Can we simultaneously estimate the state and certain model parameters?

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

$$\mathbf{y} = \mathbf{h}\left(\mathbf{x}(k), \boldsymbol{\beta}\right)$$

$$\begin{bmatrix} \mathbf{x}(k) \\ \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = ?$$

Particular Case

Can we simultaneously estimate platform's pose and the gyroscope's bias?

In our project, we implemented an EKF for estimating the system's states, for which we considered the following process model:

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

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

One of the model inputs is the **heading rate**, which we know through gyroscope measurements, whose measurements are noisy,
being the noise described by two components: a “bias” and a WGN component

$$\omega_{measured}\left(k\right)=\omega_{real}\left(k\right)+b+\mu\left(k\right)$$

$\mu\left(k\right)$: WGN of known variance.

b : unknown constant (or slowly time varying)

If we knew the bias value, **b**, we would remove it from the measurements by simply subtracting its value from the measured value

Resulting in a measurement polluted by WGN

$$\omega_{improved}(k) = \omega_{measured}(k) - b = \omega_{real}(k) + \mu(k)$$

$$\mu(k)\text{: WGN}$$

Our pose's estimation approach is able to deal with noisy inputs, if the involved noises were approximately WGN.

However, we saw that the angular rate was also affected by a BIAS

For mitigating it, may use a basic off-line approach, which requires the platform to be stationary during certain "calibration time"

Here, we propose an on-line way, to estimate the bias, jointly with the vehicle's pose. This means that we do not require that restrictive calibration procedure.

How?

First, we augment the state vector, and propose a new process model

$$\mathbf{X} = \begin{bmatrix} x & y & \varphi & b \end{bmatrix}^T$$

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

$$\Downarrow$$

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \phi(k+1) \\ b(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \\ b(k) \end{bmatrix} + T \cdot \begin{bmatrix} v(k) \cdot \cos(\phi(k)) \\ v(k) \cdot \sin(\phi(k)) \\ \omega_b(k) - b(k) \\ 0 \end{bmatrix}$$

$b(k)$: bias

$\omega_b(k)$: biased angular rate measurement

here, we consider the case in which bias, ***b*** , is constant, i.e. , b(k+1)=b(k)

Our covariance matrix, **P**, is now 4x4.

(because **X** is 4x1)

$$\mathbf{X} = \begin{bmatrix} x \\ y \\ \phi \\ b \end{bmatrix} \in R^4$$

We still have the same observations in our estimation process.

$$h(\mathbf{X}) = \begin{bmatrix} h_1(x, y, \phi, b) \\ h_2(x, y, \phi, b) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_a - x)^2 + (y_a - y)^2} \\ \tan^{-1}(y_a - y, x_a - x) - \phi \end{bmatrix}$$

$$\Downarrow$$

$$= \begin{bmatrix} \sqrt{(x_a - x)^2 + (y_a - y)^2} \\ \tan^{-1}(y_a - y, x_a - x) - \phi \end{bmatrix} = \begin{bmatrix} r \\ \alpha \end{bmatrix}$$

But now the **H** matrix has 4 columns (because X is 4x1).

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^-} = \left.\left[\begin{array}{c:c:c:c} \dfrac{\partial h_1}{\partial x} & \dfrac{\partial h_1}{\partial y} & \dfrac{\partial h_1}{\partial \phi} & \color{blue}{\dfrac{\partial h_1}{\partial b}} \\ \hdashline \dfrac{\partial h_2}{\partial x} & \dfrac{\partial h_2}{\partial y} & \dfrac{\partial h_2}{\partial \phi} & \color{blue}{\dfrac{\partial h_2}{\partial b}} \end{array}\right]\right|_{\mathbf{x}=\hat{\mathbf{x}}^-}$$

Comment: here, I am considering the case in which we process 2 observations, one range and one bearing observation, associated to a Landmark; but the concept is similar for other cases.
I am also considering the LiDAR is pose, in the vehicle's CF, is [0;0;0]

... our **H** matrix has 4 columns.

