

« Pose ta question, tu seras idiot une seconde.
Ne la pose pas, tu seras idiot toute ta vie. » Albert Einstein.

Rappels sur les réponses de systèmes dynamiques linéaires à temps continu & les méthodes de base d'identification de systèmes

Exercice 1.1 - Réponse indicielle d'un système du premier ordre

Soit le système décrit par la fonction de transfert

$$G(s) = \frac{2}{1 + 10s}$$

1. Préciser le gain statique, le pôle, la constante de temps T et la pulsation de coupure ω_c .
2. Déterminer la réponse indicielle et calculer la pente à l'origine.
3. Calculer les temps de montée $T_m^{63\%}$ et $T_m^{95\%}$ ainsi que le temps de réponse $T_r^{5\%}$.
4. Tracer précisément l'allure de la réponse indicielle et indiquer sur le tracé les paramètres caractéristiques calculés au dessus.
5. Déterminer la réponse fréquentielle $H(j\omega)$ en fonction de la pulsation de coupure. Calculer le module et l'argument de $H(j\omega)$.
6. Tracer les diagrammes de Bode en amplitude et en phase asymptotique et réel.
7. Vérifier vos réponses en exécutant le programme Matlab `td1_ident.m` disponible sur le site web du cours.

Exercice 1.2 - Réponse indicielle d'un système du premier ordre à retard

Soit un système du premier ordre ayant un gain statique de 2, une constante de temps de 10 s et un retard pur de 3 s .

1. Donner la fonction de transfert $G(s)$ du système.
2. Tracer sans calcul mais de manière précise la réponse indicielle. On indiquera sur le tracé les paramètres caractéristiques.
3. Déterminer la réponse fréquentielle $H(j\omega)$ en fonction de la pulsation de coupure. Calculer le module et l'argument de $H(j\omega)$;
4. Tracer les diagrammes de Bode en amplitude et en phase.

Exercice 1.3 - Réponse indicielle d'un système du second ordre sous-amorti

Soit le système dont le comportement dynamique est régi par l'équation différentielle suivante :

$$\ddot{y}(t) + 4\dot{y}(t) + 8y(t) = 2x(t) \quad \text{avec } \dot{y}(0) = 0, \quad y(0) = 0$$

1. Déterminer la fonction de transfert $G(s)$ du système.
2. Déterminer l'ordre du système, le gain statique K , le coefficient d'amortissement z , la pulsation propre non amortie ω_0 , les pôles et les zéros.
3. Dédire le type de régime pour la réponse indicielle. Calculer la valeur $D_{1\%}$ du premier dépassement, le temps t_{D_1} du premier dépassement, le temps de montée $T_m^{100\%}$, le temps de réponse $T_r^{5\%}$.
4. Tracer l'allure de la réponse indicielle et indiquer sur le tracé les paramètres caractéristiques calculés au dessus.
5. Déterminer la réponse fréquentielle $H(j\omega)$ en fonction de la pulsation de coupure. Calculer le module et l'argument de $H(j\omega)$;
6. Tracer les diagrammes de Bode en amplitude et en phase.

Exercice 1.4 - Identification d'un modèle à partir de la réponse indicielle d'une maquette de régulation de température

La réponse indicielle d'une maquette de régulation de température, constitué d'un organe de chauffage et d'un capteur de température, est représentée sur la figure 1.1.

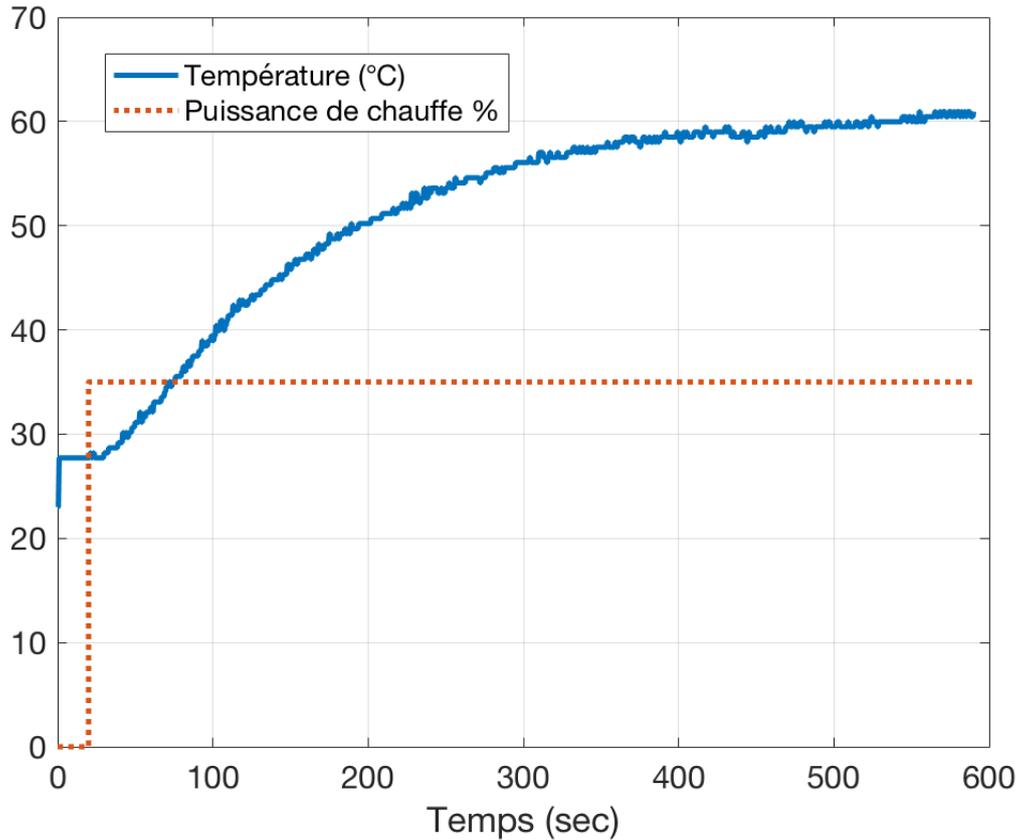


FIGURE 1.1: Réponse indicielle (données réelles) de la maquette de régulation de température.

1. D'après la réponse indicielle, proposer en justifiant une structure de modèle pour ce système.
2. Déterminer les paramètres du modèle retenu à l'aide de la méthode de votre choix.
3. Télécharger sur le site du cours les fichiers `data_maquette_temp.mat` et `Test_de_ton_modele_maquette_temp.m`.
4. Ouvrir le fichier `Test_de_ton_modele_maquette_temp.m` sous Matlab. Saisir en haut du fichier les valeurs des paramètres identifiés, puis exécuter le programme.
5. Comparer les réponses mesurée et simulée (ne pas tenir compte des premières valeurs de la réponse simulée).
6. Ajuster si nécessaire les valeurs des paramètres de votre modèle pour obtenir la meilleure adéquation possible entre les réponses.
7. Comme nous le verrons dans le cadre de ce cours, il existe aujourd'hui des algorithmes d'apprentissage/d'identification de paramètres très performants rassemblés dans des boîtes à outils comme celle développée au CRAN¹ qui permettent d'estimer les paramètres d'un modèle à temps continu directement à partir des données d'entrée/sortie mesurées.
Télécharger la boîte à outils CONTSID. Ajouter le répertoire CONTSID au chemin d'accès de Matlab (cliquer sur `setpath`) puis exécuter le fichier `ident_contsid_maquette_temp.m` sous Matlab². Relever les valeurs numériques du modèle estimé et comparer les avec les vôtres.

