

Appunti sul Controllo Predittivo basato sulla conoscenza del Modello¹

Ludovico Pinzari
Dipartimento di Informatica e Automazione
Università degli Studi "Roma TRE"



5 gennaio 2010

Indice

Indice	2
Elenco delle figure	4
Elenco delle tabelle	5
1 Introduzione	6
1.1 Perchè il Model Predictive Control	6
1.2 Tassonomia dei Sistemi di controllo	8
1.3 Receding Horizon Controls	10
1.3.1 RHC vs Optimal Control	13
1.4 Model Predictive Control Strategy	15
1.4.1 Prediction Models	18
2 Discrete-Time MPC	22
2.1 Introduzione	22
2.2 State-space Models with Embedded Integrator	22
2.2.1 Single-input and Single-output System	23
2.3 Predictive Control within One Optimization Window	25
2.3.1 Prediction of State and Output Variables	25
2.3.2 Optimization	26
2.4 Receding Horizon Control state space model	34
2.4.1 Closed-loop Control System	36
2.5 Predictive Control of MIMO Systems	40
2.5.1 General formulation of the model	45
2.5.2 Solution of Predictive Control for MIMO Systems	49
2.6 State Estimation	50
2.6.1 Basic Ideas about an Observer	50
2.6.2 Basic Results About Observability	55
2.6.3 Kalman Filter	57
2.7 State Estimate Predictive Control	58

3 Appendice	62
3.1 Appendice	62

Elenco delle figure

1.1	control system	8
1.2	state feedback control	9
1.3	output feedback control	9
1.4	tassonomia	10
1.5	One-time long term planning	11
1.6	Periodic short-time planning	12
1.7	Annual short-time planning	12
1.8	Mpc Strategy	16
1.9	Mpc Basic Structure	16
1.10	Mpc analogy	18
1.11	Impulse response	19
1.12	Step response	20
2.1	Mpc with no weighth on Δu	31
2.2	e^2 Mpc with no weighth on Δu	31
2.3	Mpc with weighth on Δu	32
2.4	e^2 Mpc with weighth on Δu	32
2.5	Block diagram of DMPC	37
2.6	MPC short $N_p=20$ no weight	41
2.7	MPC error short $N_p=20$ no weight	41
2.8	MPC short $N_p=20$ ill-conditioning	41
2.9	MPC short $N_p=20$ weight=0.5	42
2.10	MPC error short $N_p=20$ weight=0.5	42
2.11	MPC short $N_p=20$ ill-conditioning weight=0.5	42
2.12	MPC long $N_p=200$ weight=0.5	43
2.13	MPC error long $N_p=200$ weight=0.5	43
2.14	MPC error long $N_p=200$ ill-conditioning weight=0.5	43
2.15	MPC short $N_p=20$ first weight=0.5	44
2.16	MPC error short $N_p=20$ first weight=0.5	44
2.17	Estimation without observer	54
2.18	Estimation with observer	54
2.19	Motor model	55

2.20 Block diagram of DMPC with observer	59
2.21 Predictive Control of double integrator plant	61

Elenco delle tabelle

1.1 Investment planning vs Control	13
--	----

Capitolo 1

Introduzione

1.1 Perché il Model Predictive Control

Il Model Predictive Control ha una lunga storia nel campo controllistico, è una di quelle poche aree che ha avuto un crescente interesse dai ricercatori sia in ambito accademico che in quello industriale.

Il termine MPC non definisce una strategia di controllo specifica, ma individua una grande famiglia di metodologie che fanno un'utilizzo esplicito del modello impiegato per descrivere il processo. L'obiettivo principale è quello di ottenere il segnale di controllo ottimo per la minimizzazione di un certo funzionale di costo che riflette la performance del sistema.

Questi metodi di progettazione conducono alla realizzazione di controllori lineari che presentano praticamente quasi la medesima struttura e che sono caratterizzati dal fatto di essere facilmente implementabili.

Fondamentalmente le idee di base che governano tutte le strategie di controllo appartenenti a suddetta famiglia sono:

- L'utilizzo esplicito di un modello che permetta di prevedere l'evoluzione temporale del sistema su un certo orizzonte.
- Il calcolo di una traiettoria di controllo che minimizzi una funzione obiettivo.
- Una strategia che consenta di estendere l'orizzonte temporale rispetto all'ultima azione di controllo esercitata sul sistema.

Quello che differenzia i vari algoritmi è la scelta del modello adottato per descrivere il processo, i disturbi ed il funzionale di costo da minimizzare. Ciò che rende questa metodologia di progetto particolarmente attraente è legato principalmente ai seguenti aspetti:

1. Il primo aspetto riguarda la facilità d'implementazione e consente ad un personale che presenti una conoscenza piuttosto limitata delle tecniche di controllo di comprendere la legge di controllo utilizzata ed allo stesso tempo di effettuare il tuning dei parametri del processo. Quindi si può ritenere una valida alternativa ai controllori PID, in cui il procedimento di taratura è dettato da relazioni empiriche (vedi Tabelle Ziegler-Nichols) che non riflettono una visione chiara di progettazione e spesso è necessario raggiungere un compromesso tra il tuning dei parametri ed il comportamento a ciclo chiuso del sistema. In particolare in presenza di vincoli che differenziano un problema da un'altro, diventa ancor più difficile utilizzare delle tabelle empiriche, anche perfino in condizioni operative nominali dove non è sempre semplice

garantire la stabilità del sistema. Questo è il motivo per cui una gran parte dello sforzo di progettazione è basato su un gran numero di prove.

2. Può essere utilizzato per sistemi che presentano dinamiche semplici o complesse, che hanno lunghi ritardi nella risposta. Inoltre consentono di trattare sistemi instabili ed a fase non minima.
3. Un'altra caratteristica riguarda la possibilità di gestire vincoli più o meno stringenti in un sistema di controllo con più variabili. Questo aspetto risulta particolarmente interessante nel campo industriale dove inevitabilmente sono presenti dei limiti sul processo di produzione ed ovviamente degli stringenti margini di profitto.
4. Un'azione di controllo di tipo feed-forward, per compensare la presenza di disturbi misurabili sulla catena diretta. Inoltre, per la sua struttura intrinseca consente di compensare i dead times (ovvero il ritardo che intercorre tra l'emissione del segnale di controllo e la risposta della variabile di processo misurata), soprattutto in sistemi in cui la traiettoria da eseguire è nota a priori (robotica industriale).
5. Ma l'aspetto che maggiormente interessa il campo controllistico riguarda la possibilità di effettuare un'ottimizzazione del processo in TEMPO-REALE.

E' chiaro che presenta anche degli svantaggi. Infatti, sebbene la legge di controllo risulti facilmente implementabile e sia caratterizzata dal fatto di avere una complessità computazionale non onerosa, il calcolo della sua derivata è molto più complesso rispetto ai classici controllori PID.

Se le dinamiche del processo non cambiano o comunque la loro variazione è trascurabile, la derivata della legge di controllo può essere calcolata off-line, ma nel caso di un controllo di tipo adattativo, tutto il calcolo viene eseguito iterativamente ad ogni istante di campionamento. Questo naturalmente limita molto l'efficienza del nostro algoritmo, soprattutto se il sistema è soggetto a vincoli. Comunque l'ostacolo può essere facilmente superato se si dispone di potenti risorse di calcolo e memoria.

Ma la vera difficoltà risiede nella scelta di un modello appropriato per il processo in esame. La progettazione dell'algoritmo si basa soprattutto sulla conoscenza a priori del modello; quindi spesso occorre valutare i benefici che si possono ottenere adottando questa strategia in rapporto alla coerenza tra il modello adottato ed il processo reale.

In sostanza, l'MPC risulta una strategia generale abbastanza efficace nei processi industriali, nei quali una sufficiente conoscenza del processo ne permette una facile implementazione.

In questo lavoro ho cercato di fornire delle linee guida per lo sviluppo di metodologie di progettazione ed implementazione di algoritmi orientati al controllo di sistemi a tempo discreto.

In particolare, dopo aver brevemente richiamato alcuni concetti base dei sistemi di controllo, descriveremo i principi generali che governano la filosofia dell'MPC. In seguito fornirò una breve panoramica sui diversi modelli utilizzati valutando i pregi ed i difetti.

Nei capitoli a seguire concentreremo la nostra attenzione sui modelli ingresso-stato-uscita presentando una possibile formulazione del problema. Gli argomenti che verranno affrontati in modo sistematico sono i seguenti :

- **Formulazione di un Modello ingresso-stato-uscita basato sull'MPC**
- **Controllo di processi soggetti a vincoli**

- **Laguerre-function-based predictive control**
- **Predictive control using exponential data weighting**

Per quanto riguarda le simulazione e le implementazioni dei vari algoritmi, sono stati realizzati in MATLAB mentre per la simulazione è stata adottata la piattaforma di SMULINK.

