Università degli Studi di Padova

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in Ingegneria dell'Informazione

Controllo predittivo data-driven con garanzie in catena chiusa

Relatore: Prof. Alessandro Chiuso

Laureando: *Emanuele Bano*

Anno Accademico 2021/2022
Data di Laurea: 23-09-2022

Abstract

Il seguente elaborato analizza lo sviluppo recente della teoria di controllo data-driven basato sull'utilizzo di informazioni provenienti dai dati di processo per l'analisi e il progetto di sistemi di controllo, in contrasto con la teoria classica model-based.
Nella prima parte vengono esposte le motivazioni che portano alla necessità di sviluppare una nuova teoria per risolvere le sfide moderne del controllo, caratterizzate da sistemi di crescente complessità e dinamica non nota, e infine si introducono i diversi approcci con cui è possibile utilizzare i dati, facendone un confronto in termini di caratteristiche e proponendo delle motivazioni per cui potrebbe essere vantaggioso evitare lo step di identificazione.
Nella seconda parte si vuole invece formulare un framework data-driven per la risoluzione di problemi di controllo ottimo, ponendo l'enfasi sulla teoria sviluppata che permette di convalidare gli algoritmi di controllo analizzati, offrendo garanzie in catena chiusa quali la stabilità e la feasibility del problema di ottimizzazione. A tal fine viene introdotto il framework model-based MPC, mostrando come sia possibile fornire condizioni teoriche per poter garantire le proprietà richieste, e come queste condizioni vanno ad impattare sulle caratteristiche del controllore, evidenziando ancora una volta il ruolo primario della teoria alla base degli algoritmi di controllo per ottenere le migliori performance. Successivamente viene mostrato come i concetti precedentemente introdotti in un contesto model-based possano essere estesi ad un framework data-driven per mezzo del cosiddetto *fundamental lemma* che permette di fornire una descrizione di un sistema LTI tramite l'utilizzo di una sola traiettoria input-output, sotto alcune condizioni speciali sull'ingresso. Nell'ultima parte vengono quindi mostrati due algoritmi per realizzare un controllore predittivo data-driven per sistemi LTI in assenza di rumore e con rumore limitato, mostrandone le proprietà che si riescono ad ottenere e come questi concetti possano essere estesi a sistemi non-lineari. Infine viene riportato un esempio di applicazione in Matlab di un controllore data-driven MPC nel caso di sistemi lineari in condizioni nominali.

Indice

Capitolo 1

Introduzione al Data-driven control

A partire dagli anni sessanta si è sviluppata la teoria del controllo moderna basata sui modelli di stato in grado di trattare sistemi con più input ed output (sistemi MIMO). La teoria moderna include metodi di controllo sia per sistemi lineari che non lineari ed è anche chiamata *model-based control* (MBC).
Nelle sue applicazioni, il primo step consiste nella modellizzazione del processo da controllare e successivamente nella sintesi del controllore basato sul modello matematico ricavato. Per questo motivo, l'accuratezza del modello gioca un ruolo chiave nelle performance di un'approccio MBC e questo può richiedere un tempo significativo e numerose competenze tecniche specifiche del campo di interesse.
I processi reali che vengono presi in considerazione nelle applicazioni di controllo possono essere divisi in quattro categorie secondo [8], pagina 6:

C1. processi per i quali sono disponibili modelli accurati basati su nozioni a priori;

C2. processi per i quali possono essere trovati modelli con moderate incertezze;

C3. processi per i quali la modellizzazione tramite conoscenze a priori più identificazione sono troppo complicati;

C4. processi per i quali non sono disponibili conoscenze a priori sulla dinamica e non è possibile definire un modello matematico.

La teoria classica dispone di metodi per risolvere problemi di controllo in cui il processo fa parte delle categorie C1 e C2, mentre questi metodi non funzionano per le categorie C3 e C4 per i quali i controllori progettati risulterebbero troppo complicati e costosi e talvonta nemmeno realizzabili.

Questi tipi di sistemi rappresentano le nuove sfide dei controlli e sono tipicamente caratterizzati dalle seguenti proprietà:

1. dinamica fortemente non lineare;
2. dinamica non conosciuta;
3. numerosi gradi di libertà per descrivere il sistema (*high-dimensional system*).

Grazie allo sviluppo della tecnologia, molti processi industriali generano e memorizzano enormi quantità di dati che contengono informazioni sullo stato del processo nel tempo. Nel corso degli anni il numero di dati generati è cresciuto esponenzialmente col tempo (come si può vedere in Figura 1.1) e l'uso di questi dati, sia offline che online, è fondamentale soprattutto nei casi in cui risulti difficile sviluppare un modello matematico accurato.

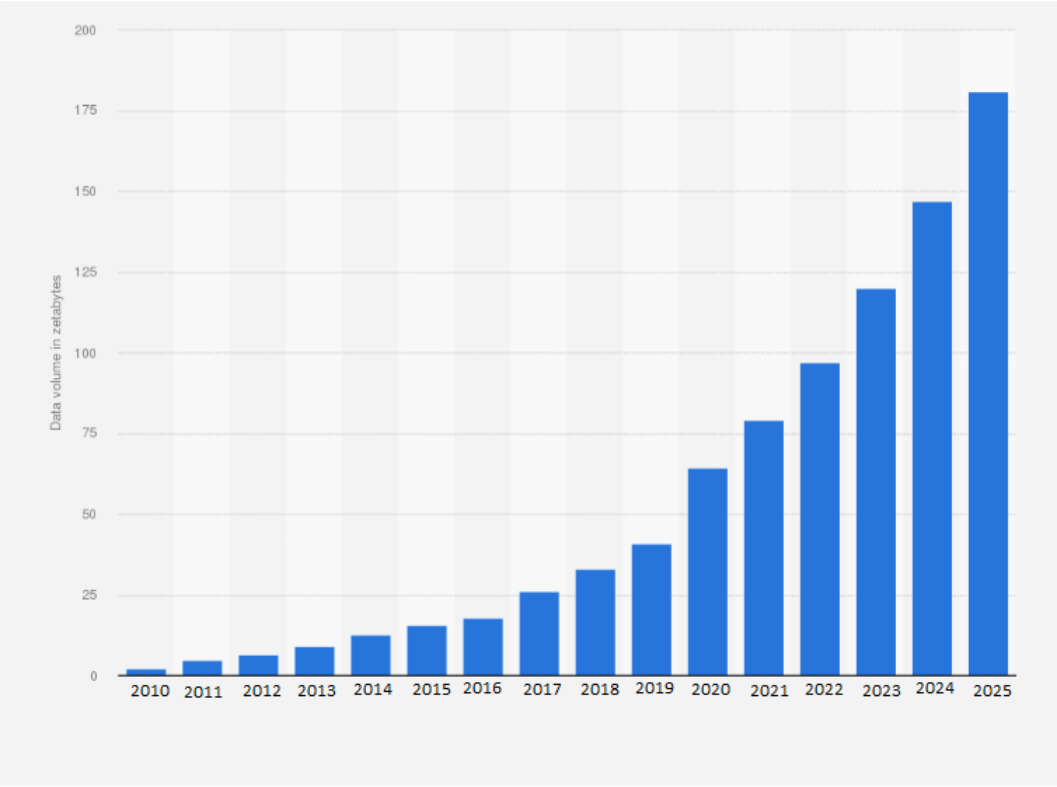


Figura 1.1: Incremento del numero di dati a disposizione negli anni

E' per questo motivo che si è resa la necessità di sviluppare una teoria data-driven che faccia utilizzo delle informazioni proveniente dai dati per progettare controllori, predirre gli stati del sistema, prendere decisioni e analizzare le performance.

L'obiettivo finale dovrebbe essere quello di creare un framework che unisca i vantaggi dei due approcci, mantenendo la facilità di applicazione di un approccio data-driven ma allo stesso tempo fornendo garanzie rigorose tipiche di un approccio model-based e fondamentali per risolvere problemi di controllo reali.

	Controllo model-based	**Controllo data-driven**
Vantaggi		
	+ teoria consolidata e ampia	+ facilità di applicazione senza conoscenza specifica del campo
	+ usata nella maggior parte delle applicazioni di controllo reali	+ indipendente dall'accuratezza del modello
	+ garanzie rigorose	+ non vi è dinamica non modellizzata
Svantaggi		
	- richiede competenze tecniche specifiche del campo	- non sempre è possibile garantire proprietà e performance
	- modelli tipicamente complicati da realizzare	- non dà informazioni sul funzionamente del sistema
	- la parte di modellizzazione occupa la maggior parte del tempo di progettazione	- impossibilità di applicazione su sistemi reali nel caso di mancanza di garanzie

Tabella 1.1: Confronto della caratteristiche dei due approcci

I problemi di controllo si basano solitamente sull'ottimizzazione di una funzione di costo che realizza un trade-off tra l'errore di tracking rispetto alla traiettoria obiettivo e il costo di attuazione.

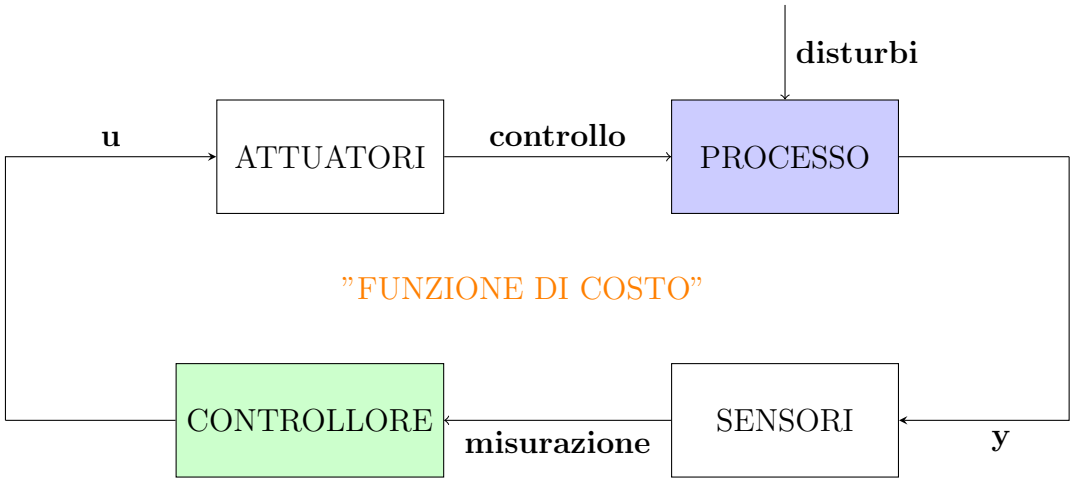


Figura 1.2: Sistema di controllo in catena chiusa con ottimizzazione di una funzione di costo

Sono stati proposti in letteratura i seguenti metodi basati sull'utilizzo dei dati per risolvere il problema di ottimizzazione:

- **Metodo indiretto data-driven**: i dati a disposizione vengono utilizzati per creare modelli data-driven risolvendo il problema dell' *unknown dinamics* tramite identificazione di sistema, realizzando un modello ridotto che approssima bene la dinamica del sistema. Questo modello può essere poi utilizzato per progettare il controllore con le tecniche classiche come ad esempio il Model-Predictive-Control;
- **Metodo diretto data-driven**: la sintesi del controllore viene fatta utilizzando direttamente i dati di misura del sistema senza un uso esplicito di un modello del sistema o di uno step di identificazione.

1.1 Confronto dei due approcci

I due approcci data-driven proposti si distinguono operativamente per la presenza o meno di uno step di identificazione di un modello matematico del processo.
Si rende quindi necessario un confronto in termini di prestazioni dei due metodi in un contesto di controllo predittivo per sistemi lineari tempo-invarianti stocastici.
Consideriamo il modello di stato:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k \end{aligned} \tag{1.1}$$

con $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}, x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^p$ e con $w_k \in \mathbb{R}^n$ rumore stocastico generato da un processo i.i.d. con distribuzione $\mathcal{N}(0, \Omega_w)$.
La presenza di rumore nel dataset introduce informazione irrilevante ai fini del controllo, e poiché il modello non è noto, le prestazioni del controllore sono fortemente dipendenti dalla capacità di estrarre le informazioni di interesse dai dati. Un'analisi consolidata e completa dei vantaggi dei due approcci non è ancora presente in letteratura e si rimanda a [6] per il punto di vista degli autori.
L'interesse recente per i metodi diretti è motivato dal fatto di rendere non necessario lo step di identificazione per la sintesi del controllore e dalla scommessa di ottenere prestazioni migliori in molti casi di interesse. Ci si chiede però quali possano essere i vantaggi di non passare attraverso un modello del sistema. Nell'approccio indiretto infatti si usano i dati a disposizione per trovare un modello di stato con le stime delle matrici $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ dalle quali viene sintetizzato il controllore. Nel caso in cui dai dati si potesse ricostruire esattamente le matrici, si otterrebbero le stesse prestazioni dell'approccio ottimo model-based basato sul modello reale. Qualora invece questo non fosse possibile, è fondamentale fornire una descrizione dell'incertezza nell'aver stimato $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$. Questa è infatti necessaria per impostare il trade-off della funzione di costo, poiché nel caso di grande incertezza è preferibile progettare controllori meno aggressivi in quanto una maggiore spesa in termini di energia di attuazione potrebbe non portare ad avere un migliore accuratezza di tracking, rischiando pure di destabilizzare il sistema. L'approccio diretto in questo caso potrebbe essere migliore (come si nota da un punto di vista simulativo), riuscendo a descrivere l'incertezza in un insieme di situazioni più ampio. Nel processo di indentificazione infatti, nella maggior parte dei casi non è noto l'ordine del sistema e deve essere fatta una scelta di trade-off tra complessità e accuratezza dei parametri stimati, rischiando di tralasciare dei poli di interesse, dovendone tener conto nell'incertezza del modello finale.

