

Diagnostic robuste et contrôle tolérant aux fautes pour systèmes singuliers par factorisation copremière

B. Marx & D. Koenig



Laboratoire d'Automatique de Grenoble
UMR CNRS-UJF-INPG
B.P. 46, 38402 Saint Martin d'Hères, France

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Plan de l'exposé

- **Introduction aux systèmes singuliers**
- **Factorisation copremière de systèmes singuliers**
- **Diagnostic robuste de défauts**
- **Commande tolérante aux défauts**
- **Conclusion**

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Définition des Systèmes singuliers

Pour la modélisation de systèmes physiques on utilise des

- relations dynamiques (loi de comportement, stockage, PFD, etc.)
- relations statiques (maillage, équilibre de bilans, etc.)

On a donc un modèle du type :

$$0 = f(\dot{x}(t), x(t), u(t))$$

$$0 = g(x(t), u(t), y(t))$$

après linéarisation, le modèle devient :

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

$$y(t) = Cx(t) + Du(t)$$

avec $\text{rang}(E) = r \leq n = \text{dim}(x)$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Définition des Systèmes singuliers

Pour la modélisation de systèmes physiques on utilise des

- relations dynamiques (loi de comportement, stockage, PFD, etc.)
- relations statiques (maillage, équilibre de bilans, etc.)

On a donc un modèle du type :

$$0 = f(\dot{x}(t), x(t), u(t))$$

$$0 = g(x(t), u(t), y(t))$$

après linéarisation, le modèle devient :

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

$$y(t) = Cx(t) + Du(t)$$

$$\text{avec } \text{rang}(E) = r \leq n = \text{dim}(x)$$

on peut supposer $D = 0$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Formes équivalentes

pour P et Q non singulières on a : $\left\{ E, \begin{pmatrix} A & B \\ C & D \end{pmatrix} \right\} = \left\{ PEQ, \begin{pmatrix} PAQ & PB \\ CQ & D \end{pmatrix} \right\}$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Formes équivalentes

pour P et Q non singulières on a : $\left\{ E, \begin{pmatrix} A & B \\ C & D \end{pmatrix} \right\} = \left\{ PEQ, \begin{pmatrix} PAQ & PB \\ CQ & D \end{pmatrix} \right\}$

Forme équivalente standard :

$$\begin{aligned} \dot{x}_1(t) &= A_1 x_1(t) + B_1 u(t) \\ N \dot{x}_2(t) &= x_2(t) + B_2 u(t) \quad (N \text{ nilpotente}) \\ y(t) &= C_1 x_1(t) + C_2 x_2(t) \end{aligned}$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Formes équivalentes

pour P et Q non singulières on a : $\left\{ E, \begin{pmatrix} A & B \\ C & D \end{pmatrix} \right\} = \left\{ PEQ, \begin{pmatrix} PAQ & PB \\ CQ & D \end{pmatrix} \right\}$

Forme équivalente standard :

sous-syst. usuel



$$\dot{x}_1(t) = A_1 x_1(t) + B_1 u(t)$$

sous -syst. non propre



$$N\dot{x}_2(t) = x_2(t) + B_2 u(t) \quad (N \text{ nilpotente})$$

$$y(t) = C_1 x_1(t) + C_2 x_2(t)$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Formes équivalentes

pour P et Q non singulières on a : $\left\{ E, \begin{pmatrix} A & B \\ C & D \end{pmatrix} \right\} = \left\{ PEQ, \begin{pmatrix} PAQ & PB \\ CQ & D \end{pmatrix} \right\}$

Forme équivalente standard :

sous-syst. usuel

$$\dot{x}_1(t) = A_1 x_1(t) + B_1 u(t)$$

sous -syst. non propre

$$N \dot{x}_2(t) = x_2(t) + B_2 u(t) \quad (N \text{ nilpotente})$$

$$y(t) = C_1 x_1(t) + C_2 x_2(t)$$

Forme équivalente par décomposition par valeurs singulières :

$$\dot{x}_1(t) = A_{11} x_1(t) + A_{12} x_2(t) + B_1 u(t)$$

$$0 = A_{21} x_1(t) + A_{22} x_2(t) + B_2 u(t)$$

$$y(t) = C_1 x_1(t) + C_2 x_2(t)$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Formes équivalentes

pour P et Q non singulières on a : $\left\{ E, \begin{pmatrix} A & B \\ C & D \end{pmatrix} \right\} = \left\{ PEQ, \begin{pmatrix} PAQ & PB \\ CQ & D \end{pmatrix} \right\}$

Forme équivalente standard :

sous-syst. usuel

$$\dot{x}_1(t) = A_1 x_1(t) + B_1 u(t)$$

sous -syst. non propre

$$N \dot{x}_2(t) = x_2(t) + B_2 u(t) \quad (N \text{ nilpotente})$$

$$y(t) = C_1 x_1(t) + C_2 x_2(t)$$

Forme équivalente par décomposition par valeurs singulières :

dynamique

$$\dot{x}_1(t) = A_{11} x_1(t) + A_{12} x_2(t) + B_1 u(t)$$

statique

$$0 = A_{21} x_1(t) + A_{22} x_2(t) + B_2 u(t)$$

$$y(t) = C_1 x_1(t) + C_2 x_2(t)$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Réponse temporelle

- Réponse temporelle

$$y(t) = C_1(e^{A_1 t} [I \ 0] Q^{-1} x_0 + \int_0^t e^{A_1(t-\tau)} B_1 u(\tau) d\tau) - C_2 \sum_{k=0}^{h-1} N^k B_2 u^{(k)}(t)$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Réponse temporelle

- Réponse temporelle

$$y(t) = C_1 \left(e^{A_1 t} [I \ 0] Q^{-1} x_0 + \int_0^t e^{A_1(t-\tau)} B_1 u(\tau) d\tau \right) + C_2 \sum_{k=0}^{h-1} N^k B_2 u^{(k)}(t)$$

sous-syst.
usuel

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Réponse temporelle

- Réponse temporelle

$$y(t) = C_1 \left(e^{A_1 t} [I \ 0] Q^{-1} x_0 + \int_0^t e^{A_1(t-\tau)} B_1 u(\tau) d\tau \right) + C_2 \sum_{k=0}^{h-1} N^k B_2 u^{(k)}(t)$$

sous-syst.
usuel

sous syst.
non propre

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Réponse temporelle

- Réponse temporelle

$$y(t) = C_1 \left(e^{A_1 t} [I \ 0] Q^{-1} x_0 + \int_0^t e^{A_1(t-\tau)} B_1 u(\tau) d\tau \right) + C_2 \sum_{k=0}^{h-1} N^k B_2 u^{(k)}(t)$$

sous-syst.
usuel

sous syst.
non propre

- **régularité** \Leftrightarrow unicité de la trajectoire $x(t)$ pour $u(t)$ et x_0 données
 $\Leftrightarrow \det(sE-A) \neq 0$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Réponse temporelle

- **Réponse temporelle**

$$y(t) = C_1 \left(e^{A_1 t} [I \ 0] Q^{-1} x_0 + \int_0^t e^{A_1(t-\tau)} B_1 u(\tau) d\tau \right) + C_2 \sum_{k=0}^{h-1} N^k B_2 u^{(k)}(t)$$

sous-syst.
usuel

sous syst.
non propre

- **régularité** \Leftrightarrow unicité de la trajectoire $x(t)$ pour $u(t)$ et x_0 données
 $\Leftrightarrow \det(sE-A) \neq 0$
- **système non impulsif**
 - \Leftrightarrow admet une représentation usuelle
 - \Leftrightarrow a une fonction de transfert propre
 - $\Leftrightarrow N=0$, dans la forme équivalente standard
 - $\Leftrightarrow A_{22}$ inversible dans la forme équivalente par SVD de E

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Intérêts des systèmes singuliers

- signification physique des variables
- combinaison de relations dynamiques **et** statiques
 - systèmes rectangulaires
 - systèmes interconnectés
 - systèmes impulsifs
- mauvais conditionnement de $E^{-1}A$, pour E inversible

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Commandabilité des syst. singuliers

Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Commandabilité des syst. singuliers

(E, A, B)
commandable

Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Commandabilité des syst. singuliers

(E, A, B)
commandable



sous-syst. usuel
commandable

sous-syst.
non propre
commandable

- Introduction
- Syst. Sing.**
- Factorisation
- Diagnostic
- Commande
- Conclusion

Commandabilité des syst. singuliers

(E, A, B)
commandable



sous-syst. usuel
commandable

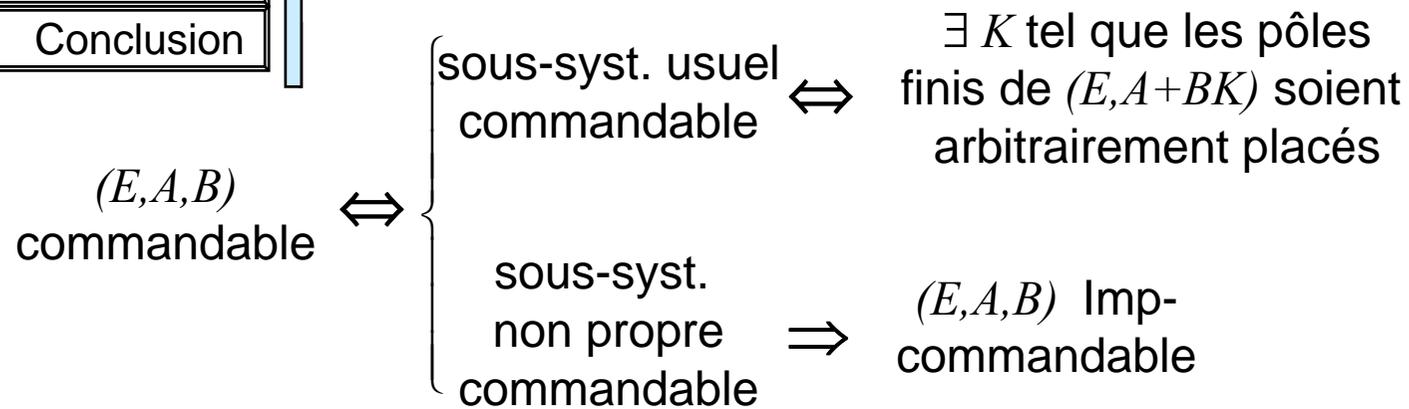
sous-syst.
non propre
commandable



$\exists K$ tel que les pôles
finis de $(E, A+BK)$ soient
arbitrairement placés

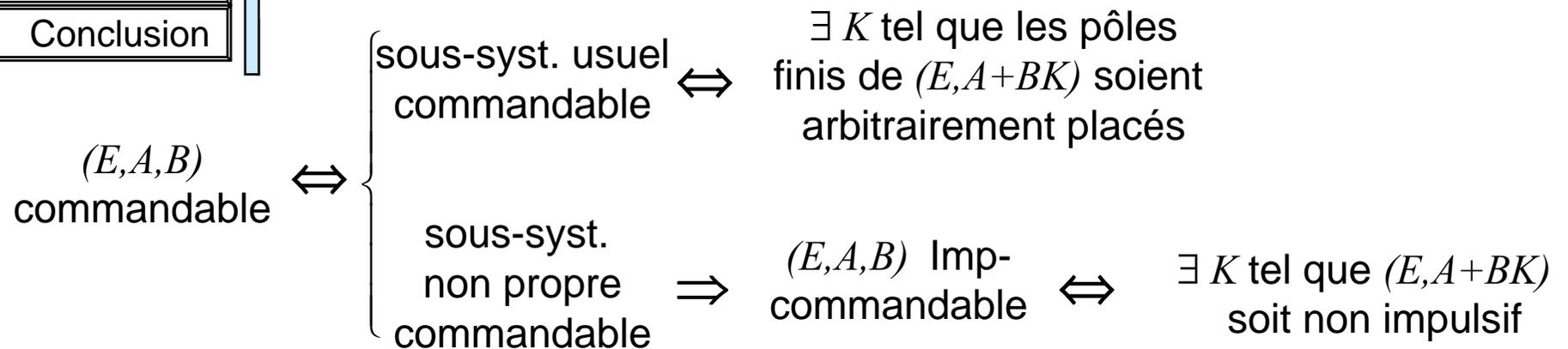
- Introduction
- Syst. Sing.**
- Factorisation
- Diagnostic
- Commande
- Conclusion

