

**Haute Ecole d'Ingénierie et de Gestion  
du canton de Vaud (HEIG-VD)**

**Département des Technologies  
Industrielles (TIN)**

***Filière Microtechniques (MI)***

***Filière Génie électrique (GE)***

***Filière Systèmes industriels (SI)***

***Filière Energie et Techniques  
Environnementales (ETE)***

**HES d'été MATLAB/Octave  
Exercices**



HAUTE ÉCOLE  
D'INGÉNIERIE ET DE GESTION  
DU CANTON DE VAUD  
[www.heig-vd.ch](http://www.heig-vd.ch)

**i**nstitut d'  
**A**utomatisation  
**i**ndustrielle



**Michel ETIQUE**  
maître d'enseignement "B",  
septembre 2016, Yverdon-les-Bains

## Table des matières

<b>1</b>	<b>Traitement des données d'un tableau</b>	<b>4</b>
<b>2</b>	<b>Mise en valeur de résultats expérimentaux</b>	<b>5</b>
2.1	Mesures pratiquées sur un foehn ou "canal aérothermique" . . . . .	5
2.1.1	Affichage de la réponse indicielle . . . . .	5
2.1.2	Valeur finale en régime permanent constant . . . . .	5
2.1.3	Caractéristique statique . . . . .	7
2.2	Débit et température d'un flux d'air . . . . .	8
2.3	Caractéristique d'un ressort . . . . .	9
<b>3</b>	<b>Modélisation et simulation d'un amortisseur</b>	<b>10</b>
3.1	. . . . .	10
3.2	. . . . .	11
<b>4</b>	<b>Modélisation et simulation d'un moteur à courant continu à excitation séparée constante</b>	<b>12</b>
4.1	. . . . .	13
<b>5</b>	<b>Simulation d'un entraînement avec transmission flexible</b>	<b>14</b>
<b>6</b>	<b>Equations</b>	<b>16</b>
6.1	Résolution d'un système d'équations . . . . .	16
6.2	Solution d'une équation de degré supérieur . . . . .	16
<b>7</b>	<b>Décomposition d'un signal carré en série de Fourier</b>	<b>17</b>
<b>A</b>	<b>Calcul de la réponse fréquentielle d'un système dynamique linéaire avec les phaseurs temporels</b>	<b>19</b>
A.1	Obtention directe de la fonction de transfert . . . . .	21
<b>B</b>	<b>Introduction de réponses fréquentielles/fonctions de transfert dans MATLAB/Octave</b>	<b>23</b>
B.1	Calcul et tracé de réponses de systèmes dynamiques linéaires . . .	24
B.1.1	Réponse temporelle . . . . .	24
B.1.2	Réponse fréquentielle . . . . .	28
B.2	Analyse des propriétés des systèmes (unités d'enseignement Régulation automatique (Régul) et Signaux et systèmes (SES)) . . . . .	29
B.3	Calcul affichage des marges de gain et de phase . . . . .	29
<b>C</b>	<b>Représentation d'un système dynamique linéaire par son modèle d'état.</b>	<b>30</b>
C.1	Exemple introductif : circuit RLC série . . . . .	30

C.1.1	Modèle entrée-sortie ("Input-output model") . . . . .	30
C.1.2	Modèle d'état . . . . .	32
C.1.3	Non-unicité de la représentation d'état . . . . .	33
C.2	Définition . . . . .	33
C.2.1	Exemple : moteur DC . . . . .	35
C.3	Forme matricielle . . . . .	36
C.3.1	Exemple : moteur DC . . . . .	37
C.4	Schéma fonctionnel . . . . .	38
C.4.1	Exemple : moteur DC . . . . .	39
C.5	Calcul de la fonction de transfert à partir du modèle d'état . . . .	39
C.5.1	Exemple : moteur DC . . . . .	42

# 1 Traitement des données d'un tableau

Le fichier

[/MATLAB/Canevas/exercice\\_1\\_matrices\\_canevas.m](#)

initialise une matrice **M** (tableau de nombres). Compléter le fichier afin d'obtenir les informations suivantes :

1. la taille de **M** (fonction MATLAB/Octave utile : `size()`);
2. la matrice transposée de **M** (opérateur MATLAB/Octave utile : `'`);
3. la 4<sup>ème</sup> ligne de **M** (opérateur MATLAB/Octave utile : `:`);
4. la 2<sup>ème</sup> colonne de **M**;
5. la sous-matrice constituée des colonnes n° 3 et 4;
6. la sous-matrice constituée des colonnes n° 6 et 2;
7. la sous-matrice constituée des lignes n° 3 et 5;
8. la sous-matrice constituée des éléments communs aux lignes n° 2 et 5 et aux colonnes n° 1 et 6;
9. la matrice constituée des colonnes n° 6 à 1 (opérateur MATLAB/Octave utile : `6:-1:1`);
10. la matrice constituée de **M** augmentée d'une ligne ne contenant que des 1 (fonction MATLAB/Octave utile : `ones()`);
11. la matrice constituée de **M** augmentée d'une colonne supplémentaire ne contenant que des 0 (fonction MATLAB/Octave utile : `zeros()`);
12. la matrice constituée de **M** dans laquelle la 3<sup>ème</sup> ligne est remplacé par des 1;
13. la matrice constituée de **M** dans laquelle la 3<sup>ème</sup> ligne est remplacé par 2<sup>composantes</sup> du 1<sup>er</sup> rang (opérateur MATLAB/Octave utile : `.`);
14. la matrice constituée de **M** dans laquelle une colonne de 0 est insérée après la colonne n° 5;
15. la matrice constituée de **M** dans laquelle la ligne n° 3 est remplacée par cette même ligne dont les composantes sont présentées dans l'ordre inverse (fonction MATLAB/Octave utile : `flip1r()`);
16. l'indice de la composante égale à 16 (fonction MATLAB/Octave utile : `find()`);
17. les composantes de **M** supérieures à 28;
18. la composante de la ligne n° 2 et de la colonne n° 5 de **M** la plus proche de  $e$  (fonctions MATLAB/Octave utiles : `min()`, `abs()`);
19. la composante de **M** la plus proche de  $3 \cdot \pi$ .

## 2 Mise en valeur de résultats expérimentaux

### 2.1 Mesures pratiquées sur un foehn ou "canal aérodynamique"

#### 2.1.1 Affichage de la réponse indicielle

Le fichier

[/Mesures/mesures\\_rep\\_ind\\_foehn.dat](#)

contient les données d'acquisition de la réponse indicielle du foehn (figure 1(a) page suivante, manipulation du laboratoire d'Automatique).

A l'aide de MATLAB/Octave, tracer cette réponse indicielle, sachant que l'organisation des données contenues dans le fichier `rep_ind_foehn.dat` mentionné ci-dessus est faite comme suit :

1<sup>ère</sup> colonne : temps en s  
2<sup>ème</sup> colonne : tension  $u(t)$  en V appliquée aux bornes du corps de chauffe  
3<sup>ème</sup> colonne : tension de sortie  $T_m$  du capteur en V (mesure de température)

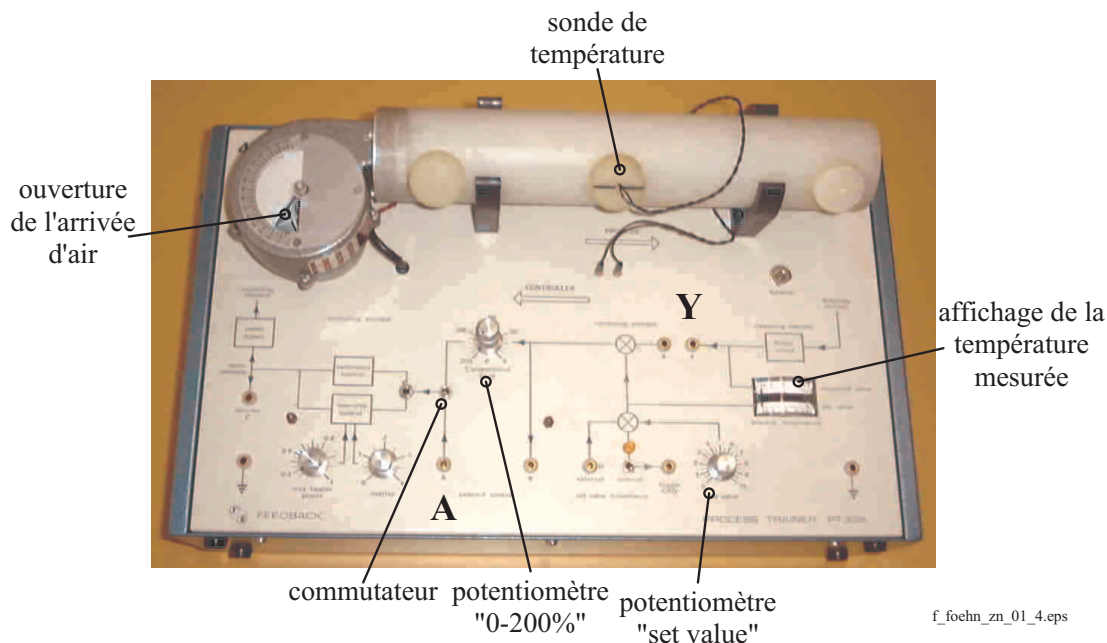
#### Indications :

- Il s'agit en fait de reproduire la figure 1(b) page suivante;
- Fonctions MATLAB/Octave utiles :
  - `load`
  - `plot`, `grid`, `title`, `xlabel`, `ylabel`, `gtext`, `legend`
  - `subplot`
- Le canevas  
[/MATLAB/Canevas/exercice\\_2\\_1\\_1\\_foehn\\_canevas.m](#)  
est à disposition.

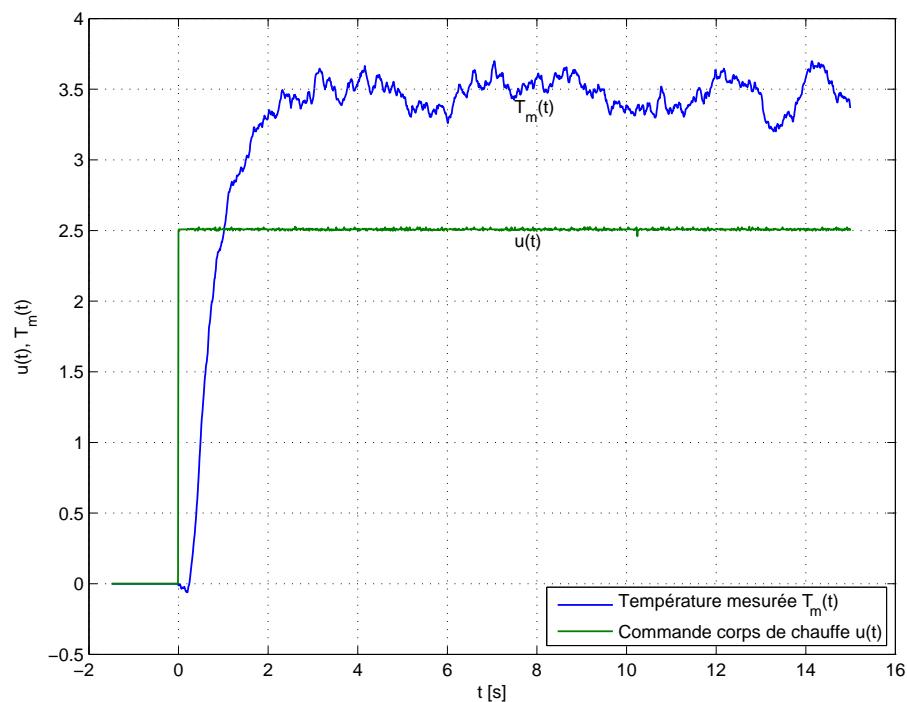
#### 2.1.2 Valeur finale en régime permanent constant

Partant des résultats obtenus au § 2.1.1, déterminer à l'aide de MATLAB/Octave la valeur moyenne  $T_{m\infty} = \lim_{t \rightarrow \infty} T_m(t)$  de la température mesurée  $T_m(t)$  en régime permanent constant.

- Fonctions MATLAB/Octave utiles :
  - `mean`
- Le canevas  
[/MATLAB/Canevas/exercice\\_2\\_1\\_2\\_foehn\\_canevas.m](#)  
est à disposition.



(a) Vue du foehn : le signal d'entrée  $u(t)$  est appliqué au point **A**, celui de sortie  $T_m(t)$  est mesuré au point **Y**.



(b) Réponse indicielle  $T_m(t)$  du foehn, i.e. réponse à un saut de tension  $u(t)$  aux bornes du corps de chauffe.

FIGURE 1 –

### 2.1.3 Caractéristique statique

Le fichier

[/Mesures/caracteristique\\_statique\\_foehn.dat](#)

contient les données d'acquisition de la caractéristique statique du foehn.

A l'aide de MATLAB/Octave, tracer la caractéristique

$$T_m = f(u)$$

sachant que l'organisation des données contenues dans le fichier mentionné ci-dessus `caracteristique_statique_foehn.dat` est faite comme suit :

1<sup>ère</sup> **colonne** : tension d'entrée  $u$  en V (commande du corps de chauffe)  
2<sup>ème</sup> **colonne** : tension de sortie  $T_m$  en V (mesure de température)  
3<sup>ème</sup> **colonne** : température mesurée sur thermomètre  $T$  en °C

Ensuite, par régression linéaire, trouver le gain statique  $K$  du système, i.e.

$$K = \left. \frac{\Delta T_m}{\Delta u} \right|_{\text{régime permanent constant}}$$

Superposer les mesures (symbolisées par exemple par des x) et la droite de régression.