1. www.cran.univ-lorraine.fr/contsid

2. Les boîtes à outils System Identification et Control de Matlab doivent également être installées sur votre machine.

Exercice 1.5 - Identification d'un système à deux entrées à partir d'essais indiciels successifs

Soit un système à deux entrées et une sortie décrit par :

$$Y(s) = G_1(s)X_1(s) + G_2(s)X_2(s)$$

Afin de déterminer les fonctions de transfert $G_1(s)$ et $G_2(s)$, on a relevé expérimentalement la réponse temporelle du système, représentée sur la figure 1.2, pour les entrées suivantes :

$$x_1(t) = \Gamma(t) \quad \text{et} \quad x_2(t) = 2\Gamma(t - 15)$$

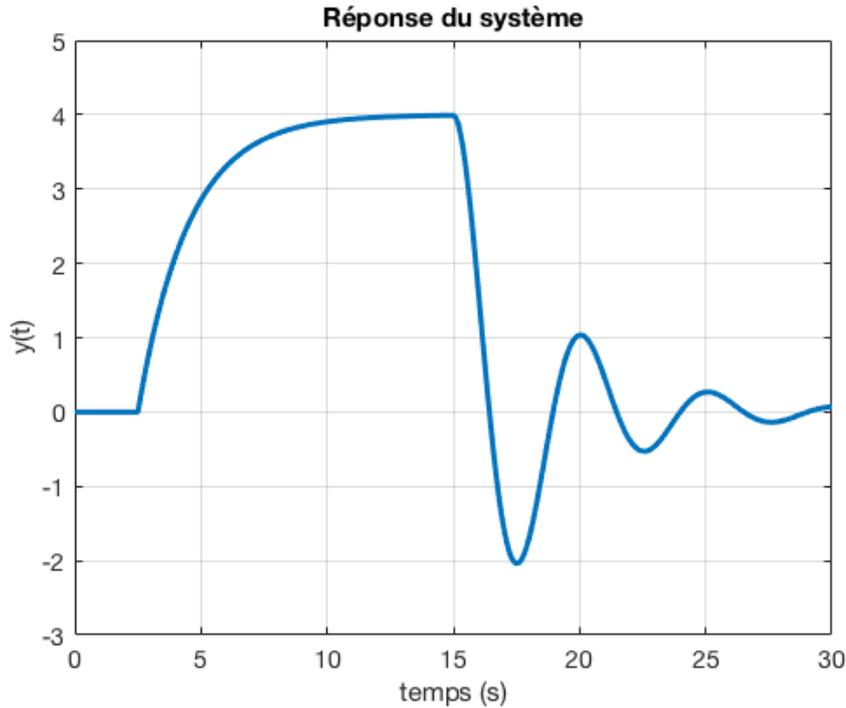


FIGURE 1.2: Réponse d'un système à deux entrées avec $x_1(t) = \Gamma(t)$ et $x_2(t) = 2\Gamma(t - 15)$

1. Représenter le système sous la forme d'un schéma fonctionnel.
2. Tracer l'évolution temporelle des deux entrées $x_1(t)$ et $x_2(t)$.
3. D'après la réponse obtenue à $x_1(t)$, proposer en justifiant un modèle pour $G_1(s)$.
4. Identifier à partir de la réponse indicielle à $x_1(t)$ les différents paramètres de $G_1(s)$.
5. D'après la réponse obtenue à $x_2(t)$, proposer en justifiant un modèle pour $G_2(s)$.
6. Identifier à partir de la réponse indicielle à $x_2(t)$ les différents paramètres de $G_2(s)$.

Les échantillons de la réponse indicielle représentée sur la figure 1.2, ainsi que les instants d'échantillonnage sont sauvegardés dans le fichier `data_syst_2entrees.mat` disponible sur le site web du cours.

Télécharger également le fichier `test_de_ton_modele_syst_2entrees.m`.

Indiquer en haut du fichier `.m` les valeurs numériques des paramètres des deux modèles que vous avez déterminés.

Exécuter ensuite le programme et évaluer la qualité de vos modèles. Ajuster si nécessaire les valeurs des paramètres pour obtenir la meilleure adéquation possible entre les réponses.

« Pose ta question, tu seras idiot une seconde.
Ne la pose pas, tu seras idiot toute ta vie. » Albert Einstein.

Modèles à temps continu et discret de systèmes linéaires

Exercice 2.1. Modèles d'un système linéaire à partir des données d'entrée/sortie échantillonnées

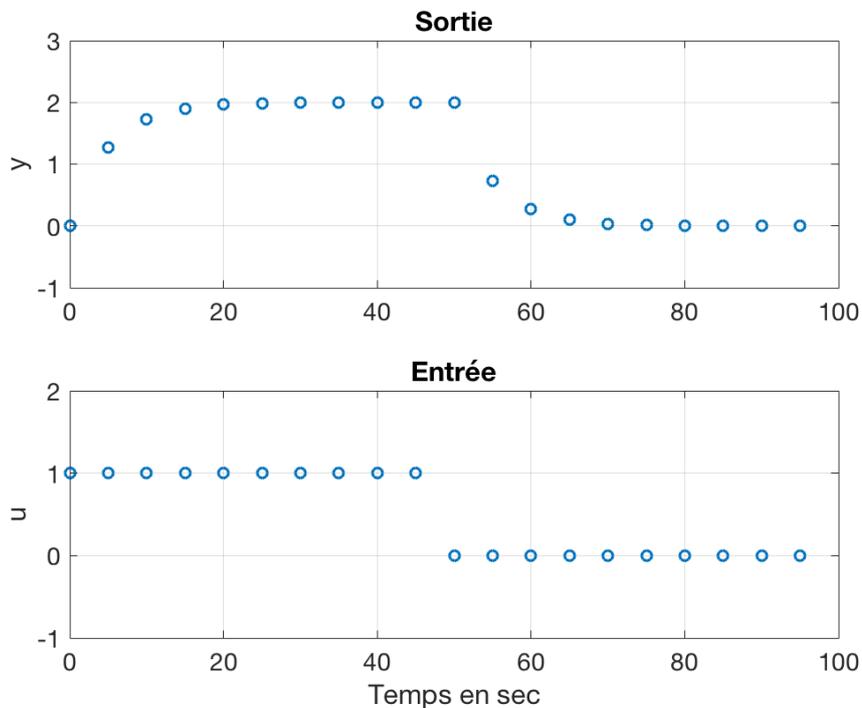


FIGURE 1.1: Échantillons des signaux d'entrée/sortie d'un système.

Sur la figure 1.1 sont représentés les échantillons déterministes des signaux d'entrée $u(t_k)$ et $y(t_k)$ d'un système linéaire à temps continu aux différents instants d'échantillonnage t_k représentés par des petits cercles.

Le but de cet exercice est de proposer la meilleure forme de modèle à temps continu et de modèle à temps discret d'après les échantillons enregistrés.

On cherche dans un premier temps à représenter le système sous la forme d'un modèle à temps continu.

