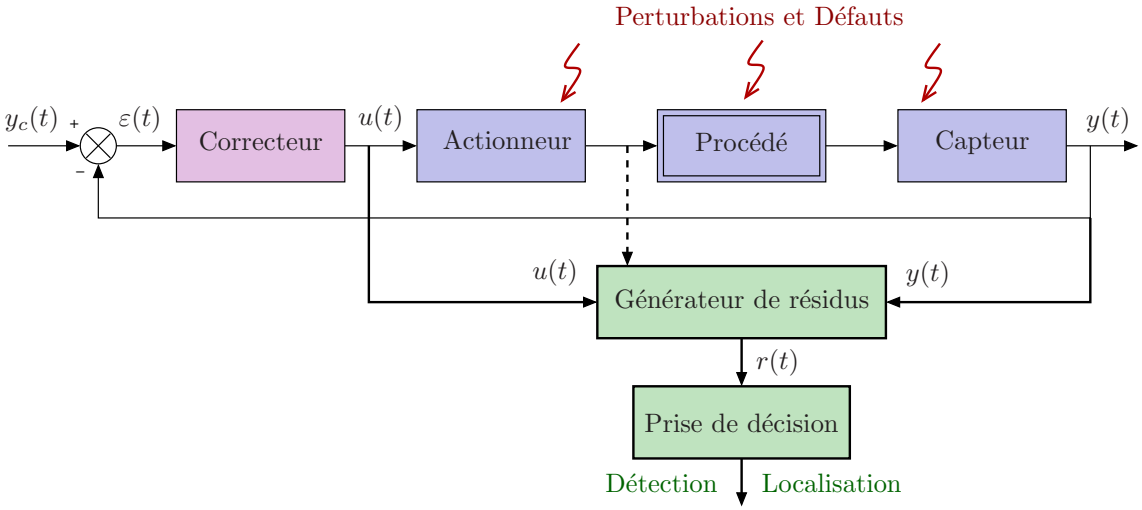
evering INGÉNIERIE AÉRONAUTIQUE / université de BORDEAUX

MASTER 2 MAINTENANCE AÉRONAUTIQUE - SPÉCIALITÉ MAA

Travaux Pratiques
Détection et Localisation de Défauts
Commande Multivariable Avancée

(4TNV906U)



2025-2026

Christophe Farges

http://evering.u-bordeaux.fr

1 Introduction

Cette étude porte sur un drone de type quadricoptère tel que celui représenté sur la figure 1. La synthèse de la loi de pilotage avec la méthode $\mathcal{H}_\infty$ sera réalisée au prochain semestre dans le cadre de l'UE *4TNV001U Automatique avancée pour les systèmes aéronautiques*. L'objectif de ce TP est de déterminer et d'implanter en simulation un module de détection et localisation de pannes avec la méthode de l'espace de parité vue dans le cours.

FIGURE 1 – Drone de type quadricoptère

2 Modélisation

Le modèle dynamique du drone est obtenu par application du principe fondamental de la dynamique en translation et en rotation. Deux repères, décrits dans la partie suivante, seront utilisés pour exprimer les forces et moments appliqués au drone. Le modèle non linéaire obtenu est implanté dans SIMULINK.

2.1 Repères utilisés

Les deux repères utilisés pour exprimer les forces et moments sont le repère inertiel $\left(\mathcal{I}, x^{\mathcal{I}}, y^{\mathcal{I}}, z^{\mathcal{I}}\right)$ (Terre, NED – North East Down) et dans le repère non inertiel $\left(\mathcal{B}, x^{\mathcal{B}}, y^{\mathcal{B}}, z^{\mathcal{B}}\right)$ (body). Ces deux repères sont représentés sur la figure 2.

