EKF (Extended Kalman Filter)

We have discussed about Bayesian estimation, conceptually

We have mentioned the KF (Kalman Filter) , for making Bayesian Estimation feasible in practical application, under certain conditions, which may be too restrictive for certain applications.

Now we will see how to extend capabilities of the KF, to deal with problems for which the standard KF is not adequate.

The name of that variant of the KF is called Extended KF (EKF)

EKF (Extended Kalman Filter)
Prediction step

The Prediction step is always necessary, as we permanently need estimates about the state of the system at time k (the present)

For doing it, we need:

1) The estimates we had at discrete time k-1
2) A process model ("nominal process model")
3) Statistical description of the errors incurred by the nominal process model.

The error of the process model is due to multiple reasons:

* The model does not consider certain aspects of the actual process (e.g. disturbances).
* The assumed inputs to the model may not be perfectly known at each time.

Still, even being imperfect, the nominal process model allows us to exploit, in a statistical fashion, our knowledge about the past state X(k-1) for helping to estimate the system state at time k, X(k).

The process model is a source of information (though not the only one) for continuously maintaining our knowledge about the state X at the present time.

Prediction step

Nominal process model

We say: “If we perfectly knew x(k) and u(k), then x(k+1) would be”

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

Here we assume we have a nominal process model, we know the inputs u(k), and that the error of the process model is zero mean WGN of known covariance matrix.

$$\boldsymbol{\xi}(k)\sim N\left(\mathbf{0},\mathbf{P}_{\xi(k)}\right)$$

This statistical model explains what happens in just one time step, from time k to time k+1.

It is telling us that even if we perfectly knew x(k) and u(k), our knowledge about x(k+1) would not be perfect, due to that instance of noise at time k.

It also tells us that we may be able to obtain a statistical description of x(k+1).

If our knowledge about x(k) was already expressed in statistical terms, via an expected value and a covariance matrix

$$\mathbf{x}(k) \sim N\left(\hat{\mathbf{x}}(k|k) \,,\, \mathbf{P}(k|k) \right)$$

$$\hat{\mathbf{x}}(k|k) \text{ : our expected value about } \mathbf{x}(k)$$

$$\mathbf{P}(k|k) \text{ : the associated covariance matrix } \left(\text{actually, about } \mathbf{x}(k) - \hat{\mathbf{x}}(k|k)\right)$$

Then we use it , with our process model, for generating a prediction about x(k+1) which will be also expressed through an expected value and an associated covariance matrix.

$$\mathbf{x}(k+1) \sim N\left(\hat{\mathbf{x}}(k+1|k) \,,\, \mathbf{P}(k+1|k) \right)$$

$$\hat{\mathbf{x}}(k+1|k) \text{ : our predicted expected value about } \mathbf{x}(k+1)$$

$$\mathbf{P}(k+1|k) \text{ : the associated covariance matrix}$$

Prediction step

We obtain

$$\hat{\mathbf{x}}(k+1|k) \quad \text{and} \quad \mathbf{P}(k+1|k)$$

Exploiting our knowledge of

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right) + \boldsymbol{\xi}(k) \\ \\ \mathbf{u}(k), \mathbf{P}_{\boldsymbol{\xi}(k)} \quad \text{and} \quad \left(\hat{\mathbf{x}}(k|k), \mathbf{P}(k|k)\right) \end{cases}$$

by performing the following calculations:

Prediction step

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

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

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

(jacobian matrix)

in which we refer to $\mathbf{P}_{\xi(k)}$ using the name $\mathbf{Q}(k)$

($\xi(k) \sim N(\mathbf{0}, \mathbf{Q}(k))$)

Remember:

The **Q** matrix (usual name in the related literature) is the covariance matrix of the additive noise which affects the process model, in just one time step.