Commandabilité des syst. singuliers



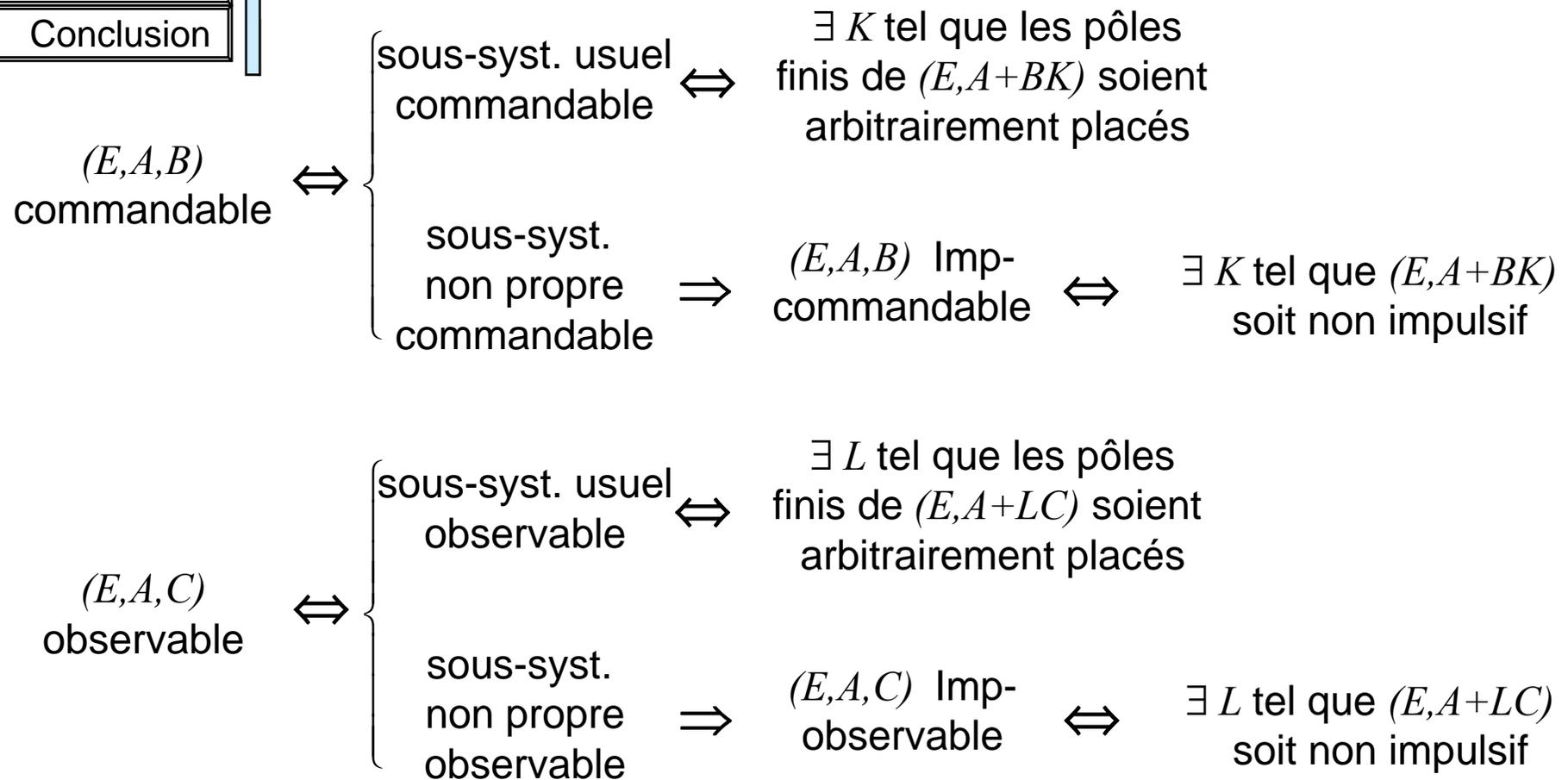
- Introduction
- Syst. Sing.**
- Factorisation
- Diagnostic
- Commande
- Conclusion

Commandabilité des syst. singuliers



- Introduction
- Syst. Sing.**
- Factorisation
- Diagnostic
- Commande
- Conclusion

Commandabilité des syst. singuliers



Introduction

Syst. Sing.

Factorisation

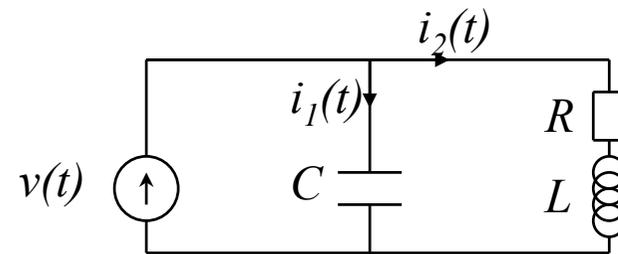
Diagnostic

Commande

Conclusion

Exemple

Circuit RLC parallèle :
à $t=1$, on applique $v=1 V$



Introduction

Syst. Sing.

Factorisation

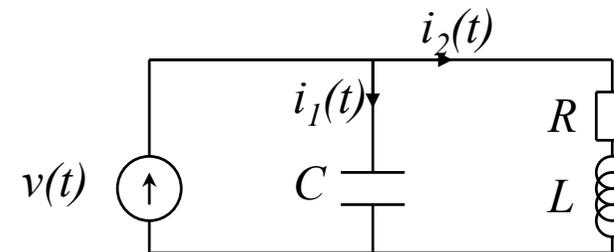
Diagnostic

Commande

Conclusion

Exemple

Circuit RLC parallèle :
à $t=1$, on applique $v=1 V$



décrit par la forme singulière

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & L \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{q} \\ \dot{i}_1 \\ \dot{i}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -R \\ 1/C & 0 & 0 \end{pmatrix} \begin{pmatrix} q \\ i_1 \\ i_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} v, \quad y = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} q \\ i_1 \\ i_2 \end{pmatrix}$$

Introduction

Syst. Sing.

Factorisation

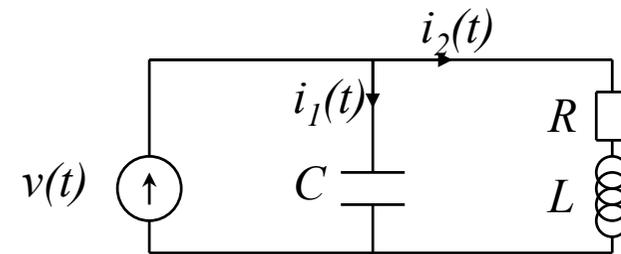
Diagnostic

Commande

Conclusion

Exemple

Circuit RLC parallèle :
à $t=1$, on applique $v=1 V$



décrit par la forme singulière

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & L \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{q} \\ \dot{i}_1 \\ \dot{i}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -R \\ 1/C & 0 & 0 \end{pmatrix} \begin{pmatrix} q \\ i_1 \\ i_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} v, \quad y = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} q \\ i_1 \\ i_2 \end{pmatrix}$$

de fonction de transfert

$$G(s) = \begin{bmatrix} Cs & 1 \\ sL + R \end{bmatrix}^T$$

Introduction

Syst. Sing.

Factorisation

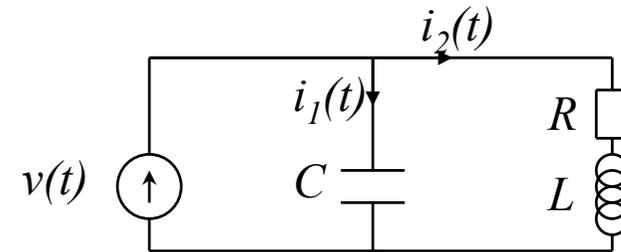
Diagnostic

Commande

Conclusion

Exemple

Circuit RLC parallèle :
à $t=1$, on applique $v=1\text{ V}$

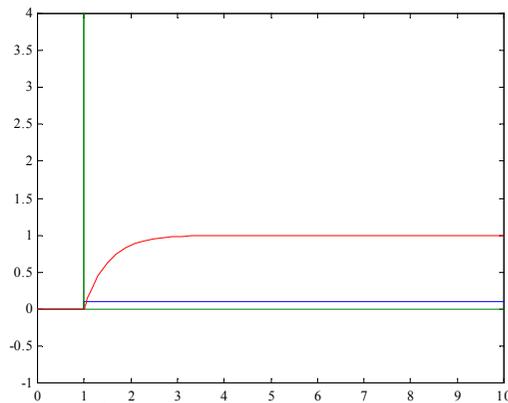


décrit par la forme singulière

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & L \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{q} \\ \dot{i}_1 \\ \dot{i}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -R \\ 1/C & 0 & 0 \end{pmatrix} \begin{pmatrix} q \\ i_1 \\ i_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} v, \quad y = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} q \\ i_1 \\ i_2 \end{pmatrix}$$

de fonction de transfert

$$G(s) = \begin{bmatrix} Cs & 1 \\ sL + R \end{bmatrix}^T$$



(E, A, B) R-commandable
et Imp-commandable

Introduction

Syst. Sing.

Factorisation

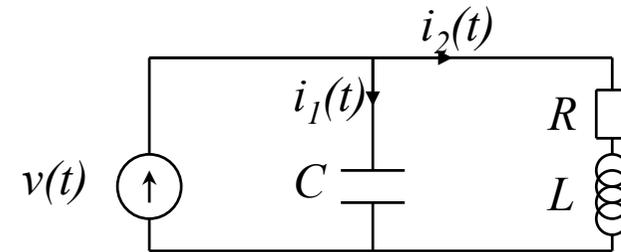
Diagnostic

Commande

Conclusion

Exemple

Circuit RLC parallèle :
à $t=1$, on applique $v=1\text{ V}$

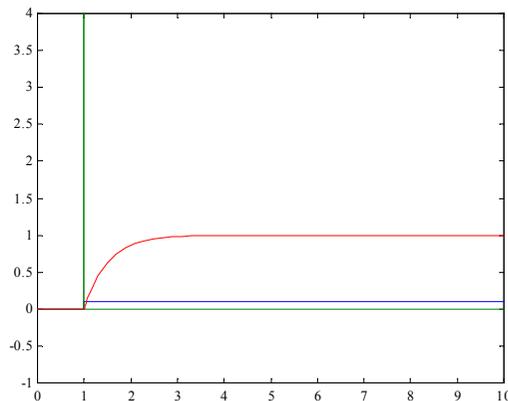


décrit par la forme singulière

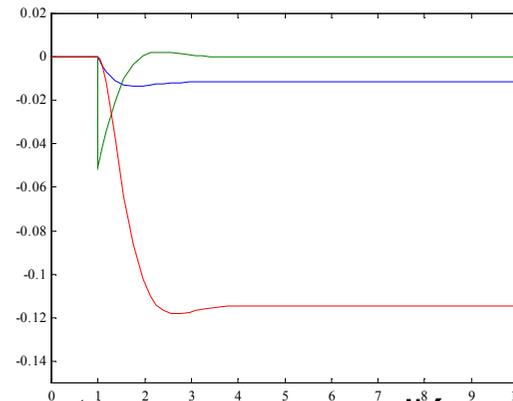
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & L \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{q} \\ \dot{i}_1 \\ \dot{i}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -R \\ 1/C & 0 & 0 \end{pmatrix} \begin{pmatrix} q \\ i_1 \\ i_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} v, \quad y = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} q \\ i_1 \\ i_2 \end{pmatrix}$$

de fonction de transfert

$$G(s) = \begin{bmatrix} Cs & \frac{1}{sL+R} \end{bmatrix}^T$$



État du système



$x(t)$, avec retour d'état normalisant

(E, A, C) R-observable et
Imp-observable

Introduction

Syst. Sing.