1. D'après l'allure de la réponse de la figure 1.1, proposer en justifiant une forme de modèle à temps continu.
2. Déterminer, d'après la réponse, la valeur numérique des paramètres du modèle retenu.
3. En déduire la fonction de transfert de Laplace $G(s) = \frac{Y(s)}{U(s)}$. On remarquera que les paramètres de $G(s)$ ne dépendent pas de la période d'échantillonnage T_e .

4. On rappelle que la réponse indicielle d'un modèle à temps continu du premier ordre de gain statique K et de constante de temps T s'écrit :

$$y(t) = K(1 - e^{-t/T})\Gamma(t) \quad (1)$$

où $\Gamma(t)$ est l'échelon unitaire.

Calculer la réponse indicielle de votre modèle aux instants $t_k = kT_e$ pour $k = 0$ à 7 et $T_e = 5$ s et comparez-les avec les valeurs enregistrées :

[0 1.2642 1.7293 1.9004 1.9634 1.9865 1.9950 1.9982]

5. Affiner si besoin la valeur numérique des paramètres de votre modèle pour faire coïncider les réponses du système et du modèle à temps continu.

On cherche à présent déterminer un modèle à temps discret *équivalent*.

On rappelle qu'il n'existe pas de méthode de discrétisation universelle de modèles à temps continu. Par exemple, la méthode de discrétisation dite du bloqueur d'ordre zéro n'est exacte que pour une entrée constante par morceaux. Pour toute autre entrée, le modèle à temps discret correspondant ne sera qu'une approximation du modèle à temps continu.

1. D'après l'allure du signal d'entrée représenté sur la figure 1.1, formuler une hypothèse sur la variation du signal d'entrée entre 2 instants d'échantillonnage.
2. La fonction de transfert en z équivalente à un modèle du 1er ordre à temps continu décrit par (1) lorsque l'entrée est constante entre deux instants d'échantillonnage s'écrit :

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_1}{z + a_1} = \frac{b_1 z^{-1}}{1 + a_1 z^{-1}} \quad (2)$$

avec

$$a_1 = -e^{-\frac{T_e}{T}} \quad b_1 = K(1 + a_1) \quad (3)$$

Notez la dépendance des coefficients de $G(z)$ vis à vis de la période d'échantillonnage T_e . Le même système échantillonné à une autre période d'échantillonnage aura donc la même forme mais des valeurs numériques différentes. Déterminer a_1 et b_1 lorsque $T_e = 5$ s.

3. Dédurre de $G(z)$ l'équation aux différences liant les échantillons des signaux d'entrée/sortie.
4. Calculer la réponse de votre modèle à temps discret aux instants $t_k = k \times T_e$ pour $k = 0$ à 7 et $T_e = 5$ s et comparez les échantillons avec les valeurs enregistrées.
5. Ecrire les trois lignes Matlab qui permettent de calculer/simuler la sortie du modèle à temps discret. Comparez votre implantation Matlab avec vos résultats précédents.
6. La fonction `filter` sous Matlab permet de calculer la réponse d'un système à temps discret (qui peut être vu comme un filtre numérique). Exploitez cette fonction pour calculer la réponse de votre modèle à temps discret aux instants $t_k = k \times T_e$ pour $k = 0$ à 7 et $T_e = 5$ s et comparez-les avec les valeurs enregistrées.
7. Une autre méthode de discrétisation consiste à approcher la dérivée des signaux par approximation numérique. Déterminer l'équation aux différences obtenue à partir de l'équation différentielle discrétisée en remplaçant la dérivée première du signal de sortie par :

$$\left. \frac{dy(t)}{dt} \right|_{t=t_k} = \frac{1}{T_e} (y(t_k) - y(t_{k-1})) \quad (4)$$

8. En déduire la fonction de transfert en z et comparez-la avec celle obtenue précédemment.
9. Calculer la réponse de votre nouveau modèle à temps discret aux instants $t_k = k \times T_e$ pour $k = 0$ à 7 et $T_e = 5$ s et comparer les échantillons avec les valeurs enregistrées.

« Pose ta question, tu seras idiot une seconde.
Ne la pose pas, tu seras idiot toute ta vie. » Albert Einstein.

Rappels sur la régression linéaire et l'estimation par moindres carrés

Exercice 3.1 - Estimation paramétrique d'un modèle de la position d'un chariot par moindres carrés

On souhaite modéliser la position d'un chariot qui se déplace le long d'un rail rectiligne.
On supposera que le chariot se déplace à accélération constante.

TABLE 1: Position du chariot au cours du temps

t_k	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$y(t_k)$	0.2	0.21	0.23	0.26	0.28	0.32	0.35	0.40	0.44	0.49

1. Les valeurs numériques de $y(t_k)$ pour quelques valeurs du temps sont rassemblées dans le tableau 1. Tracer la position du chariot au cours du temps.
2. Le modèle décrivant la position du chariot en fonction du temps est choisi comme :

$$y(t_k) = \frac{1}{2}at_k^2 + vt_k + c \quad (5)$$

Donner la signification physique et les unités des trois paramètres a, v, c du modèle.

3. Ecrire le modèle sous forme de régression linéaire.

$$y(t_k) = \phi^T(t_k)\theta \quad (6)$$

où $\theta = [a \quad v \quad c]^T$

4. A partir des $N = 10$ données du tableau 1, formuler le problème d'estimation des 3 paramètres sous forme matricielle :

$$Y = \Phi\theta \quad (7)$$

5. Préciser le nom et les dimensions de chacune des matrices.
6. L'estimation par moindres carrés ordinaires est rappelée ci-dessous :

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^TY \quad (8)$$

Implanter sous Matlab cette estimation pour déterminer les valeurs numériques des trois paramètres.

7. Vérifier que vous obtenez les mêmes estimées à l'aide de la commande Matlab suivant (où Φ représente Φ) :

```
thetaHat=Phi\Y
```

On privilégiera cette implantation à l'avenir.

8. Donner le modèle de simulation de la position du chariot et calculer les positions simulées à partir du modèle estimé.
9. Calculer et tracer l'erreur de simulation (aussi appelée résidus) aux différents instants de mesure.

10. Calculer la moyenne et la variance des résidus.

Exercice 3.2 - Estimation paramétrique d'un modèle à temps discret d'un réacteur chimique par moindres carrés

TABLE 2: Données issues du réacteur chimique

t_k	0	1	2	3	4	5	6	7	8	9	10
$u(t_k)$	1	1	1	1	1	-1	-1	-1	-1	-1	-1
$y(t_k)$	0	0.13	0.09	0.10	0.10	0.10	-0.17	-0.08	-0.11	-0.10	-0.10

1. Les mesures du flux d'entrée $u(t_k)$ et de la concentration du substrat de sortie $y(t_k)$ d'un réacteur chimique sont rassemblées dans le tableau 1.
2. Le modèle décrivant la réaction chimique est supposé pouvant s'écrire sous la forme suivante :

$$y(t_k) = b_0 u(t_k) + b_1 u(t_{k-1}) + b_2 u(t_{k-2}) + e(t_k) \quad (9)$$

où $e(t_k)$ est un bruit blanc de variance σ_e^2 .

En utilisant l'opérateur retard q^{-1} ($q^{-1}u(t_k) = u(t_{k-1})$), écrire le modèle sous forme polynomiale puis rappeler le nom de cette forme de modèle.