1.2 Tassonomia dei Sistemi di controllo

In questo paragrafo richiameremo brevemente alcuni concetti base dei sistemi di controllo al fine di poter collocare L'MPC in uno schema di progettazione generale, in modo che non risulti semplicemente una ricetta da utilizzare ogni qualvolta si presenti un problema di controllo.

Modelli

Le variabili che caratterizzano un sistema di controllo sono quelle di ingresso, stato ed uscita in cui vi sono delle uscite controllabili e misurabili. L'ingresso è naturalmente la variabile di controllo e la misura dell'output viene impiegata come feedback del sistema.

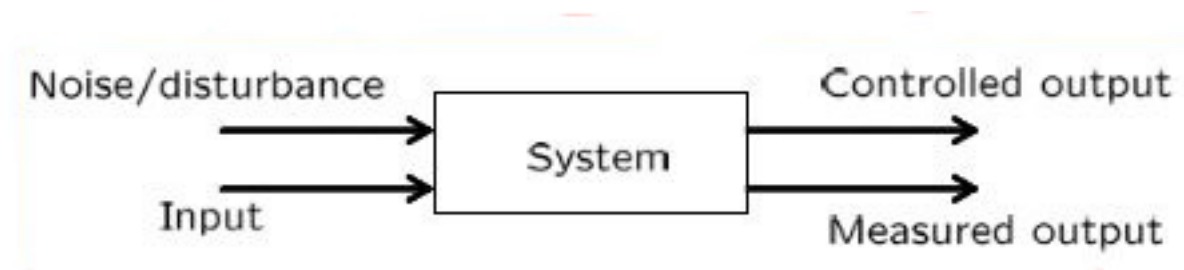


Figura 1.1: control system

Generalmente le variabili di output sono un sottoinsieme delle variabili di stato complessive del sistema. Inoltre nei sistemi dinamici deve essere tenute in considerazione effetti non desiderati legati alla presenza di disturbi, rumori, elementi non lineari, ritardi, incertezze sulle dinamiche ed i parametri del sistema ed i vincoli sugli ingressi e le uscite, etc. Tutte o una gran parte di questi elementi non desiderabili esistono in ogni sistema a seconda delle sue caratteristiche.

Un modello può essere rappresentato come un sistema stocastico affetto da rumore, oppure deterministico con disturbi. Un modello può essere un sistema lineare oppure non lineare. In genere i modelli dinamici sono descritti in termini di v.d.s oppure qualche volta come modelli ingresso-uscita.

Obiettivi del Controllo

Vi possono essere diversi obiettivi nei sistemi di controllo. La legge di controllo è progettata in maniera tale che le uscite controllabili inseguino il segnale di riferimento nel miglior modo possibile. In realtà occorre tener presente che la stabilità del sistema a ciclo chiuso e la performance con cui avviene il tracking sono altrettanto importanti, e sono ovviamente influenzate dai disturbi, rumori..etc.

Quindi al fine di poter raggiungere questi obiettivi facilmente, il sistema reale viene analizzato in due sottosistemi che sono rispettivamente: un sistema nominale che non presenta incertezze nel modello ed un

sistema in cui si considerano gli elementi che degradano le prestazioni di tale sistema. Pertanto abbiamo una fase di progettazione che parte dal sistema nominale e che va successivamente affinata considerando le incertezze del sistema.

In tal modo, gli obiettivi del controllo possono essere suddivisi in obiettivi intermedi più semplici come:

- **Stabilità nominale**
- **Performance nominale**
- **Stabilità Robusta**
- **Performance Robusta**

Le prime tre sono considerate le più importanti.

Struttura del Controllo

Se tutto lo stato del sistema è misurabile possiamo far riferimento ad un sistema con reazione dallo stato rappresentato nella figura 1.2. Se invece è possibile misurare solamente le uscite del sistema allora possiamo utilizzare un sistema con reazione dall'uscita riportato in figura 1.3.

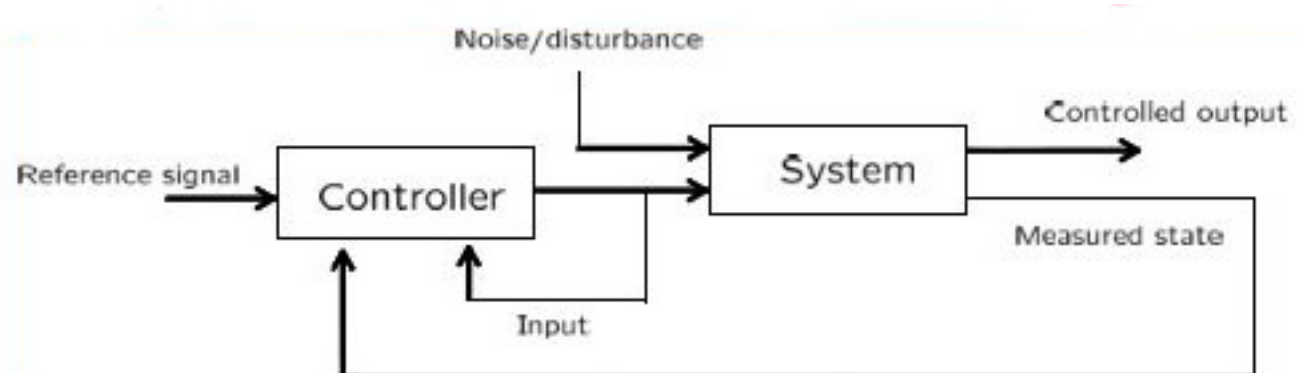


Figura 1.2: state feedback control

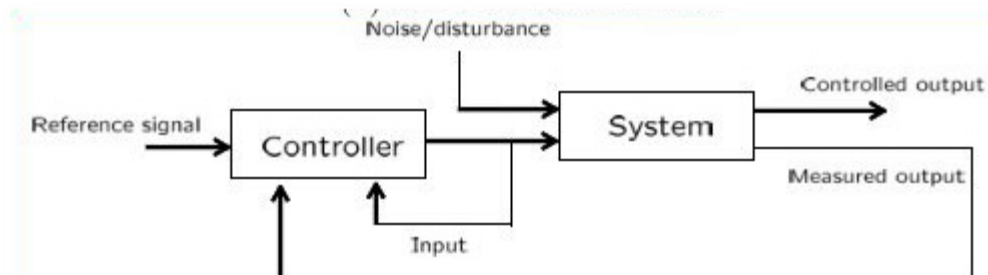


Figura 1.3: output feedback control

In generale, i sistemi di controllo con reazione dallo stato risultano più semplici da progettare rispetto a quelli dall'uscita, dato che lo stato contiene tutta l'informazione necessaria per descrivere il processo. Poi abbiamo i

sistemi di controllo con memoria finita che si basano su una combinazione lineare di un certo numero di ingressi ed uscite del sistema.

Criteri di Performance

Ci sono diversi modi di affrontare la progettazione di un sistema di controllo al fine di ottemperare ad opportune specifiche.

Il Controllo Ottimo è uno dei metodi più adottati che si basa sostanzialmente minimizzando o massimizzando un'opportuno funzionale di costo. Tale ottimizzazione viene realizzata su un'orizzonte temporale finito oppure infinito. Per i sistemi lineari, la minimizzazione avviene generalmente considerando il popolare LQ per i controllori con reazione dallo stato, mentre per quelli dall'uscita viene adottato LQG.

I controllori ottimi vengono in genere realizzati in catena aperta se si considerano dei sistemi non lineari. Sebbene tali controllori vengono realizzati per soddisfare opportuni criteri di performance devono comunque rispettare le specifiche accennate in precedenza. In particolare, per i sistemi a ciclo chiuso su un'orizzonte temporale infinito sotto opportune condizioni tale tipo di controllo fornisce dei soddisfacenti risultati per il tracking e la stabilità del sistema. Tuttavia non garantisce la stabilità robusta in presenza di incertezze sul modello.

I componenti principali possono essere quindi individuati dalla tassonomia riportata in figura 1.4.