Capitolo 2

Controllo ottimo e Model-Predictive-Control framework

Uno dei principali svantaggi di un approccio data-driven è quello di non avere nella maggior parte dei casi teorie matematiche ben fondate che permettano di ottenere garanzie sul sistema in catena chiusa, fondamentali per poter utilizzare queste tecniche in ambiti in cui la sicurezza ha un ruolo primario.
Il controllo classico model-based invece, ha sviluppato varie tecniche che permettono di studiare le proprietà del sistema controllato. Uno degli approcci più popolari e utilizzati a livello industriale è il *Model-Predictive-Control* (MPC) che offre una teoria ben consolidata in grado di stabilizzare il sistema in catena chiusa sotto alcune leggere assunzioni. L'obiettivo di questo capitolo è quello di introdurre il framework dell'MPC basato sulla conoscenza di un modello del sistema per poi estenderlo ad un framework data-driven tramite gli strumenti che verranno presentati nel capitolo 3, studiando le garanzie che si riescono ad ottenere in questo modo.
L'MPC ha tre principali ingredienti:

1. una **funzione di costo** che deve essere minimizzata risolvendo un problema di ottimizzazione;

2. una serie di **vincoli** (su ingressi, variabili di stato e d'uscita) che devono essere soddisfatti;

3. un **modello matematico** che viene utilizzato per prevedere il comportamento futuro del sistema.

I vantaggi di questo approccio sono il modo sistematico di includere vincoli e la possibilità di ottenere buone performance di controllo, mentre le principali sfide sono il fatto che il problema di controllo deve essere risolto in real-time richiedendo hardware di complessità sempre maggiore, e inoltre la stabilità e la robustezza del controllo non sono sempre garantite senza aver prima verificato alcune condizioni teoriche sui parametri di progetto. Inoltre va affrontato il problema della *infeasibility* che si verifica quando il problema di ottimizzazione diventa non risolvibile per tempi futuri e non esiste una soluzione di controllo in grado di soddisfare tutti i vincoli.
Nel paragrafo 2.1 vengono introdotti i problemi di ottimizzazione per poi poter parlare di controllo ottimo, introducendo i metodi di soluzione nel caso di assenza o presenza di vincoli, arrivando alla realizzazione dell'MPC tramite Receding Horizon Control nel paragrafo 2.5, studiandone stabilità e feasibility.

2.1 Problemi di ottimizzazione

Un problema di ottimizzazione ha formulazione generale:

$$\begin{aligned} &\inf_x && f(x) \\ &\text{subj. to} && x \in \mathcal{S} \subseteq \mathcal{X} \end{aligned} \tag{2.1}$$

dove x è il vettore delle variabili di decisione, $\mathcal{X}$ è il dominio del problema di ottimizzazione, e $\mathcal{S} \subseteq \mathcal{X}$ è l'insieme delle decisioni ammissibili (*feasible solutions*) e $f\colon \mathcal{X} \to \mathbb{R}$ è una funzione che assegna ad ogni decisione $\bar{x}$ un costo $f(\bar{x})$.
L'insieme delle soluzioni ammissibili è definito attraverso dei vincoli di uguaglianza e/o disuguaglianza

$$\begin{aligned} &\inf_x && f(x) \\ &\text{subj. to} && g_i(x) \leq 0 \quad \text{con } i = 1, \dots, m \\ & && h_j(x) = 0 \quad \text{con } j = 1, \dots, p \\ & && x \in \mathcal{X} \end{aligned} \tag{2.2}$$

ed è quindi dato in modo compatto da:

$$\mathcal{S} = \{x \in \mathcal{X} : g_i(x) \leq 0, h_j(x) = 0 \quad \forall i, j\}$$

Nel caso in cui $\mathcal{S} = \mathcal{X}$ il problema non presenta vincoli.

Risolvere il problema (2.2) significa trovare la soluzione ottima f^* tale che

$$f^* = \inf_{x \in \mathcal{S} \subseteq \mathcal{X}} f(x)$$

Si possono verificare i seguenti casi:

- $f^* = -\infty$ se il problema non è limitato inferiormente
- $\mathcal{S} = \emptyset$ se non esistono soluzioni ammissibili al problema e si pone $f^* = +\infty$
- $\exists\, x^* \in \mathcal{S} \subseteq \mathcal{X}$ tale che $f(x^*) = f^* = \min_{x \in \mathcal{S} \subseteq \mathcal{X}} f(x)$ e tale valore è detto *soluzione ottima*

In generale una soluzione analitica di (2.2) non esiste, e queste devono essere calcolate numericamente per mezzo di algoritmi iterativi che partono da un'ipotesi iniziale x_0 e si muovono nello spazio delle decisioni generando una sequenza $\{x^k\}_{k=1}^{k=k_{max}}$ con

$$x^{k+1} = \Psi(x^k, f, \mathcal{S})$$

dove Ψ è una legge di aggiornamento tale che dopo un numero di passi k_{max} si ottiene una soluzione approssimata $|f(x^{k_{max}}) - f(x^*)| < \varepsilon$ con ε costante che definisce l'accuratezza dell'approssimazione.
Una classe molto importante di problemi di ottimizzazione riguarda i problemi *convessi* con funzione di costo quadratica e vincoli lineari. Questi programmi infatti possono essere usati per formulare molti problemi pratici di controllo ottimo e possono essere risolti efficientemente.

2.2 Formulazione generale del problema di controllo ottimo

Consideriamo il seguente sistema discreto non lineare tempo invariante

$$x(k+1) = g(x(k), u(k)) \quad x \in \mathbb{R}^n,\ u \in \mathbb{R}^m \tag{2.3}$$

soggetto a n_c vincoli

$$h(x(k), u(k)) \leq 0 \quad h\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n_c} \tag{2.4}$$

dove si suppone assegnato lo stato iniziale $x(0) = x_0$.
Il problema che ci si pone è quello di determinare un vettore di ingressi

$$\mathbf{u} = [u(0), u(1), \ldots, u(L-1)] \tag{2.5}$$

che, agendo nell'intervallo finito $[0, L-1]$, renda minimo il seguente indice (o *funzione obiettivo*) $J_{0\to L}(\cdot)$

$$J_{0\to L}(x_0, \mathbf{u}) = \underbrace{p_L(x_L)}_{\text{"peso" sullo stato finale}} + \underbrace{\sum_{k=0}^{L-1} q(x_k, u_k)}_{\text{"peso" sull' evoluzione in } [0,L-1]} \tag{2.6}$$

dove L viene detto orizzonte temporale (*time horizon*), x_k[1] è lo stato predetto del sistema (2.15) al tempo k ottenuto a partire da x_0 applicando la sequenza di controllo $u_0, \ldots, u_{k-1}$, e $p_L(x_L), q(x_k, u_k)$ sono funzioni definite positive ($p \succ 0, q \succ 0$)

$$p(x,u) > 0 \ \forall x \neq 0, u \neq 0, p(0,0) = 0$$
$$q(x,u) > 0 \ \forall x \neq 0, u \neq 0, q(0,0) = 0$$

Analizzando la struttura di $J_{0\to L}(x_0, \mathbf{u})$ in (2.6) si nota che i due addendi che la compongono danno conto di due aspetti che giocano nella valutazione del comportamento del sistema in $[0, L]$:

- $p_L(x_L)$ rappresenta il costo dovuto alla distanza dello stato finale del sistema dallo stato desiderato (costo terminale);
- $\sum_{k=0}^{L-1} q(x_k, u_k)$ indica il costo dovuto allo scostamento dallo stato desiderato degli stati intermedi e all'energia spesa per il controllo.

Possiamo ora definire il seguente problema di controllo ottimo a tempo finito con vincoli:

$$\begin{aligned} J^*_{0\to L}(x_0) = \quad & \min_{u_0,\ldots,u_{L-1}} \quad && J_{0\to L}(x_0, \mathbf{u}) \\ & \text{subj. to} && x_{k+1} = g(x_k, u_k), \quad k = 0, \ldots, L-1 \\ & && h(x_k, u_k) \leq 0, \quad k = 0, \ldots, L-1 \\ & && x_L \in \mathcal{X}_f \end{aligned} \tag{2.7}$$

[1] viene indicato invece con $x(k)$ lo stato misurato al tempo k

dove $\mathcal{X}_f \subseteq \mathbb{R}^n$ viene detta *regione terminale* e rappresenta un insieme di vincoli sulla stato finale predetto che viene imposto per migliorare le proprietà del controllore.
La soluzione del problema consiste nel trovare $\mathbf{u}^*$ ottimo in modo da minimizzare l'indice $J_{0\to L}(\cdot)$.
La feasibility del problema dipende dalla scelta della condizione iniziale in relazione alla regione terminale, ossia almeno uno degli elementi in $\mathcal{X}_f$ deve essere raggiungibile a partire da x_0 applicando una particolare sequenza di ingresso $u_0, \dots, u_{L-1}$.
Possiamo determinare l'insieme delle soluzioni iniziali ammissibili in modo ricorsivo: definiamo $\mathcal{X}_{j\to L}$ l'insieme degli stati x_j al tempo j per i quali esiste un ingresso al sistema che porti ad avere una traiettoria che finisce in $\mathcal{X}_f$, ovvero

$$\begin{aligned}\mathcal{X}_{j\to L} &= \{x \in \mathbb{R}^n \colon \exists\, \mathbf{u} \text{ tale che } h(\mathbf{x},\mathbf{u}) \le 0,\ g(\mathbf{x},\mathbf{u}) \in \mathcal{X}_{j+1\to L}\} \\ &\qquad j = 0, \dots, L-1 \\ \mathcal{X}_{L\to L} &= \mathcal{X}_f\end{aligned}$$

quindi $\mathcal{X}_{0\to L}$ rappresenta l'insieme delle condizioni iniziali per cui esiste una soluzione al problema di controllo ottimo.

2.2.1 Approccio batch e programmazione dinamica

La soluzione del problema (2.7) può essere espressa e risolta come un problema di programmazione non-lineare (*approccio batch*) o in maniera ricorsiva tramite il principio di ottimalità di Bellman (*programmazione dinamica*). Nel primo caso, scrivendo esplicitamente i vincoli dati della dinamica del sistema in (2.15)

$$\begin{aligned} x_1 &= g(x_0, u_0) \\ x_2 &= g(x_1, u_1) \\ &\vdots \\ x_L &= g(x_{L-1}, u_{L-1}) \end{aligned} \tag{2.8}$$

otteniamo la seguente riscrittura del problema di ottimizzazione (2.7)

$$\begin{aligned} J^*_{0\to L}(x_0) = \quad & \min_{u_0,\dots,u_{L-1}} && J_{0\to L}(x_0, \mathbf{u}) \\ & \text{subj. to} && x_1 = g(x_0, u_0) \end{aligned}$$

$$
\begin{aligned}
& x_2 = g(x_1, u_1) \\
& \vdots \\
& x_L = g(x_{L-1}, u_{L-1}) \\
& h(x_k, u_k) \leq 0, \quad k = 0, \ldots, L-1 \\
& x_L \in \mathcal{X}_f \\
& x_0 = x(0)
\end{aligned}
\tag{2.9}
$$

che ha la forma del problema di ottimizzazione di programmazione non lineare (2.2) con variabili $\mathbf{u}, \mathbf{x}$.

Alternativamente si possono sostituire gli stati con $x_k = \overbrace{g(g(\ldots(g(x_0, u_0))}^{k \text{ volte}}, u_{k-1})$ ottenendo un problema di ottimizzazione nella sola variabile $\mathbf{u}$.
La soluzione di (2.9) porta a determinare la sequenza di ingresso ottima $\mathbf{u}^* = [u_0^*, \ldots, u_{L-1}^*]$ in funzione del particolare stato iniziale x_0.
L'approccio ricorsivo, invece, si basa sul principio di ottimalità di Bellman:

Principio di ottimalità 1 (Bellman, 1957)**.** *Una soluzione ottima ha la proprietà che qualsiasi sia lo stato iniziale e la decisione iniziale, le successive decisioni devono costituire una soluzione ottima rispetto allo stato determinato dalla prima decisione.*

Possiamo usare il principio di ottimalità per risolvere il problema di controllo ottimo ricorsivamente definendo la funzione di costo ridotta (*cost-to-go*) dall'istante j all'istante L

$$
J_{j \to L}(x_j, u_j, \ldots, u_{L-1}) = p_L(x_L) + \sum_{k=j}^{L-1} q(x_k, u_k) \tag{2.10}
$$

con il cost-to-go ottimo dato da

$$
\begin{aligned}
J^*_{j \to L}(x_j) = \quad & \min_{u_j, \ldots, u_{L-1}} \quad && J_{j \to L}(x_j, u_j, \ldots, u_{L-1}) \\
& \text{subj. to} && x_{k+1} = g(x_k, u_k), \quad k = j, \ldots, L-1 \\
& && h(x_k, u_k) \leq 0, \quad k = j, \ldots, L-1 \\
& && x_L \in \mathcal{X}_f
\end{aligned}
\tag{2.11}
$$

che dipende solo dallo stato x_j.

