Commande et estimation pour la robotique mobile

SYS5240

ESIEA 5A

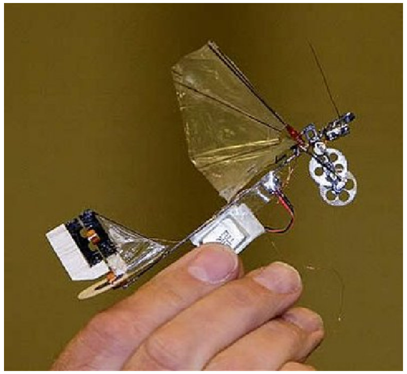
S. Bertrand
sbertrand@esiea.fr

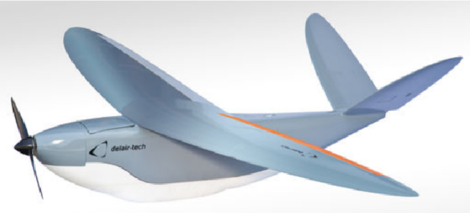
Application au drone

- Introduction
- Modélisation
- Commande
- Estimation

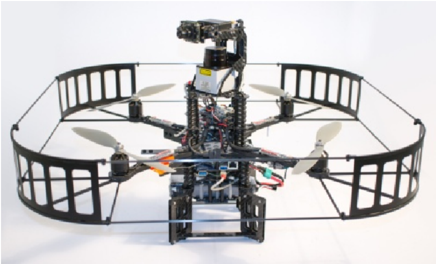
Introduction

- Différents types de véhicules
  - Voilure fixe, ailes battantes, voilure tournante
  - Toutes tailles (~avion de ligne -> « nano » drone)





- Dans ce cours :
  - drones miniatures à voilure tournante
  - quadrirotors



Introduction

- Drones quadrirotors : principe de fonctionnement
  - Différentes configurations de vol : en « X », en « + »
  - 6 degrés de liberté (3 translation, 3 rotations)
  - 4 commandes (vitesses de rotation des rotors)

  => système sous actionné
  - Couplage entre les mouvements de rotation et de translation

- Fonctionnement en « + » : translation verticale

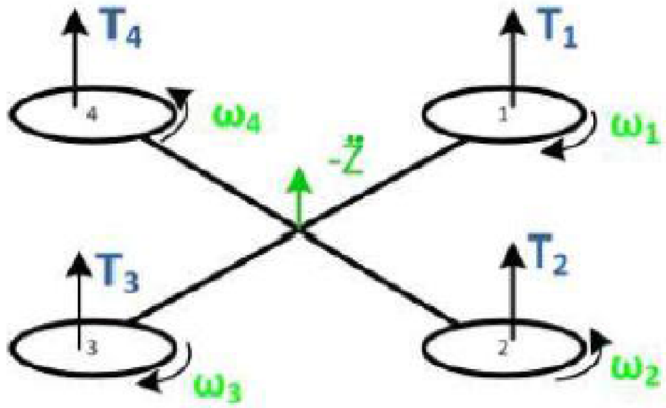


Figure 7: Altitude control

Introduction

- Fonctionnement en « + » :
  - Rotation en tangage / roulis => translation en X / Y
  - Rotation en lacet

