



الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
المدرسة العليا للعلوم التطبيقية - تلمسان
Ecole Supérieure en Sciences Appliquées-Tlemcen

Recueil de Travaux Pratiques pour les systèmes linéaires asservis continus via MATLAB/SIMULINK

Dr. Ibtissem DIDI
Dr. Amel GHOUALI

Année universitaire: 2018-2019



Avant-Propos

Ce fascicule de travaux pratiques est une modeste contribution de notre part dans le cadre de l'élaboration d'un support pédagogique conformément aux programmes officiels de la formation du second cycle au sein de l'école supérieure en sciences appliquées de TLEMCEM (ESSAT) et s'adresse principalement aux étudiants de la troisième année filières Génie industrielle, Électrotechnique et Automatique.

Au terme de ces TPs, l'élève doit être capable :

- d'analyser et de synthétiser les Systèmes Asservis Linéaires en temps continu par des approches temporelles et fréquentielles ;
- de connaître les différents types de régulateurs standards ;
- d'évaluer les performances d'un système linéaire asservi en terme de stabilité, de précision et de rapidité.

0.1 Objectif

L'objectif de ces TPs est de permettre aux élèves une meilleure compréhension de la matière concernée. Ils doivent aussi être l'occasion de développer leur esprit critique ainsi que leur autonomie.

Les TPs porteront sur :

- ✓ **TP 1 : Initiation à MATLAB/SIMULINK.**
- ✓ **TP 2 : Introduction aux systèmes linéaires continus asservis.**
- ✓ **TP 3 : Analyse des systèmes linéaires continus asservis du premier ordre.**
- ✓ **TP 4 : Analyse des systèmes linéaires continus asservis du second ordre.**
- ✓ **TP 5 : Correction des systèmes linéaires continus asservis.**
- ✓ **TP 6 : Représentation d'état, commande et observation des systèmes linéaires continus asservis.**

0.2 Consignes

Les TPs se feront par binôme. Le choix des binômes est laissé libre aux élèves. Les TPs doivent être préparés afin d'éviter de passer trop de temps pour les calculs durant les séances. Il est fortement recommandé de lire l'intégralité du TP (pas uniquement la partie préparation) afin d'avoir un bon aperçu du travail demandé. Un compte rendu sera à rendre à la fin de chaque séance (un par binôme). Le compte rendu devra être rédigé à la main.

Le compte rendu **COMPORTERA** :

- Le nom et prénoms du binôme ainsi que la date du TP.
- **Une introduction** contenant les objectifs du TP.
- **La partie théorique** éventuellement demandée en préparation.
- **La partie manipulation** qui devra évidemment comporter les valeurs numériques obtenues lors de la séance mais surtout devra mettre en évidence ce que les élèves auront observés, analysés et compris lors de la séance. En cas d'erreur pendant le TP, il est fortement conseillé d'expliquer dans le compte rendu les erreurs commises, comment vous les avez détectées et corrigées (l'erreur est formatrice).
- **Une conclusion** qui devra mettre en évidence les compétences acquises lors du TP, les problèmes rencontrés ainsi que les méthodes utilisées pour détecter les problèmes et les résoudre.

Mais le compte rendu **NE COMPORTERA PAS** :

- Une recopie de l'énoncé et des questions posées.
- Des descriptions des commandes `MATLAB` tapées.
- Des courbes non commentées et non légendées.

Durant les séances l'encadrant est présent pour répondre aux questions de fond. Pour les question type : "*Comment on fait ... sous MATLAB*", l'élève sera invité à faire preuve d'autonomie en cherchant dans la documentation du logiciel (`help`).

0.3 Évaluation

L'évaluation des TPs se fait sur la base :

1. du travail effectué pendant chacune des séances de TPs ;
2. des comptes rendus réalisés pendant chaque séance de TP ;
3. un examen de TP à la fin du semestre.

En principe, deux élèves d'un même binôme ont la même note. Cependant, si un déséquilibre important apparaît dans le travail fourni, rien n'interdit d'attribuer des notes différentes.

$$\text{Note du TP}/20 = \frac{\text{Moyenne des comptes rendus} + \text{Examen de TP}}{2}$$

**Tout retard de plus de 15 minutes ne sera pas toléré
et une note de 0/20 sera attribuée à l'étudiant
retardataire.**

Table des matières

Avant-Propos	ii
0.1 Objectif	ii
0.2 Consignes	ii
0.3 Évaluation	iii
1 TP 1 : Initiation à MATLAB/SIMULINK	1
1.1 Objectif	1
1.2 Prise en main de MATLAB	1
1.2.1 Interface principale	1
1.2.2 Aspects élémentaires	2
1.3 Tableaux et matrices	3
1.3.1 Définition d'un vecteur et quelques fonctions utiles	3
1.3.2 Définition d'une matrice et quelques fonctions utiles	4
1.3.3 Opérations sur les matrices	5
1.4 Programmation avec MATLAB	5
1.4.1 Fichier SCRIPT	5
1.4.2 Fichier FUNCTION	6
1.5 Représentation graphique	6
1.5.1 Graphiques en 2D	6
1.5.2 Graphiques en 3D	8
1.6 SIMULINK	10
1.6.1 Création d'un modèle	11
1.6.2 Quelques bibliothèques	12
1.7 Énoncé du TP	14
2 TP 2 : Introduction aux systèmes linéaires continus asservis	15
2.1 Objectif	15
2.2 Généralités sur les systèmes	15
2.2.1 Système	15
2.2.2 Système linéaire	16
2.2.3 Système continu	16
2.3 Caractérisation des systèmes asservis	17
2.3.1 Structure des systèmes asservis	17
2.3.2 Objectifs de l'asservissement	17
2.3.3 Fonction d'un système asservi	18

2.3.4	Notion de fonction de transfert et schéma-blocs	18
2.4	Énoncé du TP	23
2.4.1	Partie théorique	23
2.4.2	Partie pratique	23
3	TP 3 : Analyse des systèmes linéaires continus asservis du premier ordre	24
3.1	Objectif	24
3.2	Performances des systèmes asservis	24
3.3	Caractérisation d'un système du premier ordre	25
3.3.1	Réponse indicielle (ou réponse à un échelon)	25
3.3.2	Réponse harmonique	26
3.4	Énoncé du TP	30
3.4.1	Partie théorique	30
3.4.2	Partie pratique	32
4	TP 4 : Analyse des systèmes linéaires continus asservis du second ordre	33
4.1	Objectif	33
4.2	Caractérisation d'un système du second ordre	33
4.2.1	Réponse indicielle (ou réponse à un échelon)	34
4.2.2	Réponse harmonique	35
4.3	Énoncé du TP	36
4.3.1	Partie théorique	36
4.3.2	Partie pratique	37
5	TP 5 : Correction des systèmes linéaires continus asservis	39
5.1	Objectif	39
5.1.1	Correcteur proportionnel (P)	39
5.1.2	Correcteur proportionnel et Intégral (PI)	40
5.1.3	Correcteur proportionnel et Dérivé (PD)	40
5.1.4	Correcteur proportionnel-Intégral-Dérivé (PID)	40
5.2	Énoncé du TP	42
6	TP 6 : Représentation d'état, commande et observation des systèmes continus asservis	43
6.1	Objectif	43
6.2	Représentation d'état	43
6.3	Commandabilité et observabilité	45
6.3.1	Critère de commandabilité	45
6.3.2	Critère d'observabilité	46
6.4	Commande par retour d'état	46
6.5	Observateur de Luenberger	47
6.5.1	Observateur identité	49

TP 1 : Initiation à MATLAB/SIMULINK

1.1 Objectif

Ce TP a pour but de familiariser l'étudiant avec l'usage de MATLAB/SIMULINK afin de découvrir les fonctions et outils relatifs à l'étude des systèmes linéaires continus asservis.

1.2 Prise en main de MATLAB

De nombreux logiciels performants sont aujourd'hui à la disposition des ingénieurs et permettent de réaliser des études simples ou complexes de façon très conviviale. MATLAB est un raccourci pour « **MAT**rix **LAB**oratory ». Il possède une boîte à outils dédiée à l'étude des systèmes de commandes « Control System Toolbox », ainsi qu'un environnement graphique de simulation numérique des systèmes dynamiques : SIMULINK.

1.2.1 Interface principale

Au lancement de MATLAB, une interface apparaît comme le montre la figure (1.1).

Le logiciel propose un véritable environnement de travail composé de multiples fenêtres. Nous pouvons distinguer quatre blocs :

1. **Command Window** (console d'exécution) : à l'invite de commande "> >", l'utilisateur peut faire rentrer les instructions à exécuter. Il s'agit de la fenêtre principale de l'interface.
2. **Current Directory** (répertoire courant) : permet de naviguer et de visualiser le contenu du répertoire courant de l'utilisateur. Les programmes de l'utilisateur doivent être situés dans ce répertoire pour être visible et donc exécutable.
3. **Workspace** (espace de travail) : permet de visualiser les variables définies, leur type, la taille occupée en mémoire ...
4. **Command History** : historique des commandes que l'utilisateur a exécutées. Il est possible de faire glisser ces commandes vers la fenêtre de commande.

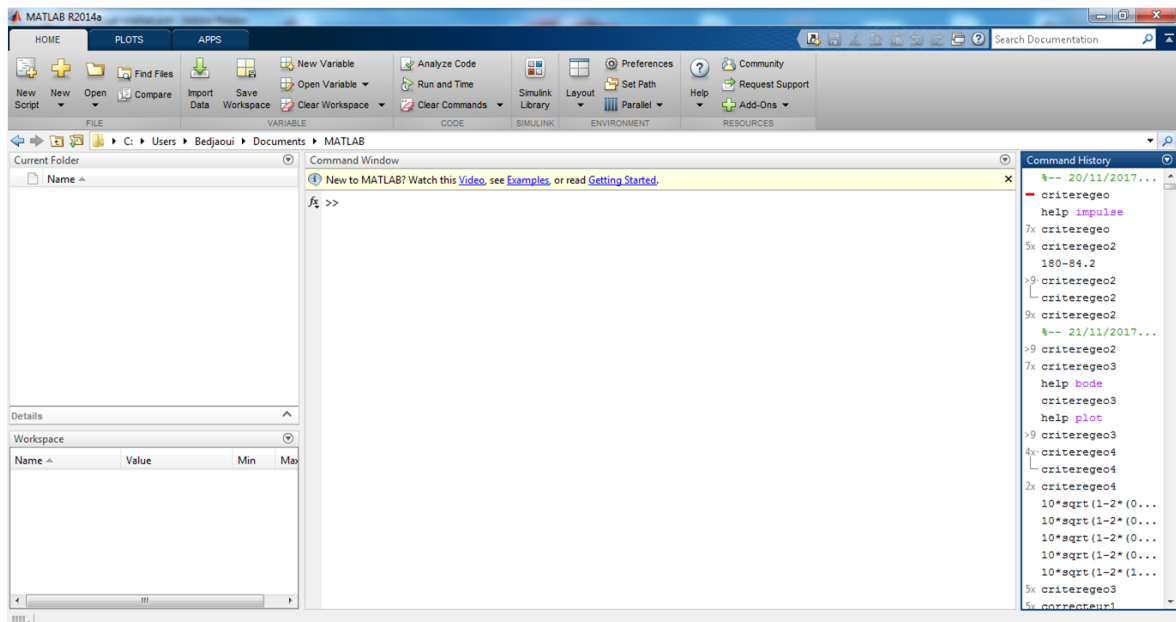


FIGURE 1.1: Interface graphique.

1.2.2 Aspects élémentaires

MATLAB gère de façon automatique les nombres entiers, réels, complexes, les chaînes de caractères ...

Ainsi, la déclaration des variables est implicite, et la mémoire nécessaire est automatiquement allouée lors de la définition de celle-ci. Le symbole d'affectation est le signe "=", par exemple :

- » `x=7` % On affecte la valeur 7 à la variable x
- » `y=8` % On affecte la valeur 8 à la variable y
- » `x=y` % On affecte la valeur contenue dans y à la variable x
- » `x+y`
- » `x+4`
- » `x=10;` %Pour ne pas afficher le résultat

Pour créer une variable de type chaîne de caractère, on utilise les quotes ' '.

- » `a='resultat'`

Remarque 1.2.1

MATLAB fait la différence entre les majuscules et les minuscules.

»x, X

On peut utiliser par exemple les commandes définies dans le tableau suivant :

clear X	Pour effacer la variable X de l'espace de travail et ainsi de la mémoire.
clear all	Pour effacer toutes les variables de l'espace de travail.
clc	Pour nettoyer la fenêtre de commande.
ctrl+c	Pour arrêter l'exécution d'un calcul à tout moment.
whos	Pour renvoyer le nom des variables de l'espace de travail, leur taille, leur occupation en mémoire et leur classe.
who	Pour afficher les variables courantes.
what	Pour afficher la liste des M-files présents.
help commande	Pour obtenir des informations relatives d'une instruction.

1.3 Tableaux et matrices

1.3.1 Définition d'un vecteur et quelques fonctions utiles

Un vecteur est considéré comme un tableau de valeurs. Il existe plusieurs façons de créer un vecteur et la plus simple est celle de l'écrire explicitement. L'ensemble des composantes est donné entre crochets et les valeurs sont séparées par un espace ou une virgule "," dans le cas d'un vecteur ligne. Un vecteur colonne est créé en utilisant un point virgule ";" comme délimiteur.

```
» V=[1 2 3 4] % Déclaration d'un vecteur ligne
V=
1 2 3 4
```

```
» V=[1;2;3;4] % Déclaration d'un vecteur colonne
V=
1
2
3
4
```

Cette méthode n'est pas pratique pour définir des vecteurs de taille importante. Une deuxième méthode utilise l'opérateur ":".

La syntaxe est la suivante `vecteur=valeur_initiale:valeur_finale`. Par défaut le pas est égal à 1. Sinon, il est possible de spécifier le pas de discrétisation.


```

» V=0:4 % Déclaration d'un vecteur V avec un pas de 1 (par défaut)
V=
0 1 2 3 4
» V=0:0.2:1 % Déclaration d'un vecteur V avec un pas de 0.2
V=
0 0.2 0.4 0.6 0.8 1

```

Le tableau ci-dessous présente un ensemble de fonctions usuelles liées à l'utilisation des tableaux ainsi que la syntaxe pour accéder aux différents éléments d'un vecteur.

linspace(x,y)	100 nombres entre x et y.
linspace(x,y,n)	n nombres entre x et y.
length(V)	revoie la taille du tableau.
V(i)	i^e coordonnée de V.
V(end)	dernière coordonnée de V .
V(i_1 :i_2)	coordonnée de i_1 à i_2 .
V(i_1 :i_2)=[]	supprimer les coordonnées de i_1 à i_2 de V.

1.3.2 Définition d'une matrice et quelques fonctions utiles

La définition d'une matrice est délimitée par des crochets "[]". Les différents éléments d'une ligne sont séparés par un espace ou une virgule et les différentes lignes sont séparées par des points virgules ";". La syntaxe est la suivante :

```

» A=[1 2 3 ; 4 5 6 ; 7 8 9]
      1 2 3
A =   4 5 6
      7 8 9

```

Le tableau ci-dessous présente la syntaxe pour accéder et modifier les éléments d'une matrice.

A(i,j)	Coefficient d'ordre i,j de A.
A(i_1 :i_2, :)	ligne i_1 à i_2 de A.
A(end, :)	dernière ligne de A.
A(i_1 :i_2, :)= []	supprimer les lignes i_1 à i_2 de A.
A(:,j_1 :j_2)	colonne j_1 à j_2 de A.
A(:,end)	dernière colonne de A.
A(:,j_1 :j_2)= []	supprimer les colonnes j_1 à j_2 de A.
[A,B]	concaténation horizontale de A et B.
[A ;B]	concaténation verticale de A et B.

