Travaux dirigés

1 Identification d'un système simulé

Considérons un système dont la relation entrée-sortie est représentée par les trois modèles suivants

$$\begin{cases} Y(z) = \dfrac{0.04\, z^{-1}}{1 - 1.6\ z^{-1} + 0.64\, z^{-2}}\, U(z) + \dfrac{1}{1 - 1.6\, z^{-1} + 0.64\, z^{-2}}\, E(z) \\ Y(z) = \dfrac{0.04\, z^{-1}}{1 - 1.6\ z^{-1} + 0.64\, z^{-2}}\, U(z) + \dfrac{1 + 0.1\, z^{-1}}{1 - 1.6\, z^{-1} + 0.64\, z^{-2}}\, E(z) \\ [1 - 1.6\, z^{-1} + 0.64\, z^{-2}]\, Y(z) = 0.04\, z^{-1} U(z) + [1 - 1.6\, z^{-1} + 0.64\, z^{-2}]\, E(z) \end{cases} \tag{1}$$

1) Exprimer la forme polynomiale réduite de chaque modèle puis tracer son diagramme fonctionnel. Pour chaque forme, spécifier le type de modèle (ARX, ARMAX, OE ou BJ) et son ordre.

1.1 Simulation du système à partir de sa forme polynomiale

Afin de générer des signaux d'entrée-sortie du système dans le cas d'un bruit de mesure i.i.d de distribution gaussienne à moyenne nulle et d'écart-type $\sigma_e = 0.1$, le système est simulé pour une période d'échantillonnage $T_e = 1$ s. Pour cela, écrire un script Matlab permettant de réaliser les étapes ci-dessous :

2) Définir les valeurs des polynômes `(A, B, C, D, F)` de la représentation polynomiale du troisième modèle de l'équation (1) puis définir la variable objet modèle exact.

```
>> A = ...; B= ...; C = ...; D = ...; F = ...;
>> modexact = idpoly(...);
```

3) Générer une réalisation du signal d'entrée qui est une séquence binaire pseudo-aléatoire d'amplitude $[-5, +5]$, de longueur 630 points et un facteur de division de fréquences $b = 10$.

```
>> u = idinput(...);
```

4) Simuler une réalisation du signal de sortie du système soumis à l'entrée précédente.

```
>> options = simOptions('AddNoise',false);
>> ysansbruit = sim(modexact, u, options);
```

5) Compléter le script pour simuler le système en présence de bruit et calculer le rapport signal sur bruit puis l'exprimer en décibels.

```
>> yavecbruit = sim(...);
>> rsb = 20*log10(std(ysansbruit)/std(yavecbruit - ysansbruit));
>> disp(['RSB (dB) = ' num2str(rsb)]);
```

6) Créer l'object données de l'identification et tracer les signaux d'entrée-sortie dans les domaines temporel et fréquentiel.

```
>> dat=iddata(...);
>> figure(1), plot(dat);
>> figure(2), plot(fft(dat));
```

1.2 Identification d'une forme polynomiale du système

Les signaux générés précédemment seront utilisés pour l'identification en employant différentes fonctions disponibles dans la boite à outils System Identification de Matlab (`cra, arx, armax, oe, bj, pem`). Pour chaque méthode, il est demandé d'analyser la capacité du modèle identifié à reproduire le comportement du système à l'aide des fonctions de validation (`resid, compare, aic, fpe`).

1) **Identification de la réponse impulsionnelle.**

   ```
   >> nb=100; modimp = cra(dat,nb), figure, plot(modimp);
   ```

   A partir de la visualisation de la réponse impulsionnelle est il possible de déduire la valeur du retard ?

2) **Identification d'un modèle ARX.** Compléter le script pour réaliser l'identification à l'aide d'un modèle ARX dont l'ordre sera recherché pour optimiser le facteur de détermination RT2 (cf. fonction `compare`) et les critères AIC et FPE. Il faut également analyser les statistiques des résidus (cf. fonction `resid`).

   ```
   >> na= ...; nb = ...;
   >> modarx = arx(dat, [na,nb]); present(modarx);
   >> [y,RT2] = compare(modarx, dat);
   >> AIC = aic(modarx);
   >> FPE = fpe(modarx);
   >> resid(dat,modarx);
   ```

3) Compléter le script de sorte à identifier les autres types de modèles en suivant les mêmes étapes que ci-dessus.

1.3 Annexe

Le script ci-dessous donne un exemple de code pour chercher le meilleur modèle au sens du critère FPE par exemple. Celui-ci peut être adapté pour d'autres critères.

```
na=1:3 ; nb=1:3; nf = 0:3;
NN = struc(na,nb,nk);
modarx = cell(size(NN,1),1);
for ct = 1:size(NN,1)
   modarx{ct} = arx(z2, NN(ct,:));
end
Compute the FPE values for the identified models.
Return the smallest value and its index.
V = fpe(modarx{:});
[Vmin, Imin] = min(V);
Return the optimal model that has the smallest FPE value.
present(modarx{Imin})
```