Filtrage de Kalman appliqué à la navigation et à l'estimation d'état

Nicolas Merlinge

ROB-312

ONERA
THE FRENCH AEROSPACE LAB

ENSTA
IP PARIS

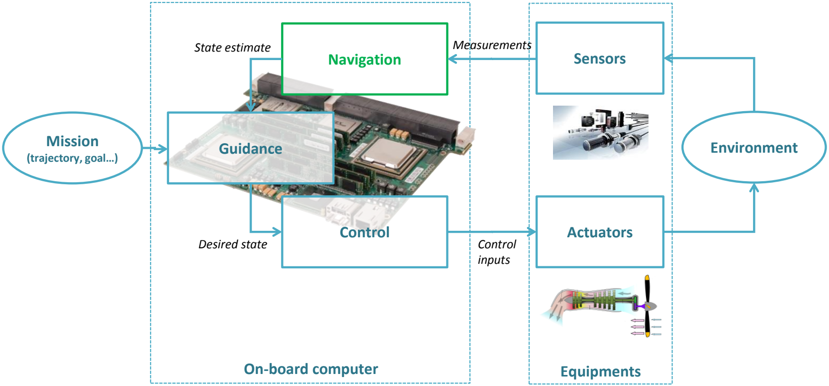
ONERA - Office National d'Etudes et de Recherches Aérospatiales

Motivation

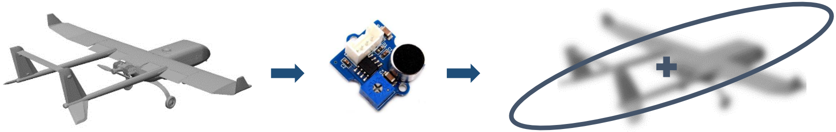
Autonomous systems need to perform accurate state estimation (e.g., self localization, objective assessment...)

... with limited embedded calculation capabilities.

Guidance, Navigation and Control (GNC)



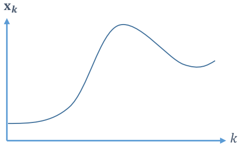
State estimation for navigation

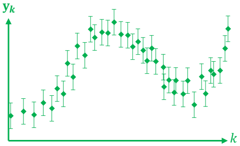


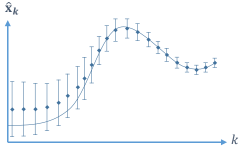
$\mathbf{x}_k$ = position, velocity...

$\mathbf{y}_k$ = measurements

$\hat{\mathbf{x}}_k$ = estimated position, velocity...







State estimation consists in retrieving the vehicle's state from noisy and incomplete measurements and uncertain evolution model.

What we know

Theoretical state evolution (dynamical model):

$$\begin{aligned} \dot{\mathbf{x}} &= F(\mathbf{x}, \mathbf{u}) \\ \mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_k) \end{aligned} \tag{1}$$

Theoretical observation equation (sensor model):

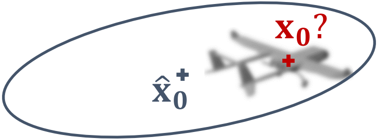
$$\mathbf{y}_k = h(\mathbf{x}_k) \tag{2}$$

However, these equations are not totally representative of the actual system, e.g.:

- unexpected wind, friction, unmodeled dynamics...
- sensor noise, unmodeled disturbances...

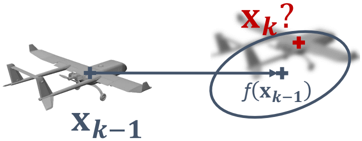
What we **don't** know (uncertainties)

- Initial state uncertainty



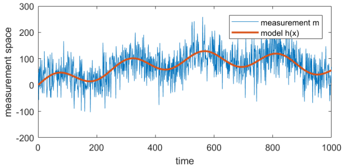
- Process noise (dynamics)

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)$$



- Measurement noise (and potentially some bias)