Factorisation

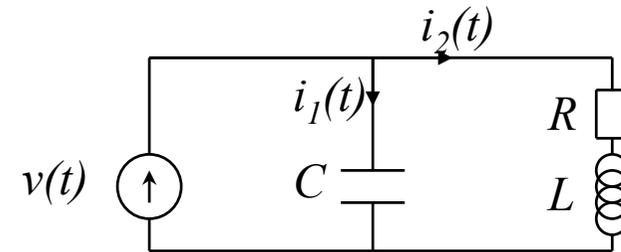
Diagnostic

Commande

Conclusion

Exemple

Circuit RLC parallèle :
à $t=1$, on applique $v=1 V$

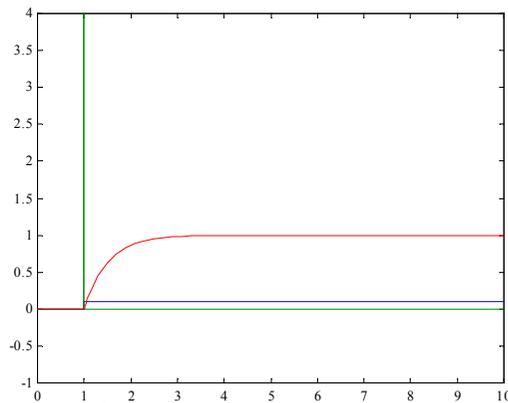


décrit par la forme singulière

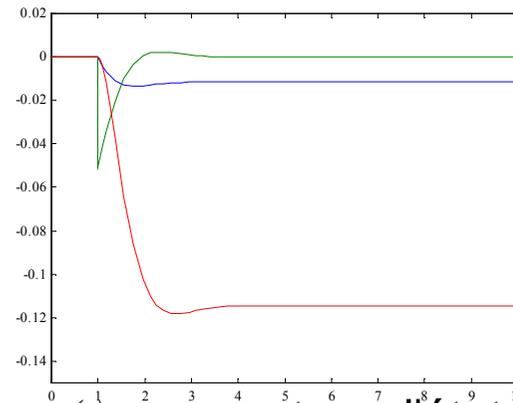
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & L \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{q} \\ \dot{i}_1 \\ \dot{i}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -R \\ 1/C & 0 & 0 \end{pmatrix} \begin{pmatrix} q \\ i_1 \\ i_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} v, \quad y = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} q \\ i_1 \\ i_2 \end{pmatrix}$$

de fonction de transfert

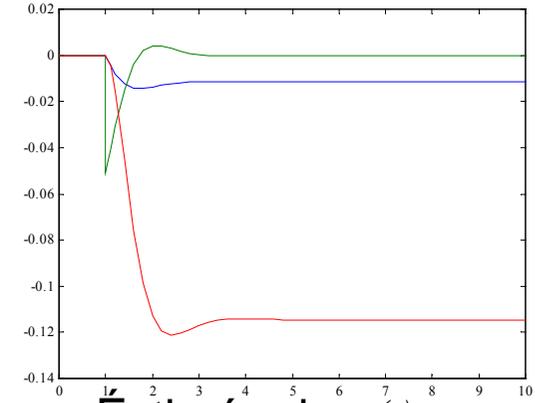
$$G(s) = \begin{bmatrix} Cs & \frac{1}{sL + R} \end{bmatrix}^T$$



État du système



$x(t)$, avec retour d'état normalisant



Estimée de $x(t)$, avec retour d'état observé

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Factorisation copremière

- **Objectif** : généraliser l'outil de factorisation copremière aux systèmes singuliers

- **Définition** (fonctions de transfert copremières):

Deux fonctions de transfert matricielles $M(s)$ et $N(s)$ de \mathbf{RH}_∞ sont dites copremières à droites (resp. à gauche) s'il existe deux fonctions de transfert matricielles $X_r(s)$ et $Y_r(s)$ (resp. $X_l(s)$ et $Y_l(s)$) telles que

$$\begin{bmatrix} X_r(s) & Y_r(s) \end{bmatrix} \begin{bmatrix} M(s) \\ N(s) \end{bmatrix} = I$$

$$\begin{bmatrix} M(s) & N(s) \end{bmatrix} \begin{bmatrix} X_l(s) \\ Y_l(s) \end{bmatrix} = I$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Factorisation copremière

- **Définition** (factorisation copremière):

Une fonction de transfert matricielle $P(s)$ de \mathbf{RH}_∞ admet une double factorisation copremière s'il existe une factorisation copremière à droite

$$P(s) = N(s)M^{-1}(s)$$

et une factorisation copremière à gauche

$$P(s) = \tilde{M}^{-1}(s)\tilde{N}(s)$$

où $M(s)$, $N(s)$, $\tilde{M}(s)$, $\tilde{N}(s)$, $X_r(s)$, $Y_r(s)$, $X_l(s)$ et $Y_l(s)$ de \mathbf{RH}_∞ vérifient

$$\begin{bmatrix} X_r(s) & Y_r(s) \\ -\tilde{N}(s) & \tilde{M}(s) \end{bmatrix} \begin{bmatrix} M(s) & -Y_l(s) \\ N(s) & X_l(s) \end{bmatrix} = I$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Factorisation copremière de systèmes singuliers

Théorème : une fonction de transfert matricielle $G(s)=C(sE-A)^{-1}B+D$ non nécessairement propre admet une double factorisation copremière si $G(s)$ est régulière, Imp-commandable et Imp-observable. On a alors

$$G(s) = N(s)M^{-1}(s) \quad G(s) = \tilde{M}^{-1}(s)\tilde{N}(s)$$

$$\begin{bmatrix} X_r(s) & Y_r(s) \\ -\tilde{N}(s) & \tilde{M}(s) \end{bmatrix} \begin{bmatrix} M(s) & -Y_l(s) \\ N(s) & X_l(s) \end{bmatrix} = I$$

et

$$\begin{bmatrix} X_r(s) & Y_r(s) \\ -\tilde{N}(s) & \tilde{M}(s) \end{bmatrix} = \left\{ E, \begin{pmatrix} A+LC & -(B+LD) & L \\ F & I & 0 \\ C & -D & I \end{pmatrix} \right\} \quad \text{et} \quad \begin{bmatrix} M(s) & -Y_l(s) \\ N(s) & X_l(s) \end{bmatrix} = \left\{ E, \begin{pmatrix} A+BF & B & -L \\ F & I & 0 \\ C+DF & D & I \end{pmatrix} \right\}$$

où L et F sont telles que $(E, A+LC)$ et $(E, A+BF)$ soient non-impulsifs

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Factorisation copremière de systèmes singuliers

Intérêts :

- Les cofacteurs $N(s)$, $M(s)$, $\tilde{M}(s)$ et $\tilde{N}(s)$ sont propres, y compris pour $G(s)$ non propre.
- Les matrices L et F peuvent être déterminées par résolution LMI pour placer les pôles finis de $(E, A+LC)$ et $(E, A+BF)$ dans une région LMI donnée, caractérisée par $\alpha = \alpha^T$ et β .

Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Placement des pôles finis de systèmes singuliers

Théorème. Pour une région LMI D il existe une matrice F telle que $(E, A+BF)$ ait ses pôles finis dans D et soit admissible, si et seulement s'il existe $P=P^T>0$, S , H et D vérifiant la condition LMI suivante

$$\alpha \otimes EPE^T + \beta \otimes (APE^T + BHE^T) + \beta^T \otimes (EPA^T + EH^TB^T) + \dots$$

$$I_{mm} \otimes (AVSU^T + BLU^T + US^TV^TA^T + UL^TB^T) < 0$$

avec $V = \text{Ker } E$ et $U = \text{Ker } E^T$,

et où I_{ij} désigne la matrice $(i \times j)$ de composantes égales à 1

F est donnée par

$$F = (HE^T + LU^T)(PE^T + VSU^T)^{-1}$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Diagnostic par factorisation copremière

- **Systeme** :
$$\begin{cases} E\dot{x} = Ax + Bu + R_1f + E_1d \\ y = Cx + Du + R_2f + E_2d \end{cases}$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Diagnostic par factorisation copremière

- **Systeme** :
$$\begin{cases} E\dot{x} = Ax + Bu + R_1f + E_1d \\ y = Cx + Du + R_2f + E_2d \end{cases}$$

- **Objectifs** :

- ◇ génération de résidus par des filtres propres
- ◇ modeler la réponse fréquentielle aux défauts
- ◇ imposer un gabarit de robustesse aux perturbations

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Diagnostic par factorisation copremière

- **Systeme** :
$$\begin{cases} E\dot{x} = Ax + Bu + R_1f + E_1d \\ y = Cx + Du + R_2f + E_2d \end{cases}$$
- **Objectifs** :
 - ◇ génération de résidus par des filtres propres
 - ◇ modeler la réponse fréquentielle aux défauts
 - ◇ imposer un gabarit de robustesse aux perturbations
- **Hypothèses** :
 - ◇ faisceau (E,A) régulier
 - ◇ système (E,A,C) Imp-observable et détectable

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Diagnostic par factorisation copremière

- **Systeme** :
$$\begin{cases} E\dot{x} = Ax + Bu + R_1f + E_1d \\ y = Cx + Du + R_2f + E_2d \end{cases}$$
- **Objectifs** :
 - ◇ génération de résidus par des filtres propres
 - ◇ modeler la réponse fréquentielle aux défauts
 - ◇ imposer un gabarit de robustesse aux perturbations
- **Hypothèses** :
 - ◇ faisceau (E,A) régulier
 - ◇ système (E,A,C) Imp-observable et détectable
- **Méthode** :
 - ◇ factorisation copremière du système
 - ◇ formulation H_∞ des objectifs

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Génération de résidus

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Génération de résidus

Le système peut s'écrire :

$$y(s) = G_u(s)u(s) + G_d(s)d(s) + G_f(s)f(s)$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Génération de résidus

Le système peut s'écrire :

$$y(s) = G_u(s)u(s) + G_d(s)d(s) + G_f(s)f(s)$$

et se factoriser sous la forme :

$$y(s) = \tilde{M}^{-1}(s) \left(\tilde{N}_u(s)u(s) + \tilde{N}_d(s)d(s) + \tilde{N}_f(s)f(s) \right)$$

Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Génération de résidus

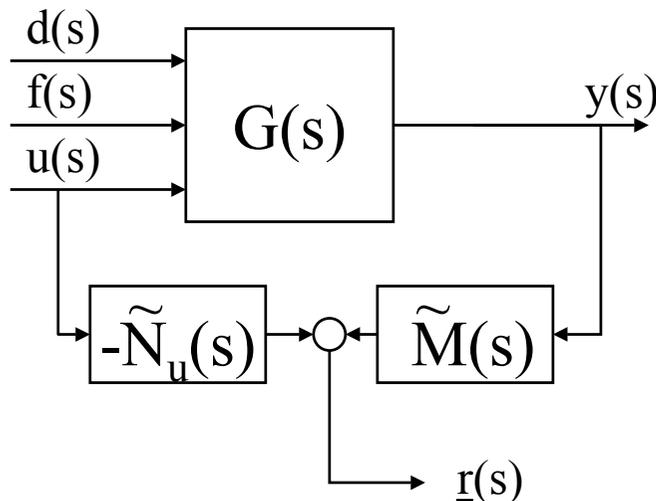
Le système peut s'écrire :

$$y(s) = G_u(s)u(s) + G_d(s)d(s) + G_f(s)f(s)$$

et se factoriser sous la forme :

$$y(s) = \tilde{M}^{-1}(s) \left(\tilde{N}_u(s)u(s) + \tilde{N}_d(s)d(s) + \tilde{N}_f(s)f(s) \right)$$

Génération de résidus par factorisation :



$$\begin{aligned} \underline{r}(s) &= \tilde{M}(s)y(s) - \tilde{N}_u(s)u(s) \\ &= \tilde{N}_f(s)f(s) + \tilde{N}_d(s)d(s) \end{aligned}$$

avec

$$\tilde{M}(s) = \left\{ E, \begin{pmatrix} A+LC & L \\ C & I \end{pmatrix} \right\} \text{ et } \tilde{N}_u(s) = \left\{ E, \begin{pmatrix} A+LC & B+LD \\ C & D \end{pmatrix} \right\}$$

Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Génération de résidus

Le système peut s'écrire :

$$y(s) = G_u(s)u(s) + G_d(s)d(s) + G_f(s)f(s)$$

et se factoriser sous la forme :

$$y(s) = \tilde{M}^{-1}(s) \left(\tilde{N}_u(s)u(s) + \tilde{N}_d(s)d(s) + \tilde{N}_f(s)f(s) \right)$$