It may be constant ,or it may be time varying.; it depends on the cases.

$$\mathbf{Q}(k)$$

$$\left(\ \boldsymbol{\xi}(k) \sim N\left(\mathbf{0},\mathbf{Q}(k)\right)\ \right)$$

Example: Pendulum

Our plant is a pendulum , whose assumed model (nominal process model) is described below.

$$\ddot{\varphi} = -10 \cdot \sin(\varphi) - 4 \cdot \dot{\varphi} + u(t)$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \varphi \\ \dot{\varphi} \end{bmatrix}; \qquad \frac{d\mathbf{x}}{dt} = \begin{bmatrix} x_2 \\ -10 \cdot \sin(x_1) - 4 \cdot x_2 + u(t) \end{bmatrix}$$

(continuous time model)

$$\Downarrow$$

discrete time version, for $\tau = 10ms = 0.01s$, including error $\boldsymbol{\xi}(k)$

$$\mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right) + \boldsymbol{\xi}(k) = \begin{bmatrix} x_1 + \tau \cdot x_2 \\ x_2 + \tau \cdot \left(-10 \cdot \sin(x_1) - 4 \cdot x_2 + u\right) \end{bmatrix}_{(k)} + \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}_{(k)}$$

Suppose we know , based on lab tests, that the error introduced by our model respect to the real plant dynamics behaves as WGN, and that it is mostly due to inaccurate knowledge of parameters of the original ODE, and also to certain nonlinearities in that ODE, which we are not able to model. There is also some minor effect due to the approximation applied to obtain a discrete time model

$$\begin{bmatrix} \xi_1(k) \\ \xi_2(k) \end{bmatrix} \sim N\left(\mathbf{0}_{2\times1} \; , \; \begin{bmatrix} 0.0005^2 & 0 \\ 0 & 0.004^2 \end{bmatrix} \right)$$

Example: Pendulum continuation

The diagonal elements of **Q** seem the be too small, but we should keep in mind that those describe noises which are added at each time step, which does occur every 0.01 second (10ms), and are expressed in radians and radians/second.

(so, maybe they are not so small as they appear to be)

$$\begin{bmatrix} \xi_1(k) \\ \xi_2(k) \end{bmatrix} \sim N\left(\mathbf{0}_{2\times 1} \quad , \quad \begin{bmatrix} 0.0005^2 & 0 \\ 0 & 0.004^2 \end{bmatrix} \right)$$

x_2 is in radians/second

x_1 is in radians

$\xi_1(k)$ has standard deviation 0.0005 rad

$\xi_2(k)$ has standard deviation 0.004 rad/second

However, they add that uncertainty every 10ms, always.

How large could be the cumulated error if we keep adding that error?

(brief calculation just by adding marginal variances)

(we do it in class)

Example: Pendulum continuation

We obtain the Jacobian **J** matrix, analytically , once , to be used as many times as needed later.

$$\mathbf{f}(\mathbf{x},\mathbf{u}) = \begin{bmatrix} x_1 + \tau \cdot x_2 \\ x_2 + \tau \cdot \left(-10 \cdot \sin(x_1) - 4 \cdot x_2 + u\right) \end{bmatrix}$$

$$\Downarrow$$

$$\left[\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right] = \begin{bmatrix} 1 & \tau \\ -\tau \cdot 10 \cdot \cos(x_1) & 1 - 4 \cdot \tau \end{bmatrix}$$

Now we see some program, using these necessary items.

