EKF Localizer

One part of Project 2 involves using EKF for solving what we have partially solved (or tried to solve) in Project 1, “vehicle localization”.

Here we will discuss about how to define the prediction and update steps of our EKF based localizer. In addition, we discuss about how to initialize the estimation process.

But first we need to answer some question.

Why using EKF?

Answer: you already saw in Project 1, that we have multiple sources of independent information which we should combine for estimating the vehicle pose

* Kinematic model
* sensor's measurements (gyroscope, speed sensor, LiDAR)

How to "fuse" them?

Some average? How to weight them? Which one should be more relevant? (or more trustworthy?

We also saw that some sources of information can be “incomplete” at certain times (but still have information)

e.g., when we see just one OOI (what did you do in Project 1 in those situations?)

We saw that it is difficult to average “apples” and “pears”, (ranges and angles) How to scale/weight them?

How to consider the quality of the sources? Even with qualities which can change with the time or conditions of operation.

What happens when they are asynchronous? (available at different rates and times)

In Bayesian estimation

…→*Prediction* → *observation* → *Prediction* → *observation* →..

Which , in Gaussian terms means:

$$..\begin{Bmatrix}\hat{\mathbf{X}}(i\,|\,i-1)\\ \mathbf{P}(i\,|\,i-1)\end{Bmatrix}\xrightarrow[update]{\mathbf{y}(i)=h(\mathbf{x}(i))}\begin{Bmatrix}\hat{\mathbf{X}}(i\,|\,i)\\ \mathbf{P}(i\,|\,i)\end{Bmatrix}\xrightarrow[prediction]{\mathbf{x}(i+1)=F(\mathbf{x}(i),\mathbf{u}(i))}\begin{Bmatrix}\hat{\mathbf{X}}(i+1\,|\,i)\\ \mathbf{P}(i+1\,|\,i)\end{Bmatrix}\xrightarrow[update]{\mathbf{y}(i+1)=h(\mathbf{x}(i+1))}\begin{Bmatrix}\hat{\mathbf{X}}(i+1\,|\,i+1)\\ \mathbf{P}(i+1\,|\,i+1)\end{Bmatrix}\rightarrow..$$

(we maintain expected value and covariance matrix)

and for it we can use the EKF approach.

Initializing the estimation process

$$\hat{\mathbf{x}}(0|0) = ? \quad , \quad \mathbf{P}(0|0) = ?$$

If we were sure that we perfectly know the initial condition of the system, **X**(0)=**Xo**, then, in that case, we could propose

$$\hat{\mathbf{x}}(0|0) = \begin{bmatrix} x_0 \\ y_0 \\ \phi_0 \end{bmatrix}; \qquad \mathbf{P}(0|0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(covariance =0, means “NO uncertainty at all”)

Our expected value will be of size 3x1, and the associated covariance matrix 3x3.

Why?

Initializing those variables. $\hat{\mathbf{x}}(0|0) = ? \quad , \quad \mathbf{P}(0|0) = ?$

If we were sure that we perfectly know the initial condition of the system, **X**(0)=**Xo**, then, in that case, we could propose

$$\hat{\mathbf{x}}(0|0) = \begin{bmatrix} x_0 \\ y_0 \\ \phi_0 \end{bmatrix}; \qquad \mathbf{P}(0|0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(covariance =0, means “NO uncertainty at all”)

$$\hat{\mathbf{x}}(0|0) = ? \quad , \quad \mathbf{P}(0|0) = ?$$

"I was told that the initial condition is $(0, 0, \pi/2)$ and that we are sure about it."

$$\Rightarrow \quad \begin{cases} \hat{\mathbf{x}}(0|0) = \begin{bmatrix} 0 \\ 0 \\ \pi/2 \end{bmatrix}; & \mathbf{P}(0|0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{cases}$$

$$\hat{\mathbf{x}}(0|0) = ? \quad , \quad \mathbf{P}(0|0) = ?$$

It you were unsure about you assumed initial pose, knowing it should be "near" **X**(0)=**Xo**, e.g., about 1m wrong in *x*, 1m in y, and 10 degrees wrong in heading, then, in that case, we could propose

$$\hat{\mathbf{x}}(0|0) = \begin{bmatrix} x_0 \\ y_0 \\ \phi_0 \end{bmatrix}; \qquad \mathbf{P}(0|0) = \begin{bmatrix} 1^2 & 0 & 0 \\ 0 & 1^2 & 0 \\ 0 & 0 & (10 \cdot \pi / 180)^2 \end{bmatrix}$$