Possiamo quindi riscrivere

$$\begin{aligned} J^*_{j-1\to L}(x_{j-1}) = \quad & \min_{u_{j-1}} && q(x_{j-1},u_{j-1}) + J^*_{j\to L}(g(x_{j-1},u_{j-1})) \\ & \text{subj. to} && h(x_{j-1},u_{j-1}) \le 0 \\ & && g(x_{j-1},u_{j-1}) \in \mathcal{X}_{j\to L} \end{aligned} \tag{2.12}$$

Per risolvere (2.7) possiamo usare il seguente algoritmo ricorsivo di programmazione dinamica partendo da

$$\begin{aligned} J^*_{L\to L}(x_L) &= p(x_L) \\ \mathcal{X}_{L\to L} &= \mathcal{X}_f \end{aligned} \tag{2.13}$$

e risolvendo a ritroso

$$\begin{aligned} J^*_{L-1\to L}(x_{L-1}) = \quad & \min_{u_{L-1}} && q(x_{L-1},u_{L-1}) + J^*_{L\to L}(g(x_{L-1},u_{L-1})) \\ & \text{subj. to} && h(x_{L-1},u_{L-1}) \le 0 \\ & && g(x_{L-1},u_{L-1}) \in \mathcal{X}_{L\to L} \\ & \vdots \\ J^*_{0\to L}(x_{L-1}) = \quad & \min_{u_0} && q(x_0,u_0) + J^*_{1\to L}(g(x_0,u_0)) \\ & \text{subj. to} && h(x_0,u_0) \le 0 \\ & && g(x_0,u_0) \in \mathcal{X}_{1\to L} \end{aligned}$$

Questo tipo di approccio può essere conveniente computazionalmente in quanto ogni step ricorsivo risolve un'ottimizzazione su una singola variabile. La principale difficoltà di questo metodo è che la struttura del cost-to-go ottimale, tranne in alcuni casi speciali, non è nota. In questi casi si può procedere con un'approccio *brute force* sfruttando la proprietà di sovrastruttura del problema: si considera $J^*_{j\to L}$ noto e si calcola $J^*_{j-1\to L}$ per ogni possibile $x_{j-1} \in \mathcal{X}_{j-1\to L}$ risolvendo il problema di programmazione non lineare (2.12).
Per questo tipo di approccio, la sequenza d'ingresso ottima $\mathbf{u}^*$ non dipende solamente dallo stato iniziale x_0, infatti ad ogni tempo j il cost-to-go ottimale

definisce una legge di controllo con feedback

$$\begin{aligned} u_j^*(x_j) = \arg\min_{u_j} \quad & q(x_j, u_j) + J^*_{j+1\to L}(g(x_j, u_j)) \\ \text{subj. to} \quad & h(x_j, u_j) \leq 0, \\ & g(x_j, u_j) \in \mathcal{X}_{j+1\to L} \end{aligned}$$

Le due soluzioni coincidono nel caso nominale, mentre in presenza di disturbi l'approccio dinamico in feedback risolta più robusto, poiché nella soluzione in catena aperta lo stato predetto e lo stato misurato potrebbe differire significativamente.

2.2.2 Orizzonte infinito

Siamo interessati a studiare le proprietà del problema (2.7) quando $L \to \infty$

$$\begin{aligned} J^*_{0\to\infty}(x_0) = \min_{u_0, u_1, \ldots} \quad & \sum_{k=0}^{\infty} q(x_k, u_k) \\ \text{subj. to} \quad & x_{k+1} = g(x_k, u_k), \quad k = 0, \ldots, \infty \\ & h(x_k, u_k) \leq 0, k = 0, \ldots, \infty \\ & x_0 = x(0). \end{aligned} \tag{2.14}$$

L'insieme degli stati iniziali per cui (2.14) ha soluzione è dato da

$$\mathcal{X}_{0\to\infty} = \{x(0) \in \mathbb{R}^n : \text{Il problema (2.14) ha soluzione e } J^*_{0\to\infty}(x_0) < +\infty\}$$

Condizione necessaria affinchè $J^*_{0\to\infty}(x_0) < +\infty$ è che

$$\lim_{k\to\infty} q(x_k, u_k) = 0$$

e poiché $q(x_k, u_k) > 0 \ \forall (x_k, u_k) \neq 0$ deve essere

$$\begin{cases} \lim_{k\to\infty} x_k = 0 \\ \lim_{k\to\infty} u_k = 0 \end{cases}$$

La sequenza di controllo generata dalla soluzione del problema a orizzonte infinito fa convergere quindi la traiettoria del sistema all'origine qualora esistano soluzioni (sistema stabilizzabile).

2.3 Controllo lineare quadratico ottimo senza vincoli

Si consideri un sistema lineare discreto deterministico con equazione di stato

$$x_{k+1} = Ax_k + Bu_k \tag{2.15}$$

dove si suppone assegnato lo stato iniziale $x(0) = x_0$ e che non vi siano vincoli sulle variabili d'ingresso e di stato. Il problema che ci si pone è quello di determinare un vettore di ingressi $\mathbf{u} = (u_0, \ldots, u_{L-1})$ che, agendo nell'intervallo finito [0,L], renda minimo il seguente indice (o funzione obiettivo) quadratico:

$$J_L(x_0, \mathbf{u}) = x_L^T P_f x_L + \sum_{k=0}^{L-1} (x_k^T W x_k + u_k^T U u_k) \tag{2.16}$$

dove P_f, W e U sono matrici semidefinite positive.

2.3.1 Soluzione tramite approccio batch

Sviluppando l'equazione di aggiornamento dello stato per istanti successivi si ottiene

$$\begin{aligned} x_1 &= Ax_0 + Bu_0 \\ x_2 &= Ax_1 + Bu_1 = A^2 x_0 + ABu_0 + Bu_1 \\ &\vdots \\ x_L &= A^T x_0 + \sum_{k=0}^{L-1} A^{L-k-1} B u_k \end{aligned}$$

da cui, mediante rappresentazione vettoriale con $\mathbf{x} = (x_1, \ldots, x_L)$ e $\mathbf{u} = (u_0, \ldots, u_{L-1})$, è possibile scrivere il sistema (2.15) come

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_L \end{bmatrix} = \begin{bmatrix} B & 0 & \cdots & \cdots & 0 \\ AB & B & \ddots & \ddots & 0 \\ A^2B & AB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{L-1}B & \cdots & \cdots & \cdots & B \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ \vdots \\ u_{L-1} \end{bmatrix} + \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^L \end{bmatrix} x_0 = F\mathbf{u} + Gx_0 \tag{2.17}$$

dove $F \in \mathbb{R}^{nL \times mL}$ e $G \in \mathbb{R}^{nL \times n}$.
É possibile quindi riscrivere, utilizzando la forma (2.17), l'espressione dell'indice di costo (2.16) nella seguente forma

$$J_L(x_0, \mathbf{u}) = \mathbf{x}^T \underbrace{\begin{bmatrix} W & & & 0 \\ & W & & \\ & & \ddots & \\ 0 & & & P_f \end{bmatrix}}_{\triangleq \overline{W} \geq 0} \mathbf{x} + \mathbf{u}^T \underbrace{\begin{bmatrix} U & & & 0 \\ & U & & \\ & & \ddots & \\ 0 & & & U \end{bmatrix}}_{\triangleq \overline{U} \geq 0} \mathbf{u} + x_0^T W x_0 =$$

$$= (F\mathbf{u} + Gx_0)^T \overline{W} (F\mathbf{u} + Gx_0) + \mathbf{u}^T \overline{U} \mathbf{u} + x_0^T W x_0 =$$

$$= \mathbf{u}^T F^T \overline{W} F \mathbf{u} + x_0^T G^T \overline{W} F \mathbf{u} + \mathbf{u}^T F^T \overline{W} G x_0 + x_0^T G^T \overline{W} G x_0 + \mathbf{u}^T \overline{U} \mathbf{u} + x_0^T W x_0 =$$

$$= \mathbf{u}^T \underbrace{(F^T \overline{W} F + U)}_{\mathcal{S}_{u,u}} \mathbf{u} + x_0^T \underbrace{G^T \overline{W} F}_{\mathcal{S}_{x_0,u}} \mathbf{u} + \mathbf{u}^T \underbrace{F^T \overline{W} G}_{\mathcal{S}_{u,x_0}^T} x_0 + x_0^T \underbrace{(W + G^T \overline{W} G)}_{\mathcal{S}_{x_0,x_0}} x_0 =$$

$$= \mathbf{u}^T \mathcal{S}_{u,u} \mathbf{u} + x_0^T \mathcal{S}_{x_0,u} \mathbf{u} + \mathbf{u}^T \mathcal{S}_{u,x_0}^T x_0 + x_0^T \mathcal{S}_{x_0,x_0} x_0$$

dove sono state definite opportunamente le matrici $\mathcal{S}_{u,u},\ \mathcal{S}_{x_0,u} = \mathcal{S}_{u,x_0}^T$ *e* $\mathcal{S}_{x_0,x_0}$.
Si vogliono dunque determinare gli ingressi ottimi $\mathbf{u}^*$ in grado di minimizzare tale funzionale di costo, cioé

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} J(u, x_0) \tag{2.18}$$

da cui si ottiene il valore minimo della funzione costo $J^*(x_0) = \min_{\mathbf{u}} J(x_0, \mathbf{u})$; questo è calcolabile in maniera esplicita azzerando la derivata di $J(\cdot)$ rispetto all'ingresso $\mathbf{u}$ data da

$$\frac{\partial J_N}{\partial \mathbf{u}} = 2\mathcal{S}_{u,u}\mathbf{u} + 2\mathcal{S}_{x_0,u} x_0 = 0 \tag{2.19}$$

Si ricava dunque l'ingresso ottimo

$$\mathbf{u}^* = -\mathcal{S}_{u,u}^{-1} \mathcal{S}_{x_0,u} x_0 \tag{2.20}$$

dove, nel caso in cui la matrice $\mathcal{S}_{u,u}$ non fosse invertibile, l'operazione di inversione viene sostituita da quella di pseudo-inversione.
E' possibile osservare come l'ingresso ottimo, cioè che minimizza la funzione costo considerata, dipende unicamente dalla condizione iniziale x_0: uno svantaggio di questa dipendenza sta nel fatto che un eventuale cambiamento delle condizioni

iniziali implica un nuovo calcolo di tutta la sequenza di ingresso (come avviene ad esempio in una implementazione receding horizon).
Con questo approccio è inoltre richiesta l'inversione di matrici di dimensione crescente con l'orizzonte temporale, causando una penalizzazione dal punto di vista computazionale che può venire parzialmente compensata sfruttando la struttura triangolare di tali matrici per una implementazione efficiente.