- Fonctions MATLAB/Octave utiles :
  - `load`
  - `plot`, `grid`, `title`, `xlabel`, `ylabel`, `gtext`, `legend`, `num2str`
  - `polyfit`, `polyval`, `poly2str`
- Le canevas

[/MATLAB/Canevas/exercice\\_2\\_1\\_3\\_foehn\\_canevas.m](#)

est à disposition.

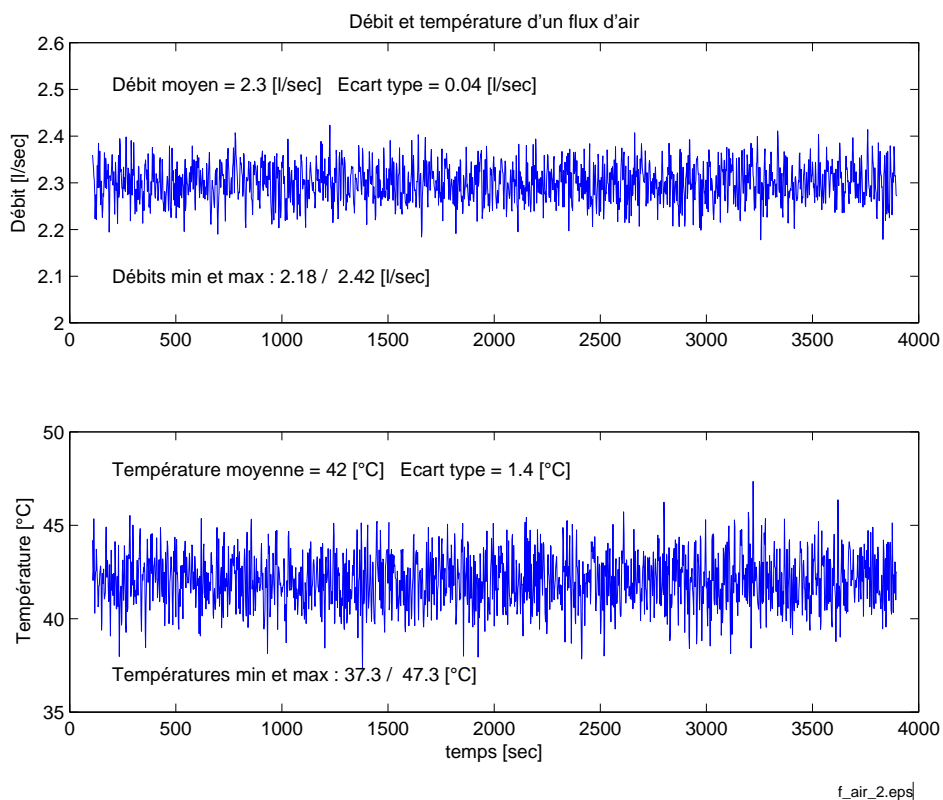


FIGURE 2 – Débit et température d'un flux d'air

## 2.2 Débit et température d'un flux d'air

Dans cet exercice, on se propose de lire des valeurs fournies par un fichier `ascii` et en tirer quelques informations utiles avant de les porter sur un graphe (figure 2). Lors d'une expérience, on a enregistré l'évolution du débit et de la température d'un flux d'air au cours du temps. Les valeurs numériques ont été portées dans un fichier `air.dat` comportant 3 colonnes :

- le temps exprimé en secondes
- le débit donné en litres par seconde
- la température mesurée en degrés Celsius :

---

1.0800000 e+002	2.3597495 e+000	4.4197579 e+001
-----------------	-----------------	-----------------

1.1000000 e+002	2.3446581 e+000	4.2045259 e+001
-----------------	-----------------	-----------------

1.1200000 e+002	2.3312441 e+000	4.2574619 e+001
-----------------	-----------------	-----------------

.....



---

3.8940000e+003    2.2748863e+000    4.2438009e+001

3.8960000e+003    2.2713996e+000    4.2516874e+001

---

Le fichier contenant ces données est disponible ici sous le lien

[/Exercices/Mesures/air.dat](#)

Ecrire un programme MATLAB/Octave reproduisant la figure 2.

## 2.3 Caractéristique d'un ressort

On considère ici un ressort que l'on tend plus ou moins tout en mesurant la force qui lui est appliquée à l'aide d'un dynamomètre dont la précision est d'environ  $500 \cdot 10^{-3}$  N. Les résultats que l'on a obtenus sont les suivants :

Longueur cm	4.2	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0
Force N( $\pm 0.50$ N)	0.0	1.1	2.0	3.2	3.9	4.6	5.8	7.0	8.3	9.0	9.5

Ces données ont été reportées dans le fichier canevas

[/MATLAB/Canevas/exercice\\_2\\_3\\_ressort\\_canevas.m](#)

Dans ce qui suit, on souhaite :

1. Tracer le graphe de la force en fonction de la longueur avec les barres d'erreurs `errorbar`.
2. Mettre en valeur le graphe à l'aide d'un titre et d'informations portées sur l'abscisse et l'ordonnée.
3. Rechercher une loi polynomiale représentant au mieux cette caractéristique ; celle d'un ressort est généralement linéaire, éventuellement cubique.
4. Mesurer la qualité des modèles proposés pour représenter le ressort.
5. Afficher les informations sur le graphe lui-même.

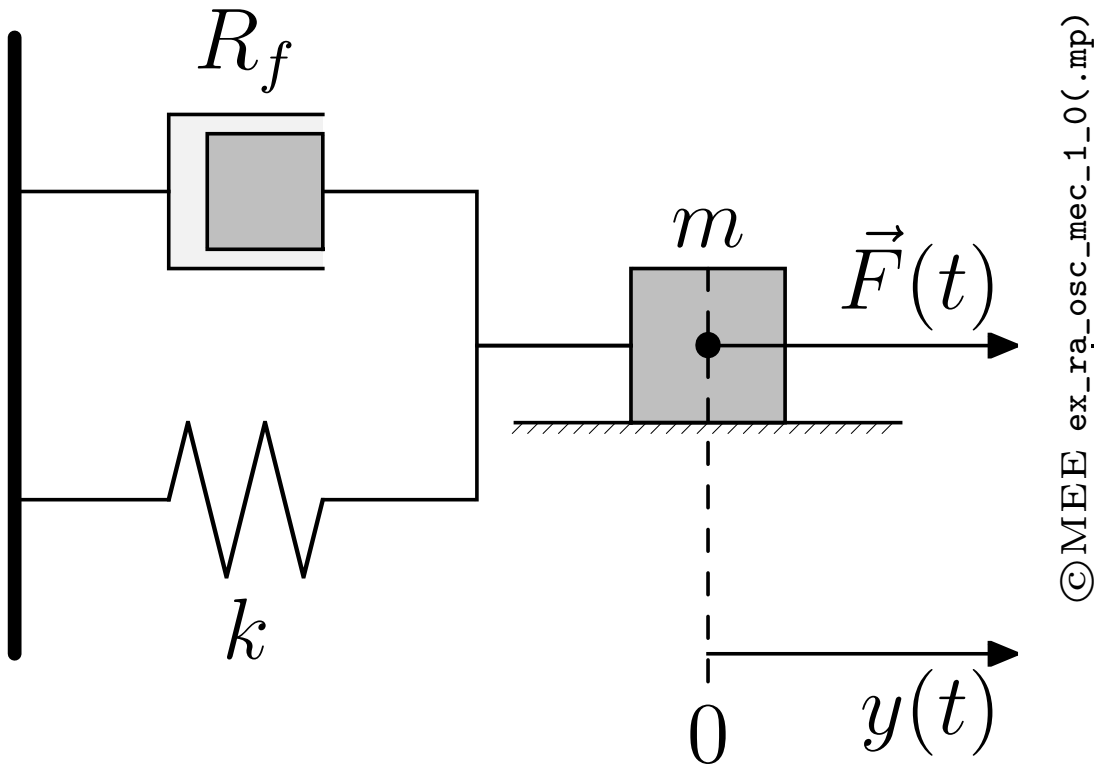


FIGURE 3 – Schéma d'un amortisseur.

### 3 Modélisation et simulation d'un amortisseur

On considère le schéma technologique suivant : La masse  $m$  subit la force externe  $F(t)$  (signal d'entrée) et sa position instantanée est donnée par  $y(t)$  (signal de sortie). Du frottement visqueux linéaire de coefficient  $R_f \frac{\text{N}}{\text{m/s}}$  agit sur  $m$  de même qu'une force de rappel provoquée par le ressort de constante de rigidité  $k \frac{\text{N}}{\text{m}}$ .

Valeurs numériques :  $m = 1 \text{ kg}$ ,  $R_f = 500 \cdot 10^{-3} \frac{\text{N}}{\text{m/s}}$ ,  $k = 10 \frac{\text{N}}{\text{m}}$

#### 3.1

Simuler la réponse  $y(t)$  de ce système à une force  $F(t)$  en forme de saut et estimer la durée du régime transitoire. Puis, exciter le système avec des entrées sinusoïdales pour différentes fréquences voire amplitudes. Observer le régime permanent sinusoïdal.

- Fonctions MATLAB/Octave utiles :
  - ss, tf
  - step, impulse, lsim
  - bode

- linspace
  - logspace
  - Les canevas
    - [/MATLAB/Canevas/amortisseur\\_equa\\_diff\\_canevas.m](#)
    - [/MATLAB/Canevas/exercice\\_3\\_1\\_amortisseur\\_canevas.m](#)
- sont à disposition.

Si le ressort avait une caractéristique statique non-linéaire

$$k = k(y) \neq \text{constante}$$

comme celui de l'exercice 2.3, comment pourrait-on en tenir compte ?

## 3.2

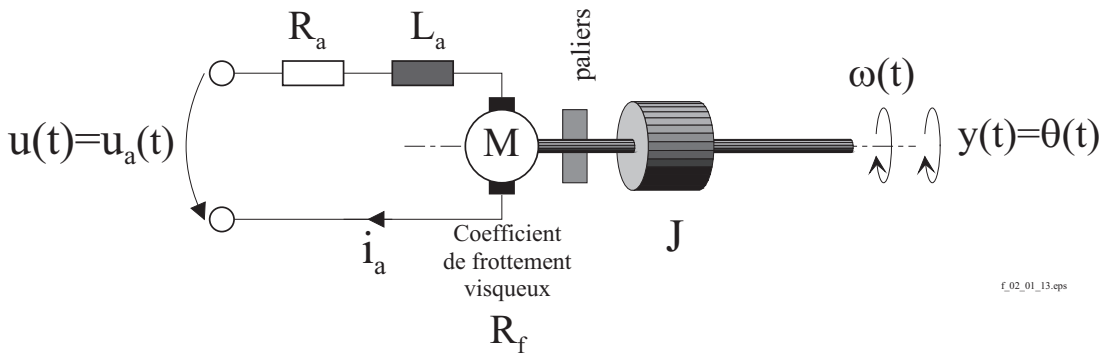
Tracer les réponses indicielles, impulsionnelle, fréquentielle du système. Simuler la réponse du système aux mêmes entrées (sinusoïdales) que celles sélectionnées à l'exercice 3.1 page précédente.

- Fonctions MATLAB/Octave utiles :
    - ss, tf
    - step, impulse, lsim
    - bode
  - Le canevas
    - [/MATLAB/Canevas/exercice\\_3\\_2\\_amortisseur\\_canevas.m](#)
- est à disposition.

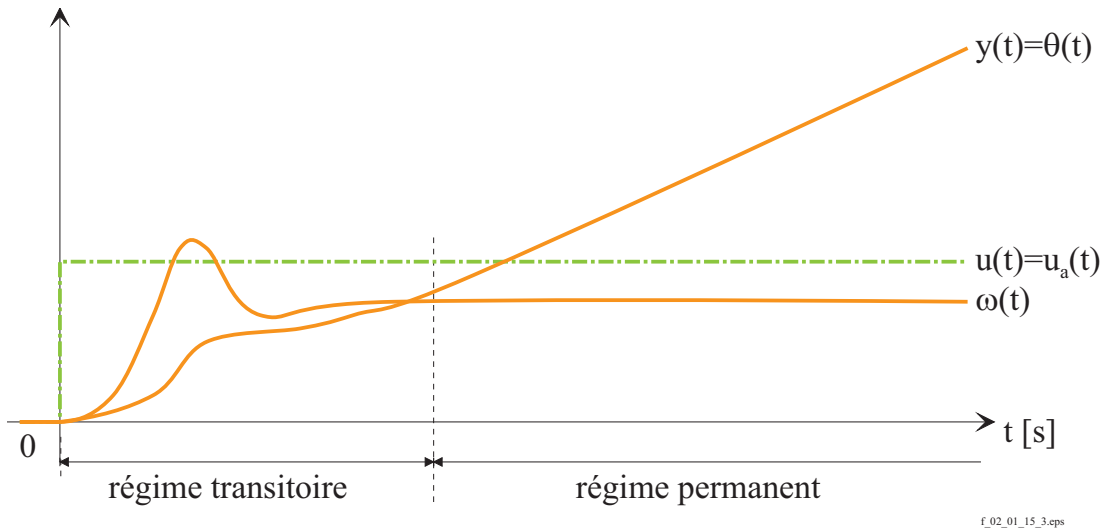
**Indication** Prendre connaissance au préalable des compléments théoriques de l'annexe A page 19.