(I am using radians, in my implementation, for representing angles)

$$\hat{\mathbf{x}}(0|0) = ? \quad , \quad \mathbf{P}(0|0) = ?$$

We can also be conservative (i.e. a bit pessimistic)(it is good to play safe)

$$\hat{\mathbf{x}}(0|0) = \begin{bmatrix} x_0 \\ y_0 \\ \phi_0 \end{bmatrix}; \qquad \mathbf{P}(0|0) = 2 \cdot \begin{bmatrix} 1^2 & 0 & 0 \\ 0 & 1^2 & 0 \\ 0 & 0 & (10 \cdot \pi / 180)^2 \end{bmatrix}$$

Prediction step, in our problem.

In general, each time we perform a prediction step we perform the following calculations:

$$\mathbf{J} = \left.\frac{\partial \mathbf{f}\left(\mathbf{x},\mathbf{u}\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}\left(k|k\right),\mathbf{u}=\breve{\mathbf{u}}\left(k\right)}$$

$$\hat{\mathbf{x}}\left(k+1|k\right) = \mathbf{f}\left(\ \hat{\mathbf{x}}\left(k|k\right), \breve{\mathbf{u}}\left(k\right)\ \right)$$

$$\mathbf{P}\left(k+1|k\right) = \mathbf{J}\cdot\mathbf{P}\left(k|k\right)\cdot\mathbf{J}^{T} + \mathbf{Q}\left(k\right)$$

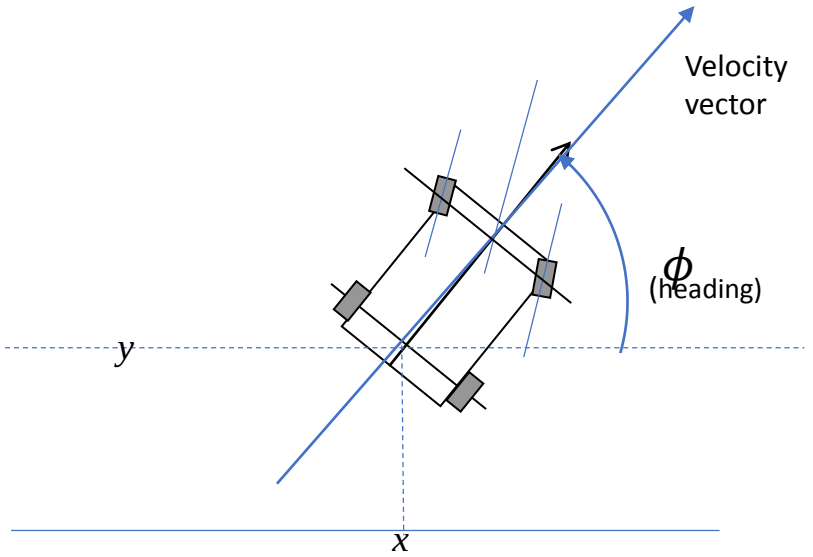
$$\mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right)$$

$$\Downarrow$$

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \phi(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \end{bmatrix} + \tau \cdot \begin{bmatrix} v(k) \cdot \cos\left(\phi(k)\right) \\ v(k) \cdot \sin\left(\phi(k)\right) \\ \omega(k) \end{bmatrix}$$

In our problem, the process model is:

$$\mathbf{x}(k) = \begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \end{bmatrix}, \quad \mathbf{u}(k) = \begin{bmatrix} v(k) \\ \omega(k) \end{bmatrix}$$



$$\mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right)$$

$$\Downarrow$$

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \phi(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \end{bmatrix} + \tau \cdot \begin{bmatrix} v(k) \cdot \cos\left(\phi(k)\right) \\ v(k) \cdot \sin\left(\phi(k)\right) \\ \omega(k) \end{bmatrix}$$

Based on our process model:

we obtain its Jacobian matrix:

$$\mathbf{f}\left(\mathbf{x}, \mathbf{u}\right) = \begin{bmatrix} x + \tau \cdot v \cdot \cos\left(\phi\right) \\ y + \tau \cdot v \cdot \sin\left(\phi\right) \\ \phi + \tau \cdot \omega \end{bmatrix}$$

$$\Downarrow$$

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \frac{\partial \mathbf{f}}{\partial [\mathrm{x}, \mathrm{y}, \phi]} = \left[\begin{array}{ccc} 1 & 0 & -\tau \cdot v \cdot \sin\left(\phi\right) \\ 0 & 1 & +\tau \cdot v \cdot \cos\left(\phi\right) \\ 0 & 0 & 1 \end{array}\right]$$

we need it, for the EKF prediction step).
Matrix **J** is evaluated at the expected values of **X** and **u**.

