

Commande prédictive (MPC) et implémentation par MATLAB

1

Contents

1. Commande prédictive (MPC)
 - 1.1. Concepts de base
 - 1.2. Algorithme MPC
 - 1.3. Types de MPC
 - 1.3.1. MPC Linéaire discret -Prédiction pour les modèles mono-sortie mono-entrée (SISO)?
* Exemples (1,2,3)
 - 1.3.2. Prédiction MPC linéaire pour les modèles MIMO
 - 1.4. Estimation prédictive de l'état MPC linéaire
* Exemple 4
 - 1.5. MPC Linéaire Continue -Prédiction pour les modèles SISO?
 - 1.6. LQR et MPC Linéaire
 - 1.6.1. Contraintes
 - 1.6.2. Problème d'optimisation LQR - dans MATLAB
 - 1.6.3. Problème d'optimisation MPC - dans MATLAB
* Exemple 5
2. MPC basé sur les inégalités matricielles linéaires (LMI)
 - 2.1. Inégalités matricielles linéaires dans le contrôle (LMI) dans Matlab
 - 2.2. Exemple 6
 - 2.3. Commande prédictive robuste (RMPC) utilisant des LMI
3. Application aux liaisons HVDC

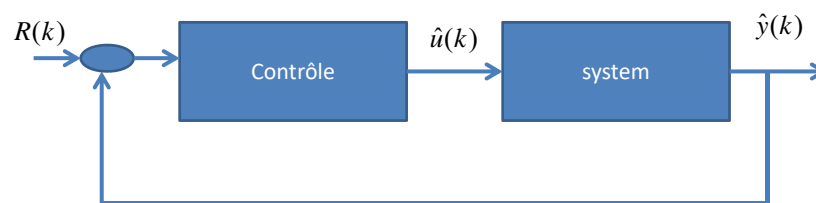
2

1. Commande prédictive (MPC)

L'objectif de MPC

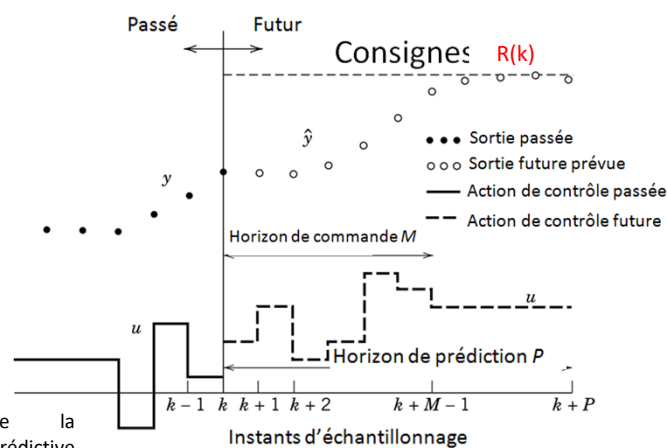
- Prédire l'entrée de contrôle
- Prédiction de la sortie du système

Dans une fenêtre temporelle



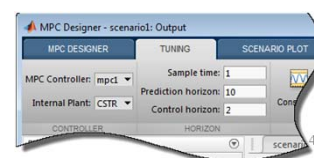
3

1.1 Concepts de base



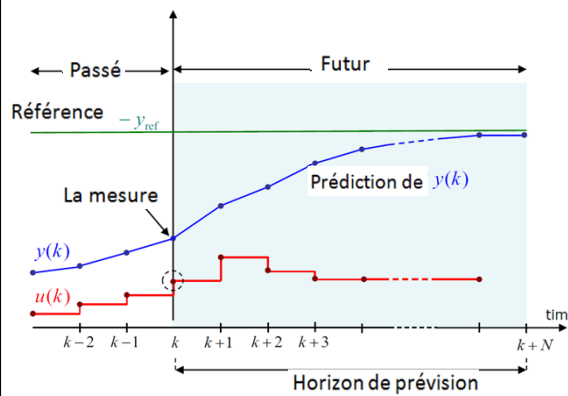
l'objectif de la commande prédictive est de Minimiser les erreurs entre sortie du system et le signal de consigne à l'instant k, dans un horizon de prédiction p,

Concept de base pour MPC



1.1 Concepts de base

Principe de la commande prédictive



$$\min_{u(k), \dots, u(k+N-1)} \sum_{i=1}^N (y_{\text{ref}} - y(k+i))^2$$

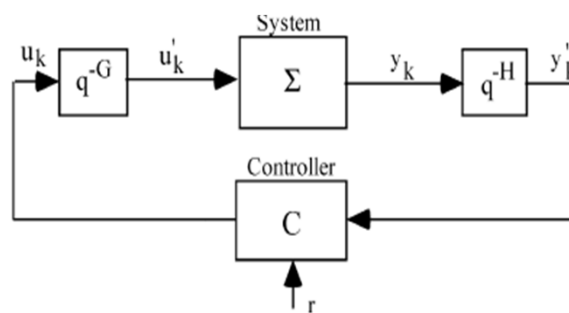
En respectant :

- Modèle du processus
- Contraintes sur l'entrée
- Contraintes sur la sortie/état

Concept de base pour MPC

5

1.1 Concepts de base



Boucle fermée générale avec retards

6

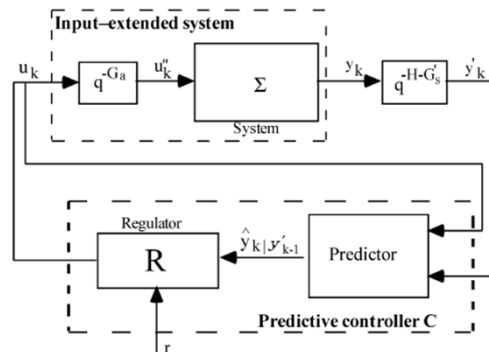
1.1 Concepts de base

Stratégie de contrôle prédictive (MPC)

- MPC est l'équivalent de deux parties :

