UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Dipartimento
di Ingegneria Gestionale,
dell'Informazione e della Produzione

Controllo Avanzato Multivariabile

L10: Model Predictive Control

CORSO DI LAUREA TRIENNALE IN
INGEGNERIA INFORMATICA

TEACHERS
Prof. Antonio Ferramosca

PLACE
Università di Bergamo

Contenuti del corso

1. **Sistemi non lineari a tempo discreto**
   1.1 Equilibrio
   1.2 Stabilità
   1.3 Teorema di Lyapunov
2. **Proprietà strutturali dei sistemi lineari multivariabili**
   2.1 Controllabilità
   2.2 Osservabilità
   2.3 Poli e zeri invarianti
3. **Analisi dei sistemi multivariabili**
   3.1 Anello aperto
   3.2 Anello chiuso
4. **Controllo ottimo**
   4.1 Proprietà
   4.2 Esempi
   4.3 Controllo Lineare Quadratico LQR
   4.4 Progetto con osservatore
5. **Controllo predittivo MPC**
   5.1 Formulazione
   5.2 Proprietà
   5.3 Stabilità
   5.4 Implementazioni
6. **Esempi di applicazioni**
   6.1 Pancreas Artificiale
   6.2 Forno SMI

Outline

1. Introduzione
2. Formulazione base
3. Soluzione

Outline

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

Introduzione

Abbiamo visto che **per ottenere una legge di controllo ottimo LQ invariante**, si formula il problema su orizzonte infinito imponendo $N \rightarrow \infty$

1. La cifra di merito diventa:

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

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

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

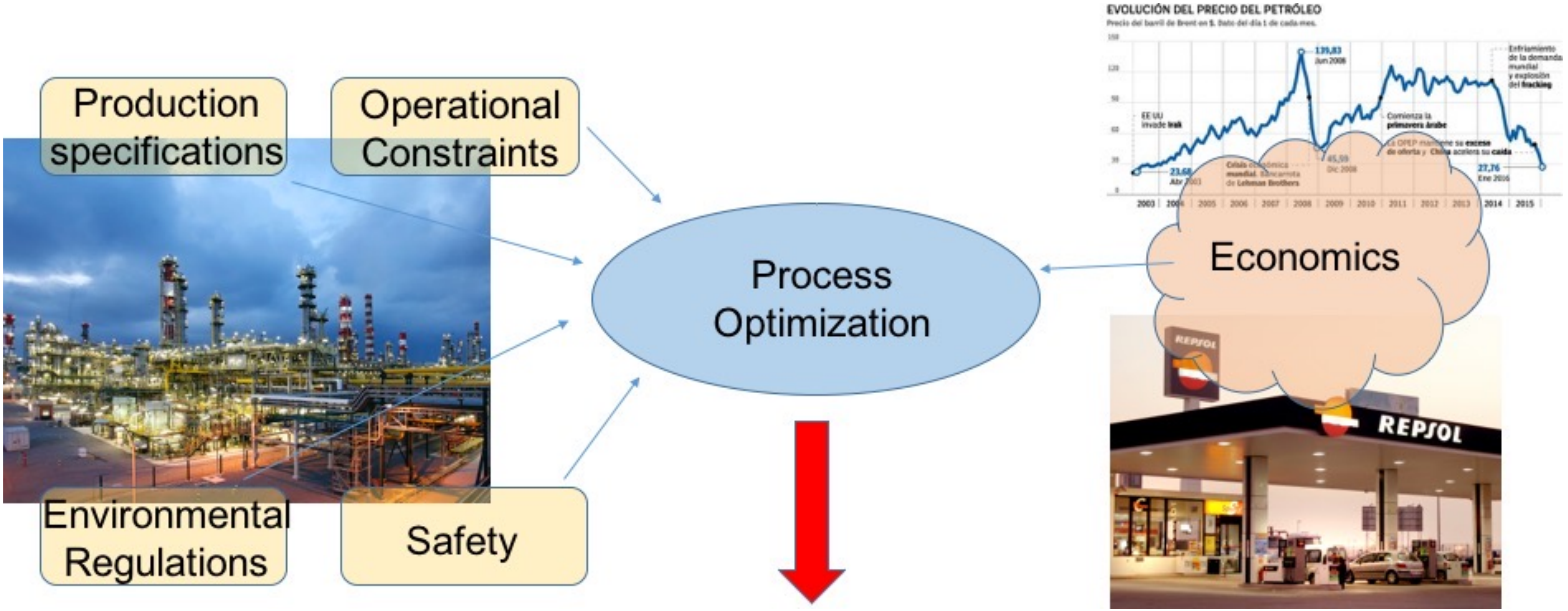
Questo perchè ha una serie di caratteristiche distintive che lo rendono flessibile e adeguato per diverse applicazioni:

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).

