UNIVERSITÀ DEGLI STUDI DI BERGAMO
Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Controllo Avanzato Multivariabile

LX: Model Predictive Control

CORSO DI LAUREA TRIENNALE IN INGEGNERIA INFORMATICA

TEACHERS
Prof. Antonio Ferramosca

PLACE
Università di Bergamo

Contenuti del corso

Outline

1. Introduzione
2. Formulazione base
3. Soluzione

Outline

1. **Introduzione**
2. Formulazione base
3. Soluzione

Introduzione

Nell'ultima lezione abbiamo visto che **per ottenere una legge di controllo ottimo LQ invariante**, si formula il problema su orizzonte infinito imponendo $N \to \infty$

1. La cifra di merito diventa:

$$J(x(0), u(.)) = \sum_{j=0}^{\infty} x(j)'Qx(j) + u(j)'Ru(j)$$

Con $S \to 0$, quando $N \to \infty$ (asintoticamente $x(N) \to 0$ quando $N \to \infty$).

2. La legge di controllo ottima è:

$$u(k) = -Kx(k)$$

Con $K = (R + B'PB)^{-1}B'PA$, dove P rappresenta la soluzione dell'eq. di Riccati.

Introduzione

Cosa succede se abbiamo vincoli sullo stato e sul controllo?

$$x_{min} \leq x \leq x_{max}, \quad u_{min} \leq u \leq u_{max}$$

In questo caso non esiste una formula chiusa che sia soluzione del problema di minimizzazione.

- Questo a causa dei vincoli che rendono impossibile la soluzione su orizzonte infinito.

Esiste un metodo pratico per risolvere il problema su orizzonte finito e ottenere la stessa ottimalità su orizzonte infinito?

Introduzione

Il **Model Predictive Control (MPC)** nasce proprio allo scopo di risolvere questa esigenza.

- È un metodo pratico per risolvere il problema di controllo ottimo vincolato su di un orizzonte infinito.

Il nome MPC include molti algoritmi diversi tra di loro ma che hanno caratteristiche comuni che permettono di classificarli come MPC.

È senza dubbio la tecnica di controllo avanzato più utilizzata a livello industriale.

Introduzione

Questo perchè ha una serie di caratteristiche distintive che lo rendono flessibile e adeguato per diverse applicazioni:

1. Il problema di controllo viene formulato come un problema di ottimizzazione, il che permette di includere diversi obiettivi di controllo.

2. Possono essere inclusi esplicitamente vincoli sugli stati e sui controlli.

3. Il regolatore è progettato sulla base del modello del sistema che si desidera controllare (questo modello può essere ottenuto in vari modi).

Introduzione

Ogni formulazione di MPC ha degli **ingredienti** fondamentali:

1. Un **modello** del sistema
2. Dei **vincoli** sul controllo, sugli stati e sugli output.
3. Un **funzionale di costo** che definisce l'obiettivo di controllo a voler soddisfare ad ogni istante di tempo k, definito su di un **orizzonte di predizione** di lunghezza finita N $[k, k+N]$.
4. Un **algoritmo** di ottimizzazione
5. L'applicazione del principio del ***receeding horizon***.