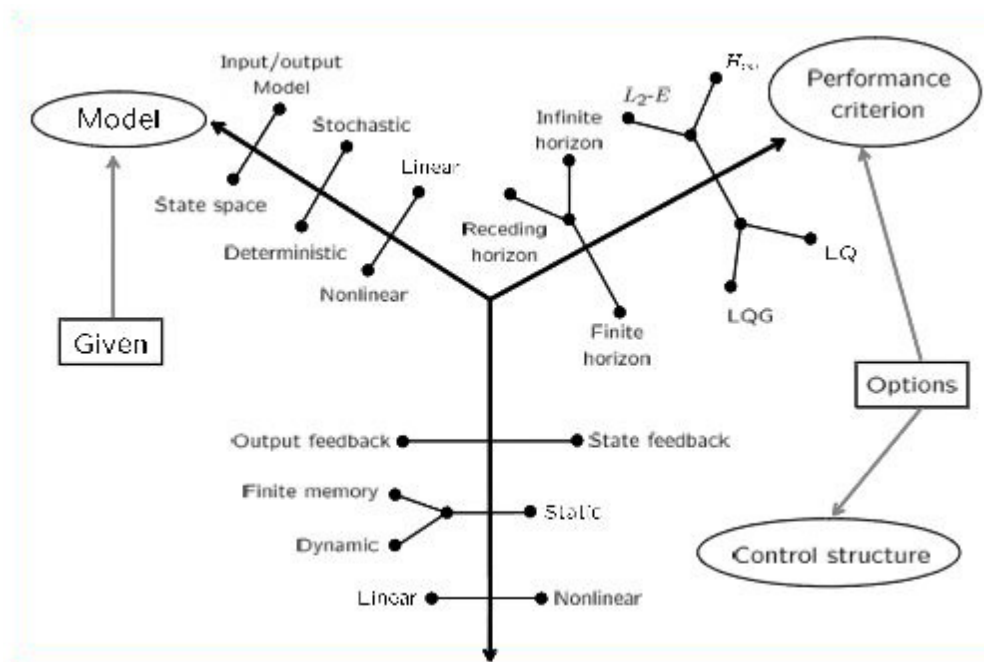


Figura 1.4: tassonomia

1.3 Receding Horizon Controls

Il Controllo Ottimo convenzionalmente viene utilizzato considerando un'orizzonte temporale finito oppure illimitato. Spesso, i sistemi di controllo a retroazionati devono essere progettati per esercitare la loro funzione

per tempi sufficientemente lunghi, soprattutto nei processi industriali come le centrali elettriche oppure gli stabilimenti chimici.

In queste circostanze risulta evidente che l'ottimizzazione non può essere effettuata su un orizzonte temporale finito; quindi è necessario ricorrere al Infinite Horizon Optimal Control. A partire da queste tecniche è possibile introdurre un nuovo tipo di controllo che considera sempre un orizzonte temporale infinito ma la sua strategia si basa sul concetto del RHC (Receding Horizon Control).

La filosofia dell' RHC è molto semplice. In ogni istante temporale si stabilisce la legge di controllo ottima (che può essere un sistema di controllo ad anello chiuso oppure ad anello aperto) su un certo orizzonte temporale predefinito; ad esempio a partire dall'istante corrente k fino a $k+N$ ovvero nell'intervallo $[k, k+N]$. Quindi all'istante temporale corrente viene implementata la prima legge di controllo e successivamente viene ripetuto tale calcolo nell'intervallo successivo $[k+1, k+1+N]$. Il termine 'Receding Horizon' viene dunque utilizzato per esprimere l'estensione temporale su cui viene considerata la legge di controllo ottima del sistema durante l'evoluzione del processo. Esiste anche un'altro tipo di controllo noto come intervalwise RHC, dove il termine 'intervalwise' sta ad indicare un'estensione temporale che viene fatta periodicamente e non in ogni istante di campionamento del sistema.

La filosofia dell'RHC può essere facilmente descritta considerando ad esempio l'attività di pianificazione degli investimenti di un'azienda che intenda ovviamente massimizzare il proprio profitto. La pianificazione viene comunque considerata negli anni a seguire come in un sistema di controllo retroazionato.

Vi possono essere tre differenti strategie:

1. Un'unica pianificazione a lungo termine

Ovvero l'attività di planning viene valutata su un'arco temporale abbastanza lungo da poter essere considerato illimitato (ovvero assimilabile a $+\infty$). Questa politica corrisponde a considerare un controllo ottimo nell'intervallo $[k, +\infty)$. Vedi figura 1.5.

2. Pianificazione periodica a breve termine

Invece di considerare un'unica pianificazione e quindi ottimizzazione, possiamo ripetere periodicamente tale attività; ad esempio ogni 5 anni. Vedi figura 1.6.

3. Pianificazione a breve termine su base annuale

Quindi ogni anno viene stabilita la politica da intraprendere per l'anno in corso e negli anni a venire viene ripetuta tale attività. Vedi figura 1.7.