Motivazione



Massimizzare i guadagni

Uso nell'industria di processo a fine '90

Table 6
Summary of linear MPC applications by areas (estimates based on vendor survey; estimates do not include applications by companies who have licensed vendor technology)[a]

Area	Aspen Technology	Honeywell Hi-Spec	Adersa[b]	Invensys	SGS[c]	Total
Refining	1200	480	280	25		1985
Petrochemicals	450	80	—	20		550
Chemicals	100	20	3	21		144
Pulp and paper	18	50	—	—		68
Air & Gas	—	10	—	—		10
Utility	—	10	—	4		14
Mining/Metallurgy	8	6	7	16		37
Food Processing	—	—	41	10		51
Polymer	17	—	—	—		17
Furnaces	—	—	42	3		45
Aerospace/Defense	—	—	13	—		13
Automotive	—	—	7	—		7
Unclassified	40	40	1045	26	450	1601
Total	1833	696	1438	125	450	4542
First App.	DMC:1985 IDCOM-M:1987 OPC:1987	PCT:1984 RMPCT:1991	IDCOM:1973 HIECON:1986	1984	1985	
Largest App.	603 × 283	225 × 85	—	31 × 12	—	

[a] The numbers reflect a snapshot survey conducted in mid-1999 and should not be read as static. A recent update by one vendor showed 80% increase in the number of applications.

[b] Adersa applications through January 1, 1996 are reported here. Since there are many embedded Adersa applications, it is difficult to accurately report their number or distribution. Adersa's product literature indicates over 1000 applications of PFC alone by January 1, 1996.

[c] The number of applications of SMOC includes in-house applications by Shell, which are unclassified. Therefore, only a total number is estimated here.

Impatto attuale e futuro

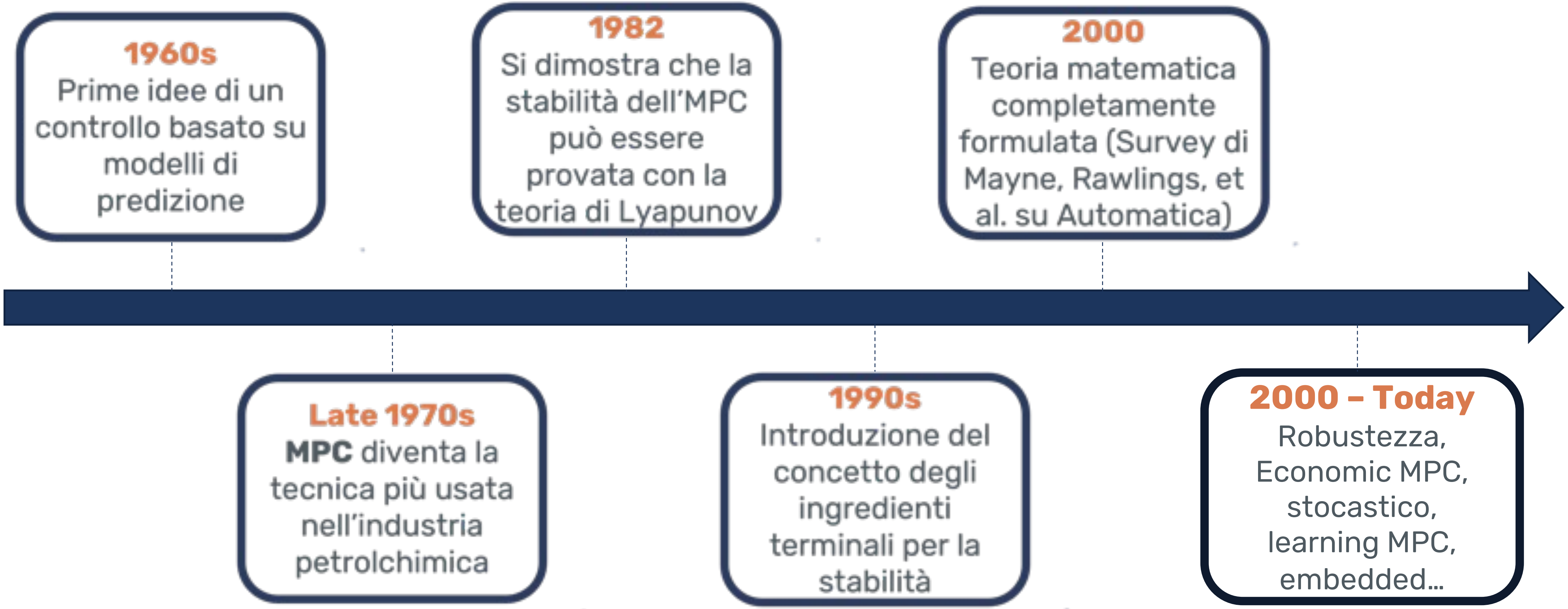
Table 2
The percentage of survey respondents indicating whether a control technology had demonstrated ("Current Impact") or was likely to demonstrate over the next five years ("Future Impact") high impact in practice.