Matrix Q? In our case, one of the sources of uncertainty is due to noise polluting inputs (our knowledge about **u**(k) is usually not perfect) . We want to approximate its effect, in terms of additive noise to the process model.

$$\mathbf{X}(k+1) = f\left(\mathbf{X}(k), \mathbf{u}(k) + \delta_{\mathbf{u}}(k)\right); \quad \delta_{\mathbf{u}}(k) \text{ is WGN}, \quad \delta_{\mathbf{u}}(k) \sim N(0, \mathbf{P}_{\mathbf{u}})$$

$$\Downarrow$$

..

$$\mathbf{X}(k+1) = f\left(\mathbf{X}(k), \mathbf{u}(k)\right) + \zeta_{\mathbf{u}}(k); \quad \zeta_{\mathbf{u}}(k) \sim N(0, ?)$$

$$\zeta_{\mathbf{u}}(k) \cong \mathbf{J}_{\mathbf{u}} \cdot \delta_{\mathbf{u}}(k)$$

$$\Downarrow$$

$$\zeta_{\mathbf{u}}(k) \sim N(0, \mathbf{Q}_{\mathbf{u}}), \quad \mathbf{Q}_{\mathbf{u}} = \mathbf{J}_{\mathbf{u}} \cdot \mathbf{P}_{\mathbf{u}} \cdot \mathbf{J}_{\mathbf{u}}^{T}$$

$$\mathbf{J}_{\mathbf{u}} = ?$$

For that, we need to linearize the function at a convenient “point of operation”. As usual , we choose the current expected values of **X** and **u**.

Note that the inputs’ noise is assumed white Gaussian noise, of known covariance $\mathbf{P}_{\mathbf{u}}$, and of expected value =0**.**

$$\mathbf{x}(k+1) = \mathbf{f}\left(\ \mathbf{x}(k), \mathbf{u}(k) + \delta_{\mathbf{u}}(k)\ \right)$$

$$\delta_{\mathbf{u}}(k) \sim N(0, \mathbf{P}_{\mathbf{u}})$$

$$\Downarrow$$

$$\mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right) + \xi_{\mathbf{u}}(k)$$

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \phi(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \end{bmatrix} + \tau \cdot \begin{bmatrix} \left(v(k) + \delta_v(k)\right) \cdot \cos\left(\phi(k)\right) \\ \left(v(k) + \delta_v(k)\right) \cdot \sin\left(\phi(k)\right) \\ \omega(k) + \delta_\omega(k) \end{bmatrix}$$

$$\mathbf{J}_{\mathbf{u}} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\mathbf{x}=\hat{\mathbf{x}}, \mathbf{u}=\breve{\mathbf{u}}} = \left[\begin{array}{cc} \tau \cdot \cos(\phi) & 0 \\ \tau \cdot \sin(\phi) & 0 \\ 0 & \tau \end{array}\right]_{\phi=\hat{\phi}}$$

(As usual, when we linearize, we evaluate the Jacobian matrix at the expected values of the involved variables.)

The Jacobian matrix respect to the inputs:

$$\mathbf{f}\left(x, y, \phi, \mathrm{v}, \omega\right) = \begin{bmatrix} x + \tau \cdot v \cdot \cos\left(\phi\right) \\ y + \tau \cdot v \cdot \sin\left(\phi\right) \\ \phi + \tau \cdot \omega \end{bmatrix} \quad \Rightarrow \frac{\partial \mathbf{f}}{\partial u} = \frac{\partial \mathbf{f}}{\partial [v, \omega]} = \left[\begin{array}{c|c} \tau \cdot \cos\left(\phi\right) & 0 \\ \hline \tau \cdot \sin\left(\phi\right) & 0 \\ \hline 0 & \tau \end{array}\right]$$

$$\Rightarrow \quad \mathbf{J_u} = \left[\begin{array}{c|c} \tau \cdot \cos\left(\hat{\phi}\left(k \mid k\right)\right) & 0 \\ \hline \tau \cdot \sin\left(\hat{\phi}\left(k \mid k\right)\right) & 0 \\ \hline 0 & \tau \end{array}\right]$$

$$\Rightarrow \quad \mathbf{Q_u} = \mathbf{J_u} \cdot \mathbf{P_u} \cdot \mathbf{J_u}^T$$

How do we set $\mathbf{P_u}$?