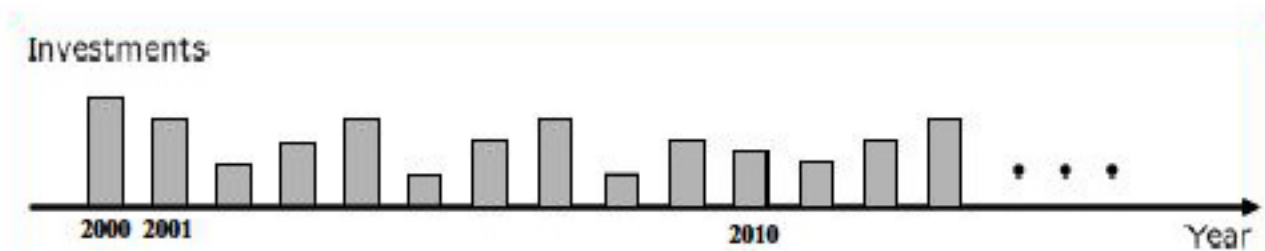


Figura 1.5: One-time long term planning

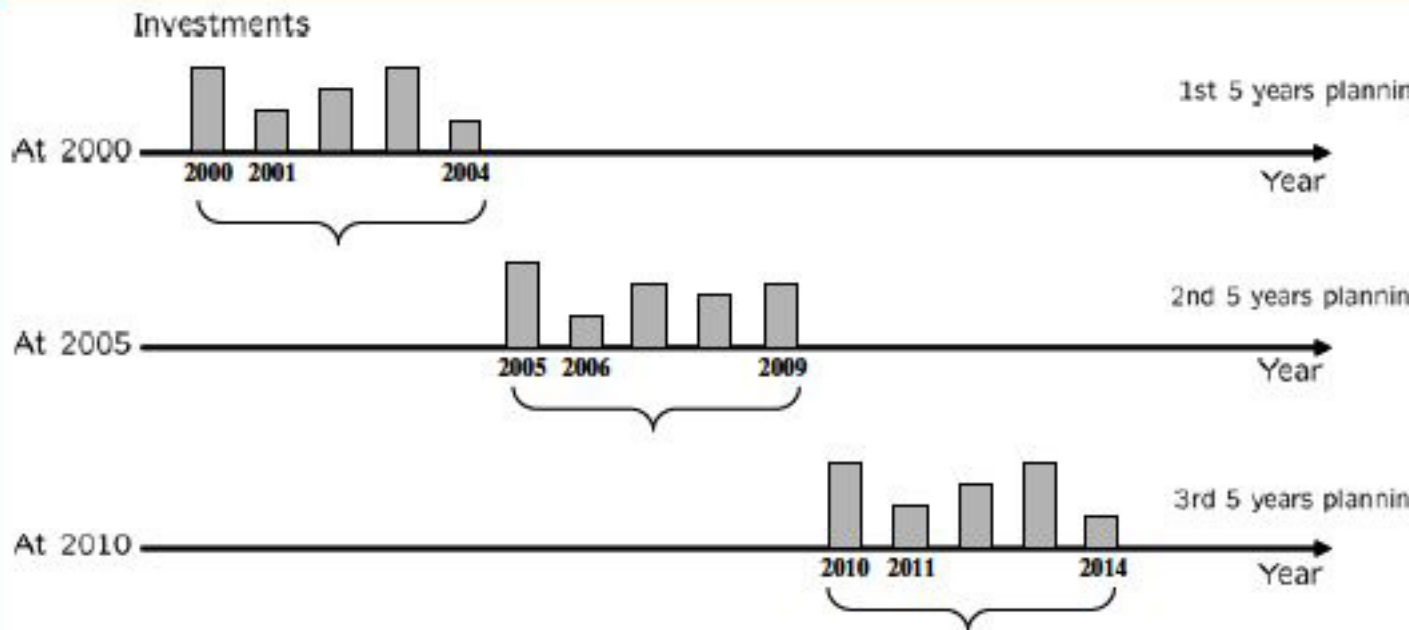


Figura 1.6: Periodic short-time planning

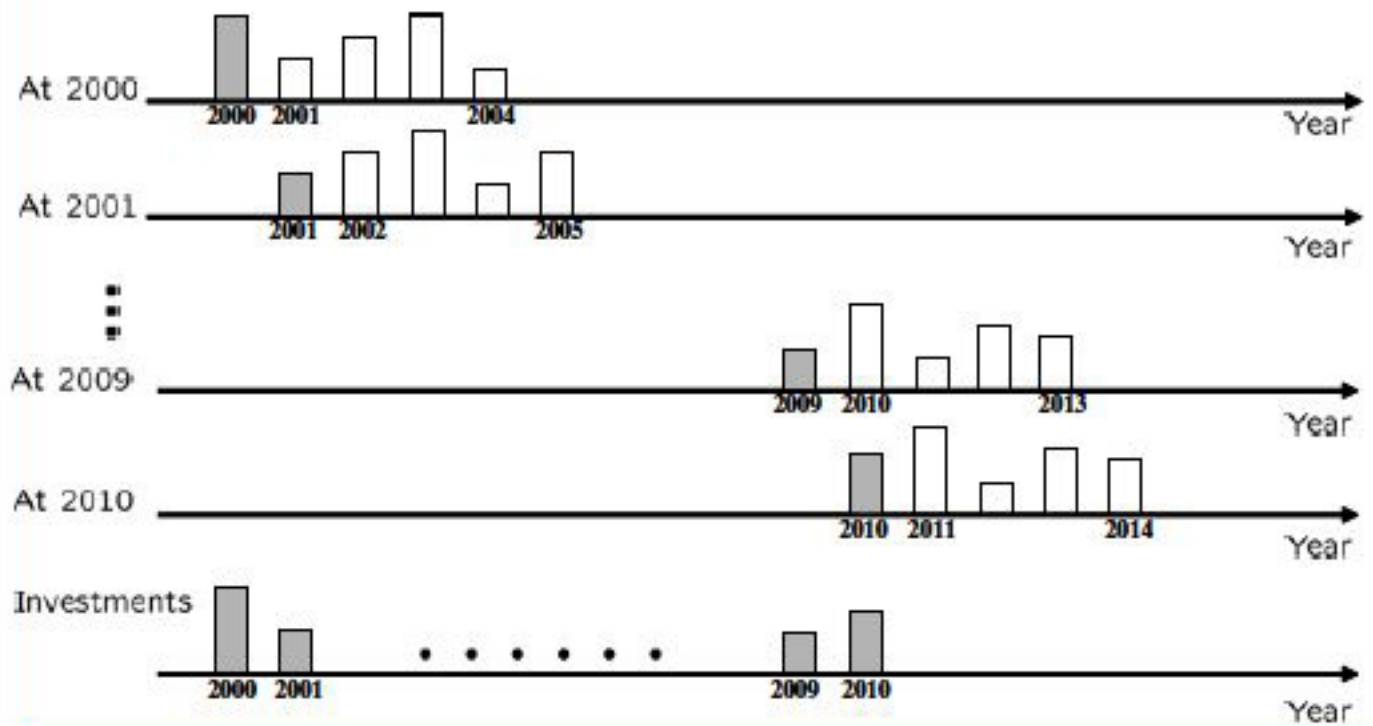


Figura 1.7: Annual short-time planning

Processo	Pianificazione	Controllo
Continui	One-time long term planning	Infinite Horizon Control
	Periodic short term planning	Intervalwise RHC
	Annual short term planning	RHC
Finiti	Short term planning	Finite Horizon Control

Tabella 1.1: Investment planning vs Control

A questo punto sorge la seguente domanda:

Quale tra le diverse strategie può essere considerata la migliore?

Ovviamente occorre definire il termine 'migliore', che risulta naturalmente un concetto soggettivo e può essere differente da persona a persona. Quindi qualunque piano può essere considerato ottimo a seconda della percezione individuale. E' chiaro che gli esempi considerati possono essere vaghi da un punto di vista matematico, sebbene siano sufficientemente adeguati per comprendere i concetti che sono alla base dell'RHC.

La politica Annual short-time planning esprime proprio il concetto di RHC, mentre la Period short-time planning è equivalente all'interval-wise RHC.

I differnti tipi d'investimento possono essere confrontati con le varie tipologie di controllo ottimo, nella tabella 1.1.

Si osserva che l'investimento a breve termine corrisponde al Finite Horizon Control, il quale risulta particolarmente adatto per processi di durata limitata come i sistemi di controllo missilistici. Generalmente i controllori ottimi vengono impiegati in sistemi in cui la legge di controllo è ad anello aperto (ovvero un sistema in cui l'azione di controllo è indipendente dall'uscita) e dipende dallo stato iniziale del sistema.

- Nel caso dell'Infinite Horizon Optimal Control tutte le leggi di controllo sono ad anello aperto ad eccezione dell'istante iniziale.
- Nel Intervalwise Receding Horizon Control, solamente la legge di controllo che viene implementata su ogni orizzonte temporale è ad anello chiuso e quelle successive sono ad anello aperto.
- In fine nel Receding Horizon Control, abbiamo sempre delle leggi di controllo ad anello chiuso dovuto alla natura iterativa con cui avviene la computazione e l'implementazione della prima legge di controllo.

1.3.1 RHC vs Optimal Control

Analizziamo i vantaggi e gli svantaggi che offre la metodologia RHC rispetto ad altri metodi di ottimizzazione e controllo.

Vantaggi:

- **Closed-loop control:** Un'approccio sistematico che consente di ottenere una legge di controllo ad anello chiuso. Mentre il controllo ottimo nei sistemi lineari e non, caratterizzati da vincoli sui segnali di ingresso e di uscita sono generalmente dei controllori ad anello aperto; L'RHC fornisce sempre dei controllori ad anello chiuso garantendo una Robustezza intrinseca al sistema se si dispone di un buon modello per le grandezze esogene.
- **Capacità di gestire più vincoli:** I sistemi lineari che presentano vincoli sugli ingressi e lo stato del processo sono piuttosto comuni in ambito industriale, l'RHC costituisce una strategia efficace e facilmente calcolabile utilizzando gli strumenti della programmazione matematica, come la Programmazione Quadratica (QP) e la programmazione semi-definita (SDD). Addirittura, per sistemi non lineari, l'RHC

può gestire i problemi connessi all'analisi numerica legata ai vincoli imposti sugli ingressi e lo stato del sistema che derivano dall'ottimizzazione della funzione obiettivo sull'orizzonte finito.