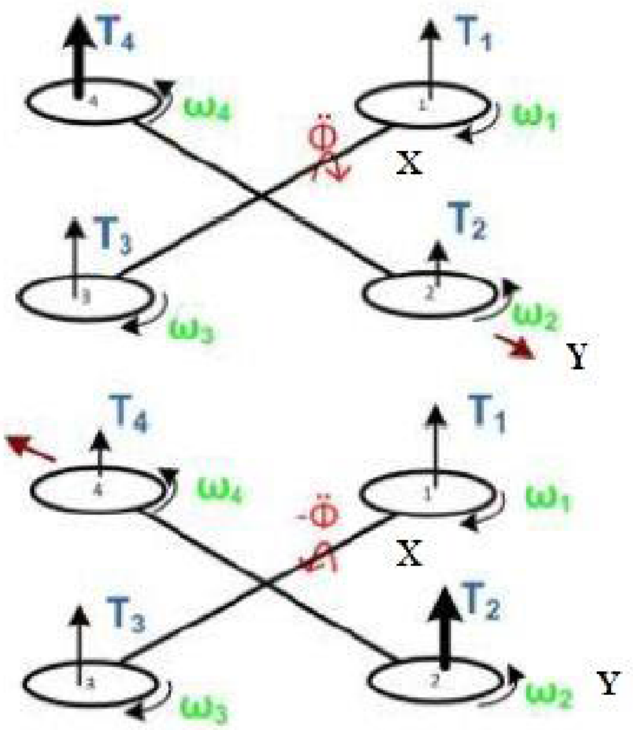


Figure 8: Roll

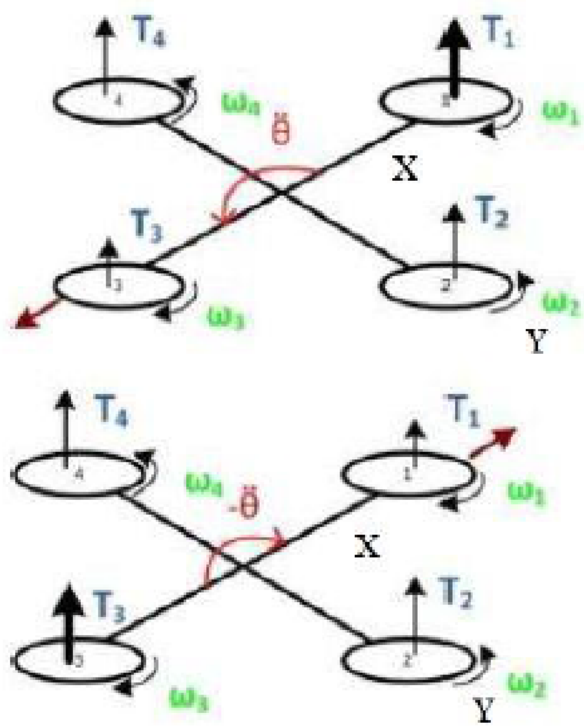


Figure 9: Pitch

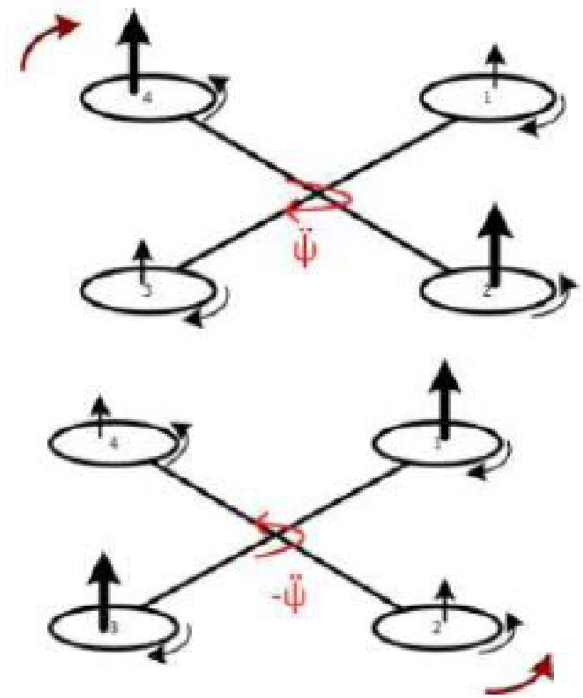
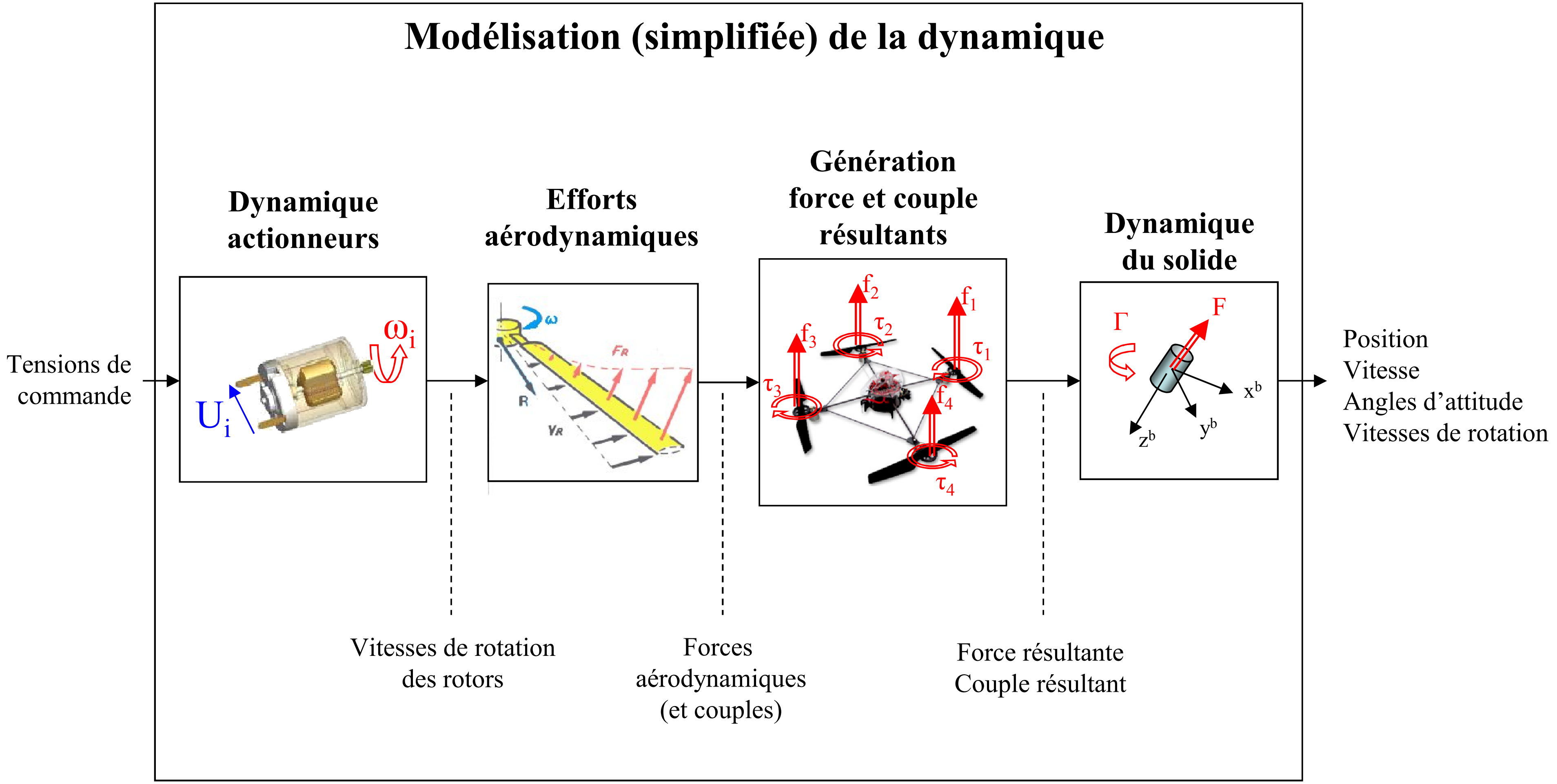


Figure 10: Yaw

Application au drone

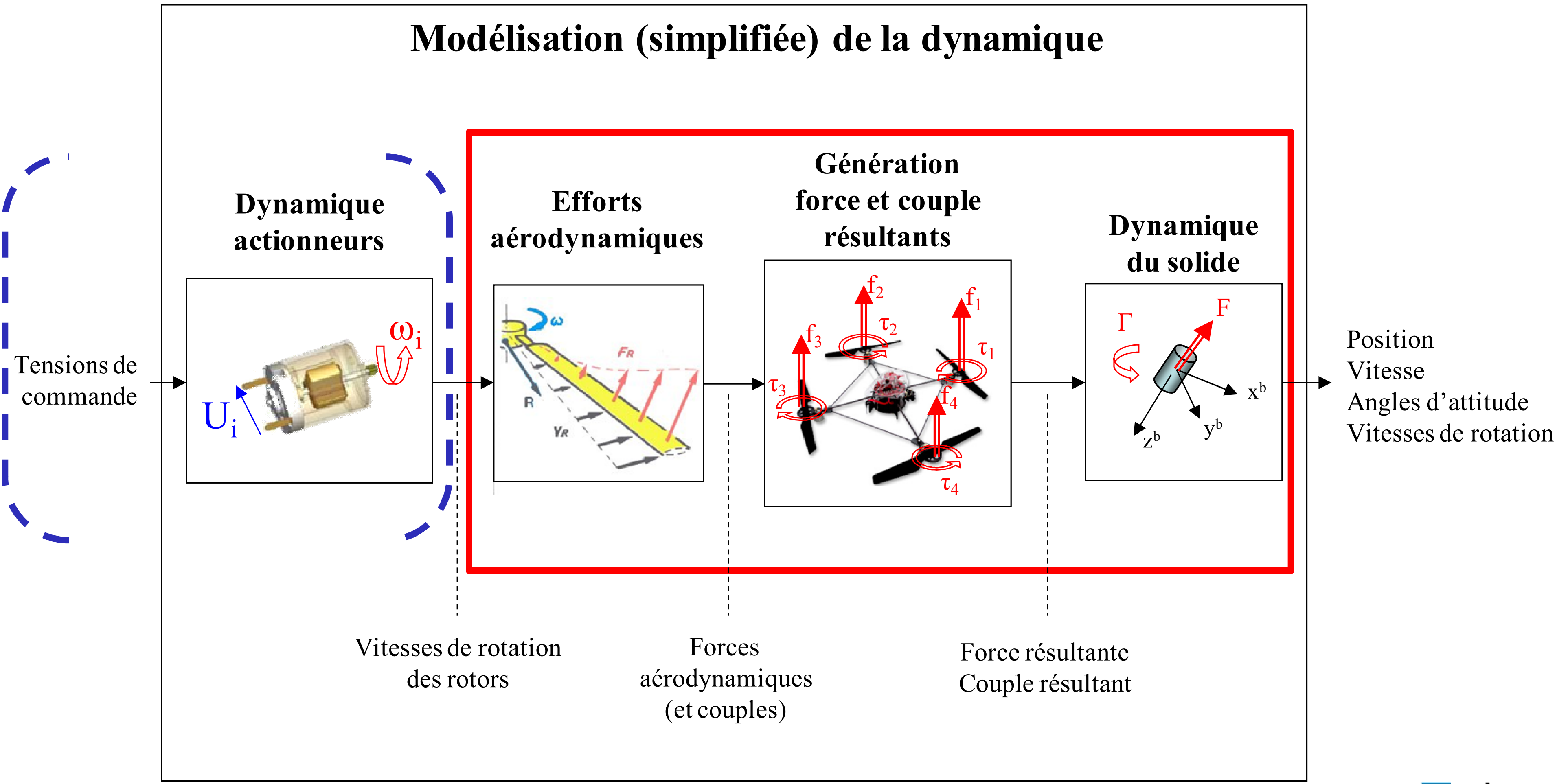
- Introduction
- Modélisation
- Commande
- Estimation