## 4 Modélisation et simulation d'un moteur à courant continu à excitation séparée constante

Le schéma technologique d'un moteur DC à excitation séparée constante est donné sur la figure 4(a). Le signal d'entrée est la tension aux bornes de l'induit  $u_a(t)$  et le signal de sortie est dans le cas de cet exemple la position angulaire  $\theta(t)$  de la charge mécanique. Pour l'essentiel, la mise en équations repose les équations



(a) Schéma technologique. Le signal d'entrée est la tension  $u_a(t)$  aux bornes de l'induit alors que le signal de sortie est la position angulaire  $\theta(t)$  de l'arbre moteur (src).



(b) Réponse indicielle du moteur DC : esquisse des allures probables de la vitesse angulaire  $\omega(t)$  et de la position angulaire  $\theta(t)$  en régime permanent constant (src).

FIGURE 4 – Moteur DC à excitation séparée constante.

différentielles de la tension  $u_a(t)$  aux bornes de l'induit, de la dynamique de la rotation du rotor et de sa charge et finalement celle établissant la position

angulaire  $\vartheta(t)$  :

$$\begin{aligned}u_a(t) &= R_a \cdot i_a(t) + L_a \cdot \frac{di_a}{dt} + e_m(t) \\e_m(t) &= K_E \cdot \omega(t) \\T_{em}(t) &= K_T \cdot i_a(t) \\J \cdot \frac{d\omega}{dt} &= T_{em}(t) - R_f \cdot \omega(t) - T_{rés}(t) \\\frac{d\theta}{dt} &= \omega(t)\end{aligned}$$

Les paramètres du moteur sont les suivants :

$$\begin{aligned}J &= 122 \cdot 10^{-3} \text{ kg m}^2 \\R_f &= 95 \cdot 10^{-3} \frac{\text{N m}}{\frac{\text{rad}}{\text{s}}} \\K_T = K_E &= 600 \cdot 10^{-3} \frac{\text{N m}}{\text{A}} \\R_a &= 1.25 \Omega \\L_a &= 7.70 \text{ mH}\end{aligned}$$

## 4.1

Tracer les réponses indicielles, impulsionnelle, fréquentielle.

— Fonctions MATLAB/Octave utiles :

- ss, tf
- step, impulse, lsim
- bode

**Indication** Prendre connaissance au préalable des compléments théoriques de l'annexe A page 19.

## 5 Simulation d'un entraînement avec transmission flexible

Soit le système mécanique suivant, constitué de deux inerties accouplées par un arbre flexible (i.e. non-infiniment rigide, élastique ...). Il s'agit par exemple du rotor d'un moteur électrique et de sa charge (figure 5).

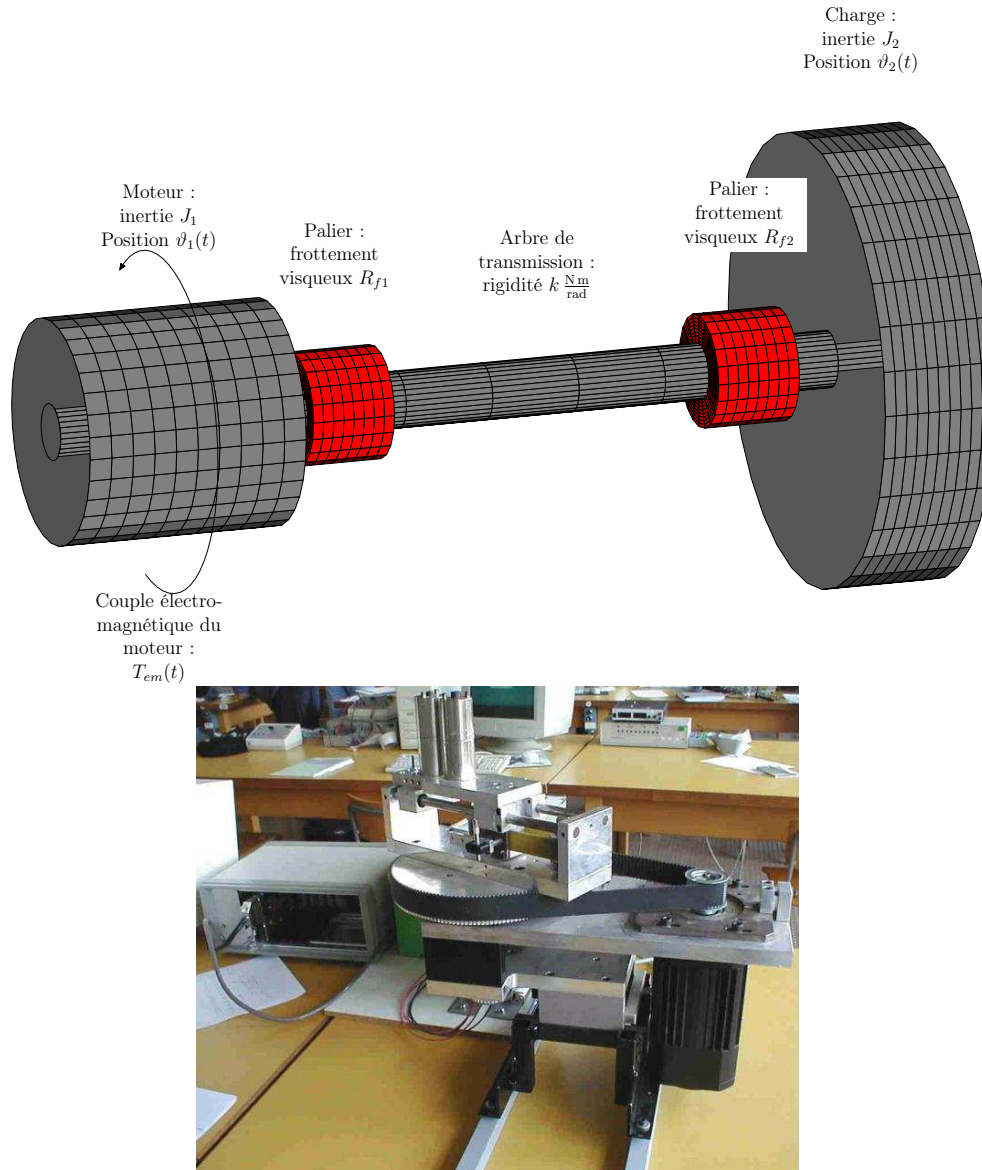


FIGURE 5 –

Les paliers créent chacun un couple de frottement visqueux de coefficients  $R_{f1}$  et  $R_{f2} \frac{\text{Nm}}{\text{rad}}$ , et l'arbre liant les masses en rotation est de rigidité  $k \frac{\text{Nm}}{\text{rad}}$ .

Paramètres :

$$J_1 = 450 \cdot 10^{-6} \text{ kg m}^2$$

$$J_2 = 250 \cdot 10^{-6} \text{ kg m}^2$$

$$R_f = 20 \cdot 10^{-3} \frac{\text{Nm}}{\frac{\text{rad}}{\text{s}}}$$

$$k = 1.74 \cdot 10^3 \frac{\text{Nm}}{\text{rad}}$$

Les équations décrivant le comportement dynamique de ce système sont

$$J_1 \cdot \frac{d\omega_1}{dt} = T_{em}(t) - R_{f1} \cdot \omega_1(t) + k \cdot (\theta_2(t) - \theta_1(t))$$

$$J_2 \cdot \frac{d\omega_2}{dt} = -R_{f2} \cdot \omega_2(t) + k \cdot (\theta_1(t) - \theta_2(t))$$

$$\frac{d\vartheta_1}{dt} = \omega_1(t)$$

$$\frac{d\vartheta_2}{dt} = \omega_2(t)$$

A l'aide de MATLAB/Octave, tracer la réponse indicielle du moteur ( $\vartheta_1(t)$ ) et de la charge ( $\vartheta_2(t)$ ).

Commandes MATLAB/Octave (*Control System Toolbox*)

#### Modèle d'état

- `sys = ss(A,B,C,D)` ;
- `step(sys)`
- `bode(sys)`

#### Fonction de transfert

- `G = tf(numG,denG)` ;
- `step(G)`
- `bode(G)`

## 6 Equations

### 6.1 Résolution d'un système d'équations

Soit le système d'équations :

$$\begin{aligned}3 \cdot x - 5 \cdot y + 0.1 \cdot z &= \pi \\ -x + 0.01 \cdot y - 30 \cdot z &= 0 \\ 10 \cdot x + 11 \cdot y - 0.01 \cdot z &= 15\end{aligned}$$

Résoudre ce système à l'aide de MATLAB/Octave.

— Commandes/fonctions MATLAB/Octave utiles

— `inv`

— Le canevas

[/MATLAB/Canevas/exercice\\_6\\_1\\_equations\\_canevas.m](/MATLAB/Canevas/exercice_6_1_equations_canevas.m)

est à disposition.

### 6.2 Solution d'une équation de degré supérieur

Trouver les solutions des équations suivantes :

1.  $10 \cdot x^3 - 7 \cdot x^2 + x = 0$

2.  $x^4 + 1 = 0$

— Commandes/fonctions MATLAB/Octave utiles

— `roots`

— Le canevas

[/MATLAB/Canevas/exercice\\_6\\_2\\_equations\\_canevas.m](/MATLAB/Canevas/exercice_6_2_equations_canevas.m)

est à disposition.



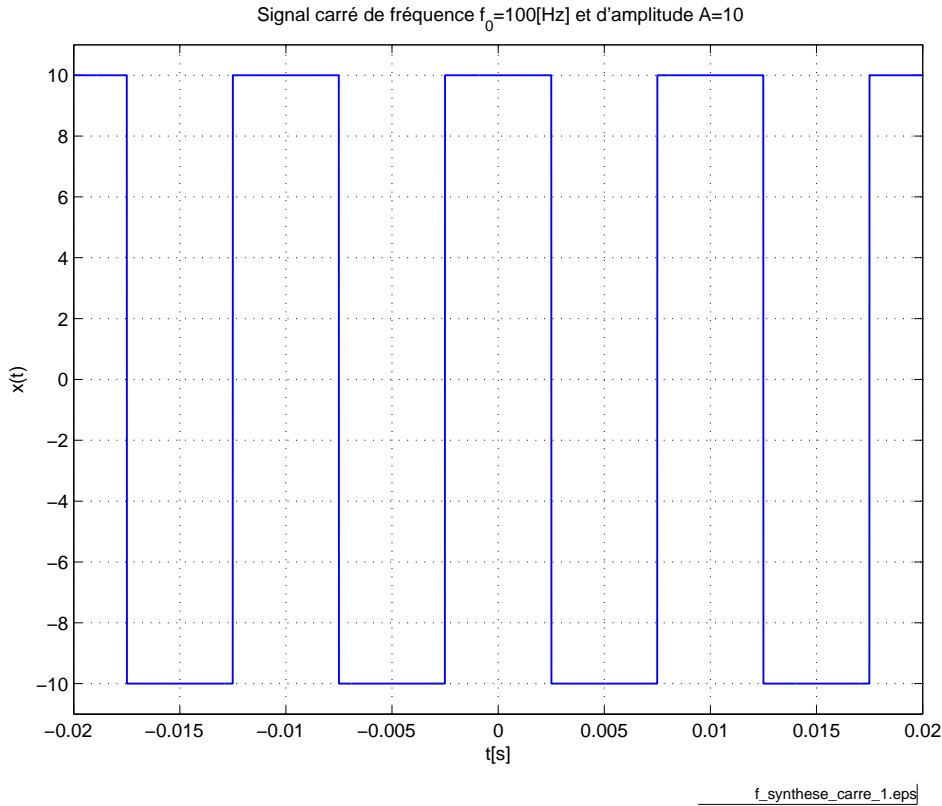


FIGURE 6 – Signal carré périodique.

## 7 Décomposition d'un signal carré en série de Fourier

On considère le signal carré  $x(t)$  périodique donné sur la figure 6. L'analyse de Fourier montre que ce signal peut être décomposé en une somme infinie de signaux cosinusoidaux

$$x(t) = A_0 + \sum_{k=1}^{\infty} A_k \cdot \cos(2 \cdot \pi \cdot k \cdot f_0 \cdot t + \alpha_k) \quad (1)$$

avec (cas général)

$$A_0 = X(j \cdot 0) \quad A_k = 2 \cdot |X(j \cdot k)| \quad \alpha_k = \arctan\left(\frac{\Im\{X(j \cdot k)\}}{\Re\{X(j \cdot k)\}}\right)$$

et (cas particulier du signal carré symétrique)

$$X(j \cdot k) = \begin{cases} 0 & \text{pour } k = 0 \text{ et } k \text{ paire} \\ A \cdot \frac{\sin(k \cdot \frac{\pi}{2})}{k \cdot \frac{\pi}{2}} = A \cdot \text{sinc}\left(k \cdot \frac{\pi}{2}\right) & \text{pour } k \text{ impaire} \end{cases}$$

Ecrire un programme MATLAB/Octave effectuant la synthèse du signal carré par la relation (1).