	Current Impact	Future Impact
Control Technology	%High	%High
PID control	91%	78%
System Identification	65%	72%
Estimation and filtering	64%	63%
Model-predictive control	62%	85%
Process data analytics	51%	70%
Fault detection and identification	48%	78%
Decentralized and/or coordinated control	29%	54%
Robust control	26%	42%
Intelligent control	24%	59%
Discrete-event systems	24%	39%
Nonlinear control	21%	42%
Adaptive control	18%	44%
Repetitive control	12%	17%
Hybrid dynamical systems	11%	33%
Other advanced control technology	11%	25%
Game theory	5%	17%

(Samad et al., Annual Reviews in Control, 2020)

MPC è attualmente una tecnologia di controllo ad alto impatto, e quella con le maggiori prospettive di impatto nei prossimi 5 anni

Breve storia…



➤ Oggi largamente utilizzato in applicazioni di vario tipo: ind. automobilistica (*traction control, motori*), ind. aerospaziale, aeronautica ed UAV, smart-grids, finanza, processi biomedici…

1960s

Prime idee di un controllo basato su modelli di predizione

Anni 60...

Discrete Dynamic Optimization Applied to On-Line Optimal Control

MARSHALL D. RAFAL and WILLIAM F. STEVENS
Northwestern University, Evanston, Illinois

A general method has been developed for controlling deterministic systems described by linear or linearized dynamics. The discrete problem has been treated in detail. Step-by-step optimal controls for a quadratic performance index have been derived. The method accommodates upper and lower limits on the components of the control vector.
A small binary distillation unit was considered as a typical application of the method. The control vector was made up of feed rate, reflux ratio, and reboiler heat load. Control to a desired state and about a load upset was effected.
Calculations are performed quite rapidly and only grow significantly with an increase in the dimension of the control vector. Extension to much larger distillation units with the same controls thus seems practical.

(Rafal and Stevens, AiChe Journal, 1968)

USE OF LINEAR PROGRAMMING METHODS FOR SYNTHESIZING SAMPLED-DATA AUTOMATIC SYSTEMS

A. I. Propoi
(Moscow)

Translated from Avtomatika i Telemekhanika, Vol. 24, No. 7, pp. 912-920, July, 1963
Original article submitted September 24, 1962

(Propoi, AiChe Journal, 1963)

Late 1970s

MPC diventa la tecnica più usata nell'industria petrolchimica

Fine anni 70…

Automatica, Vol. 14, pp. 413–428
Pergamon Press Ltd. 1978. Printed in Great Britain

0005-1098/78/0901-0413 $02.00/0

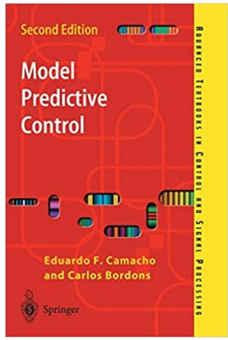
Model Predictive Heuristic Control: Applications to Industrial Processes*

J. RICHALET,† A. RAULT,† J. L. TESTUD† and J. PAPON†

Different industrial processes are being computer controlled using a new dual algorithm which identifies input–output impulse responses and computes control inputs as needed to realize desired output trajectories even when system noise, disturbances, and parameter variations occur.