- **Garantisce la Stabilità:** Nei sistemi lineari e non, con vincoli sulle norme dei segnali di ingresso e sui valori delle variabili di stato, l'RHC garantisce la stabilità sotto deboli condizioni. Il controllo ottimo su un'orizzonte temporale infinito, ovvero in una situazione di regime, costituisce una valida alternativa all'RHC; Tuttavia è applicabile solo se si è in grado di trovare una legge di controllo in forma chiusa e questo in generale è un problema molto complesso, soprattutto se il numero dei vincoli e delle variabili del problema è elevato.
- **Good Tracking Performance:** L'RHC presenta delle buone capacità d'inseguimento della traiettoria di riferimento soprattutto nei sistemi di asservimento, dove l'ingresso varia con una dinamica comparabile con quella del sistema stesso. Infatti, in molte circostanze in cui è nota a priori la traiettoria che si desidera eseguire su un orizzonte temporale finito e le operazioni del processo sono cicliche, questa strategia risulta valida, specialmente nella robotica industriale dove lo spazio di lavoro è caratterizzato da un'ambiente strutturato. Appare evidente che in questa circostanza i regolatori PID non sono altrettanto efficaci poichè il loro controllo è basato solo sul segnale di riferimento corrente, anche se è disponibile l'andamento completo del segnale lungo un certo orizzonte temporale. In particolare con i PID diventa altrettanto difficile gestire i possibili transitori che possono insorgere a seguito di improvvise variazioni della traiettoria di riferimento. Quindi l'RHC fa del suo meglio per inseguire la traiettoria futura di segnali che sono caratterizzati da una larga banda passante. Naturalmente se si considera un orizzonte temporale infinito è necessario conoscere tutti i segnali di riferimento futuri. Tuttavia, spesso non sono sempre disponibili nelle applicazioni reali ed il loro calcolo nella maggior parte dei casi è improponibile.
- **Adattamento alle variazioni parametriche:** L'RHC può essere una strategia appropriata per sistemi non stazionari, in cui si conosce l'andamento delle grandezze parametriche del sistema. L'RHC richiede la conoscenza del loro andamento su un'orizzonte temporale finito per poter calcolare il segnale di controllo corrente. Tuttavia se si considera un orizzonte temporale infinito è necessario conoscere tutte le possibili variazioni parametriche che possono avvenire in futuro nel sistema e pertanto in questo ultimo caso risulta molto difficile e richiederebbe una memoria pressochè infinita per calcolare i segnali di controllo futuri. Comunque questa difficoltà può essere superata, poichè l'RHC è una procedura iterativa, e quindi è in grado di adattarsi alle variazioni parametriche che è possibile aggiornare in tempo reale. Infatti, durante l'evoluzione temporale si acquisiscono progressivamente informazioni sul sistema a differenza del controllo ottimo (che appartiene alla categoria dell'Infinite Horizon optimal control) che non è un'algoritmo adattativo poichè lavora off-line.
- **Flessibilità progettuale:** E' noto che sotto opportune condizioni, a regime il controllo ottimo come LQ, LQG e H_∞ garantiscono una certa stabilità e robustezza del sistema. Anche l'RHC possiede suddette proprietà, inoltre presenta più parametri di progettazione (come ad esempio la dimensione della finestra temporale) che consentono di ottenere una migliore performance.
- **Complessità computazionale:** Richiede un calcolo più semplice rispetto al controllo ottimo a regime. Infatti, l'ottimizzazione viene considerata su un orizzonte temporale finito, e la soluzione può essere ricavata in forma chiusa per lo specifico istante di campionamento e corrispondente finestra temporale.
- **Largo impiego nelle applicazioni industriali:** Questo aspetto è legato principalmente al fatto che i processi industriali presentano grandi limitazioni sui segnali di controllo e richiedono che lo stato del sistema non si allontani molto da specifiche regioni operative di sicurezza. L'RHC consente quindi di gestire efficacemente entrambi questi vincoli. In particolare, la sua efficacia si manifesta soprattutto se nelle specifiche di progetto è richiesto che il sistema, per avere una performance che massimizzi il profitto, lavori intorno ad una ristretta regione che ovviamente è un sottoinsieme di quella vincolata. Per questa ragione, la capacità di gestire i vincoli è molto importante. I controllori convenzionali sono piuttosto conservativi, ovvero esercitano un'azione di controllo che sebbene soddisfi i vincoli del problema si trova comunque lontano dall'ottimo globale. Quindi sebbene si abbiano delle soluzioni ammissibili al problema, queste non sono ottime; ovvero non abbiamo prestazioni spinte. Questo è legato al problema della gestione di 'vincoli dinamici', ovvero vincoli che numericamente possono cambiare nell'evoluzione del sistema e quindi non possono essere analizzati in una fase di progettazione preliminare. Dato che

le dinamiche del processo sono relativamente lente rispetto al segnale di riferimento, grazie alla natura iterativa dell'algoritmo è possibile effettuare il calcolo ad ogni istante di campionamento per raggiungere la condizione operativa ottimale del sistema.

Svantaggi:

- **Tempo effettivo di calcolo:** Sebbene la complessità computazionale dell'RHC in ogni istante di campionamento risulta migliore rispetto a quella dell'Infinite Horizon Optimal Control, il suo tempo di calcolo complessivo è maggiore, soprattutto se i sistemi non sono lineari. Inoltre, la maggior parte degli algoritmi presenti nei pacchetti commerciali dei solutori di ottimizzazione forniscono delle soluzioni sub-ottime (ovvero hanno una valenza locale al problema di minimizzazione) in modo da velocizzare la ricerca. Questo inconveniente può ritenersi accettabile se si considerano applicazioni di tipo Real Time, dove la ricerca dell'ottimo globale ad ogni istante di campionamento potrebbe essere un problema non fattibile. Tuttavia, questa difficoltà intrinseca non fornisce alcuna attenuante all'ottimizzatore a patto che il sub-ottimo sia abbastanza vicino all'ottimo globale. Comunque, questo problema può essere in parte superato, utilizzando dei processori più veloci congiuntamente ad ulteriori miglioramenti negli algoritmi di ottimizzazione che nella maggior parte dei casi devono essere progettati ad hoc.
- **Performance and Stability Robustness:** Le proprietà del sistema, come la stabilità e la performance Robusta a seguito delle incertezze legate alla conoscenza dei parametri del sistema sono spesso intrattabili nei problemi di ottimizzazione, dove è richiesto un certo determinismo nella struttura del problema e quindi nella formulazione della funzione obiettivo e dei vincoli associati. Inoltre la natura iterativa dell'algoritmo rende difficile analizzare la robustezza del sistema che risulta più semplice da studiare seguendo una progettazione TOP-DOWN piuttosto che una BOTTOM-UP. Comunque, la robustezza rispetto alle grandezze esogene del sistema può essere in qualche modo gestita utilizzando la tecnica minimax RHC.

Nei paragrafi successivi analizzeremo come questa strategia venga adottata nel caso del Model Predictive Control.

1.4 Model Predictive Control Strategy

La metodologia utilizzata da tutti quanti i controllori che appartengono alla famiglia dell'MPC è caratterizzata dalla seguente strategia descritta nello schema riportato in figura 1.8 ed una sua implementazione in fig 1.9 che viene descritta in tre fasi.

1. Sulla base del modello prescelto per descrivere il processo, vengono stimate le possibili uscite del sistema su un'orizzonte temporale predefinito N , chiamato 'orizzonte di previsione', che in qualche maniera riflette la lungimiranza dell'algoritmo. In sintesi sto prevedendo il comportamento del sistema su una certa Time Window (finestra temporale).

Quindi l'algoritmo prevede l'andamento delle uscite del sistema $\hat{y}(t+k | t)$ per $k = 1, \dots, N$ ¹ (vettore dei campioni dell'uscita previsti nei successivi istanti di campionamento compresi nella time window), a partire dalla conoscenza di due vettori che costituiscono rispettivamente i valori degli ingressi e delle uscite passate del sistema fino all'istante corrente t ed i segnali di controllo che ho previsto di applicare al sistema nella finestra temporale considerata, ovvero $u(t+k | t)$ per $k = 0, \dots, N-1$.

Risulta evidente che una delle scelte principali è quella di valutare il numero di campioni da considerare e questo è strettamente legato alla memoria del processo. Sebbene tale scelta possa essere arbitraria essa deve rispettare il principio di causalità del sistema. Quindi, in generale, il campione dell'uscita all'istante k è una combinazione lineare dei campioni precedenti dell'uscita fino all'istante $(k-n)$ e di quelli dell'ingresso fino all'istante $(k-n+m)$ con $m \leq n$, se m è pari ad n il sistema si trova al limite di causalità. Vedremo che nel nostro modello considereremo il caso $m < n$ al fine di poter applicare la strategia RHC. In sintesi il numero dei campioni dell'ingresso non può essere in generale maggiori o uguale a quelli dell'uscita.

Quindi a partire da questi tre vettori che rappresentano rispettivamente le informazioni relative al comportamento del sistema nelle evoluzioni temporali passate e la possibile strategia di controllo che si vuole

¹il valore della variabile di output prevista all'istante $t+k$, calcolato all'istante t