Quelques matrices particulières et très utilisées, sont définies plus aisément au travers de fonctions.

zeros(m,n)	Matrice nulle de taille m,n.
ones(m,n)	Matrice de taille m,n dont les coefficients valent 1.
eye(m,n)	Matrice identité de taille $\min(m, n)$ complétée par des zéros.
diag(V)	Matrice diagonale dont la diagonale est le vecteur V.
diag(A)	Extrait la diagonale de la matrice A.
triu(A)	Annule les coefficients au-dessous de la diagonale.
tril(A)	Annule les coefficients au-dessus de la diagonale.
rand(3,3)	Matrice aléatoire 3×3 .
det(A)	Renvoie le déterminant de A.
rank(A)	Envoie le rang de A.
eig(A)	Renvoie les valeurs propres de A.

1.3.3 Opérations sur les matrices

Les opérations fondamentales sont données dans le tableau suivant :

$A + B, A - B$	Addition et soustraction terme à terme.
$A * B, B * A$	Produit matriciel standard.
$A .* B$	Multiplication élément par élément de deux tableaux.
$A.^n$	Évaluation à la puissance au sens habituel.
$A.^n$	Évaluation à la puissance terme à terme.
A/B	Si B est inversible, on a AB^{-1} .
$B \setminus A$	Si A est inversible, on a BA^{-1} .
A'	Transposition et conjugaison.

1.4 Programmation avec MATLAB

1.4.1 Fichier SCRIPT

Un fichier SCRIPT permet de regrouper des séries de commandes MATLAB. Cela évite d'avoir à saisir plusieurs fois de longues suites d'instructions. A son lancement, les instructions qu'il contient s'exécutent séquentiellement comme si elles étaient lancées depuis la fenêtre des commandes. Un SCRIPT stocke ses variables dans le `workspace`, lequel est partagé par tous les SCRIPTS. Ainsi toutes les variables créées dans le SCRIPT sont visibles depuis la Command Window et vice versa. Lorsque MATLAB détecte une erreur, le programme s'arrête et un message d'erreur s'affiche à l'écran avec le numéro de la ligne où l'erreur est détectée.

Remarque 1.4.1

Des annotations peuvent être ajoutées dans un programme MATLAB afin de le commenter. Pour cela, chaque ligne de commentaire doit être précédée par le symbole `%`. Les mots suivant ce symbole ne seront pas interprétés.

1.4.2 Fichier FUNCTION

Le principe d'une FUNCTION est d'effectuer des opérations à partir d'une ou plusieurs entrées et fournir une ou plusieurs sorties. Les variables d'entrée sont des paramètres à spécifier en argument de la fonction, tandis que les variables de sorties sont des valeurs qu'elle renvoie.

Un m-file est défini comme suit :

```
function [s1, s2, ...]=nom_fonction (e1, e2, ...)
```

Le membre de gauche regroupe les sorties renvoyées par la fonction et les variables entre parenthèses sont les entrées. **Le nom du fichier doit être identique au nom de la fonction.**

Exemple 1.4.2

Dans cet exemple, nous allons calculer le déterminant et la trace de la matrice

$$A = \begin{pmatrix} 1 & 2 & 0 \\ -1 & 5 & 6 \\ 1 & 0 & 1 \end{pmatrix}$$

```
function [D,T]=exemple (A)
%Calcul du déterminant de la matrice A
D=det(A);
%Calcul de la trace de la matrice A
T=trace(A);
```

L'appel de la fonction s'effectue de la manière suivante :

```
[Determinant,Trace]=exemple ([1 2 0;-1 5 6;1 0 1])
Determinant=
19
Trace=
7
```

1.5 Représentation graphique

Les bibliothèques de MATLAB proposent un très grand nombre de fonctions pour la manipulation d'objets graphiques. Nous ne présentons ici que quelques principes de base utiles pour le tracé de courbes, en l'occurrence la représentation graphique en 2 et 3 dimension.

1.5.1 Graphiques en 2D

Le tracé d'une courbe 2D s'effectue à partir de la commande `plot`. Cette dernière, prend en paramètre deux vecteurs et affiche sur un graphique à deux axes chaque couple de point de même indice. La syntaxe est la suivante :

```
plot(x,y,'s')
```

's' est un paramètre facultatif constitué d'une chaîne de caractère qui spécifie le type de tracé (couleur, type de trait, symbole,...). Par défaut, le tracé est continu. Pour avoir la liste des valeurs possible de 's', on utilise la commande `help plot`.

hold on	Les prochains tracés se superposeront aux tracés déjà effectués.
hold off	Le contenu de la fenêtre graphique active sera effacé lors du prochain tracé.
figure(n)	Affiche ou rend active la fenêtre graphique numéro n .
close	Ferme la fenêtre graphique active.
close all	Ferme toutes les fenêtres graphiques.
subplot(n,m,p)	Partage la fenêtre graphique active en $m \times n$ espaces graphiques et sélectionne le p -ième.

TABLE 1.1: Gestion de la fenêtre graphique

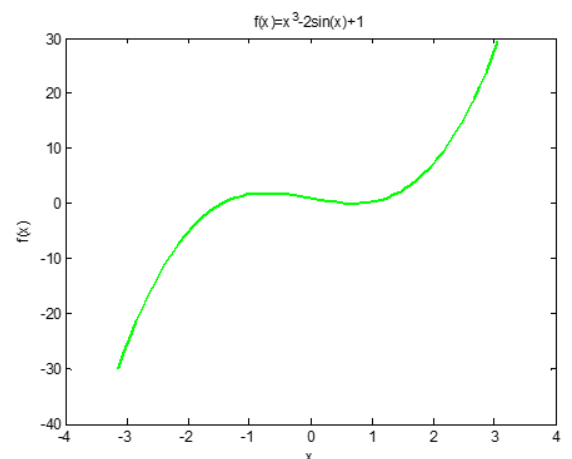
axis([xmin xmax ymin ymax])	Pour définir les échelles des axes.
grid ou grid on	Quadrillage du graphique.
grid off	Supprimer le quadrillage du graphique.
title('titre')	Titre pour le graphique.
xlabel('titre')	Légende pour l'axe des abscisses.
ylabel('titre')	Légende pour l'axe des ordonnées.
legend('titre1','titre2',...)	Légende pour chaque courbe du graphique.
text(x,y,'texte')	Texte explicatif à la position (x, y) .
gtext('texte')	Texte positionné à l'aide de la souris.

TABLE 1.2: Axes et légendes

Exemple 1.5.1

On considère la fonction $f(x) = x^3 - 2\sin(x) + 1$ qu'on va tracer dans l'intervalle $[-\pi, \pi]$ avec un pas $h = 0.1$, en utilisant la commande `plot`.

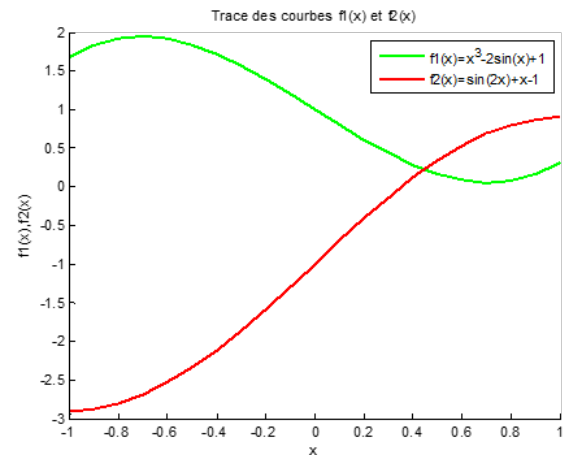
```
clear all, close all, clc
x = -pi : 0.1 : pi;
f = x^3 - 2 * sin(x) + 1;
plot(x,f, 'g', 'LineWidth', 2)
xlabel('x'),
ylabel('f(x)'), title('f(x) =
x^3 - 2 sin(x) + 1')
```



Exemple 1.5.2

On souhaite à présent tracer la courbe de la fonction $f(x) = \sin(2x) + x - 1$ dans l'intervalle $[-1, 1]$ sur la fenêtre précédente, en utilisant la commande `hold on/hold off`.

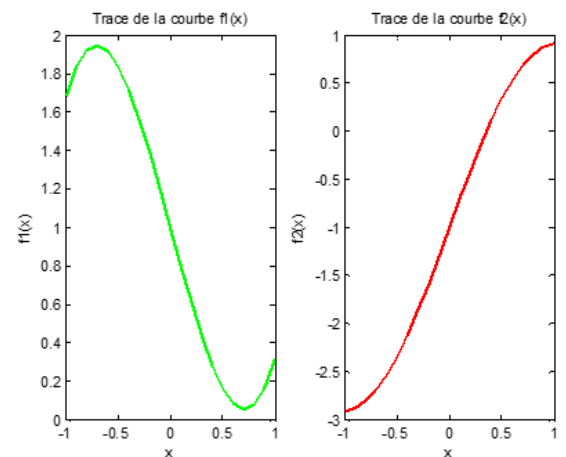
```
clear all, close all, clc
x = -1:0.1:1;
f1 = x3 - 2 * sin(x) + 1;
f2 = sin(2 * x) + x - 1;
hold on
plot(x,f1,'g','LineWidth',2)
plot(x,f2,'r','LineWidth',2)
hold off
xlabel('x'), ylabel('f1(x),
f2(x)'), title('Trace des
courbes f1(x) et f2(x)')
legend('f1 = x3 - 2 * sin(x) +
1', 'f2 = sin(2 * x) + x - 1')
```



Exemple 1.5.3

Il est possible aussi de partager la fenêtre graphique en plusieurs sous fenêtres comme suit :

```
clear all, close all, clc
x = -1:0.1:1;
f1 = x3 - 2 * sin(x) + 1;
f2 = sin(2 * x) + x - 1;
subplot(1,2,1), plot(x, f1,
'g', 'LineWidth', 2)
xlabel('x'), ylabel('f1(x)'),
title ('Trace de Cf1(x)')
subplot(1,2,2), plot(x, f2,
'r', 'LineWidth', 2)
xlabel('x'), ylabel('f2(x)'),
title ('Trace de Cf2(x)')
```



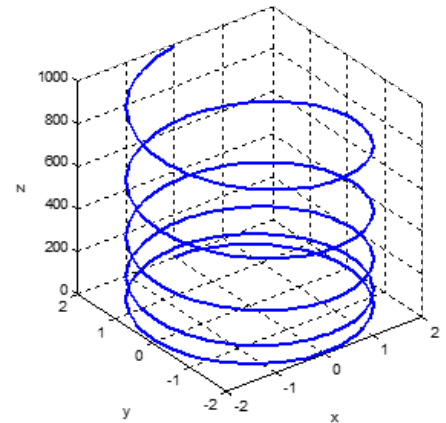
1.5.2 Graphiques en 3D

Nous montrons ici les possibilités de MATLAB en graphisme 3D sur quelques exemples.

Exemple 1.5.4

La fonction prend en argument 3 vecteurs de même taille. Son fonctionnement est similaire à celui de `plot`. Elle affiche dans un système d'axe à 3 dimensions les triplés $[x(i),y(i),z(i)]$.

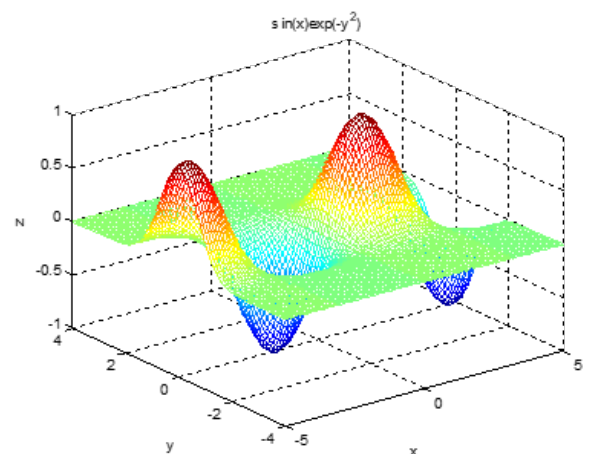
```
clear all, close all, clc
t = 0 : pi/50 : 10 * pi;
plot3 (2 * sin(t), 2 * cos(t), t^2,
'LineWidth', 2)
grid off
axis square
xlabel('x'), ylabel('y'),
zlabel('z')
```



Exemple 1.5.5

La fonction `mesh` trace un maillage (séries de lignes entre les points). Cette fonction prend en argument 2 matrices générées à partir de 2 vecteurs (x et y , pas nécessairement de même taille) et une matrice Z de dimension $(\text{length}(y), \text{length}(x))$. Pour chaque point du plan $(x(i), y(j))$, elle affiche la valeur $Z(j,i)$. Les 2 premières matrices obtenues à l'aide de la fonction `meshgrid` définissent tous les points du quadrillage. Une quatrième matrice peut être précisée pour définir une échelle de couleur associée au niveau du maillage. Cette échelle est basée sur une palette de couleurs prédéfinie par la valeur courante de la variable `colormap`.

```
clear all, close all, clc
x = -5 : 0.1 : 5;
y = -4 : 0.1 : 4;
[X,Y] = meshgrid(x,y);
f = sin(X) .* exp(-Y^2);
mesh(X,Y,f)
xlabel('x'), ylabel('y'),
zlabel('z'),
title('sin(x)exp(-y^2)')
```



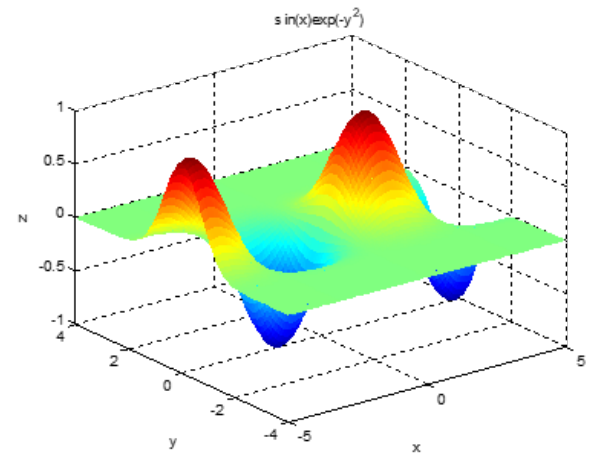
Exemple 1.5.6

Si l'on veut avoir une belle surface plutôt qu'un grillage, on peut utiliser la fonction `surf`.

```

clear all, close all, clc
x = -5:0.1:5;
y = -4:0.1:4;
[X,Y] = meshgrid(x,y);
f = sin(X).*exp(-Y^2);
surf(X,Y,f)
shading 'interp'
colormap(jet(400))
xlabel('x'), ylabel('y'),
zlabel('z'),
title('sin(x)exp(-y^2)')

```



1.6 SIMULINK

SIMULINK est un environnement graphique interactif pour la modélisation et la simulation des systèmes dynamiques linéaires et non-linéaires. SIMULINK est fondé sur l'utilisation de schémas blocs, ce qui le rend particulièrement convivial et intuitif par rapport à la boîte à outil **Control System** de MATLAB.

SIMULINK se lance en cliquant sur l'icône qui se trouve dans la barre d'outil de MATLAB (figure 1.2) ou bien, en exécutant la commande SIMULINK dans la fenêtre de commande. Une nouvelle fenêtre s'ouvre répertoriant les bibliothèques de blocs disponibles (figure 1.2).

