One part of Project 3 involves using EKF for solving what you have solved  (or tried to solve) in Project 2, “platform localization”.

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

Remember: In Bayesian estimation we keep doing:

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

Which , in Gaussian terms means:

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

(we maintain expected value and covariance matrix)

Before you try  to implement the localizer using the real data (as you did in Project 2), you should try to implement and test matters  in  simulation.

You may use, the program "DemoEKF_2021b.m", to play with the example, and to modify it, to add certain required features.

After doing that, you can focus on implementing the localizer using real data, combining your project 2 solutions and parts of he provided  solution in the simulator.

## 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)=**X**$_0$, 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")

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

"The lecturer said the initial condition was $(0, 0, \pi/2)$ and that we were sure about it."

$$\Rightarrow \left\{ \quad \hat{\mathbf{X}}(0|0) = \begin{bmatrix} 0 \\ 0 \\ \pi/2 \end{bmatrix}; \qquad \mathbf{P}(0|0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right.$$

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

It you were unsure about you assumed initial pose, knowing it should be "near" **X**(0)=**X**₀, 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 could also be conservative (i.e. a bit pessimistic)

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

## Prediction step, in our problem.

**In general, e**ach time we perform a prediction step we perform the following calculations:

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

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

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

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

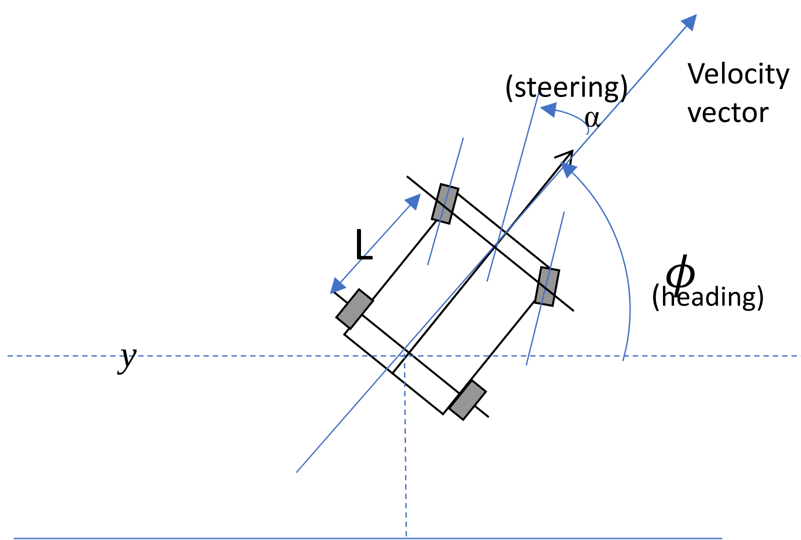
$$\Downarrow$$

In our problem, the process model is:

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

(discrete time version)

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



Based on our process model:

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

$$\Downarrow$$

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

we obtain its Jacobian matrix:

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

$$\Downarrow$$

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

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.

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

$$\Downarrow$$

..

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

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

$$\Downarrow$$

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

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

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.

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

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

$$\Downarrow$$

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

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

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

$$\mathbf{J}_{\mathbf{u}} = \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{\mathbf{X}=\hat{\mathbf{X}}, \mathbf{u}=\hat{\mathbf{u}}} = \left. \begin{bmatrix} T \cdot \cos(\phi) & 0 \\ T \cdot \sin(\phi) & 0 \\ 0 & T \end{bmatrix} \right|_{\phi=\hat{\phi}}$$

(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:

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

$$\Downarrow$$

$$\Rightarrow \quad \mathbf{J_u} = \begin{bmatrix} T \cdot \cos\left(\hat{\phi}\left(k \mid k\right)\right) & 0 \\ T \cdot \sin\left(\hat{\phi}\left(k \mid k\right)\right) & 0 \\ 0 & T \end{bmatrix}$$

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

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.