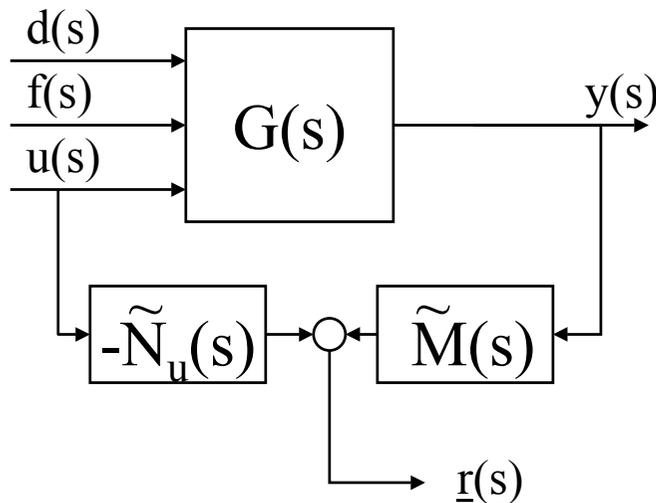
Génération de résidus par factorisation :

$$\begin{aligned} \underline{r}(s) &= \tilde{M}(s)y(s) - \tilde{N}_u(s)u(s) \\ &= \tilde{N}_f(s)f(s) + \tilde{N}_d(s)d(s) \end{aligned}$$

avec

$$\tilde{M}(s) = \left\{ E, \begin{pmatrix} A+LC & L \\ C & I \end{pmatrix} \right\} \text{ et } \tilde{N}_u(s) = \left\{ E, \begin{pmatrix} A+LC & B+LD \\ C & D \end{pmatrix} \right\}$$

⇒ **générateur de résidus propre (choix de L)**



Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Génération de résidus

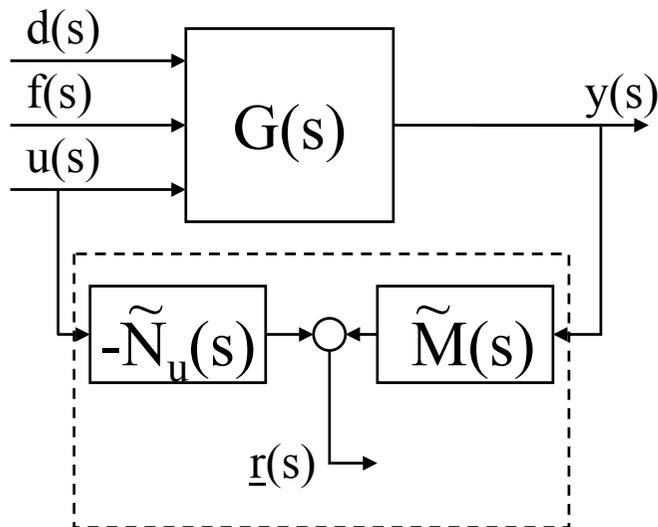
Le résidu est obtenu en 2 étapes :

Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Génération de résidus

Le résidu est obtenu en 2 étapes :

- factorisation de $G(s)$ (paramètre : L)

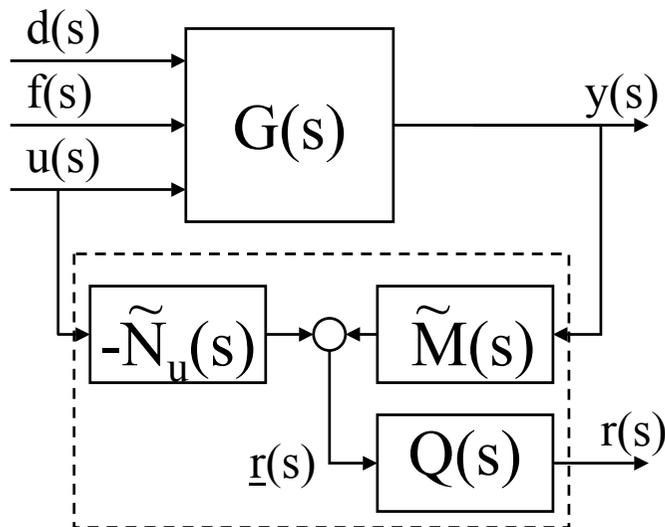


Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Génération de résidus

Le résidu est obtenu en 2 étapes :

- factorisation de $G(s)$ (paramètre : L)
- filtrage : $r(s) = Q(s)\underline{r}(s)$ (paramètre : $Q(s)$ de RH_∞)



Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Génération de résidus

Le résidu est obtenu en 2 étapes :

- factorisation de $G(s)$ (paramètre : L)
- filtrage : $r(s) = Q(s)\underline{r}(s)$ (paramètre : $Q(s)$ de \mathbf{RH}_∞)

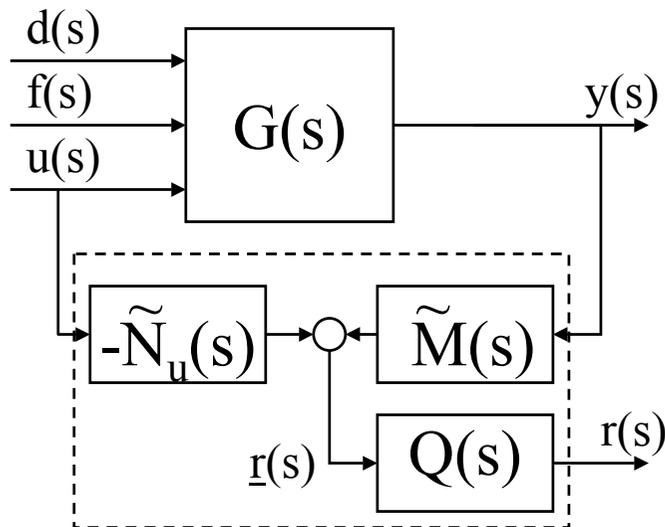
Paramétrisation de tous les générateurs de résidus :

Théorème. Sous la condition que $(E, A + L_0C)$ soit admissible, tous les générateurs de résidus sont paramétrés par $Q(s)$ de \mathbf{RH}_∞

$$r(s) = Q(s)(\tilde{M}_0(s)y(s) - \tilde{N}_0(s)u(s))$$

avec

$$\tilde{M}_0(s) = \left\{ E, \begin{pmatrix} A + L_0C & L_0 \\ C & I \end{pmatrix} \right\} \quad \tilde{N}_0(s) = \left\{ E, \begin{pmatrix} A + L_0C & B + L_0D \\ C & D \end{pmatrix} \right\}$$



Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Génération de résidus

Le résidu est obtenu en 2 étapes :

- factorisation de $G(s)$ (paramètre : L)
- filtrage : $r(s) = Q(s)\underline{r}(s)$ (paramètre : $Q(s)$ de \mathbf{RH}_∞)

Paramétrisation de tous les générateurs de résidus :

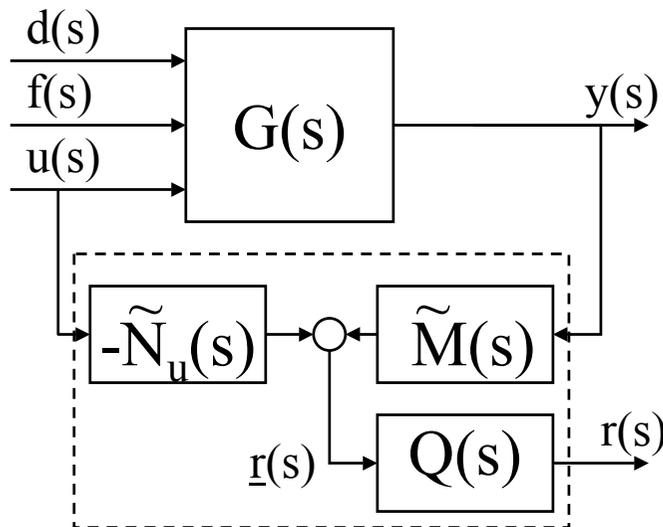
Théorème. Sous la condition que $(E, A + L_0 C)$ soit admissible, tous les générateurs de résidus sont paramétrés par $Q(s)$ de \mathbf{RH}_∞

$$r(s) = Q(s) (\tilde{M}_0(s)y(s) - \tilde{N}_0(s)u(s))$$

avec

$$\tilde{M}_0(s) = \left\{ E, \begin{pmatrix} A + L_0 C & L_0 \\ C & I \end{pmatrix} \right\} \quad \tilde{N}_0(s) = \left\{ E, \begin{pmatrix} A + L_0 C & B + L_0 D \\ C & D \end{pmatrix} \right\}$$

⇒ 1 seul paramètre : $Q(s)$



Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Synthèse du module de diagnostic

Principe : se ramener à un problème de contrôle standard H_∞ où

$$Q(s) \text{ est solution de } \min_{Q(s) \in RH_\infty} \left\| T_{wz}(s) \right\|_\infty$$

Objectifs de synthèse du module de diagnostic

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Synthèse du module de diagnostic

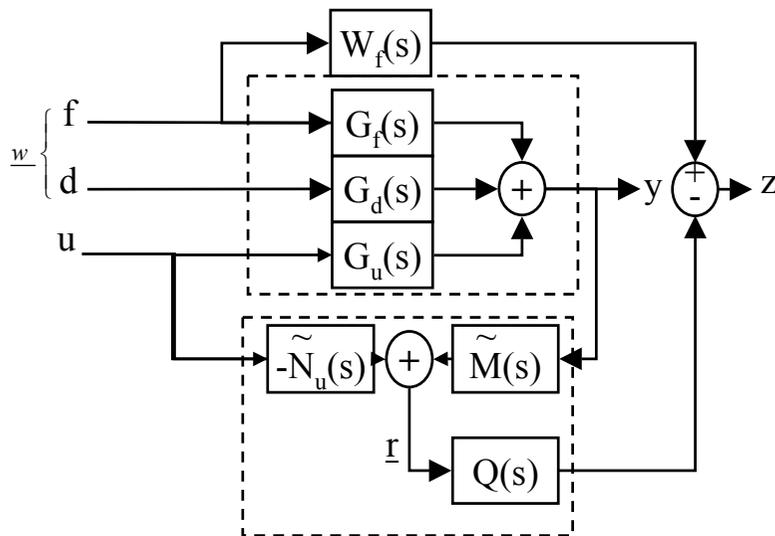
Principe : se ramener à un problème de contrôle standard H_∞ où

$$Q(s) \text{ est solution de } \min_{Q(s) \in RH_\infty} \left\| T_{wz}(s) \right\|_\infty$$

Objectifs de synthèse du module de diagnostic

1. Modeler la réponse du résidu aux fautes.

$$T_{wz}(s) = \begin{bmatrix} W_f(s) - Q(s)\tilde{N}_f(s) & -Q(s)\tilde{N}_d(s) \end{bmatrix}$$



Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Synthèse du module de diagnostic

Principe : se ramener à un problème de contrôle standard H_∞ où

$$Q(s) \text{ est solution de } \min_{Q(s) \in RH_\infty} \left\| T_{wz}(s) \right\|_\infty$$

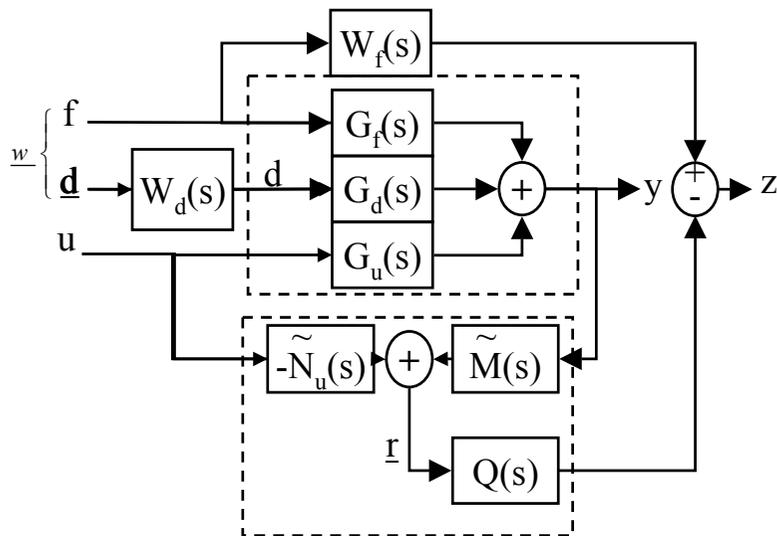
Objectifs de synthèse du module de diagnostic