In the example, we simply apply a sequence of predictions, although in real applications it is usually combined with updates.

```
function Example(X0, u0)

stds = [0.0005,0.004];  % standard deviations of noise components.

Q = diag(stds.^2);      %  A diagonal Q,  because, in this case, noises  E1,E2 are independent
Tau=0.01;
params = [-10,-0.04,1];  %model's coefficients a,b,c

% initial expected value,  and initial covariance matrix
Xe=X0;
P=zeros(2,2);

% For the sake of simplicity, in this example, I assume we keep applying u(k)=u0.
 u=u0;

 N=500;
for k=1:N,
    % get predicted covariance matrix and expected value
    [Xe,P]=DoPrediction(Xe,P, u, params,Tau,Q);

end;

end
```

```
function [X,P]=DoPrediction(X,P, u, params,Tau,Q)
   % need Jacobian to be evaluated at each discrete time k (because model is non-linear)
    J= MyJacobian(X,u,Tau,params);  % (see code inside this function)
    P = J*P*J'+Q;        % covariance of predicted x(k)
    X=modelPendulum(X,u,Tau,params);  % get expected value
end
```

```
function J=MyJacobian(X,u,T,params)
  a=params(1);b=params(2);
  J = [ [ 1,T];[T*a*cos(X(1)),1+b*T]];
end

% implement discrete time model of plant.
function Xnew=modelPendulum(X,u,Tau,params)
  a=params(1);b=params(2);c=params(3);
  Xnew = [   X(1)+Tau*X(2)  ;    X(2)  + Tau*(    a*sin(X(1))+b*X(2)+c*u)];
end
```

$$\mathbf{f}(\mathbf{x},\mathbf{u}) = \begin{bmatrix} x_1 + \tau \cdot x_2 \\ x_2 + \tau \cdot \left(-10 \cdot \sin(x_1) - 4 \cdot x_2 + u\right) \end{bmatrix}$$

$$\left[\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right] = \begin{bmatrix} 1 & \tau \\ -\tau \cdot 10 \cdot \cos(x_1) & 1 - 4 \cdot \tau \end{bmatrix}$$

How to infer Q?

The error in the process model is due to multiple factors. We can separate those sources of uncertainty to infer Q.

Model Error due to "noise" in inputs.

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

One relevant source of error may be our inaccurate knowledge about the values of the model's inputs, **u**(k).

$$\mathbf{u}(k) = \breve{\mathbf{u}}(k) + \delta\mathbf{u}(k)$$

$\breve{\mathbf{u}}(k)$: known (assumed or measured) input value

$\delta\mathbf{u}(k)$: discrepancy between actual $\mathbf{u}(k)$ and known one, $\breve{\mathbf{u}}(k)$

$\delta\mathbf{u}(k)$ assumed zero mean WGN with covariance $\mathbf{P}_{\mathbf{u}}$

Model Error due to “noise” in inputs.

One relevant source of error may be our inaccurate knowledge about the values of the model inputs, $\mathbf{u}(k)$.

Does it happen in real life?

Case: 1 We may decide the control actions, but the actuators may introduce some unknown distortion and noise.

Case: 2 We know the inputs being applied by measuring them through sensors (which introduce measurement errors)

In our case of study (being used in Project 1) , we use a kinematic model whose inputs are longitudinal velocity and heading rate, which we know from sensors’ measurements.

How that "noise" , in the inputs, does affect our prediction model?

$$\mathbf{x}(k+1)=\mathbf{f}(\mathbf{x}(k),\mathbf{u}(k)) = \mathbf{f}(\mathbf{x}(k),\breve{\mathbf{u}}(k)+\delta\mathbf{u}(k))$$

$$\cong \mathbf{f}(\mathbf{x}(k),\breve{\mathbf{u}}(k))+\mathbf{J}_{\mathbf{u}}\cdot\delta\mathbf{u}(k)=$$

$$=\mathbf{f}(\mathbf{x}(k),\breve{\mathbf{u}}(k))+\xi_{\mathbf{u}}(k)$$

$\xi_{\mathbf{u}}(k)$: approx. resulting error in proccess model due to effect of noise $\delta\mathbf{u}(k)$

$$\xi_{\mathbf{u}}(k)=\mathbf{J}_{\mathbf{u}}\cdot\delta\mathbf{u}(k)$$

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

•

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

$\xi_{\mathbf{u}}(k)$: resulting error in proccess model due to noise $\delta\mathbf{u}(k)$

$$\delta\mathbf{u}(k) \sim N\left(\mathbf{0}_{(m\times1)}, \mathbf{P}_{\mathbf{u}}\right)$$

$$\Downarrow$$

$$\hat{\xi}_{\mathbf{u}}(k) = \mathbf{J}_{\mathbf{u}} \cdot \hat{\delta}_{\mathbf{u}}(k) \sim N\left(\mathbf{0}_{(n\times1)}, \mathbf{Q}_{\mathbf{u}}\right)$$

$$\mathbf{Q}_{\mathbf{u}} = \mathbf{J}_{\mathbf{u}} \cdot \mathbf{P}_{\mathbf{u}} \cdot \mathbf{J}_{\mathbf{u}}^{T}$$