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

We use it ($\mathbf{P_u}$) and he proper Jacobian matrix to obtain the covariance ,$\mathbf{Q_u}$ , of the equivalent additive noise.

(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.

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

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 $\zeta_2(k)$

$$\mathbf{X}(k+1) = f\left(\mathbf{X}(k), \mathbf{u}(k)+\delta_{\mathbf{u}}(k)\right) + \zeta_2(k)$$

$$\zeta_2(k) \text{ is WGN, } \zeta_2(k) \sim N\left(0, \mathbf{Q}_2\right)$$

But, 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.

$$\mathbf{X}(k+1) = f\left(\mathbf{X}(k), \mathbf{u}(k)+\delta_{\mathbf{u}}(k)\right) + \zeta_2(k)$$

$$\zeta_2(k) \text{ is WGN, } \zeta_2(k) \sim N(0, \mathbf{Q}_2)$$

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

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

$$\mathbf{Q} = \mathbf{Q}_{\mathbf{u}} + \mathbf{Q}_2$$

$$\mathbf{Q}_{\mathbf{u}} = \mathbf{J}_{\mathbf{u}} \cdot \mathbf{P}_{\mathbf{u}} \cdot \mathbf{J}_{\mathbf{u}}^T = \mathbf{J}_{\mathbf{u}} \cdot \begin{bmatrix} 0.01^2 & 0 \\ 0 & (0.5 \cdot \pi / 180)^2 \end{bmatrix} \cdot \mathbf{J}_{\mathbf{u}}^T$$

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

# Update step, in our problem

## EKF update step (general expression)

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

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

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

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

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

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

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

$$\Downarrow$$

we perform these calculations:

$$\left\{\begin{matrix}\mathbf{z}(k+1)=\mathbf{y}_{measurement}(k+1)-h\left(\hat{\mathbf{X}}(k+1|k)\right)\\ \mathbf{S}=\mathbf{H}\cdot\mathbf{P}(k+1|k)\cdot\mathbf{H}^{T}+\mathbf{R}\\ \mathbf{K}(k+1)=\mathbf{P}(k+1|k)\cdot\mathbf{H}^{T}\cdot\mathbf{S}^{-1}\\ \hat{\mathbf{X}}(k+1|k+1)=\hat{\mathbf{X}}(k+1|k)+\mathbf{K}(k+1)\cdot\mathbf{z}(k+1)\\ \mathbf{P}(k+1|k+1)=\mathbf{P}(k+1|k)-\mathbf{P}(k+1|k)\cdot\mathbf{H}^{T}\cdot\mathbf{S}^{-1}\cdot\mathbf{H}\cdot\mathbf{P}(k+1|k)=\mathbf{P}(k+1|k)-\mathbf{K}\cdot\mathbf{S}\cdot\mathbf{K}^{T}\end{matrix}\right\}\Rightarrow\ \overset{\text{we obtain:}}{\left\{\begin{matrix}\hat{\mathbf{X}}(k+1|k+1)\\ \mathbf{P}(k+1|k+1)\end{matrix}\right\}}$$