1. Modeler la réponse du résidu aux fautes.

$$T_{wz}(s) = [W_f(s) - Q(s)\tilde{N}_f(s) \quad -Q(s)\tilde{N}_d(s)]$$

2. Imposer un gabarit de robustesse face aux perturbations

$$T_{wz}(s) = [W_f(s) - Q(s)\tilde{N}_f(s) \quad -Q(s)\tilde{N}_d(s)W_d(s)]$$



Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Synthèse du module de diagnostic

Principe : se ramener à un problème de contrôle standard H_∞ où

$$Q(s) \text{ est solution de } \min_{Q(s) \in RH_\infty} \left\| T_{wz}(s) \right\|_\infty$$

Objectifs de synthèse du module de diagnostic

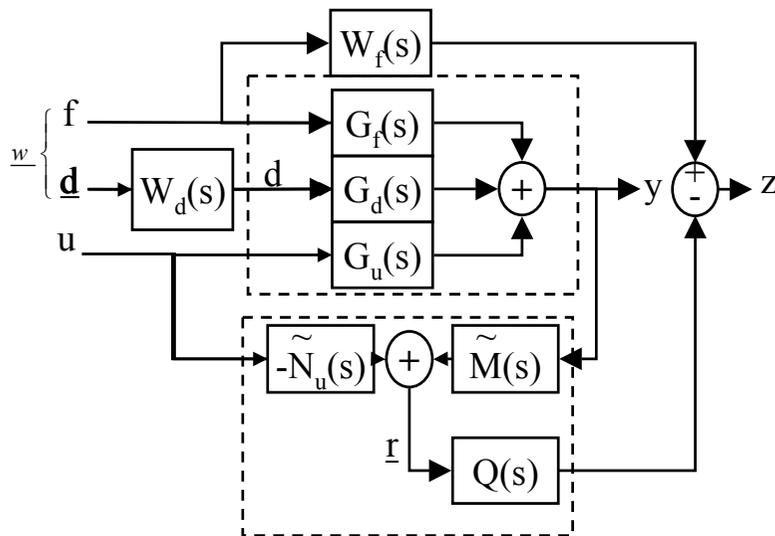
1. Modeler la réponse du résidu aux fautes.

$$T_{wz}(s) = [W_f(s) - Q(s)\tilde{N}_f(s) \quad -Q(s)\tilde{N}_d(s)]$$

2. Imposer un gabarit de robustesse face aux perturbations

$$T_{wz}(s) = [W_f(s) - Q(s)\tilde{N}_f(s) \quad -Q(s)\tilde{N}_d(s)W_d(s)]$$

3. Module de diagnostic propre : $Q(s) \in RH_\infty$



Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Synthèse du module de diagnostic

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Synthèse du module de diagnostic

- $Q(s)$ est solution du problème de poursuite de modèle suivant :

$$J_d = \min_{Q(s) \in RH_\infty} \left\| \left[W_f(s) - Q(s)\tilde{N}_f(s) \quad -W_d(s)Q(s)\tilde{N}_f(s) \right] \right\|_\infty$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Synthèse du module de diagnostic

- $Q(s)$ est solution du problème de poursuite de modèle suivant :

$$J_d = \min_{Q(s) \in RH_\infty} \left\| \begin{bmatrix} W_f(s) - Q(s)\tilde{N}_f(s) \\ -W_d(s)Q(s)\tilde{N}_f(s) \end{bmatrix} \right\|_\infty$$

- Les fonctions de transfert $W_f(s)$, $W_d(s)$, $\tilde{N}_f(s)$ et $\tilde{N}_d(s)$ sont propres

Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du module de diagnostic

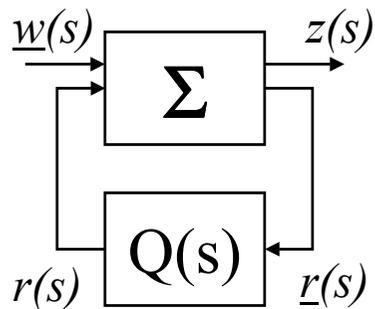
- $Q(s)$ est solution du problème de poursuite de modèle suivant :

$$J_d = \min_{Q(s) \in RH_\infty} \left\| \left[W_f(s) - Q(s)\tilde{N}_f(s) \quad -W_d(s)Q(s)\tilde{N}_f(s) \right] \right\|_\infty$$

- Les fonctions de transfert $W_f(s)$, $W_d(s)$, $\tilde{N}_f(s)$ et $\tilde{N}_d(s)$ sont propres

Solution :

résoudre le problème H_∞ standard pour le système **non singulier** :



$$\begin{cases} \dot{x} = \begin{bmatrix} \underline{A} & 0 & \underline{E}_1 C_d \\ 0 & A_f & 0 \\ 0 & 0 & A_d \end{bmatrix} x + \begin{bmatrix} \underline{R}_1 & \underline{E}_1 D_d \\ B_f & 0 \\ 0 & B_d \end{bmatrix} \begin{bmatrix} f \\ d \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} r \\ z = \begin{bmatrix} 0 & C_f & 0 \end{bmatrix} x + \begin{bmatrix} D_f & 0 \end{bmatrix} \begin{bmatrix} f \\ d \end{bmatrix} - r \\ y = \begin{bmatrix} \underline{C} & 0 & \underline{E}_2 C_d \end{bmatrix} x + \begin{bmatrix} \underline{R}_2 & \underline{E}_2 D_d \end{bmatrix} \begin{bmatrix} f \\ d \end{bmatrix} \end{cases}$$

où

$$\begin{aligned} W_f(s) &= C_f(sI - A_f)^{-1} B_f + D_f \\ \tilde{N}_f(s) &= \underline{C}(sI - \underline{A})^{-1} \underline{R}_1 + \underline{R}_2 \\ W_d(s) &= C_d(sI - A_d)^{-1} B_d + D_d \\ \tilde{N}_d(s) &= \underline{C}(sI - \underline{A})^{-1} \underline{E}_1 + \underline{E}_2 \end{aligned}$$

Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du module de diagnostic

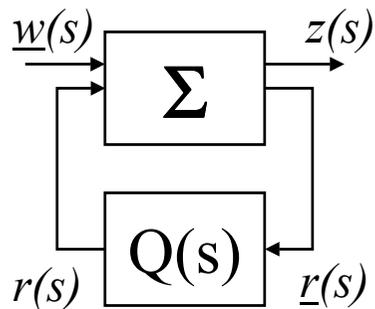
- $Q(s)$ est solution du problème de poursuite de modèle suivant :

$$J_d = \min_{Q(s) \in RH_\infty} \left\| \left[W_f(s) - Q(s)\tilde{N}_f(s) \quad -W_d(s)Q(s)\tilde{N}_f(s) \right] \right\|_\infty$$

- Les fonctions de transfert $W_f(s)$, $W_d(s)$, $\tilde{N}_f(s)$ et $\tilde{N}_d(s)$ sont propres

Solution :

résoudre le problème H_∞ standard pour le système **non singulier** :



$$\begin{cases} \dot{x} = \begin{bmatrix} \underline{A} & 0 & \underline{E}_1 C_d \\ 0 & A_f & 0 \\ 0 & 0 & A_d \end{bmatrix} x + \begin{bmatrix} \underline{R}_1 & \underline{E}_1 D_d \\ B_f & 0 \\ 0 & B_d \end{bmatrix} \begin{bmatrix} f \\ d \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} r \\ z = \begin{bmatrix} 0 & C_f & 0 \end{bmatrix} x + \begin{bmatrix} D_f & 0 \end{bmatrix} \begin{bmatrix} f \\ d \end{bmatrix} - r \\ y = \begin{bmatrix} \underline{C} & 0 & \underline{E}_2 C_d \end{bmatrix} x + \begin{bmatrix} \underline{R}_2 & \underline{E}_2 D_d \end{bmatrix} \begin{bmatrix} f \\ d \end{bmatrix} \end{cases}$$

où

$$\begin{aligned} W_f(s) &= C_f(sI - A_f)^{-1} B_f + D_f \\ \tilde{N}_f(s) &= \underline{C}(sI - \underline{A})^{-1} \underline{R}_1 + \underline{R}_2 \\ W_d(s) &= C_d(sI - A_d)^{-1} B_d + D_d \\ \tilde{N}_d(s) &= \underline{C}(sI - \underline{A})^{-1} \underline{E}_1 + \underline{E}_2 \end{aligned}$$

⇒ Solution à base de LMI (P. Gahinet & P. Apkarian, 1994)

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Diagnostic par factorisation copremière.

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Diagnostic par factorisation copremière.

Algorithme de diagnostic robuste :

1. factoriser la fonction de transfert matricielle $G_u(s)$
 - ⇒ déterminer L , tel que $(E, A+LC)$ soit admissible (LMI)
2. filtrer le pré-résidu $r(s) = Q(s)r(s)$
 - ⇒ fixer les fonctions de pondérations $W_f(s)$ et $W_d(s)$
 - ⇒ déterminer $Q(s)$: contrôle- H_∞ pour système **usuel** (LMI)

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Exemple de diagnostic par factorisation coprime

On considère le système défini par

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \dot{x} = \begin{pmatrix} -15 & 1 & 0 & 0 \\ 5 & -10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} u + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} f + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} d$$

$$y = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} f + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} d$$

avec

$$f_1(t) = \begin{cases} -u(t), & \text{pour } 1 < t < 4 \\ 0, & \text{sinon} \end{cases} \quad f_2(t) = \begin{cases} 1, & \text{pour } 3 < t < 5 \\ 0, & \text{sinon} \end{cases}$$

On applique la synthèse pour les fonctions de pondérations :

$$W_f(s) = \begin{bmatrix} \frac{1}{1+0.01s} & 0 \\ 0 & \frac{1}{1+0.01s} \end{bmatrix} \quad W_d(s) = \begin{bmatrix} \frac{1+0.001s}{1+0.01s} & 0 \\ 0 & \frac{1+0.001s}{1+0.01s} \end{bmatrix}$$

Introduction

Syst. Sing.