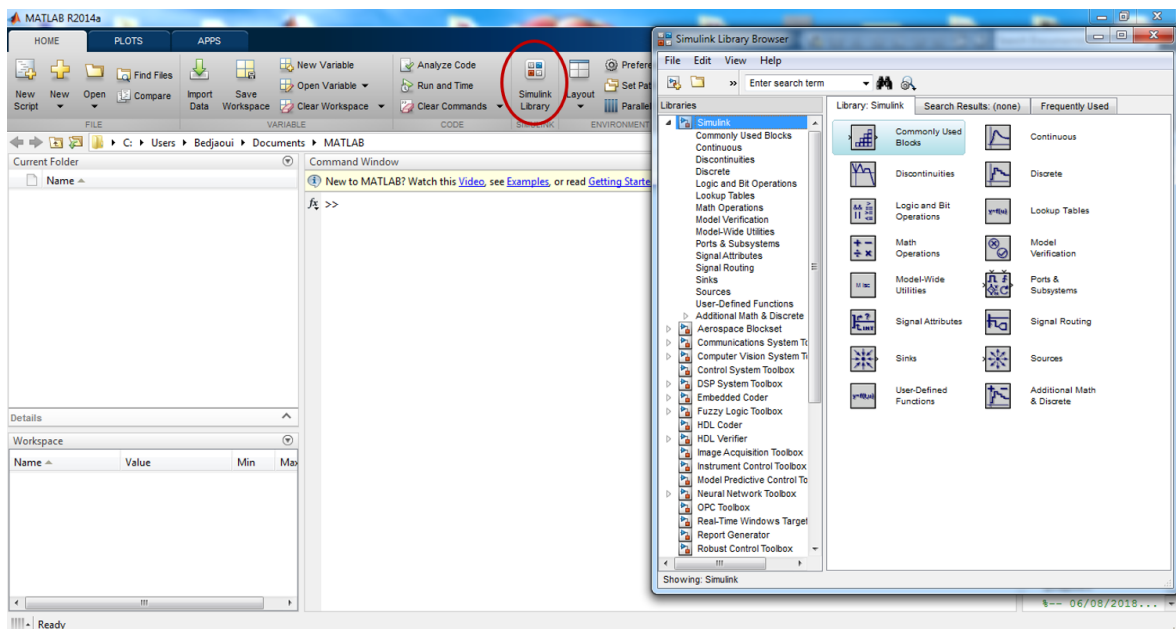


FIGURE 1.2: Répertoire des bibliothèques de blocs SIMULINK.

1.6.1 Création d'un modèle

Un nouveau modèle est créé en cliquant sur l'icône « feuille blanche » (ou dans le menu File > New > Model). Une fenêtre vide apparaît, elle va servir de support pour construire votre modèle.

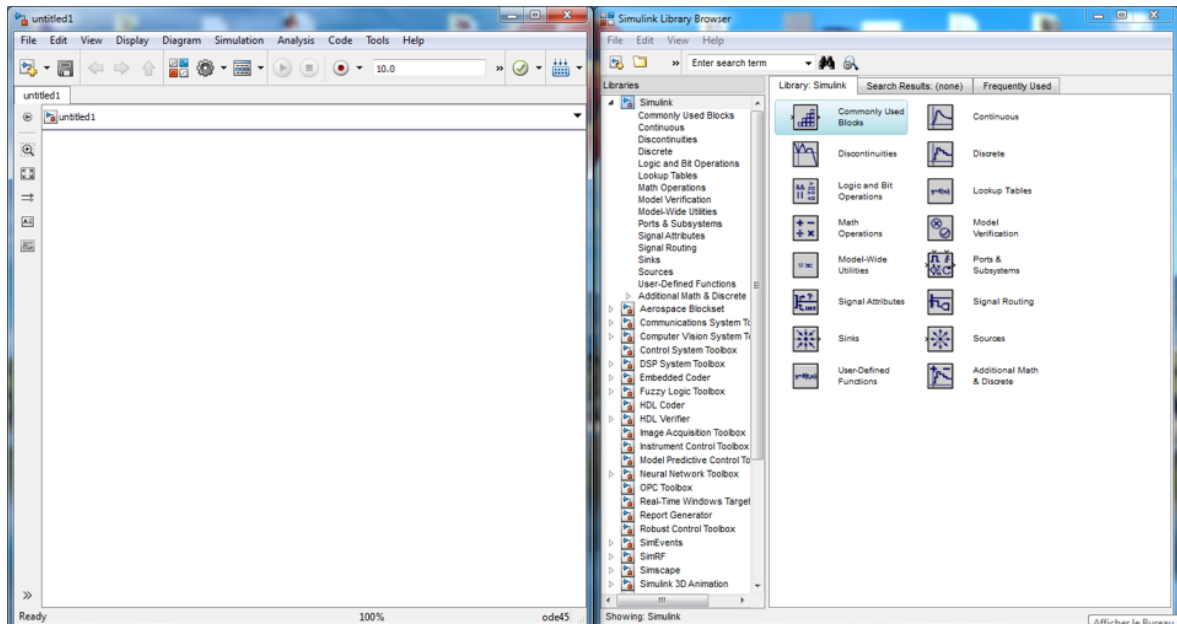


FIGURE 1.3: Création d'un modèle sous SIMULINK.

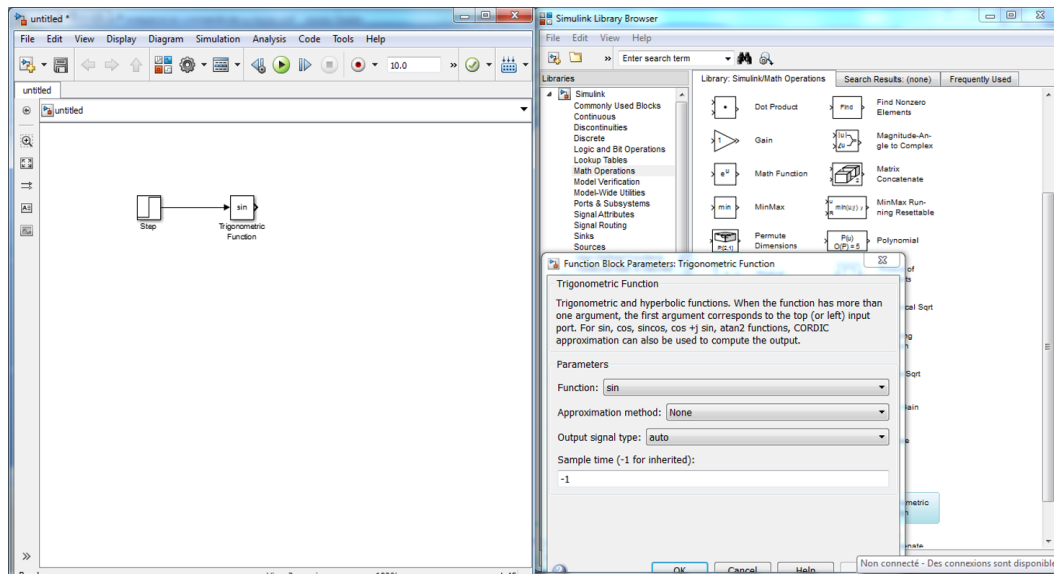
L'interface est facile à utiliser, un modèle est construit à partir des blocs fonctionnels par glisser-déposer. Les blocs sont ensuite inter-connectés par des flèches dessinées à l'aide de la souris (cliquer-maintenir sur une entrée ou une sortie puis relâcher sur une entrée ou une sortie).

Chaque bloc peut être édité (réglages de ses paramètres) en double-cliquant dessus. Cette dernière action permet d'ouvrir une fenêtre de dialogue dans laquelle une description du bloc et des champs paramétrables sont donnés.

Afin d'explicitier les différents éléments du schéma, il est possible de nommer les blocs et les liens. Pour modifier le nom du bloc, il suffit de cliquer sur son intitulé. Pour donner un nom à un lien, on double-clique sur le lien. Pour ajouter un commentaire, on double-clique dans une zone vierge de la fenêtre.

Une fois le schéma terminé, on peut l'enregistrer dans un fichier via la commande **Save As** du menu **File**. Les fichiers SIMULINK portent l'extension **.mdl**.

Une fois le modèle réalisé, l'intérêt de SIMULINK consiste à le simuler. Une simulation peut être lancée soit à partir de l'icône « lecture » en forme de triangle (ou dans le menu Simulation > Run), soit à partir de MATLAB avec la commande `sim()`. Le premier paramètre de simulation à régler est le temps de simulation. Il peut être spécifié dans le champ à droite de l'icône de lancement d'une simulation. Il doit être exprimé en seconde.



Par exemple, si l'on souhaite visualiser la réponse d'un signal sinusoïdal, on réalise le schéma bloc de la figure 1.4 à l'aide d'un bloc **Sine Wave Function** de la bibliothèque **Math Operations** et d'un bloc oscilloscope virtuel **Scope** de la bibliothèque **Sinks**.

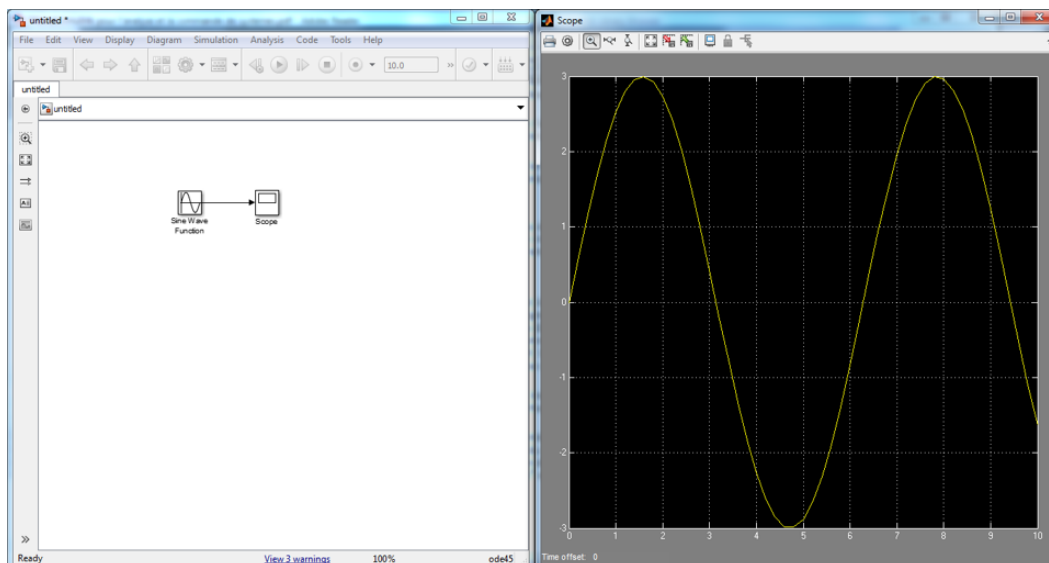
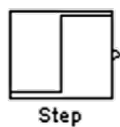


FIGURE 1.4: Représentation d'une fonction sinusoïdale sous Simulink.

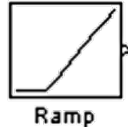
1.6.2 Quelques bibliothèques

SIMULINK possède de nombreuses bibliothèques, adaptées pour la modélisation d'une multitude de systèmes de différentes natures. Nous allons voir quelques blocs standards, souvent utilisés pour la simulation des systèmes asservis.

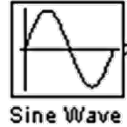
Sources. Les sources sont des blocs possédant une ou plusieurs sorties et aucune entrée. Ils sont utilisés pour la génération de signaux :



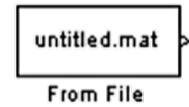
Step
Signal
échelon



Ramp
Signal
rampe

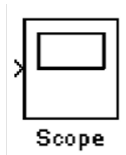


Sine Wave
Signal
sinusoïdale

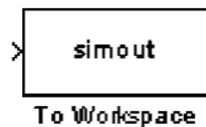


From File
Valeurs fournies
par un fichier

Sinks. Les blocs de cette librairie, à l'inverse des sources, ne possèdent qu'une ou plusieurs entrées. Ils sont utilisés pour l'affichage (digital, oscilloscope) ou la mémorisation de signaux :



Scope
Affichage
type
oscilloscope

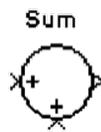


To Workspace
Stockage
dans une
variable

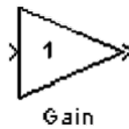


To File
Stockage
dans un
fichier

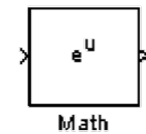
Maths Operations. Ensemble de blocs réalisant une fonction mathématique appliquée aux signaux entrants :



Sum
Somme/soustrait
des signaux



Gain
Multiplie un
signal

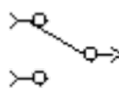


Math
Function
Applique une
fonction standard

Signal Routing. Ensemble de blocs utiles pour l'aiguillage de signaux ou la connexion de blocs :



Mux
Multiplexeur :
combine deux entrées
en une seule (vecteur)



Manual Switch
Switch : permet de
sélectionner manuellement
une entrée



Demux
Démultiplexeur : sépare
une entrée (vecteur) en
plusieurs composantes

1.7 Énoncé du TP

Exercice 1. On note A , B , et C les matrices suivantes :

$$A = \begin{pmatrix} 1 & 3 & 2 \\ -5 & 3 & 1 \\ -10 & 0 & 3 \\ 1 & 0 & -2 \end{pmatrix}, B = \begin{pmatrix} 1 & -2 & 5 \\ 6 & 1 & -1 \end{pmatrix}, C = \begin{pmatrix} 10 & -5 \\ 3 & 1 \end{pmatrix}$$

Dans la commande window :

1. Calculer les matrices AB , BA et AB^T .
2. Calculer la matrice $D = I_2 - BB^T$.
3. Calculer les déterminants des matrices A , B , C , D et $E = AA^T$.
4. Calculer les inverses des matrices A , B , C , D et E .
5. Calculer les valeurs propres de la matrice E . Quel est le rayon spectral de E ?
6. Déterminer les vecteurs propres de la matrice A .

Exercice 2. Écrire un fichier `function` nommé **f.m** pour les fonctions :

$$f_1(x) = x^2 \sin(x), f_2(x) = x|x| \text{ et } f_3(x) = x \sqrt[n]{nx}, n = \overline{1,5}$$

1. Tester ces fonctions sur les valeurs : -1, 0 et 1. Que peut-on conclure ?
2. Tracer dans la même figure l'allure des fonctions f_1 , f_2 et f_3 sur l'intervalle $[-1, 1]$ avec différentes couleurs et en rajoutant la légende, le titre et les axes.

Exercice 3. Représenter graphiquement la fonction

$$f(x) = \frac{\exp(x^2 + y^2)}{0.3 + x^2 + y^2}$$

sur le domaine $(x, y) \in [-1, 1]^2$. Utiliser pour cela la fonction `surf`.

Exercice 4. Afin de se familiariser davantage avec l'extension graphique `SIMULINK`, on vous propose de visualiser sur un scope les différents signaux d'entrées (échelon, rampe et sinusoïdale) qui seront fréquemment utilisés dans les prochains TPs.

TP 2 : Introduction aux systèmes linéaires continus asservis

2.1 Objectif

L'objectif de ce TP est d'introduire les principales définitions et notions nécessaires à l'étude des systèmes linéaires continus asservis.

2.2 Généralités sur les systèmes

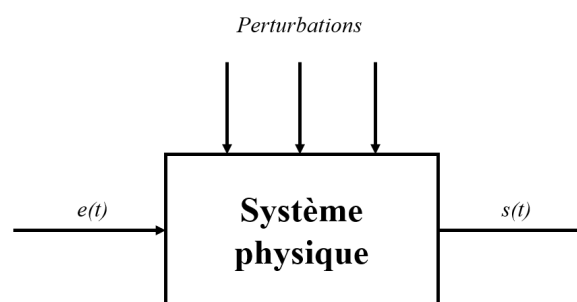
2.2.1 Système

On entend par un système un dispositif isolé soumis aux lois de la physique et caractérisé par certaines grandeurs.

Dans les systèmes monovariables, on s'intéresse à la relation entre une grandeur particulière ou entrée principale $e(t)$, correspondant à une action extérieure s'exerçant sur le système, et une grandeur de sortie caractérisant son état, que l'on désigne par $s(t)$: l'application au système des lois de la physique conduit à l'établissement d'une certaine relation entre $e(t)$ et $s(t)$.

Les autres grandeurs qui possèdent une action sur le système et qui sont susceptibles par conséquent de modifier la relation existante entre $e(t)$ et $s(t)$ sont appelées entrées parasites ou perturbations.

La relation entre l'entrée et la sortie est représentée symboliquement par le schéma suivant :



2.2.2 Système linéaire

Un système physique est linéaire si la relation entre la ou les grandeurs d'entrées et la ou les grandeurs de sortie est un système d'équations différentielles linéaires.