Key Word Index—Chemical industry; computer control; control theory; dual control; identification; Lyapunov methods; multivariable control systems; oil refining; predictive control; steam generators.

(Richalet et al., Automatica, 1978)



(Camacho and Bordons, 1992)

Dynamic matrix control??A computer control algorithm

Cutler, C. R. , Ramaker, B. L.

Details | Contributors | Fields of science | Bibliography | Quotations | Similar | Collections

Source

Joint Automatic Control Conference > 1980 > 17 > 72

Abstract

The Dynamic Matrix Control (DMC) Algorithm is a control technology that has been used successfully in process computer applications in Shell for the last six years. The general development of the DMC Algorithm to incorporate feedforward and multivariable control is covered in this paper. The DMC Algorithm evolved from a technique of representing process dynamics with a set of numerical coefficients. The numerical technique, in conjunction with a least square formulation to minimize the integral of the error/time curve, make it possible to solve complex control problems on a digital computer which are not solvable with traditional PID control concepts. The incorporation of the process dynamics into the synthesis of the design of the DMC, make it possible to maintain an awareness of deadtime and unusual dynamic behavior.

(Cutler and Ramaker, ACC, 1980)

1982

Si dimostra che la stabilità dell'MPC può essere provata con la teoria di Lyapunov

1982

Automatica, Vol. 18, No. 3, pp. 349–352, 1982
Printed in Great Britain.

0005–1098/82/030349–04$03.00/0
Pergamon Press Ltd.

Brief Paper

On Receding Horizon Feedback Control*

C. C. CHEN† and L. SHAW‡

Key Words—Receding horizon; nonlinear systems; inverse optimal control; stability; Lyapunov methods.

(Chen and Shaw, Automatica 1982)

1990s

Introduzione del concetto degli ingredienti terminali per la stabilità

Anni 90

Automatica, Vol. 28, No. 6, pp. 1229–1233, 1992
Printed in Great Britain.

0005–1098/92 $5.00 + 0.00
Pergamon Press Ltd
© 1992 International Federation of Automatic Control

Brief Paper

Stable Redesign of Predictive Control*

EDOARDO MOSCA† and JINGXIN ZHANG‡

Key Words—Predictive control; stability; nonminimum phase systems; tracking systems; discrete time systems.

(Mosca and Zhang, Automatica 1992)

Pergamon

0005–1098(94)00109–X

Automatica, Vol. 30, No. 12, pp. 2013–2015, 1994
Elsevier Science Ltd
Printed in Great Britain
0005-1098/94 $7.00 + 0.00

Technical Communique

On the Stabilizing Property of SIORHC*†

A. BEMPORAD,‡ L. CHISCI‡ and E. MOSCA‡

Key Words—Predictive control; receding horizon control; LQ control; dynamic programming.

(Bemporad, Chisci and Mosca, IEEE TAC 1994)

Constrained receding-horizon predictive control

D.W. Clarke, FEng
R. Scattolini

(Clark and Scattolini, 1992)

On the Robustness of Receding-Horizon Control with Terminal Constraints

G. De Nicolao, L. Magni, and R. Scattolini

Abstract—Robustness properties of nonlinear receding-horizon controllers with zero terminal state constraints are investigated with respect to gain and additive perturbations. Some robustness margins are derived by extending to the receding-horizon case the analysis originally proposed by Geromel and da Cruz [1] for infinite-horizon controllers. In the linear case, it is shown that the zero terminal state receding-horizon controller exhibits worse robustness margins compared to standard infinite-horizon LQ control.