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k)$$



A way to model uncertainties: probability distribution functions

- State distribution:

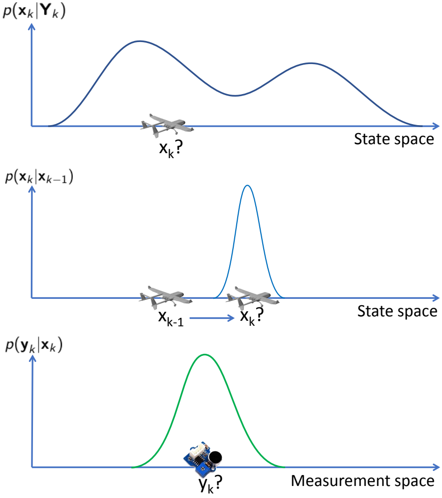
$$p(\mathbf{x}_k|\mathbf{y}_1, \ldots, \mathbf{y}_k) \triangleq p(\mathbf{x}_k|\mathbf{Y}_k)$$

- Process noise distribution

$$p(\mathbf{x}_k|\mathbf{x}_{k-1})$$

- Measurement distribution

$$p(\mathbf{y}_k|\mathbf{x}_k)$$



Optimal filter equations (Bayesian filtering)

State density propagation (dynamics, Chapman-Kolmogorov equation):

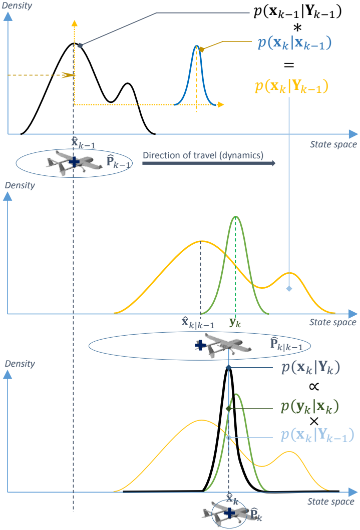
$$p(\mathbf{x}_k|\mathbf{Y}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1})\mathrm{d}\mathbf{x}_{k-1} \tag{3}$$

State density correction/update (measurements, Bayes rule):

$$p(\mathbf{x}_k|\mathbf{Y}_k) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Y}_{k-1})}{p(\mathbf{y}_k|\mathbf{Y}_{k-1})} \tag{4}$$

(Measurements $\mathbf{y}_i$ and $\mathbf{y}_j$ ($\forall\ i \neq j$) are assumed to be statistically independent)

Optimal filter equations (Bayesian filtering)



(a) Prediction

Convolution ($*$) of the **prior conditional density** with the state transition density. The transition density accounts for the deterministic dynamics f_k and its uncertainty (process noise $\mathbf{w}_k$).

(b) Predicted density, new measurement

Step (a) results in the **predicted conditional density**, whose support is usually larger than the prior density.
A measurement $\mathbf{y}_k$ is now available. It will introduce information in the estimation.

(c) Correction

The predicted density is multiplied with the measurement density, leading to the **posterior conditional density**.
Its support is usually smaller than the predicted density. It yields a refined estimate $\hat{\mathbf{x}}_k$ and covariance $\hat{\mathbf{P}}_k$.
Then, one can iterate back to step (a) for further time-steps.

Linear-Gaussian models

Linear-Gaussian state evolution (dynamical model):

$$\begin{aligned} &\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{C}_k \mathbf{u}_k + \mathbf{G}_k \mathbf{w}_k \\ &\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \qquad (\mathbf{w}_k \in \mathbb{R}^{d_w}, \mathbf{Q}_k \in \mathbb{R}^{d_w \times d_w}) \end{aligned} \tag{5}$$

Linear-Gaussian observation equation (sensor model):

$$\begin{aligned} &\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \\ &\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \qquad (\mathbf{v}_k \in \mathbb{R}^{d_m}, \mathbf{R}_k \in \mathbb{R}^{d_m \times d_m}) \end{aligned} \tag{6}$$