3. Que représentent les paramètres à estimer b_0 , b_1 et b_2 .
4. Ecrire le modèle sous forme de regression linéaire :

$$y(t_k) = \phi^T(t_k)\theta + e(t_k) \quad (10)$$

où $\theta = [b_0 \quad b_1 \quad b_2]^T$

5. A partir des $N = 11$ données du tableau 2, formuler le problème d'estimation des 3 paramètres sous forme matricielle :

$$Y = \Phi\theta + E \quad (11)$$

6. Rappeler la solution des moindres carrés en fonction de Y et Φ qui permet d'estimer le vecteur des paramètres θ .
7. On donne :

$$(\Phi^T \Phi)^{-1} = \begin{pmatrix} 0.2857 & -0.25 & 0.0357 \\ -0.25 & 0.50 & -0.25 \\ 0.0357 & -0.25 & 0.2857 \end{pmatrix} \quad \Phi^T Y = \begin{pmatrix} 0.75 \\ 0.95 \\ 0.61 \end{pmatrix} \quad (12)$$

Calculer les valeurs numériques des trois paramètres.

8. Donner le modèle de simulation de la concentration de substrat de sortie et calculer les concentrations simulées à partir du modèle estimé.
9. En déduire l'erreur de simulation aux différents instants de mesure.

Exercice 3.3 - Prédiction de la population aux USA

Le fichier `pop_usa.mat` contient les valeurs de la population aux USA sur la période 1900-2010.

On choisit de représenter l'évolution de la population aux USA par un modèle polynomial d'ordre 3 de la forme :

$$p(t_k) = \theta_0 + \theta_1 t_k + \theta_2 t_k^2 + \theta_3 t_k^3 \quad (13)$$

1. Déterminer les paramètres de ce modèle polynomial d'ordre 3 permettant de représenter l'évolution de la population aux USA de 1900 à 2010.
2. Exploiter votre modèle pour fournir une estimation de la population aux USA en 2020 puis en 2050.

« Pose ta question, tu seras idiot une seconde.
Ne la pose pas, tu seras idiot toute ta vie. » Albert Einstein.

Estimation paramétrique de modèles à temps discret

Initiation à la boîte à outils Matlab *System Identification* (SID)

Le travail réalisé fera l'objet d'un compte-rendu par binôme qui sera noté. Il est demandé de suivre les consignes présentées ci-dessous pour rédiger votre compte-rendu.

Consignes pour la rédaction de votre compte-rendu de TD Matlab

Un compte-rendu est un document scientifique. Il doit donc respecter une certaine organisation.

Votre compte-rendu qui pourra être réalisé sous Live editor de Matlab devra être structuré de la manière suivante :

- une introduction générale précisant les objectifs du TP
- Pour chaque partie ou exercice :
 - une présentation brève des attendus
 - un rappel ou une présentation (si nécessaire) des résultats théoriques
 - une description des résultats de simulation sous la forme graphique ou numérique
 - une analyse critique des résultats en lien avec ceux théoriques attendus
 - une conclusion brève
- une conclusion générale qui précise ce qui a été compris et retenu lors du TD et les éventuelles difficultés rencontrées.
- une annexe rassemblant les programmes Matlab que vous avez développés.

Votre compte-rendu à rédiger obligatoirement en binôme est à envoyer par courriel à l'intervenant sous la forme d'un fichier unique au format pdf joint au message en respectant la date limite d'envoi qui aura été communiquée durant la séance de TD.

1 Etude en simulation

Estimation paramétrique d'un modèle à temps discret par ARX, IVMA et SRIV.

1.1 Génération de signaux d'entrée/sortie

Soient les systèmes décrits par :

$$S1 : A(q^{-1})y(t_k) = q^{-nk}B(q^{-1})u(t_k) + e(t_k) \quad (1)$$

$$S2 : y(t_k) = \frac{q^{-nk}B(q^{-1})}{F(q^{-1})}u(t_k) + e(t_k) \quad (2)$$

où q^{-1} représente l'opérateur retard, u l'entrée du système, y la sortie mesurée, e un bruit blanc à temps discret, nk le nombre d'échantillons pour le retard et les polynômes $A(q^{-1})$, $B(q^{-1})$ et $F(q^{-1})$ sont définis par :

$$\begin{cases} A(q^{-1}) = 1.0 - 1.5q^{-1} + 0.7q^{-2} \\ B(q^{-1}) = 1.0 + 0.5q^{-1} \\ F(q^{-1}) = A(q^{-1}) \\ nk = 1 \end{cases}$$

1. Dédurre des équations (1) et (2) le nom (ARX ou OE) des structures de modèle des systèmes $S1$ et $S2$.
2. Préciser la fonction de transfert du modèle du système et du bruit pour chaque structure.
3. Préciser l'équation aux différences dans le cas de chaque structure.
4. Donner le nombre de paramètres à estimer pour chaque polynôme et le gain statique de la fonction de transfert.

Vous allez dans un premier temps générer des signaux d'entrée/sortie à partir desquels vous allez tenter d'estimer les paramètres des systèmes $S1$ et $S2$. Pour cela, une fois sous Matlab, entrer, dans un live script Matlab, la séquence de commandes suivante :

```
N=500;
A=[1 -1.5 0.7];
F=[1 -1.5 0.7];
B=[0 1.0 0.5];
S1 = idpoly(A,B);
S2 = idpoly(1,B,1,1,F);
u=sign(randn(N,1));
ydet=sim(S1,u);
datadet=iddata(ydet,u);
e=0.4*randn(N,1);
yS1=sim(S1,[u e]);
yS2=sim(S2,[u e]);
dataS1=iddata(yS1,u);
dataS2= iddata(yS2,u);

figure(1)
idplot(dataS1),title('Données issues du système 1')
figure(2)
idplot(dataS2),title('Données issues du système 2')
figure(3)
plot([ydet yS1 yS2]),
legend('sortie non bruitée','sortie du modèle ARX','sortie du modèle OE')
```

Vous avez à présent à votre disposition trois jeux de données contenus dans les objets de données `datadet`, `dataS1` et `dataS2` pour effectuer l'estimation paramétrique.

1.2 Estimation paramétrique

1.2.1 Choix de la structure de modèle

On supposera les ordres des modèles ainsi que le nombre d'échantillons pour le retard $nk = 1$ parfaitement connus.