Si elle n'est strictement que rarement vérifiée en pratique, cette hypothèse de linéarité peut être acceptée pour de nombreux systèmes évoluant autour d'un point d'équilibre. Un processus de linéarisation est alors nécessaire. D'un point de vue purement technique, les systèmes linéaires vérifient le principe de superposition et le principe d'homogénéité.

Le principe de superposition

La réponse $s(t)$ d'un système linéaire à une entrée $e(t)$ composée de la combinaison linéaire de plusieurs entrées $e(t) = \sum_{k=1}^n \alpha_k e_k(t)$ est la somme des réponses élémentaires $s_k(t)$ à chacune des entrées individuelles $s(t) = \sum_{k=1}^n \alpha_k s_k(t)$.

Le principe d'homogénéité

Un système vérifiant le principe d'homogénéité si pour une entrée $\alpha e(t)$, la sortie est donnée par $\alpha s(t)$.

Ainsi l'hypothèse de linéarité va permettre l'utilisation d'outils (analytiques, graphiques) très simples et puissants tels que les transformées de Laplace et de Fourier.

Toutefois, l'hypothèse de linéarité est valide dans un domaine précis et ne tient pas compte d'un certains nombres de phénomènes purement non linéaires. En effet, la plupart des systèmes physiques sont en réalité non linéaires (bras de robot) ou font apparaître des phénomènes non linéaires (hystérésis, frottement sec). Il conviendra donc de toujours justifier en pratique l'hypothèse de linéarité et d'identifier son domaine de validité.

2.2.3 Système continu

Un système physique est dit continu si toutes les grandeurs qui le caractérisent sont de nature continue : l'information que représentent ces grandeurs est disponible en chaque instant et peut prendre toutes les valeurs possibles entre deux limites. Leur évolution dans le temps est un signal continu au sens mathématique du terme.

La figure (2.1) donne un aperçu sommaire d'une classification des modèles mathématiques qui feront l'objet de ce TP.

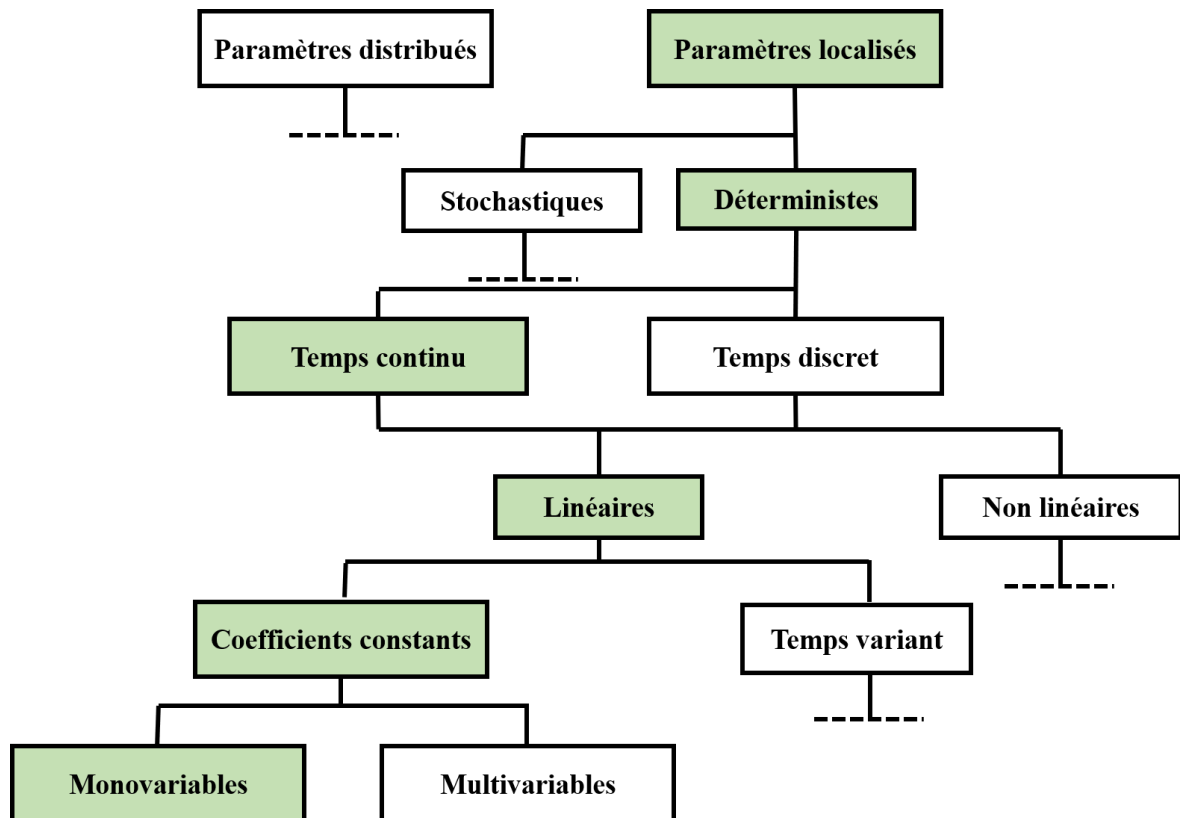


FIGURE 2.1: Nature mathématique des systèmes

2.3 Caractérisation des systèmes asservis

2.3.1 Structure des systèmes asservis

Un système asservi linéaire peut se représenter par le schéma (2.2). On y retrouve une chaîne d'action (chaîne directe) qui agit sur le système pour obtenir la sortie souhaitée en fonction de la consigne, une chaîne d'information (chaîne de retour) qui prélève une image de la sortie, cette image est comparée à la consigne à atteindre.

2.3.2 Objectifs de l'asservissement

Un système automatique est un système capable d'effectuer plusieurs opérations sans intervention de l'homme. La théorie de l'automatique permet de concevoir et d'analyser des asservissements en tenant compte de multiples objectifs (cahier des charges).

- ✓ Maintenir ou obtenir la stabilité.
- ✓ La précision.
- ✓ Obtenir des réponses rapides.

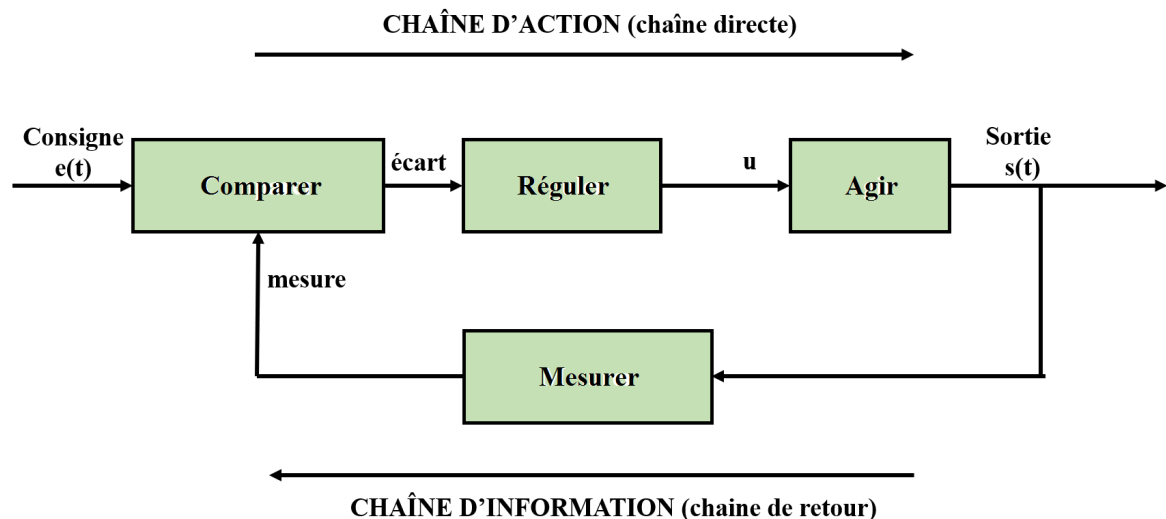


FIGURE 2.2: Structure d'un système asservi

2.3.3 Fonction d'un système asservi

✓ **Asservissement ou poursuite** : capacité pour le signal de sortie (signal à commander) de poursuivre un signal dit de consigne qui varie.

Exemple. Machine outil devant usiner une pièce selon un profil donné, missile qui poursuit une cible ...

✓ **Régulation** : dans ce cas, la consigne est fixé et le système de commande doit compenser l'effet de la perturbation. Réguler c'est maintenir la grandeur réglée la plus proche possible de la valeur de consigne en agissant sur la grandeur réglante.

Exemple. Réguler la température d'un four, le niveau d'eau d'un barrage ...

2.3.4 Notion de fonction de transfert et schéma-blocs

Fonction de transfert

La fonction de transfert est un modèle de comportement entrée/sortie qui s'obtient à partir de l'équation différentielle linéaire à coefficients constants. Plutôt que de chercher à obtenir $s(t)$ en fonction de $e(t)$, l'on cherche à obtenir $S(p) = L(s(t))$ en fonction de $E(p) = L(e(t))$. Comme il s'agit de déterminer un modèle qui soit indépendant des conditions initiales, ces dernières sont considérées nulles et l'on applique tout simplement la transformée de Laplace à l'équation différentielle suivante :

$$a_n \frac{d^n s(t)}{dt^n} + a_{n-1} \frac{d^{n-1} s(t)}{dt^{n-1}} + \dots + a_1 \dot{s}(t) + a_0 s(t) = b_m \frac{d^m e(t)}{dt^m} + b_{m-1} \frac{d^{m-1} e(t)}{dt^{m-1}} + \dots + b_1 \dot{e}(t) + b_0 e(t)$$

ce qui conduit à l'expression suivante :

$$\frac{S(p)}{E(p)} = G(p) = \frac{N(p)}{D(p)} = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0}$$

$G(p)$ est une fraction rationnelle dépendant de la variable de Laplace, de numérateur $N(p)$ et de dénominateur $D(p)$. Elle est appelée fonction de transfert et se révèle très utile pour l'étude des systèmes linéaires monovariables.

$G(p)$ est dite d'ordre n et selon la règle de causalité, l'on aura généralement $m \leq n$. L'on note que les racines de $N(p)$ sont appelées zéros du système et celles de $D(p)$ sont appelées pôles du système. Bien que les zéros aient une influence difficile à estimer sur le comportement du système, il convient surtout de se souvenir que les pôles ont une influence encore plus grande en termes de stabilité et de régime transitoire de la réponse du système. Cette influence des pôles est beaucoup plus facile à prévoir.

Remarque 2.3.1

Un système asservi linéaire est stable si les parties réelles des pôles (solution de $D(p) = 0$) sont négatives.

La Control Toolbox permet de définir des objet de type `tf`, c'est-à-dire *transfert function*. Elle prend en argument deux vecteurs contenant les coefficients des polynômes du numérateur et du dénominateur. Ainsi une fonction de transfert sera définie par la commande `tf(num,den)`.

Exemple 2.3.2

Soit le système décrit par

$$G(p) = \frac{2p + 1}{p^2 + 2p + 1}$$

```
» G=tf([2 1],[1 2 1])
```

On peut obtenir diverses informations sur le système défini par sa représentation.

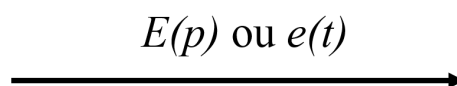
```
» pole(G) % donne les pôles du système
» zero(G) % donne les zéros du système
» zpk(G) % convertie n'importe quel système dynamique en une
représentation zéro - pôle - gain
» pzmap(G) % place les pôles et les zéros dans le plan complexe
```

Schéma fonctionnel

Le schéma fonctionnel d'un système est une représentation graphique des fonctions de chaque composant élémentaire constituant le système ainsi que le flux des signaux utiles.

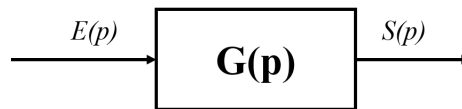
Un schéma fonctionnel est composé :

✓ **d'arcs orientés.**



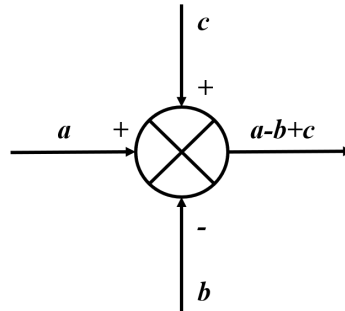
L'arc orienté représente le flux de signaux donnés regroupés dans un vecteur.

✓ de blocs fonctionnels.



Le bloc fonctionnel est le symbole représentant l'opération mathématique appliquée à l'entrée du bloc et produisant sa sortie. Cette opération mathématique est très souvent mais pas exclusivement représentée par une fonction de transfert.

✓ de blocs sommateurs.



Le bloc sommateur traduit une relation purement algébrique entre les signaux d'entrée et de sortie.

Remarque 2.3.3

- Un schéma fonctionnel n'est pas unique.
- Il est indépendant de la nature physique du système modélisé.

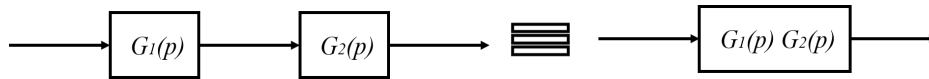
Principes de construction

1. Écrire les équations de la physique associées à chaque élément constituant le système.
2. En appliquant la transformée de Laplace, calculer la fonction de transfert associée à chaque élément en supposant les conditions initiales nulles.
3. Identifier les relations inter-signaux et les relations signaux-blocs pour tracer le schéma fonctionnel.

Opérations sur les fonctions de transfert

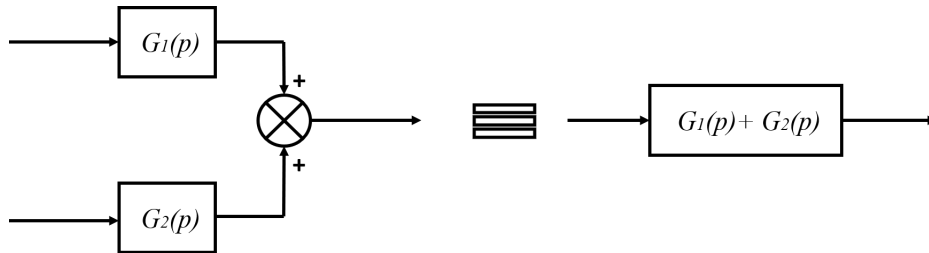
Il est possible d'assembler les fonctions de transfert entre-elles de plusieurs manières. Soient G_1 et G_2 les représentations de deux systèmes. Les combinaisons de base sont :

En série :



» $G1 \cdot G2$ ou » `series(G1,G2)`

En parallèle :



» $G1 + G2$ ou » `parallel(G1,G2)`

Fonction de transfert en boucle ouverte et en boucle fermée

Le schéma d'un système asservi donné par la figure (2.2) fait apparaître un bouclage de la sortie $s(t)$ sur l'entrée $e(t)$, réalisant ainsi un système de contre-réaction ou système bouclé (asservi) du type de la figure (2.3).

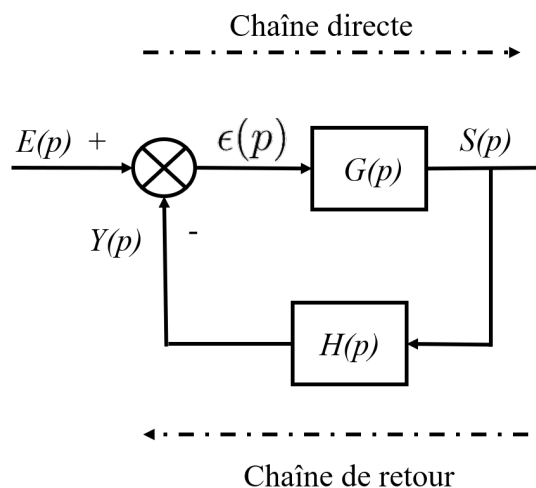


FIGURE 2.3: Asservissement à contre-réaction

On définit $E(p)$ comme l'entrée de consigne, $Y(p)$ comme la sortie mesurée et $\epsilon(p)$ comme l'erreur.