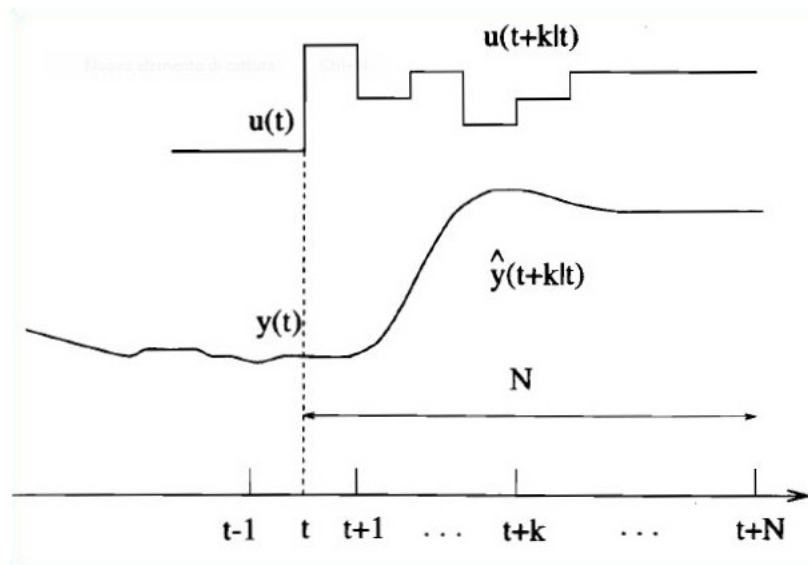


Figura 1.8: Mpc Strategy

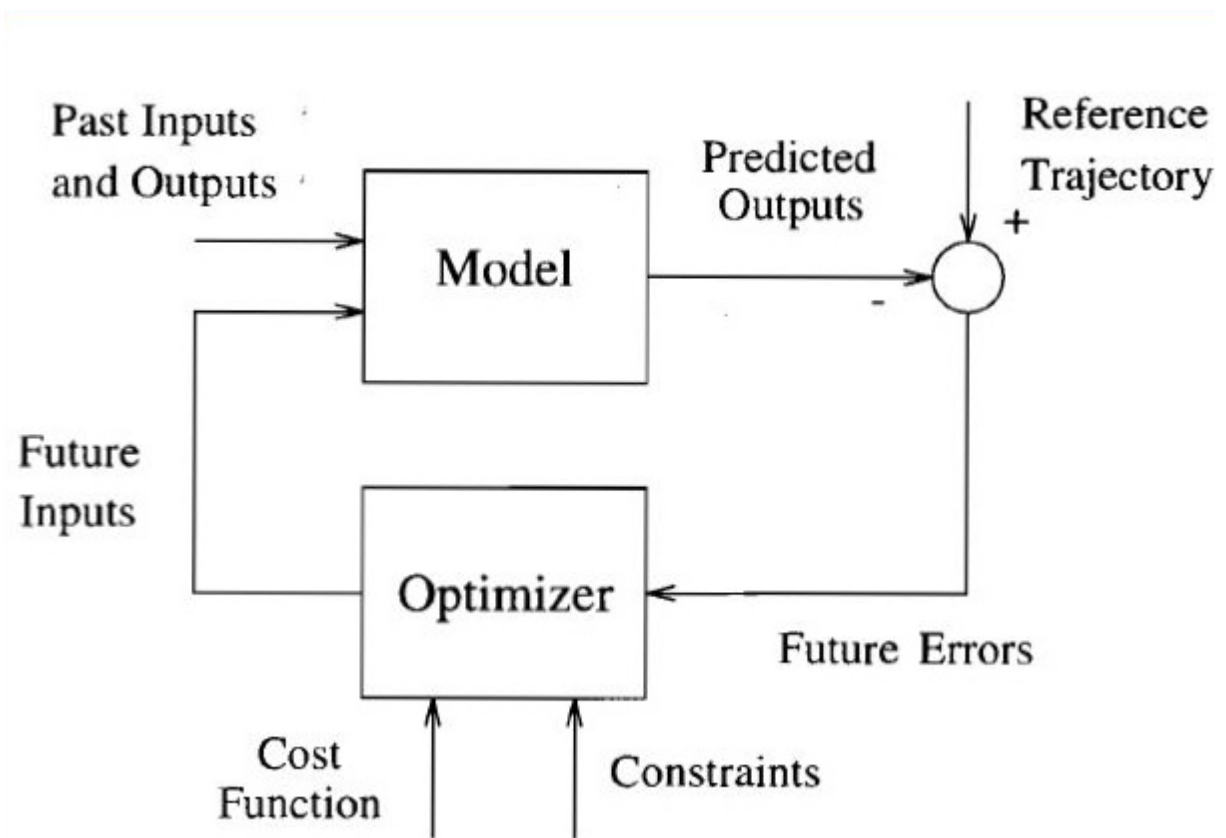


Figura 1.9: Mpc Basic Structure

applicare al sistema; il modello adottato per descrivere il processo, se messo in un'opportuna forma di predizione, ci consente di valutare come si comporterà il sistema in un certo arco temporale. A questo punto è naturale porsi la seguente domanda: Come viene pianificata la legge di controllo sull'orizzonte temporale considerato?

Sebbene i vettori degli Inputs e degli Outputs sono delle informazioni acquisite, questo non risulta altrettanto vero per il vettore delle uscite previste.

2. L'insieme dei successivi segnali di controllo vengono calcolati sulla base di un criterio di ottimizzazione arbitrario in modo tale che l'evoluzione del processo non si allontani di molto dalla traiettoria di riferimento $\mathbf{w}(\mathbf{t}+\mathbf{k})$ per $\mathbf{k} = 1, \dots, \mathbf{N}$ (la quale ad esempio può essere un valore di set-point, ovvero un riferimento costante). L'obiettivo in generale è sempre quello di ridurre il più possibile la norma dell'errore, ma come vedremo tra breve, l'errore in questo tipo di strategia viene concepito in maniera differente rispetto ai classici regolatori PID.

Il criterio adottato generalmente considera la minimizzazione di un funzionale di costo quadratico che penalizza lo scostamento che potrebbe avere il sistema dal comportamento desiderato nella Time Window. In sostanza, l'ottimizzazione è calcolata sugli eventuali errori futuri, ottenuti dalla differenza tra i valori dei segnali di output previsti e la traiettoria di riferimento pianificata. Questo è l'ingrediente fondamentale che caratterizza l'algoritmo che non si basa sugli errori passati del sistema ma su quelli futuri e questo è ciò che contraddistingue tale strategia dai classici regolatori PID. Da questa osservazione si comprende anche il motivo per cui compare il termine 'Predictive' nell'acronimo MPC.

Lo sforzo del controllo è generalmente incorporato nella funzione obiettivo da minimizzare. Se, la funzione è quadratica ed il modello risulta lineare e non vincolato, allora è possibile ottenere una soluzione esplicita al problema di ottimizzazione, altrimenti è necessario ricorrere ad algoritmi di ottimizzazione iterativi come quello del Gradiente (Steepest Gradient Descendent). In alcuni casi vengono fatte delle ipotesi preliminari riguardo la struttura possibile della legge di controllo da implementare, come ad esempio il fatto di essere costante a partire da un certo istante temporale in poi. Ovviamente questo facilita molto il lavoro di ottimizzazione.

3. Il segnale di controllo $\mathbf{u}(t_0 | t_0)$ ² viene inviato al processo mentre i segnali di controllo ottenuti dall'ottimizzazione precedente ($\mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(t_0+\mathbf{N}-1)$) vengono scartati, poichè al successivo istante di campionamento ($t = t_0+1$) si ha disposizione il campione dell'uscita $y(t_0)$ e quindi riparte la procedura dallo step iniziale.

Con questo nuovo valore tutte le sequenze vengono nuovamente aggiornate e sfruttando il principio dell'estensione temporale del RHC l'ottimizzatore mi fornirà il valore di $\mathbf{u}(t_0+1 | t_0+1)$, il quale in linea di principio potrebbe essere differente da $\mathbf{u}(t_0+1 | t_0)$ in seguito alla nuova informazione disponibile.

Possiamo quindi elencare gli ingredienti essenziali per un'algoritmo basato sull'MPC:

1. **Moving Horizon Window:** Una finestra temporale che parte da un'arbitrario istante t_0 fino a $t_0+\mathbf{N}$. La lunghezza della finestra \mathbf{N} rimane costante. Risulta chiaro che t_0 , che definisce l'inizio della finestra di ottimizzazione, viene incrementato ad ogni passo di campionamento.
2. **Prediction Horizon:** Descrive la nostra capacità di prevedere il futuro. Questo parametro corrisponde alla dimensione \mathbf{N} della nostra finestra temporale.
3. **Receding Horizon Control:** Sebbene ad ogni istante temporale si calcoli la traiettoria ottima per la Time Window considerata viene solamente implementato il primo campione della sequenza, scartando i rimanenti.
4. **System Information:** Nella fase di predizione abbiamo bisogno delle informazioni rilevanti relative al sistema fino all'istante t_0 . Queste informazioni possono essere direttamente misurate, oppure se non sono accessibili è possibile stimarle con un'opportuno osservatore dallo stato.
5. **Model:** Un buon modello che consenta di descrivere in modo soddisfacente le dinamiche del sistema. Un buon modello dinamico ovviamente consente di fare una stima accurata delle possibili uscite del sistema.

²dove il pedice zero indica che è l'ingresso calcolato alla prima iterazione per l'istante iniziale $t = t_0$

6. **Optimizer:** Definire un criterio di ottimizzazione per stabilire la miglior decisione da prendere. La variabile della funzione obiettivo è definita dall'errore dato dalla differenza tra la risposta desiderata e la risposta del sistema. Questa funzione obiettivo viene spesso indicata con la lettera J e l'azione di controllo ottima si ottiene minimizzando una funzione di costo nella finestra di ottimizzazione.

Si osserva che la strategia dell'MPC è molto simile a quella di guidare un'automobile.