1.2.2 Estimation par moindres carrés simples

On s'intéresse dans un premier temps à l'identification du système par la méthode des moindres carrés simples. Le modèle recherché est du type ARX :

$$\mathcal{M}_{\text{ARX}} : A(q^{-1})y(t_k) = q^{-nk}B(q^{-1})u(t_k) + e(t_k) \quad (3)$$

1. Rappeler l'équation aux différences associée au modèle.
2. En déduire l'écriture du modèle sous forme de régression linéaire à l'instant $t = t_k$.
3. En supposant N mesures des signaux d'entrée/sortie, formuler le problème sous forme matricielle (voir transparents de cours). Donner la dimension de chaque matrice.
4. Rappeler la solution de l'estimateur des moindres carrés.
5. Planter cette solution sous Matlab dans le cas des données déterministes `datadet` puis bruitées `dataS1` et `dataS2`. Commentaires. On rappelle que l'on réalise ici une estimation ponctuelle et que l'on ne peut parler de biais de l'estimateur mais plutôt d'erreur d'estimation. Vous pouvez exécuter plusieurs fois votre programme pour dégager la tendance au niveau du biais des estimées.
6. La boîte à outils Matlab *System Identification* contient différentes fonctions permettant de réaliser l'estimation des paramètres d'un modèle. Elles ont toutes la structure suivante :

```
M=fonction(data,nn)
```

avec :

- `M` : le modèle estimé;
- `fonction` : le nom de la fonction SID : `arx`, `iv4`, `oe`, `pem`, ...
- `data` : l'objet *données* contenant les signaux d'entrée/sortie construit à la partir de la fonction `iddata` : `data=iddata(y,u)` ;
- `nn` : un vecteur ligne permettant de définir le nombre de paramètres des polynômes du modèle et la valeur du retard (supposé être un multiple entier de la période d'échantillonnage $\tau = nk \times T_e$) : `[na nb nk]` par exemple. Attention il s'agit du nombre de paramètres à estimer de chaque polynôme du modèle et non du degré de chaque polynôme.

La commande suivante permet d'afficher les paramètres du modèle estimé :

```
present(M);
```

Si vous désirez obtenir plus d'informations sur l'utilisation d'une fonction, vous pouvez entrer la commande suivante à tout moment :

```
help fonction
```

La fonction SID `arx` permet d'estimer les paramètres d'un modèle ARX :

$$\mathcal{M}_{\text{ARX}} : y(t_k) = \frac{q^{-nk}B(q^{-1})}{A(q^{-1})}u(t_k) + \frac{1}{A(q^{-1})}e(t_k) \quad (4)$$

par la méthode des moindres carrés simples. La commande est la suivante :

```
Marx=arx(data,[na nb nk]);
```

`na` et `nb` correspondent respectivement au nombre de paramètres à estimer des polynômes $A(q^{-1})$ et $B(q^{-1})$ et `nk` représente le nombre d'échantillons pour le retard. Les paramètres estimés peuvent être visualisés par la commande `present(Marx)`. L'écart-type estimé de chaque paramètre est affiché entre parenthèse.

7. Entrer les commandes suivantes pour estimer les paramètres dans le cas des trois jeux de données par la méthode des moindres carrés simples :

```
Marxdet=arx(datadet,[2 2 1]);present(Marxdet);
```

```
MarxS1= arx(dataS1,[2 2 1]);present(MarxS1);
```

```
MarxS2= arx(dataS2,[2 2 1]);present(MarxS2);
```

8. Pour une réalisation du bruit, comparer les résultats de votre implantation avec ceux obtenus via la fonction de la boîte à outils SID.

1.2.3 Estimation par variable instrumentale à modèle auxiliaire

On s'intéresse à présent à l'identification d'un modèle ARX par la méthode de la variable instrumentale à modèle auxiliaire (IVMA) vue en cours.

1. En supposant N mesures des signaux d'entrée/sortie. Rappeler la solution de l'estimateur de variable instrumentale à modèle auxiliaire. Préciser la dimension de chaque matrice.
2. Implanter cette solution sous Matlab (voir transparents de cours). Commentaires.
Il n'existe hélas, pas de fonction dans la boîte à outils SID qui implante directement cette méthode de variable instrumentale.

1.2.4 Estimation par variable instrumentale itérative

La fonction `sriv` disponible dans la boîte à outils CONTSID³ permet d'estimer les paramètres d'un modèle du type OE :

$$\mathcal{M}_{\text{OE}} : y(t_k) = \frac{q^{-nk}B(q^{-1})}{F(q^{-1})}u(t_k) + e(t_k) \quad (5)$$

par la méthode de variable instrumentale itérative. Les estimées délivrées sont alors optimales (sans biais et à minimum de variance) lorsque le bruit additif $e(t_k)$ sur la sortie est blanc (système vrai de type OE). La commande est la suivante :

```
Msriv=sriv(data,[nb nf nk]);
```

```
% Attention à la définition du nombre de paramètres à estimer
```

`nb` et `nf` correspondent respectivement au nombre de paramètres à estimer des polynômes $B(q^{-1})$ et $F(q^{-1})$ du modèle et `nk` représente le nombre d'échantillons pour le retard. Les paramètres estimés ainsi que leur écart-type peuvent être visualisés par la commande :

```
present(Msriv);
```

1. Entrer les commandes suivantes pour estimer les paramètres du modèle OE à l'aide de la fonction `sriv` :

```
Msriv=sriv(data,[2 2 1]);present(Msriv);
```
2. Comparer avec les résultats obtenus avec les méthodes de moindres carrés simples et de variable instrumentale à modèle auxiliaire.

1.3 Analyse comparative des performances des estimateurs ARX, IVMA et SRIV par simulation de Monte Carlo

Ecrire un programme qui permet de comparer les performances des estimateurs ARX, IVMA et SRIV par simulation de Monte Carlo (voir transparents de cours) dans le cas des systèmes **S1** et **S2**.

Résumer vos conclusions dans un tableau synthétique permettant de résumer les propriétés statistiques (biais, variance) des 3 estimateurs étudiés dans le cas des systèmes **S1** et **S2**.

3. www.cran.univ-lorraine.fr/contsid/

*« Pose ta question, tu seras idiot une seconde.
Ne la pose pas, tu seras idiot toute ta vie. » Albert Einstein.*

Estimation paramétrique de modèles à temps continu

Initiation à la boîte à outils *CONtinuous-Time System IDentification* (CONTSID)

Le travail réalisé fera l'objet d'un compte-rendu par binôme qui sera noté. Il est demandé de suivre les consignes présentées ci-dessous pour rédiger votre compte-rendu.

Consignes pour la rédaction de votre compte-rendu de TD Matlab

Un compte-rendu est un document scientifique. Il doit donc respecter une certaine organisation.

Votre compte-rendu, qui pourra être réalisé sous Live Editor de Matlab, devra être structuré de la manière suivante :

- une introduction générale précisant les objectifs du TP
- Pour chaque partie ou exercice :
 - une présentation brève des attendus
 - un rappel ou une présentation (si nécessaire) des résultats théoriques
 - une description des résultats de simulation sous la forme graphique ou numérique
 - une analyse critique des résultats en lien avec ceux théoriques attendus
 - une conclusion brève
- une conclusion générale qui précise ce qui a été compris et retenu lors du TD et les éventuelles difficultés rencontrées.
- une annexe rassemblant les programmes Matlab que vous avez développés.

Votre compte-rendu à rédiger obligatoirement en binôme est à envoyer par courriel à l'intervenant sous la forme d'un fichier unique au format pdf joint au message en respectant la date limite d'envoi qui aura été communiquée durant la séance de TD.

1 Etude en simulation

Identification d'un modèle à temps continu du 2^{ème} ordre par LSSVF, IVSVF et SRIVC.