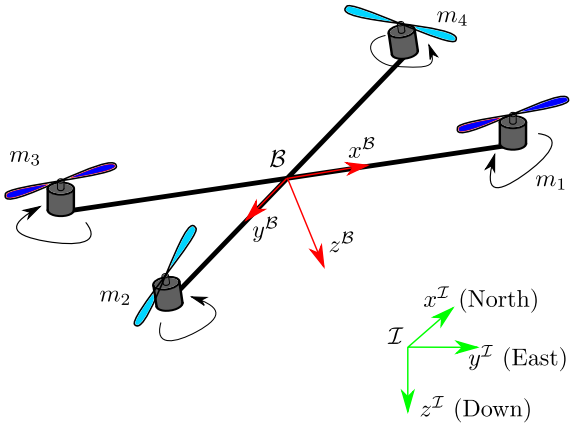


FIGURE 2 – Sens de rotation des hélices et repères inertiel / body

Le passage du repère inertiel vers le repère body est obtenue par la rotation successive d'angles ψ, θ et ϕ (appelés angles d'Euler) comme indiqué sur la figure 3

Un vecteur X exprimé dans le repère body pourra s'écrire dans le repère inertiel comme suit :

$$X^{\mathcal{I}} = R_{\mathcal{B}}^{\mathcal{I}} X^{\mathcal{B}} \tag{1}$$

où $R_{\mathcal{B}}^{\mathcal{I}}$ est la matrice de passage du repère body au repère inertiel s'écrivant :

$$R_{\mathcal{B}}^{\mathcal{I}} = R_z(\psi) R_y(\theta) R_x(\phi) \tag{2}$$

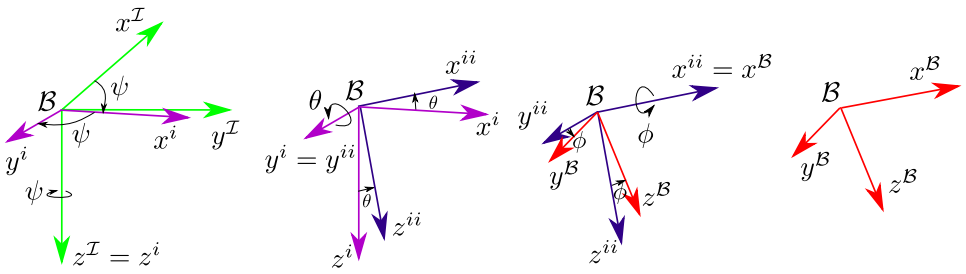


FIGURE 3 – Passage du repère inertiel au repère body

avec :

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{3}$$

2.2 Principe fondamental de la dynamique en translation

Les principales forces s'appliquant sur le drone sont la gravité $\vec{G}$ ainsi que la poussée de chaque moteur m_i notée $\vec{F}_i$. Les forces aérodynamiques sont ici négligées car elles sont faibles dans le cas des multicoptères. Ainsi, le principe fondamental de la dynamique en translation dans un repère inertiel $\mathcal{I}$ s'écrit ici :

$$m\vec{a}^{\mathcal{I}} = \vec{G}^{\mathcal{I}} + \sum_{i=1}^{4} \vec{F}_i^{\mathcal{I}} = \vec{G}^{\mathcal{I}} + R_{\mathcal{B}}^{\mathcal{I}} \sum_{i=1}^{4} \vec{F}_i^{\mathcal{B}} \tag{4}$$

où $\vec{a}^{\mathcal{I}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}$ représente l'accélération du centre de gravité du drone exprimée dans le repère inertiel (x, y et z étant la position du drone dans le repère inertiel).

La gravité s'exprime de la façon suivante dans le repère inertiel :

$$\vec{G}^{\mathcal{I}} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{5}$$

La poussée de chaque moteur dans le repère body est donnée par :

$$\vec{F}_i^{\mathcal{B}} = \begin{bmatrix} 0 \\ 0 \\ -k\omega_i^2 \end{bmatrix}, \ \forall i \in 1, \dots \ 4 \tag{6}$$

où k est une constante et ω_i représente la vitesse de rotation du moteur i.