-Contrôle optimal
- et observateur

- l'observateur pour estimer les états du système



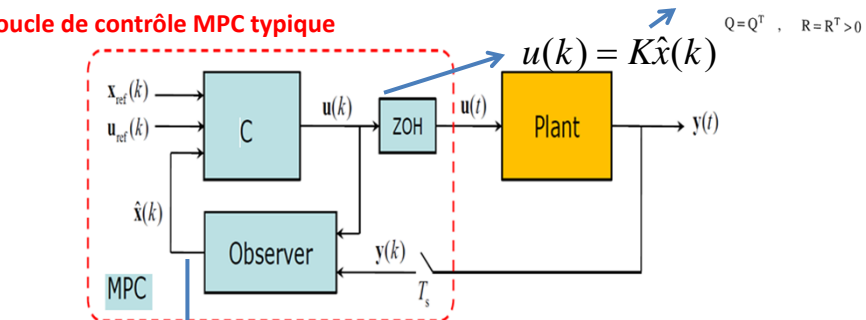
Stratégie de contrôle prédictive

7

1.2. Algorithme MPC

$$[K] = LQR(A, B, Q, R)$$

Boucle de contrôle MPC typique



Estimation d'état

Filtre de Kalman

Observateur

$$\hat{x}(k+1) = A_m \hat{x}(k) + B_m u(k) + K_{obs} (y(k) - \hat{y}(k))$$

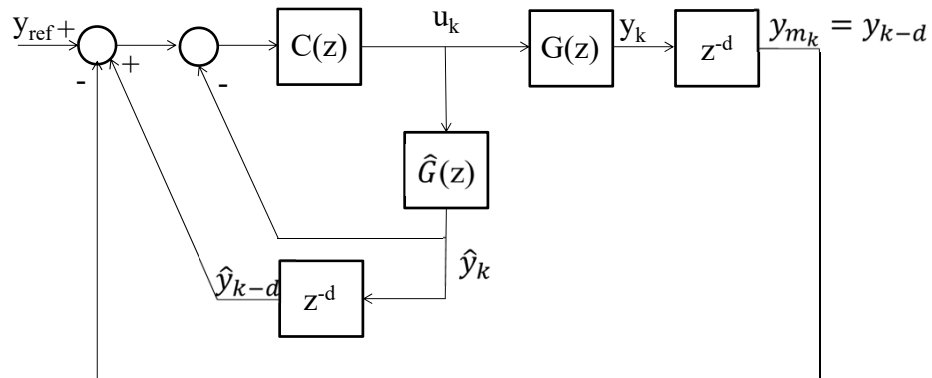
$$\hat{y}(k) = C_m \hat{x}(k)$$

Nous utilisons la commande MATLAB 'place' pour produire la matrice de gain
Si Pôle = observateur pôles désirés

$$K_{obs} = \text{place}(A', C', \text{Pole})$$

8

Autres rédictes pour MPC



Smith predictor:

Autres rédictes pour MPC

Prédicteur d'état avec mesure d'état:

$$\hat{x}_{k+d|k} = A^d x_k + \sum_{j=k+1}^{k+d} A^{k+d-j} B u_{j-d}$$

1.3 Types de MPC

MPC linéaire: il utilise un modèle linéaire du système $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$

MPC non linéaire: il utilise un modèle non linéaire du système $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$

1.3.1. Prédiction pour les modèles mono-sortie mono-entrée (SISO)

1.3.2. Prédiction pour les modèles multi-sortie multi-entrées (MIMO)

Dans le MPC linéaire, nous pouvons utiliser n'importe quel modèle linéaire

- Modèle d'état
- Réponse impulsionnelle
- Modèle ARX
- Réponse à un échelon

11

1.3.1. Prédiction pour les modèles mono-sortie mono-entrée (SISO)

Modèle d'état

Le signal de commande prédictif optimal est donné par

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (\bar{R}_r r(k) - Fx(k))$$

Preuve

1. Trouver le modèle augmenté, qui sera utilisé dans la conception du contrôle prédictif.

Système à **mono-sortie mono-entrée**, décrit par

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \quad (1)$$

$$y(k) = C_m x_m(k) \quad (2)$$

u est la commande; y est la sortie; and x_m est le vecteur d'état avec une dimension supposée $n1$.

A_m , B_m et C_m sont de dimension $n1 \times n1$, $n1 \times 1$ et $1 \times n1$, respectivement

en décalant de 1 pas dans le passé

$$x_m(k) = A_m x_m(k-1) + B_m u(k-1) \quad (3)$$

En soustrayant (1) par (3)

$$x_m(k+1) - x_m(k) = A_m (x_m(k) - x_m(k-1)) + B_m (u(k) - u(k-1)) \quad (4)$$

Posons :

$$\Delta x_m(k+1) = x_m(k+1) - x_m(k)$$

$$\Delta x_m(k) = x_m(k) - x_m(k-1) \quad (5)$$

$$\Delta u(k) = u(k) - u(k-1)$$

L'équation d'état est donnée par :

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \quad (6)$$

12

1.3.1. Prédiction pour les modèles mono-sortie mono-entrée (SISO)

Modèle d'état

De même $y(k) = C_m x_m(k)$ (2)

en décalant de 1 pas dans le futur $y(k+1) = C_m x_m(k+1)$ (7)

En soustrayant (7) par (2) $y(k+1) - y(k) = C_m (x_m(k+1) - x_m(k))$ (8)

De (5) $y(k+1) - y(k) = C_m \Delta x_m(k+1)$ (9)

De (6) $y(k+1) - y(k) = C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k)$ (10)

$$y(k+1) = y(k) + C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) \quad (11)$$

En assemblant (6) avec (11), cela conduit au modèle d'espace d'état suivant:

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & 0_m^T \\ C_m A_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) \quad (12)$$

Puis $y(k) = \begin{bmatrix} 0_m^T & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$ (13)

13

1.3.1. Prédiction pour les modèles mono-sortie mono-entrée (SISO)

Modèle d'état

Ensuite, le modèle augmenté

$$\begin{aligned} x(k+1) &= A x(k) + B \Delta u(k) \\ y(k) &= C x(k) \end{aligned} \quad (14)$$

où

$$x(k+1) = \begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix}, \quad A = \begin{bmatrix} A_m & 0_m^T \\ C_m A_m & 1 \end{bmatrix}, \quad x(k) = \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}, \quad B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}, \quad C = \begin{bmatrix} 0_m^T & 1 \end{bmatrix}$$

où

$$0_m^T = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}^{n_1} \quad \text{Avec la dimension } n_1$$

Le triplet (A, B, C) est appelé le modèle augmenté, qui sera utilisé dans la conception du contrôle prédictif.

Basé sur le le modèle augmenté (14), trouvez la commande de prédiction dans une fenêtre d'optimisation

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k))$$

14

1.3.1. Prédiction pour les modèles mono-sortie mono-entrée (SISO)

Modèle d'état

2. Trouvez la commande de prédiction dans une fenêtre d'optimisation

2.1. Trouver la prédiction des variables d'état et de sortie

En supposant qu'à l'instant d'échantillonnage k , $k > 0$, le vecteur de variable d'état $x(k)$ est mesurable, l'état $x(k)$ fournit les informations actuelles du système.

La futur trajectoire de commande est notée :

$$\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+N_c-1)$$

- où N_c est l'Horizon de commande .