Fonction de Transfert en Boucle Ouverte (FTBO). La FTBO ou gain de boucle est définie comme :

$$\frac{Y(p)}{E(p)} = G(p)H(p)$$

Fonction de Transfert en Boucle Fermée (FTBF). La FTBF est définie comme :

$$\frac{S(p)}{E(p)} = \frac{G(p)}{1 + G(p)H(p)}$$

Elle correspond au transfert global de la boucle d'asservissement.

» `feedback(G,H,-1)` ou » `feedback(G,H) % G bouclé par H`

Un cas particulier que l'on rencontre fréquemment est celui des systèmes bouclés à retour unitaire.

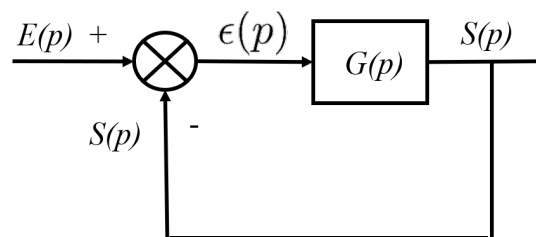


FIGURE 2.4: Système bouclé à retour unitaire

Un système bouclé est dit à retour unitaire si le transfert de la chaîne de retour est égal à un. La fonction de transfert en boucle ouverte est alors égale au transfert de la chaîne directe $G(p)$ et la fonction de transfert en boucle fermée est donnée par :

$$\frac{S(p)}{E(p)} = \frac{G(p)}{1 + G(p)}$$

» `feedback(G,1)`

2.4 Énoncé du TP

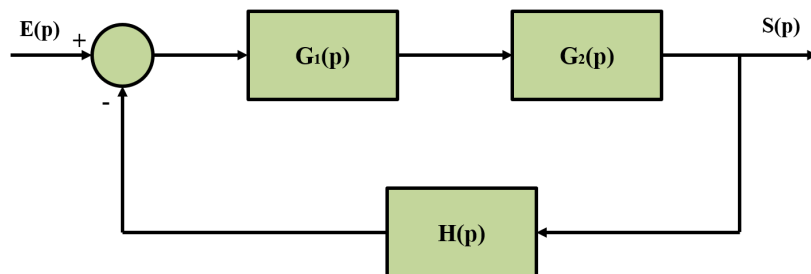
2.4.1 Partie théorique

On considère les systèmes régissent par les équations différentielles suivantes :

$$\frac{d^2s(t)}{dt^2} + 4\frac{ds(t)}{dt} + 4s(t) = \frac{d^2e(t)}{dt^2} + 3\frac{de(t)}{dt} + 2e(t)$$

$$\frac{d^2s(t)}{dt^2} + 3\frac{ds(t)}{dt} + 2s(t) = 5\frac{de(t)}{dt} + 5e(t)$$

1. Calculer les fonctions de transfert G_1 et G_2 de chaque système et calculer ses pôles et ses zéros.
2. Réécrire G_1 et G_2 sous la forme factorisée.
3. Trouver les fonctions de transfert en série G_s et en parallèle G_p .
4. Soit le schéma bloc suivant :



- (a) Déterminer la fonction de transfert T_1 en boucle fermée de ce schéma avec $H(p) = \frac{1}{p+1}$.
- (b) Calculer les pôles de T_1 puis conclure sur la stabilité.
- (c) Déterminer la fonction de transfert T_2 en boucle fermée de ce schéma avec $H(p) = 1$.
- (d) Calculer les pôles de T_2 puis conclure sur la stabilité.

2.4.2 Partie pratique

Dans un fichier script :

1. Faire entrer les deux fonctions de transfert G_1 et G_2 , puis G_s et G_p avec les commandes adéquates. Que peut-on dire sur les résultats trouvés dans la partie théorique.
2. Utiliser l'instruction `minreal` pour calculer G_s et G_p . Commentez les résultats ?
3. En utilisant l'instruction `feedback`, déterminer la FTBF T_1 puis T_2 .
4. Utiliser l'instruction `minreal` pour calculer T_1 puis T_2 . Que peut-on dire sur la stabilité des deux systèmes ?
5. Tracer les réponses indicielles de T_1 et T_2 en utilisant la commande `step`.
6. Reprendre le schéma bloc défini dans la partie théorique en utilisant l'extension graphique `SIMULINK` avec $H(p) = \frac{1}{p+1}$ puis $H(p) = 1$.

TP 3 : Analyse des systèmes linéaires continus asservis du premier ordre

3.1 Objectif

Dans ce TP, on commence par introduire les éléments de l'analyse des systèmes asservis du premier ordre. Pour un système donné que l'on projette d'asservir, il se pose un certain nombre de questions :

Est-il suffisamment précis ? Est-il stable ? Comment se comporte-t-il durant le régime transitoire ? Est-il suffisamment rapide ? Quelles sont ses sensibilités à diverses perturbations ?

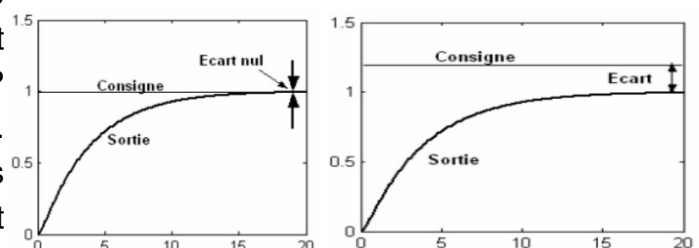
Pour répondre à ces questions, il faut effectuer :

1. **L'analyse Temporelle** : elle permet de déterminer le temps de réponse, le dépassement, le temps de montée, les erreurs en régime permanent et la stabilité.
2. **L'analyse Fréquentielle** : elle permet de déterminer les fréquences de coupure, la bande passante, la stabilité et les marges de gain et de phase.

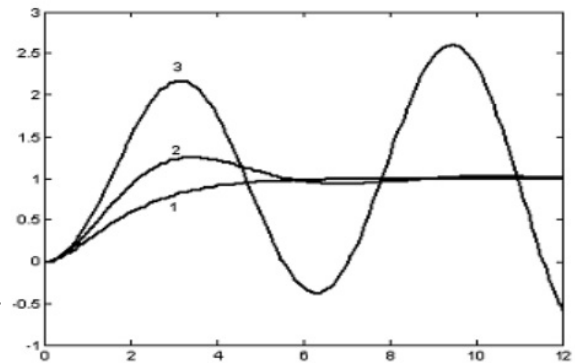
3.2 Performances des systèmes asservis

Dans le TP précédent, nous avons défini globalement les objectifs d'un asservissement comme étant l'amélioration des performances (précision, stabilité et rapidité). Dans ce TP, nous allons définir plus précisément ces performances.

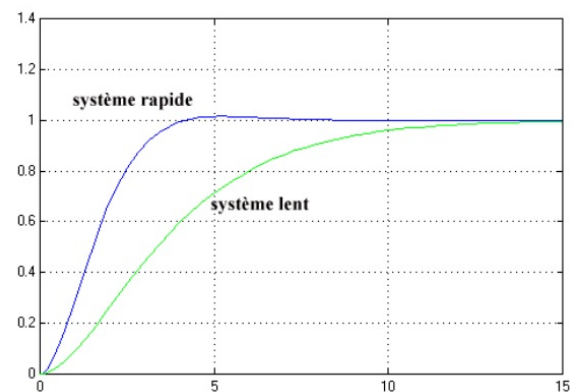
✓ **La précision.** Pour un type d'entrée donné, quel est l'écart entre la consigne et la sortie ? (erreurs statique et de traînage). Plus l'écart entre ces grandeurs est petit, plus l'asservissement est précis.



✓ **La stabilité.** On dit qu'un asservissement est stable si pour une consigne bornée en amplitude, tous les autres signaux sont aussi bornés en amplitude. Les courbes 1 et 2 de la figure ci-dessous sont caractéristiques d'un asservissement stable. En effet, pour une entrée constante, la sortie évolue et se stabilise à une valeur finale constante. La courbe 3 est caractéristique d'un asservissement instable : la sortie diverge. Une boucle instable est une boucle inutilisable.



✓ **La rapidité.** Plus le temps de réponse est faible, plus l'asservissement est dit rapide. La rapidité d'un système asymptotiquement stable se mesure par la durée de son régime transitoire. Il est pratique de la mesurer grâce à la réponse indicielle. Pour un système asymptotiquement stable, plus la réponse indicielle converge vite vers la valeur finale, plus le système est rapide.



3.3 Caractérisation d'un système du premier ordre

Définition 3.3.1

Un système est dit *de premier ordre*, lorsqu'il est régi par une équation différentielle linéaire à coefficients constants du premier ordre de la forme :

$$\tau \frac{ds(t)}{dt} + s(t) = Ke(t) \quad (3.1)$$

où τ et K sont des constantes réelles non nulles ; τ est la constante de temps du système et K son gain statique.

Ainsi, en appliquant la transformation de Laplace à l'équation différentielle précédente (avec des conditions initiales nulles) on obtient :

$$G(p) = \frac{K}{1 + \tau p}$$

D'où la fonction de transfert d'un système du premier ordre.

3.3.1 Réponse indicielle (ou réponse à un échelon)

Traditionnellement, on caractérise le comportement temporel du système par sa réponse indicielle. La solution de l'équation différentielle pour $u(t) = \mathcal{U}(t)$ s'écrit :

$$y(t) = \alpha + \beta e^{-t/\tau} \quad (3.2)$$

avec α et β deux constantes réelles dépendant des CI. A l'instant $t = 0$, l'équation (3.2) donne :

$$y(0) = \alpha + \beta. \quad (3.3)$$

On n'étudie que le cas où $\tau > 0$, de sorte que le système se stabilise pour t tendant vers l'infini. Lorsque le système a atteint sa valeur finale $\frac{dy(t)}{dt} = 0$, ce que l'on peut supposer vrai pour t tendant vers l'infini. Alors, d'après (4.1) et (3.2) :

$$\lim_{t \rightarrow +\infty} y(t) = K \text{ et } \lim_{t \rightarrow +\infty} \frac{dy(t)}{dt} = 0$$

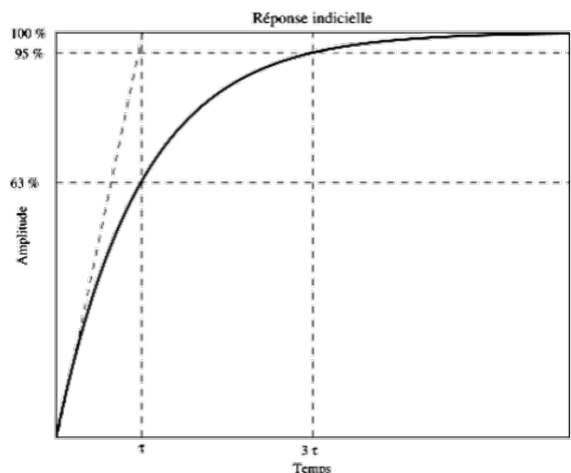
Il en résulte que $\alpha = K$, puis, en utilisant (3.3), que $\beta = y(0) - K$. Finalement :

$$y(t) = K + (y(0) - K) e^{-t/\tau}$$

Cette réponse comporte :

- un terme constant : le gain statique K , qui représente le régime permanent de la sortie (**steady state**) ;
- une partie variable : $(y(0) - K)e^{-t/\tau}$, qui représente le régime transitoire (**transient response**).

La valeur finale atteinte en régime permanent par $y(t)$ est k fois la valeur de l'entrée. Pour $t = \tau$, $y(\tau)$ atteint 63% de la valeur finale. Pour $t = 3\tau$, $y(3\tau)$ atteint 95% de la valeur finale c'est-à-dire la valeur finale à 5% près. La pente à l'origine de la réponse indicielle vaut quant à elle $\frac{K}{\tau}$.



Sous MATLAB, la réponse indicielle est obtenue en utilisant la commande

```
»step(G)
```

En cliquant sur le tracé, bouton droit, il est possible de modifier les propriétés du graphe et surtout d'obtenir des renseignements sur le temps de réponse, temps de montée, l'erreur ...

3.3.2 Réponse harmonique

La description de la réponse harmonique d'un système se fait aisément en étudiant son comportement asymptotique en régime permanent sinusoïdal. L'analyse de la réponse harmonique se fait en complexe ($p=j\omega$). Ainsi, pour un système du premier ordre

$$G(j\omega) = \frac{K}{1 + j\omega\tau}$$

En générale les paramètres étudiés sont le gain et le déphasage :

$$\text{Gain} = |G(j\omega)| \quad \text{Déphasage} = \arg(G(j\omega))$$

que l'on représente sous forme de diagramme de Bode, Black et Nyquist.

Diagramme de Bode

La représentation dans le plan de Bode d'une fonction de transfert $G(j\omega)$ consiste à tracer le module en décibels $G_{dB} = 20 \log |G(j\omega)|$ et la phase en degrés $\varphi = \arg(G(j\omega))$ en fonction de la pulsation ω en (rad/s) portée sur une échelle logarithmique. Le comportement asymptotique de la réponse harmonique est résumé dans le tableau suivant :

ω	$G(j\omega)$ équivalent	gain	gain (dB)	phase (deg)
$\omega \ll \frac{1}{\tau}$, soit $\omega\tau \ll 1$	K	K	$K_{dB} = 20 \log_{10} K$	0
$\frac{1}{\tau}$	$\frac{K}{1+j}$	$\frac{K}{\sqrt{2}}$	$K_{dB} - 3$	-45
$\omega \gg \frac{1}{\tau}$, soit $\omega\tau \gg 1$	$-\frac{jK}{\tau\omega}$	$\frac{K}{\tau\omega}$	$K_{dB} = -20 \log_{10} \tau - 20 \log_{10} \omega$	-90

TABLE 3.1: Réponse harmonique d'un système du premier ordre

On remarque que pour les pulsations grandes devant $\omega_c = \frac{1}{\tau}$, la courbe du gain en dB suit une direction asymptotique qui est une droite de pente -20 dB/décade dans le diagramme de Bode (voir figure 3.1). A ces pulsations le gain diminue en effet de 20 dB chaque fois que la pulsation est multipliée par 10. La pulsation ω_c est appelée pulsation de coupure. Pour un système du premier ordre elle correspond à une atténuation du gain de 3 dB par rapport au gain aux faibles pulsations et à un déphasage de -45 deg. La bande passante (bandwidth) du système est donnée par la fréquence $f_c = \frac{\omega_c}{2\pi}$ associée à cette pulsation de coupure.