Factorisation

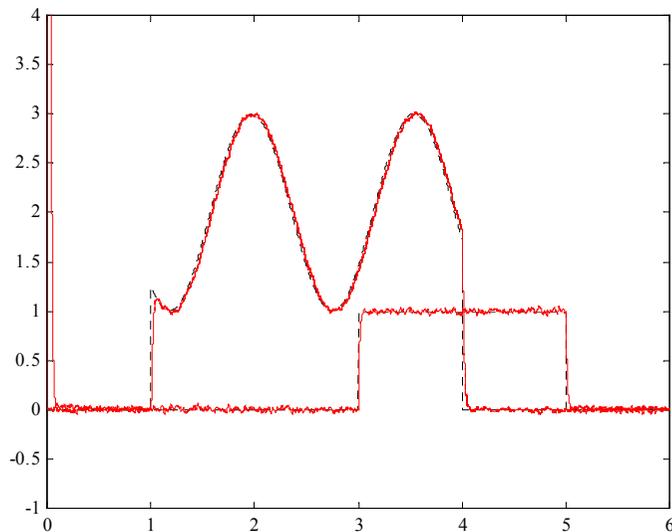
Diagnostic

Commande

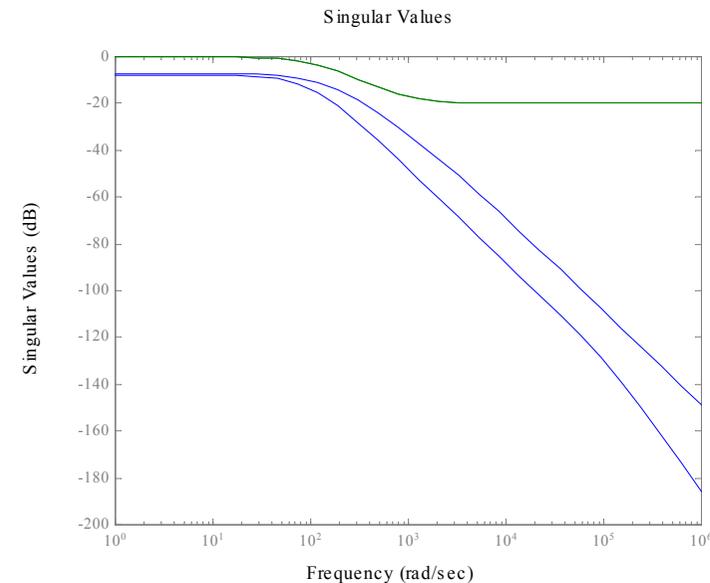
Conclusion

Exemple de diagnostic par factorisation coprime

Résultats obtenus :



Fautes (pointillés noirs)
et résidus (traits rouge)



Valeurs singulières de la sensibilité à d (bleu), et gabarit (vert)

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Commande tolérante aux fautes

- **Systeme** :
$$\begin{cases} E\dot{x} = Ax + Bu + R_1f + E_1d \\ y = Cx + Du + R_2f + E_2d \end{cases}$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Commande tolérante aux fautes

- **Systeme** :
$$\begin{cases} E\dot{x} = Ax + Bu + R_1f + E_1d \\ y = Cx + Du + R_2f + E_2d \end{cases}$$

- **Objectifs** :

- ◇ performances nominales
- ◇ tolérance aux fautes et aux perturbations
- ◇ mise en œuvre de filtres propres

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Commande tolérante aux fautes

- **Systeme** :
$$\begin{cases} E\dot{x} = Ax + Bu + R_1f + E_1d \\ y = Cx + Du + R_2f + E_2d \end{cases}$$
- **Objectifs** :
 - ◇ performances nominales
 - ◇ tolérance aux fautes et aux perturbations
 - ◇ mise en œuvre de filtres propres
- **Hypothèses** :
 - ◇ (E, A) régulier
 - ◇ (E, A, B) Imp-commandable et stabilisable
 - ◇ (E, A, C) Imp-observable et détectable

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Commande tolérante aux fautes

- **Système** :
$$\begin{cases} E\dot{x} = Ax + Bu + R_1f + E_1d \\ y = Cx + Du + R_2f + E_2d \end{cases}$$
- **Objectifs** :
 - ◇ performances nominales
 - ◇ tolérance aux fautes et aux perturbations
 - ◇ mise en œuvre de filtres propres
- **Hypothèses** :
 - ◇ (E, A) régulier
 - ◇ (E, A, B) Imp-commandable et stabilisable
 - ◇ (E, A, C) Imp-observable et détectable
- **Méthode** :
 - ◇ factorisation copremière
 - ◇ formalisme H_∞ standard pour systèmes usuels

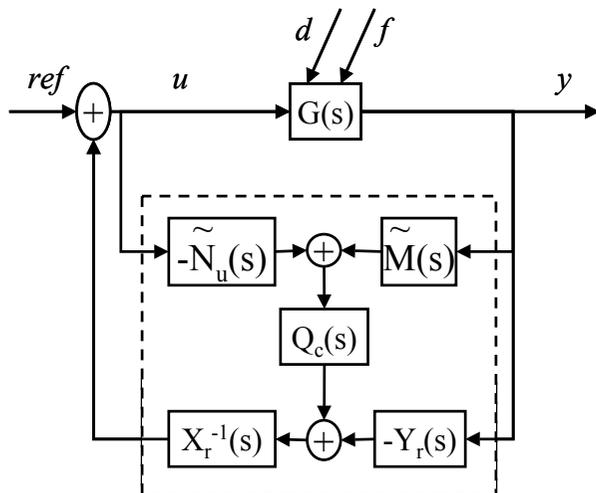
Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du correcteur pour la commande tolérante aux fautes

Lemme. (Takaba *et al.*, 1994) Tous les correcteurs stabilisants de $G_u(s)$ sont paramétrés par

$$u(s) = (X_r(s) + Q(s)\tilde{N}_u(s))(Q(s)\tilde{M}_u(s) - Y_r(s))y(s)$$

où $Q(s) \in RH_\infty$



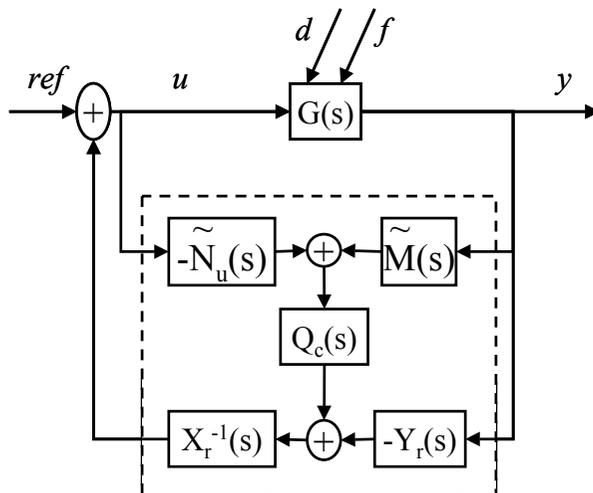
Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du correcteur pour la commande tolérante aux fautes

Lemme. (Takaba *et al.*, 1994) Tous les correcteurs stabilisants de $G_u(s)$ sont paramétrés par

$$u(s) = (X_r(s) + Q(s)\tilde{N}_u(s))(Q(s)\tilde{M}_u(s) - Y_r(s))y(s)$$

où $Q(s) \in RH_\infty$



Structure du correcteur proposé (Youla)

- commande :

$$u(s) = (X_r(s) + Q_c(s)\tilde{N}_u(s))X_r(s)ref(s) + \dots \\ (X_r(s) + Q_c(s)\tilde{N}_u(s))(Q_c(s)\tilde{M}(s) - Y_r(s))y(s)$$

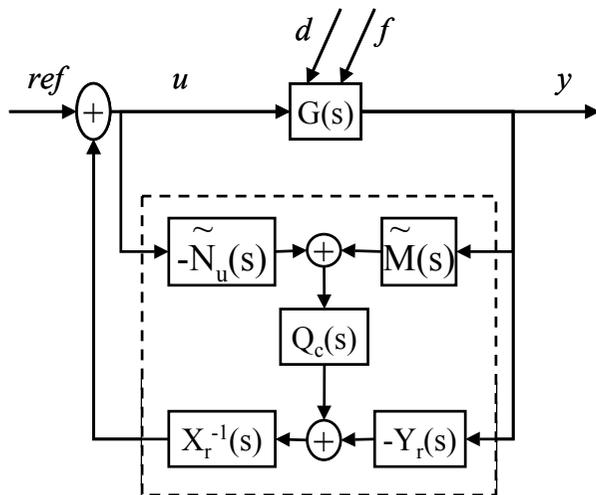
Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du correcteur pour la commande tolérante aux fautes

Lemme. (Takaba *et al.*, 1994) Tous les correcteurs stabilisants de $G_u(s)$ sont paramétrés par

$$u(s) = (X_r(s) + Q(s)\tilde{N}_u(s))(Q(s)\tilde{M}_u(s) - Y_r(s))y(s)$$

où $Q(s) \in RH_\infty$



Structure du correcteur proposé (Youla)

- commande :

$$u(s) = (X_r(s) + Q_c(s)\tilde{N}_u(s))X_r(s)ref(s) + \dots \\ (X_r(s) + Q_c(s)\tilde{N}_u(s))(Q_c(s)\tilde{M}(s) - Y_r(s))y(s)$$

- sortie en boucle fermée :

$$y(s) = X_l(s)\tilde{N}_u(s)ref(s) + (X_l(s) + N_u(s)Q_c(s))(\tilde{N}_f(s)f(s)\tilde{N}_d(s)d(s))$$

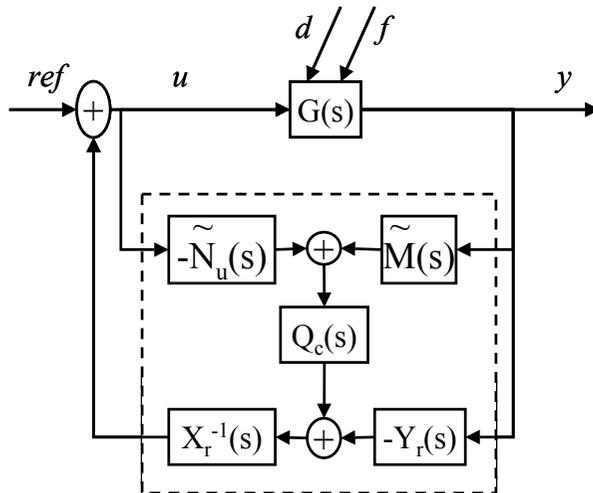
Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du correcteur pour la commande tolérante aux fautes

Lemme. (Takaba *et al.*, 1994) Tous les correcteurs stabilisants de $G_u(s)$ sont paramétrés par

$$u(s) = (X_r(s) + Q(s)\tilde{N}_u(s))(Q(s)\tilde{M}_u(s) - Y_r(s))y(s)$$

où $Q(s) \in RH_\infty$



Structure du correcteur proposé (Youla)

- commande :

$$u(s) = (X_r(s) + Q_c(s)\tilde{N}_u(s))X_r(s)ref(s) + \dots \\ (X_r(s) + Q_c(s)\tilde{N}_u(s))(Q_c(s)\tilde{M}(s) - Y_r(s))y(s)$$

- sortie en boucle fermée :

$$y(s) = X_l(s)\tilde{N}_u(s)ref(s) + (X_l(s) + N_u(s)Q_c(s))(\tilde{N}_f(s)f(s)\tilde{N}_d(s)d(s))$$

réponse nominale

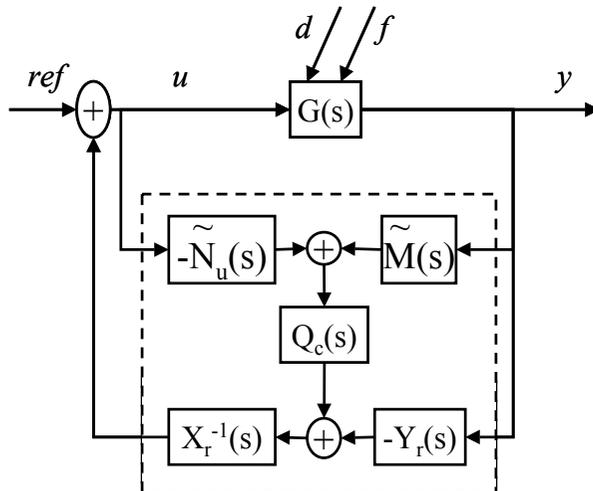
Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du correcteur pour la commande tolérante aux fautes

Lemme. (Takaba *et al.*, 1994) Tous les correcteurs stabilisants de $G_u(s)$ sont paramétrés par

$$u(s) = (X_r(s) + Q(s)\tilde{N}_u(s))(Q(s)\tilde{M}_u(s) - Y_r(s))y(s)$$

où $Q(s) \in RH_\infty$



Structure du correcteur proposé (Youla)

- commande :

$$u(s) = (X_r(s) + Q_c(s)\tilde{N}_u(s))X_r(s)ref(s) + \dots \\ (X_r(s) + Q_c(s)\tilde{N}_u(s))(Q_c(s)\tilde{M}(s) - Y_r(s))y(s)$$

- sortie en boucle fermée :

$$y(s) = \underbrace{X_l(s)\tilde{N}_u(s)ref(s)}_{\text{réponse nominale}} + \underbrace{(X_l(s) + N_u(s)Q_c(s))(\tilde{N}_f(s)f(s)\tilde{N}_d(s)d(s))}_{\text{écart dû aux signaux exogènes}}$$

réponse nominale

écart dû aux signaux exogènes

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Synthèse du correcteur pour la commande tolérante aux fautes