Resulting prediction step:

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

$$\mathbf{P}(k+1 \mid k) = \mathbf{J} \cdot \mathbf{P}(k \mid k) \cdot \mathbf{J}^T + \mathbf{Q}_{\mathbf{u}}$$

$$\mathbf{Q}_{\mathbf{u}} = \mathbf{J}_{\mathbf{u}} \cdot \mathbf{P}_{\mathbf{u}} \cdot \mathbf{J}_{\mathbf{u}}^T$$

in which

$\breve{\mathbf{u}}(k)$: measured/known input value

$\mathbf{P}_{\mathbf{u}}$: covariance matrix of noise which affects inputs

$$\mathbf{J} = \left[\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}\right]_{\substack{\mathbf{x} = \hat{\mathbf{x}}(k \mid k) \\ \mathbf{u} = \breve{\mathbf{u}}(k)}}$$

$$\mathbf{J}_{\mathbf{u}} = \left[\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\right]_{\substack{\mathbf{x} = \hat{\mathbf{x}}(k \mid k) \\ \mathbf{u} = \breve{\mathbf{u}}(k)}}$$

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

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

$$\mathbf{Q}_{\mathbf{u}} = \mathbf{J}_{\mathbf{u}} \cdot \mathbf{P}_{\mathbf{u}} \cdot \mathbf{J}_{\mathbf{u}}^T$$

This is relevant to us, because we use a kinematic model which has inputs whose values we measure via noisy sensors. E.g. ,the gyroscope.

What happens if we also know the process model is inaccurate by itself?

We combine the sources of error

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

$$\mathbf{P}(k+1|k) = \mathbf{J} \cdot \mathbf{P}(k|k) \cdot \mathbf{J}^T + \mathbf{Q}_{\mathbf{u}} + \mathbf{Q}_f$$

$\mathbf{Q}_f$: covariance matrix of process model error (not accounting for noise affecting $\mathbf{u}(k)$)

We know how to apply a prediction step for the cases of nonlinear process model.

We need to solve the update step, too, for cases in which the observation ,model is nonlinear.

KF Update (discussed in week 5)

We have $\left\{\hat{\mathbf{x}}(k|k-1), \mathbf{P}(k|k-1)\right\}$ PRIOR $\{$expected value and , covariance matrix$\}$

We want $\left\{\hat{\mathbf{x}}(k|k), \mathbf{P}(k|k)\right\}$ POSTERIOR $\{$expected value and , covariance matrix$\}$

$$\Downarrow$$

$$\mathbf{z}(k) = \mathbf{y}_{measurement}(k) - \mathbf{H} \cdot \hat{\mathbf{x}}(k|k-1)$$

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

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

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

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

Relevant "actors" in the Update step

$\{\hat{\mathbf{x}}(k|k-1), \mathbf{P}(k|k-1)\}$ {expected value and , covariance matrix of PRIOR}

$$\begin{cases} \hat{\mathbf{x}}(k|k-1) & \text{: expected value of } \mathbf{x}(k) \text{ before update} \\ \mathbf{P}(k|k-1) & \text{: covariance matrix before update} \end{cases}$$

$\mathbf{H}\cdot\hat{\mathbf{x}}(k|k-1)$ expected measurement of output variable

(the value of the output variable, $\mathbf{y}(k)$, if $\mathbf{x}(k)$ was $= \hat{\mathbf{x}}(k|k-1)$, based on our assumed output model $\mathbf{y}(k) = \mathbf{H}\cdot\mathbf{x}(k)$)

$\mathbf{y}_{measurement}(k)$: actual measurement of output variable

(affected by sensor noise and other uncertainties)