2.3.2 Soluzione tramite programmazione dinamica

Alternativamente si può usare l'algoritmo di programmazione dinamica per determinare la soluzione ricorsivamente.
L'obiettivo ora è quello di minimizzare il cost-to-go

$$J_j^*(x_j) \triangleq \min_{u_j,\dots,u_{L-1}} x_L^T P_f x_L + \sum_{k=j}^{L-1} (x_k^T W x_k + u_k^T U u_k) \tag{2.21}$$

che sfruttando il principio di ottimalità può essere riscritto come

$$\begin{aligned} J_j^*(x_j) &= \min_{u_j,\dots,u_{L-1}} x_j^T W x_j + u_j^T U u_j + \sum_{k=j+1}^{L-1} (x_k^T W x_k + u_k^T U u_k) + x_L^T P_f x_L \\ &= \min_{u_j} x_j^T W x_j + u_j^T U u_j + J_{j+1}^*(x_{j+1}) \end{aligned}$$

con

$$J_L^*(x_L) = x_L^T P_f x_L \tag{2.22}$$

Si ha quindi

$$J_{L-1}^*(x_{L-1}) = \min_{u_{L-1}} x_{L-1}^T W x_{L-1} + u_{L-1}^T U u_{L-1} + x_L^T P_f x_L \tag{2.23}$$

con

$$x_L = A x_{L-1} + B u_{L-1} \tag{2.24}$$

Sostituendo (2.24) in (2.23) si ottiene

$$\begin{aligned} J_{L-1}^*(x_{L-1}) = \min_{u_{L-1}} \{ & x_{L-1}^T W x_{L-1} + u_{L-1}^T U u_{L-1} \\ & + (A x_{L-1} + B u_{L-1})^T P_f (A x_{L-1} + B u_{L-1}) \} \\ = x_{L-1}^T W x_{L-1} + u_{L-1}^T U u_{L-1} & + (x_{L-1}^T A^T + u_{L-1}^T B^T) P_f (A x_{L-1} + B u_{L-1}) \end{aligned}$$

$$= x_{L-1}^T W x_{L-1} + u_{L-1}^T U u_{L-1} + x_{L-1}^T A^T P_f A x_{L-1} + x_{L-1}^T A^T P_f B u_{L-1} +$$

$$+ u_{L-1}^T B^T P_f A x_{L-1} + u_{L-1}^T B^T P_f B u_{L-1}$$

Per determinare l'ingresso ottimo u_{L-1}^* si calcola la derivata e, ponendola uguale a zero,

$$\frac{\partial J_{L-1}(x_{L-1}, u_{L-1})}{\partial u_{L-1}} = 0$$

$$\frac{\partial}{\partial u_{L-1}} \left(u_{L-1}^T (B^T P_f B + U) u_{L-1} + 2 u_{L-1}^T B^T P_f A x_{L-1} \right) = 0$$

$$2(B^T P_f B + U) u_{L-1} + 2 B^T P_f A x_{L-1} = 0$$

si arriva a determinare

$$u_{L-1}^* = - \underbrace{(B^T P_f B + U)^{-1} B^T P_f A}_{\triangleq G_{L-1}} x_{L-1} \tag{2.25}$$

con il costo ottimale

$$J_{L-1}^*(x_{L-1}) = x_{L-1}^T P_{L-1} x_{L-1} \tag{2.26}$$

dove si ha definito

$$\begin{aligned} P_L &= P_f \\ P_{L-1} &= W + A^T P_L A - A^T P_L B (B^T P_L B + R)^{-1} B^T P_L A \end{aligned} \tag{2.27}$$

Il prossimo passo di ottimizzazione è dato da

$$J_{L-2}^*(x_{L-2}) = \min_{u_{L-2}} x_{L-2}^T W x_{L-2} + u_{L-2}^T U u_{L-2} + x_{L-1}^T P_{L-1} x_{L-1} \tag{2.28}$$

con

$$x_{L-1} = A x_{L-2} + B u_{L-2} \tag{2.29}$$

Si riconosce che (2.28),(2.29) ha la stessa struttura di (2.23),(2.24), per cui possiamo determinare la soluzione ottimale direttamente

$$u^*_{L-2} = -\underbrace{\left(B^T P_{L-1} B + U\right)^{-1} B^T P_{L-1} A}_{\triangleq G_{L-2}} x_{L-2}$$

$$J^*_{L-2}(x_{L-2}) = x^T_{L-2} P_{L-2} x_{L-2}$$

$$P_{L-2} = W + A^T P_{L-1} A - A^T P_{L-1} B (B^T P_{L-1} B + R)^{-1} B^T P_{L-1} A$$

Continuando ricorsivamente si ottiene la soluzione ottima al problema di controllo

$$\begin{aligned} u^*(k) &= -\left(B^T P_{k+1} B + U\right)^{-1} B^T P_{k+1} A\, x(k), \quad k = 0, \ldots, L-1 \\ &= G_k\, x(k) \end{aligned} \tag{2.30}$$

con

$$P_k = \begin{cases} W + A^T P_{k+1} A - A^T P_{k+1} B (B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad 0 \le k \le L-1 \\ P \qquad k = L \end{cases}$$

chiamata *equazione alle differenze di Riccati.*
Il costo ottimo è dato da

$$J^*_k(x(k)) = x^T(k) P_k x(k) \tag{2.31}$$

Si nota da (2.30) che l'azione di controllo ottima $u^*(k)$ è ottenuta sotto forma di legge in retroazione come funzione lineare dello stato misurato $x(k)$ al tempo k, mentre da (2.31) si vede che il costo ottimo è una funzione quadratica dello stato al tempo k.

2.3.3 Proprietà del problema ad orizzonte infinito

Dato il sistema (2.15), obiettivo del problema di controllo ottimo ad orizzonte infinito è minimizzare il funzionale di costo

$$J^*_\infty(x(0)) = \min_{u_0, u_1, \ldots} \sum_{k=0}^{\infty} \left(x_k^T W x_k + u_k^T U u_k\right) \tag{2.32}$$

Diversamente al caso tempo finito, il termine relativo al costo terminale non avrebbe alcun senso pratico, per cui esso non è presente nel funzionale di costo. Poiché $L \to \infty$, l'applicazione del metodo batch richiederebbe l'inversione di matrici di dimensione infinita risultando computazionalmente impossibile. L'applicazione del metodo di programmazione dinamica rimane però praticabile.
Inizializziamo l'equazione alla differenze di Riccati

$$P_k = W + A^T P_{k+1} A - A^T P_{k+1} B (B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A \tag{2.33}$$

con $P_0 = W$ e risolviamo ricorsivamente per $k \to -\infty$.
Se si ha convergenza ad una soluzione P_∞ questa soddisfa *l'equazione algebrica di Riccati*

$$P_\infty = W + A^T P_\infty A - A^T P_\infty B (B^T P_\infty B + R)^{-1} B^T P_\infty A \tag{2.34}$$

L'azione di controllo in retroazione ottima è quindi

$$u^*(k) = -\underbrace{(B^T P_\infty B + U)^{-1} B^T P_\infty A}_{G_\infty} x(k) \tag{2.35}$$

e il costo ottimale

$$J^*_\infty(x(0)) = x^T(0) P_\infty x(0) \tag{2.36}$$

E' facile mostrare che se la coppia (A, B) è raggiungibile allora esiste una soluzione unica per il problema di ottimizzazione che stabilizza il sistema in catena chiusa.

Teorema 2.3.1 (Stabilità in catena chiusa con orizzonte infinito [3], Teorema 8.1)**.** *Se (A,B) è raggiungibile, allora il controllo ad orizzonte infinito con* $W, U \succ 0$ *porta ad una risposta in catena chiusa convergente all'origine.*

2.3.4 Set-point tracking

Il problema di controllo ottimo descritto finora è un problema di regolarizzazione, tuttavia è possibile riscriverlo sotto forma di set-point tracking.
Ipotizziamo che si voglia portare l'uscita del sistema controllato y ad un valore costante y_{sp} associato ad un punto d'equilibrio del sistema (x_s, u_s). Allora la seguente equazione è verificata

$$\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ y_{sp} \end{bmatrix} \tag{2.37}$$

Se esiste una soluzione possiamo definire un nuovo sistema di riferimento avente come origine il punto di equilibrio desiderato

$$\tilde{x}(k) = x(k) - x_s, \qquad \tilde{u}(k) = u(k) - u_s \tag{2.38}$$

che risolvono

$$\tilde{x}(k+1) = A\tilde{x}(k) + B\tilde{u}(k)$$

In questo modo è possibile risolvere il problema di regolazione nelle nuove variabili introdotte utilizzando i metodi precedentemente descritti ottenendo che l'ingresso ottimo u^* per il problema iniziale è dato da

$$u^* = \tilde{u}^* + u_s \tag{2.39}$$

2.4 Controllo ottimo con vincoli

Consideriamo il problema di regolarizzazione per il sistema n-dimensionale, discreto, tempo-invariante con m ingressi e p uscite descritto dall'equazione in spazio di stato

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ x(k) &= x_k \\ y(k) &= Cx(k) \end{aligned} \tag{2.40}$$

rispetto all'orizzonte temporale $[k, k+L]$. Definiamo la sequenza di controllo $\mathbf{u}_k = [u(k), u(k+1), \ldots, u(k+L-1)]$.
La funzione di costo è

$$J(x(k), \mathbf{u}_k, k) = \sum_{i=0}^{L-1} \left[x^T(k+i)Wx(k+i) + u^T(k+i)Uu(k+i)\right] + x^T(k+L)P_f x(k+L)$$

con $W \succeq 0$, $U \succ 0$, $P_f \succeq 0$.

Il problema di controllo ottimo vincolato a tempo finito è formulato come segue:

$$
\begin{aligned}
J_0(x(k)) = & \min_{\mathbf{u}_k} J(x(k), \mathbf{u}_k) \qquad (2.41)\\
\text{subj. to} \quad & x(k+i+1) = Ax(k+i) + Bu(k+i),\ i = 0, 1, \ldots, L-1\\
& x(k+i) \in \mathcal{X},\ u(k+i) \in \mathcal{U},\ i = 0, 1, \ldots, L-1\\
& x(k+L) \in \mathcal{X}_f\\
& x(k) = x_k
\end{aligned}
$$

dove $\mathcal{X}$,$\mathcal{U}$,$\mathcal{X}_f$ sono regioni che possono essere descritte mediante disuguaglianze lineari (regioni poliedriche).
Definiamo

$$
X(k) = \begin{bmatrix} x(k) \\ x(k+1) \\ \vdots \\ x(k+L-1) \\ x(k+L) \end{bmatrix} \quad U(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+L-1) \\ u(k+L) \end{bmatrix}
$$

$$
G = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{L-1} \\ A^L \end{bmatrix} \quad F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ B & 0 & \cdots & \cdots & \cdots & 0 \\ AB & B & \cdots & \cdots & \cdots & 0 \\ A^2B & AB & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ A^{L-1}B & \cdots & \cdots & \cdots & \cdots & B \end{bmatrix}
$$

$$
\overline{W} = \begin{bmatrix} W & & & 0 \\ & W & & \\ & & \ddots & \\ 0 & & & P_f \end{bmatrix} \quad \overline{U} = \begin{bmatrix} U & & & 0 \\ & U & & \\ & & \ddots & \\ 0 & & & U \end{bmatrix}
$$

2.4.1 Modellizzazione dei vincoli

Gli ingressi di controllo della maggior parte dei sistemi fisici sono limitati. Includiamo questi vincoli sotto forma di diseguaglianze lineari

$$u_{min} \leq u(k) \leq u_{max}$$
$$\Updownarrow$$
$$Eu(k) \leq e, \quad E = \begin{bmatrix} I \\ -I \end{bmatrix} \qquad e = \begin{bmatrix} u_{max} \\ u_{min} \end{bmatrix} \qquad k \geq 0$$

Analogamente è possibile imporre vincoli sui valori delle variabili di stato o d'uscita. E' importante tenere in considerazione il fatto che tipicamente i vincoli sull'ingresso sono limiti fisici (*hard constraints*) che vengono forzati dal sistema reale anche nel caso in cui non vengano rispettati dal controllore, mentre i vincoli su variabili di stato e d'uscita sono tipicamente desiderabili e possono essere "rilassati" (*soft constraints*). Consideriamo separatamente i vincoli su u, x, y ([4], pagine 91-92):

1. **Vincoli sull'ingresso**: $\mathcal{U} = \{u \mid E_u u \leq e_u\}$ possono essere formulati come

$$G_{in} U(k) \leq w_{in} + E_{in}\, x(k)$$

$$G_{in} = \begin{bmatrix} E_u & 0 & \cdots & 0 \\ 0 & E_u & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & E_u \end{bmatrix} \quad E_{in} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad w_{in} = \begin{bmatrix} e_u \\ e_u \\ \vdots \\ e_u \end{bmatrix}$$

2. **Vincoli sullo stato**: $\mathcal{X} = \{x \mid E_x x \leq e_x\},\ \mathcal{X}_f = \{x \mid E_f x \leq e_f\}$ possono essere formulati come

$$G_{stato} U(k) \leq w_{stato} + E_{stato}\, x(k)$$

$$G_{stato} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ E_x B & 0 & \cdots & 0 \\ E_x AB & E_x B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ E_f A^{L-1} B & E_f A^{L-2} B & \cdots & E_f B \end{bmatrix}$$

$$E_{stato} = \begin{bmatrix} -E_x \\ -E_x A \\ -E_x A^2 \\ \vdots \\ -E_f A^L \end{bmatrix} \quad w_{stato} = \begin{bmatrix} e_x \\ e_x \\ e_x \\ \vdots \\ e_f \end{bmatrix}$$

3. **Vincoli sull'uscita**: possono essere riscritti in termini di vincoli sullo stato $\mathcal{X} = \{y \mid E_y y \leq e_y\}$ usando

$$y(k+i) = CA^i x(k) + \sum_{j=0}^{i-1} CA^{i-j-1} Bu(k+j)$$

Vengono formulati come

$$G_{out} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ E_y CB & 0 & \cdots & 0 \\ E_y CAB & E_y CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ E_y CA^{L-1}B & E_y CA^{L-2}B & \cdots & E_y CB \end{bmatrix}$$

$$E_{out} = \begin{bmatrix} -E_y C \\ -E_y CA \\ -E_y CA^2 \\ \vdots \\ -E_y CA^L \end{bmatrix} \quad w_{out} = \begin{bmatrix} e_y \\ e_y \\ e_y \\ \vdots \\ e_f \end{bmatrix}$$

Mettendo assieme si può scrivere in maniera compatta

$$GU(k) \leq w + Ex(k) \tag{2.42}$$

$$G = \begin{bmatrix} G_{in} \\ G_{stato} \\ G_{out} \end{bmatrix} \quad E = \begin{bmatrix} E_{in} \\ E_{stato} \\ E_{out} \end{bmatrix} \quad w = \begin{bmatrix} w_{in} \\ w_{stato} \\ w_{out} \end{bmatrix}$$

Il problema (2.41) può essere riscritto in maniera compatta come

$$
\begin{aligned}
J_0(x(k)) &= \min_{U(k)} J(x(k), U(K)_k, k) \qquad (2.43)\\
&= \min_{U(k)} U^T(k)\, F^T \overline{W} F + \overline{U}\, U(k) + 2x^T(k)\, G^T \overline{W} F^{\,T} U(k) + x^T(k)\, G^T \overline{W} G\, x(k)\\
\text{subj. to} \quad & GU(k) \leq w + Ex(k)
\end{aligned}
$$

che appartiene alla classe dei problemi di ottimizzazione quadratici (*Quadratic Programming QP*) che possono essere risolti efficientemente.