(De Nicolao, Magni and Scattolini, IEEE TAC 1995)

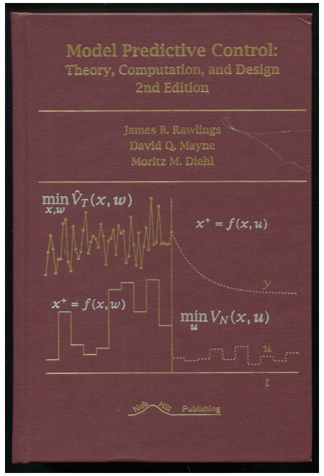
2000

Teoria matematica completamente formulata (Survey di Mayne, Rawlings, et al. su Automatica)

2000

(David Mayne)

(James B. Rawlings)



PERGAMON

Automatica 36 (2000) 789–814

www.elsevier.com/locate/automatica

Survey Paper

Constrained model predictive control: Stability and optimality☆

D. Q. Mayne[a,*], J. B. Rawlings[b], C. V. Rao[b], P. O. M. Scokaert[c]

[a]*Department of Electrical and Electronic Engineering, Imperial College of Science, Technology and Medicine, London SW7 2BT, UK*
[b]*Department of Chemical Engineering, University of Wisconsin, Madison, USA*
[c]*Centre National d'Etudes des Telecommunications, France Telecom, France*

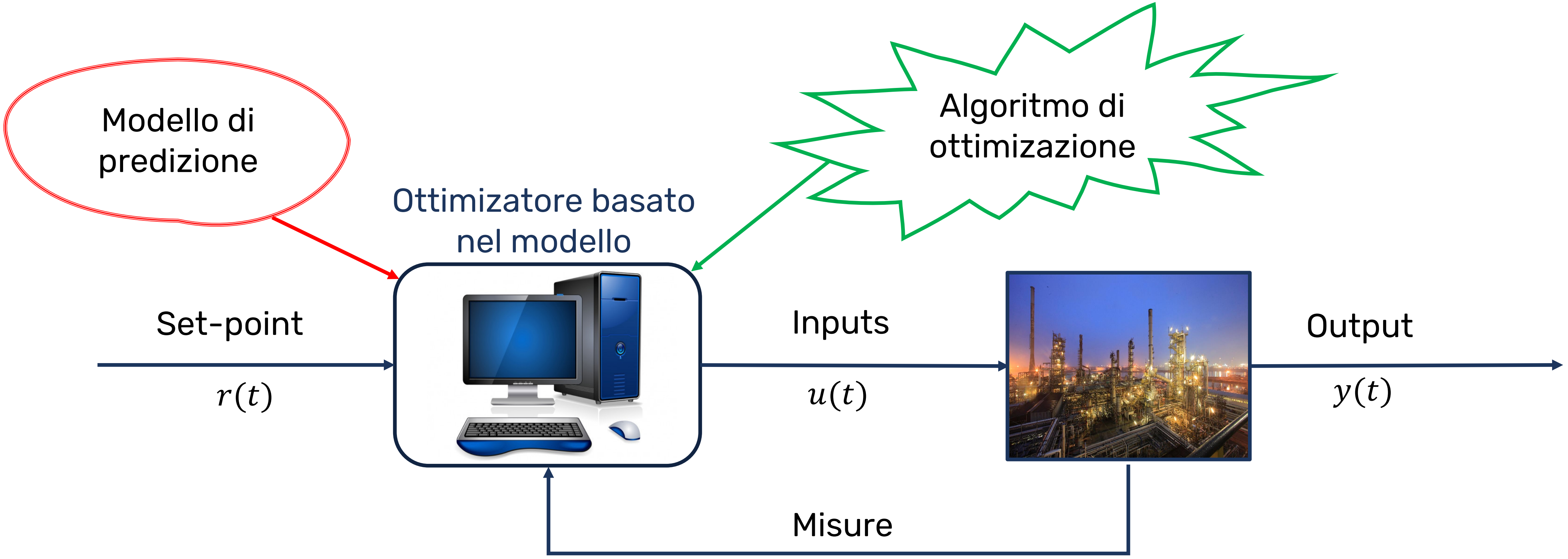
Received 29 May 1998; revised 6 September 1999; received in final form 8 November 1999