### Indications

1. Générer le signal à l'aide de la fonction `square` ; prendre garde à la phase du signal (se référer à la figure 6 page précédente)
2. Ecrire une boucle effectuant le calcul des coefficients  $X(j \cdot k)$ ,  $A_k$ ,  $\alpha_k$
3. Calculer dans la boucle

$$x_k(t) = A_k \cdot \cos(2 \cdot \pi \cdot k \cdot f_0 \cdot t + \alpha_k)$$

4. Calculer hors de la boucle

$$x_N(t) = A_0 + \sum_{k=1}^N x_k(t)$$

— Fonctions MATLAB/Octave utiles :

- `for ... end`
- `exp`, `atan2`
- `abs`, `angle`
- `cumsum`
- `etc`

— Le canevas

[/MATLAB/Canevas/exercice\\_7\\_fourier\\_canevas.m](#)

est à disposition.

## A Calcul de la réponse fréquentielle d'un système dynamique linéaire avec les phaseurs temporels

L'équation de Newton de l'oscillateur mécanique montré sur la figure 3 s'écrit :

$$m \cdot \frac{d^2x}{dt^2} = \sum_i F_{\text{ext}i} = F(t) - R_f \cdot \frac{dx}{dt} - k \cdot x(t)$$

On peut la présenter sous la forme

$$m \cdot \frac{d^2x}{dt^2} + R_f \cdot \frac{dx}{dt} + k \cdot x(t) = F(t) \quad (2)$$

En régime permanent sinusoïdal, l'excitation  $F(t)$  est sinusoïdale ; sous forme générale (amplitude et phase quelconques), on peut l'exprimer comme suit :

$$F(t) = \hat{F} \cdot \sin(\omega \cdot t + \varphi_F)$$

Le système étudié étant par hypothèse linéaire, une fois le régime transitoire passé, l'état de régime permanent sinusoïdal est atteint et le signal de sortie ( $x(t)$ , position de la masse  $m$ ) est également sinusoïdal :

$$x(t) = \hat{x} \cdot \sin(\omega \cdot t + \varphi_x)$$

Introduisant ces expressions dans l'équation différentielle (2), on a :

$$m \cdot \frac{d^2}{dt^2} (\hat{x} \cdot \sin(\omega \cdot t + \varphi_x)) + R_f \cdot \frac{d}{dt} (\hat{x} \cdot \sin(\omega \cdot t + \varphi_x)) + k \cdot \hat{x} \cdot \sin(\omega \cdot t + \varphi_x) = \hat{F} \cdot \sin(\omega \cdot t + \varphi_F) \quad (3)$$

De même, en régime permanent cosinusoidal, on a :

$$F(t) = \hat{F} \cdot \cos(\omega \cdot t + \varphi_F)$$

$$x(t) = \hat{x} \cdot \cos(\omega \cdot t + \varphi_x)$$

L'introduction des ces expressions dans l'équation différentielle (2) donne

$$m \cdot \frac{d^2}{dt^2} (\hat{x} \cdot \cos(\omega \cdot t + \varphi_x)) + R_f \cdot \frac{d}{dt} (\hat{x} \cdot \cos(\omega \cdot t + \varphi_x)) + k \cdot \hat{x} \cdot \cos(\omega \cdot t + \varphi_x) = \hat{F} \cdot \cos(\omega \cdot t + \varphi_F) \quad (4)$$

L'addition terme à terme des 2 équations (4) et (3), en multipliant préalablement (3) par  $j$ , donne :

$$\begin{aligned} m \cdot \frac{d^2}{dt^2} (\hat{x} \cdot \cos(\omega \cdot t + \varphi_x)) + j \cdot m \cdot \frac{d^2}{dt^2} (\hat{x} \cdot \sin(\omega \cdot t + \varphi_x)) \\ + R_f \cdot \frac{d}{dt} (\hat{x} \cdot \cos(\omega \cdot t + \varphi_x)) + j \cdot R_f \cdot \frac{d}{dt} (\hat{x} \cdot \sin(\omega \cdot t + \varphi_x)) \\ + k \cdot \hat{x} \cdot \cos(\omega \cdot t + \varphi_x) + j \cdot k \cdot \hat{x} \cdot \sin(\omega \cdot t + \varphi_x) \\ = \hat{F} \cdot \cos(\omega \cdot t + \varphi_F) + j \cdot \hat{F} \cdot \sin(\omega \cdot t + \varphi_F) \end{aligned}$$

La dérivée d'une somme de fonctions étant égale à la somme des dérivées des fonctions, on peut regrouper les termes :

$$\begin{aligned}
 m \cdot \frac{d^2}{dt^2} (\hat{x} \cdot \cos(\omega \cdot t + \varphi_x) + j \cdot \hat{x} \cdot \sin(\omega \cdot t + \varphi_x)) \\
 + R_f \cdot \frac{d}{dt} (\hat{x} \cdot \cos(\omega \cdot t + \varphi_x) + j \cdot \hat{x} \cdot \sin(\omega \cdot t + \varphi_x)) \\
 + k \cdot (\hat{x} \cdot \cos(\omega \cdot t + \varphi_x) + j \cdot k \cdot \hat{x} \cdot \sin(\omega \cdot t + \varphi_x)) \\
 = \hat{F} \cdot \cos(\omega \cdot t + \varphi_F) + j \cdot \hat{F} \cdot \sin(\omega \cdot t + \varphi_F)
 \end{aligned}$$

Comme  $e^{j \cdot \varphi} = \cos(\varphi) + j \cdot \sin(\varphi)$  (relation d'Euler), on a :

$$m \cdot \frac{d^2}{dt^2} (\hat{x} \cdot e^{j \cdot \varphi_x} \cdot e^{j \cdot \omega \cdot t}) + R_f \cdot \frac{d}{dt} (\hat{x} \cdot e^{j \cdot \varphi_x} \cdot e^{j \cdot \omega \cdot t}) + k \cdot \underbrace{\hat{x} \cdot e^{j \cdot \varphi_x}}_{\underline{\mathbf{X}}} \cdot e^{j \cdot \omega \cdot t} = \underbrace{\hat{F} \cdot e^{j \cdot \varphi_F}}_{\underline{\mathbf{F}}} \cdot e^{j \cdot \omega \cdot t} \quad (5)$$

On peut désormais introduire la notion de *phaseurs temporels*  $\underline{\mathbf{F}}$  et  $\underline{\mathbf{X}}$  de  $F(t)$  et  $x(t)$  :

$$\begin{aligned}
 \underline{\mathbf{F}} &= \hat{F} \cdot e^{j \cdot \varphi_F} \\
 \underline{\mathbf{X}} &= \hat{x} \cdot e^{j \cdot \varphi_x}
 \end{aligned}$$

On note que le module du phaseur temporel d'un signal sinusoïdal en fournit la valeur de crête alors que son argument en donne la phase :

$$\begin{aligned}
 |\underline{\mathbf{F}}| &= \hat{F} \quad \arg \{\underline{\mathbf{F}}\} = \varphi_F \\
 |\underline{\mathbf{X}}| &= \hat{x} \quad \arg \{\underline{\mathbf{X}}\} = \varphi_x
 \end{aligned}$$

La pulsation du signal est implicitement  $\omega$ , considéré comme constante (régime permanent sinusoïdal). En tenant compte de la définition des phaseurs dans (5), on a :

$$m \cdot \frac{d^2}{dt^2} (\underline{\mathbf{X}} \cdot e^{j \cdot \omega \cdot t}) + R_f \cdot \frac{d}{dt} (\underline{\mathbf{X}} \cdot e^{j \cdot \omega \cdot t}) + k \cdot \underline{\mathbf{X}} \cdot e^{j \cdot \omega \cdot t} = \underline{\mathbf{F}} \cdot e^{j \cdot \omega \cdot t} \quad (6)$$

Les dérivées successives se calculent facilement, sachant que les phaseurs ne sont pas dépendants de  $t$  :

$$\begin{aligned}
 \frac{d}{dt} \{\underline{\mathbf{X}} \cdot e^{j \cdot \omega \cdot t}\} &= \underline{\mathbf{X}} \cdot \frac{d}{dt} e^{j \cdot \omega \cdot t} = \underline{\mathbf{X}} \cdot j \cdot \omega \cdot e^{j \cdot \omega \cdot t} \\
 \frac{d^2}{dt^2} (\underline{\mathbf{X}} \cdot e^{j \cdot \omega \cdot t}) &= \frac{d}{dt} (\underline{\mathbf{X}} \cdot j \cdot \omega \cdot e^{j \cdot \omega \cdot t}) = \underline{\mathbf{X}} \cdot (j \cdot \omega)^2 \cdot e^{j \cdot \omega \cdot t}
 \end{aligned}$$

Dériver un phaseur par rapport à  $t$  revient à le multiplier simplement par  $j \cdot \omega$  ! En en tenant compte dans l'équation (6) :

$$m \cdot \underline{\mathbf{X}} \cdot (j \cdot \omega)^2 \cdot e^{j \cdot \omega \cdot t} + R_f \cdot \underline{\mathbf{X}} \cdot j \cdot \omega \cdot e^{j \cdot \omega \cdot t} + k \cdot \underline{\mathbf{X}} \cdot e^{j \cdot \omega \cdot t} = \underline{\mathbf{F}} \cdot e^{j \cdot \omega \cdot t} \quad (7)$$

On peut alors simplifier les termes en  $e^{j \cdot \omega \cdot t}$  :

$$m \cdot \underline{\mathbf{X}} \cdot (j \cdot \omega)^2 + R_f \cdot \underline{\mathbf{X}} \cdot j \cdot \omega + k \cdot \underline{\mathbf{X}} = \underline{\mathbf{F}} \quad (8)$$

On en déduit la *réponse harmonique* ou *réponse fréquentielle* du système :

$$\begin{aligned} G(j \cdot \omega) &= \frac{\underline{\mathbf{X}}}{\underline{\mathbf{F}}} \\ &= \frac{1}{m \cdot (j \cdot \omega)^2 + R_f \cdot j \cdot \omega + k} \\ &= \frac{\frac{1}{k}}{1 + \frac{R_f}{k} \cdot j \cdot \omega + \frac{m}{k} \cdot (j \cdot \omega)^2} \end{aligned} \quad (9)$$

La dernière correspondant à la *forme de Bode*.  $G(j \cdot \omega)$  est un nombre complexe qui peut être exprimé sous forme polaire :

$$\begin{aligned} G(j \cdot \omega) &= \frac{\underline{\mathbf{X}}}{\underline{\mathbf{F}}} \\ &= \frac{\hat{X} \cdot e^{j \cdot \varphi_x}}{\hat{F} \cdot e^{j \cdot \varphi_F}} \\ &= \frac{\hat{X}}{\hat{F}} \cdot e^{j \cdot (\varphi_x - \varphi_F)} \\ &= |G(j \cdot \omega)| \cdot e^{j \cdot \arg\{G(j \cdot \omega)\}} \\ &= A(\omega) \cdot e^{j \cdot \varphi(\omega)} \end{aligned}$$

Le module de  $G(j \cdot \omega)$  fournit donc le gain  $A(\omega)$  apporté par le système considéré en fonction de  $\omega$  alors que son argument donne le déphasage  $\varphi(\omega)$  apporté par le même système en fonction de  $\omega$ . Le signal d'entrée  $F(t) = \sin(\omega \cdot t)$  est donc :

- amplifié d'un facteur  $A(\omega) = |G(j \cdot \omega)|$
- déphasé d'un angle égal à  $\varphi(\omega) = \arg\{G(j \cdot \omega)\}$

$$G(j \cdot \omega) \quad \left\{ \begin{array}{l} A(\omega) = |G(j \cdot \omega)| \\ \varphi(\omega) = \arg\{G(j \cdot \omega)\} \end{array} \right.$$

## A.1 Obtention directe de la fonction de transfert

On peut obtenir la fonction de transfert 9 directement à partir de l'équation 2 en faisant simplement les substitutions

$$\begin{array}{lll} x(t) & \longrightarrow & \underline{\mathbf{X}} \\ F(t) & \longrightarrow & \underline{\mathbf{F}} \\ \frac{d}{dt} & \longrightarrow & j \cdot \omega \\ \frac{d^2}{dt^2} & \longrightarrow & (j \cdot \omega)^2 \\ \dots & \dots & \dots \\ \frac{d^n}{dt^n} & \longrightarrow & (j \cdot \omega)^n \end{array}$$

En effet

$$m \cdot \frac{d^2 x}{dt^2} + R_f \cdot \frac{dx}{dt} + k \cdot x(t) = F(t)$$

$$m \cdot (j \cdot \omega)^2 \cdot \underline{\mathbf{X}} + R_f \cdot j \cdot \omega \cdot \underline{\mathbf{X}} + k \cdot \underline{\mathbf{X}} = \underline{\mathbf{F}}$$

d'où

$$\underline{\mathbf{H}} = \frac{\underline{\mathbf{X}}}{\underline{\mathbf{F}}} = \frac{1}{m \cdot (j \cdot \omega)^2 + R_f \cdot j \cdot \omega + k}$$

Son introduction dans MATLAB/Octave (§ B page suivante) se fait par les commandes :

---

---

```
numH=[1];
denH=[m, Rf, k];
H = tf(numH,denH);
```

---

---

## B Introduction de réponses fréquentielles/fonctions de transfert dans MATLAB/Octave

L'introduction de fonctions de transfert s'effectue en deux temps, les numérateur et dénominateur devant être donnés séparément. Le principe est simple : les numérateur et dénominateur apparaissent sous forme de vecteurs-ligne, les composantes desquels étant les coefficients des puissances décroissantes de  $j \cdot \omega$  (ou  $s$  (systèmes analogiques) avec l'outil transformée de Laplace ou de  $z$  (systèmes numériques)).