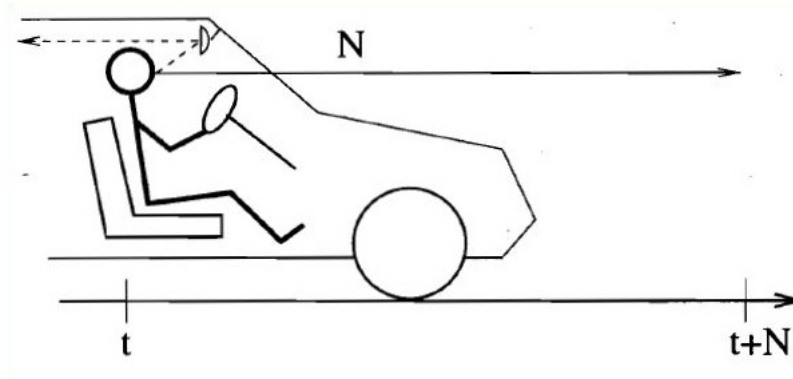


Figura 1.10: Mpc analogy

Il guidatore decide preliminarmente la traiettoria di riferimento desiderata che vuole mantenere per un'orizzonte temporale finito. Tenendo in considerazione le caratteristiche dell'auto (modello mentale della vettura) decide quale azione di controllo (accelerare, frenare e sterzare) prendere per seguire la traiettoria desiderata. Solo la prima azione di controllo viene esercitata ad ogni istante temporale e la procedura viene nuovamente eseguita per le successive decisioni di controllo in modo analogo all'RHC.

Diversamente nei classici sistemi di controllo, come i PID, le azioni di controllo si basano sugli errori commessi negli istanti precedenti. In questo caso considerando sempre l'analogia precedente il modo di guidare l'auto secondo un controllore PID sarebbe equivalente a guidare una macchina unicamente guardando lo specchietto retrovisore³. Comunque non possiamo essere così spietati nei confronti dei PID, poiché comunque l'MPC richiede come informazione la traiettoria di riferimento. Tuttavia possiamo dire che effettivamente le differenze tra le due strategie non possono ritenersi proprio abissali, dipende dalle circostanze e dalla prospettiva di analisi del problema.

Nel sottoparagrafo successivo passeremo in rassegna alcuni dei Modelli adottati nella progettazione di un'algoritmo di controllo basato sull'MPC, cercando di analizzare i vantaggi e gli svantaggi di ciascuno.

1.4.1 Prediction Models

Il modello costituisce l'elemento principale dell'MPC, il quale dovrebbe essere sufficientemente completo per cogliere a pieno le dinamiche del processo ed allo stesso tempo da poter consentire una formulazione altrettanto semplice per calcolare le uscite del sistema ed eventualmente effettuare un'analisi delle sue proprietà strutturali. Dato che l'MPC non è un'unica tecnica ma un'insieme di differenti metodologie che si basano sulla filosofia dell'RHC, esistono diversi modelli nelle varie formulazioni.

Come è stato discusso in precedenza, il ruolo principale del modello è quello di determinare il vettore $\hat{y}(t+k | t)$ per $k = 1, \dots, N$. Quindi questi modelli esprimono la relazione tra le uscite e gli ingressi misurabili, di cui una parte sono le variabili manipolabili, ovvero quelle sulle quali è possibile agire direttamente e le altre sono i disturbi misurabili, che possono essere compensati da un'azione di tipo feed-forward. E' richiesta anche la necessità di fornire un modello dei disturbi che riguarda l'effetto degli ingressi che non possono essere direttamente misurati come il rumore e gli errori di modellazione. Quindi il modello si scinde in due parti: Uno caratterizzato da variabili endogene ed uno da variabili esogene le quali possono essere anche di natura stocastica. Appare evidente che quando si affronta lo studio del sistema occorre analizzare congiuntamente i

³Questo paragone è stato utilizzato all'epoca della commercializzazione di questa tecnologia

due modelli, al fine di poter formulare un modello di predizione adeguato che tenga in considerazione sia la natura deterministica, che ergodica del processo.

Modelli di Predizione

- **Truncated Impulse Response:** Nota anche come sequenza ponderatrice o modello convoluzionale, è molto semplice poichè richiede di misurare la risposta del sistema quando in ingresso riceve un'impulso. La relazione che lega l'uscita all'ingresso è la seguente:

$$y(t) = \sum_{i=1}^N h_i \cdot u(t - i) \quad (1.1)$$

Dove h_i è il campione in uscita quando al processo viene applicato un'impulso unitario.

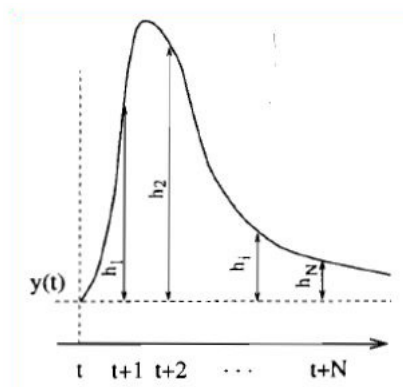


Figura 1.11: Impulse response

Questa sommatoria viene troncata e solamente N valori vengono presi in considerazione, tramite la seguente equazione:

$$y(t) = \sum_{i=1}^N h_i \cdot u(t - i) = H(z^{-1}) \cdot u(t) \quad (1.2)$$

Dove $H(z^{-1}) = h_1 \cdot z^{-1} + h_2 \cdot z^{-2} + \dots + h_N \cdot z^{-N}$ dove z^{-1} è l'operatore di ritardo detto anche backward shift operator.

Uno dei principali svantaggi di questo metodo è legato alla possibilità di modellare solamente processi stabili privi di integratori. Un'altro inconveniente di questo modello è la richiesta di un gran numero di parametri, poichè N è generalmente un valore molto grande che dipende dalle dinamiche del processo e dalla scelta degli intervalli di campionamento. Per esempio, un sistema del I ordine può essere descritto mediante una funzione di trasferimento caratterizzata da 3 parametri (Guadagno, Costante di tempo ed il ritardo o Dead Time) mentre la risposta impulsiva richiede almeno tra i 45-50 coefficienti per descrivere le medesime dinamiche.

Il modello in forma di predizione è caratterizzato dalla seguente equazione:

$$\hat{y}(t + k|t) = \sum_{i=1}^N h_i \cdot u(t + k - i|t) = H(z^{-1}) \cdot u(t + k|t) \quad (1.3)$$

Il motivo per cui il TFIR (Truncated Finite Impulse Response) è molto diffuso nel campo industriale è legato al fatto che risulta molto intuitivo ed esprime in modo chiaro e diretto l'influenza di ogni singola

variabile manipolabile su ciascun output del sistema. Inoltre si può facilmente studiare le specifiche temporali, come: tempo di salita, ritardo, guadagno, etc. Si osserva che se il processo è caratterizzato da più variabili, le differenti uscite riflettono l'effetto degli m ingressi nella seguente maniera:

$$y_j(t) = \sum_{k=1}^m \sum_{i=1}^N h_i^{kj} \cdot u^k(t-i) \quad (1.4)$$

Un'altro dei principali vantaggi di questo metodo è che non è richiesta alcuna informazione preliminare del processo, e quindi il processo di identificazione viene semplificato ed allo stesso tempo consente di descrivere dinamiche complesse come ritardi e sistemi a fase non minima.

- **Step Response:** E' molto simile al TFIR con l'eccezione che l'ingresso è un gradino. Per i sistemi stabili la risposta troncata è:

$$y(t) = y_0 + \sum_{i=1}^N g_i \Delta u(t-i) = y_0 + G(z^{-1}) \cdot (1 - z^{-1}) \cdot u(t) \quad (1.5)$$

dove g_i sono i valori dei campioni dell'uscita quando in ingresso viene applicato il gradino $\Delta u(t) = u(t) - u(t-1)$ come è mostrato nella figura 1.12.

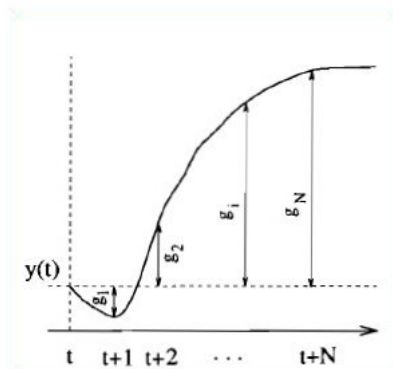


Figura 1.12: Step response

Il valore iniziale y_0 può essere assunto pari a 0 senza perdita di generalità ed esprimere il modello in forma di predizione nella seguente maniera:

$$\hat{y}(t+k|t) = \sum_{i=1}^N g_i \cdot \Delta u(t+k-i|t) \quad (1.6)$$

Analogamente ad un'impulso che può essere considerato come la differenza tra due gradini con un ritardo di un campione, possiamo formulare lo stesso modello facendo le seguenti sostituzioni:

$$h_i = g_i - g_{i-1} - 1$$

$$g_i = \sum_{j=1}^i h_j$$

Questo metodo presenta gli stessi vantaggi e svantaggi della risposta impulsiva. Sebbene in generale le specifiche nel dominio del tempo vengono spesso date in termini di risposta indiciale soprattutto per i sistemi di regolazione.