1.1 Génération de signaux d'entrée/sortie

Soit le système décrit par :

$$\mathcal{S} : y(t_k) = \frac{B(p)}{F(p)} u(t_{k-nk}) + e(t_k) \quad (1)$$

où p représente l'opérateur de différentiation, u l'entrée du système, y la sortie mesurée, e un bruit blanc à temps discret, nk représente le nombre d'échantillons pour le retard et les polynômes $B(p)$ et $F(p)$ sont définis par :

$$\begin{cases} B(p) = 2 \\ F(p) = p^2 + 4p + 3 \\ nk = 0 \end{cases}$$

1. Dédurre de l'équation (1) la structure du modèle du système (CARX ou COE)
2. Donner les caractéristiques du système à temps continu : gain statique, constantes de temps et pulsation de coupure associée à chaque constante de temps
3. Préciser l'équation différentielle du système à l'instant $t = t_k$

Vous allez dans un premier temps générer des signaux d'entrée/sortie à partir desquels vous allez tenter d'estimer les paramètres du système (1). Pour cela, une fois sous Matlab, entrer, dans un live script Matlab, la séquence de commandes suivante :

```
N=1500;
B=[2];
F=[1 4 3];
Te=0.01;
t=(0:N-1)'*Te;
u=prbs(4,100);
e=0.2*randn(N,1);
x=lsim(B,F,u,t);
y=x+e;
datadet=iddata(x,u,Te);
data=iddata(y,u,Te);
figure(1)
idplot(datadet)
figure(2)
idplot(data)
```

Vous avez à présent à votre disposition deux jeux de données contenus dans les objets de données `datadet` et `data` pour effectuer l'estimation paramétrique.

1.2 Estimation paramétrique

1.2.1 Choix de la structure de modèle

On supposera les ordres du modèle connus.

1.2.2 Estimateur LSSVF

On s'intéresse dans un premier temps à l'identification du système par la méthode des moindres carrés/filtre de variable d'état (LSSVF). Le modèle recherché est du type ARX continu :

$$\mathcal{M}_{\text{CARX}} : A(p)y(t_k) = B(p)u(t_{k-nk}) + e(t_k) \quad (2)$$

1. Rappeler l'équation différentielle associée au modèle.
2. En déduire l'écriture du modèle sous forme de régression linéaire à l'instant $t = t_k$.

3. En supposant N mesures des signaux d'entrée/sortie, formuler le problème sous forme matricielle. Donner la dimension de chaque matrice.
4. Rappeler la solution de l'estimateur LSSVF (voir transparents de cours).
5. Implanter cette solution sous Matlab dans le cas des données déterministes puis bruitées. Comparer avec les résultats obtenus.
On rappelle qu'on réalise ici une estimation ponctuelle et qu'on ne peut parler de biais de l'estimateur mais plutôt d'erreur d'estimation. Vous pouvez exécuter plusieurs fois votre programme pour dégager la tendance au niveau du biais des estimées.
6. La boîte à outils *CONTSID*⁴ contient différentes fonctions permettant de réaliser l'estimation des paramètres d'un modèle. Elles ont toutes la structure suivante :

```
M=fonction(data,nn,hyperparametre)
```

avec :

- M : le modèle estimé;
- **fonction** : le nom de la fonction CONTSID : `lssvf`, `ivsvf`, `srivc`, ...
- **data** : l'objet *données* contenant les signaux d'entrée/sortie construit à la prtir de la fonction `iddata` : `data=iddata(y,u,Te)` ;
- **nn** : un vecteur ligne permettant de définir le nombre de paramètres des polynômes du modèle et la valeur du retard (supposé être un multiple entier de la période d'échantillonnage $\tau = nk \times T_e$) : `[na nb nk]` par exemple. Attention il s'agit du nombre de paramètres à estimer de chaque polynôme du modèle et non du degré de chaque polynôme.
- **hyperparametre** : hyper-paramètre éventuel de la méthode d'estimation : pulsation de coupure (rad/s) du filtre SVF par exemple.

La commande suivante permet d'afficher les paramètres du modèle estimé :

```
present(M);
```

Si vous désirez obtenir plus d'informations sur l'utilisation d'une fonction, vous pouvez entrer la commande suivante à tout moment :

```
help fonction
```

La fonction CONTSID `lssvf` permet d'estimer les paramètres d'un modèle à temps continu par LSSVF. La commande est la suivante :

```
Mlssvf=lssvf(data,[na nb nk],lambda);
```

`na` et `nb` correspondent respectivement au nombre de paramètres à estimer des polynômes $A(p)$ et $B(p)$ et `nk` représente le nombre d'échantillons pour le retard, `lambda` représente la pulsation de coupure du filtre SVF (en rad/s).

Les paramètres estimés peuvent être visualisés par la commande `present(Mlssvf)` :

7. Entrer les commandes suivantes pour estimer les paramètres du modèle par la méthode des moindres carrés/filtre de variable d'état (LSSVF) :

```
Mlssvf1=lssvf(datadet,[2 1 0],3);present(Mlssvf1);
Mlssvf2=lssvf(data,[2 1 0],3);present(Mlssvf2);
```

8. Pour une réalisation du bruit, comparer les résultats de votre implantation avec ceux obtenus via la fonction de la boîte à outils CONTSID.

1.2.3 Estimateur IVSVF

On s'intéresse à présent à l'identification d'un modèle CARX par la méthode de la variable instrumentale à modèle auxiliaire/filtre de variable d'état IVSVF.

1. En supposant N mesures des signaux d'entrée/sortie. Rappeler la solution de l'estimateur IVSVF. Préciser la dimension de chaque matrice.
2. Implanter cette solution sous Matlab (voir transparents de cours). Commentaires.
3. La fonction `ivsvf` permet d'estimer les paramètres d'un modèle CARX par la méthode de la variable instrumentale associée à la technique des SVF. La commande est la suivante :

```
Mivsvf=ivsvf(data,[na nb nk],lambda);
```

4. www.cran.univ-lorraine.fr/contsid/

4. Entrer les commandes suivantes pour estimer les paramètres du modèle par IVSVF :

```
Mivsvf= ivsvf(data,[2 1 0],3);
present(Mivsvf);
```

5. Comparer avec les résultats obtenus avec `lssvf` sur un même jeu de données.

1.3 Estimateur SRIVC

La fonction `srivc` permet d'estimer les paramètres d'un modèle du type COE :

$$\mathcal{M}_{\text{COE}} : y(t_k) = \frac{B(p)}{F(p)}u(t_k - nk) + e(t_k) \quad (3)$$

par la méthode itérative de variable instrumentale. Les estimées délivrées sont alors optimales (sans biais et à minimum de variance) lorsque le bruit additif $e(t_k)$ sur la sortie est blanc (système vrai de type COE). La commande est la suivante :

```
Msrivc=srivc(data,[nb nf nk]);
```

```
% Attention à la définition du nombre de paramètres à estimer
```

`nb` et `nf` correspondent respectivement au nombre de paramètres à estimer des polynômes $B(p)$ et $F(p)$ du modèle et `nk` représente le nombre d'échantillons pour le retard. Les paramètres estimés ainsi que leur écart-type peuvent être visualisés par la commande :

```
present(Msrivc);
```

1. Entrer les commandes suivantes pour estimer les paramètres du modèle COE à l'aide de la fonction `srivc` :

```
Msrivc=srivc(data,[1 2 0]);present(Msrivc);
```

2. Comparer avec les résultats obtenus avec `lssvf` et `ivsvf`.

1.3.1 Analyse comparative des performances des estimateurs LSSVF, IVSVF et SRIVC par simulation de Monte Carlo

Ecrire un programme qui permet de comparer les performances des estimateurs LSSVF, IVSVF et SRIVC par simulation de Monte Carlo dans le cas du système (1).