$$\mathbf{H} = \left.\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^-} = \left.\begin{bmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} & \frac{\partial h_1}{\partial \phi} & \frac{\partial h_1}{\partial b} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} & \frac{\partial h_2}{\partial \phi} & \frac{\partial h_1}{\partial b} \end{bmatrix}\right|_{\mathbf{x}=\hat{\mathbf{x}}^-}$$

$$\mathbf{H} = \left.\begin{bmatrix} -\frac{(x_a - x)}{\sqrt{(x_a - x)^2 + (y_a - y)^2}} & -\frac{(y_a - y)}{\sqrt{(x_a - x)^2 + (y_a - y)^2}} & 0 & 0 \\ \frac{(y_a - y)}{(x_a - x)^2 + (y_a - y)^2} & \frac{-(x_a - x)}{(x_a - x)^2 + (y_a - y)^2} & -1 & 0 \end{bmatrix}\right|_{\mathbf{X}=\hat{\mathbf{X}}^-}$$

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

→ However, the rest of the update operations are as before (but using our new definitions of H, P, etc.)

Prediction step?

We use the new nominal model.

$$\mathbf{x} = \begin{bmatrix} x & y & \varphi & b \end{bmatrix}^T$$

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

$$\Downarrow$$

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \phi(k+1) \\ b(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \\ b(k) \end{bmatrix} + T \cdot \begin{bmatrix} v(k) \cdot \cos\left(\phi(k)\right) \\ v(k) \cdot \sin\left(\phi(k)\right) \\ \omega_b(k) - b(k) \\ 0 \end{bmatrix}$$

here b(k+1) = b(k)

Prediction step

$$\mathbf{x} = \begin{bmatrix} x & y & \varphi & b \end{bmatrix}^T$$

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

$$\Downarrow$$

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \phi(k+1) \\ b(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \\ b(k) \end{bmatrix} + T \cdot \begin{bmatrix} v(k) \cdot \cos(\phi(k)) \\ v(k) \cdot \sin(\phi(k)) \\ \omega_b(k) - b(k) \\ 0 \end{bmatrix}$$

We have a slightly different Jacobian matrix,

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

Why is it 4x4? → because **X** is now 4x1

How is **Ju** now? (you do that part) (what is the size of that Jacobian matrix? →(4x2) (why?)

Initializing the expected value of X.

Expected value of **b**? We usually assume b(0|0)=0
(except we knew something which could give us a better clue).

The rest of X(0|0) is initialized as we usually do.

Initializing the covariance matrix P

$$\mathbf{P}(0\,|\,0) = \begin{bmatrix} .. & .. & .. & 0 \\ .. & .. & .. & 0 \\ .. & .. & .. & 0 \\ 0 & 0 & 0 & p_{4,4}(0) \end{bmatrix}$$

Case: suppose we know from the gyroscope's manual (or from laboratory experiments) that the bias can be in the range [-0.5,+0.5] degrees/second.
So, we may try to play safe , so we set

$$p_{4,4}(0) = \left(\frac{1 \cdot \pi}{180}\right)^2$$

Initializing the covariance matrix P

$$\mathbf{P}(0|0)=\begin{bmatrix} .. & .. & .. & 0 \\ .. & .. & .. & 0 \\ .. & .. & .. & 0 \\ 0 & 0 & 0 & p_{4,4}(0) \end{bmatrix}$$

What would it happen if we specified a very low value $p_{4,4}(0)$? (e.g. $p_{4,4}(0)=0$)

➔ the estimator would trust on the initially proposed expected value of the bias; consequently, would not update it adequately.

So, we must not lie in the initialization.
Being overconfident is a risky matter.

And Q?

$$\mathbf{Q} = \begin{bmatrix} .. & .. & .. & 0 \\ .. & .. & .. & 0 \\ .. & .. & .. & 0 \\ 0 & 0 & 0 & q_{4,4} \end{bmatrix}$$

We need to “tell “ the EKF how much we trust on that model (“bias is constant”)

more➔

And Q?

$$\mathbf{Q} = \begin{bmatrix} .. & .. & .. & 0 \\ .. & .. & .. & 0 \\ .. & .. & .. & 0 \\ 0 & 0 & 0 & q_{4,4} \end{bmatrix}$$

We need to "tell " the EKF how much we trust on that model ("bias is constant")