Soit par exemple la réponse fréquentielle / fonction de transfert

$$G(j \cdot \omega) = \frac{\underline{Y}}{\underline{U}} = 36 \cdot \frac{1 + j \cdot \omega \cdot 10}{1 + j \cdot \omega \cdot 2 + (j \cdot \omega)^2 \cdot 0.1}$$

Son introduction dans MATLAB/Octave se fait par les commandes :

---

---

```
numG=36*[10,1];
denG=[0.1,2,1];
G = tf(numG,denG);
```

---

---

Dans le cas de la réponse fréquentielle de l'amortisseur (9), on écrira :

---

---

```
numG=[1/k];
denG=[k/m,Rf/m,1];
G = tf(numG,denG);
```

---

---

Le coefficient de  $(j \cdot \omega)^0$ , i.e. le coefficient du terme constant, **qu'il soit nul ou non**, doit toujours être donné. Ainsi, la réponse fréquentielle / fonction de transfert de l'intégrateur

$$G(j \cdot \omega) = \frac{\underline{Y}}{\underline{U}} = \frac{2.71828}{j \cdot \omega}$$

est introduite sous la forme :

---

---

```
numG=2.71828*[1];
denG=[1,0];
G = tf(numG,denG);
```

---

---

Les noms donnés aux numérateurs et dénominateurs sont libres. On aura toutefois intérêt à être organisé et rigoureux dans les notations employées.

## B.1 Calcul et tracé de réponses de systèmes dynamiques linéaires

### B.1.1 Réponse temporelle

Pour le calcul des réponses temporelles, les fonctions `impulse`, `step`, `lsim` sont disponibles. Leur signification sont données dans le tableau B.1.1 page suivante.

Lorsqu'aucun argument de sortie n'est spécifié (*lhs*, *left handside argument*), le calcul et le tracé des réponses sont simultanés.

Les paramètres `t`, `n`, et `w` sont des optionnels et `MATLAB/Octave` les détermine lui-même par défaut. Si l'on souhaite les fixer, on peut par exemple le faire selon les indications du tableau B.1.1.

Lors du calcul de la réponse harmonique de systèmes numériques, il convient de spécifier la période d'échantillonnage  $h$ .

Il est souvent utile d'effectuer ces calculs de réponse sans tracer celle-ci immédiatement. Dans ce cas, on procède comme ci-dessus en spécifiant toutefois des arguments de sortie dans lesquels `MATLAB/Octave` sauvegardera les résultats de ses calculs :

---

```
[y, t]=impulse(G, t);  
[y, t]=step(G, t);  
[y]=lsim(G, u, t);
```

---

`y` est un vecteur-colonne contenant la réponse cherchée et `t` un vecteur-ligne contenant les instants auxquels elle a été calculée. La figure suivante montre le calcul préalable des réponses impulsionnelle et indicielle avant leur tracé à l'aide de la fonction **plot**.

---

```
[ys, t] = step(Gyw);  
[yi] = impulse(Gyw, t);  
figure  
plot(t, [yi, ys])  
title('Réponses impulsionnelle et indicielle en boucle fermée')  
xlabel('t[s]')
```

---



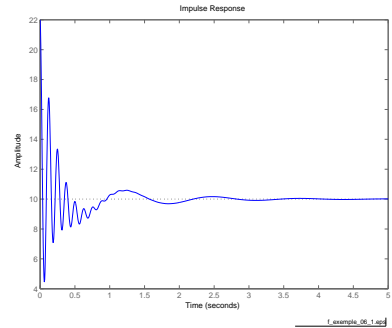
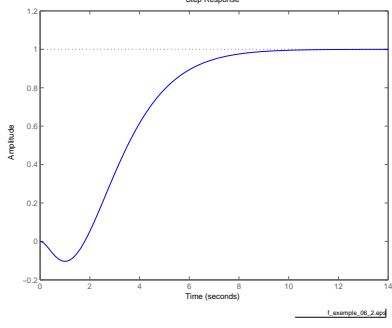
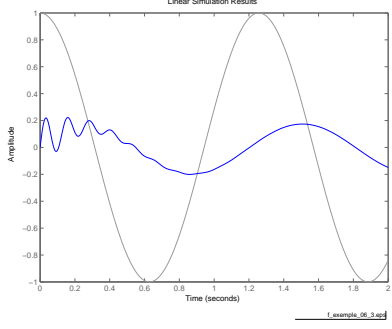
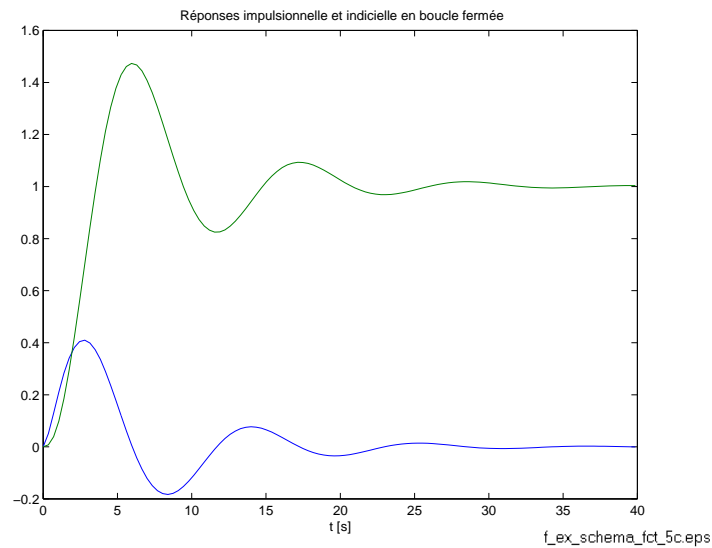
Commande	Réponse	Exemple
<code>impulse(G)</code>	réponse impulsionnelle	 <p>Impulse Response plot showing Amplitude vs Time (seconds). The response starts at 22, oscillates, and settles around 10. The plot is titled 'Impulse Response' and the file name is 'I_example_06_1.spj'.</p>
<code>step(G)</code>	réponse indicielle	 <p>Step Response plot showing Amplitude vs Time (seconds). The response starts at 0, dips slightly, and then rises to settle at 1. The plot is titled 'Step Response' and the file name is 'I_example_06_2.spj'.</p>
<code>lsim(G,u,t)</code>	réponse à une entrée $u$ quelconque définie par l'utilisateur	 <p>Linear Simulation Results plot showing Amplitude vs Time (seconds). The plot displays multiple oscillating signals, including a high-frequency one and a lower-frequency one. The plot is titled 'Linear Simulation Results' and the file name is 'I_example_06_3.spj'.</p>
<code>initial(A,B,C,D,x0,t)</code>	réponse du système lorsque ses conditions initiales sont $x_0$	

TABLE 1 –

Paramètre	Signification	Exemple
t	(facultatif) vecteur-ligne de temps, spécifiant les instants où la réponse doit être calculée (ce paramètre est obligatoire si l'on utilise <code>lsim</code> )	<code>t=[0 :0.01 :0.2];</code> ou <code>t=linspace(0,0.2,201);</code> (crée un vecteur temps variant entre 0 s et $200 \cdot 10^{-3}$ s par pas de $10 \cdot 10^{-3}$ s) ou $t = t_{\max}$
w	(facultatif) vecteur-ligne de pulsation, spécifiant les pulsations où la réponse doit être calculée	<code>w=logspace(2,5,100);</code> (crée un vecteur pulsation de 100 points espacés logarithmiquement et variant entre $1.00 \cdot 10^3 \frac{\text{rad}}{\text{s}}$ et $1.00 \cdot 10^6 \frac{\text{rad}}{\text{s}}$ )
u	vecteur-colonne de l'entrée simulée du système (ce paramètre est obligatoire si l'on utilise <code>lsim</code> )	<code>u=1.5*sin(2*pi*10*t)';</code> (crée un vecteur u représentant une entrée sinusoïdale d'amplitude 1.5 et de fréquence 10 Hz. u est calculé pour chaque instant défini dans le vecteur t, par exemple celui donné ci-dessus)

TABLE 2 –



De même, l'exemple suivant montre le tracé de la réponse d'un système asservi à une rampe de consigne, en faisant notamment usage de la fonction `lsim`.

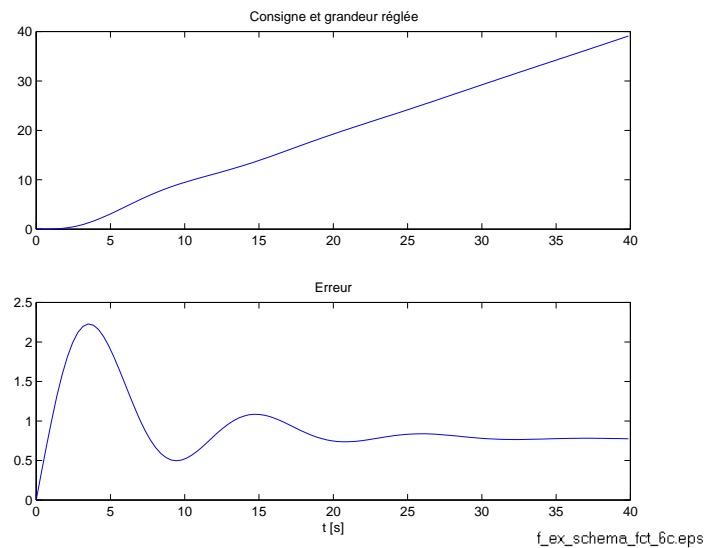
---

```

wr = t';
yr = lsim(Gyw,wr,t);
figure
subplot(211),plot(t,yr)
title('Consigne et grandeur réglée')
subplot(212),plot(t,wr-yr)
title('Erreur')
xlabel('t [s]')
subplot(111)

```

---



### B.1.2 Réponse fréquentielle

Les fonctions `bode`, `nyquist` et `nichols` sont disponibles. Leur utilisation est la suivante :

---

---

```
bode(G)
bode(G,w)
nyquist(G)
nyquist(G,w)
```

---

---

Si les résultats des calculs doivent être sauvegardés en vue d'un traitement ultérieur, on indique également des arguments de sortie ; dans le cas de la fonction `bode`, on aura par exemple :

---

---

```
[A, phi ,w] = bode(G);
[A, phi ] = bode(G,w);
```

---

---

`A` et `phi` sont des vecteurs-colonne contenant respectivement le gain et la phase et `w` un vecteur-ligne contenant les pulsations correspondantes.

## B.2 Analyse des propriétés des systèmes (unités d'enseignement **Régulation automatique (Régul)** et **Signaux et systèmes (SES)**)

<code>dcgain(G)</code>	Gain statique d'un système.
<code>damp(G)</code>	Taux d'amortissement $\zeta$ , pulsation propre non-amortie $\omega_n$ et pôles d'un système

## B.3 Calcul affichage des marges de gain et de phase

<code>[Am,phim]=margin(Go)</code>	Marges de phase et de gain d'un système.	Si les arguments de sortie ne sont pas spécifiés, trace le diagramme de Bode et montre graphiquement la valeur des marges
-----------------------------------	--	---

## C Représentation d'un système dynamique linéaire par son modèle d'état.

### C.1 Exemple introductif : circuit RLC série

#### C.1.1 Modèle entrée-sortie ("Input-output model")

On considère le circuit électrique suivant : En admettant que les paramètres

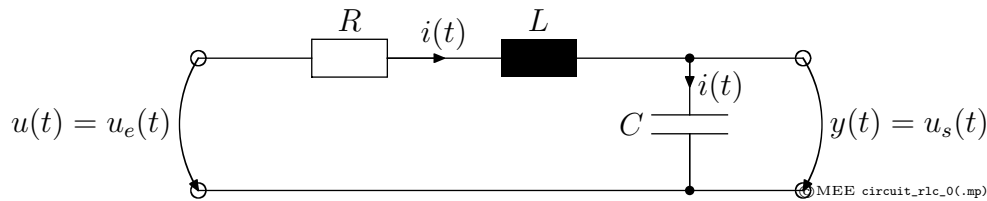


FIGURE 7 – Circuit RLC série (src).

$R$ ,  $L$  et  $C$  soient constants, la relation mathématique liant la tension de sortie  $u_s(t)$  à celle d'entrée  $u_e(t)$  peut être trouvée en écrivant l'équation (intégré-) différentielle régissant le circuit :

$$u_e(t) = R \cdot i(t) + L \cdot \frac{di}{dt} + \frac{1}{C} \cdot \int_{-\infty}^t i(\tau) \cdot d\tau \quad (10)$$

Notant que :

$$i(t) = \frac{dq}{dt} = C \cdot \frac{du_s}{dt} \quad (11)$$

$q(t)$  étant la charge instantanée du condensateur, et que

$$u_s(t) = \frac{1}{C} \cdot \int_{-\infty}^t i(\tau) \cdot d\tau \quad (12)$$

l'équation différentielle d'ordre 2 devient :

$$u_e(t) = R \cdot C \cdot \frac{du_s}{dt} + L \cdot C \cdot \frac{d^2u_s}{dt^2} + u_s(t) \quad (13)$$

soit encore :

$$\frac{d^2u_s}{dt^2} + \frac{R}{L} \cdot \frac{du_s}{dt} + \frac{1}{L \cdot C} \cdot u_s(t) = \frac{1}{L \cdot C} \cdot u_e(t) \quad (14)$$

Sa résolution fournit la relation cherchée entre

l'entrée  $u_e(t)$

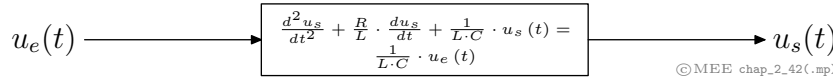


FIGURE 8 – Description du circuit de la figure 7 page précédente par un modèle de connaissance prenant la forme d’une équation différentielle d’ordre 2. Le modèle indique le lien entre l’entrée  $u_e(t)$  et la sortie  $u_s(t)$  : il s’agit d’un modèle *entrée sortie* (src).

et

la **sortie**  $u_s(t)$

du système.