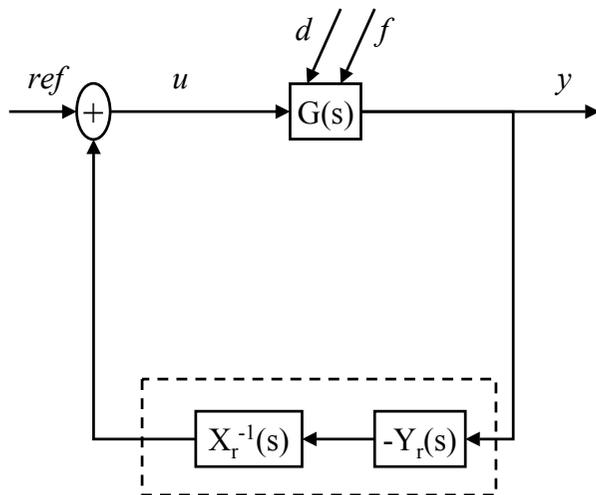
Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du correcteur pour la commande tolérante aux fautes

- Réponse nominale \Leftrightarrow retour d'état observé

$$X_l(s)\tilde{N}_u(s) = \left\{ \begin{array}{c|c} \begin{pmatrix} E & 0 \\ 0 & E \end{pmatrix} & \begin{pmatrix} A+BF & -LC & -LD \\ 0 & A+LC & B+LD \end{pmatrix} \\ \hline C+DF & C & D \end{array} \right\}$$

L et F déterminés afin de placer les pôles
 \Rightarrow résolution de LMI strictes



$$y(s) = X_l(s)\tilde{N}_u(s)ref(s)$$

Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du correcteur pour la commande tolérante aux fautes

- **Réponse nominale** \Leftrightarrow retour d'état observé

$$X_l(s)\tilde{N}_u(s) = \left\{ \begin{array}{c|c} \begin{pmatrix} E & 0 \\ 0 & E \end{pmatrix} & \begin{pmatrix} A+BF & -LC \\ 0 & A+LC \end{pmatrix} \\ \hline C+DF & C \end{array} \begin{array}{c} -LD \\ B+LD \\ D \end{array} \right\}$$

L et F déterminés afin de placer les pôles
 \Rightarrow résolution de LMI strictes

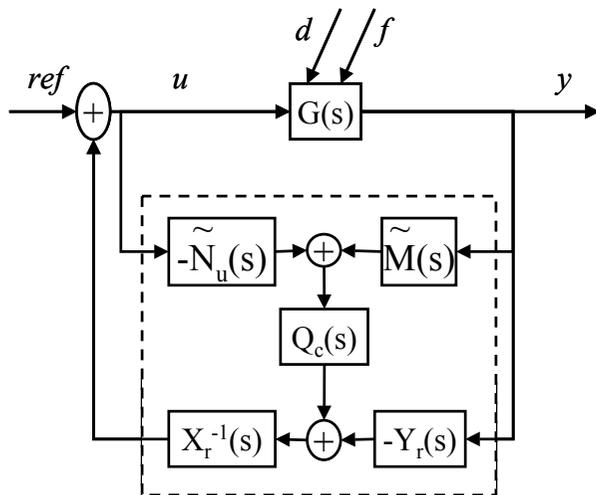
- **Tolérance aux fautes** \Leftrightarrow poursuite de modèle

$$J_c = \min_{Q_c(s) \in RH_\infty} \left\| (X_l(s) + N_u(s)Q_c(s)) \begin{bmatrix} \tilde{N}_f(s) & \tilde{N}_d(s) \end{bmatrix} \right\|_\infty$$

$Q_c(s)$ déterminé par synthèse d'un correcteur H_∞
 pour système non singulier

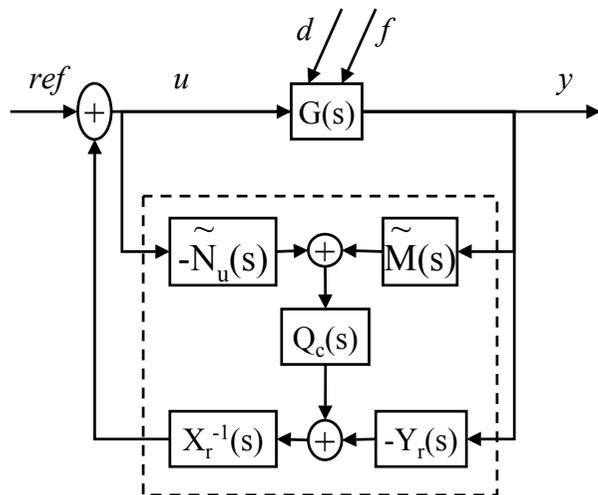
\Rightarrow résolution par LMI

$$y(s) = X_l(s)\tilde{N}_u(s)ref(s) + (X_l(s) + N_u(s)Q_c(s))(\tilde{N}_f(s)f(s)\tilde{N}_d(s)d(s))$$



Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du correcteur pour la commande tolérante aux fautes



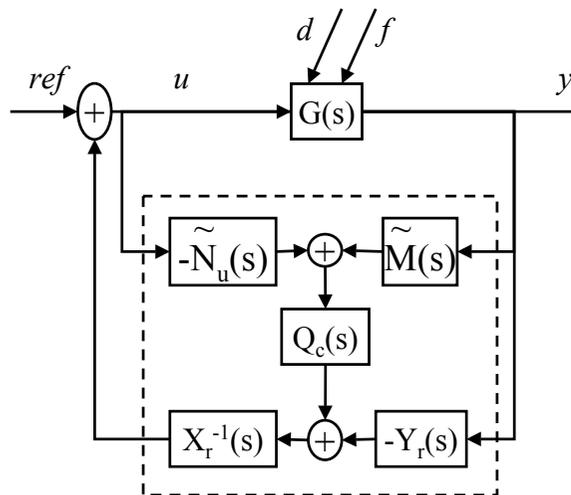
Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du correcteur pour la commande tolérante aux fautes

- **Algorithme de contrôle tolérant aux fautes**

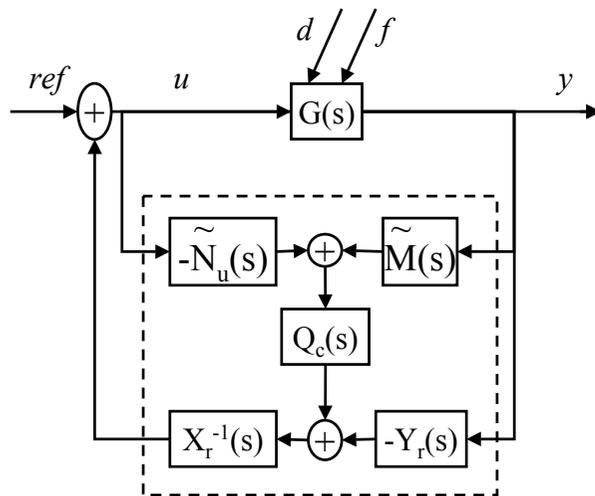
1. Déterminer F et L , afin de satisfaire les objectifs de contrôle nominal.

2. Déterminer $Q_c(s)$, afin de minimiser la déviation de sortie due à $d(s)$ et $f(s)$



Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Synthèse du correcteur pour la commande tolérante aux fautes



- **Algorithme de contrôle tolérant aux fautes**

1. Déterminer F et L , afin de satisfaire les objectifs de contrôle nominal.
2. Déterminer $Q_c(s)$, afin de minimiser la déviation de sortie due à $d(s)$ et $f(s)$

- **Intérêts de cette approche**

1. Séparer performances nominales // tolérance aux fautes
2. Implémentation de filtres propres
3. Eviter de masquer $d(s)$ et $f(s)$ par la commande.

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Commande tolérante aux fautes adaptative

mais...

Commande tolérante à TOUS les signaux exogènes

⇒ pessimisme

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Commande tolérante aux fautes adaptative

mais...

Commande tolérante à TOUS les signaux exogènes

⇒ pessimisme

...amélioration proposée

1. Introduire un module de diagnostic $Q_d(s)$

⇒ localisation des défauts occurents

2. Choisir *en ligne* un filtre de contrôle parmi un banc

⇒ correcteur adapté au(x) défaut(s) survenu(s)

Introduction

Syst. Sing.

Factorisation

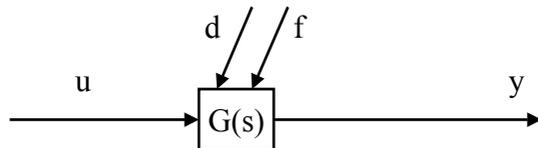
Diagnostic

Commande

Conclusion

Commande tolérante aux fautes adaptative

Algorithme de contrôle tolérant aux fautes adaptatif.

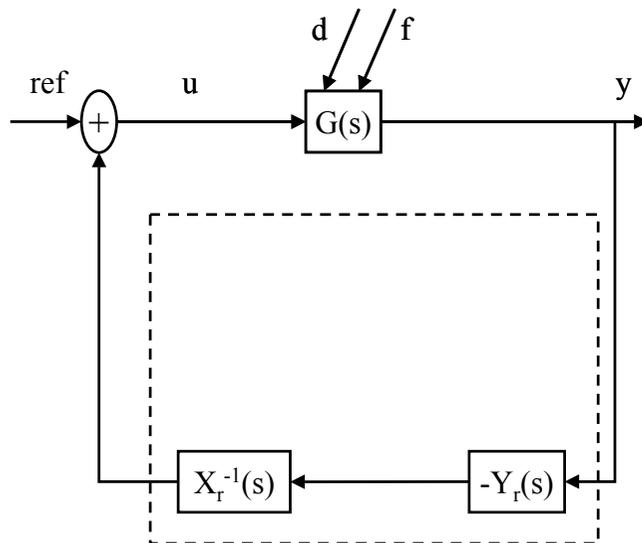


Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Commande tolérante aux fautes adaptative

Algorithme de contrôle tolérant aux fautes adaptatif.

- 1 Déterminer F et L tels que $(E, A+BF)$ et $(E, A+LC)$ satisfassent les objectifs de contrôle nominal



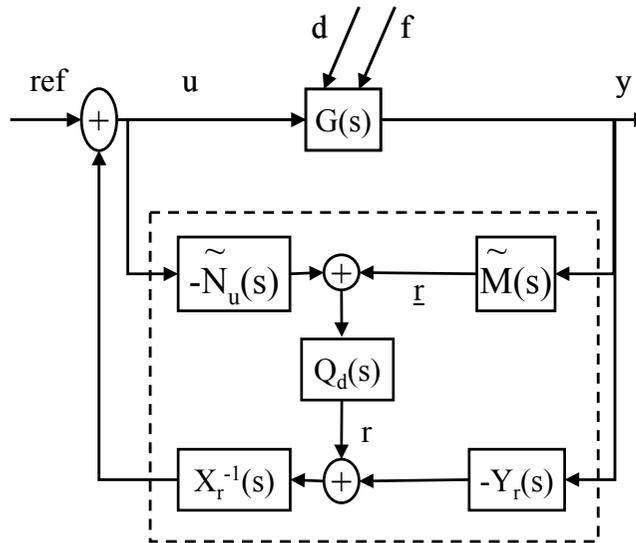
Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Commande tolérante aux fautes adaptative

Algorithme de contrôle tolérant aux fautes adaptatif.

1 Déterminer F et L tels que $(E, A+BF)$ et $(E, A+LC)$ satisfassent les objectifs de contrôle nominal

2 Déterminer le filtre de diagnostic robuste $Q_d(s)$ qui minimise $J_d = \min_{Q_d(s) \in RH_\infty} \|W_f(s) - Q_d(s)\tilde{N}_d(s)W_d(s)\|_\infty$



Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

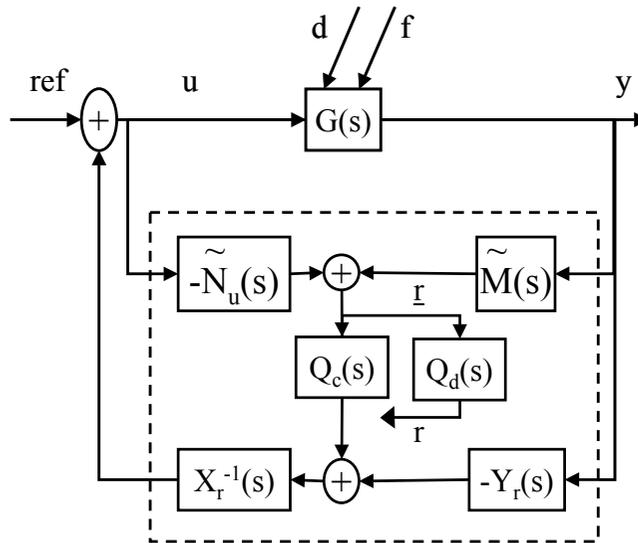
Commande tolérante aux fautes adaptative

Algorithme de contrôle tolérant aux fautes adaptatif.