2.5 Receding Horizon control

Nei paragrafi precedenti si è dimostrato come risolvere problemi di controllo ottimo in assenza e presenza di vincoli per sistemi lineari, sia nel caso ad orizzonte finito che ad orizzonte infinito. Si è visto inoltre che il controllore realizzato ad orizzonte infinito offre buone proprietà al sistema in catena chiusa, in particolare se il problema di ottimizzazione è feasibible ed ha soluzione finita allora le variabili di stato e d'ingresso convergeranno all'origine in catena chiusa. Un controllore sub-ottimo può essere progettato usando un'approccio chiamato *Receding Horizon Control*: ad ogni istante di campionamento, partendo dallo stato corrente, si risolve un problema di ottimizzazione in catena aperta su un orizzonte finito e viene applicato l'ingresso calcolato solo durante l'intervallo di campionamento $[k, k+1]$. Al successivo istante di campionamento $k+1$ viene risolto un nuovo problema di controllo ottimo basato sulle nuove misure dello stato su un orizzonte traslato (*shifted horizon*).
Il controllore risultate viene chiamato *Receding Horizon Controller* (RHC), e quando la legge di controllo ottima viene calcolata risolvendo il problema di ottimizzazione in real-time si parla di *Model Predictive Control* (MPC).

Algoritmo 1 Model-Predictive-Control lineare

Input: *modello* A, B, C, x_0 *vincoli* G, E, w *pesi* W, U, P_f, time horizon L

1: Misura lo stato $x(k)$
2: Risolvi (2.43)
3: Applica $\bar{u}_0^*$
4: $k \leftarrow k+1$ e torna ad 1)

Lo scopo di questo controllore è quello di portare ad un comportamento in catena chiusa simile a quello ottenibile usando un orizzonte infinito. Purtroppo però si possono presentare due principali problemi:

- il controllore potrebbe portare ad una situazione in cui dopo alcuni step il problema di controllo ad orizzonte finito diventa infeasible;
- anche se non vi è il problema della feasibility, la sequenza di controllo calcolata potrebbe non portare le traiettorie a convergere all'origine.

In generale quindi la feasibility e la stabilità non sono garantite dall'approccio RHC. Vengono quindi derivate delle condizioni sui pesi P_f e l'insieme dei vincoli terminali $\mathcal{X}_f$ affinchè queste due proprietà siano garantite. Per ottenere questi risultati è necessario introdurre alcune nozioni presenti nella sottosezione (2.5.1)

2.5.1 Controllability, reachability and invariance

Consideriamo il sistema dinamico vincolato

$$\begin{aligned} &x(k+1) = f(x(k), u(k)), \\ &x(k) \in \mathcal{X},\ u(k) \in \mathcal{U} \quad \forall k \geq 0 \end{aligned} \tag{2.44}$$

con $\mathcal{X}$,$\mathcal{U}$ poliedri (vincoli lineari).
Siamo interessati a rispondere a due tipi di domande:

1. per una dato controllore in feedback $u = g(x)$, trovare l'insieme degli stati iniziali per cui la traiettoria del sistema non viola i vincoli imposti;
2. trovare l'insieme delle condizioni iniziali per cui esiste un controllore tale da non violare i vincoli del sistema.

Al fine di rispondere a queste domande introduciamo alcune definizioni.

Definizione 2.5.1. Per il sistema (2.44), *l'insieme controllabile ad un passo* dato l'insieme $\mathcal{S}$ è

$$Pre(\mathcal{S}) = \{x \in \mathbb{R}^n \mid \exists u \in \mathcal{U} \text{ tale che } f(x,u) \in \mathcal{S}\}.$$

$Pre(\mathcal{S})$ è l'insieme degli stati che possono essere portati nell'insieme target $\mathcal{S}$ in un passo soddisfando i vincoli su input e stato.

Definizione 2.5.2. Per il sistema (2.44), *l'insieme raggiungibile ad un passo* dato l'insieme $\mathcal{S}$ è

$$Reach(\mathcal{S}) = \{x \in \mathbb{R}^n \mid \exists u(0) \in \mathcal{U} \text{ tale che } x = f(x(0), u(0))\}.$$

$Reach(\mathcal{S})$ rappresenta l'insieme degli stati che possono essere raggiunti in un passo partendo all'interno dell'insieme $\mathcal{S}$.

Definizione 2.5.3. Per il sistema (2.44), *l'insieme controllabile ad N-passi* $\mathcal{K}_N(\mathcal{S})$ dato l'insieme $\mathcal{S} \subseteq \mathcal{X}$ è definito ricorsivamente come:

$$\mathcal{K}_j(\mathcal{S}) = Pre(\mathcal{K}_{j-1}(\mathcal{S})) \cap \mathcal{X}, \quad \mathcal{K}_0(\mathcal{S}) = \mathcal{S}, \quad j = 1, \ldots, N$$

Tutti gli stati $x_0 \in \mathcal{K}_N(\mathcal{S})$ possono essere portati, tramite un'opportuna sequenza di controllo, nel set target $\mathcal{S}$ in N passi, soddisfando contemporaneamente i vincoli del sistema.

Definizione 2.5.4. Per il sistema (2.44), *l'insieme massimo controllabile* $\mathcal{K}_\infty(\mathcal{S})$ dato l'insieme $\mathcal{S} \subseteq \mathcal{X}$ è l'unione di tutti gli insiemi controllabili a N-passi contenuti in $\mathcal{X}, N \in \mathbb{N}$.

Definizione 2.5.5. Per il sistema (2.44), *l'insieme raggiungibile ad N-passi* $\mathcal{R}_N(\mathcal{X}_0)$ dato l'insieme $\mathcal{X}_0 \subseteq \mathcal{X}$ è definito ricorsivamente come:

$$\mathcal{R}_{i+1}(\mathcal{X}_0) = Reach(\mathcal{R}_i(\mathcal{X}_0)), \quad \mathcal{R}_0(\mathcal{X}_0) = \mathcal{X}_0, \quad i = 0, \ldots, N-1$$

$\mathcal{R}_N(\mathcal{X}_0)$ rappresenta l'insieme degli stati che posso raggiungere in N-step partendo da $x_0 \in \mathcal{X}_0$

Definizione 2.5.6. Per il sistema (2.44), *l'insieme massimo raggiugibile* $\mathcal{R}_\infty(\mathcal{X}_0)$ dato l'insieme $\mathcal{X}_0 \subseteq \mathcal{X}$ è l'unione di tutti gli insiemi raggiungibili a N-passi contenuti in $\mathcal{X}, N \in \mathbb{N}$.

Gli insiemi controllabili e raggiungibili per sistemi lineari vincolati sono facilmente calcolabili poichè $Pre(\mathcal{X}), Reach(\mathcal{X})$ sono il risultato di operazioni lineari sui poliedri $\mathcal{X}$,$\mathcal{U}$ e sono quindi a loro volta poliedri.
Consideriamo ora il sistema lineari autonomo vincolato

$$\begin{aligned} &x(k+1) = f(x(k)), \\ &x(k) \in \mathcal{X}, \quad \forall k \geq 0 \end{aligned} \tag{2.45}$$

Definizione 2.5.7. Un insieme $\mathcal{O} \subseteq \mathcal{X}$ è detto *insieme positivo invariante* per il sistema (2.45) se

$$x(0) \in \mathcal{O} \quad \Rightarrow \quad x(k) \in \mathcal{O}, \quad \forall k \in \mathbb{N}_+.$$

Definizione 2.5.8. Un insieme $\mathcal{O}_\infty \subseteq \mathcal{X}$ è detto *insieme massimo positivo invariante* per il sistema (2.45) se $\mathcal{O}_\infty$ è positivo invariante e contiene tutti gli insiemi positivo invarianti contenuti in $\mathcal{X}$

Questi tipi di insieme sono utili per rispondere alla prima domanda, ovvero dato un controllore trovare l'insieme degli stati iniziali per cui la traiettoria non viola i vincoli di sistema.

Teorema 2.5.9. *Un insieme $\mathcal{O}$ è positivo invariante se e solo se*

$$\mathcal{O} \subseteq Pre(\mathcal{O}) \iff Pre(\mathcal{O}) \cap \mathcal{O} = \mathcal{O}$$

Questo teorema suggerisce un algoritmo per calcolare $\mathcal{O}_\infty$

Algoritmo 2 Calcolo di $\mathcal{O}_\infty$

Input: $f, \mathcal{X}$

Output: $\mathcal{O}_\infty$

1: $\Omega_0 \leftarrow \mathcal{X}, \quad k \leftarrow 0$
2: $\Omega_1 \leftarrow Pre(\Omega_0) \cap \Omega_0$
3: **while** $\Omega_{k+1} \neq \Omega_k$ **do**
4: $\quad k \leftarrow k + 1$
5: $\quad \Omega_{k+1} \leftarrow Pre(\Omega_k) \cap \Omega_k$
6: **end while**
7: $\mathcal{O}_\infty \leftarrow \Omega_k$

Definizione 2.5.10. Un insieme $\mathcal{C} \subseteq \mathcal{X}$ è detto *controllo-invariante* per il sistema (2.44) se

$$x(k) \in \mathcal{C} \Rightarrow \exists\, u(k) \in \mathcal{U} \text{ tale che } f(x(k), u(k) \in \mathcal{C}, \quad \forall k \in \mathbb{N}_+.$$

Definizione 2.5.11. L'insieme $\mathcal{C}_\infty$ è detto *insieme massimo controllo-invariante* se è controllo-invariante e contiene tutti gli insieme controllo-invarianti contenuti in $\mathcal{X}$.

Questi insiemi rispondono alla seconda domanda posta, cioè contengono gli stati iniziali per cui esiste un controllore che non viola i vincoli di sistema.

Teorema 2.5.12. *Un insieme $\mathcal{C}$ è controllo invariante per* (2.44) *se e solo se*

$$\mathcal{C} \subseteq Pre(\mathcal{C}) \iff Pre(\mathcal{C}) \cap \mathcal{C} = \mathcal{C}$$

Analogamente a prima è possibile calcolare $\mathcal{C}_\infty$ usando questo algoritmo

Algoritmo 3 Calcolo di $\mathcal{C}_\infty$

Input: $f, \mathcal{X}, \mathcal{U}$

Output: $\mathcal{C}_\infty$

1: $\Omega_0 \leftarrow \mathcal{X}, \quad k \leftarrow 0$
2: $\Omega_1 \leftarrow Pre(\Omega_0) \cap \Omega_0$
3: **while** $\Omega_{k+1} \neq \Omega_k$ **do**
4: $\quad k \leftarrow k + 1$
5: $\quad \Omega_{k+1} \leftarrow Pre(\Omega_k) \cap \Omega_k$
6: **end while**
7: $\mathcal{C}_\infty \leftarrow \Omega_k$

2.5.2 Feasibility

Consideriamo ora il problema della feasibility del controllore MPC.
Il sistema in catena chiusa è dato da

$$\begin{aligned} &x(k+1) = Ax(k) + Bu_o(x(k)) = f(x(k)) \\ &u(k) \in \mathcal{U},\ x(k) \in \mathcal{X},\ k \geq 0 \end{aligned} \tag{2.46}$$

E' facile trovare esempio per i quali la feasibility al tempo iniziale $x(0) \in \mathcal{X}_0$ non implica la feasibility per i tempi futuri. E' invece necessario poter garantire la feasibility ad ogni istante, chiamata *recursive feasibility*. L'obiettivo è quello di mostrare come questa proprietà sia influenzata dalla formulazione del problema di controllo e dalla scelta dei parametri ([3], Lemma 12.1).

Teorema 2.5.13. *Sia $\mathcal{O}_\infty$ l'insieme massimo positivo invariante per il sistema in catena chiusa* (2.46)*. Allora il controllore MPC è recursively feasible se e solo se $\mathcal{O}_\infty = \mathcal{X}_0$.*

Si ha che $\mathcal{X}_0$ dipende solamente da $\mathcal{X}, \mathcal{U}$, dall'orizzonte temporale L e i vincoli terminali $\mathcal{X}_f$. Tuttavia $\mathcal{O}_\infty$ dipende anche dai parametri della funzione di costo

e quindi il Teorema 2.5.13 mostra che solo alcune scelte di questi parametri sono accettabili.
Sono presenti in letteratura numerosi teoremi che dimostrano come varia $\mathcal{X}_0$ in base alla scelta dei vincoli terminali e di N, e di conseguenza quali sono alcune condizioni sufficienti sui parametri che portano a garantire la proprietà di recursive feasibility.
L'aspetto fondamentale che si vuole mettere in luce è il fatto che esistono condizioni teoriche sui parametri di progetto del controllore che portano a garantire la risolubilità del problema di ottimizzazione in catena chiusa, andando ad impattare sulle performance. E' quindi necessario progettare il controllore in modo da garantire le proprietà richieste ottenendo il massimo delle performance disponibili.