Suppose we assume that the gyroscope's measurements have noise whose standard deviation is 0.5 degrees /second, and that the speed measurements are polluted by noise having standard deviation 1cm/second. We also know that both noises seem to be independent.

$$\mathbf{P_u} = \begin{bmatrix} 0.01^2 & 0 \\ 0 & (0.5 \cdot \pi / 180)^2 \end{bmatrix}$$

We use it ($\mathbf{P_u}$) and the proper Jacobian matrix to obtain the covariance , $\mathbf{Q_u}$ (Note that I scaled the standard deviation of the gyro's noise, because I am using radians/second in my model.)

We also remark that because $\mathbf{J_u}$ depends on the current expected value of the estimated heading, it usually varies at each prediction step. Consequently, $\mathbf{Q_u}$ does also change at different times, even if the covariance matrix $\mathbf{P_u}$ is usually constant,

$$\mathbf{Q_u} = \mathbf{J_u} \cdot \mathbf{P_u} \cdot \mathbf{J_u}^T$$

Our model has additional sources of error.

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \phi(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \end{bmatrix} + \tau \cdot \begin{bmatrix} \left(v(k)+\delta_v(k)\right)\cdot\cos\left(\phi(k)\right)+\delta_x(k) \\ \left(v(k)+\delta_v(k)\right)\cdot\sin\left(\phi(k)\right)+\delta_y(k) \\ \omega(k)+\delta_\omega(k) \end{bmatrix}$$

Those are due to skidding and other discrepancies between nominal model and the real platform. Those can usually be studied through experiments.
In our UGV, operating indoor, on a surface allowing good traction, we expect a discrepancy of less than 3 cm (in x and in y) every 1 second, when the machine is moving. This discrepancy is mostly due to instantaneous translation which is not in the longitudinal direction (e.g. due to skidding)
So, we assume a second additive noise $\xi_f(k)$

$$\mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)+\delta_{\mathbf{u}}(k)\right) + \xi_f(k)$$

$$\xi_f(k) \text{ is WGN, } \xi_f(k) \sim N(0, \mathbf{Q}_2)$$

(… our model has additional sources of error.)

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \phi(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \end{bmatrix} + T \cdot \begin{bmatrix} \left(v(k)+\delta_v(k)\right)\cdot\cos\left(\phi(k)\right)+\delta_x(k) \\ \left(v(k)+\delta_v(k)\right)\cdot\sin\left(\phi(k)\right)+\delta_y(k) \\ \omega(k)+\delta_\omega(k) \end{bmatrix}$$

If we assume that additional discrepancy of less than 3 cm (in x and in y) every 1 second, when the machine is moving.

$$\mathbf{X}(k+1) = f\left(\mathbf{X}(k), \mathbf{u}(k)+\delta_{\mathbf{u}}(k)\right)+\xi_f(k)$$

$$\xi_f(k) \text{ is WGN, } \xi_f(k) \sim N(0, \mathbf{Q}_2)$$

$$\mathbf{Q}_2 = \begin{bmatrix} (\tau\cdot 0.03)^2 & 0 & 0 \\ 0 & (\tau\cdot 0.03)^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Finally, the combined affect of both dominant sources of error in our process model, can be modelled by the covariance matrix Q

$$\mathbf{Q} = \mathbf{Q_u} + \mathbf{Q_2}$$

$$\mathbf{Q_u} = \mathbf{J_u} \cdot \mathbf{P_u} \cdot \mathbf{J_u}^T = \mathbf{J_u} \cdot \begin{bmatrix} 0.01^2 & 0 \\ 0 & (0.5 \cdot \pi / 180)^2 \end{bmatrix} \cdot \mathbf{J_u}^T$$

$$\mathbf{Q_2} = \begin{bmatrix} (T \cdot 0.03)^2 & 0 & 0 \\ 0 & (T \cdot 0.03)^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Update step, in our problem