Les étapes de modélisation

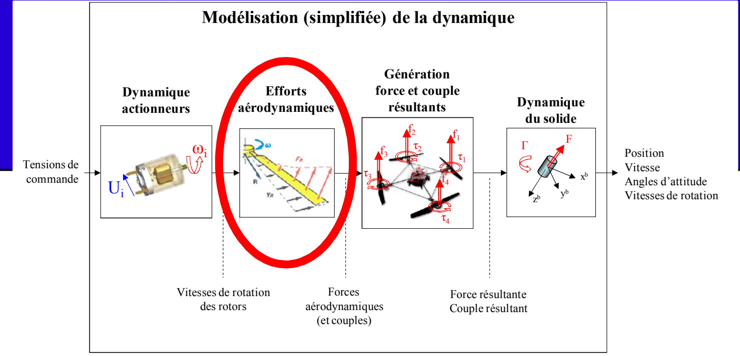


Modélisation
Moteurs

- On supposera l'asservissement moteur réalisé et on négligera sa constante de temps



Modélisation
Efforts aérodynamiques

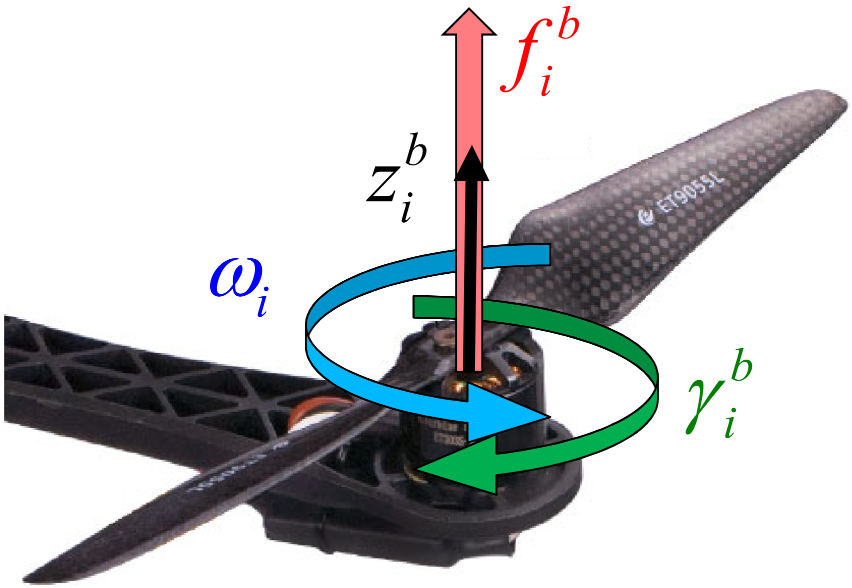


- Force générée par un rotor : $f_i^b = b.\omega_i^2.z_i^b$

- Couple résistant généré par un rotor :

$$\gamma_i^b = \pm d.\omega_i^2.z_i^b$$

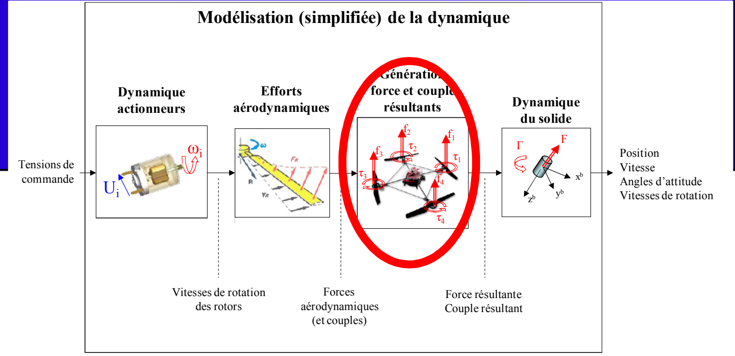
Selon le sens de rotation du rotor



ω_i : vitesse de rotation rotor *i*

b, d : constantes aérodynamiques

Force et couple résultants



- Force résultante

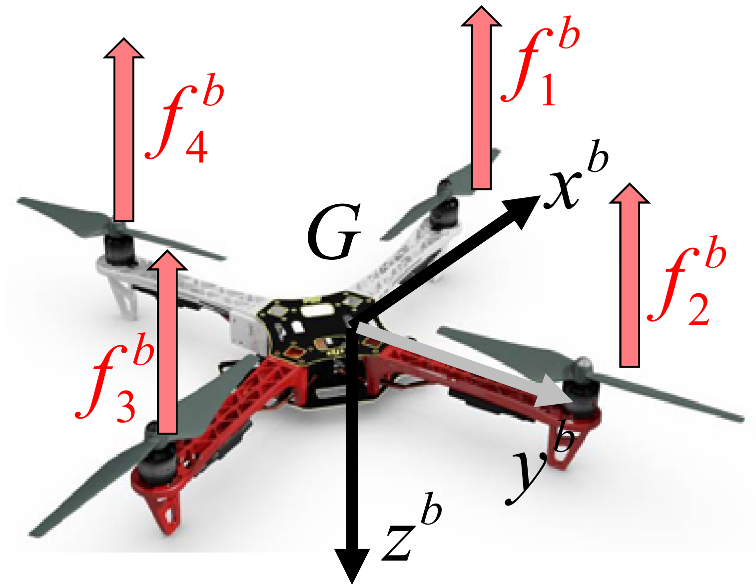
$$F^b = \sum_{i=1}^{4} f_i^b = \sum_{i=1}^{4} -b.\omega_i^2.z^b$$