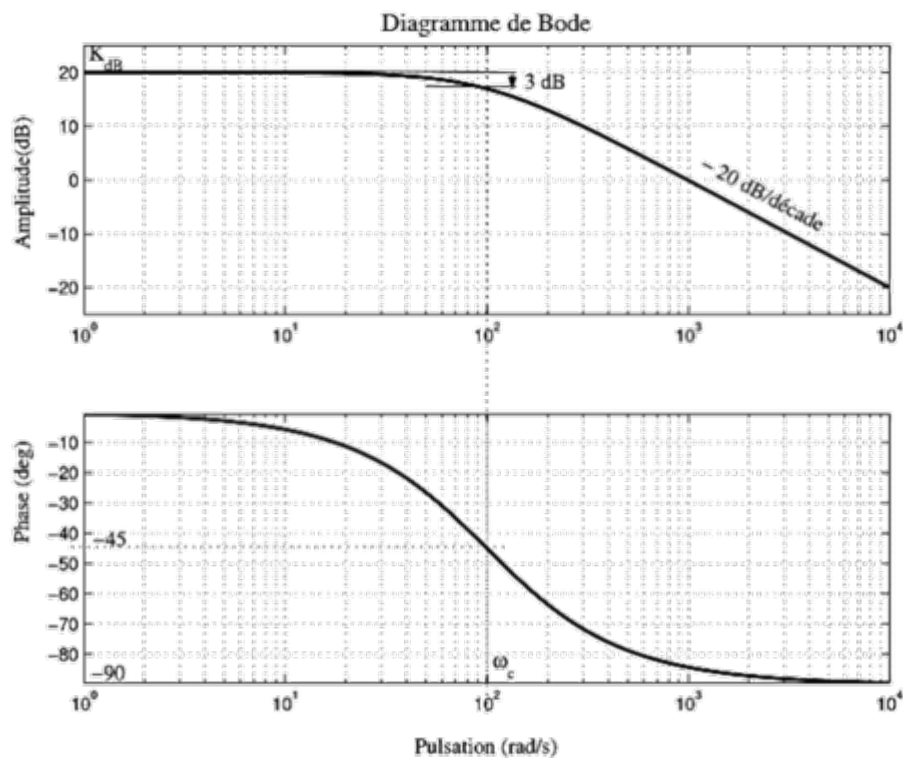


FIGURE 3.1: Diagramme de Bode d'un système du premier ordre

Le tracé du diagramme de Bode peut être obtenu simplement en utilisant la fonction :

```
»bode(G) % Trace le diagramme de Bode
```

Diagramme de Nyquist

Le diagramme de Nyquist est la représentation de la fonction de transfert dans le plan complexe. Il est défini comme la représentation de la partie imaginaire de $G(j\omega)$ en fonction de sa partie réelle dans un repère cartésien. Pour obtenir le diagramme de Nyquist à l'aide MATLAB, procéder comme suit :

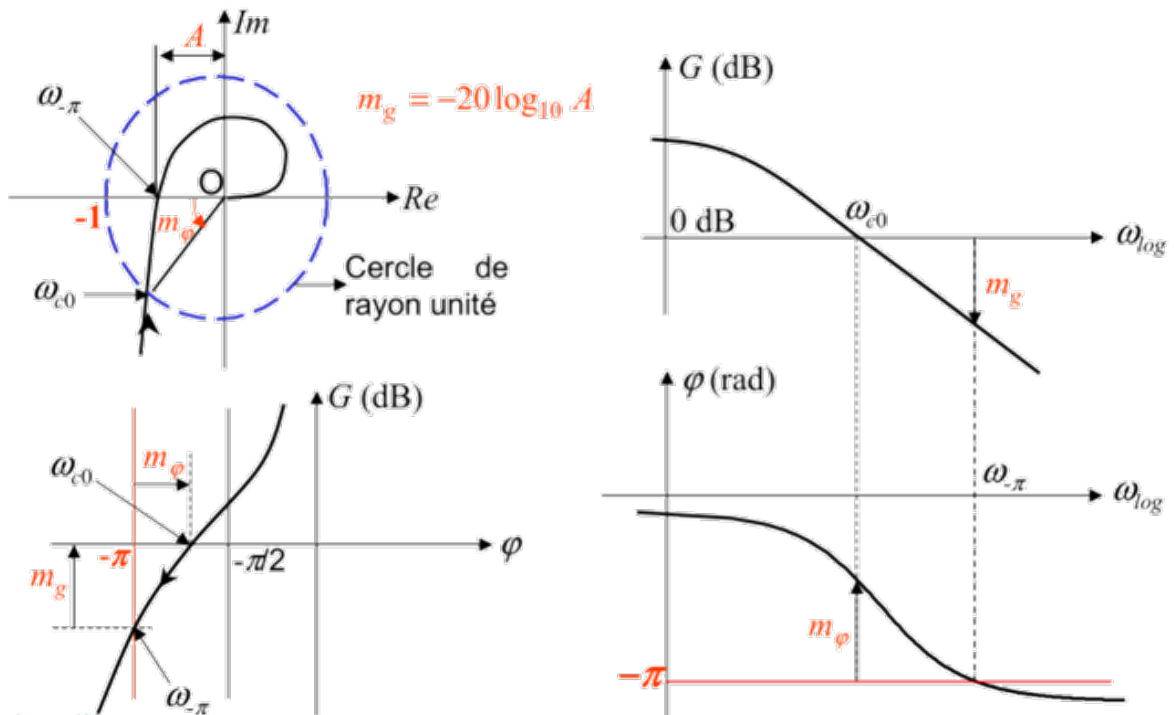
```
»nyquist(G) % Trace le diagramme de Nyquist
```

Diagramme de Black

Le diagramme de Black représente la courbe du modèle de la fonction de transfert en décibels en fonction de sa phase en degré. Pour tracer le diagramme de Black, taper :

```
»nichols(G) % Trace le diagramme de Black-Nichols
```

Marges de stabilité



- Si $m_g > 0$ et $m_\varphi > 0$ alors le système est stable.
- Si $m_g < 0$ et $m_\varphi < 0$ alors le système est instable.
- Si $m_g = 0$ et $m_\varphi = 0$ alors le système est à la limite de stabilité.

Les marges de gain et de phase sont déterminées à l'aide de la fonction :

```
»margin(G)
```

Cette fonction trace le diagramme de Bode et affiche ces marges avec les pulsations correspondantes.

3.4 Énoncé du TP

3.4.1 Partie théorique

Nous nous intéressons dans cette partie au montage représenté à la figure (3.2).

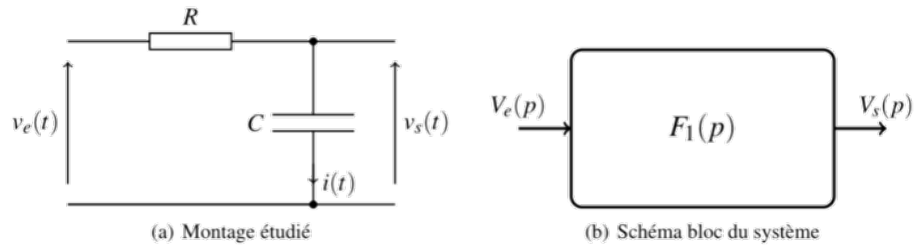


FIGURE 3.2: Circuit RC

1. Quelle est la relation liant le courant $i(t)$ et la tension $v_s(t)$?
2. À partir de la loi des mailles, exprimer l'équation différentielle du premier ordre liant $v_e(t)$ et $v_s(t)$.
3. À l'aide des propriétés de la transformée de Laplace, déterminer la fonction de transfert liant $V_e(p)$ et $V_s(p)$. La tension de sortie à l'instant $t = 0$ est considérée nulle.
4. La forme standard d'une fonction de transfert du premier ordre est la suivante :

$$F_1(p) = \frac{K}{1 + \tau p}$$

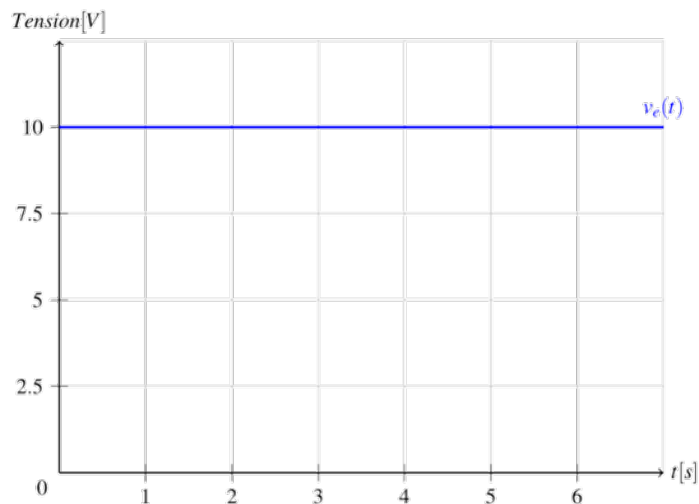
Exprimer sous cette forme la fonction de transfert $F_1(p) = \frac{V_s(p)}{V_e(p)}$ définie à la question précédente. Donner l'expression des constantes K et τ .

5. La tension d'entrée est constante : $v_e(t) = E$. A l'aide des transformées de Laplace usuelles, déterminer la transformée de Laplace de la tension d'entrée ($V_e(p)$).
6. Déterminer l'expression de la tension de sortie $V_s(p)$.
7. A l'aide des transformées de Laplace usuelles, déterminer l'expression de la tension de sortie $v_s(t)$.
8. **Application numérique** : $E = 10V$, $R = 1000\Omega$, $C = 1mF$. Déterminer la valeur de la tension de sortie pour $t = 0$, $t = \tau$, $t = 3\tau$ et $t = 5\tau$.

Valeurs de la tension de sortie :

t	$v_s(t)$	Signification par rapport à la valeur finale
0		
τ		
3τ		
5τ		

9. Tracer ci-dessous l'évolution de la tension de sortie en fonction du temps.



On s'intéresse cette fois-ci au système $F_1(p)$, placé dans une boucle à retour unitaire, représenté dans la figure (3.3).

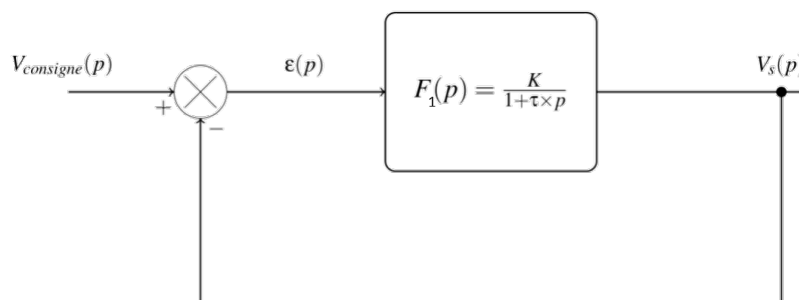


FIGURE 3.3: Système du premier ordre en boucle fermé

On cherche à exprimer la fonction de transfert en boucle fermée :

$$F_2(p) = \frac{V_s(p)}{V_{consigne}(p)}$$

1. Exprimer $V_s(p)$ en fonction de $\epsilon(p)$ et $F_1(p)$.
2. Exprimer $\epsilon(p)$ en fonction de $V_s(p)$ et $V_{consigne}(p)$.
3. Dédire des deux relations précédentes la fonction de transfert en boucle fermée $F_2(p)$.
4. Mettre $F_2(p)$ sous la forme suivante :

$$F_2(p) = \frac{K_{BF}}{1 + \tau_{BF} \times p}$$

Vous donnerez les expressions de K_{BF} et τ_{BF} .

5. À partir des valeurs de $\tau = 1$ et $K = 2$, donner les valeurs de K_{BF} et τ_{BF} .

Le même système $F_1(p)$ est désormais placé au sein d'une boucle d'asservissement représentée à la figure (3.4). La grandeur à asservir est désormais mesurée par un capteur (de gain C) avant d'être envoyée dans le comparateur.

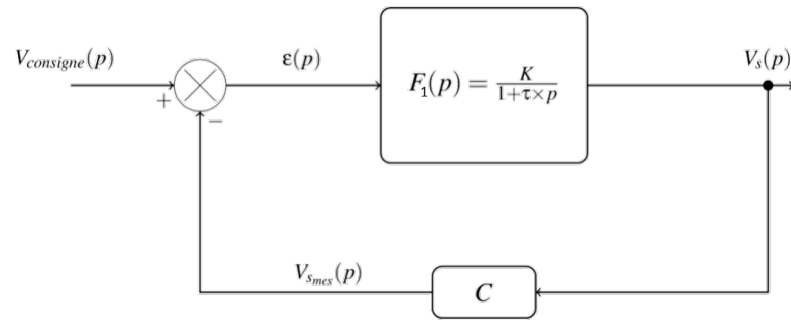


FIGURE 3.4: Capteur dans la boucle de retour

1. Déterminer la fonction de transfert en boucle fermée :

$$F_3(p) = \frac{V_s(p)}{V_{consigne}(p)}$$

Vous donnerez les expressions des nouvelles constantes K_{BF} et τ_{BF} .

2. **Application numérique** : le gain du capteur est de 0.5. Déterminer les valeurs numériques des constantes K_{BF} et τ_{BF} .

3.4.2 Partie pratique

Créer un fichier script MATLAB qui :

1. ferme les fenêtre ouvertes, efface les variables existantes,
2. contient deux lignes sur lesquelles on définit K et τ ,
3. définit automatiquement la fonction de transfert du premier ordre $F_1(p)$,
4. définit la fonction de transfert en boucle fermée $F_2(p)$,
5. définit la nouvelle fonction de transfert en boucle fermée avec capteur $F_3(p)$,
6. affiche les 3 réponses à un échelon unitaire pour $K = 2$ et $\tau = 1$ dans la même figure.
 - ✘ Parmi les deux systèmes $F_1(p)$ et $F_2(p)$ qui est le plus rapide et pourquoi ?
 - ✘ Parmi les trois systèmes $F_1(p)$, $F_2(p)$ et $F_3(p)$ qui est le plus précis et pourquoi ?
 - ✘ Visualiser les trois systèmes $F_1(p)$, $F_2(p)$ et $F_3(p)$ sur le même Scope puis recopier le schéma bloc sur la feuille.

TP 4 : Analyse des systèmes linéaires continus asservis du second ordre

4.1 Objectif

Le but de ce TP consiste à faire une analyse complète d'un système du second ordre dans les domaines temporel et fréquentiel.

4.2 Caractérisation d'un système du second ordre

Définition 4.2.1

Un système linéaire invariant à temps continu d'ordre 2 est décrit par une équation différentielle d'ordre 2 à coefficients constants reliant son entrée $u(t)$ et sa sortie $y(t)$. On considère des systèmes dont l'équation différentielle se met sous la forme canonique

$$\omega_n^2 y(t) + 2\xi\omega_n \frac{dy(t)}{dt} + \frac{d^2 y(t)}{dt^2} = K\omega_n^2 u(t) \quad (4.1)$$

où ξ et K sont des constantes réelles strictement positives et ω_n une constante réelle non nulle. Pour ce système ξ est le coefficient d'amortissement du système (damping ratio), ω_n sa pulsation naturelle (natural frequency) ou pulsation propre non amortie et K son gain statique.

Ainsi, en appliquant la transformation de Laplace à l'équation différentielle (4.1) (avec des conditions initiales nulles) on obtient :

$$G(p) = \frac{Y(p)}{U(p)} = \frac{K\omega_n^2}{\omega_n^2 + 2\xi\omega_n p + p^2}$$

D'où la fonction de transfert d'un système du second ordre. Cette forme particulière facilite le calcul des pôles du système.

En effet les pôles du système sont les solutions de l'équation

$$\omega_n^2 + 2\xi\omega_n p + p^2 = 0$$

qui valent :

$$p_{1,2} = -\left(\xi \pm j\sqrt{1-\xi^2}\right)\omega_n \text{ si } 0 < \xi \leq 1$$

$$p_{1,2} = -\left(\xi \pm j\sqrt{\xi^2-1}\right)\omega_n \text{ si } \xi \geq 1$$

On peut représenter graphiquement ces pôles dans le plan complexe, figure (4.1), comme cela est fait à la figure ci-contre. D'après ce schéma, les notions de coefficient d'amortissement et de pulsation naturelle peuvent être étendues à un pôle quelconque, quel que soit le degré du système.

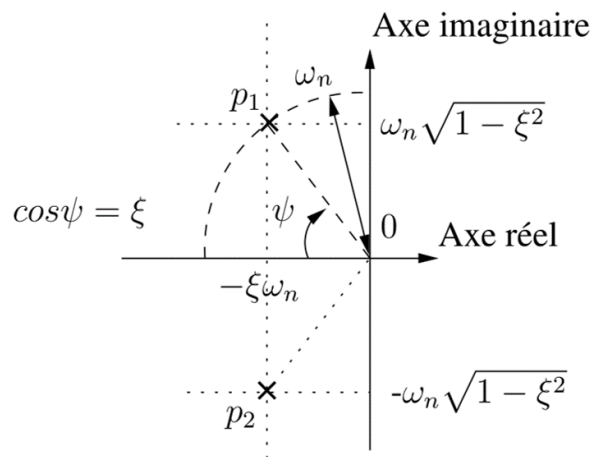


FIGURE 4.1: Les pôles dans le plan complexe

4.2.1 Réponse indicielle (ou réponse à un échelon)

La solution de l'équation (4.1) pour $u(t) = \mathcal{U}(t)$ dépend de la valeur de ξ , à l'image des pôles du système. Selon la valeur du coefficient d'amortissement ξ , la réponse indicielle du système prend différentes formes :

$$\begin{cases} \alpha + \beta e^{-\xi\omega_n t} \sin\left(\sqrt{1-\xi^2}\omega_n t + \varphi\right) & \text{si } 0 < \xi < 1 \\ \alpha + (\beta + \gamma t) e^{-\xi\omega_n t} & \text{si } \xi = 1 \\ \alpha + \beta e^{-(\xi+\sqrt{\xi^2-1})\omega_n t} + \gamma e^{-(\xi-\sqrt{\xi^2-1})\omega_n t} & \text{si } \xi > 1 \end{cases}$$

avec α, β et $\gamma \in \mathbb{R}$ dépendant des conditions initiales. Il serait un peu fastidieux d'établir toutes les expressions en fonction des conditions initiales. On retiendra donc la forme des différentes réponses pour des conditions initiales nulles.