Abstract

Model predictive control is a form of control in which the current control action is obtained by solving, at *each* sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant. An important advantage of this type of control is its ability to cope with hard constraints on controls and states. It has, therefore, been widely applied in petro-chemical and related industries where satisfaction of constraints is particularly important because efficiency demands operating points on or close to the boundary of the set of admissible states and controls. In this review, we focus on model predictive control of constrained systems, both linear and nonlinear and discuss only briefly model predictive control of unconstrained nonlinear and/or time-varying systems. We concentrate our attention on research dealing with stability and optimality; in these areas the subject has developed, in our opinion, to a stage where it has achieved sufficient maturity to warrant the active interest of researchers in nonlinear control. We distill from an extensive literature essential principles that ensure stability and use these to present a concise characterization of most of the model predictive controllers that have been proposed in the literature. In some cases the finite horizon optimal control problem solved on-line is exactly equivalent to the same problem with an infinite horizon; in other cases it is equivalent to a modified infinite horizon optimal control problem. In both situations, known advantages of infinite horizon optimal control accrue.

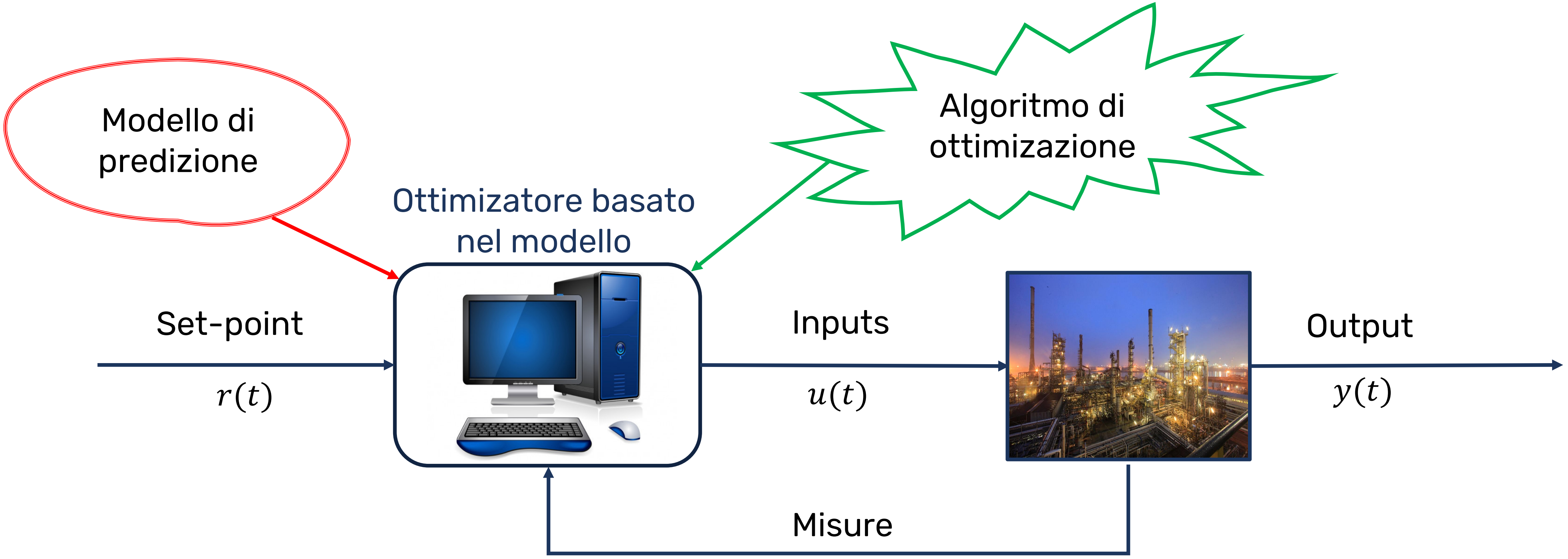
Keywords: Model predictive control; Stability; Optimality; Robustness

Cos'è il Model Predictive Control?



MPC usa un modello dinamico del processo per predire il suo comportamento futuro e calcolare la migliore azione di controllo

Cos'è il Model Predictive Control?