Les variables d'état futures sont notées :

$$x(k_i + 1|_{k_i}), x(k_i + 2|_{k_i}), \dots, x(k_i + m|_{k_i}), \dots, \Delta u(k_i + N_p|_{k_i})$$

- N_p est l'Horizon de prédiction .
- N_p est également la longueur de la fenêtre d'optimisation.

où $x(k_i + m|_{k_i})$ est l'état prédit à $k_i + m$ avec l'information actuelle du système donnée par $x(k_i)$. L'horizon de contrôle N_c est choisi inférieur (ou égal) à l'horizon de prédiction N_p .

$$N_c \leq N_p$$

15

1.3.1. Prédiction pour les modèles mono-sortie mono-entrée (SISO)

Modèle d'état

Sur la base du modèle d'espace d'état (A, B, C), les variables d'état futures sont calculées séquentiellement en utilisant l'ensemble des commandes futures:

Les variables d'état prédites

$$x(k+1) = Ax(k) + B\Delta u(k) \quad \leftarrow \text{modèle augmenté} \quad (15)$$

$$x(k+2) = Ax(k+1) + B\Delta u(k+1) \quad \leftarrow \text{(en décalant de 1 pas dans le futur)}$$

$$= A(Ax(k) + B\Delta u(k)) + B\Delta u(k+1)$$

$$x(k+2) = A^2x(k) + AB\Delta u(k) + B\Delta u(k+1) \quad (16)$$

$$\vdots$$

$$x(k+N_p) = A^{N_p}x(k) + A^{N_p-1}B\Delta u(k) + A^{N_p-2}B\Delta u(k+1) + \dots + A^{N_p-N_c}B\Delta u(k+N_c-1) \quad (17)$$

À partir des variables d'état prédites, les variables de sortie prédites sont, par substitution

$$y(k+1) = CAx(k) + CB\Delta u(k) \quad (18)$$

$$y(k+2) = CA^2x(k) + CAB\Delta u(k) + CB\Delta u(k+1) \quad (19)$$

$$\vdots$$

$$y(k+N_p) = CA^{N_p}x(k) + CA^{N_p-1}B\Delta u(k) + CA^{N_p-2}B\Delta u(k+1) + \dots + CA^{N_p-N_c}B\Delta u(k+N_c-1) \quad (20)$$

16

Notez que toutes les variables prédites sont formulées en fonction d'informations sur les variables d'état courant $x(k)$ et sur les commandes futures $\Delta u(k+j)$, où $j = 0, 1, \dots, N_p - 1$.

Définissons des vecteurs

$$Y = [y(k+1) \ y(k+2), \dots, y(k+N_p)]^T$$

$$\Delta U = [\Delta u(k) \ \Delta u(k+1) \ \Delta u(k+2), \dots, \Delta u(k+N_c-1)]^T$$

Nous concaténons (18) à (20) ensemble dans une forme de matrice compacte

$$Y = F x(k) + \Phi \Delta U \quad (21)$$

où

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}; \Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix}$$

17

2.2. Optimisation

Pour un signal de consigne donné $r(k)$ à l'instant k , dans un horizon de prédiction, l'objectif de la commande prédictive est d'amener la sortie prédite aussi proche que possible du signal de consigne,

la fonction d'erreur entre le point de consigne et la sortie prédite est minimisée.

En supposant que le vecteur de données contenant les informations de point de consigne est

$$R_s^T = \overbrace{[1 \ 1 \ \dots 1]}^{N_p} r(k)$$

Nous définissons la fonction de coût J qui reflète l'objectif de contrôle

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (22)$$

Minimiser les erreurs entre la sortie prévue et le signal de consigne

Notre objectif est uniquement de rendre l'erreur $(R_s - Y)^T (R_s - Y)$ aussi petite que possible

$$\bar{R} = r_w I_{N_c \times N_c}$$

où r_w est utilisé comme paramètre de réglage pour les performances souhaitées en boucle fermée

18

Pour trouver le ΔU optimal qui minimisera J ,

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (23)$$

remplacer

$$Y = Fx(k) + \Phi \Delta U$$

On obtient

$$J = (R_s - Fx(k))^T (R_s - Fx(k)) - 2\Delta U^T \Phi^T (R_s - Fx(k)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U \quad (24)$$

La première dérivée de la fonction de coût J est :

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T (R_s - Fx(k)) + 2(\Phi^T \Phi + \bar{R}) \Delta U \quad (25)$$

La condition nécessaire du minimum J est obtenue si

$$\frac{\partial J}{\partial \Delta U} = 0 \quad (26)$$

$$-2\Phi^T (R_s - Fx(k)) + 2(\Phi^T \Phi + \bar{R}) \Delta U = 0 \quad (27)$$

à partir de laquelle nous trouvons la solution optimale pour le signal de contrôle

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k)) \quad (28)$$

19

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k)) \quad (29)$$

La matrice $(\Phi^T \Phi + \bar{R})^{-1}$ est la matrice hessienne de l'optimisation

$$R_s = \overbrace{[1 \ 1 \ \dots 1]^T}^{N_p} r(k) = \bar{R}_s r(k)$$

où

$$\bar{R}_s = \overbrace{[1 \ 1 \ \dots 1]^T}^{N_p}$$

La solution optimale du signal de commande est liée au signal de consigne $r(k)$

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (\bar{R}_s r(k) - Fx(k)) \quad (30)$$

où

$$\bar{R} = r_w I_{N_c \times N_c} \quad \bar{R}_s = \overbrace{[1 \ 1 \ \dots 1]^T}^{N_c}$$

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}; \Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix}$$

20

Exemple 1

Considérons un modèle à temps discret sous la forme suivante:

$$x_m(k+1) = A_m x_m(k) + B_m u(k)$$

$$y(k) = C_m x_m(k)$$

où les matrices du système sont

$$A_m = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$B_m = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

$$C_m = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Trouver les matrices du triplets (A, B, C) correspondant au modèle augmenté (14)

Solution:

$$x(k+1) = Ax(k) + B\Delta u(k)$$

$$y(k) = Cx(k)$$

$$A = \begin{bmatrix} A_m & 0_m^T \\ C_m A_m & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1 \\ 0.5 \end{bmatrix} \quad C = \begin{bmatrix} 0_m & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

21

Exemple 2

Supposons qu'un système du premier ordre soit décrit par l'équation d'état:

$$x_m(k+1) = ax_m(k) + bu(k)$$

$$y(k) = C_m x_m(k)$$

où $a = 0.8$ et $b = 0.1$ sont des scalaires.

- Trouvez le modèle d'espace d'état augmenté (14).
- En supposant un horizon de prédiction $N_p = 10$ et un horizon de contrôle $N_c = 4$, calculez les composantes qui **forment la prédiction de la sortie futur Y**
- Calculez les grandeurs $\Phi^T \Phi$, $\Phi^T F$ et $\Phi^T \bar{R}_i$.
- En supposant qu'à un instant k_i ($k_i = 10$ pour cet exemple), $r(k_i) = 1$ et le vecteur d'état $x(k_i) = [0.1 \ 0.2]^T$, trouvez la solution optimale ΔU dans les cas où $rw = 0$ et $rw = 10$.