- Couple résultant

$$\Gamma^b = \begin{bmatrix} l.b.\left(\omega_2^2 - \omega_4^2\right) \\ l.b.\left(-\omega_1^2 + \omega_3^2\right) \\ d.\left(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2\right) \end{bmatrix}$$

couples résistants

application des f_i^b avec un bras de levier

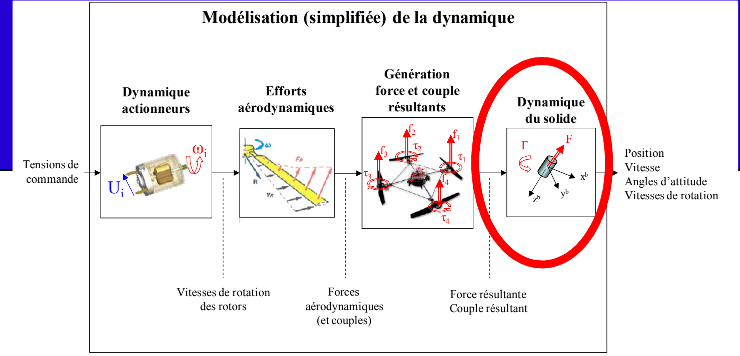


l: longueur d'un bras

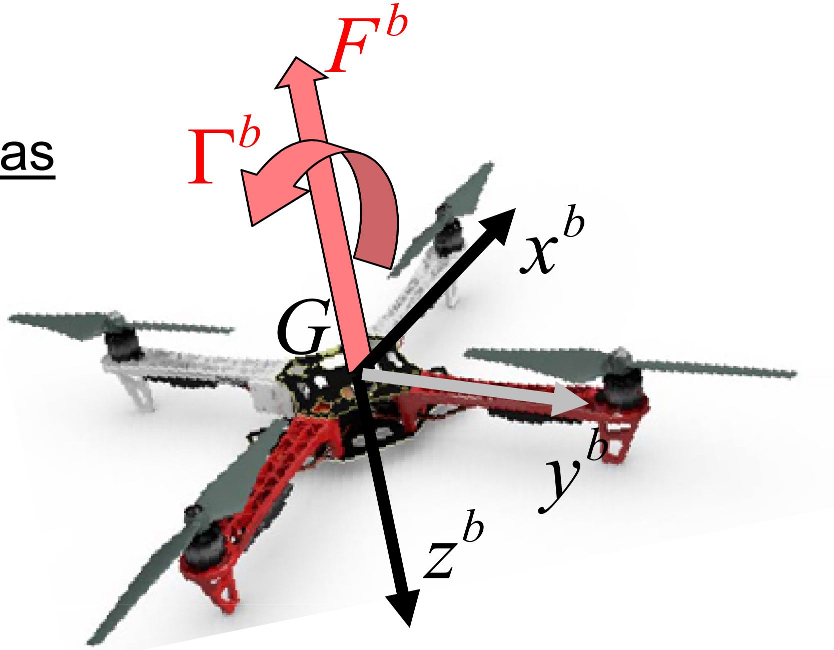
- Repère drone: (G, x^b, y^b, z^b) : <u>convention aéronautique : z^b vers le bas</u>

  convention robotique : z^b vers le haut

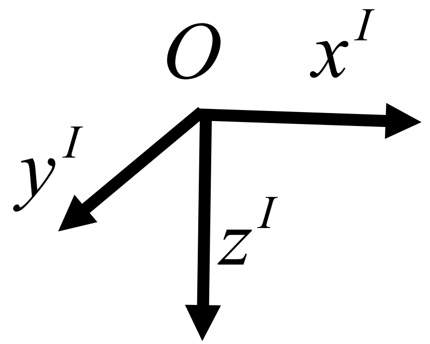
Modélisation
Mécanique du solide



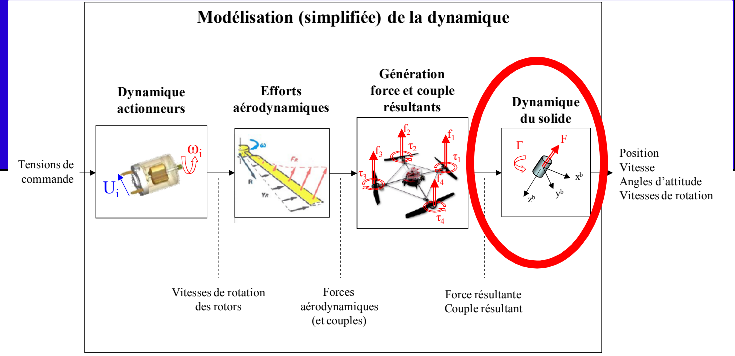
- Repère drone $R^b = (G, x^b, y^b, z^b)$
- Repère fixe $R^I = (O, x^I, y^I, z^I)$
- <u>!! convention aéronautique : z vers le bas</u>
- !! convention robotique : z vers le haut



- Paramétrisation du mouvement :
  - Position
  - Vitesse
  - Orientation
  - Vitesse de rotation



Mécanique du solide



- Orientation du véhicule
  - Angles d'Euler
    - Lacet : ψ, tangage : θ, roulis : φ
  - Matrice de rotation

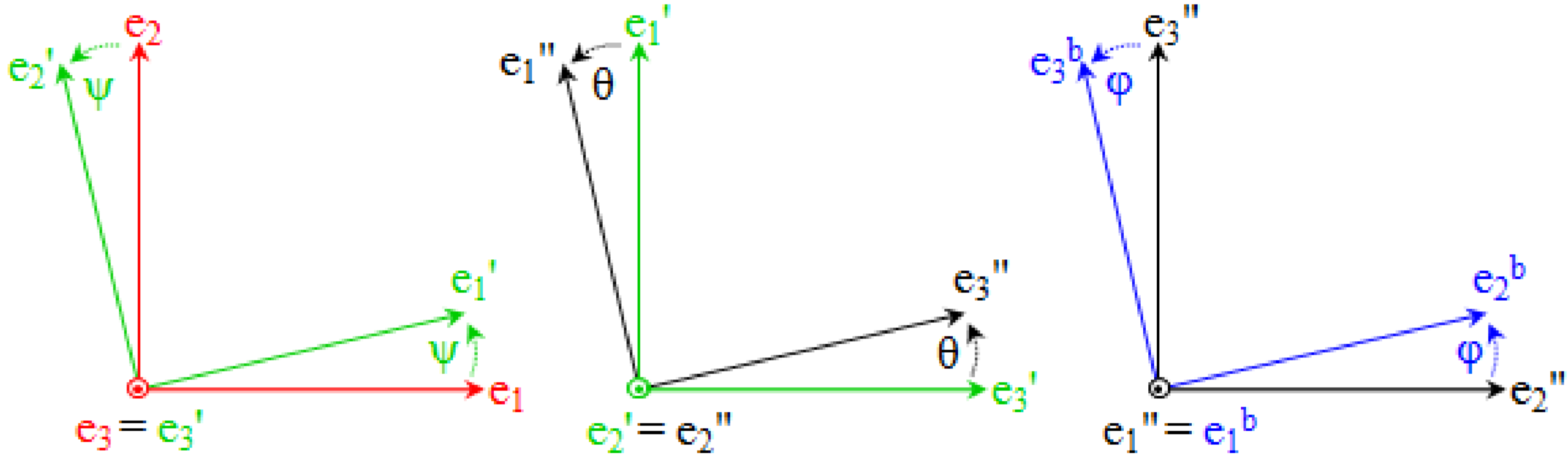
$$R = \quad R_{B \to E} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}$$