If we were sure that the bias is constant then we would propose $q_{4,4} = 0$

But the bias may be a slowly time varying variable. So, we need to specify some distrust about the nominal model "bias is constant"

bias slowly time varying variable ➔

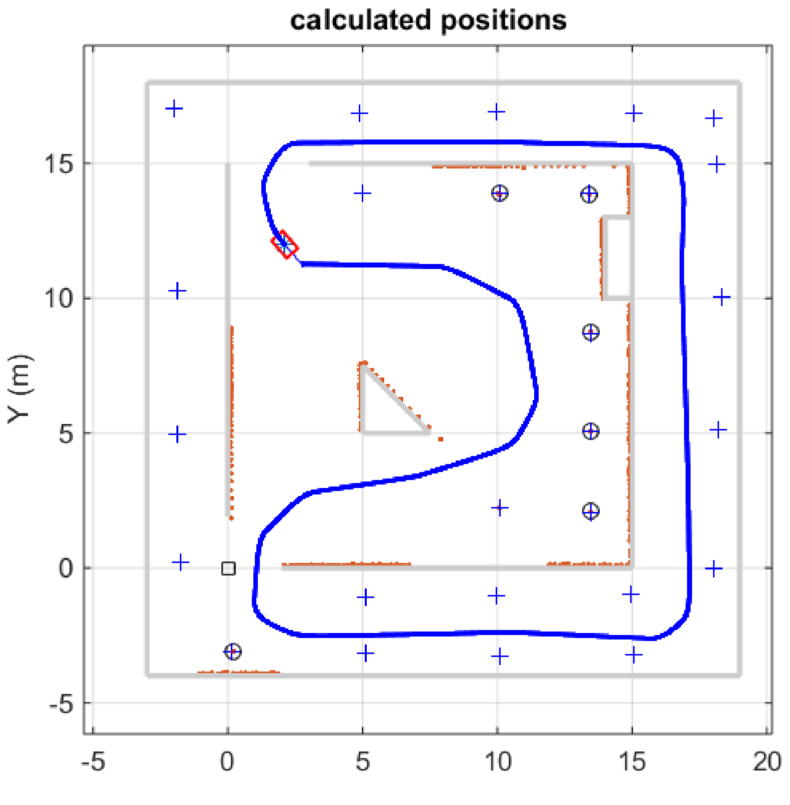
Case in which the bias does change (slowly time varying)