22

Solution: (a) L'équation d'état-espace augmentée est

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} a & 0 \\ a & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} b \\ b \end{bmatrix} \Delta u(k)$$

$$y(k) = [0 \ 1] \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}.$$

(b) Basées sur (21), les matrices F et Φ prennent les formes suivantes:

$$Y = F x(k) + \Phi \Delta U$$

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}; \Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix}$$

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \\ CA^6 \\ CA^7 \\ CA^8 \\ CA^9 \\ CA^{10} \end{bmatrix}; \Phi = \begin{bmatrix} CB & 0 & 0 & 0 \\ CAB & CB & 0 & 0 \\ CA^2B & CAB & CB & 0 \\ CA^3B & CA^2B & CAB & CB \\ CA^4B & CA^3B & CA^2B & CAB \\ CA^5B & CA^4B & CA^3B & CA^2B \\ CA^6B & CA^5B & CA^4B & CA^3B \\ CA^7B & CA^6B & CA^5B & CA^4B \\ CA^8B & CA^7B & CA^6B & CA^5B \\ CA^9B & CA^8B & CA^7B & CA^6B \end{bmatrix}$$

$N_p = 10$ et $N_c = 4$

23

(c) Avec les paramètres du système $a = 0.8$ et $b = 0.1$, $N_p = 10$ et $N_c = 4$, nous calculons les quantités

$$\bar{R}_s = \overbrace{[1 \ 1 \ \dots 1]^T}^{N_p}$$

$$\Phi^T \Phi = \begin{bmatrix} 1.1541 & 1.0407 & 0.9116 & 0.7726 \\ 1.0407 & 0.9549 & 0.8475 & 0.7259 \\ 0.9116 & 0.8475 & 0.7675 & 0.6674 \\ 0.7726 & 0.7259 & 0.6674 & 0.5943 \end{bmatrix}$$

$$\Phi^T F = \begin{bmatrix} 9.2325 & 3.2147 \\ 8.3259 & 2.7684 \\ 7.2927 & 2.3355 \\ 6.1811 & 1.9194 \end{bmatrix}; \Phi^T \bar{R}_s = \begin{bmatrix} 3.2147 \\ 2.7684 \\ 2.3355 \\ 1.9194 \end{bmatrix}.$$

(d) Quand $r_w = 0$. Ensuite, le ΔU optimal est trouvé par le calcul

$$\bar{R} = r_w I_{N_c \times N_c} \longrightarrow \bar{R} = 0$$

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (\bar{R}_s r(k) - Fx(k))$$

$$\Delta U = (\Phi^T \Phi)^{-1} (\Phi^T \bar{R}_s - \Phi^T Fx(k_i)) = [7.2 \ -6.4 \ 0 \ 0]^T.$$

$$x(k_i) = [0.1 \ 0.2]^T$$

$$r(k_i) = 1$$

$$\bar{R}_s = \overbrace{[1 \ 1 \ \dots 1]^T}^{N_p}$$

$$R_s = \overbrace{[1 \ 1 \ \dots 1]^T}^{N_p} r(k) = \bar{R}_s r(k)$$

$$R_s = [1 \ 1 \ \dots 1]^T$$

24

3. Système de contrôle en boucle fermée

le vecteur de paramètres optimal ΔU est obtenu en utilisant

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \bar{R}_s r(k) - \Phi^T F x(k)) \quad (31)$$

ΔU à l'instant k est donc :

$$\Delta u(k) = K_y r(k) - K_{mpc} x(k) \quad (32)$$

où K_y est le premier élément de $(\Phi^T \Phi + \bar{R})^{-1} \Phi^T \bar{R}_s$

K_{mpc} est la première ligne de $(\Phi^T \Phi + \bar{R})^{-1} \Phi^T F$

alors

$$K_y = [1 \ 0 \ \dots \ 0] (\Phi^T \Phi + \bar{R})^{-1} \Phi^T \bar{R}_s \quad K_{mpc} = [1 \ 0 \ \dots \ 0] (\Phi^T \Phi + \bar{R})^{-1} \Phi^T F \quad (33)$$

Donc le modèle augmenté est

$$x(k+1) = A x(k) + B \Delta u(k) \quad (34)$$

L'équation en boucle fermée

$$x(k+1) = A x(k) + B(K_y r(k) - K_{mpc} x(k)) \quad (35)$$

$$x(k+1) = (A - B K_{mpc}) x(k) + B K_y r(k) \quad (36)$$

25

3. Système de contrôle en boucle fermée

Le diagramme montre la structure de retour d'état pour la commande de prédiction de modèle discret (DMPC) avec action intégrale dans laquelle le bloc $1/(1-q^{-1})$ désigne l'intégrateur à temps discret

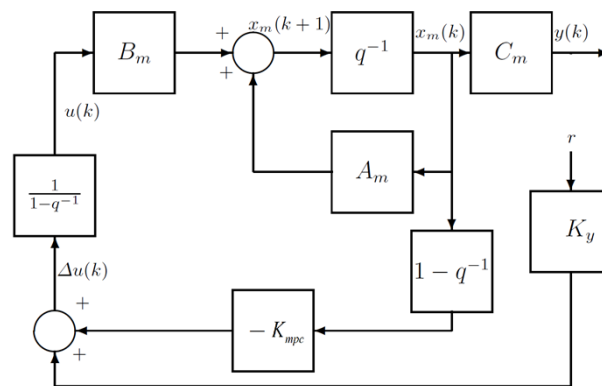


Schéma fonctionnel du système de contrôle prédictif à temps discret

26

EXEMPLE 3

Cet exemple examinera les matrices de gain de rétroaction en boucle fermée générées à partir de l'exemple 2 et les valeurs propres du système en boucle fermée avec un poids $rw = 0$

Solution: Lorsque le poids $rw = 0$, nous avons

de (33) et (36)

$$K_y = [1 \ 0 \ 0 \ 0] (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}) = 10$$

$$K_{mpc} = [1 \ 0 \ 0 \ 0] (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T F) = [8 \ 10]$$

les valeurs propres de la matrice en boucle fermée $\bar{A} - BK_{mpc}$

$$A = \begin{bmatrix} 0.8 & 0 \\ 0.8 & 1 \end{bmatrix}; B = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

$$\lambda_1 = -6.409 \times 10^{-7} \text{ and } \lambda_2 = 6.409 \times 10^{-7}$$

27

1.3.2. Prédiction pour les modèles multi-sortie multi-entrées (MIMO)

Modèle d'état

1. Trouver le modèle augmenté, qui sera utilisé dans la conception du contrôle prédictif.

système à entrées multiples et à sorties multiples, décrit par

$$x_m(k+1) = A_m x_m(k) + B_m \Delta u(k) \quad (37)$$

$$y(k) = C_m x_m(k) \quad (38)$$

A_m , B_m et C_m ont respectivement la dimension $n1 \times n1$, $n1 \times m$ and $q \times n1$.

Avec la même méthode

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \bar{R}_s r(k) - \Phi^T F x(k)) \quad (39)$$

ΔU à l'instant k est

$$\Delta u(k) = K_y r(k) - K_{mpc} x(k) \quad (40)$$

où

$$K_y = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T \bar{R}_s \quad K_{mpc} = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T F \quad (41)$$

28

1.4. Estimation prédictive de l'état MPC linéaire

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - F\hat{x}(k)) \quad (42)$$

Dans la mise en œuvre du contrôle prédictif, un observateur est utilisé pour les cas où la variable d'état $x(k)$ à l'instant k n'est pas mesurable