Come funziona

Goal: trovare la miglior sequenza di N azioni di controllo all'istante k

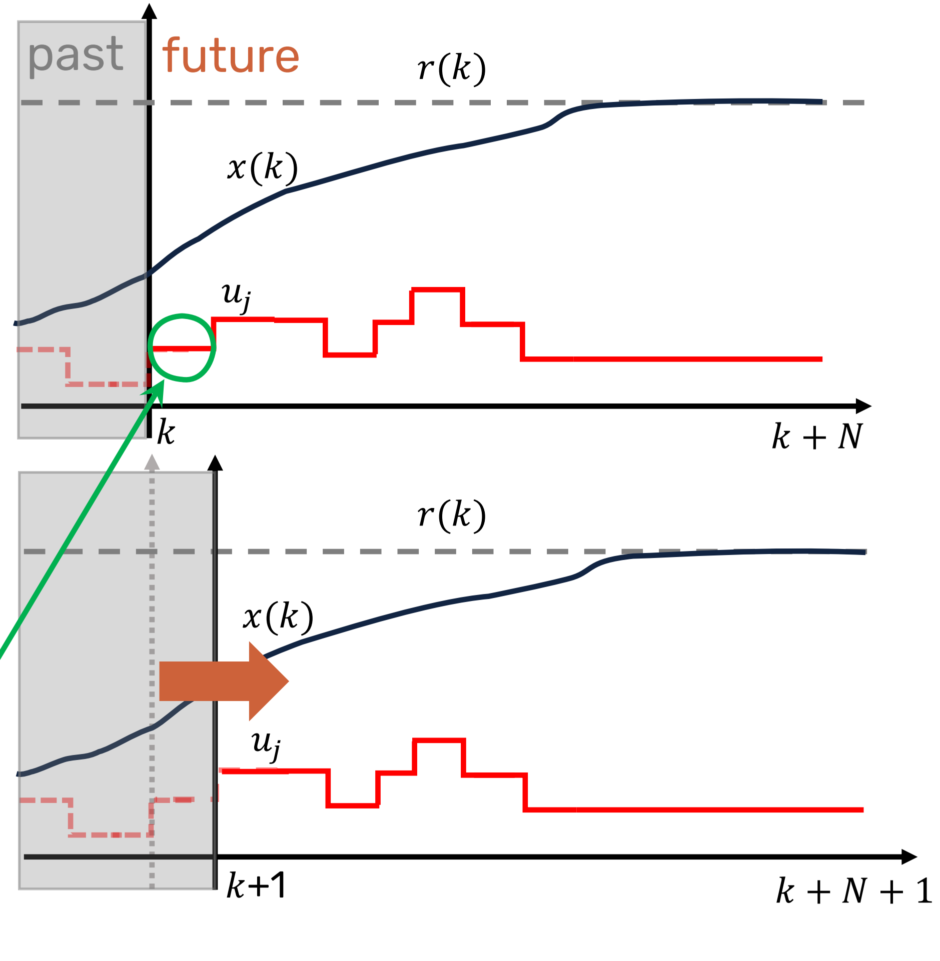
$$\min_{\boldsymbol{u}} \sum_{j=0}^{N-1} ||x_j - r(k)||_Q^2 + ||u_j - u_r(j)||_R^2$$

$s.t.$

$$x(j+1) = f(x(j), u(j)) \qquad y(j) = g(x(j))$$ *modello*

$$u_{min} \leq u_j \leq u_{max} \qquad x_{min} \leq x_j \leq x_{max}$$ *vincoli*

$$x_0 = x(k)$$ *Feedback di stato*



Ad ogni istante di tempo k:

1. Ottieni la nuova misura dello stato $x(k)$
2. Risolvi il proble di ottimizzazione e trova $\boldsymbol{u} = \{u(j), \dots, u(N-1)\}$
3. Applica solo il primo elemeno della sequeza, $u(k) = u_0^*$

Principio del receeding horizon (RH)

Cos'è dice questo principio?

« ad ogni istante di tempo k, sulla base delle informazioni disponibili (i.e. misura dello stato), si risolva il problema di ottimizzazione all'istante di tempo k per determinare la sequenza di N future azioni di controllo ottime $[u(k), u(k+1), \dots, u(k+N-1)]$, e si applichi solo la prima di esse $u(k)$. Dopodiché ad ogni istante di tempo $k+1$, si sposti la finestra di predizione un passo in avanti, e sulla base delle informazioni disponibili al tempo $k+1$, si risolva il nuovo problema di ottimizzazione lungo l'orizzonte $[k+1, k+N+1]$ per determinare la nuova sequenza di N future azioni di controllo ottime»