En notant $\begin{bmatrix} x & y & z \end{bmatrix}^T$ la position du drone dans le repère inertiel, et en remplaçant les forces données par les équations (5) et (6) dans le principe fondamental (4) ainsi que la matrice $R_{\mathcal{B}}^{\mathcal{I}}$ par son expression (2), il vient :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + R_z(\psi) R_y(\theta) R_x(\phi) \begin{bmatrix} 0 \\ 0 \\ -k \sum_{i=1}^{4} \omega_i^2 \end{bmatrix} \tag{7}$$

où $R_z(\psi)$, $R_y(\theta)$ et $R_x(\phi)$ sont les matrices de rotation donnée par (3).

2.3 Principe fondamental de la dynamique en rotation

Le principe fondamental de la dynamique en rotation dans le repère non inertiel $\mathcal{B}$ s'écrit :

$$I \frac{d\vec{\Omega}}{dt} = -\vec{\Omega} \wedge I\vec{\Omega} + \vec{M}^{\mathcal{B}} \tag{8}$$

où I représente la matrice d'inertie du drone exprimée dans le repère drone (drone supposé symétrique) :

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \tag{9}$$

Ω est le vecteur vitesse de rotation du drone exprimé dans le repère drone :

$$\Omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{10}$$

et $\vec{M}^{\mathcal{B}}$ est l'ensemble des moments que subit le drone :

$$\vec{M}^{\mathcal{B}} = \begin{bmatrix} lk(\omega_4^2 - \omega_2^2) \\ lk(\omega_1^2 - \omega_3^2) \\ d(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \end{bmatrix} \tag{11}$$

avec l et d des constantes dépendant des caractéristiques des hélices (profil, angle d'attaque et taille). La composante autour de $x_{\mathcal{B}}$ correspond au roulis, celle autour de $y_{\mathcal{B}}$ au mouvement de tangage et celle autour de $z_{\mathcal{B}}$ au lacet.

Les vitesses angulaires p, q et r dans le repère body sont liées aux dérivées des angles d'euler $\dot{\phi}$, $\dot{\theta}$ et $\dot{\psi}$ selon :

$$\begin{aligned} \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} p \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ q \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} \\ &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_x(\phi)^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_x(\phi)^T R_y(\theta)^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \end{aligned} \tag{12}$$

soit :

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{13}$$

L'inversion de la relation (13) conduit à :

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{14}$$

2.4 Modèle d'état non linéaire

L'application du principe fondamental de la dynamique en translation et en rotation détaillé dans les parties 2.2 et 2.3 permet d'obtenir un modèle d'état décrivant la dynamique du drone en position et en attitude.

L'état de ce modèle est choisi comme suit :

$$X = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \phi & \theta & \psi & p & q & r \end{bmatrix}^T \tag{15}$$

Les entrées de commande sont les vitesses de rotation des quatre moteurs :

$$U = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T \tag{16}$$

Les sorties mesurées sont les positions x, y et z ainsi que ψ, soit :

$$Y = \begin{bmatrix} x & y & z & \psi \end{bmatrix}^T \tag{17}$$

Le modèle d'état non linéaire s'écrit ainsi :

$$\begin{cases} \dot{X} = f(X, U) \\ Y = g(X, U) \end{cases} \tag{18}$$

où la fonction non linéaire f est obtenue à partir des équations des parties 2.1, 2.2 et 2.3.

2.5 Valeur numérique des paramètres

Les valeurs numériques des différents paramètres pour le quadricoptère considéré sont indiquées dans le tableau 1.