$\mathbf{R}(k)$: covariance matrix

of the uncertainty that pollutes the measurement of the output variable

---

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

$\{\hat{\mathbf{x}}(k|k), \mathbf{P}(k|k)\}$ {expected value and , covariance matrix of POSTERIOR}

$$\begin{cases} \hat{\mathbf{x}}(k|k) & \text{: expected value of } \mathbf{x}(k) \text{ as result of the update} \\ \mathbf{P}(k|k) & \text{: covariance matrix as result of the update} \end{cases}$$

Update for nonlinear output equation

Now we consider the case:

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k))$$

$$\mathbf{y}_{measurement}(k) = \mathbf{h}(\mathbf{x}(k)) + \boldsymbol{\eta}(k)$$

$\mathbf{y}_{measurement}(k)$: measurement of output variable $\mathbf{y}$ at time k.

$\boldsymbol{\eta}(k)$ error, due to sensor noise and to inaccuracies in output model

$\boldsymbol{\eta}(k)$ assumed WGN, of known variance

$$\boldsymbol{\eta}(k) \sim N(\mathbf{0}, \mathbf{R}(k))$$

Update for nonlinear output equation

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

$$\mathbf{H} = \left[\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right]\Bigg|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)}$$

$$\left.\begin{cases} \mathbf{S} = \mathbf{H}\cdot\mathbf{P}(k\mid k-1)\cdot\mathbf{H}^T + \mathbf{R}(k) \\ \mathbf{K}(k) = \mathbf{P}(k\mid k-1)\cdot\mathbf{H}^T\cdot\mathbf{S}^{-1} \\ \hat{\mathbf{x}}(k\mid k) = \hat{\mathbf{x}}(k\mid k-1) + \mathbf{K}(k)\cdot\mathbf{z}(k) \\ \mathbf{P}(k\mid k) = \mathbf{P}(k\mid k-1) - \mathbf{P}(k\mid k-1)\cdot\mathbf{H}^T\cdot\mathbf{S}^{-1}\cdot\mathbf{H}\cdot\mathbf{P}(k\mid k-1) \end{cases}\right\}\text{(as for the standard KF)}$$

(similar to the linear case, except in how we calculate the expected measurement and the **H** matrix.)

Example.

If we had

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = (3 \cdot x_1 + 5 \cdot x_2)^2$$

$$\hat{\mathbf{x}}(k \mid k-1) = [1.15 \quad 2.25]^T$$

then the **H** matrix and the expected value of the output variable would be:

$$\mathbf{H} = \left[\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right] \Bigg|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)} = \left[2 \cdot (3 \cdot x_1 + 5 \cdot x_2) \cdot 3 \quad 2 \cdot (3 \cdot x_1 + 5 \cdot x_2) \cdot 5 \right]_{\substack{x_1=1.15 \\ x_2=2.25}} = [\triangle \qquad \square]$$

$$\hat{\mathbf{y}}(k) = \mathbf{h}(\mathbf{x})\big|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)} = (3 \cdot 1.15 + 5 \cdot 2.25)^2 = ..$$

(note that, in this case, matrix **H** is 1 $\times$ 2 as function **h** is $\mathbb{R}^2 \to \mathbb{R}^1$
i.e. $\mathbf{x} \in \mathbb{R}^2, \mathbf{y} \in \mathbb{R}^1$)

Example.