Esempi di MPC quotidiani

➢MPC è come giocare a scacchi…

➢… o guidare

Outline

1. Introduzione
2. **Formulazione base**
3. Soluzione

Enunciato del problema

1. Si consideri il sistema lineare a tempo discreto

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k)$$

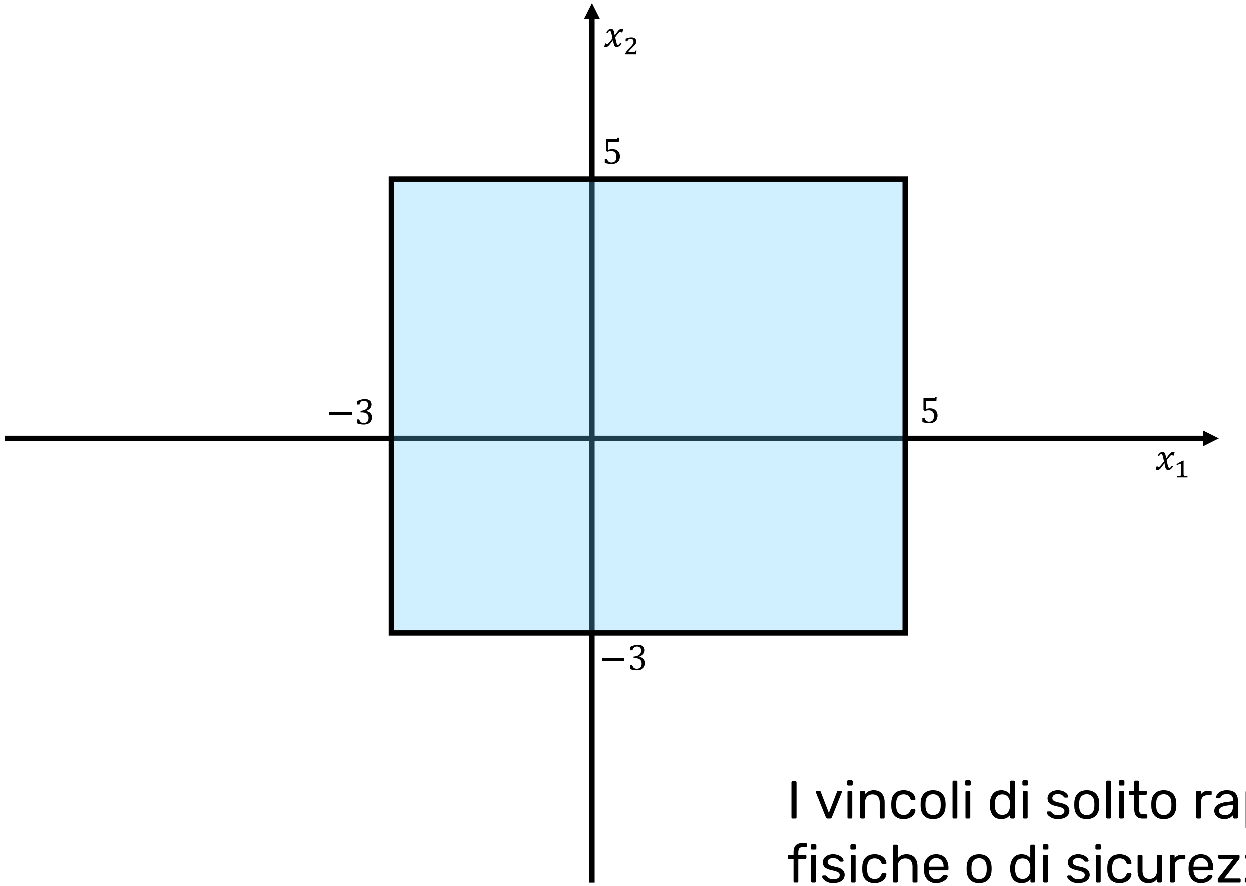
In cui $k_0 = 0$ e $x \in \mathbb{R}^n, u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$. Lo stato si suppone accessibile. Sia (A, B) ***raggiungibile***.