Résumer vos conclusions dans un tableau synthétique permettant de résumer les propriétés statistiques (biais, variance) des 3 estimateurs étudiés dans le cas du système S.

*« Pose ta question, tu seras idiot une seconde.
Ne la pose pas, tu seras idiot toute ta vie. » Albert Einstein.*

Model order selection and model validation

Le travail réalisé fera l'objet d'un compte-rendu par binôme qui sera noté. Il est demandé de suivre les consignes présentées ci-dessous pour rédiger votre compte-rendu.

Consignes pour la rédaction de votre compte-rendu de TD Matlab

Un compte-rendu est un document scientifique. Il doit donc respecter une certaine organisation.

Votre compte-rendu, qui pourra être réalisé sous Live Editor de Matlab, devra être structuré de la manière suivante :

- une introduction générale précisant les objectifs du TP
- Pour chaque partie ou exercice :
 - une présentation brève des attendus
 - un rappel ou une présentation (si nécessaire) des résultats théoriques
 - une description des résultats de simulation sous la forme graphique ou numérique
 - une analyse critique des résultats en lien avec ceux théoriques attendus
 - une conclusion brève
- une conclusion générale qui précise ce qui a été compris et retenu lors du TD et les éventuelles difficultés rencontrées.
- une annexe rassemblant les programmes Matlab que vous avez développés.

Votre compte-rendu à rédiger obligatoirement en binôme est à envoyer par courriel à l'intervenant sous la forme d'un fichier unique au format pdf joint au message en respectant la date limite d'envoi qui aura été communiquée durant la séance de TD.

1 Model order selection and model validation

One of the most difficult stages in the data-driven model learning procedure is to select a model structure and choose the different polynomial degrees (or equivalently fix the number of parameters to be determined for each polynomial).

To select a model structure, a suitable path is to follow the procedure recommended during the lectures which is summarized below :

1. Start with non-parametric response or estimates (step response or frequency-responses). This can provide useful information about the time-delay, system order and bandwidth.
2. Begin with a COE model structure.
3. Identify COE models for a range of possible polynomial degrees and time-delays. Select the model order that has the most negative YIC with the highest associated coefficient of determination R_T^2 according to the flexibility/parsimony trade-off.
4. Check out the properties of the estimated COE model, like its ability to simulate the measured data and other traditional model validation tests (residual analysis, physical interpretation of the main features of the identified model, ...) and cross-validation if another dataset is available.
5. Try a more complex model structure such as a hybrid Box-Jenkins model, if necessary.

To become familiar with the recommended start-up procedure, run the CONTSID main demonstration program through the command

```
>> contsid_demo
```

Select *Tutorials* in the menu window.

In the new menu window, select *Getting started* and follow the demo displayed in the command window.

When the first demo is finished, select then *Determining Model order and Input Delay* and follow the demo displayed in the command window.

Quit to come back to the main menu window.

2 A simulated system

The goal is to test first your new model learning skills on simulated data.

From the course website, download the file named : **Simulated_data.zip** which includes three files named :

- **Simulated_stepdata.mat** contains the step response,
- **Simulated_data1.mat** contains the system response to a PRBS input,
- **Simulated_data2.mat** contains the system response to a different PRBS input.

The input of this simulated system is stored in **u** while the output is stored in **y**. The sampling period is specified in **Ts**.

The three datasets have been generated from a simulated system. There is thus a "correct solution" (a true model) to it.

1. By using the start-up procedure outlined in Section 1, try to figure out the best model for the data.
2. Check out the properties of the estimated model, like its ability to simulate the measured data and other traditional model validation tests (residual analysis, etc).
3. Use the second file to cross-validate your model.
4. Compute the static gain of your model (see **dcgain** routine along with the cut-off frequencies (in rad/s) and damping ratio(s) (see **damp** routine of your estimated model).
5. Present the main characteristics of your identified model in a table.

3 Real-life mechanical system data

The goal is now to test your linear model learning skills on real-life data.

From the course website, download the file named : `Mechanical_data.zip` which includes three files named :

- `Mechanical_stepdata.mat` contains the step response,
- `Mechanical_data1.mat` contains the system response to a PRBS input,
- `Mechanical_data2.mat` contains the system response to a different PRBS input.

The three real-life datasets have been measured from a mechanical three-mass oscillator system shown in Figure 1.1.

The input of this real-life system is the synchronous motor torque (N.m) stored in `u` while the angular velocity (in rad/s) of the third mass represents the system output stored in `y`. The sampling period is fast and given in `Ts`. The collected data have been pre-processed and can be considered as ready to use without additional pre-treatments.

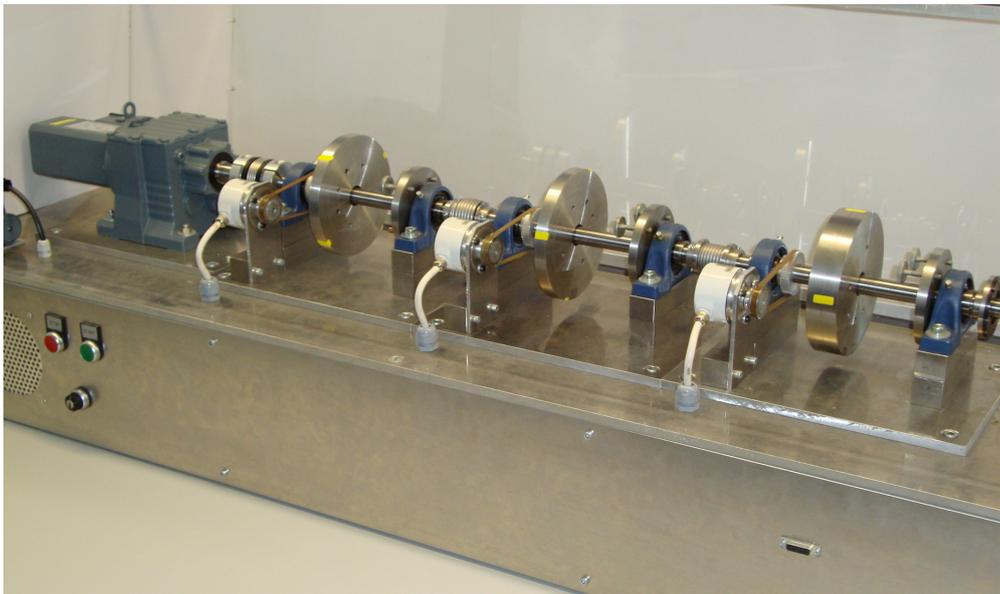


FIGURE 1.1: Mechanical three-mass oscillator system.

Note that for this practical case, there is thus no "correct solution" (a true model) to it, but you might use some mechanical insight to estimate/validate your model.

1. By using the start-up procedure outlined in Section 1, try to figure out the best model for the data.
2. Check out the properties of the estimated model, like its ability to simulate the measured data and other traditional model validation tests (residual analysis, etc).
3. Use the second dataset to cross-validate your model.
4. Compute the static gain of your model along with the underdamped resonance frequencies (in rad/s) and damping ratios for each resonance mode in your model.
5. Present the main characteristics of your identified model in a table and interpret the main features of your identified model in a physical sense.