Paramètre	Valeurs	Unité
g	9.81	$\mathrm{m\,s^{-2}}$
m	0.486	kg
l	0.25	m
d	3.232×10^{-7}	$\mathrm{kg\,m^2}$
k	2.9842×10^{-5}	kg m
I_{xx}	3.83×10^{-3}	$\mathrm{kg\,m^2}$
I_{yy}	3.83×10^{-3}	$\mathrm{kg\,m^2}$
I_{zz}	7.66×10^{-3}	$\mathrm{kg\,m^2}$

TABLE 1 – Valeurs numériques des paramètres

2.6 Simulateur du comportement dynamique du drone

Le comportement dynamique du drone, donné par le modèle d'état non linéaire de la partie 2.4, est codé dans le fichier `quad_simu_lin_etu.slx` fourni sous moodle et correspond au schéma de la figure 4. Ce schéma fait appel à une S-fonction nommée `quad_dyn.m` également fournie sous moodle.

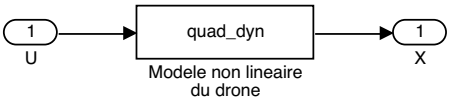


FIGURE 4 – Schéma pour le codage du modèle dynamique et pour la linéarisation

3 Pilotage automatique du quadricoptère

Le pilotage automatique du drone consiste à déterminer les vitesses de rotation des quatre moteurs de sorte à suivre des trajectoires désirées en position et angle de lacet. Les angles de tangages et de lacet sont laissés libres. La loi de commande est ici déterminée à partir d'un modèle linéarisé autour d'un point d'équilibre correspondant au vol stationnaire du quadricoptère.

3.1 Recherche d'un point d'équilibre

Le point d'équilibre considéré correspond au vol stationnaire du drone à une altitude de 5 m. Le vecteur d'état à l'équilibre est donc :

$$\bar{X} = \begin{bmatrix} 0 & 0 & -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{19}$$

Q 1 *Calculer la commande à l'équilibre correspondante, c'est-à-dire $\bar{U}$ tel que :*

$$f(\bar{X}, \bar{U}) = 0 \tag{20}$$

Q 2 *Enregistrer une copie du fichier* `quad_simu_lin_etu.slx` *sous le nom* `quad_simu_equilibre.slx`. *Modifier le schéma de sorte à valider le point d'équilibre déterminé à la question 1 en appliquant dans une première simulation $U(t) = \bar{U}$ puis $U(t) = 0.95 \cdot \bar{U}$ puis $U(t) = 1.05 \cdot \bar{U}$. Commenter les résultats obtenus.*

3.2 Linéarisation

Il est proposé ici d'effectuer la linéarisation autour du point d'équilibre $\{\bar{X}, \bar{U}\}$ de manière numérique à partir du simulateur `quad_simu_lin_etu.slx` réalisé dans la partie 2.6 de sorte à obtenir un modèle d'état linéarisé de la forme :

$$P : \begin{cases} \Delta\dot{X} = \tilde{A}\Delta X + \tilde{B}\Delta U \\ \Delta Y = \tilde{C}\Delta X + \tilde{D}\Delta U \end{cases} \tag{21}$$

avec :

$$\Delta X = X - \bar{X} \qquad \Delta U = U - \bar{U} \qquad \Delta Y = Y - \bar{Y} \tag{22}$$

Q 3 *Réaliser la linéarisation autour du point d'équilibre* $\{\bar{X}, \bar{U}\}$ *grâce à la commande* `linmod` *et stocker le modèle obtenu dans une variable espace d'état* P *via la commande* `ss`.

3.3 Synthèse d'une loi de commande $\mathcal{H}_\infty$

Le modèle linéarisé dans la partie 3.2 permet de déterminer un régulateur $\mathcal{H}_\infty$ selon le schéma de synthèse de la figure 5.