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = (3 \cdot x_1 + 5 \cdot x_2)^2$$

$$\hat{\mathbf{x}}(k \mid k-1) = \begin{bmatrix} 1.15 & 2.25 \end{bmatrix}^T$$

$$\mathbf{y}_{measured}(k) = 122.5$$

standard deviation of measurement noise =3.33

$$\Downarrow$$

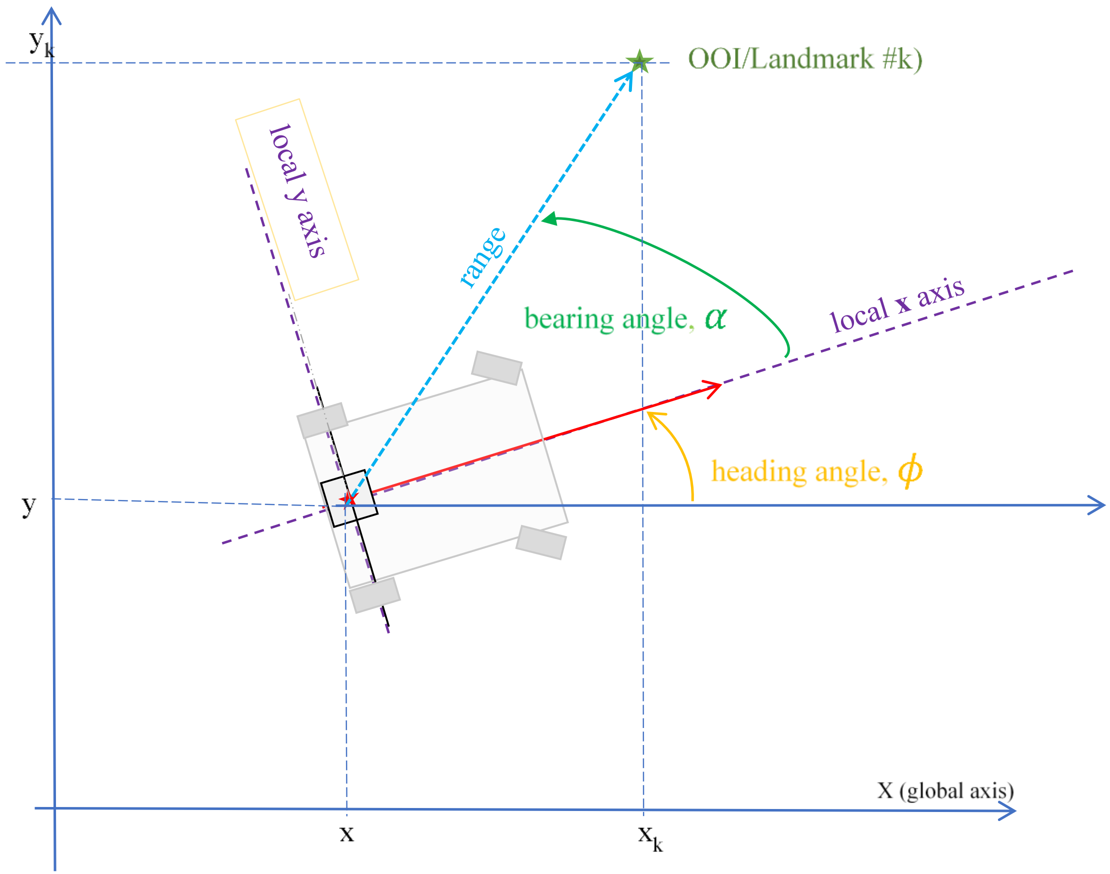
variance $\mathbf{R} = (3.33)^2$

$$\mathbf{H} = \left[\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right] \Bigg|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)} = \begin{bmatrix} 2 \cdot (3 \cdot x_1 + 5 \cdot x_2) \cdot 3 & 2 \cdot (3 \cdot x_1 + 5 \cdot x_2) \cdot 5 \end{bmatrix}_{\substack{x_1=1.15 \\ x_2=2.25}} = \begin{bmatrix} \triangle & \square \end{bmatrix}$$

$$\hat{\mathbf{y}}(k) = \mathbf{h}(\mathbf{x})\big|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)} = (3 \cdot 1.15 + 5 \cdot 2.25)^2 = \square$$ (expected measurement)

$$z(k) = 122.5 - \square$$ (z=actual measurement - expected measurement)

Example: Simplified case. LiDAR installed at origin of car's coordinate frame.



function relationship between output variables and state variables:

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

(r, α): range and bearing to detected OOI (measured/calculated locally)

(x_k, y_k): position of OOI's associated landmark, in GCF (known parameters, from given map of landmarks)

Using EKF for localizing our platform.