EKF update step (general expression)

$$\mathbf{z}(k+1) = \mathbf{y}_{measurement}(k+1) - \mathbf{h}\left(\hat{\mathbf{x}}(k+1|k)\right)$$

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k+1|k)}$$

$$\mathbf{S} = \mathbf{H} \cdot \mathbf{P}(k+1|k) \cdot \mathbf{H}^T + \mathbf{R}(k+1)$$

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k) \cdot \mathbf{H}^T \cdot \mathbf{S}^{-1}$$

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1) \cdot \mathbf{z}(k+1)$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{P}(k+1|k) \cdot \mathbf{H}^T \cdot \mathbf{S}^{-1} \cdot \mathbf{H} \cdot \mathbf{P}(k+1|k)$$

$$\text{we have} \left\{ \begin{array}{c} \mathbf{R} = ? \\ \mathbf{H} = ? \\ \mathbf{y}_{measurement}(k+1) \end{array} \right\} \text{ and } \left\{ \begin{array}{c} \hat{\mathbf{x}}(k+1|k) \\ \mathbf{P}(k+1|k) \end{array} \right\}$$

$$\Downarrow$$

we perform these calculations:

$$\left\{ \begin{array}{c} \mathbf{z}(k+1) = \mathbf{y}_{measurement}(k+1) - h\left(\hat{\mathbf{x}}(k+1|k)\right) \\ \mathbf{S} = \mathbf{H} \cdot \mathbf{P}(k+1|k) \cdot \mathbf{H}^T + \mathbf{R} \\ \mathbf{K}(k+1) = \mathbf{P}(k+1|k) \cdot \mathbf{H}^T \cdot \mathbf{S}^{-1} \\ \hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1) \cdot \mathbf{z}(k+1) \\ \mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{P}(k+1|k) \cdot \mathbf{H}^T \cdot \mathbf{S}^{-1} \cdot \mathbf{H} \cdot \mathbf{P}(k+1|k) = \mathbf{P}(k+1|k) - \mathbf{K} \cdot \mathbf{S} \cdot \mathbf{K}^T \end{array} \right\} \Rightarrow \left\{ \begin{array}{c} \hat{\mathbf{x}}(k+1|k+1) \\ \mathbf{P}(k+1|k+1) \end{array} \right\}$$

we obtain:

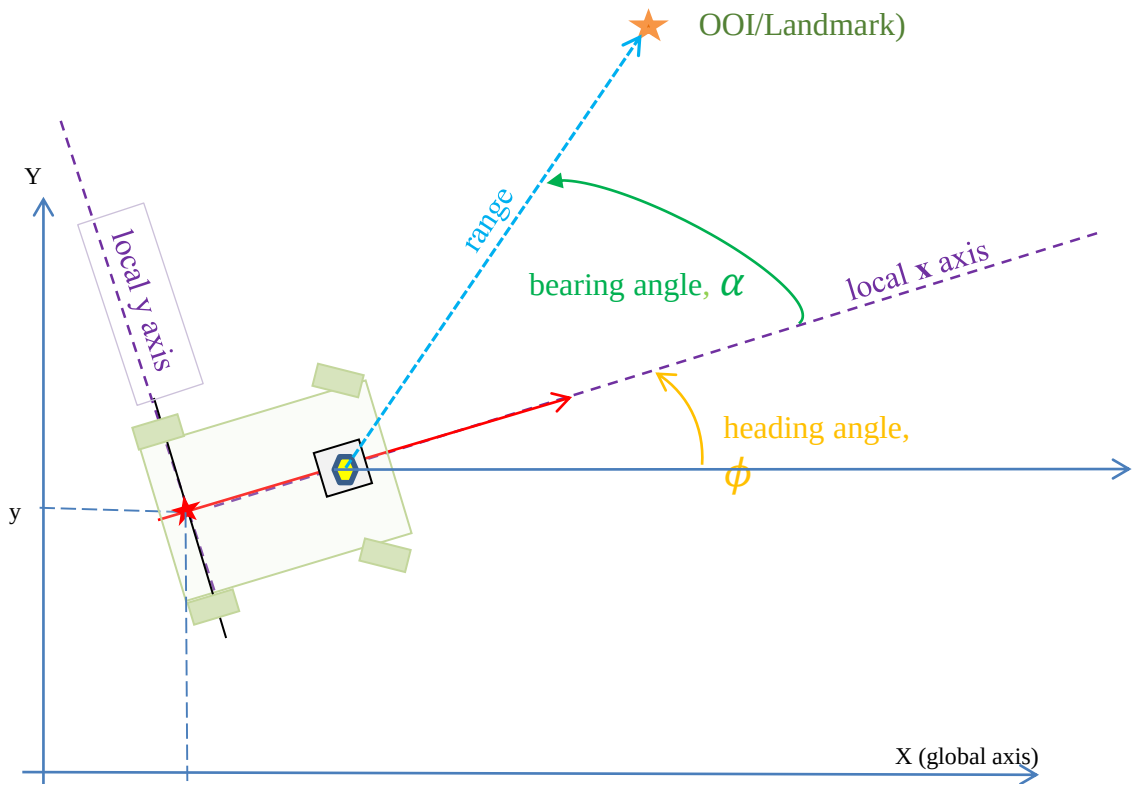
Let's see some source code, implementing it…

```
function [Xe,P]=DoUpdate(P,Xe,H,R,z)
        %here Xe and P are the PRIOR  expected value and covariance matrix
        S = R + H*P*H' ;
         iS=inv(S);   %in  a case in which S is 1x1 : inv(S) is just 1/S

        K = P*H'*iS ;          % Kalman gain
        % ----- finally, we do it...We obtain  X(k+1|k+1) and P(k+1|k+1)

        Xe = Xe+K*z ;       % update the  expected value
        P = P-K*H*P ;       % update the Covariance % i.e. "P = P-P*H'*iS*H*P"  )

        %now Xe and P are the POSTERIOR expected value and covariance
        % update done!
  return;
  end
```