4.2.2 Réponse harmonique

La réponse harmonique d'un tel système est de la forme

$$G(j\omega) = \frac{K\omega_n^2}{\omega_n^2 - \omega^2 + 2j\xi\omega_n\omega}$$

Son comportement asymptotique de la réponse harmonique est résumé dans le tableau suivant :

ω	$G(j\omega)$ équivalent	gain	gain (dB)	phase (deg)
$\omega \ll \omega_n$	K	K	$K_{dB} = 20 \log_{10} K$	0
ω_n	$\frac{K}{2j\xi}$	$\frac{K}{2\xi}$	$K_{dB} - 6 - 20 \log_{10} \xi$	-90
$\omega \gg \omega_n$	$-\frac{K\omega_n^2}{\omega^2}$	$\frac{K\omega_n^2}{\omega^2}$	$K_{dB} + 40 \log_{10} \omega_n - 40 \log_{10} \omega$	-180

TABLE 4.1: Réponse harmonique d'un système du second ordre

On remarque que pour les pulsations grandes devant ω_n , la courbe de gain suit une direction asymptotique qui est une droite de pente -40 dB/décade dans le diagramme de Bode. A la pulsation ω_n le déphasage vaut -90 degré et le gain dépend de la valeur de ξ et l'on peut, selon le cas, avoir atténuation ou, au contraire, résonance. Pour cela il faut que ξ soit inférieur à 0.7. Alors, à la pulsation $\omega = \sqrt{1 - 2\xi^2}\omega_n$, le gain passe par une valeur maximale supérieur au gain statique.

4.3 Énoncé du TP

4.3.1 Partie théorique

Nous nous intéressons dans cette partie au montage représenté à la figure (4.2).

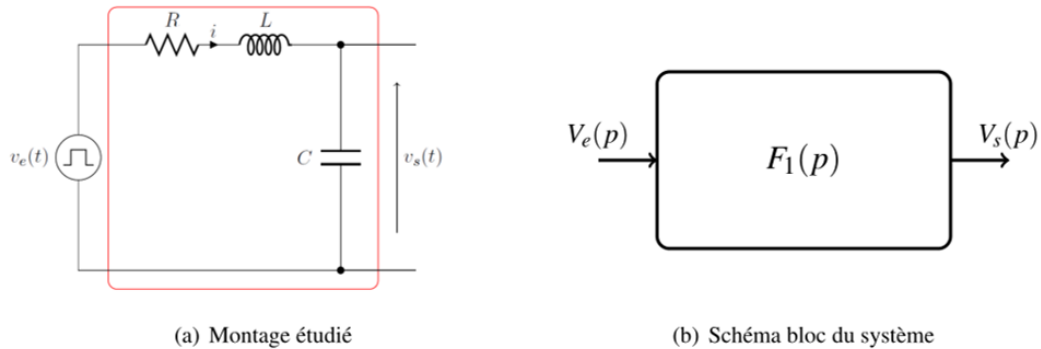


FIGURE 4.2: Circuit RLC

1. Quelle est la relation liant le courant $i(t)$ et la tension $v_s(t)$? Quelle est la relation liant la tension aux bornes de l'inductance et le courant $i(t)$.
2. À partir de la loi des mailles, exprimer l'équation différentielle du second ordre liant $v_e(t)$ et $v_s(t)$.
3. À l'aide des propriétés de la transformée de Laplace, déterminer la fonction de transfert liant $V_e(p)$ et $V_s(p)$. La tension de sortie à l'instant $t = 0$ est considérée nulle.
4. La forme standard d'une fonction de transfert du premier ordre est la suivante :

$$F_1(p) = \frac{K}{1 + \frac{2\xi}{\omega_n}p + \frac{1}{\omega_n^2}p^2}$$

Exprimer sous cette forme la fonction de transfert $F_1(p) = \frac{V_s(p)}{V_e(p)}$ définie à la question précédente. Vous donnerez l'expression des constantes K , ω_n et ξ . Afin d'asservir un système, il est impératif de savoir l'identifier. Une manière usuelle de procéder est de solliciter le système avec un échelon indiciel en entrée et d'observer comment évolue la sortie. La réponse indicielle d'un système $F_1(p)$ est représentée à la figure (4.3).

5. Déterminer le gain statique K du système.
6. Mesurer la valeur du premier dépassement du système.
7. A l'aide du graphique ci-dessous, déterminer la valeur du coefficient d'amortissement ξ du système.
8. Mesurer la valeur de la pseudo période T_p du système, figure (4.3).

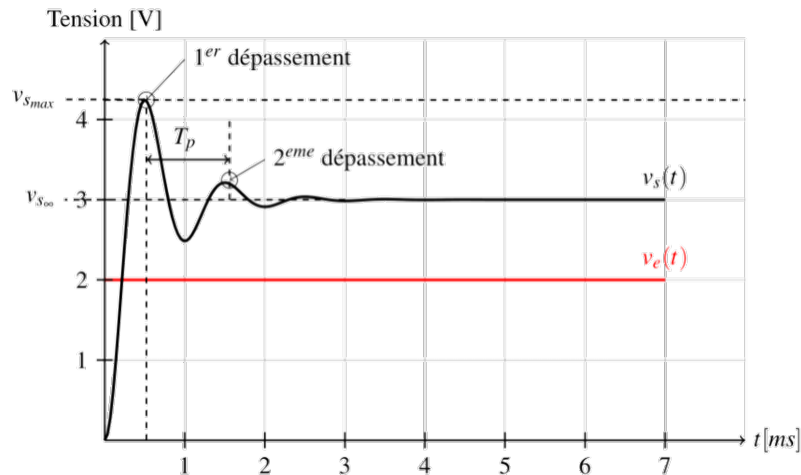
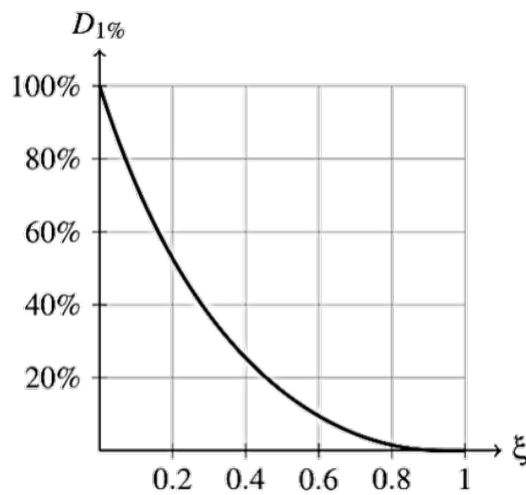


FIGURE 4.3: Réponse indicelle d'un système du second ordre



9. En déduire, à l'aide de la relation ci-dessous, la valeur de la pulsation propre du système ω_n .

$$\omega_n = \frac{2\pi}{T_p \times \sqrt{1 - \xi^2}}$$

4.3.2 Partie pratique

Créer un fichier script MATLAB qui :

1. ferme les fenêtres ouvertes, efface les variables existantes,
2. contient trois lignes sur lesquelles on définit K , ξ et ω_n
3. définit automatiquement la fonction de transfert du second ordre $F_1(p)$,
4. calcule les pôles et les zéros du système $F_1(p)$, et les trace dans le plan complexe (pzmap), que peut-on dire sur la stabilité du système ?
5. affiche la réponse à un échelon d'amplitude E pour les valeurs de K , ξ et ω_n trouvées plus haut.
6. affiche le diagramme de Bode du système F_1 pour les valeurs de K , ξ et ω_n trouvées plus haut.

7. afin de voir l'influence des paramètres K , ξ et ω_n sur le système, on propose de fixer deux paramètres et varier le troisième.
 - (a) En fixant $\xi = 0.28$ et $\omega_n = 6544$, visualiser la réponse indicielle ainsi que le tracé de Bode pour des valeurs différentes de $K = 1, 1.5, 3$ et 5 . Que peut-on conclure ?
 - (b) En fixant $K = 1.5$ et $\omega_n = 6544$, visualiser la réponse indicielle ainsi que le tracé de Bode pour des valeurs différentes de $\xi = 0.28, 0.4, 0.7$ et 1 . Que peut-on conclure ?
 - (c) En fixant $\xi = 0.28$ et $K = 1.5$, visualiser la réponse indicielle ainsi que le tracé de Bode pour des valeurs différentes de $\omega_n = 2000, 6544$ et 8000 . Que peut-on conclure ?
8. Visualiser dans le même Scope les réponses indicielles en fixant $\xi = 0.28$ et $\omega_n = 6544$ et en variant $K = 1.5, 3, 5$.
9. Visualiser dans le même Scope les réponses indicielles en fixant $K = 1.5$ et $\omega_n = 6544$ et en variant $\xi = 0.28, 0.7, 1$.
10. Visualiser dans le même Scope les réponses indicielles en fixant $\xi = 0.28$ et $K = 1.5$ et en variant $\omega_n = 2000, 6544, 8000$.

TP 5 : Correction des systèmes linéaires continus asservis

5.1 Objectif

On s'intéresse dans cette partie à l'étude des actions du module **PID** composé par les trois actions de base. On mettra en évidence l'effet produit par chaque action dans une boucle de régulation, ses avantages ainsi que ses limitations.

5.1.1 Correcteur proportionnel (P)

Le correcteur le plus simple est le correcteur proportionnel qui commande l'actionneur à l'aide d'un signal proportionnel au signal d'erreur. Sa fonction de transfert est :

$$C(p) = K_p$$

L'augmentation de K_p améliore la précision statique et peut aussi améliorer les performances en dynamique mais un gain trop important peut déstabiliser un système en boucle fermée.

Exemple. Soit un modèle qui peut représenter deux systèmes mécaniques, composé d'une masse m , un ressort de constante de raideur k , d'un effet de frottements de coefficient b . On souhaite contrôler l'élongation $x(t)$ lorsqu'on applique une force extérieure $F(t)$.

L'équation du modèle du système est donnée par :

$$m\ddot{x} + b\dot{x} + kx = F$$

1. Trouver la fonction de transfert G de ce système entre le déplacement x et l'entrée F .

Posons : $M = 1kg$, $b = 10Ns/m$, $k = 20N/m$ et $F = 1$.

2. Afficher la réponse indicielle du système en boucle fermée. Déterminer le gain statique, le temps de montée et le temps de réponse.

On souhaite concevoir un correcteur qui réduira le temps de montée, le temps de réponse et éliminera l'erreur statique.

On commencera par essayer la commande proportionnelle.

- Calculer la fonction de transfert $F1$ en boucle fermée du système avec un correcteur proportionnel.
- Afficher la réponse indicielle de $F1$, en prenant $K_p = 350$.
- Donner des conclusions.

5.1.2 Correcteur proportionnel et Intégral (PI)

La fonction de transfert de ce correcteur est :

$$C(p) = K_p + \frac{K_i}{p}$$

Il supprime l'erreur de position en régime permanent pour un problème de poursuite ou de régulation mais peut diminuer les marges de stabilité car il s'agit d'une action à retard de phase. Il peut déstabiliser le système s'il est mal choisi.

Exemple (suite).

- Calculer la fonction de transfert $F3$ en boucle fermée du système avec un correcteur proportionnel-Intégral.
- Afficher la réponse indicielle de $F3$, en prenant $K_p = 30$ et $K_i = 70$.
- Donner des conclusions.

5.1.3 Correcteur proportionnel et Dérivé (PD)

La fonction de transfert de ce correcteur est :

$$C(p) = K_p + K_d p$$

Ce correcteur améliore les performances dynamiques du système (temps de réponse, dépassement, ...). L'inconvénient pratique de ce dernier est l'accentuation des bruits de hautes fréquences provenant de l'entrée.

Exemple (suite).

- Calculer la fonction de transfert $F2$ en boucle fermée du système avec un correcteur proportionnel-Dérivé.
- Afficher la réponse indicielle de $F2$, en prenant $K_p = 300$ et $K_d = 10$.
- Donner des conclusions.

5.1.4 Correcteur proportionnel-Intégral-Dérivé (PID)

Le correcteur PID combine les actions des 3 correcteurs P, I et D. La fonction de transfert est :

$$C(p) = K_p + \frac{K_i}{p} + K_d p$$

Ce correcteur cumule les bénéfices de chacune des actions proportionnelle, intégrale et dérivée. Sa synthèse est toutefois complexe car si chaque action

peut améliorer le comportement si elle est bien placée, elle peut aussi le dégrader si elle est mal placée !!

Exemple (suite).

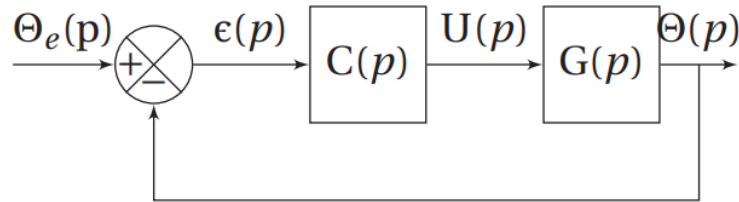
- Calculer la fonction de transfert $F4$ en boucle fermée du système avec un correcteur proportionnel-Intégral.
- Afficher la réponse indicielle de $F4$, en prenant les bonnes valeurs de K_p , K_d et K_i afin de répondre au cahier de charge.
- Donner des conclusions.

Sous forme d'un tableau récapitulatif, on résume les avantages et les limitations des actions de base des régulateurs **PID** :

Action	Points forts	Points faibles
P	Action instantanée	Ne permet pas d'annuler une erreur statique mais permet de la réduire
I	Annule l'erreur statique	Action lente et ralentit le système (effet déstabilisant)
D	Action très dynamique, améliore la rapidité et apporte un effet stabilisant	Sensibilité aux bruits et forte sollicitation de l'organe de commande

5.2 Énoncé du TP

Le système étudié est un réacteur, dont on se propose de réguler la température sous MATLAB.



L'équation différentielle reliant la température $\theta(t)$ à la tension de commande $u(t)$ s'écrit :

$$\frac{d^2\theta}{dt^2} + (\alpha + 4)\frac{d\theta}{dt} + 4\alpha\theta(t) = 4u(t)$$

Le paramètre α a été identifié par plusieurs essais mais sa valeur, dépendante aux produits introduits dans le réacteur, n'est connue que par un encadrement : $16 \leq \alpha \leq 20$. La régulation complète est représentée par le schéma bloc ci contre.

- $\Theta_e(p)$: la transformée de Laplace de la température de consigne $\theta_e(t)$;
- $\Theta(p)$ et $U(p)$ les transformées de Laplace de $\theta(t)$ et $u(t)$;
- $C(p)$: le régulateur.

Dans un premier temps, on choisit un régulateur proportionnel :

- (a) Déterminer la fonction de transfert $G(p)$.
- (b) Déterminer la fonction de transfert en boucle fermée $G_F(p) = \frac{\Theta(p)}{\Theta_e(p)}$.
Mettre sous forme canonique.
- (c) Le système est-il stable ?
- (d) Déterminer K_p pour obtenir un coefficient d'amortissement $\xi > 0.5$.
- (e) Déterminer l'erreur indicielle pour un échelon de température de $\Theta_0 = 20$ degré, en fonction du paramètre α . Conclure sur la précision.