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - F\hat{x}(k)) \quad (43)$$

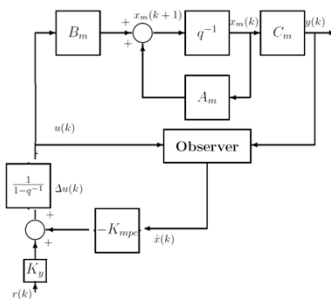
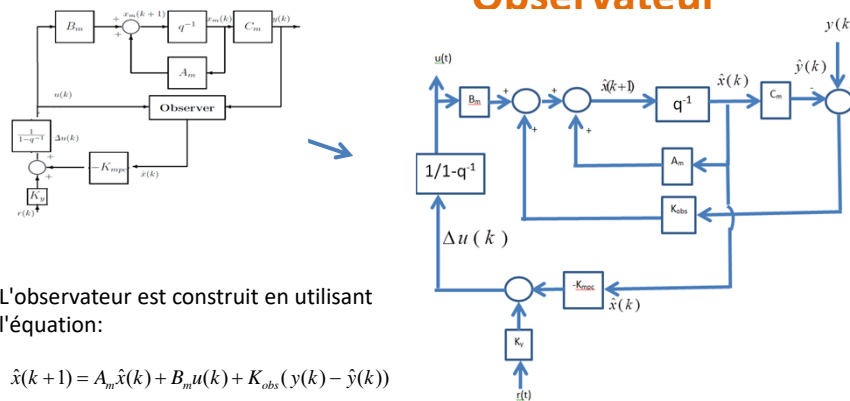


Schéma de principe de MPC avec observateur

29

Observateur



L'observateur est construit en utilisant l'équation:

$$\hat{x}(k+1) = A_m \hat{x}(k) + B_m u(k) + K_{obs} (y(k) - \hat{y}(k))$$

$$\hat{y}(k) = C_m \hat{x}(k)$$

Nous utilisons la commande MATLAB 'place' pour produire la matrice de gain d'observateur K_{obs} .

Si Pôle = observateur pôles désirés

$$K_{obs} = \text{place}(A_m', C_m', \text{Pole})$$

30

Exemple 4

Le modèle augmenté pour un système (voir exemple 1) est donné par

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) \\ v(k) &= Cx(k) \end{aligned}$$

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}; \quad B = \begin{bmatrix} 0.5 \\ 1 \\ 0.5 \end{bmatrix}; \quad C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

. L'observateur est conçu en utilisant la méthode de placement de pôles où les pôles d'observateurs en boucle fermée sont 0.01, 0.0105, 0.011, correspondant à une vitesse de réponse dynamique rapide de l'observateur.

Calculez K_{obs} par la commande matlab

Solution:

```
Pole=[0.01 0.0105 0.011];
K_obs=place(A',C',Pole);
```

31

1.5. MPC en continu

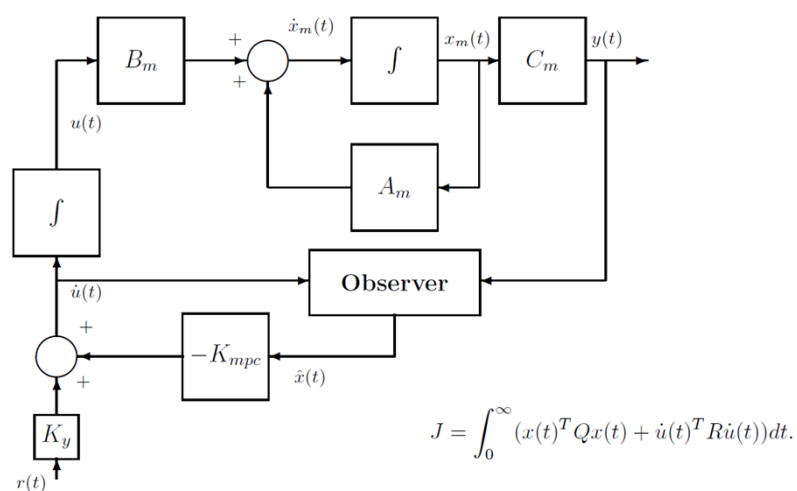


Schéma de principe CMPC avec un observateur

32

1.6. LQR et MPC linéaire

Pour simplifier, Supposons que les références sont mises à zéro.

LQR

$$\min_{\mathbf{x}_0, \mathbf{u}_0} \sum_{k=1}^{\infty} \mathbf{x}(k)^T \mathbf{Q} \mathbf{x}(k) + \mathbf{u}(k)^T \mathbf{R} \mathbf{u}(k)$$

En respectant :

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \quad k = 0, 1, \dots, \infty$$

Avantages :

- Solution linéaire
- Faible temps de calcul

Désavantages :

- Les contraintes ne sont pas prises en compte
- Pas de capacité prédictive

MPC linéaire

$$\min_{\mathbf{x}_N, \mathbf{u}_N} \sum_{k=1}^N \mathbf{x}(k)^T \mathbf{Q} \mathbf{x}(k) + \sum_{k=0}^{N-1} \mathbf{u}(k)^T \mathbf{R} \mathbf{u}(k)$$

En respectant :

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \quad k = 0, 1, \dots, N-1,$$

$$|\mathbf{x}(k)| \leq \mathbf{x}_{\max}, \quad k = 1, 2, \dots, N,$$

$$|\mathbf{u}(k)| \leq \mathbf{u}_{\max}, \quad k = 0, 1, \dots, N-1,$$

Avantages :

- Prend en compte les contraintes
- Comportement prédictif

Désavantages :

- Temps de calcul important

Si les contraintes ne sont pas prises en compte, alors le MPC et le LQR donnent la même solution

33

1.6.1. Contraintes

- Contraintes dures
- Contraintes douces

$$u_{\min}(t) \leq u(t) \leq u_{\max}(t),$$

$$\Delta u_{\min}(t) \leq \Delta u(t) \leq \Delta u_{\max}(t)$$

$$y_{\min}(t) \leq y(t) \leq y_{\max}(t)$$

$$x_{\min}(t) \leq x(t) \leq x_{\max}(t)$$

34

1.6.1. Contraintes

- Contraintes dures

$$\begin{aligned} u_{\min}(t) &\leq u(t) \leq u_{\max}(t) , \\ \Delta u_{\min}(t) &\leq \Delta u(t) \leq \Delta u_{\max}(t) \end{aligned}$$

$$\begin{aligned} y_{\min}(t) &\leq y(t) \leq y_{\max}(t) \\ x_{\min}(t) &\leq x(t) \leq x_{\max}(t) \end{aligned}$$

35

1.6.1. Contraintes

- Contraintes douces

$$\begin{aligned} u_{\min}(t) &\leq u(t) \leq u_{\max}(t) , \\ \Delta u_{\min}(t) &\leq \Delta u(t) \leq \Delta u_{\max}(t) \end{aligned}$$

$$\begin{aligned} y_{\min}(t) &\leq y(t) \leq y_{\max}(t) \\ x_{\min}(t) &\leq x(t) \leq x_{\max}(t) \end{aligned}$$

36

1.6.2. Problème d'optimisation LQR - dans MATLAB

L'équation algébrique de Riccati

$$KA + A^T K - KBR^{-1}B^T K + Q = 0$$

LQR dans MATLAB

$$[K] = \text{LQR}(A, B, Q, R)$$

Contrôleur par retour d'état

$$QA^T + AQ + (BY)^T + (BY) \leq 0$$

$$u = -R^{-1}B^T Kx$$

Exemple

$$\begin{bmatrix} \dot{x} \\ \dot{\dot{x}} \\ \dot{\phi} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.1818 & 2.6727 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.4545 & 31.1818 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.8182 \\ 0 \\ 4.5455 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad R = 1;$$