Dans le cas de conditions initiales nulles, on peut extraire la **fonction de transfert** :

$$G(s) = \frac{U_s(s)}{U_e(s)} = \frac{1}{1 + s \cdot R \cdot C + s^2 \cdot L \cdot C} \quad (15)$$

Il s’agit là à nouveau d’une relation

**entrée-sortie**

où aucune des grandeurs **internes** du circuit n’intervient, bien que leur connaissance puisse être importante ; on pense notamment

- au courant  $i(t)$  ;
- au flux totalisé  $\Psi(t) = L \cdot i(t)$  ;
- à la charge instantanée du condensateur  $q(t)$  ;
- au champ électrique  $E(t)$  entre les armatures du condensateur.

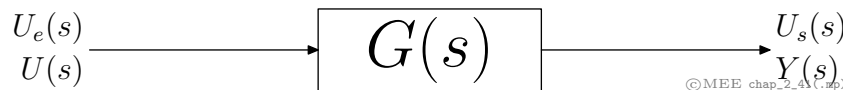


FIGURE 9 – Description du circuit de la figure 7 page précédente par un modèle de connaissance prenant la forme d’une fonction de transfert d’ordre 2. Comme le modèle de la figure 8, il s’agit également d’un modèle *entrée sortie* (src).

Un courant  $i(t)$  trop élevé peut provoquer une saturation magnétique se manifestant directement sur le flux totalisé  $\Psi(t)$ , alors qu’une charge exagérée du condensateur peut engendrer un champ électrique  $E$  supérieur au champ disruptif. Dans un cas comme dans l’autre, les hypothèses de linéarité sont démenties, mais aucune de ces grandeurs n’apparaît dans l’un ou l’autre des modèles entrée-sortie (équation différentielle d’ordre 2 et fonction de transfert) obtenus.

### C.1.2 Modèle d'état

La représentation dans l'espace d'état (**State space model**) offre une alternative au modèle entrée-sortie en proposant un modèle liant non seulement les signaux d'entrée et de sortie d'un système dynamique tout en gardant "à l'oeil" certaines grandeurs internes essentielles, les **variables d'état**.

Pour l'obtenir, il suffit de décrire le système dynamique par  $n$  **équations différentielles d'ordre 1** en lieu et place d'une seule équation différentielle d'ordre  $n$ . Pour le circuit électrique considéré, on pourrait écrire :

$$\begin{cases} u_e(t) = R \cdot i(t) + L \cdot \frac{di}{dt} + \frac{1}{C} \cdot q(t) \\ i(t) = \frac{dq}{dt} \end{cases} \quad (16)$$

où  $q(t)$  est la charge électrique instantanée du condensateur. En plaçant les dérivées premières dans les membres de gauche et en mettant en forme, on a :

$$\begin{cases} \frac{di}{dt} = -\frac{R}{L} \cdot i(t) - \frac{1}{L \cdot C} \cdot q(t) + \frac{1}{L} \cdot u_e(t) \\ \frac{dq}{dt} = i(t) \end{cases} \quad (17)$$

Ces deux équations, mises ainsi sous forme canonique, modélisent le comportement dynamique du circuit. Elles sont les **équations d'état** du système. L'expression de la tension de sortie  $u_s(t)$  est alors simplement

$$u_s(t) = \frac{1}{C} \cdot q(t) \quad (18)$$

qui est appelée **équation d'observation**.

En profitant de la notation matricielle, on peut présenter les trois dernières équations sous forme compacte :

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} i \\ q \end{bmatrix} &= \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L \cdot C} \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} i \\ q \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \cdot u_e \\ u_s &= \begin{bmatrix} 0 & \frac{1}{C} \end{bmatrix} \cdot \begin{bmatrix} i \\ q \end{bmatrix} \end{aligned} \quad (19)$$

La résolution de la première de ces équations (i.e. l'équation d'état) fournit  $i(t)$  et  $q(t)$  en fonction de  $u_e(t)$ . Le calcul de  $u_s(t)$  n'est alors plus qu'une simple formalité (combinaison linéaire des états  $i(t)$  et  $q(t)$ ) en faisant usage de la seconde équation, i.e. l'équation d'observation.

Les **variables d'états** du système sont ici

$$i(t) \quad \text{et} \quad q(t) \quad (20)$$

Elles ont été réunies dans le **vecteur d'état**

$$\vec{x} = \begin{bmatrix} i \\ q \end{bmatrix} \quad (21)$$



### C.1.3 Non-unicité de la représentation d'état

Remarquons que d'autres grandeurs pourraient faire office d'états. En faisant les substitutions

$$\begin{aligned} i(t) &= \frac{1}{L} \cdot \Psi(t) \\ u_s(t) &= \frac{1}{C} \cdot q(t) \end{aligned} \quad (22)$$

et en réécrivant les équations du circuit comme suit

$$\frac{1}{L} \cdot \frac{d\Psi}{dt} = -\frac{R}{L^2} \cdot \Psi(t) - \frac{1}{L} \cdot u_s(t) + \frac{1}{L} \cdot u_e(t) \quad (23)$$

$$C \cdot \frac{du_s}{dt} = \frac{1}{L} \cdot \Psi(t) \quad (24)$$

on a finalement, après avoir multiplié la première équation par  $L$  et la seconde par  $\frac{1}{C}$ ,

$$\frac{d}{dt} \begin{bmatrix} \Psi \\ u_s \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -1 \\ \frac{1}{L \cdot C} & 0 \end{bmatrix} \cdot \begin{bmatrix} \Psi \\ u_s \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot u_e \quad (25)$$

$$u_s = \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Psi \\ u_s \end{bmatrix} \quad (26)$$

ce qui montre déjà que la représentation d'état n'est pas unique.

## C.2 Définition

La représentation d'état d'un système dynamique linéaire est un modèle par lequel non seulement la relation *entrée-sortie* entre  $u(t)$  et  $y(t)$  est déterminée,

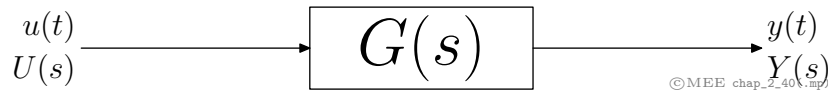


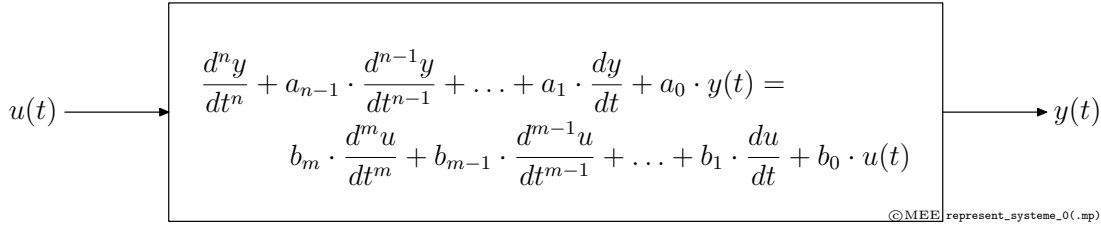
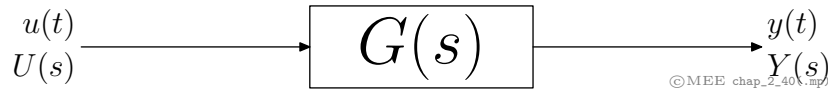
FIGURE 10 – Modèle entrée-sortie (src).

comme c'est déjà le cas avec

— l'équation différentielle d'ordre  $n$ ,

$$\frac{d^n y}{dt^n} + a_{n-1} \cdot \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_1 \cdot \frac{dy}{dt} + a_0 \cdot y = b_m \cdot \frac{d^m u}{dt^m} + b_{m-1} \cdot \frac{d^{m-1} u}{dt^{m-1}} + \dots + b_1 \cdot \frac{du}{dt} + b_0 \cdot u \quad (27)$$

— la réponse impulsionnelle  $g(t)$  ou la fonction de transfert  $G(s)$ ,

FIGURE 11 – Représentation d'un système dynamique linéaire par une équation différentielle d'ordre  $n$  (src).FIGURE 12 – Représentation d'un système dynamique linéaire par sa fonction de transfert (src).

mais également le comportement des grandeurs internes  $x_1 \dots x_n$  au système, appelées **variables d'état**.

$$\begin{cases} \frac{dx_1}{dt} = a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1n} \cdot x_n + b_1 \cdot u \\ \frac{dx_2}{dt} = a_{21} \cdot x_1 + a_{22} \cdot x_2 + \dots + a_{2n} \cdot x_n + b_2 \cdot u \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \frac{dx_n}{dt} = a_{n1} \cdot x_1 + a_{n2} \cdot x_2 + \dots + a_{nn} \cdot x_n + b_n \cdot u \end{cases} \quad (28)$$

Les variables d'état  $x_1$  à  $x_n$  sont au nombre de  $n$ ,  $n$  étant l'ordre du système. Elles apparaissent naturellement lors de la mise en équations d'un système. Si l'on s'astreint à modéliser celui-ci par un ensemble de  $n$  équations différentielles du 1<sup>er</sup> ordre, les grandeurs d'état sont alors celles faisant l'objet de la dérivée. Les  $n$  équations différentielles d'ordre 1 sont les **équations d'état** du système.

Bien qu'une définition claire des variables d'état soit relativement difficile à trouver dans la littérature, on proposera néanmoins la suivante :

Les variables d'état d'un système dynamique d'ordre  $n$  sont les  $n$  grandeurs  $x_1$  à  $x_n$  qu'il est nécessaire et suffisant de connaître à l'instant  $t_0$  pour calculer la réponse  $y(t)$  du système à toute entrée  $u(t)$ ,  $t \geq t_0$ .

Cela signifie que si  $x_1(t)$  à  $x_n(t)$  sont connues à un instant  $t_0$ , la connaissance des équations du système ainsi que du signal d'entrée  $u(t)$  qui lui est appliqué permet de calculer la réponse  $y(t)$  pour  $t \geq t_0$ . Dans ce sens, les variables d'état  $x_1(t_0)$  à  $x_n(t_0)$  à l'instant  $t_0$  coïncident avec les conditions initiales du système.

La connaissance à un instant donné des variables d'état du système permet donc d'en définir rigoureusement l'état, à l'instar par exemple des registres d'état ("status registers") d'un processeur. Toute autre donnée est alors superflue, hormis bien sûr les valeurs des paramètres ( $R$ ,  $L$ ,  $C$ ,  $J$ ,  $R_f$ , etc).

Le jeu de  $n$  équations différentielles ci-dessus doit en principe être complété par une équation définissant la relation entre les grandeurs d'état  $x_1(t)$  à  $x_n(t)$

et la sortie  $y(t)$  du système :

$$y = c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_n \cdot x_n + d \cdot u \quad (29)$$

Il s'agit de l'**équation d'observation**, dans laquelle le signal de sortie  $y(t)$  apparaît comme une combinaison linéaire des états  $x_1$  à  $x_n$ .

### C.2.1 Exemple : moteur DC

On considère un moteur DC à excitation séparée dont tous les paramètres sont supposés constants :

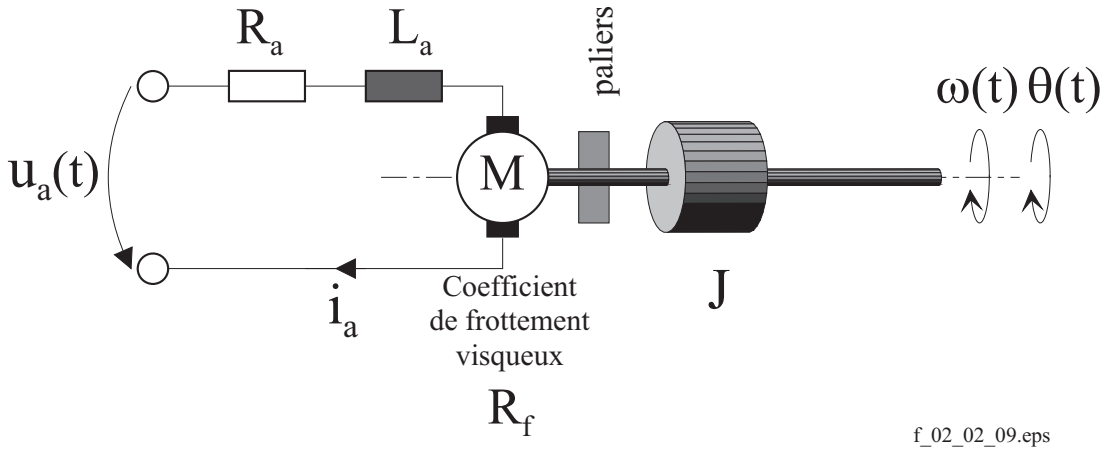


FIGURE 13 – Schéma technologique d'un moteur DC ([src](#)).