MPC usa un modello dinamico semplificato del processo per predire il suo probabile comportamento futuro e calcolare una buona azione di controllo

Come funziona

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

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

$s.t.$

$$x(j+1) = f(x(j), u(j))$$
$$y(j) = g(x(j))$$

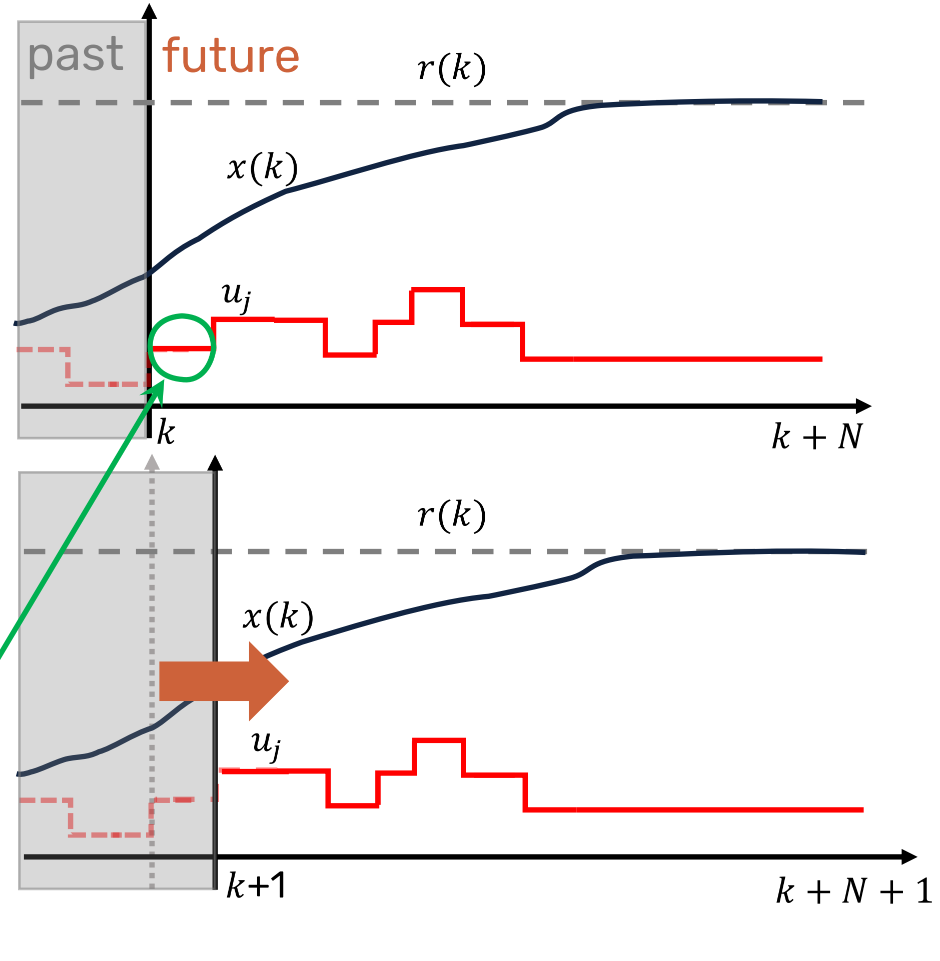
modello

$$u_{min} \leq u_j \leq u_{max}$$
$$x_{min} \leq x_j \leq x_{max}$$

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

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***.

Outline

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***.