  - Quaternion

$$Q = \begin{bmatrix} q_0 \\ \hline q_x \\ q_y \\ q_z \end{bmatrix}, q_0 \in \mathbb{R}, q \in \mathbb{R}^3$$

Représentations de l'orientation

- angles d'Euler
  - Lacet : ψ
  - Tangage : θ
  - Roulis : φ



Représentations de l'orientation

- Matrice de rotation
  - Lacet :
  - Tangage :
  - Roulis :

$$R(z,\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R(y,\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R(x,\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

$$R_{E\to B} = R(x,\phi)\, R(y,\theta)\, R(z,\psi)$$

$$[v]_E = R_{B\to E}\,[v]_B$$

$$R = R_{B\to E} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}$$

- Propriété : $R^{-1} = R^T = R_{E\to B}$

$$\det(R) = 1$$

Représentations de l'orientation

- Quaternion unitaire

$$Q = \begin{bmatrix} q_0 \\ \hline q_x \\ q_y \\ q_z \end{bmatrix}, q_0 \in \mathbb{R}, q \in \mathbb{R}^3 \qquad q_0^2 + q^T q = 1$$

Pour une rotation d'angle γ autour d'un axe défini par le vecteur unitaire k :

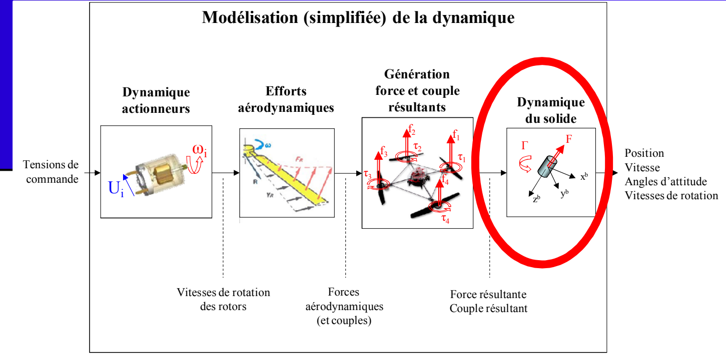
$$q = \hat{k} \sin(\frac{\gamma}{2}), \quad q_0 = \cos(\frac{\gamma}{2})$$

transformations.py

```
from tf.transformations import euler_from_quaternion

euler_from_quaternion(quaternion, axes='sxyz'):
```

Mécanique du solide



- Dynamique de translation :

- Relation cinématique (dans R^I)

$$\dot{\xi} = v$$

- Principe fondamental de la dynamique (dans R^I)

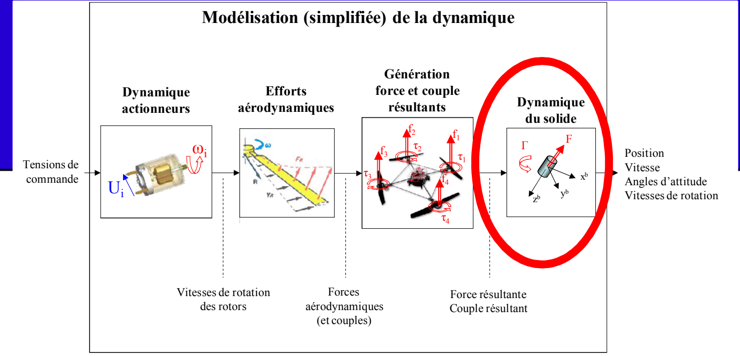
$$m.\dot{v} = m.g.z^b + R.F^b$$

$$\begin{cases} \dot{\xi} = v \\ \dot{v} = g.z^b - \dfrac{T}{m} R.z^b \end{cases}$$

m : masse du drone

$$T = \left\| F^b \right\| = b \sum_{i=1}^{4} \omega_i^2$$

Mécanique du solide



- Dynamique de rotation

- Relation cinématique (dans R^b)

$$\dot{R} = R\Omega_{\times} \qquad \text{avec} \qquad \Omega_{\times} = \begin{bmatrix} 0 & -\omega_r & \omega_q \\ \omega_r & 0 & -\omega_p \\ -\omega_q & \omega_p & 0 \end{bmatrix}$$

Remarque :

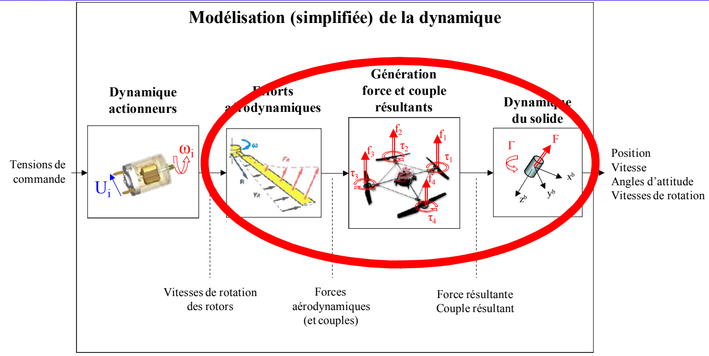
- Il existe des relations cinématiques pour les représentations en angles d'Euler et en quaternion

- Théorème des moments (dans R^b)

$$J\dot{\Omega} = -\Omega \times J\Omega + \Gamma^b$$

J : matrice d'inertie du drone

Modélisation
Dynamique complète



$$\begin{cases} \dot{\xi} = v \\ \dot{v} = g.z^b - \dfrac{T}{m} R.z^b \end{cases}$$

$$\begin{cases} \dot{R} = R\Omega_\times \\ J\dot{\Omega} = -\Omega \times J\Omega + \Gamma^b \end{cases}$$

Couplage entre
les deux dynamiques
par l'orientation R

Entrées de commande

$$\begin{bmatrix} T \\ \Gamma_1^b \\ \Gamma_2^b \\ \Gamma_3^b \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -l.b & 0 & l.b \\ l.b & 0 & -l.b & 0 \\ d & -d & d & -d \end{bmatrix} \cdot \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

Représentation d'état

- Etat : $X = \begin{bmatrix} \xi \\ v \\ \eta \\ \Omega \end{bmatrix} \in \mathbb{R}^{12}$ avec $\eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$

- Commande : $U = \begin{bmatrix} \mathcal{T} \\ \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \end{bmatrix} \in \mathbb{R}^4$ $\dot{X} = f(X, U)$

$$U \quad = \quad A \quad * \quad U'$$

$$\begin{bmatrix} \mathcal{T} \\ \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -l.b & 0 & l.b \\ l.b & 0 & -l.b & 0 \\ d & -d & d & -d \end{bmatrix} . \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