Les signaux d'entrée  $u(t)$  et de sortie  $y(t)$  sont ici la tension  $u_a(t)$  appliquée aux bornes de l'induit ainsi que la position angulaire  $\theta(t)$  respectivement. La mise en équations donne :

$$u_a(t) = R_a \cdot i_a(t) + L_a \cdot \frac{di_a}{dt} + K_E \cdot \omega(t) \quad (30)$$

$$T_{em}(t) = K_T \cdot i_a(t) \quad (31)$$

$$J \cdot \frac{d\omega}{dt} = T_{em}(t) - R_f \cdot \omega(t) \quad (32)$$

$$\frac{d\vartheta}{dt} = \omega(t) \quad (33)$$

Par simple mise en forme, on peut en déduire les **équations d'état**, en choi-

issant  $i_a$ ,  $\omega$  et  $\theta$  comme variables d'état :

$$\frac{di_a}{dt} = -\frac{R_a}{L_a} \cdot i_a - \frac{K_E}{L_a} \cdot \omega + \frac{1}{L_a} \cdot u_a \quad (34)$$

$$\frac{d\omega}{dt} = \frac{K_T}{J} \cdot i_a - \frac{R_f}{J} \cdot \omega \quad (35)$$

$$\frac{d\theta}{dt} = \omega \quad (36)$$

La connaissance de ces trois équations est nécessaire et suffisante pour décrire le comportement dynamique du système considéré, lequel est donc d'ordre  $n = 3$ .

La sortie  $y$  du système est donnée par l'équation d'observation et coïncide dans cet exemple avec l'un des états :

$$y = \theta \quad (37)$$

### C.3 Forme matricielle

Les équations différentielles d'ordre 1 ci-dessus sont avantageusement représentées en faisant usage de la notation matricielle :

$$\frac{d\vec{x}}{dt} = A \cdot \vec{x} + B \cdot u \quad (38)$$

$$y = C \cdot \vec{x} + D \cdot u \quad (39)$$

— le vecteur

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (40)$$

est le **vecteur d'état** ; c'est un vecteur colonne de dimension  $n \times 1$ . Ses composantes sont les  $n$  états du système.

— la matrice

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad (41)$$

est la **matrice d'état** ou **matrice système** ; c'est une matrice carrée de dimension  $n \times n$ .

Dans le cas d'un système mono-variable (une entrée  $u$ , une sortie  $y$ ),

— la matrice

$$B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (42)$$

est la **matrice d'entrée** prenant la forme d'un vecteur-colonne de dimension  $n \times 1$  ;

— la matrice

$$C = \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix} \quad (43)$$

est la **matrice de sortie**, vecteur-ligne de dimension  $1 \times n$  ;

— la matrice

$$D = [d] \quad (44)$$

est la **matrice de transfert direct**. Elle se réduit à un scalaire dans le cas mono-variable. Si elle est non-nulle, cela indique que l'entrée  $u$  intervient directement sur la sortie  $y$ , ce qui traduit un comportement statique (voir figure 15 page 39).

L'équation

$$\boxed{\frac{d\vec{x}}{dt} = A \cdot \vec{x} + B \cdot u} \quad (45)$$

est l'**équation d'état**. Elle seule détermine le comportement des états  $x_1$  à  $x_n$ , i.e. le comportement dynamique du système.

L'équation

$$\boxed{y = C \cdot \vec{x} + D \cdot u} \quad (46)$$

est l'**équation d'observation** ou encore équation de sortie ; elle n'a aucune influence sur les états. Elle permet de construire la/les sortie(s) du système par simple combinaison linéaire des états.

### C.3.1 Exemple : moteur DC

On reprend l'exemple du moteur DC à excitation séparée précédemment traité. Sous forme matricielle, ses équations d'état s'écrivent :

$$\underbrace{\frac{d}{dt} \begin{bmatrix} i_a \\ \omega \\ \theta \end{bmatrix}}_{\vec{x}} = \underbrace{\begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_E}{L_a} & 0 \\ \frac{K_T}{J} & -\frac{R_f}{J} & 0 \\ 0 & 1 & 0 \end{bmatrix}}_A \cdot \underbrace{\begin{bmatrix} i_a \\ \omega \\ \theta \end{bmatrix}}_{\vec{x}} + \underbrace{\begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix}}_B \cdot u_a \quad (47)$$

$$\underbrace{\theta}_y = \underbrace{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}}_C \cdot \underbrace{\begin{bmatrix} i_a \\ \omega \\ \theta \end{bmatrix}}_{\vec{x}} + \underbrace{[0]}_D \cdot \underbrace{u_a}_u$$

où le vecteur d'état est

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} i_a \\ \omega \\ \theta \end{bmatrix} \quad (48)$$

La représentation dans l'espace d'état constitue par ailleurs la forme idéale pour la simulation ; en effet, la plupart des méthodes de résolution de systèmes d'équations de 1<sup>er</sup> ordre linéaires ou non-linéaires (Runge-Kutta, Euler, etc) requièrent la forme dite *canonique*, où les dérivées premières (des états) apparaissent dans le membre de gauche, le membre de droite comprenant des combinaisons linéaires ou non-linéaires des états.

Par exemple, dans le cas linéaire, les réponses impulsionnelle, indicielle ou harmonique du système étudié sont facilement obtenues avec **MATLAB/Octave**, par les commandes respectives (offertes dans *Control System Toolbox*) :

- `step(A,B,C,D)`
- `impz(A,B,C,D)`
- `bode(A,B,C,D)`

exécutées après avoir introduit les valeurs numériques des matrices  $A$ ,  $B$ ,  $C$  et  $D$ . On a par exemple pour la réponse indicielle :

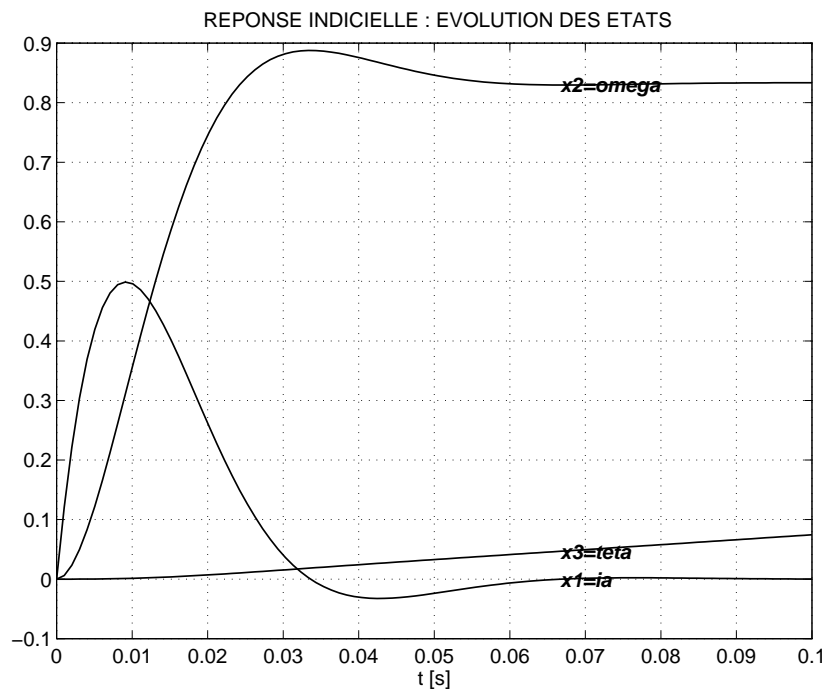


FIGURE 14 – La réponse indicielle du modèle d'état du moteur DC montre l'évolution des 3 états  $i(t)$ ,  $\omega(t)$  et  $\theta(t)$ .

## C.4 Schéma fonctionnel

Les équations d'état

$$\begin{aligned} \frac{d\vec{x}}{dt} &= A \cdot \vec{x} + B \cdot u \\ y &= C \cdot \vec{x} + D \cdot u \end{aligned} \quad (49)$$

peuvent être représentées graphiquement par le schéma fonctionnel général correspondant (figure 15). Ce schéma met en évidence le rôle capital joué par la

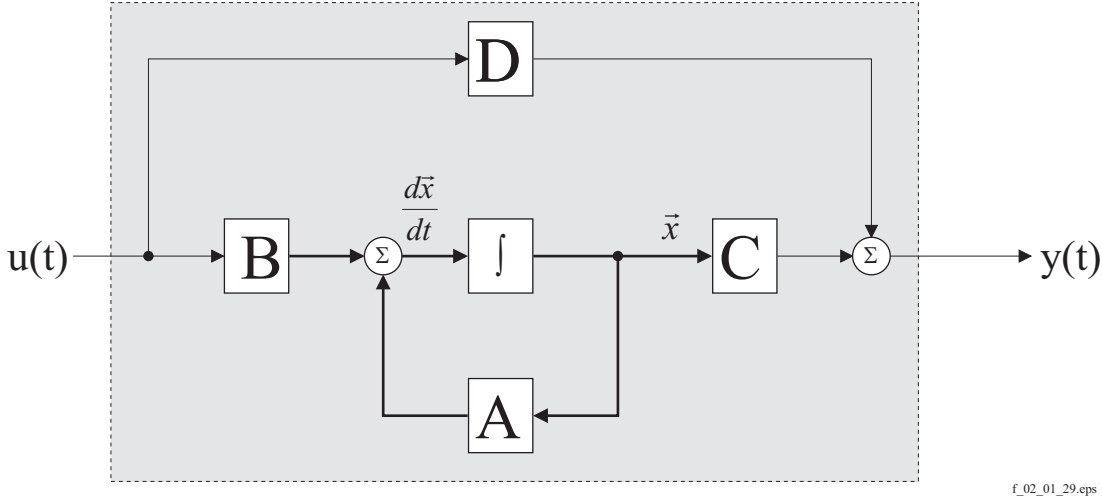


FIGURE 15 – Schéma fonctionnel (ou structurel) associé à une représentation par un modèle d'état. On observe que la matrice d'état  $A$  détermine les contre-réactions des états du système ( $\text{src}$ ).

matrice d'état  $A$ , laquelle détermine les contre-réactions internes au système. Il sera montré ultérieurement qu'elle seule détermine en fait la stabilité du système, ses valeurs propres coïncidant avec les pôles dudit système.

#### C.4.1 Exemple : moteur DC

Les équations d'état du moteur DC peuvent être représentées sous forme graphique. Un schéma fonctionnel possible est celui des figures 16 page suivante et 17 page 44 où les seuls éléments dynamiques intervenant sont des intégrateurs. L'avantage de ces schémas est que l'on peut voir au premier coup d'oeil la structure interne du système, notamment les relations existant entre les différentes grandeurs.

### C.5 Calcul de la fonction de transfert à partir du modèle d'état

On se propose ici de calculer la fonction de transfert du système sur la base des équations d'état. Notons que l'opération inverse est également possible.

Dans le cas de conditions initiales nulles, la transformée de Laplace des deux membres des équations d'état donne, pour un système mono-variable :

$$\begin{aligned} \frac{d\vec{x}}{dt} &= A \cdot \vec{x} + B \cdot u \\ y &= C \cdot \vec{x} + D \cdot u \end{aligned} \quad (50)$$

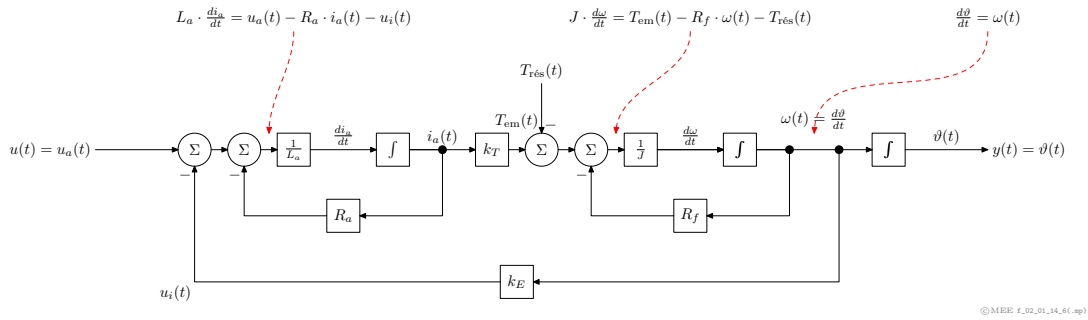


FIGURE 16 – Une représentation graphique possible des équations d'état du moteur DC (src).