2. Si assuma inoltre l'esistenza di vincoli su stato e controllo $x \in \mathcal{X}, u \in \mathcal{U}$

$$x \in \mathcal{X} \Longrightarrow x_{min} \leq x \leq x_{max},$$
$$u \in \mathcal{U} \;\Longrightarrow\; u_{min} \leq u \leq u_{max}$$

Esempio vincoli

Sia ad esempio $x \in \mathcal{X}$ dato da $\begin{bmatrix} -3 \\ -3 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix}$,



I vincoli di solito rappresentano limitazioni fisiche o di sicurezza sul sistema.

Enunciato del problema

4. Si voglia risolvere il problema di regolazione dello stato del sistema all'origine.

Per farlo, si assuma una cifra di merito quadratica, ovvero

$$
\begin{aligned}
J(x(k), u(.)) &= \sum_{j=0}^{N-1} ||x(j)||_Q^2 + ||u(j)||_R^2 + ||x(N)||_S^2 \\
&= \sum_{j=0}^{N-1} x(j)'Qx(j) + u(j)'Ru(j) + x(N)'Sx(N)
\end{aligned}
$$

Con $Q = Q' \geq 0, S = S' \geq 0, R = R' > 0.$

Enunciato del problema

Il problema di ottimizzazione sarà quindi dato da

$$\min_{\boldsymbol{u}} J(x(k), u(.))$$

$$s.t.\ \ x(0) = x(k)$$ *Condizione iniziale = Feedback di stato*

$$x(j+1) = Ax(j) + Bu(j)$$

$$x(j) \in \mathcal{X}$$

$$u(j) \in \mathcal{U}$$

Il problema di ottimizzazione cosí formulato è un problema di **programmazione quadratica (QP)**.

- Funzionale di costo quadratico
- Vincoli lineari.

Enunciato del problema

5. La soluzione del problema è la sequenza di N azioni di controllo ottime

$$\boldsymbol{u} = \{u(0), u(1), \dots u(N-1)\}$$

Di esse applichiamo solo la prima al sistema: $u(k) = u(0)$

$$x(k+1) = Ax(k) + Bu(k)$$

Al tempo $k+1$ misuriamo lo stato che sarà la nuova condizione iniziale del problema di ottimizzazione.

Ad ogni iterazione il problema di ottimizzazione è lo stesso, eccezion fatta per il feedback di stato.

Outline

1. Introduzione
2. Formulazione base
3. **Soluzione**

Soluzione in anello chiuso

Supponiamo di non avere vincoli sullo stato e sugli ingressi. La soluzione del problema in anello chiuso può essere ottenuta usando la teoria del LQR

Gli elementi $u(j)$ della sequenza ottima $\boldsymbol{u}$ al tempo k saranno, per $j = 0, \dots N-1$

$$u(j) = -K(j)x(j)$$

Dove $K(j) = -((R + B'P(j+1)B)')^{-1}B'P(j+1)A'$

E $P(j)$ è la soluzione dell'equazione di Riccati

$$P(j) = (\mathrm{Q} + \mathrm{A}'P(j+1)\mathrm{A}) - A'P(j+1)B(R + B'P(j+1)B)^{-1}B'P(j+1)A$$

Con condizione iniziale $P(N) = S$, e $x(0) = x(k)$.

Osservazioni

1. La soluzione è una legge di controllo in anello chiuso, dato che ogni azione di controllo $u(j)$ dipende da uno stato $x(j)$.

2. In virtù del principio del receeding horizon, solamente il primo elemento della sequenza sarà applicato al sistema. Quindi per ogni istante di tempo k, il controllo realmente applicato (la legge di controllo del MPC) è
$$u(k) = -K(0)x(k)$$

3. A differenza del caso LQR, la legge di controllo è invariante nel tempo: una volta calcolato $K(0)$, non cambia più.

Soluzione in anello aperto

Rimaniamo sempre nel caso in cui non si hanno vincoli sullo stato e sugli ingressi.