On choisit maintenant un régulateur proportionnel intégral :

- (a) Déterminer la FTBO pour les valeurs suivantes de K_i et α ;
 - $K_i = 20, K_i = 16$.
 - $\alpha = 20, \alpha = 16$.
- (b) À partir du tracé des diagrammes de Bode ou de Nyquist de la FTBO pour chacun des cas (on prend $K_p = 1$), déterminer le couple (K_i, K_p) tel que m_φ soit supérieure à 45 degré pour toutes les valeurs de α .
- (c) Déterminer l'erreur indicielle pour un échelon de température de $\Theta_0 = 20$ degré.

TP 6 : Représentation d'état, commande et observation des systèmes continus asservis

6.1 Objectif

L'analyse par variables d'état est une approche moderne d'étude des systèmes née dans les années 60. Parmi les domaines d'application de cette théorie, l'automatique qui prend une place privilégiée. Les représentations d'état sont à l'origine de méthodes puissantes d'analyse, de commande et d'observabilité. Il est à noter que la propriété de commandabilité répond à la question de savoir si les états d'un système dynamique peuvent être arbitrairement modifiés par les entrées de commande alors que la propriété d'observabilité indique si l'information fournie par les signaux de sortie mesurés suffit à reconstruire l'état initial du système dynamique. Tous ces aspects sont au cœur de ce TP.

6.2 Représentation d'état

La théorie moderne de la commande des systèmes repose en grande partie sur le concept d'état d'un système et sur le modèle associé. L'état dynamique d'un système peut être caractérisé par un ensemble de variables internes appelées **variables d'état**. Le choix de cet ensemble n'est pas unique mais doit comporter un nombre toujours identique de variables d'état **indépendantes**.

Définition 6.2.1 (*Représentation d'état linéaire*)

Tout système dynamique Σ peut être représenté par ses équations d'état définies comme un ensemble d'équations différentielles du premier ordre appelées **équations dynamiques** et un ensemble d'équations algébriques appelées **équations de sortie** ou **de mesure**.

Si Σ vérifie l'hypothèse de linéarité alors sa représentation d'état est donnée par :

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\quad (6.1)$$

où

- $A \in R^{n \times n}$ est la matrice dynamique.
- $B \in R^{n \times m}$ est la matrice de commande ou d'entrée.
- $C \in R^{r \times n}$ est la matrice de mesure ou de sortie.
- $D \in R^{r \times m}$ est la matrice de transmission directe.

Dans ce cas de figure, le type de commande utilisé sous MATLAB est le type `ss` (state-space). La fonction associée, `ss()` prend en argument les 4 matrices définissant le système LTI (Linéaire Temps Invariant).

`ss(A,B,C,D)`

Cette représentation d'état a pour équivalent le schéma fonctionnel suivant :

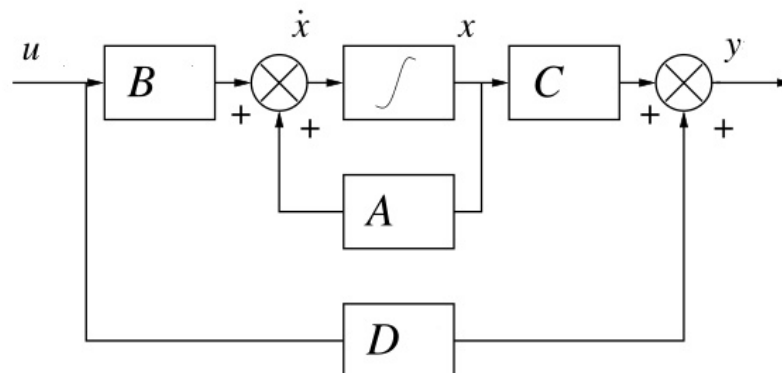


FIGURE 6.1: Schéma fonctionnel d'une représentation d'état.

Exercice 6.2.2 Soit le circuit électrique RLC de la figure (6.2). En écrivant les lois de Kirchoff (nœuds et mailles), n obtient alors les deux équations différentielles suivantes :

$$\begin{aligned}C \frac{dv_C}{dt} &= u(t) - i_L \\ L \frac{di_L}{dt} &= -Ri_L + v_C\end{aligned}$$

En choisissant comme variables d'état la tension aux bornes de la capacité et le courant dans l'inductance, soit $[x_1 \ x_2]^T = [v_C \ i_L]^T$. Donner l'équation dynamique d'état du réseau électrique.

Pour $C = 2F$, $L = 1H$ et $R = 3\Omega$. Calculer le modèle d'état, le polynôme caractéristique et les pôles pour le circuit électrique théoriquement puis sous MATLAB en utilisant les commandes adéquates.

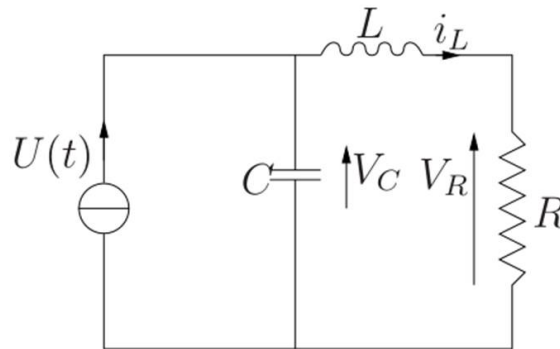


FIGURE 6.2: Circuit électrique RLC.

Du fait de la linéarité de l'opérateur de Laplace, il est possible de l'appliquer aux matrices. Partant des équations (6.1), il vient

$$\begin{cases} pX(p) = AX(p) + BU(p) \iff X(p) = (pI - A)^{-1} BU(p) \\ Y(p) = CX(p) + DU(p) \end{cases}$$

les conditions initiales étant considérées nulles puisqu'il s'agit de déterminer $G(p)$. De toute évidence, ceci amène à :

$$G(p) = C(pI - A)^{-1} B + D$$

Lors de l'étude d'un système il est souvent pratique de passer d'une représentation à une autre en fonction des objectifs. En effet, les fonctions de transfert et la représentation d'état sont deux formalismes qui possèdent leurs propres outils d'analyse et méthodes de synthèse de lois de commande.

```
% fonction de transfert → représentation d'état
G=tf([3],[4 12 2])
ss(G)
```

```
% représentation d'état → fonction de transfert
A=[0 -1/2;1 -3];B=[1/2;0];
C=[0 3];D=0;
sys=ss(A,B,C,D)
G=tf(sys)
```

6.3 Commandabilité et observabilité

6.3.1 Critère de commandabilité

La commandabilité est une propriété caractéristique du couplage entre l'entrée et la sortie du système et fera donc intervenir les matrices A et B .

Théorème 6.3.1 (*Critère de Kalman*)

Un système LTI d'équation dynamique d'état,

$$\dot{x}(t) = Ax(t) + Bu(t)$$

est commandable si et seulement si la matrice de commandabilité, \mathcal{C} est de rang n ,

$$\text{rang}(\mathcal{C}) = \text{rang} \left(\begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} \right) = n$$

$$\text{rang}(\mathcal{C}) = \text{ctrb}(A, B)$$

6.3.2 Critère d'observabilité

La notion d'observabilité fait intervenir les matrices A et C .

Théorème 6.3.2 (*Critère de Kalman*)

Un système LTI d'équations dynamique et de mesure (6.1), est observable si et seulement si la matrice d'observabilité, \mathcal{O} est de rang n ,

$$\text{rang}(\mathcal{O}) = \text{rang} \left(\begin{bmatrix} C \\ \dots \\ CA \\ \dots \\ \vdots \\ \dots \\ CA^{n-1} \end{bmatrix} \right) = n$$

$$\mathcal{O} = \text{obsv}(A, C)$$

Exercice 6.3.3 (Circuit électrique) Calculer les matrices de commandabilité et d'observabilité puis dire si le système est commandable et/ou observable.

6.4 Commande par retour d'état

Ce système (6.1) peut être représenté par le schéma bloc suivant : On désire as-

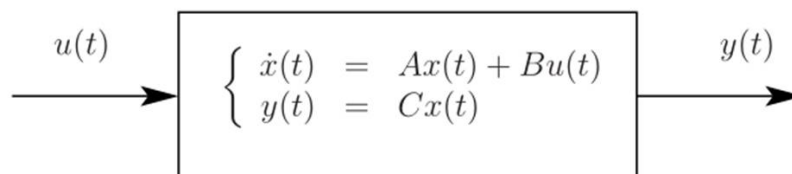


FIGURE 6.3: Schéma bloc d'un modèle espace d'état.

servir le système à une valeur $y_{ref}(t)$ tout en imposant les dynamiques du régime transitoire et en maintenant une erreur petite ou nulle en régime permanent. Modifier le régime transitoire du système (6.1), c'est modifier les pôles de la matrice dynamique A . On implante ainsi une loi de commande par retour d'état (par placement

de pôles) qui prend en compte les valeurs de l'état à l'instant t . La loi de commande s'écrit alors :

$$u(t) = -Kx(t) + v(t)$$

où $K \in R^{m \times n}$ est une matrice appelée gain du retour d'état et $v(t)$ est une nouvelle entrée pour le système en boucle fermée, comme le montre la figure (6.4). Le

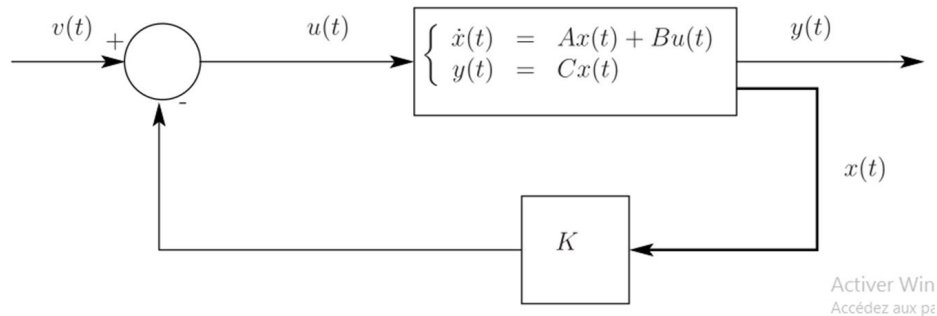


FIGURE 6.4: Le schéma bloc du retour d'état

système en boucle fermée s'écrit donc :

$$\begin{aligned}\dot{x}(t) &= (A - BK)x(t) + Bv(t) \\ y(t) &= Cx(t)\end{aligned}$$

Exercice 6.4.1 (Circuit électrique) On désire placer les pôles du système électrique en -3 et -4 .

1. Calculer la matrice K théoriquement puis sous *MATLAB* en utilisant la commande **place**.
2. Donner la réponse du système électrique en boucle ouverte et boucle fermée, asservi par un retour d'état.

6.5 Observateur de Luenberger

L'objectif est de reconstruire l'état interne d'un système à l'aide d'un algorithme dynamique. Ce dernier dispositif est appelé Observateur.

Soit le système (6.1) dont nous voulons observer l'état $x(t)$. L'observateur est un système dynamique qui s'écrit :

$$\begin{aligned}\dot{z}(t) &= Fz(t) + Gy(t) + Hu(t) \\ \hat{x}(t) &= Mz(t) + Ny(t)\end{aligned}$$

où $u(t)$ et $y(t)$ sont respectivement l'entrée et la sortie du système. Remarquons que ce sont également les deux entrées de l'observateur. La sortie de l'observateur est la variable $\hat{x}(t)$.

On définit un signal d'erreur $\epsilon(t)$ comme suit : $z(t) = Tx(t) + \epsilon(t)$. L'objectif est de construire l'observateur c'est-à-dire de choisir convenablement F, G, H, M, N et T telle que

$$\lim_{t \rightarrow \infty} \hat{x}(t) - x(t) = 0$$

Or, en utilisant l'expression de $z(t)$ et de $y(t)$, on peut réécrire $\hat{x}(t)$ de la manière suivante :

$$\hat{x}(t) = (MT + NC)x(t) + M\epsilon(t)$$

Posons

$$MT + NC = I$$

il vient que $\hat{x}(t) = x(t) + M\epsilon(t)$. Si l'erreur converge asymptotiquement vers zéro, $\lim_{t \rightarrow \infty} \epsilon(t) = 0$, alors $\hat{x}(t)$ converge asymptotiquement vers $x(t)$.

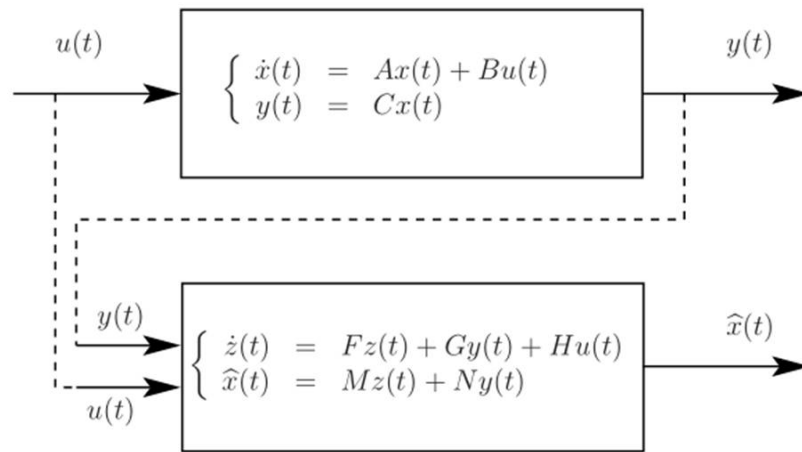


FIGURE 6.5: Le schéma bloc de l'observateur

Calculons la dynamique de l'erreur, c'est-à-dire

$$\begin{aligned} \dot{\epsilon}(t) &= \dot{z}(t) - T\dot{x}(t) \\ \dot{\epsilon}(t) &= F\epsilon(t) + (FT + GC - TA)x(t) + (H - TB)u(t) \end{aligned}$$

Imposons les relations suivantes :

$$\begin{aligned} FT + GC - TA &= 0 \\ H &= TB \end{aligned}$$

Nous obtenons alors la relation

$$\dot{\epsilon}(t) = F\epsilon(t)$$

La matrice F doit alors être choisie de manière à assurer la stabilité asymptotique de $\epsilon(t)$ et sa convergence vers 0. F est donc une matrice stable.

6.5.1 Observateur identité

Afin de résoudre les équations matricielles, on choisit :

$$T = I, M = I, N = 0$$

L'unique relation à résoudre est :

$$F = A - GC$$

On obtient alors l'observateur identité :

$$\dot{z}(t) = Fz(t) + Gy(t) + Bu(t)$$

$$\hat{x}(t) = z(t)$$

$$z(t) = x(t) + \epsilon(t)$$

Il faut ensuite déterminer le gain de l'observateur G tel que $F = A - GC$ soit une matrice stable. Ce calcul s'effectue en imposant le polynôme caractéristique de F , noté $P_{obs}(\lambda)$ et en résolvant l'équation $P_{obs}(\lambda) - P_{A-GC}(\lambda)$. Celui-ci peut également s'écrire :

$$\dot{\hat{x}}(t) = A\hat{x}(t) + G(y(t) - \hat{y}(t)) + Bu(t)$$

$$\hat{y}(t) = C\hat{x}(t)$$

Cela correspond finalement à une recopie du système original auquel un terme correctif dépendant de l'erreur $y(t) - \hat{y}(t)$ est ajouté.

Exercice 6.5.1 (Circuit électrique) *On souhaite construire un observateur identité ayant pour pôles -2 et -5.*

1. Calculer la matrice K théoriquement puis sous *MATLAB* en utilisant la commande **place**.
2. Faire des simulations des états $x_1(t)$ et $x_2(t)$ et leurs estimations dans la même figure.
3. Dans *SIMULINK*, simuler le système en associant l'observateur à la commande par retour d'état, tout en supposant que l'état n'est pas accessible par une mesure. Nous avons seulement la connaissance de $y(t)$.