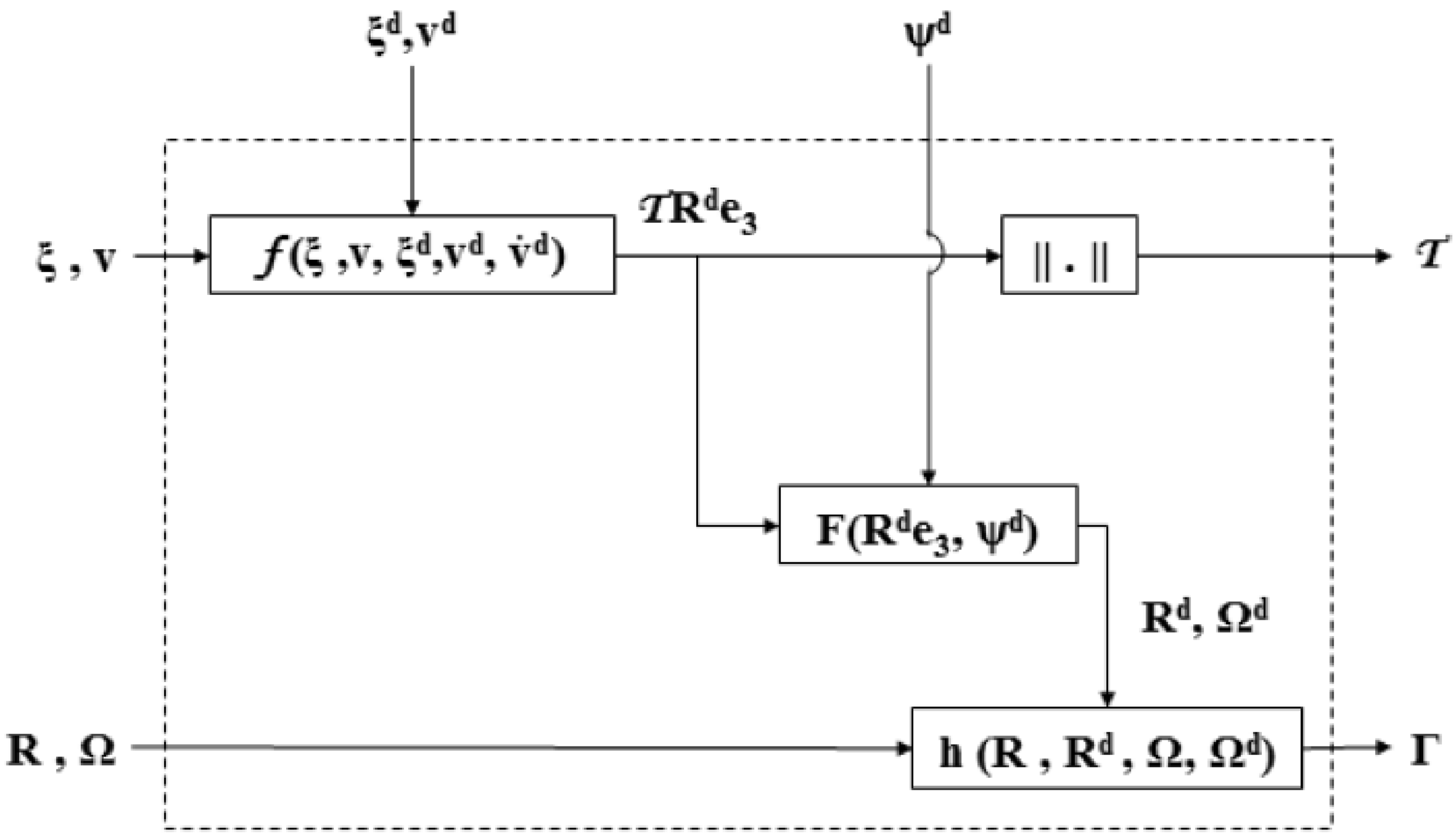
Application au drone

- Introduction
- Modélisation
- Commande
- Estimation

Commande

- Sorties commandables :
  - position x, y, z, angle de lacet ψ
- Variables internes :
  - angles de tangage θ et de roulis φ
- Commande hiérarchique (cf. robot 2D)
  - Commande en position
    - Calculer T et R^r pour que $x \to x^r$, $y \to y^r$, $z \to z^r$
  - Commande en orientation
    - Calculer Γ pour que $R \to R^r$
  - Réglage des gains pour que la dynamique de rotation converge <u>beaucoup</u> plus rapidement que la dynamique de translation

Commande



Commande en position

- Dynamique de translation :

$$\dot{\xi} = v$$
$$\dot{v} = g_0 z_I - \frac{\mathcal{T}}{m} R z_I = u$$

- Commande type proportionnelle dérivée :

$$u = -k_p(\xi - \xi^r) - k_d(v - v^r) \qquad k_p > 0\ ,\ k_v > 0$$

pour stabilisation sur une position fixe : $\xi^r = \text{cste},\ v^r = 0$

$$\mathcal{T} R z_I = m\left(g_0 z_I + k_x(\xi - \xi^r) + k_v(v - v^r)\right)$$

$$\mathcal{T} = \|\mathcal{T} R z_I\|$$

$$R z_I = \frac{\mathcal{T} R z_I}{\|\mathcal{T} R z_I\|} \xrightarrow[\Psi^r \text{ donné}]{} \theta^r\ \varphi^r$$

Commande en orientation

- Dynamique de rotation $\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = R_r^{-1}\Omega$ $\quad R_r = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}$

$$J\dot{\Omega} = -\Omega \times J\Omega + \Gamma$$

- Hypothèse : vol quasi stationnaire => linéarisation autour des petits angles et faibles vitesses angulaires

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \approx \Omega \qquad J\dot{\Omega} \approx \Gamma$$

$$J_x\ddot{\phi} = \Gamma_1$$
$$J_y\ddot{\theta} = \Gamma_2$$
$$J_z\ddot{\psi} = \Gamma_3$$

- Commande type proportionnelle dérivée

$$\Gamma_1 = -k_p(\phi - \phi^r) - k_d(\Omega_\phi - \underbrace{\Omega_\phi^r}_{})$$

etc.

= 0 (stabilisation à un angle cst)

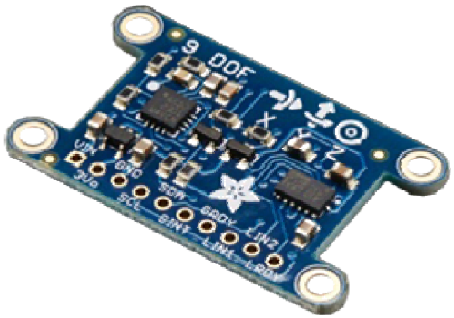
Illustration

Application au drone

- Introduction
- Modélisation
- Commande
- Estimation

Estimation
Capteurs embarqués

- Centrale inertielle (accéléromètre, gyromètre et magnétomètre 3 axes)
- Baro-altimètre
- GPS
- Télémètre(s) : ultrason, lidar, etc…
- Caméra(s)



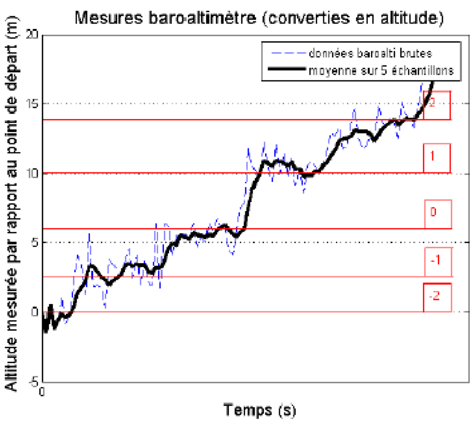


FIGURE 29: Mesure d'altitude donnée par le baroaltimètre (essai dynamique)