I movimenti del sistema (libero e forzato) sono, per $j = 0, \dots N-1$

$$x(j) = A^j x(0) + \sum_{i=0}^{j-1} A^{j-i-1} B u(j)$$

Con $x(0) = x(k)$.

In virtù di questa equazione, detta equazione di Lagrange, possiamo ottenere la soluzione in anello aperto del problema MPC non vincolato.

Soluzione in anello aperto

Considerando $\boldsymbol{u} = \{u(0), u(1), \dots u(N-1)\}$ al tempo k, si definiscano:

$$\boldsymbol{u}(k) = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-2) \\ u(N-1) \end{bmatrix} \quad \boldsymbol{X}(k) = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \\ x(N) \end{bmatrix} \quad \mathcal{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N-1} \\ A^N \end{bmatrix} \quad \mathcal{B} = \begin{bmatrix} B & 0 & 0 & \dots & 0 & 0 \\ AB & B & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ A^{N-2}B & A^{N-3}B & A^{N-4}B & \dots & B & 0 \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \dots & AB & B \end{bmatrix}$$

In base a queste definizioni, le predizioni in anello aperto sono date da:

$$\boldsymbol{X}(k) = \mathcal{A}x(k) + \mathcal{B}\boldsymbol{u}(k)$$

Con $x(0) = x(k)$.

Si noti che si tratta della formula di Lagrange raccolta in forma matriciale.

Soluzione in anello aperto

Si definiscano inoltre le seguenti matrici diagonali a blocchi:

$$\mathcal{Q} = \begin{bmatrix} Q & 0 & \dots & 0 & 0 \\ 0 & Q & \dots & 0 & 0 \\ \dots & \dots & \ddots & \dots & \dots \\ 0 & 0 & \dots & Q & 0 \\ 0 & 0 & \dots & 0 & S \end{bmatrix} \qquad \mathcal{R} = \begin{bmatrix} R & 0 & \dots & 0 & 0 \\ 0 & R & \dots & 0 & 0 \\ \dots & \dots & \ddots & \dots & \dots \\ 0 & 0 & \dots & R & 0 \\ 0 & 0 & \dots & 0 & R \end{bmatrix}$$

Si noti che $\mathcal{R}$ ha N blocchi, mentre $\mathcal{Q}$ ha N+1 blocchi

Il funzionale di costo da minimizzare può quindi essere scritto come

$$J(x(k), u(.)) = \sum_{j=0}^{N-1} ||x(j)||_Q^2 + ||u(j)||_R^2 + ||x(N)||_S^2$$

$$= \boldsymbol{X}'(k)\mathcal{Q}\boldsymbol{X}(k) + \boldsymbol{u}'(k)\mathcal{R}\boldsymbol{u}(k)$$

Soluzione in anello aperto

Considerando che $\boldsymbol{X}(k) = \mathcal{A}x(k) + \mathcal{B}\boldsymbol{u}(k)$, allora:

$$J(x(k), u(.)) = \boldsymbol{X}'(k)\mathcal{Q}\boldsymbol{X}'(k) + \boldsymbol{u}'(k)\mathcal{R}\boldsymbol{u}(k)$$

$$= (\mathcal{A}x(k) + \mathcal{B}\boldsymbol{u}(k))'\mathcal{Q}(\mathcal{A}x(k) + \mathcal{B}\boldsymbol{u}(k)) \;+ \boldsymbol{u}'(k)\mathcal{R}\boldsymbol{u}(k)$$