Afin d'extraire la relation entrée sortie entre  $U(s)$  et  $Y(s)$ , on élimine  $\vec{X}(s)$  entre les deux équations :

$$\begin{aligned} s \cdot \vec{X}(s) - A \cdot \vec{X}(s) &= B \cdot U(s) \\ (s \cdot I - A) \cdot \vec{X}(s) &= B \cdot U(s) \\ \vec{X}(s) &= (s \cdot I - A)^{-1} \cdot B \cdot U(s) \end{aligned} \quad (51)$$

En introduisant cette dernière expression dans la seconde équation (l'équation d'observation)

$$Y(s) = C \cdot (s \cdot I - A)^{-1} \cdot B \cdot U(s) + D \cdot U(s) \quad (52)$$

on en déduit finalement la fonction de transfert recherchée :

$$G(s) = \frac{Y(s)}{U(s)} = C \cdot (s \cdot I - A)^{-1} \cdot B + D \quad (53)$$

**Rappel : inversion d'une matrice** L'inverse d'une matrice  $A$  est obtenu en divisant la transposée de la matrice des cofacteurs par le déterminant de  $A$ .  
Cas particulier : matrice 2 sur 2.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad A^{-1} = \frac{1}{a_{11} \cdot a_{22} - a_{12} \cdot a_{21}} \cdot \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \quad (54)$$

On peut ainsi obtenir la fonction de transfert du système décrit dans l'espace d'état à partir des matrices  $A$ ,  $B$ ,  $C$  et  $D$ . On voit qu'il est nécessaire d'inverser la matrice  $(s \cdot I - A)$  qui est d'ordre  $n$ , ce qui peut constituer un travail considérable.



L'expression de  $G(s)$  peut encore être développée en tenant compte de l'expression de l'inverse de  $(s \cdot I - A)$  :

$$\begin{aligned} G(s) &= \frac{Y(s)}{U(s)} = C \cdot \frac{[\text{cof}(s \cdot I - A)]^T}{|s \cdot I - A|} \cdot B + D \\ &= \frac{C \cdot [\text{cof}(s \cdot I - A)]^T \cdot B + D \cdot |s \cdot I - A|}{|s \cdot I - A|} = \frac{\text{polynôme en } s}{\text{polynôme en } s} \end{aligned} \quad (55)$$

On observe que le dénominateur de  $G(s)$  n'est autre que le déterminant de  $(s \cdot I - A)$ . Les racines du dénominateur étant les pôles  $s_1 \dots s_n$  de  $G(s)$ , on voit que ceux-ci correspondent aux *valeurs propres* de  $A$ , obtenues en résolvant :

$$d_c(s) = |s \cdot I - A| = 0 \Rightarrow \begin{cases} s_1 \\ s_2 \\ \vdots \\ s_n \end{cases} \quad (56)$$

## C.5.1 Exemple : moteur DC

La fonction de transfert  $G(s) = \frac{Y(s)}{U(s)} = \frac{\Theta(s)}{U_a(s)}$  est obtenue en procédant par étapes :

$$\begin{aligned}
 (s \cdot I - A) &= s \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_E}{L_a} & 0 \\ \frac{K_T}{J} & -\frac{R_f}{J} & 0 \\ 0 & 1 & 0 \end{bmatrix} \\
 &= \begin{bmatrix} s + \frac{R_a}{L_a} & +\frac{K_E}{L_a} & 0 \\ -\frac{K_T}{J} & s + \frac{R_f}{J} & 0 \\ 0 & -1 & s \end{bmatrix} \\
 (s \cdot I - A)^{-1} &= \frac{[\text{cof}(s \cdot I - A)]^T}{|s \cdot I - A|} \\
 \text{cof}(s \cdot I - A) &= \begin{bmatrix} s \cdot \left(s + \frac{R_f}{J}\right) & s \cdot \frac{K_T}{J} & \frac{K_T}{J} \\ -\frac{K_T}{L_a} \cdot s & s \cdot \left(s + \frac{R_a}{L_a}\right) & \left(s + \frac{R_a}{L_a}\right) \\ 0 & 0 & \left(s + \frac{R_a}{L_a}\right) \cdot \left(s + \frac{R_f}{J}\right) + \frac{K_T \cdot K_E}{J \cdot L_a} \end{bmatrix} \\
 [\text{cof}(s \cdot I - A)]^T &= \begin{bmatrix} s \cdot \left(s + \frac{R_f}{J}\right) & -s \cdot \frac{K_T}{L_a} & 0 \\ \frac{K_T}{J} \cdot s & s \cdot \left(s + \frac{R_a}{L_a}\right) & 0 \\ \frac{K_T}{J} & \left(s + \frac{R_a}{L_a}\right) & \left(s + \frac{R_a}{L_a}\right) \cdot \left(s + \frac{R_f}{J}\right) + \frac{K_T \cdot K_E}{J \cdot L_a} \end{bmatrix} \\
 |s \cdot I - A| &= s \cdot \left( \left(s + \frac{R_a}{L_a}\right) \cdot \left(s + \frac{R_f}{J}\right) + \frac{K_T \cdot K_E}{J \cdot L_a} \right) \\
 &= s \cdot \left( s^2 + \left( \frac{R_a \cdot J + R_f \cdot L_a}{L_a \cdot J} \right) \cdot s + \frac{K_T \cdot K_E + R_a \cdot R_f}{L_a \cdot J} \right) \\
 C \cdot (s \cdot I - A)^{-1} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \frac{\begin{bmatrix} s \cdot \left(s + \frac{R_f}{J}\right) & -s \cdot \frac{K_T}{L_a} & 0 \\ \frac{K_T}{J} \cdot s & s \cdot \left(s + \frac{R_a}{L_a}\right) & 0 \\ \frac{K_T}{J} & \left(s + \frac{R_a}{L_a}\right) & \left(s + \frac{R_a}{L_a}\right) \cdot \left(s + \frac{R_f}{J}\right) + \frac{K_T \cdot K_E}{J \cdot L_a} \end{bmatrix}}{s \cdot \left( s^2 + \left( \frac{R_a \cdot J + R_f \cdot L_a}{L_a \cdot J} \right) \cdot s + \frac{K_T \cdot K_E + R_a \cdot R_f}{L_a \cdot J} \right)} \\
 &= \frac{\begin{bmatrix} \frac{K_T}{J} & \left(s + \frac{R_a}{L_a}\right) & \left(s + \frac{R_a}{L_a}\right) \cdot \left(s + \frac{R_f}{J}\right) + \frac{K_T \cdot K_E}{J \cdot L_a} \end{bmatrix}}{s \cdot \left( s^2 + \left( \frac{R_a \cdot J + R_f \cdot L_a}{L_a \cdot J} \right) \cdot s + \frac{K_T \cdot K_E + R_a \cdot R_f}{L_a \cdot J} \right)} \quad (57)
 \end{aligned}$$

On voit ici que la connaissance de la matrice  $C$  peut permettre d'abrégier le calcul de l'inverse de  $(s \cdot I - A)$ , certaines composantes de cette dernière n'étant de toute façon pas prises en compte dans le calcul. La même remarque s'applique également à la matrice  $B$  intervenant dans le calcul suivant. Pour gagner du temps lors de l'extraction de  $(s \cdot I - A)^{-1}$  en évitant le calcul de certaines de ses composantes, on aura donc intérêt à prendre en compte la forme de  $B$  et  $C$  dès

le départ.

$$C \cdot (s \cdot I - A)^{-1} \cdot B = \frac{\begin{bmatrix} \frac{K_T}{J} & \left(s + \frac{R_a}{L_a}\right) & \left(s + \frac{R_a}{L_a}\right) \cdot \left(s + \frac{R_f}{J}\right) + \frac{K_T \cdot K_E}{J \cdot L_a} \end{bmatrix}}{s \cdot \left(s^2 + \left(\frac{R_a \cdot J + R_f \cdot L_a}{L_a \cdot J}\right) \cdot s + \frac{K_T \cdot K_E + R_a \cdot R_f}{L_a \cdot J}\right)} \cdot \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix}$$

$$= \frac{\frac{K_T}{J} \cdot \frac{1}{L_a}}{s \cdot \left(s^2 + \left(\frac{R_a \cdot J + R_f \cdot L_a}{L_a \cdot J}\right) \cdot s + \frac{K_T \cdot K_E + R_a \cdot R_f}{L_a \cdot J}\right)} \quad (58)$$

d'où finalement :

$$G(s) = \frac{Y(s)}{U(s)} = C \cdot (s \cdot I - A)^{-1} \cdot B + \underbrace{D}_0 = \frac{\frac{K_T}{L_a \cdot J}}{s \cdot \left(s^2 + \left(\frac{R_a \cdot J + R_f \cdot L_a}{L_a \cdot J}\right) \cdot s + \frac{K_T \cdot K_E + R_a \cdot R_f}{L_a \cdot J}\right)} \quad (59)$$

Le calcul symbolique ci-dessus est fastidieux et pourrait être aisément réalisé au moyen de logiciels de calcul symbolique comme **Mathematica**, **Maple**, **Mathcad** (qui comprend quelques primitives de calcul de Maple) ou **MATLAB/Octave** et sa boîte à outil **Symbolic** (à nouveau un extrait de **Maple**). Ce long calcul peut aussi être évité si l'on se contente d'une solution numérique, laquelle est aisément obtenue avec **MATLAB/Octave** au moyen de **ss2tf** ("State Space to Transfer Function")

Combiné avec **printsys(numG,denG)**, le résultat est :

```
>> [numG,denG]=ss2tf(A,B,C,D);
>> printsys(numG,denG)
num/den =
-5.457e-012 s + 1.277e+004
-----
s^3 + 162.4 s^2 + 1.533e+004 s
```

Du déterminant de  $(s \cdot I - A)$  peuvent être extraites les valeurs propres, i.e. les pôles  $s_1$  à  $s_3$  du système. Numériquement, cela peut se faire à l'aide de **MATLAB/Octave** par la fonction **eig** ("*eigenvalues*"), ce qui donne ici :

```
>> eig(A)
ans =
0
-81.1766 +93.4977i
-81.1766 -93.4977i
```

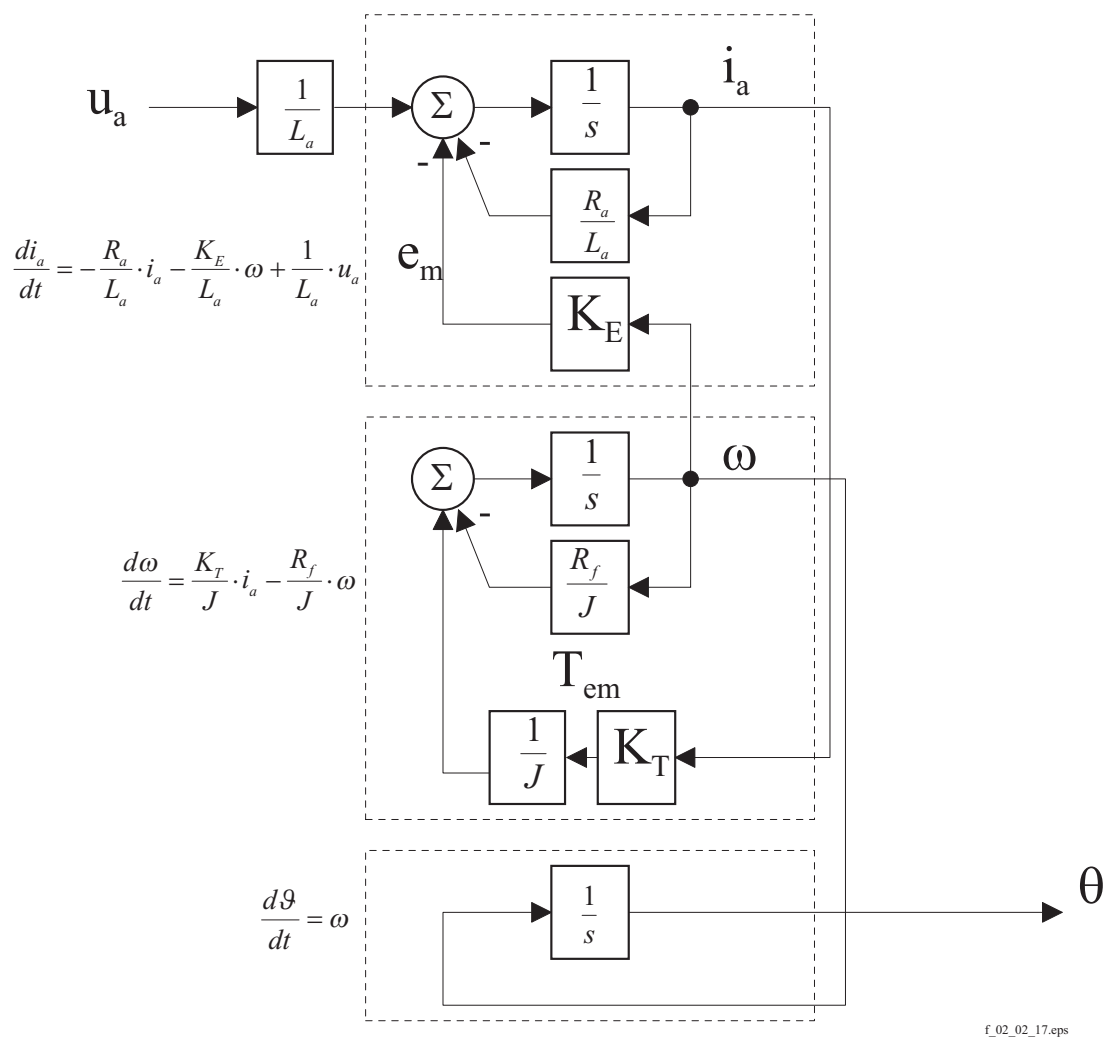


FIGURE 17 – Une autre représentation graphique des équations d'état du moteur DC (src).