Solution

$$K = \text{lqr}(A, B, Q, R) \longrightarrow u = -R^{-1}B^T Kx$$

37

1.6.3. Problème d'optimisation MPC - dans MATLAB

Le problème d'optimisation LCQP suivant,

$$\begin{aligned} \min_{\mathbf{x}_N, \mathbf{u}_N} & \sum_{i=1}^N (\mathbf{x}(k+i) - \mathbf{x}_{\text{ref}}(k+i))^T Q (\mathbf{x}(k+i) - \mathbf{x}_{\text{ref}}(k+i)) + \sum_{i=0}^{N-1} (\mathbf{u}(k+i) - \mathbf{u}_{\text{ref}}(k+i))^T R (\mathbf{u}(k+i) - \mathbf{u}_{\text{ref}}(k+i)) \\ \text{subject to} & \quad \mathbf{x}(k+1+i) = \mathbf{A}\mathbf{x}(k+i) + \mathbf{B}\mathbf{u}(k+i), \quad i = 0, 1, \dots, N-1, \\ & \quad |\mathbf{u}(k+i)| \leq \mathbf{u}_{\text{max}}, \quad i = 0, 1, \dots, N-1, \\ & \quad |\mathbf{x}(k+i)| \leq \mathbf{x}_{\text{max}}, \quad i = 1, 2, \dots, N, \end{aligned} \quad (44)$$

avec

$$\mathbf{x}_N \in \mathfrak{R}^{n_x \cdot N} = [\mathbf{x}(k+1); \mathbf{x}(k+2); \dots; \mathbf{x}(k+N)], \quad \mathbf{u}_N \in \mathfrak{R}^{n_u \cdot N} = [\mathbf{u}(k); \mathbf{u}(k+1); \dots; \mathbf{u}(k+N-1)]$$

Est résolu par la fonction Matlab :

$$[x, fval] = \text{quadprog}(H, f, A, b, Aeq, Beq, l, u)$$

38

(44) peut être réécrit comme un problème LCQP (Linearly Constrained Quadratic Program) de la manière suivante

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^\top Hx + f^\top x \\ \text{sujet à} \quad & Ax \preceq b \\ & A_{\text{eq}}x = b_{\text{eq}} \\ & l \preceq x \preceq u \end{aligned}$$

$$[x, \text{fval}] = \text{quadprog}(H, f, A, b, A_{\text{eq}}, B_{\text{eq}}, l, u)$$

39

Exemple 5

Résoudre un problème de programmation quadratique simple: trouver les valeurs de x qui minimisent

$$\begin{aligned} \min_{x,y} \quad & \frac{1}{2}x^2 + 3x + 4y \\ \text{sujet à} \quad & x, y \geq 0 \\ & x + 3y \geq 15 \\ & 2x + 5y \leq 100 \\ & 3x + 4y \leq 80 \end{aligned}$$

Solution

$$\begin{aligned} \min_{x,y} \quad & \frac{1}{2} \begin{bmatrix} x \\ y \end{bmatrix}^\top \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix}^\top \begin{bmatrix} x \\ y \end{bmatrix} \\ & \begin{bmatrix} -1 & -3 \\ 2 & 5 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \preceq \begin{bmatrix} -15 \\ 100 \\ 80 \end{bmatrix} \\ & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \preceq \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned}$$



$$\begin{aligned} H &= \text{diag}([1; 0]); \\ f &= [3; 4]; \\ A &= [-1 \ -3; 2 \ 5; 3 \ 4]; \\ b &= [-15; 100; 80]; \\ l &= \text{zeros}(2,1); \end{aligned}$$

$$[x, \text{fval}] = \text{quadprog}(H, f, A, b, [], [], l, []);$$

40

2. MPC basé sur les inégalités matricielles linéaires (LMI)

41

2.1. Inégalités matricielles linéaires dans le contrôle (LMI)

Forme Standard

$$F(x) := F_0 + x_1 F_1 + \cdots + x_n F_n > 0$$

où

$$x := \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, F_i \in \mathbb{S}^m \text{ } m \times m \text{ symmetric matrix}$$

Exemple

$$\begin{bmatrix} 1 & x \\ x & 1 \end{bmatrix} > 0 \Leftrightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + x \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} > 0.$$

42

2.1. Inégalités matricielles linéaires dans le contrôle (LMI)

Definiteness

- Semidefinite positive

$$F \geq 0 \iff \text{si tous les principaux mineurs sont } \geq 0$$

- Négatif défini

$$F < 0 \iff \text{si tout mineur principal impair est } < 0 \text{ et même premier mineur principal est } > 0$$

- Négatif Semi-défini

$$F \leq 0 \iff \text{si chaque mineur principal impair est } \leq 0 \text{ et même principal mineur est } \geq 0$$

$$F > 0 \iff -F < 0$$

$$F \geq 0 \iff -F \leq 0$$

43

2.1. Inégalités matricielles linéaires dans le contrôle (LMI)

Exemple

$$y > 0, y - x^2 > 0, \iff \begin{bmatrix} y & x \\ x & 1 \end{bmatrix} > 0$$

Nous pouvons l'écrire sous forme standard

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} + y \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + x \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} > 0$$

44

2.1. Inégalités matricielles linéaires dans le contrôle (LMI)

Schur Complément

$$\text{Si } M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

Schur Complément du Bloc D de la matrice M

$$M / D = A - BD^{-1}C$$

Schur Complément du Bloc a de la matrice M

$$M / A = D - CA^{-1}B$$

Exemple

$$\text{Si } A := \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}.$$

$$S_{\text{ch}}(A_{11}) := A_{22} - A_{21}A_{11}^{-1}A_{12}$$

$$S_{\text{ch}}(A_{22}) := A_{11} - A_{12}A_{22}^{-1}A_{21}$$

45

Les LMI sont résolues par YALMIP

YALMIP: Une boîte à outils pour la modélisation et l'optimisation dans MATLAB

Pour l'implémentation de YALMIP dans le Matlab nous avons quatre étapes

1. Déclarer les variables

[sdpvar](#) est utilisé pour définir les variables de décision symboliques de YALMIP.