$$= x'(k)\mathcal{A}'\mathcal{Q}\mathcal{A}x(k) + \boldsymbol{u}'(k)\mathcal{B}'\mathcal{Q}\mathcal{B}\boldsymbol{u}(k) + 2x'(k)\mathcal{A}'\mathcal{Q}\mathcal{B}\boldsymbol{u}(k) + \boldsymbol{u}'(k)\mathcal{R}\boldsymbol{u}(k)$$

$$= {\color{red}x'(k)\mathcal{A}'\mathcal{Q}\mathcal{A}x(k)} + {\color{blue}2x'(k)\mathcal{A}'\mathcal{Q}\mathcal{B}\boldsymbol{u}(k) + \boldsymbol{u}'(k)(\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})\boldsymbol{u}(k)}$$

1. *Questa è una forma quadratica della variabile $\boldsymbol{u}(k)$, definita positiva in quanto $\mathcal{R} > 0$.*
2. *Possiamo calcolarne il minimo eguagliando a zero la sua derivata rispetto a $\boldsymbol{u}(k)$.*
3. *La parte rossa dipende solo da $x(k)$ che è un parametro noto, quindi non influenza il risultato.*

Soluzione in anello aperto

Considerando che $\boldsymbol{X}(k) = \mathcal{A}x(k) + \mathcal{B}\boldsymbol{u}(k)$, allora:

$$\arg\min_{\boldsymbol{u}(k)}[J(x(k), u(.))] \text{ è } tale\ che$$

$$\frac{\partial[x'(k)\mathcal{A}'\mathcal{Q}\mathcal{A}x(k) + 2x'(k)\mathcal{A}'\mathcal{Q}\mathcal{B}\boldsymbol{u}(k) + \boldsymbol{u}'(k)(\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})\boldsymbol{u}(k)]}{\partial\boldsymbol{u}(k)} = 0$$

Da cui:

$$\boldsymbol{u}^o(k) = -(\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})^{-1}\mathcal{B}'\mathcal{Q}\mathcal{A}x(k)$$

Questa soluzione si definisce in **anello aperto** perchè si ottiene a partire dalle predizioni in anello aperto dello stato, noto $x(k)$.

Soluzione in anello aperto

Possiamo infatti scrivre la soluzione come

$$\boldsymbol{u}^o(k) = -\begin{bmatrix} \mathcal{K}(0) \\ \mathcal{K}(1) \\ \vdots \\ \mathcal{K}(N-1) \end{bmatrix} x(k) \qquad \mathcal{K} = (\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})^{-1}\mathcal{B}'\mathcal{Q}\mathcal{A} = \begin{bmatrix} \mathcal{K}(0) \\ \mathcal{K}(1) \\ \vdots \\ \mathcal{K}(N-1) \end{bmatrix}$$

Ovvero $u^o(j) = -\mathcal{K}(j)x(k)$, per $j = 0,1, \dots N-1$.

Questa soluzione si ottiene a partire da $x(k)$.

Applicando il principio del Receeding Horizon:

$$\boldsymbol{u}^{MPC}(k) = -\mathcal{K}(0)x(k)$$

Osservazioni

- Nel caso **nominale**, ovvero in assenza di disturbi o errori di modello, la soluzione in anello aperto **coincide** con quella in anello chiuso.

- In caso di disturbi/errori di modello, solamente i primi elementi delle soluzioni ottime coincidono $\mathcal{K}(0) = K(0)$.

- La soluzione in anello aperto si ottiene calcolando la predizione dei futuri stati sulla base dello stato corrente misurato $x(k)$. Quest'idea può essere generalizzata al caso di altre rappresentazioni: funzioni di trasferimento, sistemi non lineari, etc.

- La soluzione in anello aperto permette di **considerare esplicitamente** i vincoli sullo stato e sugli ingressi. In questo caso non è possibile ottenere una soluzione esplicita, ma va risolto un problema di ottimizzazione per mezzo di specifici algoritmi (**programmazione quadratica**).