- 1 Déterminer F et L tels que $(E, A+BF)$ et $(E, A+LC)$ satisfassent les objectifs de contrôle nominal
- 2 Déterminer le filtre de diagnostic robuste $Q_d(s)$ qui minimise $J_d = \min_{Q_d(s) \in RH_\infty} \left\| W_f(s) - Q_f(s)\tilde{N}_f(s) - Q_d(s)\tilde{N}_d(s)W_d(s) \right\|_\infty$
- 3 Déterminer n_f filtres $Q_{ci}(s)$ dédiés à une faute f_i , et un filtre $Q_{c0}(s)$ pour le cas "sans faute", par minimisation de

$$J_{ci} = \min_{Q_{ci}(s) \in RH_\infty} \left\| (X_l(s) + N_u(s)Q_c(s)) \begin{bmatrix} \tilde{N}_{fi}(s) & \tilde{N}_d(s) \end{bmatrix} \right\|_\infty$$

$$J_{c0} = \min_{Q_{c0}(s) \in RH_\infty} \left\| (X_l(s) + N_u(s)Q_c(s))\tilde{N}_d(s) \right\|_\infty$$



Introduction
Syst. Sing.
Factorisation
Diagnostic
Commande
Conclusion

Commande tolérante aux fautes adaptative

Algorithme de contrôle tolérant aux fautes adaptatif.

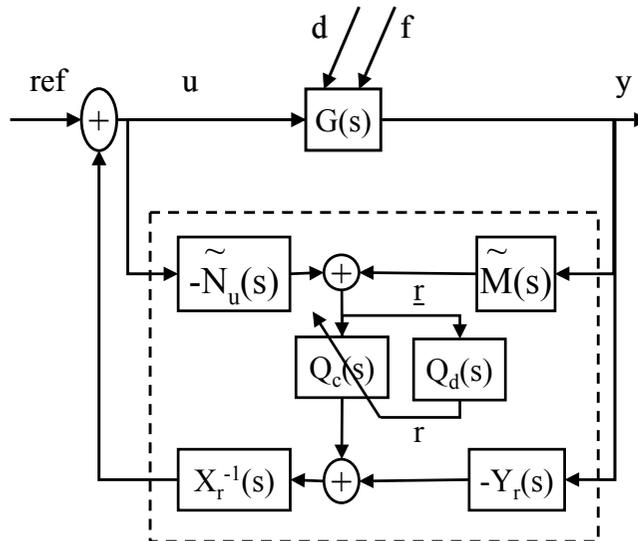
- 1 Déterminer F et L tels que $(E, A+BF)$ et $(E, A+LC)$ satisfassent les objectifs de contrôle nominal
- 2 Déterminer le filtre de diagnostic robuste $Q_d(s)$ qui minimise $J_d = \min_{Q_d(s) \in RH_\infty} \|W_f(s) - Q_f(s)\tilde{N}_f(s) - Q_d(s)\tilde{N}_d(s)W_d(s)\|_\infty$
- 3 Déterminer n_f filtres $Q_{ci}(s)$ dédiés à une faute f_i , et un filtre $Q_{c0}(s)$ pour le cas "sans faute", par minimisation de

$$J_{ci} = \min_{Q_{ci}(s) \in RH_\infty} \left\| (X_l(s) + N_u(s)Q_c(s)) \begin{bmatrix} \tilde{N}_{fi}(s) & \tilde{N}_d(s) \end{bmatrix} \right\|_\infty$$

$$J_{c0} = \min_{Q_{c0}(s) \in RH_\infty} \left\| (X_l(s) + N_u(s)Q_c(s))\tilde{N}_d(s) \right\|_\infty$$

- 4 Implémenter le contrôleur ci-contre, avec

$$Q_c(s) = \begin{cases} Q_{ci}(s), & \text{si } r_i > J_d \\ Q_{c0}(s), & \text{sinon} \end{cases}$$



Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Application de la commande tolérante aux fautes

On considère le système défini par

$$\begin{cases} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \dot{x} = \begin{pmatrix} -15 & 1 & 0 & 0 \\ 5 & -10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} u + \begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} f + \begin{pmatrix} 0,01 \\ 0 \\ 0 \\ 0 \end{pmatrix} d \\ y = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} f \end{cases}$$

avec

$$\begin{cases} f_1(t) = \begin{cases} 2, & \text{pour } 1 < t < 2 \\ 0, & \text{sin on} \end{cases} \\ f_2(t) = \begin{cases} 2, & \text{pour } 3,5 < t < 4,5 \\ 0, & \text{sin on} \end{cases} \end{cases}$$

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

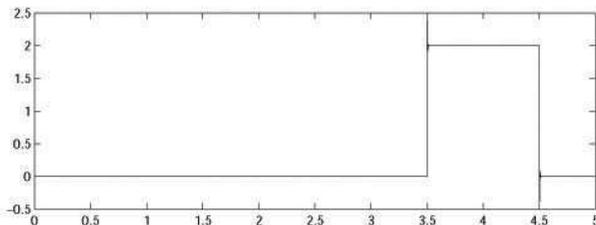
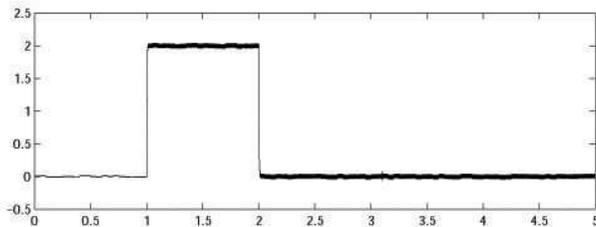
Application de la commande tolérante aux fautes

On considère le système défini par

$$\begin{cases} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \dot{x} = \begin{pmatrix} -15 & 1 & 0 & 0 \\ 5 & -10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} u + \begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} f + \begin{pmatrix} 0,01 \\ 0 \\ 0 \\ 0 \end{pmatrix} d \\ y = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} f \end{cases}$$

avec

$$\begin{cases} f_1(t) = \begin{cases} 2, & \text{pour } 1 < t < 2 \\ 0, & \text{sin on} \end{cases} \\ f_2(t) = \begin{cases} 2, & \text{pour } 3,5 < t < 4,5 \\ 0, & \text{sin on} \end{cases} \end{cases}$$



Résidus fournis par le module de diagnostic

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

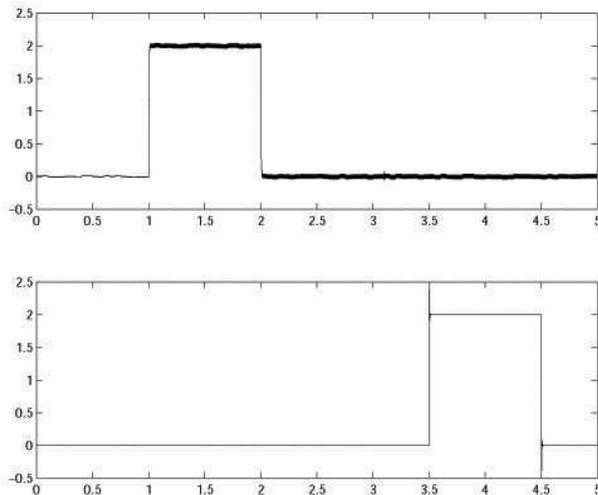
Application de la commande tolérante aux fautes

On considère le système défini par

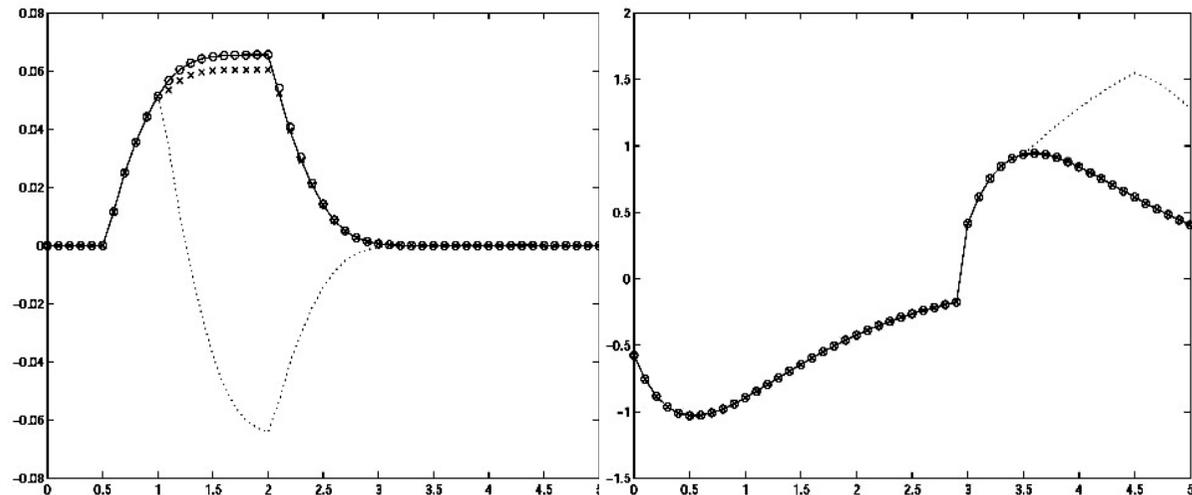
$$\begin{cases} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \dot{x} = \begin{pmatrix} -15 & 1 & 0 & 0 \\ 5 & -10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} u + \begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} f + \begin{pmatrix} 0,01 \\ 0 \\ 0 \\ 0 \end{pmatrix} d \\ y = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} f \end{cases}$$

avec

$$\begin{cases} f_1(t) = \begin{cases} 2, & \text{pour } 1 < t < 2 \\ 0, & \text{sin on} \end{cases} \\ f_2(t) = \begin{cases} 2, & \text{pour } 3,5 < t < 4,5 \\ 0, & \text{sin on} \end{cases} \end{cases}$$



Résidus fournis par le module de diagnostic



Sorties, contrôle nominal avec $d=f=0$ (ronds), contrôle par retour d'état observé (pointillés), CTF (croix) et CTFA (trait).

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Conclusion

On a proposé une méthode pour traiter les problèmes de

1/ diagnostic robuste

2/ commande tolérante aux fautes

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Conclusion

On a proposé une méthode pour traiter les problèmes de

1/ diagnostic robuste

2/ commande tolérante aux fautes



problème de contrôle standard H_∞ de systèmes usuels

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Conclusion

On a proposé une méthode pour traiter les problèmes de

1/ diagnostic robuste

2/ commande tolérante aux fautes



problème de contrôle standard H_∞ de systèmes usuels



Solution à base de LMI

(boîtes à outils Matlab ou Scilab)

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Conclusion

On a proposé une méthode pour traiter les problèmes de

1/ diagnostic robuste

2/ commande tolérante aux fautes



problème de contrôle standard H_∞ de systèmes usuels



Solution à base de LMI

(boîtes à outils Matlab ou Scilab)



Implémentation simple (filtres propres)

Introduction

Syst. Sing.

Factorisation

Diagnostic

Commande

Conclusion

Publications

diagnostic
robuste

- ◇ B. Marx, D. Koenig et D. Georges, Robust fault diagnosis for descriptor systems – a coprime factorization approach, Proc of the *IFAC SAFEPROCESS'03*, Washington, USA, 2003.

placement de
pôles

- ◇ B. Marx, D. Koenig et D. Georges, Robust pole-clustering in LMI regions for descriptor systems, a characterization via strict LMIs, Proc of the *European Control Conference*, Cambridge, UK, 2003.

contrôle tolérant
aux fautes

- ◇ B. Marx, D. Koenig et D. Georges, Fault Diagnosis and Robust Fault Tolerant Control for Descriptor Systems, soumis à *Automatica* (2003)