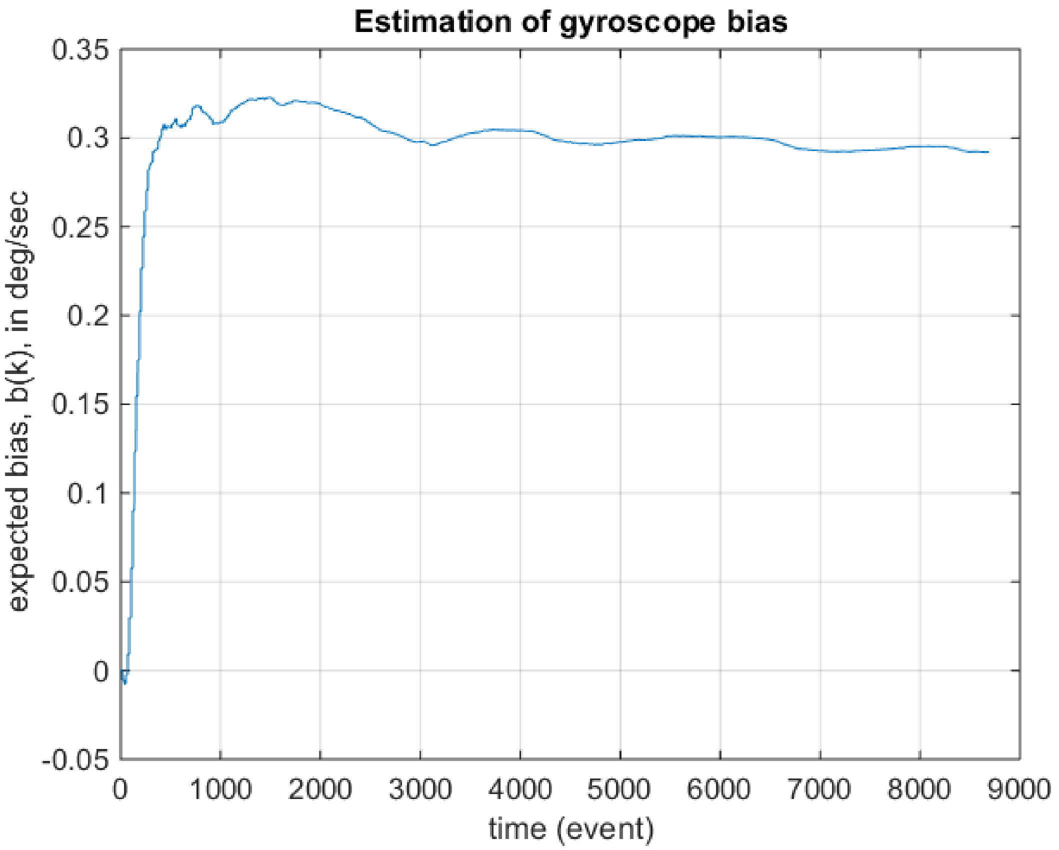
Suppose the case in which we suspect that the bias may change, we know that $b(k+1)=b(k)$ is not perfectly correct. However, we do not know the dynamics of the variation of b, so we can not include any better process model for describing the evolution of the state ***b***. Consequently, we keep our nominal model $b(k+1)=b(k)$; however, in this case, we do not assume that it is "perfect". In order to include certain distrust about our model about ***b*** we set a proper value for $q_{4,4}>0$. We are saying now that the model is $b(k+1)=b(k)+\xi_b(k)$, where the variable $\xi_b(k)$ is assumed to be a random variable, whose time sequence is WHITE (WGN).

Example, if we know that the bias can change up to 1 degree/second in 10 minutes, then we can define its variance $q_{4,4}=\left(T\cdot 1\cdot(\pi/180)\cdot 1/(10\cdot 60)\right)^2$ (a covariance associated to a standard deviation twice this value would be also adequate); where T is the sample time of the discrete process model. You can note that we set the covariance entries $q_{4,i}=q_{i,4}=0 \quad \left(\forall \quad i\in\{1,2,3\}\right)$, because we are sure that the uncertainty in the bias process model does not have dependency with other sources of uncertainty in the rest of the process model.

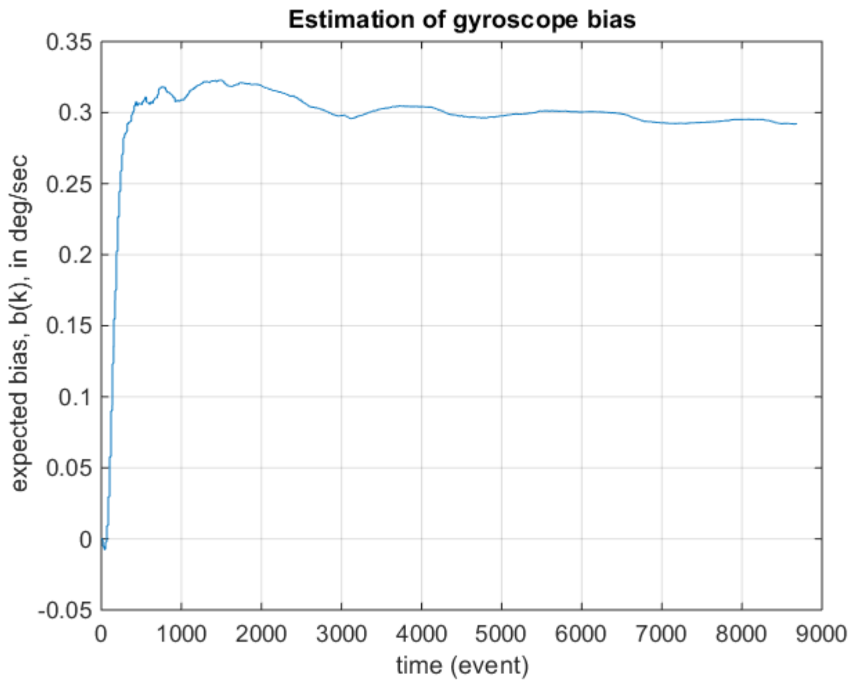
What performance should we expect?

It depends on how rich are the observations, and how much uncertainty affects the rest of the process model.





How is that? I do not see "b" participating in h(X).
How is that we estimate b?? Why is it possible?



OBSERVABILITY.

There is statistical dependency between our RVs . If we get information about a subset of them e.g. from certain "$h(X_1)$ =measurement" (i.e. a likelihood function) we also help the rest of the RVs that are statistical dependent with X_1

Where is that statistical dependency coming from?

From our prediction and observation models, being continuously applied, several times, since time t_0 (k=0)

Where is that statistical dependency represented / stored?

In the covariance matrix P
(which we always keep maintaining)

We see it working (using data from Project 2)

General case

What we discussed is a particular case of a more general problem, in which we simultaneously estimate states and certain parameters.

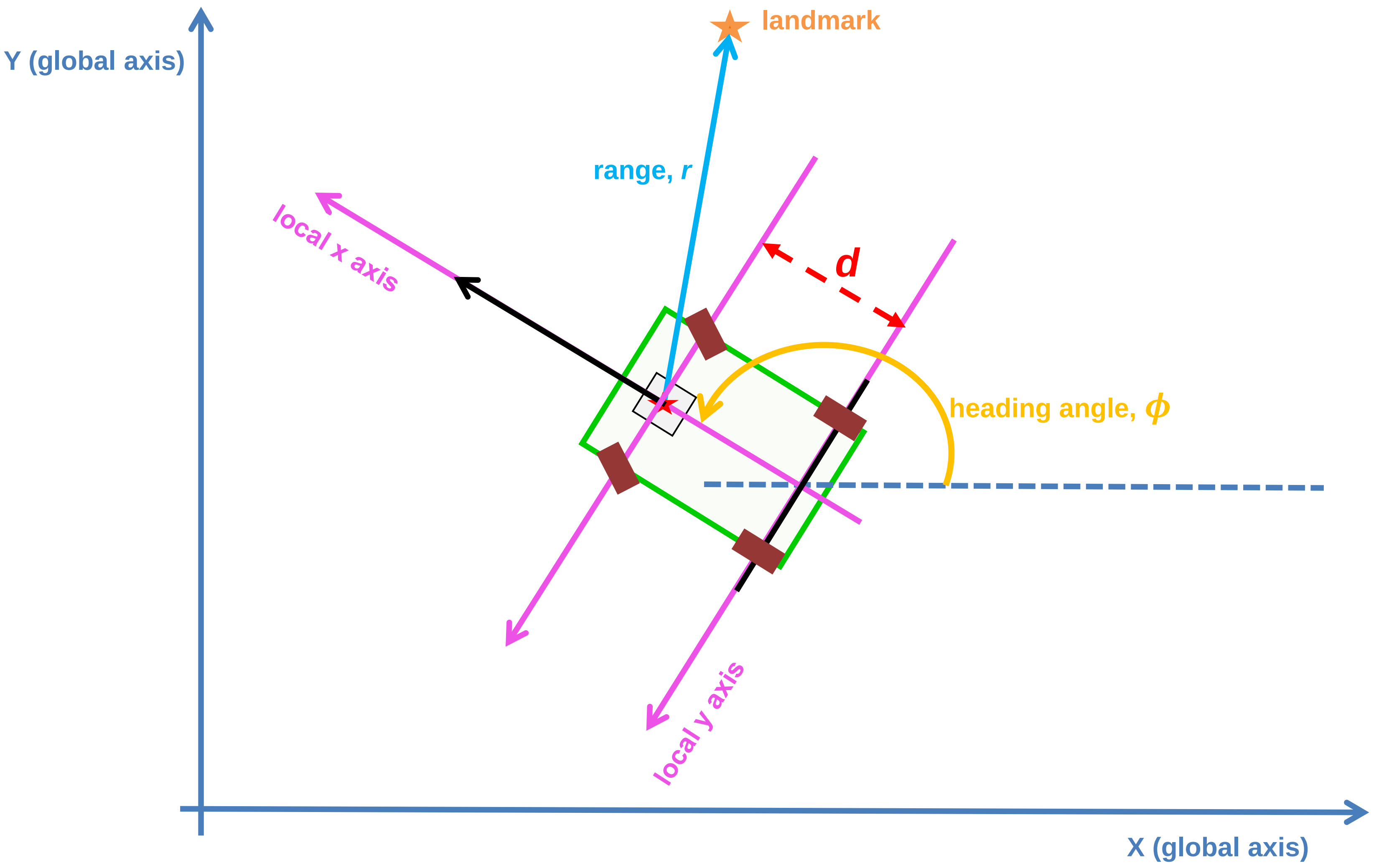
$$\mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\alpha}\right)$$
$$y = \mathbf{h}\left(\mathbf{x}(k), \boldsymbol{\beta}\right)$$

$$\Downarrow$$

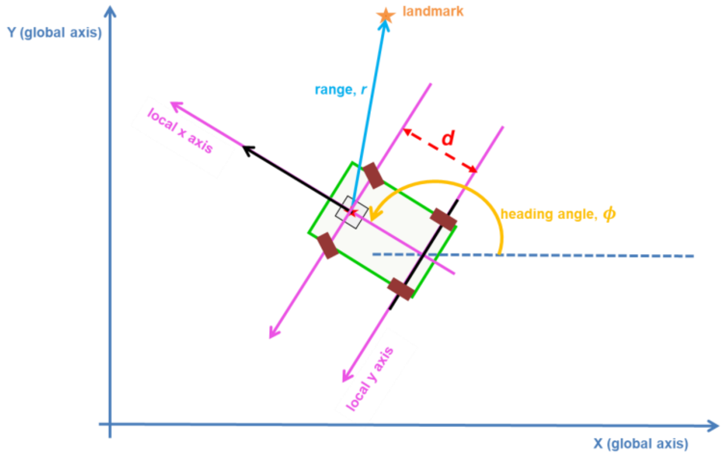
$$\begin{bmatrix} \mathbf{X}(k+1) \\ \boldsymbol{\alpha}(k+1) \\ \boldsymbol{\beta}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{f}\left(\mathbf{X}(k), \mathbf{u}(k), \boldsymbol{\alpha}(k)\right) \\ \boldsymbol{\alpha}(k) \\ \boldsymbol{\beta}(k) \end{bmatrix}$$

Can we estimate the LiDAR position on the platform?

(at least Parameter “Lx” in Projects 1 and 2.)



Note: In this example, the LiDAR not located at the position described by the state of the system, as the one in our UGV.
(Parameter "*d*" is our "Lx" in project 1)



$$\begin{bmatrix} \sqrt{(x_a - x_s)^2 + (y_a - y_s)^2} \\ \tan^{-1}(y_a - y_s, x_a - x_s) - \phi \end{bmatrix} = \begin{bmatrix} r \\ \alpha \end{bmatrix}$$

$$(x_s, y_s, \phi) : \text{Lidar's pose } \left(\text{ aligned with vehicle, } \phi_s = \phi \ \right)$$

$$x_s = x + d \cdot \cos(\phi)$$

$$y_s = y + d \cdot \sin(\phi)$$

$$\Downarrow$$

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \sqrt{(x_a - x - d \cdot \cos(\phi))^2 + (y_a - y - d \cdot \sin(\phi))^2} \\ \tan^{-1}(y_a - y - d \cdot \sin(\phi), x_a - x - d \cdot \cos(\phi)) - \phi \end{bmatrix} = \begin{bmatrix} r \\ \alpha \end{bmatrix} = \mathbf{h}(x, y, \phi, d)$$

Can we estimate the LiDAR position on the platform?

$$\mathbf{g}(\mathbf{x}_s) = \begin{bmatrix} g_1(x_s, y_s, \phi_s) \\ g_2(x_s, y_s, \phi_s) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_a - x_s)^2 + (y_a - y_s)^2} \\ \tan^{-1}(y_a - y_s, x_a - x_s) - \phi_s \end{bmatrix} = \begin{bmatrix} r \\ \alpha \end{bmatrix}$$

$$x_s = x + d \cdot \cos(\phi)$$

$$y_s = y + d \cdot \sin(\phi)$$

$$\phi_s = \phi$$

$$\Downarrow$$

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \sqrt{(x_a - x - d \cdot \cos(\phi))^2 + (y_a - y - d \cdot \sin(\phi))^2} \\ \tan^{-1}(y_a - y - d \cdot \sin(\phi), x_a - x - d \cdot \cos(\phi)) - \phi \end{bmatrix} = \begin{bmatrix} r \\ \alpha \end{bmatrix} = \mathbf{h}(x, y, \phi, d)$$

for obtaining H = $\dfrac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}$ we may use the chain rule $\dfrac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} = \dfrac{\partial \mathbf{h}}{\partial \mathbf{x}_s} \cdot \dfrac{\partial \mathbf{x}_s}{\partial \mathbf{x}}$

"It is magical, we can keep augmenting X, to include more variables (e.g. model parameters)"

Not so easy! → ONLY if the resulting system is observable.!

OBSERVABILITY

It is difficult to infer observability of systems which present these or some of these characteristics:

1) Nonlinear and/or time-varying (non LTI)
2) Asynchronous
3) multiple output variables
4) multi-rate
5) involving diverse sensors qualities
6) sporadic unavailability of certain sensors' measurements (e.g. " I cannot see landmarks here")

We verify it in practical terms (in MTRN4010, and in many application contexts).

1) A priori, we perform some simulations, to verify convergency, empirically.
2) In runtime: we keep an eye on the covariance matrix, P. Easy, by just inspecting the diagonal elements (which are the variances o f the marginal PDFs, associated to the estimates of each of the states and parameters being estimated.

→ However, covariance matrix "may lie", if we lie to the EKF such as assuming over optimistic variances about noises, or if nonlinearities are too acute to be linearized. Other usual cases is when the assume that a noise is WGN, for a noise that is far from behaving in that way (such as the gyroscope bias, that is why we preferred estimating it than assuming it was simply WGN!)

Case:

What about the LIDAR not being aligned with the chassis, or whose shift in orientation are not accurately known to us? Could we estimate it?

→ Think about it, and about related equations.

(we **do not** implement it, in Project 2)

Let run some cases, now, with the data you are using in Project 2.

Cases

1) simple EKF
2) Estimating Gyro bias.
3) Estimating “Lx”

Other cases, in other areas of application?

Estimating pendulum states and certain parameter, in run time?

Suppose we know the friction was changed, during operation, but we do not know the new friction coefficient.

Would we be able to adapt those that change?

(tutorial problem)

Augmenting state vector
Other interesting /real cases.

Suppose we do not have speed sensor.
Can we deal with that?
Can we estimate the longitudinal component ("speed") of the velocity vector?

Answer: In the context and conditions of operation we have in Project 2, **YES**.
(because under those conditions, the augmented system is observable. We can verify it empirically)

(we do not implement it in 2022, and we do not include it in any assessment)

Project 2, Part B involves estimating the gyroscope bias, in runtime.

In some dataset, the bias is slowly time varying (e.g. like an IMU being affected by temperature change)

So, we will need to consider $q(4,4)>0$.

Can we estimate the gyroscope’s in an easier way?

Traditional calibration.

If the vehicle stays still / static during an interval of time, we can exploit that fact to estimate the bias.

➔ We average the measured values during that period of time.
➔ That average value is usually well close to the value of the bias.

$$\omega_{measured}(k) = \omega_{real}(k) + b + \mu(k)$$

$$\omega_{real}(k) = 0 \text{ when the vehicle is perfectly still}$$

$\mu(k)$: WGN of known variance (WGN has mean = 0)

b : constant

Conclusion:

We have seen how to augment the state vector, for including parameters of interest, which we want/need to estimate

We considered a particular case. However, that “trick” can be applied to other cases, in which the values of certain model parameters need to be known.

(We end here)

Next topic: Exploiting Optimization for state and parameter estimation

Go to new presentation ApplyingBasicOptimization ➔