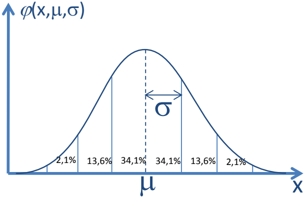
Gaussian initial state uncertainty:

$$\mathbf{x}_0 \sim \mathcal{N}(\mathbf{s}_0, \mathbf{P}_0) \qquad (\mathbf{s}_0 \in \mathbb{R}^{d}, \mathbf{P}_0 \in \mathbb{R}^{d \times d})$$

Gaussian density (recall)

Only two parameters:

- Expectancy (theoretical mean)
- Covariance matrix (dispersion and correlations)

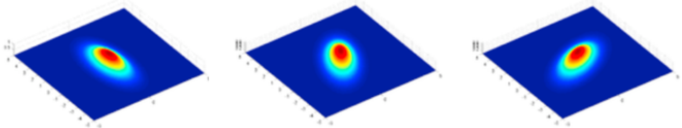


(the variance is the square of the standard deviation σ)

Gaussian pdf:

$$\varphi(X, \boldsymbol{\mu}, \mathbf{P}) = \frac{1}{\sqrt{(2\pi)^d |\mathbf{P}|}} \exp\left(-\frac{1}{2}(X - \boldsymbol{\mu})^T \mathbf{P}^{-1}(X - \boldsymbol{\mu})\right)$$

Illustration in 2D: $\mathbf{P} = \begin{bmatrix} \sigma_a^2 & cov(a,b) \\ cov(a,b) & \sigma_b^2 \end{bmatrix} > 0$



Gaussian Bayesian filter

State density propagation (dynamics, Chapman-Kolmogorov equation):

$$\begin{array}{ccccc} p(\mathbf{x}_k|\mathbf{Y}_{k-1}) & = \int & p(\mathbf{x}_k|\mathbf{x}_{k-1}) & p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1}) & \mathrm{d}\mathbf{x}_{k-1} \\ \mathcal{N}(\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}) & \leftarrow & \mathcal{N}(\mathbf{x}_k, \mathbf{Q}_k) & \mathcal{N}(\mathbf{x}_{k-1}, \mathbf{P}_{k-1}) & \end{array} \tag{7}$$

State density correction/update (measurements, Bayes rule):

$$\begin{array}{cccc} p(\mathbf{x}_k|\mathbf{Y}_k) & \propto & p(\mathbf{y}_k|\mathbf{x}_k) & p(\mathbf{x}_k|\mathbf{Y}_{k-1}) \\ \mathcal{N}(\mathbf{x}_k, \mathbf{P}_k) & \leftarrow & \mathcal{N}(\mathbf{y}_k, \mathbf{R}_k) & \mathcal{N}(\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}) \end{array} \tag{8}$$

$\Rightarrow$ Gaussian models yield Gaussian conditional state densities

Kalman Filter equations [1]

Prediction step:

$$\begin{aligned}\widehat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k\widehat{\mathbf{x}}_{k-1} + \mathbf{C}_k\mathbf{u}_k \\ \widehat{\mathbf{P}}_{k|k-1} &= \mathbf{F}_k\widehat{\mathbf{P}}_{k-1}\mathbf{F}_k^T + \mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^T\end{aligned} \tag{9}$$

Correction step (where $\mathbf{y}_k - \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1}$ is the *innovation* term and $\mathbf{K}_k$ is the Kalman gain):