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

(r, α): range and bearing to detected OOI (measured/calculated locally)

(x_k, y_k): position of landmark, in GCF, associated to the locally detected OOI

(x_k, y_k) (are known parameters, from a-priori given map of landmarks)

Case:, in numbers

We detect an OOI, which in the car's CF, has position (10,10) (expressed in metres)

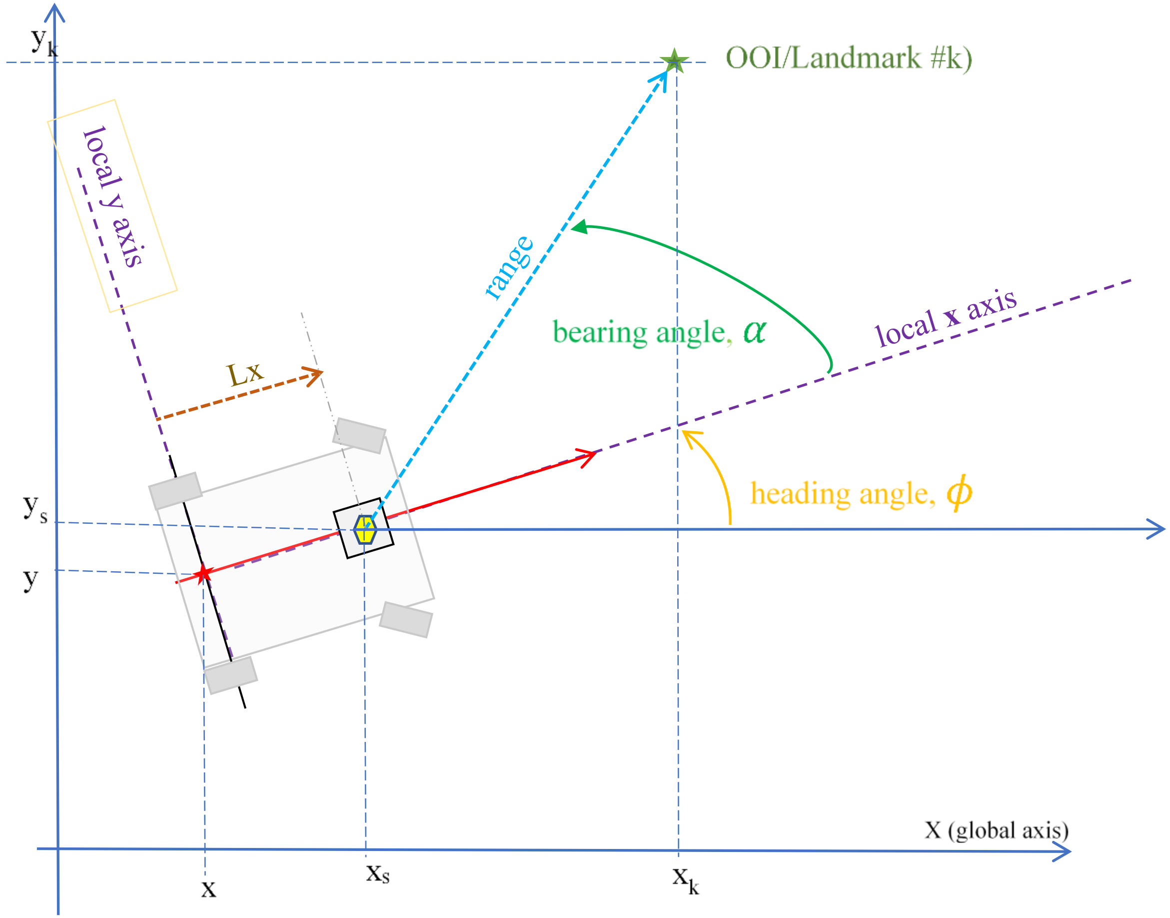
We, based on a Data Association module, know that that OOI is a landmark whose position (in GCF) is (122,60) , expressed in metres.