2.5.3 Stabilità

Persistent feasibility non garantisce che le traiettorie in catena chiusa convergano al punto di equilibrio imposto. Un approccio molto popolare per garantire anche la stabilità in catena chiusa è l'utilizzo di un insieme di vincoli terminali $\mathcal{X}_f$ controllo-invariante e di un costo terminale P_f.

Teorema 2.5.14 (Stabilità controllore MPC vedi [3], Teorema 12.2)**.** *Siano*

H1. $U, W, P_f \succ 0$

H2. $\mathcal{X}, \mathcal{X}_f, \mathcal{U}$ *insiemi chiusi che contengono interiormente l'origine*

H3. $\mathcal{X}_f \subseteq \mathcal{X}$ *insieme controllo-invariante*

H4.

$$\min_{u \in \mathcal{U}, Ax+Bu \in \mathcal{X}_f} \Big(-x^T P_f x + \left[x^T W x + u^T U u \right] + (Ax + Bu)^T P_f (Ax + Bu) \Big) \leq 0 \quad \forall\, x \in \mathcal{X}_f$$

Allora

1. *lo stato del sistema in catena chiusa converge all'origine*

$$\lim_{k \to \infty} x(k) = 0;$$

2. *l'origine del sistema in catena chiusa è asintoticamente stabile*

La dimostrazione di questo risultato si basa sul provare che la funzione di costo del problema di controllo ottimo è una funzione di Lyapunov per il sistema in catena chiusa ([3], pagina 254).

2.5.4 Esempio di implementazione in Matlab

Consideriamo il sistema instabile

$$x(k+1) = \begin{bmatrix} 2 & 1 \\ 0 & 0.5 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k)$$

con ingresso limitato

$$-1 \leq u(k) \leq 1, \quad k = 0, \ldots, L-1$$

e vincoli sullo stato

$$\begin{bmatrix} -10 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \quad k = 0, \ldots, L-1$$

con funzione di costo

$$J(x_o, \mathbf{u}) = x_L^T P_f x_L + \sum_{k=0}^{L-1} (x_k^T W x_k + u_k^T U u_k)$$

Consideriamo il controllore RHC omettendo i vincoli terminali $\mathcal{X}_f = \mathbb{R}^2$ e il costo terminale $P_f = 0$, con $Q = I$ (questo viene fatto per semplicità di realizzazione, non rispettando le condizioni sufficienti che garantiscono la feasibility e stabilità per ogni condizione iniziale). Viene implementata un funzione *optimization* che risolve il problema di ottimizzazione (2.43).

```
function [u_0,x1,x2] = optimization(x1_0,x2_0,L,W,P,U)
    problem = optimproblem;
    %bound sulle variabili di stato
    x_variable1 = optimvar('x_variable1',1,L,'LowerBound',-10,'
     UpperBound',10);
    x_variable2 = optimvar('x_variable2',1,L,'LowerBound',-10,'
     UpperBound',10);
    x_variable = [x_variable1' x_variable2']';
    %bound sugli ingressi
    u = optimvar('u',1,L, 'LowerBound',-1,'UpperBound',1);
    %vincoli dati dalla dinamica del sistema
```

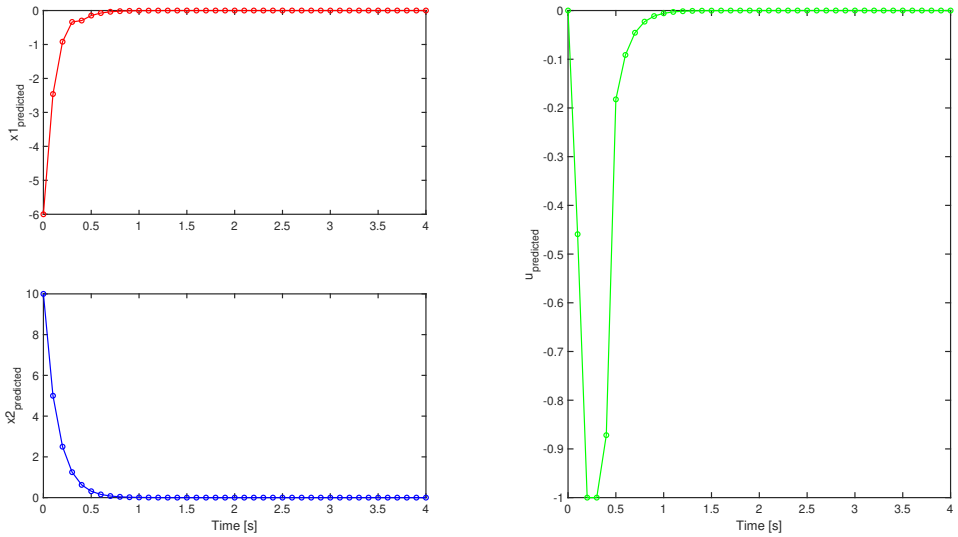
```
    constraints = optimconstr(2*L);
    constraints(1) = x_variable(1,1) == 2*x1_0 + x2_0  + u(1,1);
    constraints(2) = x_variable(2,1) == 0.5*x2_0;
    for k=1:L-1
        constraints(2*k+1) = x_variable(1,k+1) == 2*x_variable(1,
     k) + x_variable(2,k) + u(1,k+1);
        constraints(2*k+2) = x_variable(2,k+1) == 0.5*x_variable
     (2,k);
    end
    problem.Constraints.constraints = constraints;
    %funzione di costo quadratica
    cost = 0;
    for k = 1:L-1
        cost = cost + x_variable(:,k)'*W*x_variable(:,k) + u(1,k
     +1)'*U*u(1,k+1);
    end
    cost = cost + x_variable(:,L)'*P*x_variable(:,L);
    problem.Objective = cost;
    solution = solve(problem);
    u_0 = solution.u(1); %ingresso ottimo
    x1 = solution.x_variable1(1); %componente dello stato
     predetta x1_{k+1}
    x2 = solution.x_variable2(1); %componente dello stato
     predetta x2_{k+1}
end
```

Viene poi realizzato il controllore Receding

```
%% definizione parametri del controllore

Q = eye(2);
P = zeros(2,2);
R = 1;
L = 40; %time horizon

%% simulazione

x1_0 = -6; %condizioni iniziali
x2_0 = 10;

%sequenze predizioni
x1 = x1_0;
x2 = x2_0;
u = 0;
```

```
T = 0.1; %tempo di campionamento
t = 0:T:4; %tempo simulazione

for i = 1:length(t)-1
    [u_0,x1_0,x2_0] = optimization(x1_0,x2_0,L,Q,P,R);
    x1 = [x1 x1_0];
    x2 = [x2 x2_0];
    u = [u u_0];
end

%% visualizzazione risultati

figure;
subplot(211); plot(t,x1,'r-o');
ylabel('x1_{predicted}');
subplot(212);plot(t,x2,'b-o');
xlabel('Time [s]'); ylabel('x2_{predicted}');
figure;
plot(t,u,'k-o');
xlabel('Time [s]'); ylabel('u_{predicted}');
figure;
plot(x1,x2,'k',x1(1),x2(1),'o');
xlim([-8 8]); ylim([-10 10]);
grid on;
legend('traiettoria in closed-loop','condizione iniziale','
    Location','southeast');
```

Vengono mostrati i risultati della simulazione in Figura 2.1-2.2 dai quali si può vedere che il controllore porta la traiettoria in catena chiusa all'origine.



(a) *Variabili di stato* **(b)** *Input*

Figura 2.1: Simulazione controllore MPC

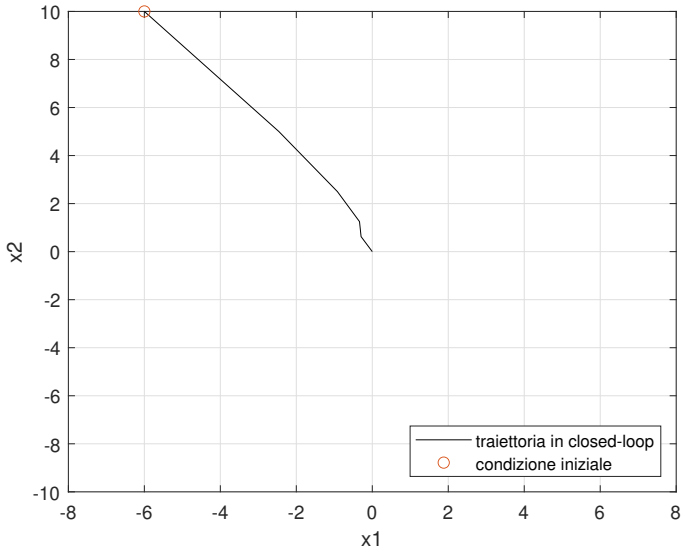


Figura 2.2: Traiettoria in catena chiusa

Capitolo 3

Descrizione data-driven di un sistema LTI

Nel capitolo (2) si è mostrato come è possibile controllare sistemi vincolati usando la tecnica del Model-Predictive-Control che richiede di risolvere ciclicamente un problema di controllo ottimo in catena aperta.
Uno degli elementi fondamentali per poter implementare questo approccio è la formulazione di un modello del sistema da controllare.
Consideriamo un sistema $\mathcal{G}$ con le seguenti caratteristiche:

(i) lineare tempo-invariante (LTI)

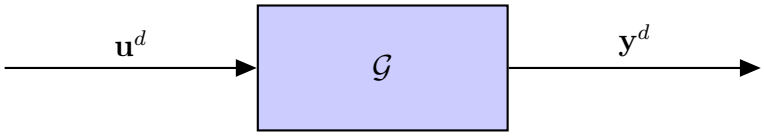
(ii) ordine n

(iii) controllabile

avente il seguente modello di stato

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned} \tag{3.1}$$

Tramite simulazioni o esperimenti sul sistema fisico è possibile ottenere dati sulle traiettorie del sistema $\{u_k^d, y_k^d\}_{k=0}^{N-1}$, con N rappresentante il numero di coppie ingresso-uscita che si hanno a disposizione.



Un'approccio classico indiretto consiste nell'usare questi dati per identificare un modello del sistema da poter poi utilizzare ad esempio all'interno del framework dell'MPC.
Il problema che ci pone qui è invece quello di rispondere a questa domanda:

> **Possiamo usare i dati $\{\mathbf{u_k^d}, \mathbf{y_k^d}\}_{\mathbf{k=0}}^{\mathbf{N-1}}$ per descrivere il sistema $\mathcal{G}$ senza identificare un modello del sistema?**

La risposta viene data dal cosidetto *Fundamental Lemma* [1] che dimostra che, sotto alcune ipotesi sull'ingresso del sistema (*"persistently exciting input"*), l'insieme delle traiettorie di lunghezza finita di un sistema LTI è dato dallo span di una speciale matrice ottenuta da una singola traiettoria nota del sistema.

3.1 Persistently exciting input

Al fine di poter determinare quando è possibile descrivere il comportamente del sistema usando solo una traiettoria nota, è necessario caratterizzare quando una sequenza di dati è informativa.
Non è possibile ad esempio ottenere alcuna informazione dalla traiettoria misurata del sistema quando quest'ultimo non è sollecitato (i.e. $u = 0$) poiché si sa già a priori in questo caso che qualunque sistema LTI risponderà con uscita nulla.
Assumiamo che sia nota la risposta $\{y_k\}_{k=0}^{N-1}$, $y_k \in \mathbb{R}^p$ di $\mathcal{G}$ di lunghezza N e consideriamo per un qualche L, $1 \leq L \leq N$ le finestre di lunghezza L:

$$
\begin{aligned}
&[y_0, y_1, \dots, y_{L-1}] \\
&[y_1, y_2, \dots, y_L] \\
&\dots \\
&[y_{N-L}, y_{N-L+1}, \dots, y_{N-1}]
\end{aligned}
$$

Sotto quali condizioni è possibile ottenere tutte le traiettorie del sistema a partire da queste finestre di lunghezza L?
Definiamo la *matrice di Hankel* associata

$$
H_L(y) = \begin{bmatrix} y_0 & y_1 & \cdots & y_{N-L} \\ y_1 & y_2 & \cdots & y_{N-L+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{L-1} & y_L & \cdots & y_{N-1} \end{bmatrix}
$$

nella quale le diagonali a pendenza positiva sono costanti e le colonne sono date dalle 'finestre' di lunghezza L. Equivalentemente si può definire la matrice di Hankel associata all'ingresso u

$$H_L(u) = \begin{bmatrix} u_0 & u_1 & \dots & u_{N-L} \\ u_1 & u_2 & \dots & u_{N-L+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{L-1} & u_L & \dots & u_{N-1} \end{bmatrix}$$

Definizione 3.1.1 (**Persistently exciting input**)**.** Il segnale d'ingresso $u \in \mathbb{R}^m$ è detto *persistently exciting di ordine* L se $rank(H_L(u)) = mL$

Viene fatto notare che avere un'ingresso persistently exciting di ordine L impone un lower bound sulla lunghezza dei dati a disposizione N

$$N - L + 1 \geq mL \quad \iff \quad N \geq (m+1)L - 1 \tag{3.2}$$

3.2 Fundamental Lemma

Sotto le ipotesi sul sistema $\mathcal{G}$ (3.1), in assenza di rumore si perviene al seguente risultato fondamentale per la descrizione data-driven di un sistema.