Semi-Definite Programming (sdp)

`Q = sdpvar(n,n);`

`Y = sdpvar(m,n);`

2. Définir des contraintes

`L1=Q > 0;`

`L2 = QAT + AQ + (BY)T + (BY) ≤ 0`

`L = L1 + L2`

3. Résolution du problème d'optimisation

`olvesdp(L, objective);` ou `optimize(L, objective);`

4. Calculer les variables

`Q=double(Q);`

`Y=double(Y);`

46

2.1. Inégalités matricielles linéaires dans le contrôle (LMI) dans Matlab

1. Systèmes linéaires et stabilité de Lyapunov

Système en boucle ouverte

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (45)$$

Supposons que le contrôleur optimal est donné par

$$u(t) = Kx(t) \quad (46)$$

Boucle fermée (45) et (46)

$$\dot{x}(t) = (A + BK)x(t) \quad (47)$$

2. Trouver K à l'aide de LMI

Le système est stable si, il existe une fonction de Lyapunov quadratique

$$V(x(t)) = x^T(t)Px(t) \quad (48)$$

Tel que

$$\dot{V}(x(t)) \leq 0 \quad (49)$$

$$P \geq 0 \quad \text{Est une matrice définie positive} \quad (50)$$

47

De (49), on obtient la dérivée

$$\frac{dV(x(t))}{dx(t)} = \dot{x}^T(t)Px(t) + x^T(t)P\dot{x}(t) \quad (51)$$

De (47) et (51)

$$\frac{dV(x(t))}{dx(t)} = [(A + BK)x(t)]^T Px(t) + x^T(t)P[(A + BK)x(t)] \quad (52)$$

$$\frac{dV(x(t))}{dx(t)} = x^T(t)(A + BK)^T Px(t) + x^T(t)P(A + BK)x(t) \quad (xz)^T = z^T x^T$$

$$\frac{dV(x(t))}{dx(t)} = x^T(t)[(A + BK)^T P + P(A + BK)]x(t) \quad P^T = P$$

Sachant que $\dot{V}(x(t)) \leq 0$

alors $(A + BK)^T P + P(A + BK) \leq 0 \quad (53)$

48

$$(A + BK)^T P + P(A + BK) \leq 0 \quad (53)$$

$$A^T P + (BK)^T P + PA + P(BK) \leq 0 \quad (54)$$

$$A^T P + (BK)^T P + PA + P(BK) \leq 0 \quad (55)$$



(55) Est une inégalité matricielle non linéaire (NLMI)

Pour convertir cette NLMI en LMI

Multiplier (55) (gauche et droite) par P^{-1}

$$P^{-1} A^T P P^{-1} + P^{-1} (BK)^T P P^{-1} + P^{-1} P A P^{-1} + P^{-1} P (BK) P^{-1} \leq 0 \quad (56)$$

$$P^{-1} A^T + P^{-1} (BK)^T + A P^{-1} + (BK) P^{-1} \leq 0 \quad (57)$$

Posons : $Q = P^{-1}$

$$Q A^T + Q (BK)^T + A Q + (BK) Q \leq 0 \quad (58)$$

$$Q A^T + A Q + (BK Q)^T + (BK Q) \leq 0 \quad (59)$$

Posons : $Y = K Q$

On obtient la LMI $Q A^T + A Q + (B Y)^T + (B Y) \leq 0 \quad (60)$ 49

2.1. Inégalités matricielles linéaires dans le contrôle (LMI) dans Matlab

L'LMI $Q A^T + A Q + (B Y)^T + (B Y) \leq 0 \quad (60)$

$$Q \geq 0 \quad (61)$$

Trouver des gains de contrôleur basés sur l'IMI en utilisant MATLAB

```
>> Q = sdpvar(n,n);
>> Y = sdpvar(1,n);
>> L1 = Q>=0;
>> L2=A'*Q+Q*A+(B*Y)' +B*Y <= 0;
>> L=L1+L2;
>> optimize(F);
>> Q = double(Q);
>> Y = double(Y);
>> K = Y*inv(Q);
```

[sdpvar](#) est utilisé pour définir les variables de décision symboliques de YALMIP.

Semi-Definite Programming (sdp)

$$Y = K Q$$



$$K = Y Q^{-1}$$

Exemple 6

Pendule inversé: méthodes par espace d'état pour la conception du contrôleur

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.1818 & 2.6727 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.4545 & 31.1818 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.8182 \\ 0 \\ 4.5455 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

Trouver des gains du contrôleur par LMI en utilisant MATLAB

```
>> A = [0 1 0 0; 0 -0.1818 2.6727 0; 0 0 0 1; 0 -0.4545 31.1818 0]
>> B = [0; 1.8182; 0; 4.5455]
>> Q = sdpvar(4,4);
>> Y = sdpvar(1,4);
>> L1 = Q>=0;
>> L2=A'*Q+Q*A+(B*Y)'+B*Y <= 0;
>> L=L1+L2;
>> optimize(F);
>> Q = double(Q);
>> Y = double(Y);
>> K = Y*inv(Q);
```

51

2.3. Contrôle prédictif de modèle robuste (RMPC) utilisant des IMI

Formulation

x0 étant donné, résoudre

$$\left[\begin{array}{ll} \min_K & J_\infty(\mathbf{x}_\infty, \mathbf{u}_\infty) = \sum_{k=0}^{\infty} (x_k^T Q x_k + u_k^T R u_k) \\ \text{s.t.} & u_k \in \mathcal{U} \quad k = 0, \dots, \infty \\ & x_k \in \mathcal{X} \quad k = 1, \dots, \infty \\ & x_{k+1} = A x_k + B u_k \quad k = 0, \dots, \infty \\ & u_k = K x_k \quad k = 0, \dots, \infty \end{array} \right. \quad (62)$$

2.3. Comande prédictive robuste (RMPC) utilisant des LMI

L'objectif

Minimize La fonction objectif **utilisant des LMI et Lyapunov**

$$J_{\infty} = \sum_{i=0}^{\infty} \{X(k)^T \bar{Q} X(k) + u(k)^T R u(k)\} \quad (63)$$

sujet à

$$u^T(k) u(k) < X^T(k) N^{-1} X(k) \quad (64)$$

$$N = N^T > 0$$

Supposons que le MPC Command dynamique

$$x_c(k+1) = A_c x_c(k) + B_c y(k)$$

$$u(k) = C_c x_c(k)$$

Trouver Paramètres MPC

$$A_c, B_c, C_c$$

Si, la condition suivante est satisfaite

$$V(X(k+1)) - V(X(k)) = -\{X(k)^T \bar{Q} X(k) + u(k)^T R u(k)\} \leq \gamma \quad (65)$$

Formulation du problème

Système en boucle ouverte

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \quad (66)$$

Supposons que le MPC Command dynamique

$$\begin{aligned} x_c(k+1) &= A_c x_c(k) + B_c y(k) \\ u(k) &= C_c x_c(k) \end{aligned} \quad (67)$$

L'objectif

Trouver Paramètres MPC

$$A_c, B_c, C_c$$

Boucle fermée (66) et (67)

$$\begin{aligned} X(k+1) &= \bar{A} X(k) + \bar{B} u(k) \\ u(k) &= K X(k) \end{aligned} \quad (68)$$

$$y(k) = \bar{C} X(k)$$

où

$$\bar{A} = \begin{bmatrix} A & 0 \\ B_c C & A_c \end{bmatrix}, \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, X(k) = \begin{bmatrix} x(k) \\ x_c(k) \end{bmatrix}, K = \begin{bmatrix} 0 & C_c \end{bmatrix}, \bar{C} = \begin{bmatrix} C & 0 \end{bmatrix} \quad (69)$$