$$\mathbf{y}_{measured} = \begin{bmatrix} 14.2 \\ 45^{o} \end{bmatrix} =$$

observation model:

$$\mathbf{y} = \begin{bmatrix} r \\ \alpha \end{bmatrix} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1(x, y, \phi) \\ h_2(x, y, \phi) \end{bmatrix} = \begin{bmatrix} \sqrt{(122 - x)^2 + (60 - y)^2} \\ \mathrm{atan2}(\ 60 - y\ , 122 - x\ ) - \phi \end{bmatrix}$$

LiDAR displaced

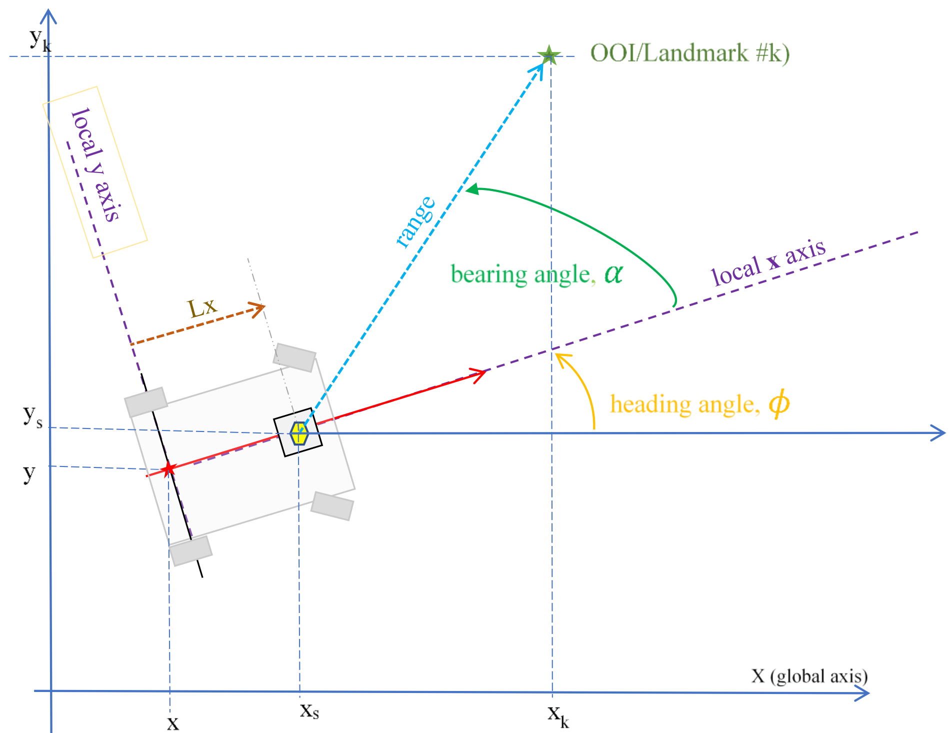


function relationship between output variables and state variables:

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

$$(x_s, y_s) = (x, y) + L_x \cdot (\cos(\phi), \sin(\phi))$$

LiDAR displaced



Alternative way (trick) : express OOI's position in vehicle's CF (transform from LiDAR's CF to vehicle's CF), and obtain polar coordinates there (range and bearing), then use the simplified observation function.

Observation model

Uncertainty affecting measurements

From LiDAR sensor

Quantization error (1/2 cm)

Angular resolution (1/2 degree)

error due to LiDAR inclination (due to Pitch and roll), which are assumed to be =0, i.e. perfectly horizontal.

But platform may oscillate due to terrain not being perfectly flat/horizontal and its interaction with platform suspension system.

Vibrations of vehicle chassis may also affect.

Poles not being perfectly vertical.

CoG of poles may also introduce some error.

In our case, considering all those sources of error, we estimated that the resulting error can be assumed to be WGN having a standard deviation in 10cm for range measurements, and 2 degrees for bearing measurements.

(slides end here)