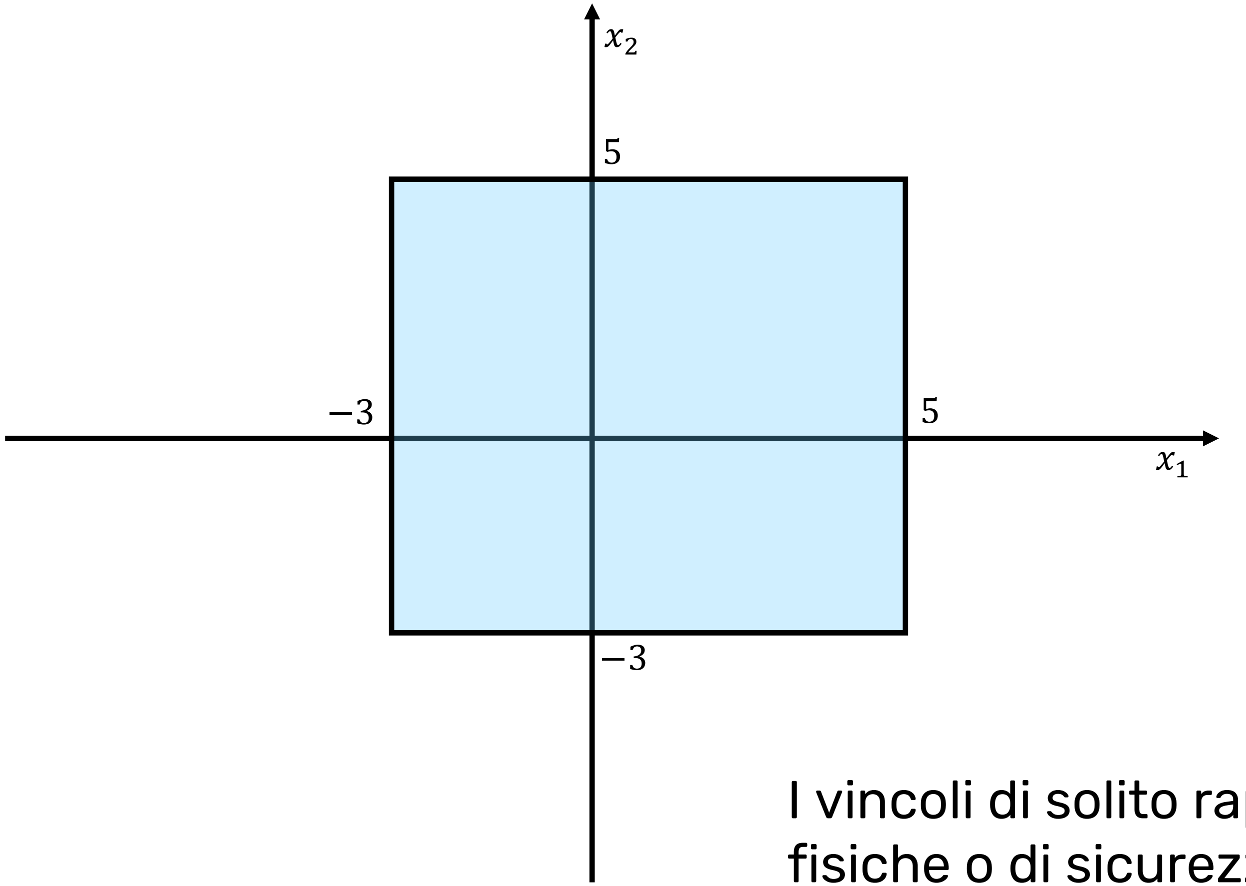
2. Si assuma inoltre l'esistenza di vincoli su stato e controllo $x \in \mathcal{X}, u \in \mathcal{U}$

$$x \in \mathcal{X} \Longrightarrow x_{min} \leq x \leq x_{max},$$

$$u \in \mathcal{U} \;\Longrightarrow\; u_{min} \leq u \leq u_{max}$$

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

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

$$= \sum_{j=0}^{N-1} x(j)'Qx(j) + u(j)'Ru(j) + x(N)'Sx(N)$$

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:

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

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

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 (anello chiuso, anello aperto, programmazione quadratica) 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:

$$\boldsymbol{K}_{cl}^{o} = \begin{bmatrix} K(0) \\ K(1) \\ K(2) \\ K(3) \\ K(4) \end{bmatrix} = \begin{bmatrix} 0.4249 & 1.2498 \\ 0.4458 & 1.2846 \\ 0.5517 & 1.4385 \\ 0.9710 & 1.9613 \\ 0 & 0.9901 \end{bmatrix} \qquad \boldsymbol{u}_{cl}^{o} = \begin{bmatrix} 0.0249 \\ -0.4250 \\ -0.3250 \\ -0.1748 \\ -0.0991 \end{bmatrix}$$

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 $\begin{bmatrix} -2 \\ -2 \end{bmatrix} \leq x \leq \begin{bmatrix} 4 \\ 4 \end{bmatrix}, -0.5 \leq u \leq 0.5$

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 quadratica, 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. Otteniamo:

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

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: $-0.3 \leq \boldsymbol{u} \leq 0.3$

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