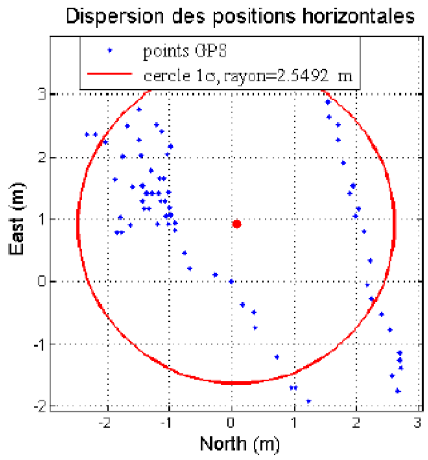


FIGURE 31: Position dans le plan horizontal donnée par le GPS (essai statique)

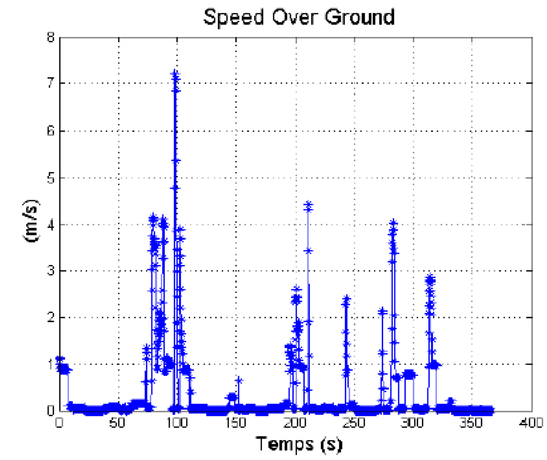
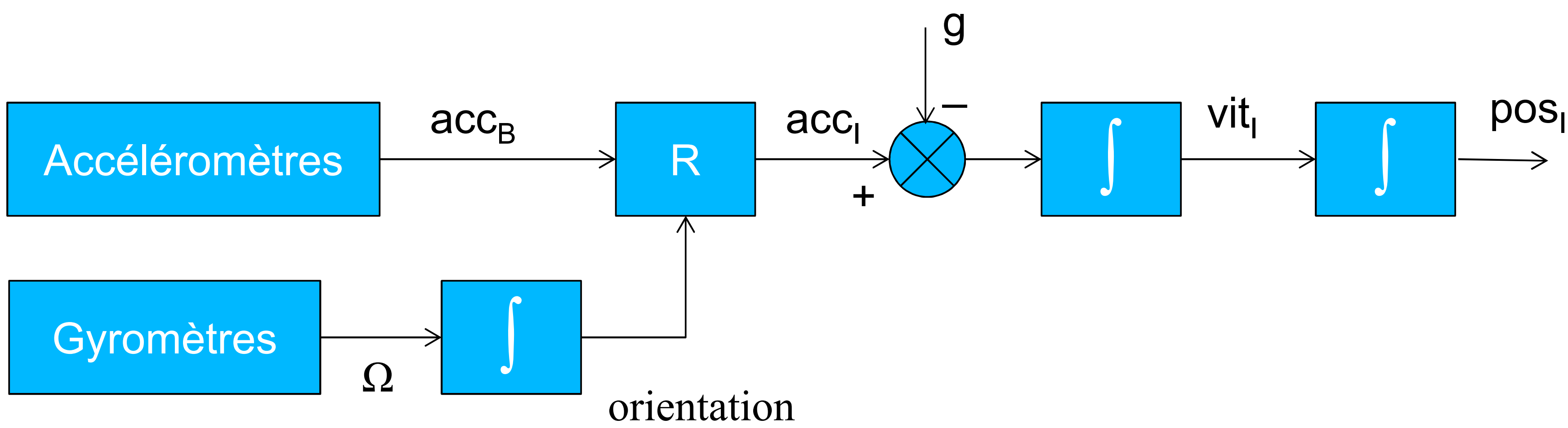


FIGURE 32: Norme de la vitesse dans le plan horizontal (essai statique)

Estimation
Navigation inertielle

- Principe de base : estimer l'état du véhicule à partir des capteurs de l'IMU



- Points critiques :
  - estimation de l'orientation R
  - capteurs imparfaits : biais, bruit, facteur d'échelle

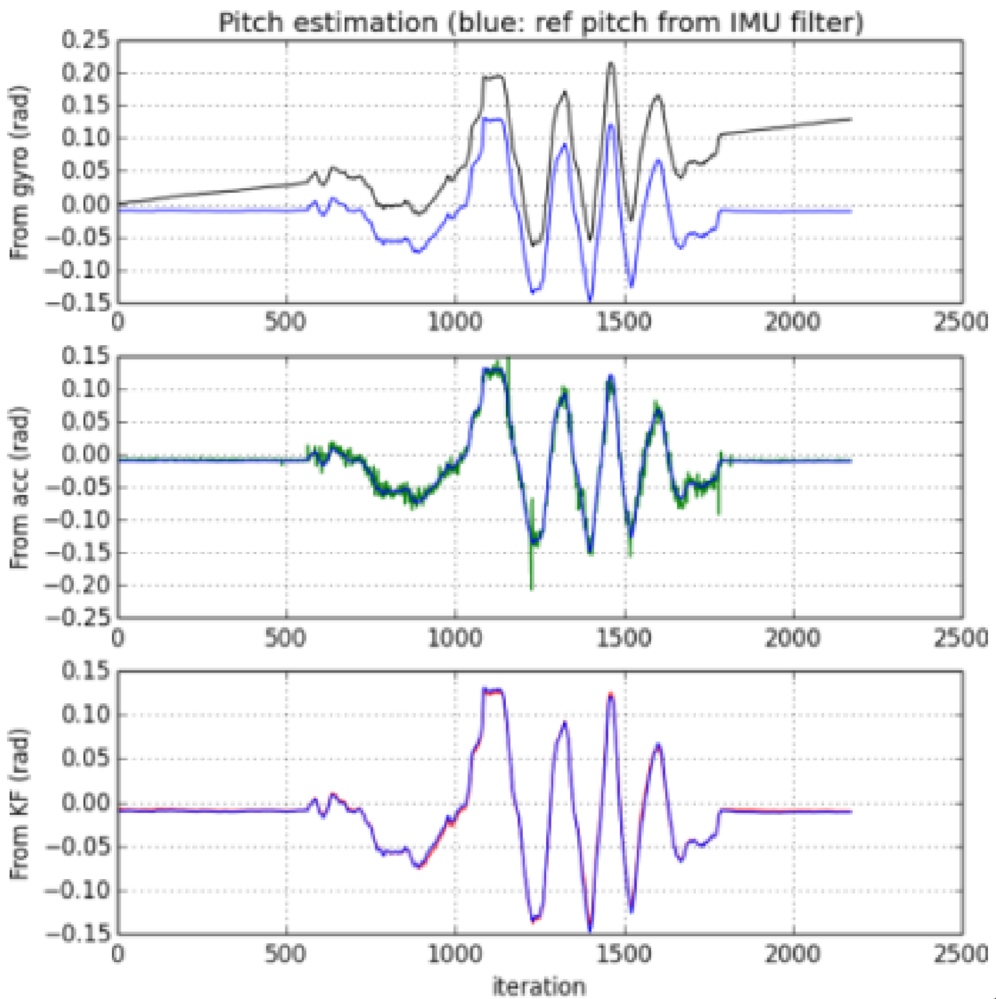
- Modélisation des capteurs

  mesuré = (facteur d'échelle)*vrai + biais + bruit

- Estimation d'attitude à partir des mesures IMU : (cf. TD)