Quelques fonctions de transfert importantes

Systèmes du premier ordre

$$G(s) = \frac{K}{1 + Ts}$$

Les 2 paramètres caractéristiques d'un système du 1er ordre

— K : gain statique

$$K = \frac{y(\infty) - y(0)}{u(\infty) - u(0)} = \lim_{s \rightarrow 0} G(s)$$

— T : constante de temps (ou pulsation de coupure $\omega_c = \frac{1}{T}$)

Valeurs caractéristiques de la réponse indicielle d'un système du premier ordre

Temps de montée à 63% $T_m^{63\%} = T$

Temps de montée à 95% $T_m^{95\%} \approx 3T$

Temps de réponse à 5 % $T_r^{5\%} \approx 3T$

Systèmes du deuxième ordre

$$G(s) = \frac{K}{\frac{s^2}{\omega_0^2} + 2\frac{z}{\omega_0}s + 1} = \frac{K\omega_0^2}{s^2 + 2z\omega_0s + \omega_0^2}$$

Les 3 paramètres caractéristiques d'un système du 2nd ordre

— K : gain statique

— z : coefficient d'amortissement ($z > 0$)

— ω_0 : pulsation propre non amortie

Valeurs caractéristiques de la réponse indicielle d'un système du second ordre pseudo-périodique

Valeur du 1er dépassement en % $D_{1\%} = \frac{y(T_{D_1}) - y(\infty)}{y(\infty) - y(0)} \times 100 = e^{\frac{-\pi z}{\sqrt{1-z^2}}} \times 100$

Instant du 1er dépassement $T_{D_1} = \frac{\pi}{\omega_0 \sqrt{1-z^2}}$

Valeur du $n^{\text{ième}}$ dépassement en % $D_{n\%} = -(-D_1)^n \times 100$

Instant du $n^{\text{ième}}$ dépassement $T_{D_n} = n T_{D_1}$

Pseudo-période $T_p = \frac{2\pi}{\omega_0 \sqrt{1-z^2}}$

Temps de réponse à 5 % $T_r^{5\%} \approx \frac{3}{\omega_0 z}$

Temps de réponse à x % $T_r^{x\%} = \frac{\ln\left(\frac{100}{x\sqrt{1-z^2}}\right)}{\omega_0 z}$

Temps de montée (100%) $T_m^{100\%} = \frac{\pi - \arccos(z)}{\omega_0 \sqrt{1-z^2}}$

Identification des systèmes à partir de la réponse indicielle

Identification d'un système du premier ordre

$$G(s) = \frac{K}{1 + Ts}$$

Il faut déterminer à partir de la réponse indicielle le gain statique K et la constante de temps T . La marche à suivre est la suivante :

1. Relever les valeurs finale et initiale de la réponse et de l'échelon. On en déduit K :

$$K = \frac{y(+\infty) - y(0)}{u(+\infty) - u(0)}$$

2. Relever $y(T_r^{5\%})$, en déduire $T_r^{5\%}$ puis T :

$$T = \frac{T_r^{5\%}}{3}$$

Identification d'un système du second ordre pseudo-périodique

$$G(s) = \frac{K\omega_0^2}{s^2 + 2z\omega_0s + \omega_0^2}$$

Il faut déterminer à partir de la réponse indicielle le gain statique K , le coefficient d'amortissement z et la pulsation propre non amortie ω_0 . La marche à suivre est la suivante :

1. Relever les valeurs finales et initiales de la réponse et de l'échelon. On en déduit K :

$$K = \frac{y(+\infty) - y(0)}{u(+\infty) - u(0)}$$

2. Relever les valeurs finale et initiale de la réponse ainsi que celle du premier dépassement $y(t_{D_1})$. On en déduit D_1 , puis z :

$$D_1 = \frac{y(T_{D_1}) - y(\infty)}{y(\infty) - y(0)}; \quad z = \sqrt{\frac{(\ln(D_1))^2}{(\ln(D_1))^2 + \pi^2}}$$

3. Relever l'instant du premier dépassement T_{D_1} . On en déduit ω_0 :

$$\omega_0 = \frac{\pi}{T_{D_1}\sqrt{1 - z^2}}$$

Identification d'une réponse apériodique par un modèle de Broïda

Broïda a proposé d'approcher la réponse apériodique de tout système d'ordre n par un premier ordre avec retard pur (fictif)

$$G(s) = \frac{Ke^{-\tau s}}{1 + Ts}$$

Il faut déterminer à partir de la réponse indicielle le gain statique K , la constante de temps T et le retard pur τ . La marche à suivre est la suivante :

1. Relever les valeurs finales et initiales de la réponse et de l'échelon. On en déduit K :

$$K = \frac{y(+\infty) - y(0)}{u(+\infty) - u(0)}$$

2. Relever $y(T_m^{28\%})$ et $y(T_m^{40\%})$, en déduire $T_m^{28\%}$ et $T_m^{40\%}$ puis :

$$\tau = 2,8T_m^{28\%} - 1,8T_m^{40\%}$$

$$T = 5,5 \left(T_m^{40\%} - T_m^{28\%} \right)$$

Il faut rappeler que le retard pur fictif τ n'a pas d'existence réelle. Si le système comporte un retard pur naturel $\bar{\tau}$, il faudra l'ajouter au retard pur τ . Ce retard pur naturel doit être déterminé graphiquement comme le temps mis par le système pour commencer à réagir après l'envoi de l'échelon (instant de décollement de la réponse). Le modèle complet identifié prend alors la forme :

$$G(s) = \frac{Ke^{-(\bar{\tau} + \tau)s}}{1 + Ts}$$

Propriétés utiles de la transformée de Laplace

$$L(\alpha x(t) + \beta y(t)) = \alpha X(s) + \beta Y(s)$$

$$L(x(t) * y(t)) = X(s) \times Y(s)$$

$$L(tx(t)) = -\frac{dX(s)}{ds}$$

$$L(x(t - t_0)) = e^{-t_0 s} X(s)$$

$$L(\dot{x}(t)) = sX(s) - x(0)$$

$$L(\ddot{x}(t)) = s^2 X(s) - sx(0) - \dot{x}(0)$$

$$\lim_{t \rightarrow 0^+} x(t) = \lim_{s \rightarrow +\infty} sX(s)$$

$$\lim_{t \rightarrow +\infty} x(t) = \lim_{s \rightarrow 0} sX(s) \quad \text{si la limite existe}$$

Table de transformées de Laplace

Signal	Transformée de Laplace
$\delta(t)$	1
$\Gamma(t)$	$\frac{1}{s}$
$r(t) = t\Gamma(t)$	$\frac{1}{s^2}$
$t^2\Gamma(t)$	$\frac{2}{s^3}$
$e^{-at}\Gamma(t)$	$\frac{1}{s+a}$
$t^n e^{-at}\Gamma(t)$	$\frac{n!}{(s+a)^{n+1}}$
$\cos(\omega_0 t)\Gamma(t)$	$\frac{s}{s^2 + \omega_0^2}$
$\sin(\omega_0 t)\Gamma(t)$	$\frac{\omega_0}{s^2 + \omega_0^2}$
$e^{-at} \cos(\omega_0 t)\Gamma(t)$	$\frac{s+a}{(s+a)^2 + \omega_0^2}$
$e^{-at} \sin(\omega_0 t)\Gamma(t)$	$\frac{\omega_0}{(s+a)^2 + \omega_0^2}$