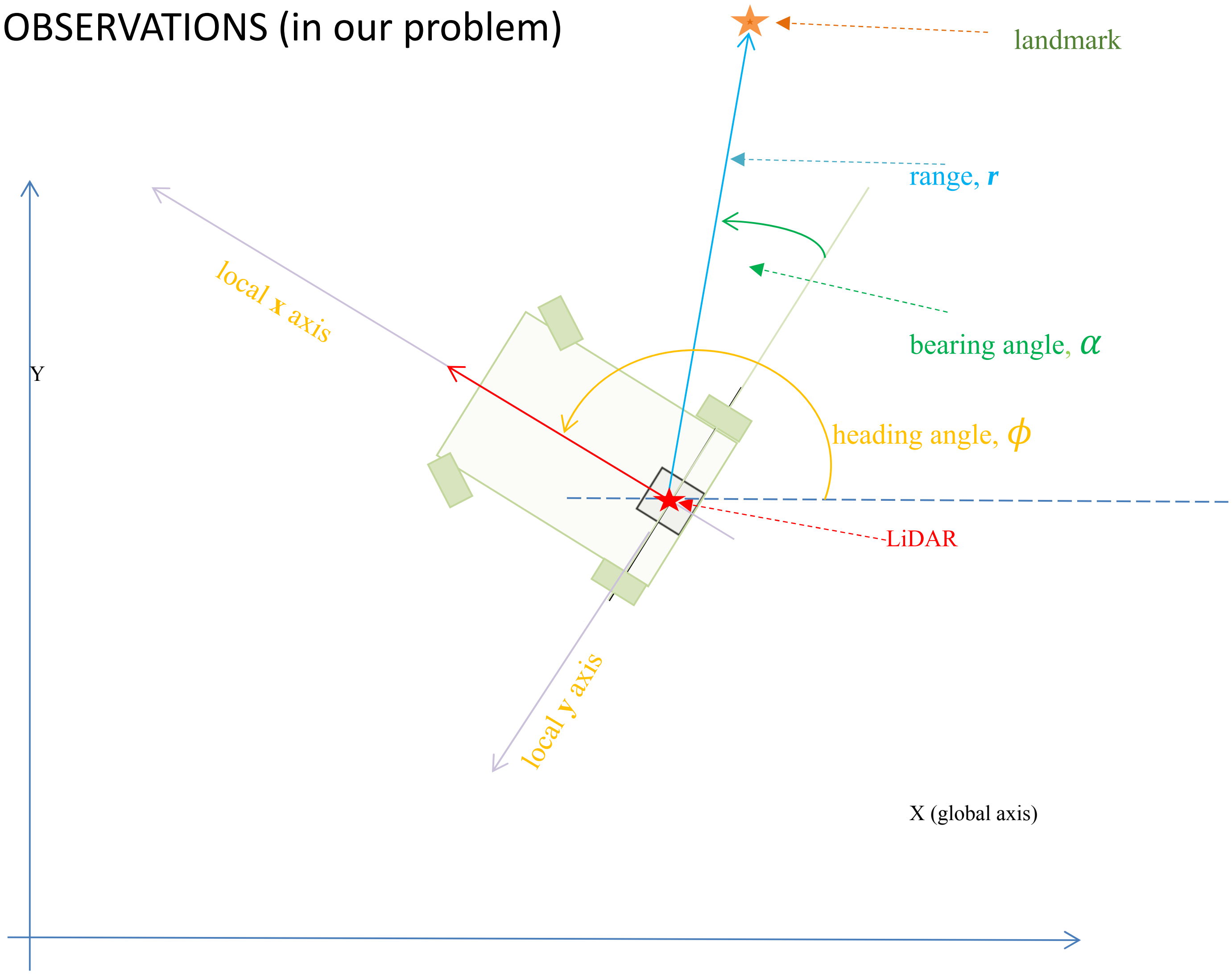
Let’s see some source code, implementing it…

```
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.

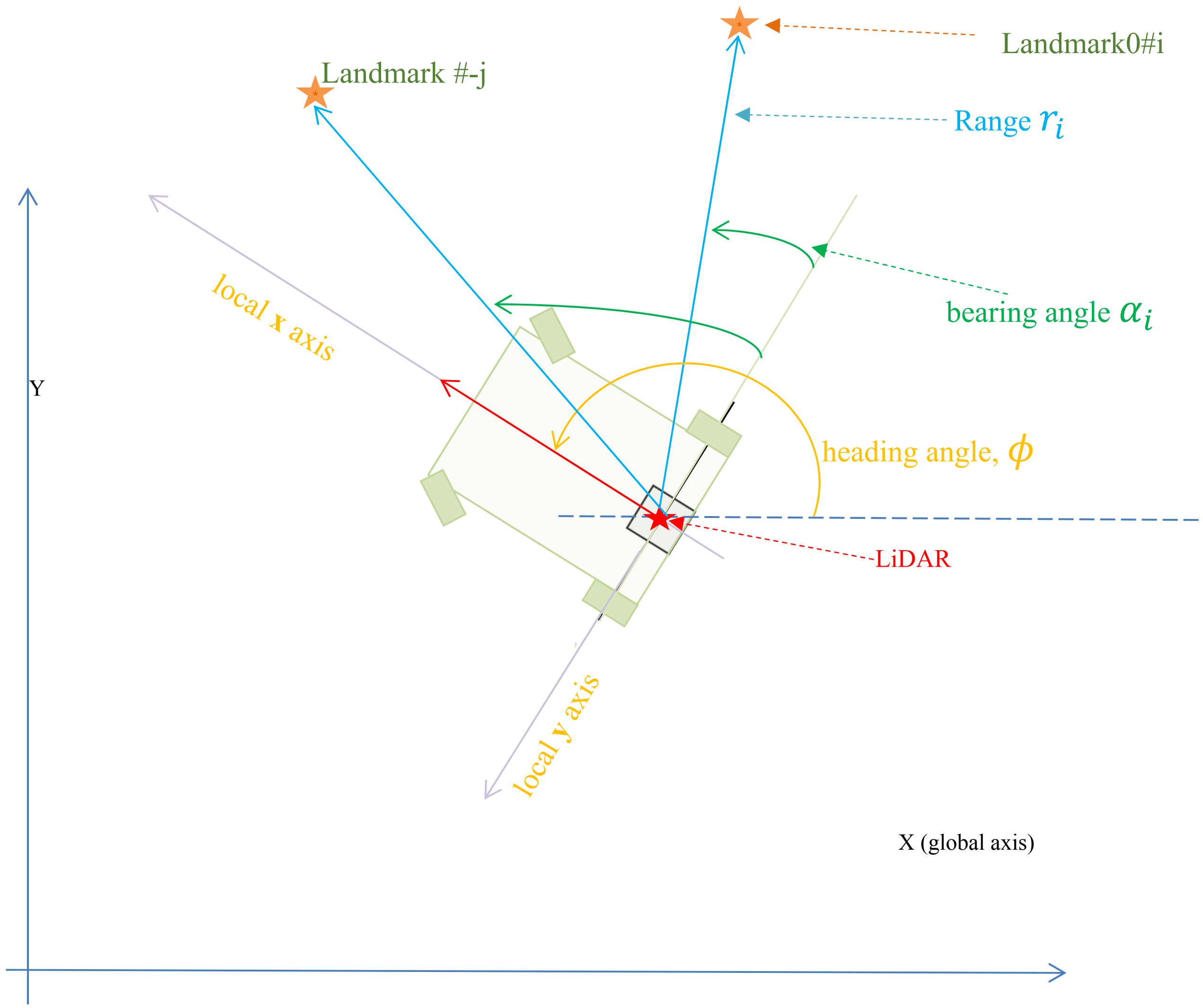
Note2: we can refine the function to be a bit more efficient (using some auxiliary program variables to avoid repeating operations)