Teorema 3.2.1 (**Fundamental Lemma**)**.** *Sia $\mathcal{G}$ un sistema LTI di ordine n controllabile in condizioni nominali di assenza di disturbi e $\{u_k^d, y_k^d\}_{k=0}^{N-1}$ una traiettoria input-output con $u^d \in \mathbb{R}^m$ persistently exciting di ordine $L + n$. Allora $\{\bar{u}_k, \bar{y}_k\}_{k=0}^{N-1}$ è una traiettoria di $\mathcal{G}$ se e solo se esiste $\alpha \in \mathbb{R}^{N-L+1}$ tale che*

$$\begin{bmatrix} H_L(u^d) \\ H_L(y^d) \end{bmatrix} \alpha = \begin{bmatrix} \bar{u} \\ \bar{y} \end{bmatrix} \tag{3.3}$$

Questo risultato ci dice che usando una singola traiettoria (u^d, y^d) in catena aperta possiamo descrivere tutte le traiettorie di un sistema non noto e ci permette di ottenere una descrizione puramente data-driven del sistema senza alcuna informazione sul modello del sistema.
In particolare

> Qualsiasi approccio di controllo model-based di analisi o sintesi può essere riformulato in un contesto data-driven tramite (3.3)

Viene fatto notare che per verificare la condizione su u^d è necessario conoscere l'ordine del sistema n ma il risultato del teorema rimane vero anche se n viene sostituito con un upper bound. Infatti, se un segnale è persistently exciting di ordine di ordine L allora è anche persistently exciting di ordine $\tilde{L}$ per ogni $\tilde{L} \leq L$.

Capitolo 4

Data-driven predictive control

Sostituendo la rappresentazione in spazio di stato classica con la corrispondente parametrizzazione basata sui dati tramite (3.3) è possibile implementare schemi di controllo MPC che non usano conoscenze a priori sul sistema. Questo tipo di framework è stato recentemente sviluppato ed è stato applicato con successo per risolvere problemi reali. Tuttavia, per applicazioni complesse o applicazioni in cui la sicurezza gioca un ruolo chiave, è necessario garantire alcune proprietà fondamentali sul sistema in catena chiusa, come ad esempio la stabilità, ammettendo anche la possibilità realistica che i dati a disposizione siano affetti da rumore.
Nel seguente capitolo vengono presentati gli schemi di controllo data-driven MPC per sistemi LTI in condizioni nominali e per sistemi LTI in presenza di rumore, discutendo alcune estensioni possibili per sistemi non lineari e mostrando un esempio di possibile implementazione in Matlab.

4.1 Data-driven MPC per sistemi lineari in assenza di rumore

Consideriamo il sistema LTI con modello di stato

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned} \tag{4.1}$$

con $x_k \in \mathbb{R}^n,\ u_k \in \mathbb{R}^m,\ y_k \in \mathbb{R}^p$ e le matrici A, B, C, D non note.
Assumiamo che (A, B) si controllabile e (A, C) sia osservabile, che sia noto un upper bound dell'ordine n del sistema e che si abbia a disposizione una misura di una traiettoria input-output $\{u_k^d, y_k^d\}_{k=0}^{N-1}$, con u^d persistently exciting.

Lo scopo di controllo è quello di

- stabilizzare il punto di equilibrio (u^s, y^s);
- rispettare i vincoli su ingresso e uscita $u_k \in \mathcal{U},\ y_k \in \mathcal{Y}$.

Poiché non è disponibile un modello del sistema definiamo un punto di equilibrio tramite coppie input-output.

Definizione 4.1.1. Diciamo che una coppia input-output $(u^s, y^s) \in \mathbb{R}^{m+p}$ è un equilibrio di un sistema LTI $\mathcal{G}$ se la sequenza $\{\bar{u}_k, \bar{y}_k\}_{k=0}^{n}$ con $(\bar{u}_k, \bar{y}_k) = (u^s, y^s)$ per ogni $k \geq 0$ è una traiettoria di $\mathcal{G}$.

Sia x^s lo stato corrispondente a (u^s, y^s). L'obiettivo di rendere x^s asintoticamente stabile richiede la convergenza locale a x^s, tuttavia nelle applicazioni è spesso necessario disporre di una stima del tempo necessario perché lo stato perturbato ritorni in x_s. Si introduce quindi il seguente concetto di stabilità:

Definizione 4.1.2. Un punto di equilibrio x^s si dice *esponenzialmente stabile* se esistono costanti positive α, λ e c tali che:

$$|x(t) - x^s| \leq \alpha\, |x_0 - x^s|\, \mathrm{e}^{-\lambda t}, \quad \forall\, t > 0, \quad \forall\, |x_0 - x^s| \leq c$$

In pratica, si richieste che esista un intorno di x^s a partire dal quale la traiettoria perturbata converge a x^s con velocità almeno esponenziale. Inoltre la stabilità esponenziale implica la stabilità asintotica e non viceversa.
Nel caso data-driven supponiamo di avere a disposizione solo misurazioni sull'uscita senza necessità di avere osservatori dello stato. La funzione di costo viene quindi riformulata in termini dell'uscita e non dello stato.
Data una matrice definita positiva $P = P^T \succ 0$ definiamo $\|x\|_P^2 \coloneqq x^T P x$.
Consideriamo la seguente funzione di costo quadratica:

$$J = \sum_{k=0}^{L-1} \|\bar{u}_k - u^s\|_R^2 + \|\bar{y}_k - y^s\|_Q^2 \tag{4.2}$$

con $Q, R \succ 0$ pesi che possono essere scelti dal progettista.
Per definire il problema di controllo ottimale è necessario disporre delle condizioni iniziali al tempo t per predirre le traiettorie future. Poiché supponiamo di avere solo dati input-output e nessuna misura dello stato disponibile, usiamo le ultime n misure su input-output $\{u_k^d, y_k^d\}_{k=t-n}^{t-1}$ per determinare univocamente una

traiettoria di sistema. A causa di queste condizioni iniziali l'orizzonte temporale di predizione ha lunghezza $L + n$.
E' inoltre noto dalla teoria che è opportuno imporre dei vincoli terminali poiché un approccio MPC che non li utilizza richiede un orizzonte temporale sufficientemente grande altrimenti la sua applicazione potrebbe portare a destabilizzare anche un sistema stabile in catena aperta.
Il problema di controllo ottimo riscritto in una contesto data-driven tramite (3.3) è quindi:

$$\min_{\alpha(t),\bar{u}(t),\bar{y}(t)} \quad \sum_{k=0}^{L-1} \|\bar{u}_k - u^s\|_R^2 + \|\bar{y}_k - y^s\|_Q^2 \tag{4.3a}$$

$$\text{subj. to} \quad \begin{bmatrix} \bar{u}_{[-n,L-1]}(t) \\ \bar{y}_{[-n,L-1]}(t) \end{bmatrix} = \begin{bmatrix} H_{L+n}(u^d) \\ H_{L+n}(y^d) \end{bmatrix} \alpha(t) \tag{4.3b}$$

$$\begin{bmatrix} \bar{u}_{[-n,-1]}(t) \\ \bar{y}_{[-n,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-n,t-1]} \\ y_{[t-n,t-1]} \end{bmatrix} \tag{4.3c}$$

$$\begin{bmatrix} \bar{u}_{[L-n,L-1]}(t) \\ \bar{y}_{[L-n,L-1]}(t) \end{bmatrix} = \begin{bmatrix} u_n^s \\ y_n^s \end{bmatrix} \tag{4.3d}$$

$$\bar{u}_k(t) \in \mathcal{U}, \quad \bar{y}_k(t) \in \mathcal{Y}, \quad k = 0, \dots, L-1 \tag{4.3e}$$

dove (4.3b) rappresenta la parametrizzazione di tutte le traiettorie del sistema tramite il Fundamental Lemma, (4.3c) rappresenta l'inizializzazione delle predizioni fino al tempo $t-1$ poiché l'input all'istante t potrebbe già influenzare l'output al tempo t e (4.3d) rappresentano vincoli terminali imposti sulle ultime n coppie input-output predette per poter garantire la stabilità in catena chiusa (u_n^s, y_n^s sono vettori colonni che contengono n volte rispettivamente u_s, y_s).
Nel caso di vincoli lineari ($\mathcal{U}, \mathcal{Y}$ poliedri) (4.3) è un classico problema di quadratic programming che può essere risolto efficientemente. L'applicazione di (4.3) in un contesto MPC è data dal seguente algoritmo:

Algoritmo 4 Data-Driven MPC per sistemi lineari in assenza di rumore

Input: time horizon L, matrici di costo $Q, R \succ 0$, vincoli $\mathcal{U}, \mathcal{Y}$, setpoint (u^s, y^s), dati $\{u_k^d, y_k^d\}_{k=0}^{N-1}$

1: Al tempo t, prendi le ultime n misurazioni $\{u_k^d, y_k^d\}_{k=t-n}^{t-1}$ e risolvi (4.3)

2: Applica l'ingresso $u_t = u_0^*(t)$

3: $t \leftarrow t+1$ e torna a 1)

Quando il precedente algoritmo viene applicato al sistema (4.1) si possono dimostrare le seguenti caratteristiche in catena chiusa ([2], Teorema 2).

Teorema 4.1.3. *Sia $L \geq n$, u^d persistently exciting di ordine $L+2n$ e il costo ottimale di* (4.3) *limitato superiormente da $c_u\|x_t - x^s\|_2^2$ per un qualche $c_u > 0$. Se il problema* (4.3) *è feasible per $t = 0$ allora*

- *è feasible per ogni t*
- *la risposta in catena chiusa soddisfa i vincoli di sistema $u_t \in \mathcal{U},\ y_t \in \mathcal{Y}$ per ogni t*
- *x^s è un punto di equilibrio esponenzialmente stabile per il risultante sistema in catena chiusa*

La dimostrazione del teorema ([5], pagina 4) è simile alla dimostrazione per il model-based MPC con l'ulteriori complicazioni. Questo approccio si caratterizza per la semplicità di applicazione in quanto usa una singola traiettoria misurata mentre nel caso model-based sono necessarie informazioni a priori sul sistema e step di identificazione.
La complessità di (4.3) è comparabile alla complessità degli schemi classici model-based MPC. In particolare si può riscrivere il problema in forma compatta considerando come unica variabile $\alpha(t) \in \mathbb{R}^{N-L-n+1}$, mentre invece il problema di ottimizzazione model-based ha mL variabili di decisione. Essendo u^d persistently exciting di ordine $L+2n$, da (3.2) deve essere $N - L - 2n + 1 \geq m(L+2n)$ e quindi sembrerebbe che l'approccio data-driven abbia una complessità maggiore. Viene mostrato però in [7], pagina 8-10, che, sfruttando la decomposizione LQ della matrice di Hankel, il vettore α può essere scomposto in tre contributi dei quali due vengono fissati dalle condizioni iniziali e terminali. L'ottimizzazione data-driven viene quindi fatta sullo stesso numero di variabili dell'approccio model-based, portando alla stessa complessità computazione, senza richiedere misurazione dello stato né informazioni sul sistema/step di identificazione.