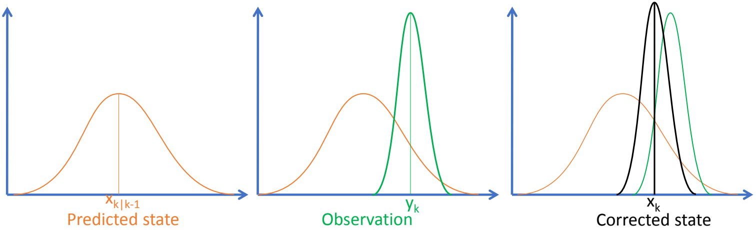
$$\begin{aligned}\widehat{\mathbf{x}}_k &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1}\right) \\ \widehat{\mathbf{P}}_k &= (\mathbf{I}_d - \mathbf{K}_k\mathbf{H}_k)\,\widehat{\mathbf{P}}_{k|k-1} \\ \mathbf{K}_k &= \widehat{\mathbf{P}}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1} \\ \mathbf{S}_k &= \mathbf{R}_k + \mathbf{H}_k\widehat{\mathbf{P}}_{k|k-1}\mathbf{H}_k^T\end{aligned} \tag{10}$$

1D illustration of the correction step

Assume $H = 1$, $x_{k|k-1} \in \mathbb{R}$, $y_k \in \mathbb{R}$, then:

$$K = \frac{P_k}{R_k + P_k}$$

$$x_k = x_{k|k-1} + K(y_k - x_{k|k-1}) = \frac{\frac{x_{k|k-1}}{P_{k|k-1}} + \frac{y_k}{R_k}}{\frac{1}{P_{k|k-1}} + \frac{1}{R_k}} = \frac{x_{k|k-1}R_k + y_k P_{k|k-1}}{P_{k|k-1} + R_k}$$



2D illustration: $\mathbf{x}_k = [p_k, v_k]^T \qquad \mathbf{y}_k = [p_k + w_k]$

- $\mathbf{x}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \upsilon_k \qquad \upsilon_k \sim \mathcal{N}\left(0, \begin{bmatrix} 1^2 & 0 \\ 0 & 0.001^2 \end{bmatrix}\right)$
- $y_k = p_k + w_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_k + w_k \qquad w_k \sim \mathcal{N}(0, 1^2))$

Initialization: $\widehat{\mathbf{x}}_0 = \begin{bmatrix} 0 \\ 10 \end{bmatrix} \quad \mathbf{P}_0 = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \quad \left(\mathbf{x}_0 = \begin{bmatrix} 20 \\ 12 \end{bmatrix}\right)$

Prediction step ($\Delta t = 1$ s):

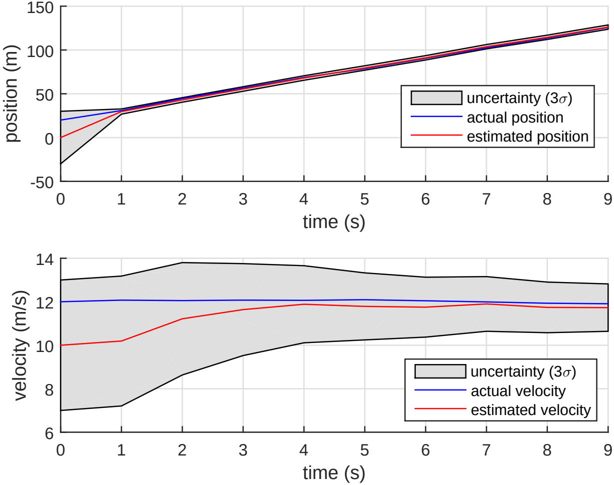
$$\widehat{\mathbf{x}}_{1|0} = \begin{bmatrix} 10 \\ 10 \end{bmatrix} \quad \widehat{\mathbf{P}}_{1|0} = \begin{bmatrix} 102 & 1 \\ 1 & 1.001 \end{bmatrix} \qquad \left(\mathbf{x}_1 = \begin{bmatrix} 30.83 \\ 12.07 \end{bmatrix}\right)$$

Correction step:

$$\mathbf{K}_1 = \begin{bmatrix} 0.9903 \\ 0.0097 \end{bmatrix} \qquad y_k - \mathbf{H}_k \mathbf{x}_k = 29.91 - 10 = 19.91$$

$$\widehat{\mathbf{x}}_1 = \begin{bmatrix} 29.7 \\ 10.2 \end{bmatrix} \qquad \widehat{\mathbf{P}}_1 = \begin{bmatrix} 0.9903 & 0.0097 \\ 0.0097 & 1.0903 \end{bmatrix}$$