# OBSERVATIONS (in our problem)



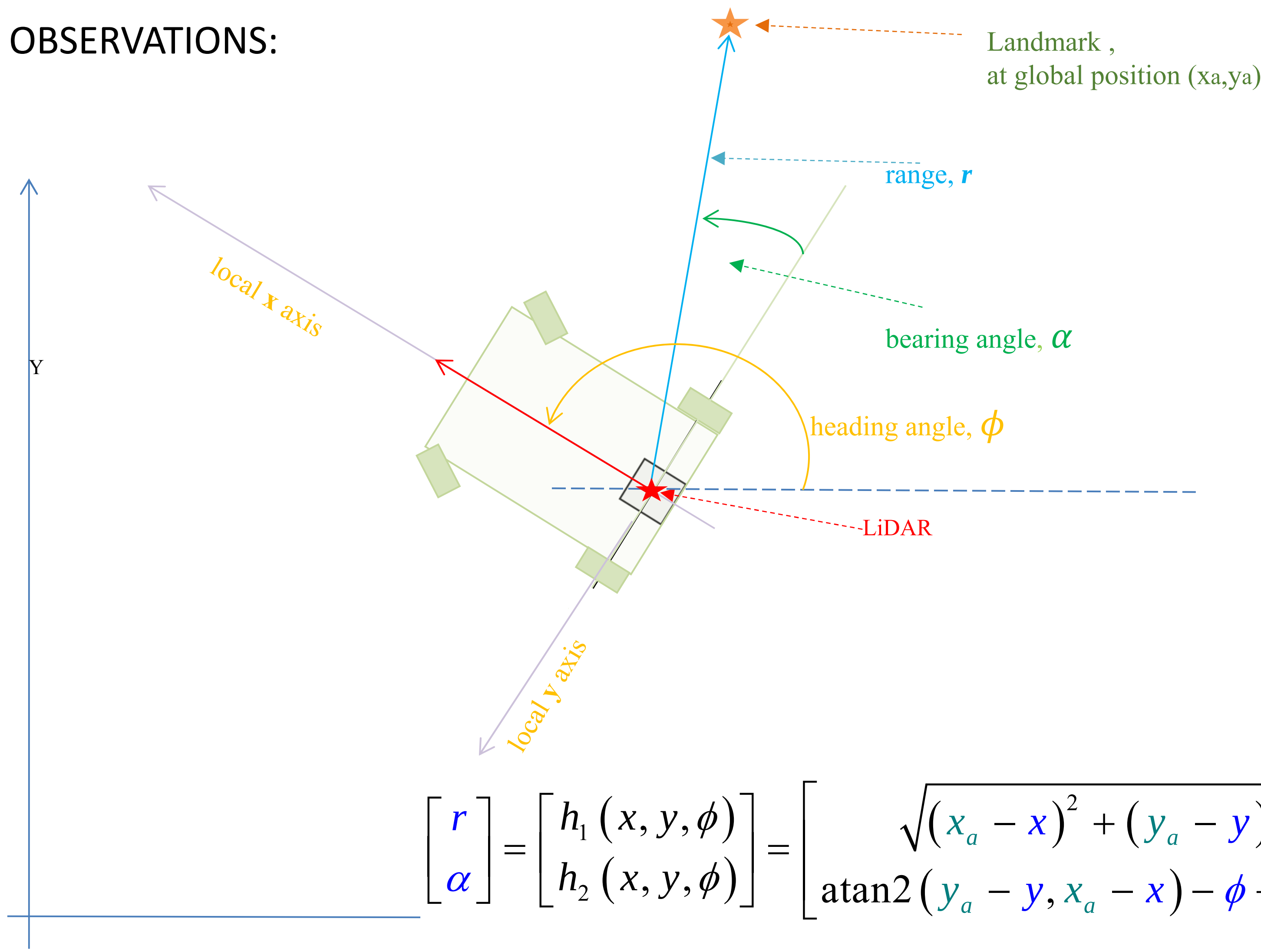

Note: In this example, the LiDAR is not located at the real position , as the one in our UGV.

# OBSERVATIONS:




If we have more landmarks → more observations → better.

## OBSERVATIONS:




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

Note: In this example, the LiDAR is not located at the real position

R?

I would recommend to assume standard deviation of noise in range measurements = 10cm, and for that in polluting the bearing =3 degrees.

Why?

We are ready for solving that part of Project 3
(almost, you do the rest…)

(We end  this discussion here..)