4.2 Robust data-driven MPC per sistemi lineari

Cosa succede se non è possibile misurare direttamente (u, y)?
Consideriamo uno scenario più realistico nel quale è presente un rumore di misura limitato sull'output

- sui dati iniziali: $\tilde{y}_s^d = y_k^d + \varepsilon_k^d$ con $\left\|\varepsilon_k^d\right\|_\infty \leq \bar{\varepsilon}$
- sui dati online: $\tilde{y}_s = y_k + \varepsilon_k$ con $\|\varepsilon_k\|_\infty \leq \bar{\varepsilon}$

Il Fundamental Lemma (3.3) è applicabile solo nel caso di assenza di rumore. Di conseguenza l'immagine dello stack delle matrici di Hankel dei segnali non genera esattamente lo spazio delle traiettorie e quindi le traiettorie di output non possono essere predette accuratamente. Vegono quindi introdotte le seguenti modifiche:

1. viene aggiunta una "*slack variable*" σ per rendere l'equazione feasible:

$$\begin{bmatrix} \bar{u} \\ \bar{y} + \sigma \end{bmatrix} = \begin{bmatrix} H_{L+n}(u^d) \\ H_{L+n}(\tilde{y}^k) \end{bmatrix} \alpha$$

2. viene aggiunto un termine di regolarizzazione λ_σ di σ nella funzione di costo per evitare che σ sia troppo grande: $+\lambda_\sigma \|\sigma\|_2^2,\ \lambda_\sigma > 0$

3. viene aggiunto un termine di regolarizzazione λ_α di α nella funzione di costo per ridurre l'influenza del rumore sull'accuratezza dei risultati: $+\lambda_\alpha \|\alpha\|_2^2$, $\lambda_\alpha > 0$

Queste modifiche portano al seguente problema di ottimizzazione data-driven MPC:

$$\min_{\substack{\alpha(t),\sigma(t) \\ \bar{u}(t),\bar{y}(t)}} \quad \sum_{k=0}^{L-1} \|\bar{u}_k - u^s\|_R^2 + \|\bar{y}_k - y^s\|_Q^2 + \lambda_\sigma \|\sigma(t)\|_2^2 + \lambda_\alpha \|\alpha(t)\|_2^2 \tag{4.4a}$$

$$\text{subj. to} \quad \begin{bmatrix} \bar{u}(t) \\ \bar{y}(t) + \sigma(t) \end{bmatrix} = \begin{bmatrix} H_{L+n}(u^d) \\ H_{L+n}(\tilde{y}^d) \end{bmatrix} \alpha(t) \tag{4.4b}$$

$$\begin{bmatrix} \bar{u}_{[-n,-1]}(t) \\ \bar{y}_{[-n,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-n,t-1]} \\ \tilde{y}_{[t-n,t-1]} \end{bmatrix} \tag{4.4c}$$

$$\begin{bmatrix} \bar{u}_{[L-n,L-1]}(t) \\ \bar{y}_{[L-n,L-1]}(t) \end{bmatrix} = \begin{bmatrix} u_n^s \\ y_n^s \end{bmatrix}, \quad \bar{u}_k(t) \in \mathcal{U} \tag{4.4d}$$

Per semplicità non sono stati considerati vincoli sull'uscita, $\mathcal{Y} = \mathbb{R}^p$, anche se è possibile estendere i risultati anche in questo caso. Viene studiato in questo caso uno schema MPC a n-step poiché si può dimostrare che offre proprietà teoriche migliori rispetto allo schema ad 1-step. Dopo aver risolto online il problema (4.4), vengono applicati al sistema i primi n input della sequenza calcolata e l'orizzonte temporale viene traslato di n step. L'algoritmo che deriva ([5], pagina 6) da questo idea è il seguente:

Algoritmo 5 Robust Data-Driven MPC

Input: : time horizon L, matrici di costo $Q, R \succ 0$, vincoli $\mathcal{U}$, setpoint (u^s, y^s), dati $\{u_k^d, \tilde{y}_k^d\}_{k=0}^{N-1}$, upper bound sull'ordine di sistema n, parametri di regolarizzazione $\lambda_\alpha, \lambda_\sigma > 0$, bound sul rumore $\bar{\varepsilon} > 0$

1: Al tempo t, prendi le ultime n misurazioni $\{u_k^d, \tilde{y}_k^d\}_{k=t-n}^{t-1}$ e risolvi (4.4)
2: Nei prossimi n step, applica l'ingresso $u_{[t,t+n-1]} = \bar{u}^*_{[0,n-1]}(t)$
3: $t \leftarrow t + n$ e torna a 1)

Similmente al caso nominale, è possibile fornire le seguenti garanzie in catena chiusa ([2], Teorema 3).

Teorema 4.2.1. *Sia $L \geq 2n$, u^d persistently exciting di ordine $L + 2n$, allora esistono dei parametri $\lambda_\alpha, \lambda_\sigma > 0$ scelti opportunamente e un bound al rumore $\bar{\varepsilon}$ sufficientemente piccolo tale che il setpoint (u^s, y^s) è esponenzialmente stabile in catena chiusa. Inoltre la regione di attrazione aumenta e l'errore di tracking diminuisce se*

- *il bound al rumore $\bar{\varepsilon}$ diminuisce*
- *l'energia dell'ingresso $\|u^d\|$ aumenta*
- *il numero di dati N aumenta*

4.3 Estensioni per sistemi non lineari

Una delle più grandi sfide nel data-driven control è lo sviluppo di metodi per controllare sistemi non-lineari con dinamica non nota con garanzie teoriche in catena chiusa. Lo sviluppo di schemi data-driven MPC con garanzie in catena chiusa è tema di ricerca attuale e non viene presentato in questo elaborato. Si vuole però mostrare qui che è possibile realizzare uno schema MPC basato sul Teorema 3.2.1 simile a quello realizzato nelle sezioni precedenti. E' possibile

infatti risolvere il problema della dinamica non lineare aggiornando in tempo reale i dati $\{u_k^d, \tilde{y}_k^d\}_{k=0}^{N-1}$ usati per le predizioni badandosi sulle misurazioni correnti. Questo approccio è equivalente allo sfruttare il fatto che il sistema non lineare può essere linearizzato localmente.

4.4 Esempio di implementazione in Matlab per sistemi LTI in assenza di rumore

Consideriamo il sistema (*doppio integratore*):

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \tag{4.5}$$

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(k)$$

con vincoli di saturazione sull'input dati da $-1 \leq u(k) \leq 1$ e matrici di costo $Q = I, R = 0.1$.
Viene mostrata una possibile implementazione con Matlab, ottenuta realizzando una funzione che imposta il problema di ottimizzazione (4.3) con setpoint $(0, 0)$.

```
function [u_0,y] = data_driven_MPC(L,n,N,u_initial,y_initial,H_u,
    H_y,R,Q)
    problem = optimproblem;
    %variabili di ottimizzazione
    alpha = optimvar('alpha',N-L-n+1,1);
    u_prediction = optimvar('u_prediction',L+n,1,'LowerBound',-1,
    'UpperBound',1);
    y_prediction = optimvar('y_prediction',L+n,1);
    constraints1 = optimconstr(2*(L+n));
    %vincoli sulla dinamica
    constraints1 = [u_prediction' y_prediction']' == [H_u' H_y
    ']'*alpha;
    constraints2 = optimconstr(2*n);
    %imposizione condizioni iniziali
    constraints2 = [u_prediction(1:n)' y_prediction(1:n)']' == [
    u_initial y_initial]';
    constraints3 = optimconstr(2*n);
    %condizioni terminali
    constraints3 = [u_prediction(L+1:L+n)' y_prediction(L+1:L+n)
    ']' == zeros(2*n,1);
    constraints = [constraints1' constraints2' constraints3']';
```

```
    problem.Constraints.constraints = constraints;
    %definizione funzione di costo
    cost = 0;
    for k = 1:L
        cost = cost + u_prediction(k,1)*R*u_prediction(k,1) +
     y_prediction(k,1)*Q*y_prediction(k,1);
    end
    problem.Objective = cost;
    solution = solve(problem);
    %ingresso ottimo da applicare al prossimo istante di
     campionamento
    u_0 = solution.u_prediction(n+1);
    %uscita predetta al prossimo istante di campionamento
     applicando u_0
    y = solution.y_prediction(n+1);
end
```

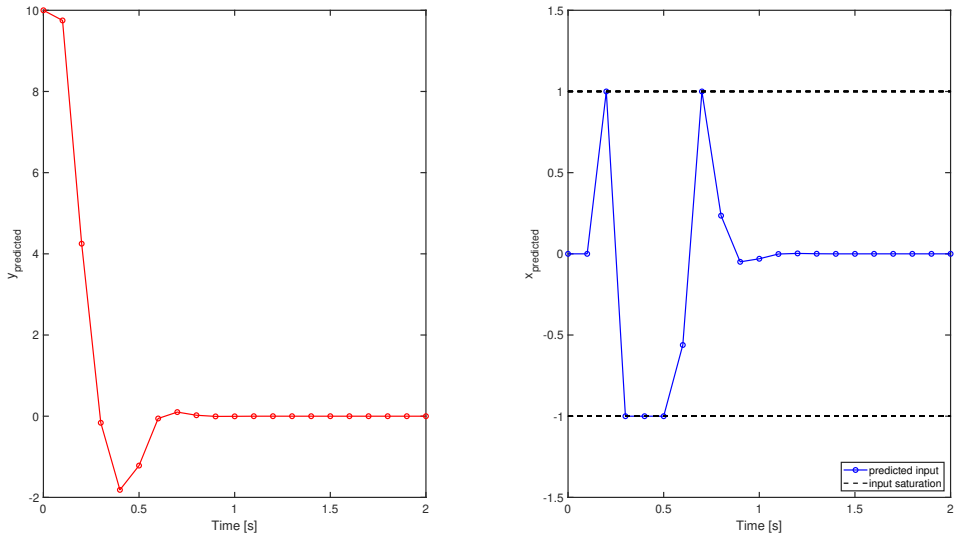
Viene poi implementato il controllore Receding Horizon con $L = 3, N = 15$ (che rispettano le condizioni del Teorema 4.1.3).

```
close all;
clear all;

%% definizione processo da controllare

n = 2; %dimensione sistema
m = 1; %dimensione input
A = [1 1; 0 1]; %matrici modelli di stato
B = [1;0]; %[0;1];
C = [1 0]; %[1 1];
D = 0;
init = [10, -0.25]; %condizioni iniziali
T = 0.1; % tempo di campionamento

%% simulazione risposta con persistently exciting input

N = 15; %numero di coppie input-output raccolte
L = 3; %time horizon
u = [zeros(1,L+2*n-1) ones(1,N-L-2*n+1)]; %input persistently
     exciting di ordine L+2n
t = n*T:T:n*T+N/10-0.1; %vengono raccolti n dati per y_initial e
     poi viene applicato l'input
sim("simulazione_processo.slx"); %simulazione processo in
     simulink
y_data = y.signals.values'; %dati simulazione
```

```
u_initial = zeros(1,n); %inizialmente non viene applicato
    ingresso
y_initial = y_data(1:n); %primi n dati per condizioni iniziali
y_data_offline = y_data(n+1:N+n); %dati offline per fundamental
    lemma

%% generazione matrici di Hankel di ordine L+n per input e output

H_uLn = []; %matrice di Hankel di ordine L+n per l'input
H_yLn = []; %matrice di Hankel di ordine L+n per l'output
for i = 1:N-L-n+1
    H_uLn = [H_uLn u(i:L+n+i-1)'];
    H_yLn = [H_yLn y_data_offline(i:L+n+i-1)'];
end

%% data-driven MPC

R = 0.1; %parametri funzione di costo
Q = 1;
u = u_initial; %sequenza ingressi
y = y_initial; %sequenza uscite

simulation_time = 0:T:2; %vengono simulati i primi 10 secondi

for i = 1:length(simulation_time)-2
    %prende gli ultime n coppie input-output per condizioni
     iniziali
    u_initial = u(i:length(u));
    y_initial = y(i:length(y));
    %soluzione problema di ottimizzazione
    [u_0,y1] = data_driven_MPC(L,n,N,u_initial,y_initial,H_uLn,
     H_yLn,R,Q);
    y = [y y1];
    u = [u u_0];
end

%% visualizzazione risultati
figure;
plot(simulation_time,y,'r-o');
xlabel('Time [s]'); ylabel('y_{predicted}');
figure;
plot(simulation_time,u,'b-o');
ylim([-1.5 1.5]);
hold on;
```

```
plot(simulation_time,-1*ones(length(simulation_time)),'k--',
     simulation_time,1*ones(length(simulation_time)),'k--');
xlabel('Time [s]'); ylabel('x_{predicted}');
legend('predicted input','input saturation','Location','southeast
     ');
```

Vengono riportati in figura 4.1 i risultati della simulazione dove si può vedere che il setpoint viene raggiunto.



(a) *Output* **(b)** *Input*

Figura 4.1: Simulazione controllore data-driven MPC sul sistema (4.5)

Bibliografia

[1] J.C. Willems, P.Rapisarda, I.Markovsky, and B. L. De Moor, *"A note on persistency of excitation"*, Systems & Control Letters, vol. 54, no. 4, pp. 325-329, 2005

[2] Julian Berberich, Johannes Kühler, Matthias A. Müller and Frank Allgöwer, *"Data-driven model predictive control: closed-loop guarantees and experimental results"*, arXiv preprint at arXiv:2107.00966

[3] Francesco Borrelli, Alberto Bemporad, Manfred Morari, *"Predictive Control for Linear and Hybrid Systems"*, Cambridge University

[4] Alessandro Beghi, *"Lecture Notes on Adaptive and Model Predictive Control"*, Università di Padova

[5] Julian Berberich, Johannes Kühler, Matthias A. Müller and Frank Allgöwer, *"Data-Driven Model Predictive Control with Stability and Robustness Guarantees"*, IEEE Transactions on Automatic Control, 2020.

[6] Vishaal Krishnan, Fabio Pasqualetti, *"On Direct vs Indirect Data-Driven Predictive Control"*, arXiv preprint arXiv:2103.14936, 2021

[7] Valentina Breschi, Alessandro Chiuso, Simone Formentin, *"The role of regularization in data-driven predictive control"*, arXiv preprint arXiv:2203.10846, 2022.

[8] Zhong-Sheng Hou, Zhuo Wang, *"From model-based control to data-driven control: Survey, classification and perspective"*, Information Sciences,vol. 235, 2013.