Note1: we simply maintain two variables, **Xe** and **P**; which I use for storing the expected value, and the covariance.

Note2: we can refine the function to be a bit more efficient (using some auxiliary program variables to avoid repeating operations)

OBSERVATIONS (in our problem)



$$\begin{bmatrix} r \\ \alpha \end{bmatrix} = \begin{bmatrix} h_1(x, y, \phi) \\ h_2(x, y, \phi) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_a - x)^2 + (y_a - y)^2} \\ \text{atan2}(y_a - y, x_a - x) - \phi \end{bmatrix}$$

(Attention : this observation model assumes the LiDAR is not displaced on the car frame)

OBSERVATIONS (in our problem)

If we want to process, simultaneously, a range and bearing associated to a detected OOI

$$\begin{bmatrix} r \\ \alpha \end{bmatrix} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1(x, y, \phi) \\ h_2(x, y, \phi) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_a - x)^2 + (y_a - y)^2} \\ \text{atan2}(y_a - y, x_a - x) - \phi \end{bmatrix}$$

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} = \left[\begin{array}{ccc} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} & \frac{\partial h_1}{\partial \phi} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} & \frac{\partial h_2}{\partial \phi} \end{array}\right]_{\mathbf{x}=\hat{\mathbf{x}}} =$$

$$= \left[\begin{array}{ccc} -\frac{(x_a - x)}{\sqrt{(x_a - x)^2 + (y_a - y)^2}} & -\frac{(y_a - y)}{\sqrt{(x_a - x)^2 + (y_a - y)^2}} & 0 \\ \frac{(y_a - y)}{(x_a - x)^2 + (y_a - y)^2} & \frac{-(x_a - x)}{(x_a - x)^2 + (y_a - y)^2} & -1 \end{array}\right]_{\mathbf{x}=\hat{\mathbf{x}}}$$

R?

Suppose the standard deviation of noise in range measurements = 10cm, and that of the noise polluting the bearing measurements is 2 degrees.

$$\mathbf{R} = \begin{bmatrix} 0.1^2 & 0 \\ 0 & \left(2 \cdot \frac{\pi}{180}\right)^2 \end{bmatrix}_{(2\times 2)}$$

How do we process multiple observations in the same update stage? (because we have many OOIs)

→ As the sources of information are independent , we can process them separately, in a sequence of updates. Being the posterior of an individual update the prior of the following one.

The order in which we process them is relevant (each time we process one update, it is like processing a likelihood function in the Bayesian case)

We are ready for solving that part of Project 2
(almost, you do the rest…)

(we see some demo, in Matlab)

(We end this discussion here..)