Esempio

Si consideri il sistema dato da: $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, x(0) = \begin{bmatrix} -3 \\ 1 \end{bmatrix}$

Confrontiamo le soluzioni di un MPC con $\mathrm{Q} = I, R = 1, N = 5, P(N) = S = 100I$

1. Calcoliamo la soluzione in anello chiuso: $u(j) = -K(j)x(j)$

Dove $K(j) = -((R + B'P(j+1)B)')^{-1}B'P(j+1)A'$

$$P(j) = (\mathrm{Q} + \mathrm{A}'P(j+1)\mathrm{A}) - A'P(j+1)B(R + B'P(j+1)B)^{-1}B'P(j+1)A$$

$$x(j+1) = Ax(j) + Bu(j) = (A - BK(j))x(j)$$

Esempio

1. Il risultato che otteniamo è:

$$\boldsymbol{K}_{ol}^{o} = \begin{bmatrix} K(0) \\ K(1) \\ K(2) \\ K(3) \\ K(4) \end{bmatrix} = \begin{bmatrix} 0.4249 & 1.2498 \\ 0.4458 & 1.2846 \\ 0.5517 & 1.4385 \\ 0.9710 & 1.9613 \\ 0 & 0.9901 \end{bmatrix}$$

$$\boldsymbol{u}_{cl}^{o} = \begin{bmatrix} 0.0249 \\ -0.4250 \\ -0.3250 \\ -0.1748 \\ -0.0991 \end{bmatrix}$$

2. Calcoliamo la soluzione in anello aperto:

$$\boldsymbol{u}_{ol}^{o} = -\begin{bmatrix} \mathcal{K}(0) \\ \mathcal{K}(1) \\ \mathcal{K}(2) \\ \mathcal{K}(3) \\ \mathcal{K}(4) \end{bmatrix} x(k) = \begin{bmatrix} 0.4249 & 1.2498 \\ -0.1001 & 0.1248 \\ -0.1501 & -0.1252 \\ -0.0999 & -0.1248 \\ -0.0742 & -0.1235 \end{bmatrix} x(k) = \begin{bmatrix} 0.0249 \\ -0.4250 \\ -0.3250 \\ -0.1748 \\ -0.0991 \end{bmatrix}$$

Esempio

3. Calcoliamo la soluzione con l'algoritmo di quadratic programming:

$$\boldsymbol{u}_{qp}^{o} = quadprog(H, f)$$

Dove (H, f) rappresentanto rispettivamente i termini quadratico e lineare del funzionale di costo scritto come nella soluzione in anello aperto

$$J(x(k), u(.)) = x'(k)\mathcal{A}'\mathcal{Q}\mathcal{A}x(k) + 2x'(k)\mathcal{A}'\mathcal{Q}\mathcal{B}\boldsymbol{u}(k) + \boldsymbol{u}'(k)(\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})\boldsymbol{u}(k)$$

Ovvero:

$$H = (\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})$$

$$f = 2x'(k)\mathcal{A}'\mathcal{Q}\mathcal{B}$$

$$\boldsymbol{u}_{qp}^{o} = \begin{bmatrix} 0.0249 \\ -0.4250 \\ -0.3250 \\ -0.1748 \\ -0.0991 \end{bmatrix}$$

Esempio

Si noti come le 3 soluzioni siano uguali (siamo nel caso nominale)

$$\boldsymbol{u}_{cl}^{o} = \begin{bmatrix} 0.0249 \\ -0.4250 \\ -0.3250 \\ -0.1748 \\ -0.0991 \end{bmatrix} \qquad \boldsymbol{u}_{ol}^{o} = \begin{bmatrix} 0.0249 \\ -0.4250 \\ -0.3250 \\ -0.1748 \\ -0.0991 \end{bmatrix} \qquad \boldsymbol{u}_{qp}^{o} = \begin{bmatrix} 0.0249 \\ -0.4250 \\ -0.3250 \\ -0.1748 \\ -0.0991 \end{bmatrix}$$

La legge di controllo ottima del MPC (non vincolato) al tempo k è quindi:

$$\boldsymbol{u}^{MPC}(k) = -\mathcal{K}(0)x(k) = -K(0)x(k) = 0.0249$$

Esempio 2

Supponiamo ore di avere dei vincoli sullo stato e sugli ingressi.