2D illustration: $\mathbf{x}_k = [p_k, v_k]^T \qquad \mathbf{y}_k = [p_k + w_k]$



Nonlinear-Gaussian models

Nonlinear-Gaussian state evolution (dynamical model):

$$\begin{aligned} &\mathbf{x}_k = f(\widehat{\mathbf{x}}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \\ &\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \qquad (\mathbf{w}_k \in \mathbb{R}^d, \mathbf{Q}_k \in \mathbb{R}^{d \times d}) \end{aligned} \tag{11}$$

Nonlinear-Gaussian observation equation (sensor model):

$$\begin{aligned} &\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k) \\ &\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \qquad (\mathbf{v}_k \in \mathbb{R}^{d_m}, \mathbf{R}_k \in \mathbb{R}^{d_m \times d_m}) \end{aligned} \tag{12}$$

Gaussian initial state uncertainty:

$$\mathbf{x}_0 \sim \mathcal{N}(\mathbf{s}_0, \mathbf{P}_0) \qquad (\mathbf{s}_0 \in \mathbb{R}^d, \mathbf{P}_0 \in \mathbb{R}^{d \times d})$$

Extended Kalman Filter for nonlinear models

Prediction step:

$$
\begin{aligned}
\widehat{\mathbf{x}}_{k|k-1} &= f(\widehat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \\
\widehat{\mathbf{P}}_{k|k-1} &= \mathbf{F}_k \widehat{\mathbf{P}}_{k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \\
\mathbf{F}_k &= \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\widehat{\mathbf{x}}_{k-1}} \in \mathbb{R}^{d \times d} \\
\mathbf{G}_k &= \left.\frac{\partial f}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}_k} \in \mathbb{R}^{d \times d_w}
\end{aligned}
\tag{13}
$$

Correction step (where $\mathbf{y}_k - h(\widehat{\mathbf{x}}_{k|k-1})$ is the *innovation* term and $\mathbf{K}_k$ is the Kalman gain):

$$
\begin{aligned}
\widehat{\mathbf{x}}_k &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{y}_k - h(\widehat{\mathbf{x}}_{k|k-1}) \right) \\
\widehat{\mathbf{P}}_k &= (\mathbf{I}_d - \mathbf{K}_k \mathbf{H}_k) \widehat{\mathbf{P}}_{k|k-1} \\
\mathbf{K}_k &= \widehat{\mathbf{P}}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \\
\mathbf{S}_k &= \mathbf{R}_k + \mathbf{H}_k \widehat{\mathbf{P}}_{k|k-1} \mathbf{H}_k^T \\
\mathbf{H}_k &= \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\widehat{\mathbf{x}}_k} \in \mathbb{R}^{d_m \times d}
\end{aligned}
\tag{14}
$$

Limitations and potential solutions

Extended Kalman Filter limitations:

- Local linearization of f and h: instability for severe non-linearities or large initial state uncertainty
- Gaussian approximations: no longer valid for non-linear models: introduce some bias

Advanced non-linear Gaussian filters:

- Unscented Kalman Filter (UKF [2]): avoids linarization (Unscented Transform)
- Lie Group and invariant Kalman Filters (LG-EKF [3], IKF [4]): Kalman Filter with an exact linearization

Advanced non-linear non-Gaussian filter:

- Particle Filter (PF [5]): Monte-Carlo approximation of the Optimal Filter

Inertial Measurement Unit

Relativity principle:

Imagine a box without access to the outside world.

What can you measure inside the box?

Inertial Measurement Unit

Relativity principle:
Imagine a box without access to the outside world.

What can you measure inside the box?

- Specific accelerations (i.e., non gravitational)
- Angular velocities

What **cannot** you measure?

Inertial Measurement Unit

Relativity principle:
Imagine a box without access to the outside world.

What can you measure inside the box?

- Specific accelerations (i.e., non gravitational)
- Angular velocities

What **cannot** you measure?

- Gravity field

Inertial Measurement Unit

Accelerometer measurement:

$$\gamma_m = b + (1 + f)\gamma + w_\gamma \tag{15}$$

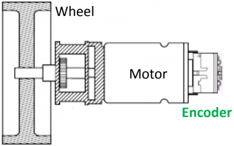
γ	:	actual specific acceleration
γ_m	:	measured specific acceleration
b	:	bias (constant)
f	:	scale factor
w_γ	:	sensor noise

Gyrometer measurement:

$$\omega_m = d + (1 + \tau)\omega + w_\omega \tag{16}$$

ω	:	actual absolute angular velocity
ω_m	:	measured angular velocity
d	:	angular drift (constant)
τ	:	scale factor
w_ω	:	sensor noise

Wheel odometry



Model 1:

$$\Delta p_m = (1+f)\Delta p + w \tag{17}$$

Δp	:	actual wheel position variation since last measureme
Δp_m	:	measured wheel position variation
f	:	scale factor
$w \sim \mathcal{N}(0, \sigma^2)$	:	sensor noise

Model 2:

$$\Delta p_m = \Delta p + w_{\Delta p} \tag{18}$$

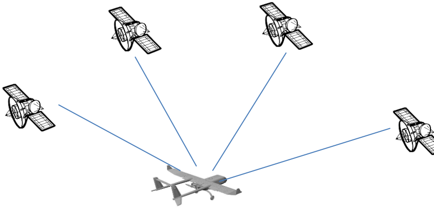
$w_{\Delta p} \sim \mathcal{N}(0, (\Delta p \sigma)^2)$: sensor noise

Global Navigation System by Satellite (GNSS)

Pseudo-measurements for satellite i:

$$\begin{aligned} PR_i &= \|\mathbf{p} - \mathbf{p}_i\| + cB_h + w_{pr} \\ PV_i &= \left\langle \mathbf{v} - \mathbf{v}_i, \frac{\mathbf{p}-\mathbf{p}_i}{\|\mathbf{p}-\mathbf{p}_i\|} \right\rangle + c\dot{B}_h + w_{pv} \end{aligned} \tag{19}$$

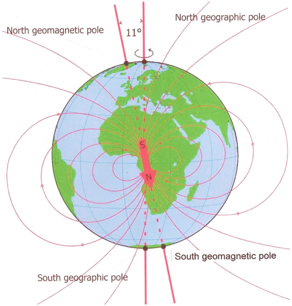
- $\mathbf{p}$, $\mathbf{v}$: receptor's position and velocity
- $\mathbf{p}_i$, $\mathbf{v}_i$: satellite's position and velocity
- c : speed of light
- B_h : receptor's clock bias
- w_{pr}, w_{pv} : measurement noises.



From a minimum of 4 satellites, instantaneous navigation solution:

$$\begin{aligned} \mathbf{p}_m &= \mathbf{p} + \mathbf{w}_p \\ \mathbf{v}_m &= \mathbf{v} + \mathbf{w}_v \end{aligned} \tag{20}$$

Magnetometer

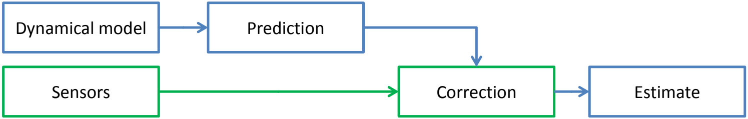


$$\mathbf{b}_m = \mathbf{R}_{earth \to body} \mathbf{b} + \mathbf{w} \tag{21}$$

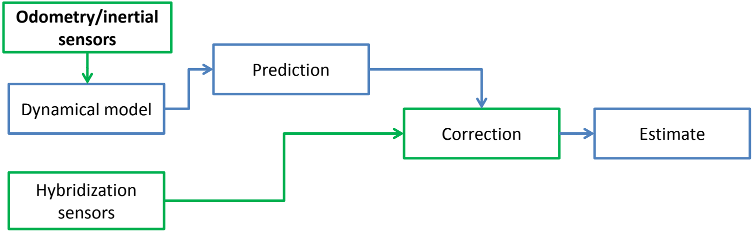
$\mathbf{b}$	:	actual local magnetic field
$\mathbf{b}_m$	:	measured magnetic field
$\mathbf{R}_{earth \to body}$	:	sensor attitude matrix in Earth frame
$\mathbf{w}$	:	sensor noise

Two possible implementations of a Kalman filter

- Implementation with a theoretical dynamical model

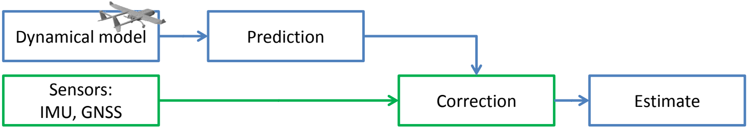


- Navigation by odometry/inertial Hybridization

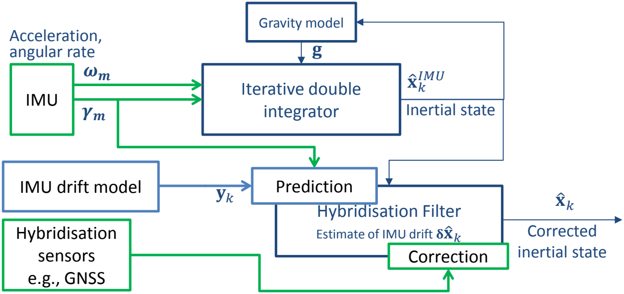


Example: IMU hybridization

- Implementation with a specific aircraft dynamical model

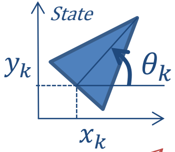


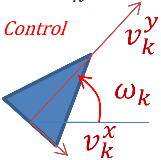
- Navigation by odometry/inertial Hybridization



Non-linear model: Prediction

State (ground frame) : $\mathbf{x}_k = [x_k, y_k, \theta_k]^T$,
Control (robot frame) : $\mathbf{u}_k = [v_k^x, v_k^y, \omega_k]^T$
Noised Odometry (robot frame) :
$\tilde{\mathbf{u}}_k = [\tilde{v}_k^x, \tilde{v}_k^y, \tilde{\omega}_k]^T \sim \mathcal{N}(\mathbf{u}_k, \mathbf{Q}_k)$
Dynamics: $\mathbf{x}_k = f(\mathbf{x}_{k-1}, \tilde{\mathbf{u}}_k)$





$$\mathbf{x}_k = \begin{bmatrix} x_{k-1} + (\tilde{v}_k^x \cos\theta_{k-1} - \tilde{v}_k^y \sin\theta_{k-1})\Delta t \\ y_{k-1} + (\tilde{v}_k^x \sin\theta_{k-1} + \tilde{v}_k^y \cos\theta_{k-1})\Delta t \\ \theta_{k-1} + \tilde{\omega}_k \Delta t \end{bmatrix}$$

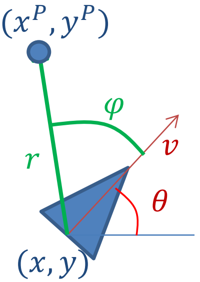
$$\mathbf{F}_k = \frac{\partial f}{\partial \mathbf{x}} = \begin{bmatrix} 1 & 0 & \dots \\ 0 & 1 & \dots \\ 0 & 0 & \dots \end{bmatrix} \qquad \mathbf{G}_k = \frac{\partial f}{\partial \mathbf{w}} = \frac{\partial f}{\partial \tilde{\mathbf{u}}} = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

Non-linear model: Correction

Measurements (e.g., radar):

$$\mathbf{y}_k = h(\mathbf{x}_k) = \begin{bmatrix} r_k \\ \varphi_k \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{(x_k^P - x_k)^2 + (y_k^P - y_k)^2} \\ \operatorname{atan} \frac{y_k^P - y_k}{x_k^P - x_k} - \theta_k \end{bmatrix}$$



$$\mathbf{H}_k = \frac{\partial h}{\partial \mathbf{x}} = \begin{bmatrix} -\frac{x_k^P - x_k}{\sqrt{(x_k^P - x_k)^2 + (y_k^P - y_k)^2}} & -\frac{y_k^P - y_k}{\sqrt{(x_k^P - x_k)^2 + (y_k^P - y_k)^2}} & 0 \\ \frac{y_k^P - y_k}{(x_k^P - x_k)^2 + (y_k^P - y_k)^2} & -\frac{x_k^P - x_k}{(x_k^P - x_k)^2 + (y_k^P - y_k)^2} & -1 \end{bmatrix}$$

The map of landmarks is *a priori* known and is encoded in matrix *Map*. At each time-step k, one landmark is randomly selected to simulate a measurement (the current landmark index in the map is *iFeature*).

TP

1. Prendre en main la structure du code et repérer les différents paramètres du filtre. Expliquer comment s'agencent les grandes parties du code (simulation du véhicule, des capteurs, de l'odométrie, du filtre de Kalman...).
2. Compléter le code avec les équations du filtre EKF (slide 19), le modèle dynamique (*motion_model*), le modèle de mesure (*observation_model*), les matrices jacobiennes (*get_obs_jac*, A, B) et commenter les résultats; (remarque: on pourra s'aider des fonctions utilitaires, par exemple *tcomp*)
3. Modifier la fréquence des mesures en utilisant la variable *dt_mesure*, qu'observe-t-on ? Expliquer.
4. Faire varier le bruit de dynamique du filtre (matrice *QEst*) qu'observe-t-on ? Expliquer.
5. Faire varier le bruit de mesure du filtre (matrice *REst*), qu'observe-t-on ? Expliquer.
6. Simuler un trou de mesures entre $t = 2500$ s et $t = 3500$ s en utilisant la variable *notValidCondition* et expliquer les résultats.
7. Faire varier le nombre d'amers et étudier les performances du filtre en fonction du nombre d'amers. Régler le filtre pour obtenir les meilleures performances possibles et expliquer les résultats.
8. Simuler le cas où seulement les mesures de distance sont disponibles (*range only*): modifier le code, régler le filtre (matrices de covariances QEst et REst) et expliquer les résultats. On pourra au besoin jouer sur le nombre d'amers.
9. Simuler le cas où seulement les mesures de direction sont disponibles (*angles only*): modifier le code, régler le filtre (matrices de covariances QEst et REst) et expliquer les résultats. On pourra au besoin jouer sur le nombre d'amers.

/!\ Les rapports de TP sont individuels. La moitié des points sera consacrée aux résultats, et l'autre moitié portera sur votre appropriation personnelle des concepts et l'interprétation des résultats.

Bibliography

1. Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of basic Engineering, 82(1), 35-45.
2. Wan, E. A., Van Der Merwe, R. (2001). The unscented Kalman filter. Kalman filtering and neural networks, 221-280.
3. Bourmaud, G., Mégret, R., Arnaudon, M., Giremus, A. (2015). Continuous-discrete extended Kalman filter on matrix Lie groups using concentrated Gaussian distributions. Journal of Mathematical Imaging and Vision, 51(1), 209-228.
4. Barrau, A., Bonnabel, S. (2018). Invariant kalman filtering. Annual Review of Control, Robotics, and Autonomous Systems, 1(1), 237-257.
5. Ristic, B., Arulampalam, S., Gordon, N. (2004). Beyond the Kalman filter. IEEE Aerospace and Electronic Systems Magazine, 19(7), 37-38.