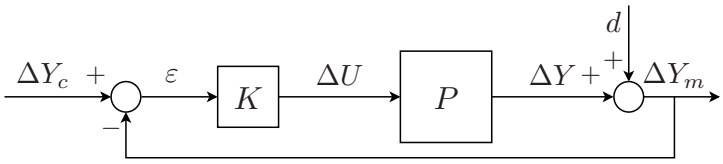


FIGURE 5 – Schéma de commande considéré

Les spécifications sont les suivantes (identiques quelle que soit la mesure considérée : x, y, z ou ψ) :
- temps de montée inférieur à 3 s ;
- erreur statique inférieure à 1 % ;
- marge de module supérieure à 0.5 ;
- rejet de perturbation : les bruits de mesure $d(t)$ de fréquence supérieur à 3000 rad s^{-1} ne doivent pas être amplifiés vis-à-vis du signal de commande, les bruits de mesure $d(t)$ de très haute fréquence doivent être atténués d'un facteur 2, une perturbation basse fréquence $d(t)$ ne doit pas être amplifiée de plus de 50 dB ;

Cette loi de commande, qui sera déterminée au cours du semestre prochain, est fournie pour ce TP et peut être chargée avec l'instruction : `load K.mat`.

3.4 Génération de trajectoire

Le signal de consigne $\Delta Y_d(t)$ contient les positions $\Delta x(t)$, $\Delta y(t)$ et $\Delta z(t)$ désirées au cours du temps (trajectoire à suivre par le drone autour du point d'équilibre) ainsi que l'angle de lacet $\Delta\psi(t)$ désiré autour du point d'équilibre.

Plutôt que de choisir ces consignes sous la forme d'échelons difficiles à suivre pour le drone, des trajectoires plus *lisses* vont être ici générées sous la forme de polynômes. Les coefficients des polynômes sont calculées à l'aide de la fonction `interp_poly5` fournie sous moodle. Les trajectoires seront calculées dans SIMULINK avec le bloc `Generation de trajectoire` également fourni.

3.5 Validation en simulation

La simulation sera réalisée avec le fichier SIMULINK `quad_simu_control_etu.slx` fourni sous moodle correspondant au schéma de la figure 6.

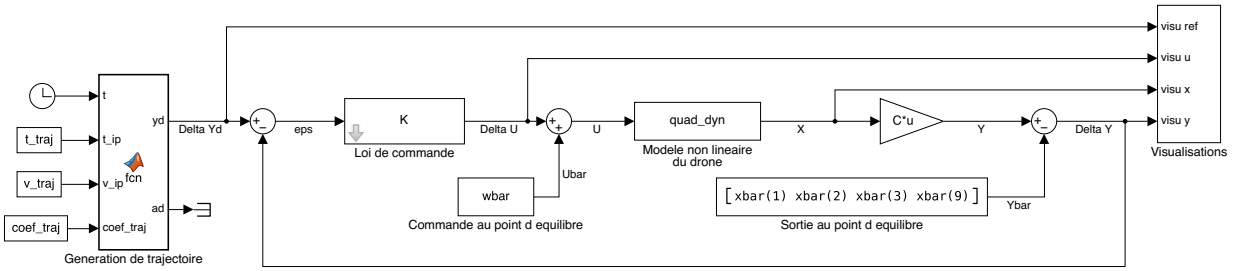


FIGURE 6 – Simulateur pour la commande du drone

Q 4 *Simuler le comportement dynamique du système bouclé et comparez notamment les trajectoires de consigne pour Δx, Δy, Δz et $\Delta\psi$ avec les trajectoires réellement suivies par le drone. Étudiez également le comportement en tangage et roulis réalisé pour suivre ces trajectoires ainsi que les vitesses de rotation des quatre moteurs. Commenter les résultats obtenus.*

4 Détection et localisation de défaillances

L'objectif de cette partie est de mettre en place des algorithmes pour détecter et localiser les défaillances suivantes :

1. perte de poussée des quatres moteurs ;
2. biais sur la mesure d'altitude ;
3. biais sur la mesure de lacet.

4.1 Obtention d'un modèle échantillonné prenant en compte les défauts

Q 5 *Utiliser la fonction* `c2d` *comme suit pour obtenir les matrices de la représentation d'état du modèle échantillonné avec un pas de $T_e = 0.01\,\mathrm{s}$:*

```
Pe=c2d(P,Te,'zoh');
[A,B,C,D]=ssdata(Pe);
```

Q 6 *Déterminer les matrices B_f et D_f de la représentation d'état ci-dessous*

$$\begin{cases} \Delta X(k+1) = A\Delta X(k) + B\Delta U(k) + B_f F(k) \\ \Delta Y(k) = C\Delta X(k) + D\Delta U(k) + D_f F(k) \end{cases} \tag{23}$$

en considérant les 3 défauts cités plus haut.