- Fusion gyro/acc/mag

  Deux directions inertielles (acc, mag) utilisées pour fusion avec gyro



- Estimation de la position

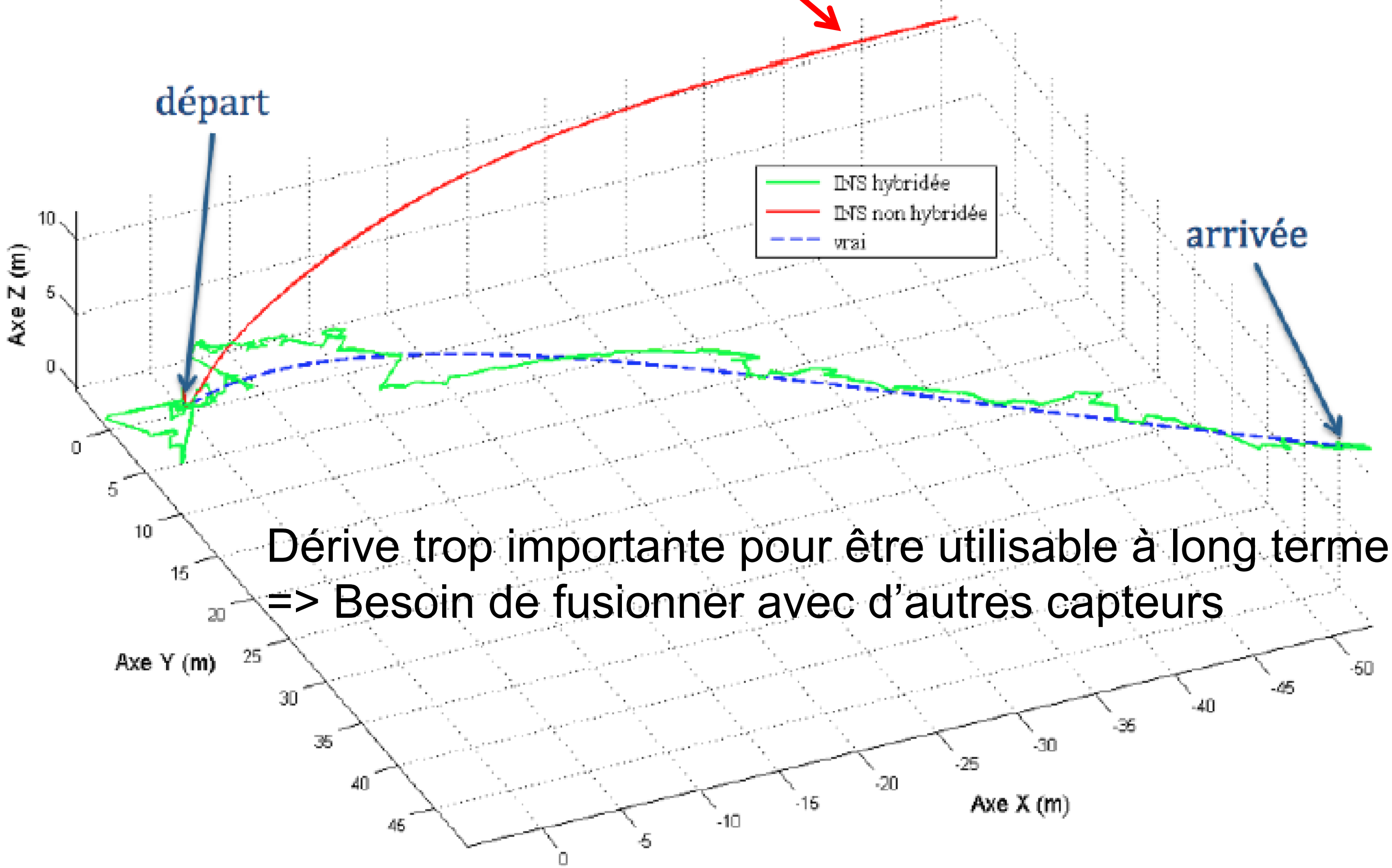


FIGURE 55: Trajectoire dans l'espace avec et sans hybridation

Fusion IMU + mesure de position

- Utilisation d’un filtre de Kalman
  - Prédiction à partir des mesures accélérométriques (entrée)
  - Correction à partir de mesures de position
    - ex : GPS, vision, baro-altimètre, télémètre

- Hypothèse :
  - orientation correctement estimée
    -> utilisée pour transformer les mesures accélérométriques en repère inertiel

Estimation
Fusion IMU + mesure de position

- Modèle de prédiction : $\left\{\begin{array}{l} \dot{p}_I = v_I \\ \dot{v}_I = a_I \end{array}\right.$

Entrée du modèle :

H $\quad a_I^m = a_I + b_I' + n_1'$

$a_I = a_I^m + b_I + n_1$

$(a_I^m = R.a_B^m - g)$

Mesure accéléro

b_I = cste

$$\Rightarrow \left\{\begin{array}{l} \dot{p}_I = v_I \\ \dot{v}_I = a_I^m + b_I + n_1 \\ \dot{b}_I = n_2 \end{array}\right.$$

- Mesure de position : $p_I^m = p_I + n_3$

Fusion IMU + mesure de position

- Mise en équation pour application du KF :
  - Discrétisation du modèle de prédiction

$$\begin{cases} p_I(k+1) = p_I(k) + T_e.v_I \\ v_I(k+1) = v_I(k) + Te.a_I^m(k) + T_e.b_I(k) + T_e.n_1(k) \\ b_I(k+1) = b_I(k) + Te.n_2(k) \end{cases}$$

  - Représentation d'état : $X = \begin{bmatrix} p_I \\ v_I \\ b_I \end{bmatrix}$ $U = [a_I^m]$ $Y = [p_I^m]$

$$X(k+1) = \begin{bmatrix} 1 & T_e & 0 \\ 0 & 1 & T_e \\ 0 & 0 & 1 \end{bmatrix} X(k) + \begin{bmatrix} 0 \\ T_e \\ 0 \end{bmatrix} U(k) + W(k)$$

$$Y(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} X(k) + V(k)$$

Conclusion

On sait donc :

- modéliser la dynamique d'un drone miniature (type quadrirotor)
  - plusieurs représentation de l'orientation

- calculer les commandes en position et en orientation pour stabiliser le drone à un point donné

- fusionner les mesures des capteurs à bord du véhicule pour estimer son orientation (cf. TD), sa position et sa vitesse