Sia ad esempio $x \in \mathcal{X}$ dato da $\begin{bmatrix} -2 \\ -2 \end{bmatrix} \leq x \leq \begin{bmatrix} 4 \\ 4 \end{bmatrix}, -0.3 \leq u \leq 0.3$

La presenza dei vincoli non ci permette di usare soluzioni esplicite né in anello aperto, né in anello chiuso.

Dobbiamo assolutamente risolvere il problema con l'algoritmo di programmazione dinamica, imponendo i vincoli.

$$\boldsymbol{X}_{min} \leq \boldsymbol{X}(k) = \mathcal{A}x(k) + \mathcal{B}\boldsymbol{u}(k) \leq \boldsymbol{X}_{max} \qquad \boldsymbol{U}_{min} \leq \boldsymbol{u}(k) \leq \boldsymbol{U}_{max}$$

Esempio 2

Dato che la nostra variabile di ottimizzazione è $\boldsymbol{u}(k)$, dobbiamo convertire i vincoli sullo stato in vincoli su $\boldsymbol{u}(k)$.

Se prendiamo i vincoli di massimo, ad esempio:

$$\boldsymbol{X}(k) \leq \boldsymbol{X}_{max} \Longrightarrow \mathcal{A}x(k) + \mathcal{B}\boldsymbol{u}(k) \leq \boldsymbol{X}_{max}$$

Da cui banalmente:

$$\mathcal{B}\boldsymbol{u}(k) \leq \boldsymbol{X}_{max} - \mathcal{A}x(k)$$

Seguiremo un ragionamento simile per i vincoli di minimo.

Esempio 2

Inoltre, l'algoritmo quadprog di Matlab, in caso di vincoli, assume la forma

$$\boldsymbol{u}_{qp}^{o} = quadprog(H, f, A_{qp}, b_{qp})$$

Dove (A_{qp}, b_{qp}) conterranno i vincoli sulla variabile di ottimizzazione in maniera tale che

$$A_{qp}\,\boldsymbol{u}(k) \leq b_{qp}$$

Possiamo quindi riscrivere tutti i nostri vincoli per ottenere le matrici (A_{qp}, b_{qp})

$$A_{qp} = \begin{bmatrix} \mathcal{B} \\ -\mathcal{B} \\ \boldsymbol{I} \\ -\boldsymbol{I} \end{bmatrix} \qquad b_{qp} = \begin{bmatrix} \boldsymbol{X}_{max} - \mathcal{A}x(k) \\ -\boldsymbol{X}_{min} + \mathcal{A}x(k) \\ U_{max} \\ -U_{min} \end{bmatrix}$$

Esempio 2

Utilizzando dunque la seguente funzione $\boldsymbol{u}_{qp}^{o} = quadprog(H, f, A_{qp}, b_{qp})$

otteniamo $$\boldsymbol{u}_{qp}^{o} = \begin{bmatrix} 0.0249 \\ -0.4250 \\ -0.3250 \\ -0.1748 \\ -0.0991 \end{bmatrix}$$

La soluzione è identica alle precedenti. Questo perchè la soluzione ottima rispetta i vincoli

Proviamo a stringere i vincoli sugli ingressi: $-0.3 \leq u \leq 0.3$

otteniamo $$\boldsymbol{u}_{qp}^{o} = \begin{bmatrix} -0.0654 \\ -0.3000 \\ -0.3000 \\ -0.2381 \\ -0.0956 \end{bmatrix}$$

La soluzione è differente perchè non possiamo più accettare una sequenza ottima come l'anteriore a causa dei vincoli più stringenti

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione