

$\mathbf{J}_{x_t}$ The Jacobian matrix of **f** w.r.t the state vector.

$\mathbf{J}_{u_t}$ The Jacobian matrix of **f** w.r.t the odometry.

$$\bar{\Sigma}_t = \mathbf{J}_{x_t} \mathbf{\Sigma}_{t-1} \mathbf{J}^T_{x_t} + \mathbf{J}_{u_t} \mathbf{R} \mathbf{J}^T_{u_t}$$

If at time step **t** we have mapped **n** landmarks, what is the dimension of these matrices?

The prediction step

The same matrices from the localization case

$$\mathbf{J}_{x_t} = \begin{bmatrix} \mathbf{J}_{x_t^r} & 0_{3\times 2n} \\ 0_{2n\times 3} & I_{2n\times 2n} \end{bmatrix} \qquad \mathbf{J}_{u_t} = \begin{bmatrix} \mathbf{J}_{u_t^r} \\ 0_{2n\times 2} \end{bmatrix}$$

$$\bar{\Sigma}_t = \mathbf{J}_{x_t}\mathbf{\Sigma}_{t-1}\mathbf{J}_{x_t}^T + \mathbf{J}_{u_t}\mathbf{R}\mathbf{J}_{u_t}^T$$

$$\bar{\Sigma}_t = \mathbf{J}_{x_t}\boldsymbol{\Sigma}_{t-1}\mathbf{J}_{x_t}^T + \mathbf{J}_{u_t}\mathbf{R}\mathbf{J}_{u_t}^T$$

$$\begin{bmatrix} \mathbf{J}_{x_t^r} & 0_{3\times 2n} \\ 0_{2n\times 3} & I_{2n\times 2n} \end{bmatrix} \begin{bmatrix} \Sigma_{\mathbf{x}^r} & \Sigma_{\mathbf{x}^r l_1} & \cdots & \Sigma_{\mathbf{x}^r l_n} \\ \Sigma_{l_1 \mathbf{x}^r} & \Sigma_{l_{11}} & \cdots & \Sigma_{l_{1n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{l_n \mathbf{x}^r} & \Sigma_{l_{n1}} & \cdots & \Sigma_{l_{nn}} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{x_t^r} & 0_{3\times 2n} \\ 0_{2n\times 3} & I_{2n\times 2n} \end{bmatrix}^T$$

We use the same treatment with this term as well.

The same set of equations

Prediction step:

$$\bar{\mu}_t = f(\mu_{t-1}, \mathbf{u}_t)$$
$$\bar{\Sigma}_t = \mathbf{J}_{x_t} \mathbf{\Sigma}_{t-1} \mathbf{J}_{x_t}^T + \mathbf{J}_{u_t} \mathbf{R} \mathbf{J}_{u_t}^T$$

Update step:

For each landmark $\mathbf{z}_t^i$ do:

$$\bar{\mu}_t = \bar{\mu}_t + \mathbf{K}_t^i (\mathbf{z}_t^i - h(\bar{\mu}_t, i))$$
$$\bar{\mathbf{\Sigma}}_t = (\mathbf{I} - \mathbf{K}_t^i \mathbf{G}_t^i) \bar{\mathbf{\Sigma}}_t$$

end

$$\mu_t = \bar{\mu}_t$$
$$\mathbf{\Sigma}_t = \bar{\mathbf{\Sigma}}_t$$

Similar to the mapping case but with the fact that the pose of the robot is now part of the state vector

$$\mathbf{K}_t = \bar{\mathbf{\Sigma}}_t \mathbf{G}_t^T (\mathbf{G}_t \bar{\mathbf{\Sigma}}_t \mathbf{G}_t^T + \mathbf{Q})^{-1}$$

The same measurement function we used for localization and for mapping.

$$h(\bar{\mu}_t, i) = \begin{bmatrix} r^i \\ \beta^i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_r - x_{l_i})^2 + (y_r - y_{l_i})^2} \\ atan2(y_{l_i} - y_r, x_{l_i} - x_r) - \theta_r \end{bmatrix}$$

Range and bearing with respect to robot's own frame of reference at time step t.

Coordinates of landmark i from the vector $\bar{\mu}_t$

The predicted pose of the robot from the vector $\bar{\mu}_t$

The Jacobian matrix of the measurement function

$$\mathbf{G}_t^i = \frac{\partial h(\bar{\mu}_t, i)}{\partial \bar{\mu}_t}$$

$$= \begin{bmatrix} -\frac{x_{l_i} - x_r}{r} & -\frac{y_{l_i} - y_r}{r} & 0 & \cdots & \frac{x_{l_i} - x_r}{r} & \frac{y_{l_i} - y_r}{r} & \cdots \\ \frac{y_{l_i} - y_r}{r^2} & -\frac{x_{l_i} - x_r}{r^2} & -1 & \cdots & -\frac{y_{l_i} - y_r}{r^2} & \frac{x_{l_i} - x_r}{r^2} & \cdots \end{bmatrix}$$

zeros

The same set of equations

Prediction step:

$$\bar{\mu}_t = f(\mu_{t-1}, \mathbf{u}_t)$$

$$\bar{\Sigma}_t = \mathbf{J}_{x_t} \mathbf{\Sigma}_{t-1} \mathbf{J}_{x_t}^T + \mathbf{J}_{u_t} \mathbf{R} \mathbf{J}_{u_t}^T$$

Update step:

For each landmark $\mathbf{z}_t^i$ do:

$$\bar{\mu}_t = \bar{\mu}_t + \mathbf{K}_t(\mathbf{z}_t^i - h(\bar{\mu}_t, i))$$

$$\bar{\mathbf{\Sigma}}_t = (\mathbf{I} - \mathbf{K}_t^i \mathbf{G}_t^i) \bar{\mathbf{\Sigma}}_t$$

end

$$\mu_t = \bar{\mu}_t$$

$$\mathbf{\Sigma}_t = \bar{\mathbf{\Sigma}}_t$$

But what if we see a landmark for the first time?

$$\mathbf{K}_t = \bar{\mathbf{\Sigma}}_t \mathbf{G}_t^T (\mathbf{G}_t \bar{\mathbf{\Sigma}}_t \mathbf{G}_t^T + \mathbf{Q})^{-1}$$

Landmark initialization

$$\bar{\mu_t}^* = \begin{bmatrix} \bar{\mu_t} \\ l_{new} \end{bmatrix} = \begin{bmatrix} \bar{\mu_t} \\ x_{l_{new}} \\ y_{l_{new}} \end{bmatrix}$$

You already know how to find these as we already encountered them in the mapping case during last lecture.

Simply expand the state vector with the coordinates of the new landmark in the map!

The landmark initialisation function

$$\mathbf{z} = \begin{bmatrix} r \\ \beta \end{bmatrix}$$

$$l_{new} = q(\mathbf{z}_t^{new}, \bar{\mu}_t)$$

Sensor measurement to a never seen before landmark.

$$l_{new} = \begin{bmatrix} x_r + r \times cos(\theta_r + \beta) \\ y_r + r \times sin(\theta_r + \beta) \end{bmatrix}$$

The first three components of $\bar{\mu}_t$

The Jacobian of the landmark initialisation function w.r.t the **z**

$$\mathbf{L}_z = \frac{\partial q(\mathbf{z}, \bar{\mu}_t)}{\partial \mathbf{z}}$$

$$= \begin{bmatrix} cos(\theta_r + \beta) & -r \times sin(\theta_r + \beta) \\ sin(\theta_r + \beta) & r \times cos(\theta_r + \beta) \end{bmatrix}$$

$$\mathbf{z} = \begin{bmatrix} r \\ \beta \end{bmatrix}$$

What about the covariance matrix?

$$\bar{\Sigma}_t^* = \begin{bmatrix} \bar{\Sigma}_t & 0 \\ 0 & \mathbf{L_z Q L_z^T} \end{bmatrix}$$

The covariance matrix expands as well!

Zero matrices!

The covariance of the sensor noise.

Putting it all together

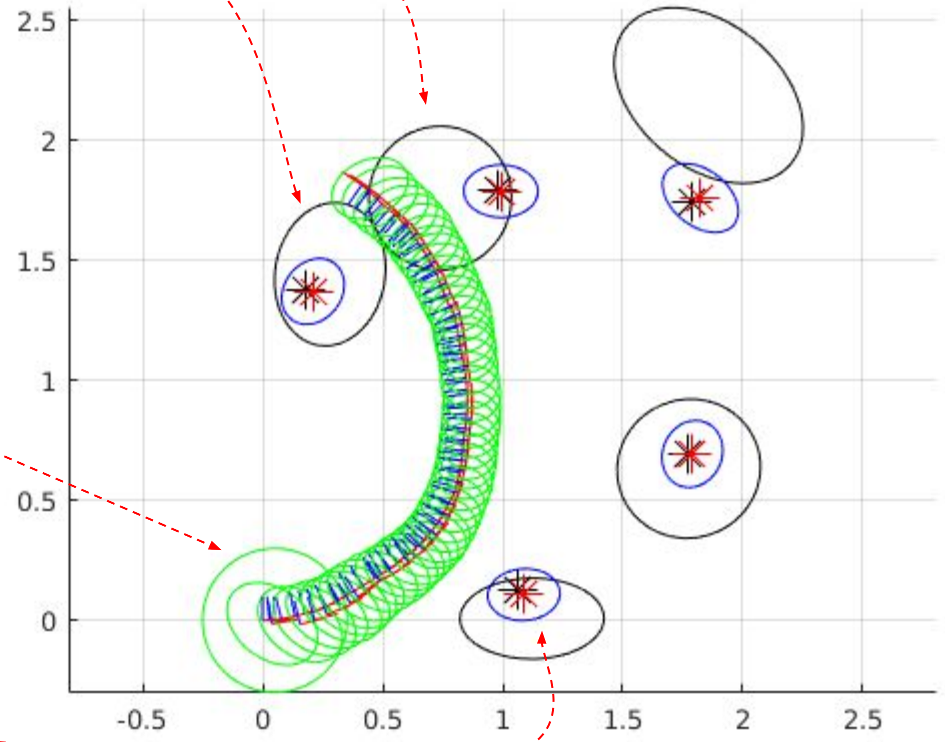
1. Move.
2. Perform the **prediction step** which updates the mean and covariance.
3. Make a new Measurement (range and bearing to a landmark).
4. if we have not seen the landmark before:
    - Do landmark **initialization** based on the robot estimated pose and expand the mean and the covariance.

    else

    - Perform the **update step** and update the mean and the covariance.
5. Go to 1.

The uncertainty on the positions of the landmarks on initialization.

The uncertainty on the pose of the robot (green ellipses)



The uncertainty on the position of the landmarks after 50 steps (**blue ellipses**)