Boucle fermée

$$X(k+1) = (\bar{A} + \bar{B} K) X(k) \quad (70)$$

54

Formulation du problème

La fonction objectif

$$J_{\infty} = \sum_{i=0}^{\infty} \{X(k)^T \bar{Q} X(k) + u(k)^T R u(k)\} \quad (71)$$

où

$$\bar{Q} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}$$

$$Q \geq 0, R \geq 0$$

Introduisons la fonction quadratique **Lyapunov**:

$$V(X(k)) = X^T(k) P X(k), \quad P > 0 \quad (72)$$

Si

$$X(\infty) = 0, \quad V(X(\infty)) = 0 \quad (73)$$

Alors, la condition suivante est satisfaite

$$V(X(k+1)) - V(X(k)) = -\{X(k)^T \bar{Q} X(k) + u(k)^T R u(k)\} \leq \gamma \quad (74)$$

de

$$(72) \quad V(X(k+1)) = X^T(k+1) P X(k+1), \quad (75)$$

55

Formulation du problème

de (70) et (72)

$$\begin{aligned} ((\bar{A} + \bar{B}K)X(k))^T P (\bar{A} + \bar{B}K)X(k) - X^T(k) P X(k) \\ = -X(k)^T \bar{Q} X(k) - (KX(k))^T R KX(k) \end{aligned} \quad (76)$$

$$\begin{aligned} (KX)^T &= X^T K^T \\ (KX)^T &= X^T K^T \end{aligned}$$

$$\begin{aligned} X(k)^T (\bar{A} + \bar{B}K) P (\bar{A} + \bar{B}K) X(k) - X^T(k) P X(k) \\ = -X(k)^T \bar{Q} X(k) - X(k)^T K R K X(k) \end{aligned} \quad (77)$$

$$X(k)^T [(\bar{A} + \bar{B}K) P (\bar{A} + \bar{B}K) - P + \bar{Q} + K R K] X(k) < 0 \quad (78)$$

56

Formulation du problème

de (71) et (75), On obtient

$$(\bar{A} + \bar{B}K)^T P (A + BK) - P + \bar{Q} + K^T R K < \gamma \quad (79)$$

(72) Est une Inégalité matricielle non linéaires (NLMI)

Pour convertir cette NLMI en LMI

Posons : $P^{-1} = \begin{bmatrix} \bar{X} & U \\ U^T & \hat{X} \end{bmatrix}, \quad P = \begin{bmatrix} \bar{Y} & V \\ V^T & \hat{Y} \end{bmatrix} \quad (80)$

Schur Complément Pour (79)

$$\begin{bmatrix} P^{-1} & (\bar{A} + \bar{B}K)P^{-1} & 0 \\ * & P^{-1} - P^{-1} Q P^{-1} & P^{-1} K^T \\ * & * & R^{-1} \end{bmatrix} \quad (81)$$

57

Multiplier (81) (gauche et droite) par \bar{T}_2^T

où $\bar{T}_2 = \begin{bmatrix} T & 0 & 0 \\ 0 & T & I \end{bmatrix}$

On obtient

$$\begin{bmatrix} T^T P^{-1} T & \psi & 0 \\ * & \lambda & T^T P^{-1} K^T \\ * & * & R^{-1} \end{bmatrix} > 0 \quad (82)$$

où $\psi = T^T (\bar{A} + \bar{B}K) P^{-1} T \quad \psi = T^T P^{-1} T - T^T P^{-1} \bar{Q} T$

remplacer (69) et (80) à (82) Et Définissons des vecteurs

$$F = V B_c \quad L = C_c U^T \quad Z = \bar{Y} A \bar{X} + F C \bar{X} + \bar{Y} B L + V A_c U^T \quad (83)$$

On obtient la LMI

$$\begin{bmatrix} \bar{Y} & I & \eta & Z & 0 & 0 \\ * & \bar{X} & A & \Delta & 0 & 0 \\ * & * & \omega & \Omega & 0 & 0 \\ * & * & * & \bar{X} & L^T & \bar{X} Q^{0.5} \\ * & * & * & * & R^{-1} & 0 \\ * & * & * & * & * & I \end{bmatrix} > 0 \quad (84)$$

58

où $\eta = \bar{Y}A + FC$ $\Delta = A\bar{X} + BL$

$\omega = \bar{Y} - Q$ $\Omega = I - Q\bar{X}$

En respectant les contraintes suivantes sur la commande :

$$u^T(k)u(k) < X^T(k)N^{-1}X(k) \quad (85)$$

$$N = N^T > 0$$

Posons :

$$(KX)^T KX < X^T N^{-1}X \quad (86)$$

$$X^T K^T KX < X^T N^{-1}X \quad (87)$$

alors

$$K^T K < N^{-1} \quad (88)$$

Multiplier (88) (gauche et droite) par N

$$NK^T KN < N \quad (89)$$

59

$$NK^T KN < N \quad (90)$$

Posons :

$$G = KN$$

$$G^T G < N \quad (91)$$

Schur Complément Pour (91)

$$\begin{bmatrix} 1 & G \\ G^T & N \end{bmatrix} \geq 0 \quad (92)$$

de (84)et (92), On obtient

$$\begin{bmatrix} \bar{Y} & I & \eta & Z & 0 & 0 \\ * & \bar{X} & A & \Delta & 0 & 0 \\ * & * & \omega & \Omega & 0 & 0 \\ * & * & * & \bar{X} & L^T & \bar{X}Q^{0.5} \\ * & * & * & * & R^{-1} & 0 \\ * & * & * & * & * & I \end{bmatrix} > 0 \quad (93)$$

$$\begin{bmatrix} 1 & G \\ G^T & N \end{bmatrix} \geq 0 \quad (94)$$

60

Résultat principal

Le signal de commande prédictif optimal est donné par

$$\min \gamma \quad (95)$$

sujet à

$$\begin{bmatrix} \bar{Y} & I & \eta & Z & 0 & 0 \\ * & \bar{X} & A & \Delta & 0 & 0 \\ * & * & \omega & \Omega & 0 & 0 \\ * & * & * & \bar{X} & L^T & \bar{X}Q^{0.5} \\ * & * & * & * & R^{-1} & 0 \\ * & * & * & * & * & I \end{bmatrix} > 0 \quad (96)$$

$$\begin{bmatrix} 1 & G \\ G^T & N \end{bmatrix} \geq 0 \quad (97)$$

alors

Paramètres MPC

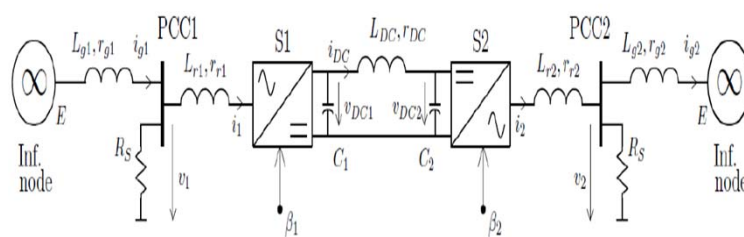
$$V = (1 - \bar{Y}\bar{X})(U^T)^{-1} \rightarrow A_c = V^{-1}Z(U^T)^{-1} \quad (98)$$

$$B_c = V^{-1}F \quad (99)$$

$$C_c = L(U^T)^{-1} \quad (100)$$

61

3. HVDC Application



62