4.2 Détermination du générateur de résidus par la méthode espace de parité

Q 7 *Est-il possible de générer un résidu insensible à l'état à l'aide de la méthode espace de parité statique ?*

Q 8 *Quelle est la taille minimum de fenêtre pour calculer un résidu par la méthode espace de parité dynamique ?*

Q 9 *Déterminer une matrice de parité W pour cette taille de fenêtre.*

4.3 Validation en simulation

La simulation sera réalisée avec le fichier SIMULINK `quad_simu_diag_control_etu.slx` fourni en TP correspondant au schéma de la figure 7.

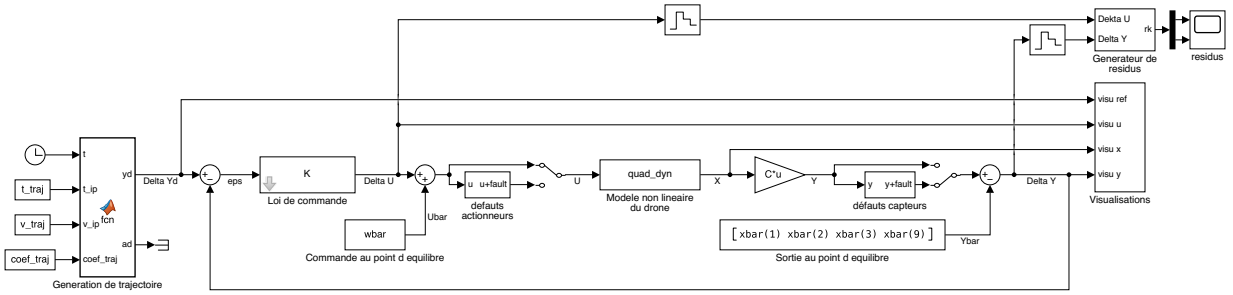


FIGURE 7 – Simulateur pour la commande et le diagnostic du drone

Q 10 *Compléter le bloc* `Generateur de residus` *avec le schéma bloc correspondant au générateur de résidus déterminé dans la partie 4.2.*

Q 11 *Tracer l'évolution temporelle des résidus sans défauts, dans le cas de défauts actionneurs et dans le cas de défauts capteurs. Indiquer les instants d'apparition des différents défauts.*

4.4 Localisation de défaillances

Q 12 *Déterminer la forme d'évaluation des résidus :*

$$r(k) = W\Phi_F F(k-s,k) = W_F F(k-s,k) \tag{24}$$

Q 13 *Déterminer la forme d'évaluation des résidus en régime permanent dans le cas de défauts constants :*

$$r(k) = W_{Fss} f(k) \tag{25}$$

En déduire quels défauts sont fortement détectables et faiblement détectables. Commenter ce résultat vis-à-vis des tracés obtenus à la question 11.

Q 14 *À partir de l'analyse des colonnes de la matrice W_F et à partir des résultats de simulation, localiser chacun des défauts capteurs et déterminer leur amplitude.*

5 Commandes Matlab

`>> m=rank(M)` : calcule le rang m d'une matrice M
`>> W=null(Q')'` : calcule une matrice Q tq $WQ = 0$ (renvoie `Empty matrix` si W n'existe pas)
`>> [V,LAMBDA]=eig(M,N)` : permet de calculer les valeurs propres et vecteurs propres du faisceau de matrices (M, N). Les valeurs propres sont données dans une matrice `LAMBDA` diagonale et les vecteurs propres associés constituent les colonnes de la matrice `V` (vecteurs propres et valeurs sont classés dans le même ordre).