- **Transfer function:** I modelli basati sulla Funzione di Trasferimento forniscono una descrizione più parsimoniosa delle dinamiche del processo e può essere utilizzata sia per i sistemi stabili che instabili. Detta $G = B/A$ la F.d.T possiamo quindi esprimere l'output nella seguente maniera:

$$A(z^{-1}) \cdot y(t) = B(z^{-1}) \cdot u(t) \quad (1.7)$$

con

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n} \\ B(z^{-1}) &= b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_n \cdot z^{-n} \end{aligned}$$

Pertanto il modello in forma di predizione è dato da:

$$\hat{y}(t+k|t) = \frac{B(z^{-1})}{A(z^{-1})} \cdot u(t+k|t) \quad (1.8)$$

Questa rappresentazione richiede pochi parametri, sebbene sia fondamentale una conoscenza preliminare del processo, in quanto i coefficienti dell'equazione alle differenze inglobano tutte le informazioni relative ai parametri fisici del sistema. In particolare è importante stabilire il grado dei due polinomi.

Tuttavia il suo utilizzo risulta estremamente complicato nel caso di processi con più variabili; per tale motivo in questo lavoro si è preferito un'approccio basato su una rappresentazione ingresso-stato-uscita, che descriveremo dettagliatamente nel capitolo successivo.

Capitolo 2

Discrete-Time MPC

2.1 Introduzione

In questo capitolo verrà proposto un possibile modello di predizione nel caso in cui la progettazione del sistema di controllo parta da una rappresentazione **ingresso-stato-uscita**. Il lavoro è stato sviluppato nella seguente maniera:

1. Nel primo paragrafo verrà impostato il problema nel caso di un sistema SISO (Single-Input and Single-Output State-Space Model). Il modello proposto fa riferimento ad un sistema dinamico nello spazio di stato con un'azione integrale necessaria per ottenere il modello in forma di predizione (al quale faremo riferimento nei seguenti stadi di progettazione ed analisi). E' chiaro che questa scelta non pretende di essere una soluzione definitiva, ma semplicemente un possibile modello di predizione.
2. Una volta scelto il modello di riferimento, si procede alla progettazione del controllore che ha lo scopo di prevedere la sequenza di campioni da applicare nella Time Window considerata. In questa seconda fase verrà affrontato il problema dell'ottimizzazione, che costituisce l'altro elemento essenziale dell'MPC.
3. Sulla base dei risultati ottenuti in seguito all'ottimizzazione sull'orizzonte temporale predefinito, si affronterà il problema del RHC (ovvero su come effettuare l'estensione temporale della finestra). In fine si discuterà delle matrici di guadagno (state feedback gain matrices) e della configurazione ad anello chiuso del controllore.
4. I risultati ottenuti nello stadio di progettazione preliminare verranno quindi estesi nel caso di sistemi con più ingressi e più uscite: MIMO (Multi-Input and Multi-Output).
5. All'interno del framework generale nel quale abbiamo definito i diversi stadi di progettazione dell'MPC, è essenziale la presenza di un'osservatore. Infatti, in un sistema di controllo non è sempre possibile accedere a tutte le variabili di stato che lo caratterizzano; Pertanto affronteremo in questo ultimo stadio il problema della stima.
6. Finalmente considerando congiuntamente l'azione dello Stimatore e del Controllore Predittivo formuleremo il: **State Estimate Predictive Control**.

2.2 State-space Models with Embedded Integrator

Abbiamo più volte sottolineato che il primo passo fondamentale nella progettazione di un sistema basato sul controllo predittivo è la scelta di un modello matematico che descriva il processo. Il modello a cui faremo riferimento è quello nello spazio di stato. A partire dal framework generale che caratterizza un modello in termini di variabili di stato che risulta ovviamente comune a tutte le differenti progettazioni, si procederà alla scelta soggettiva del modello in forma di predizione che si desidera adottare per lo specifico sistema.

2.2.1 Single-input and Single-output System

Consideriamo inizialmente un sistema costituito da un solo ingresso ed una sola uscita. Il modello è descritto dalla classica rappresentazione nello spazio di stato a tempo-discreto:

$$\begin{cases} x_m(k+1) = A_m \cdot x_m(k) + B_m \cdot u(k) \\ y(k) = C_m \cdot x_m(k) \end{cases} \quad (2.1)$$

dove u rappresenta l'ingresso; y l'uscita del processo; x_m è il vettore delle variabili di stato la cui dimensione la indicheremo convenzionalmente con n_1 . E' opportuno osservare che nella classica formulazione generale di un sistema dinamico nello spazio di stato è presente anche un termine che esprime l'influenza diretta dell'ingresso $u(k)$ sull'uscita $y(k)$, ovvero:

$$y(k) = C_m \cdot x_m(k) + D_m \cdot u(k)$$

Tuttavia, in seguito al principio del RHC, nella quale l'informazione sullo stato corrente del processo viene impiegata per la fase di predizione e controllo, abbiamo implicitamente supposto che l'ingresso $u(k)$ non esercita alcun effetto sulla variabile di uscita $y(k)$ nello stesso istante temporale. In parole povere stiamo ipotizzando che il sistema non si trova al limite di causalità, quindi $D_m = 0$.

E' chiaro che il modello nella sua formulazione generale tiene in considerazione solo l'ingresso all'istante k , quindi se desideriamo estendere il nostro modello in modo tale da contenere qualche informazione aggiuntiva sulla sua 'storia' passata, risulterà opportuno introdurre un'integratore che tenga in considerazione i campioni precedenti. Risulta quindi chiaro che il nostro scopo sarà quello di aggiungere l'informazione inerente al comportamento del sistema nell'istante di campionamento precedente e di esprimere la sua transizione allo stadio corrente. Quindi abbiamo:

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

Quindi per esprimere cosa è avvenuto tra l'istante di campionamento precedente e quello corrente consideriamo la differenza tra i due stati del sistema.

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

Per esprimere in maniera compatta la transizione del sistema tra uno stato e quello successivo introduciamo le seguenti differenze per lo stato:

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

mentre per quanto concerne l'ingresso si ha:

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

Queste rappresentano rispettivamente le variazioni dello stato $x_m(k)$ e dell'ingresso $u(k)$ del sistema. Con questa trasformazione è possibile formulare l'equazione di transizione dello stato in termini di 'memoria':

$$\Delta x_m(k+1) = A_m \cdot \Delta x_m(k) + B_m \cdot \Delta u(k) \quad (2.2)$$

Il passo successivo è quello di definire la relazione che lega la variazione $\Delta x_m(k)$ all'uscita del sistema $y(k)$. A questo proposito è necessario definire un nuovo vettore di stato:

$$x(k) = [\Delta x_m(k)^T \quad y(k)]^T$$

analogamente allo stato, esprimiamo la variazione dell'output:

$$y(k+1) - y(k) = C_m \cdot (x_m(k+1) - x_m(k)) = C_m \cdot \Delta x_m(k+1)$$

Quindi sostituendo la relazione (2.2) si ottiene:

$$y(k+1) - y(k) = C_m \cdot A_m \cdot \Delta x_m(k) + C_m \cdot B_m \cdot \Delta u(k) \quad (2.3)$$

A questo punto, considerando la relazione (2.2) e la (2.3) possiamo finalmente formulare il modello esteso (augmented model) al quale d'ora in avanti faremo riferimento:

$$\left\{ \begin{array}{l} \overbrace{\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix}}^{x(k+1)} = \overbrace{\begin{bmatrix} A_m & o_m^T \\ C_m \cdot A_m & 1 \end{bmatrix}}^A \cdot \overbrace{\begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}}^{x(k)} + \overbrace{\begin{bmatrix} B_m \\ C_m \cdot B_m \end{bmatrix}}^B \cdot \Delta u(k) \\ y(k) = \overbrace{\begin{bmatrix} o_m & 1 \end{bmatrix}}^C \cdot \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \end{array} \right. \quad (2.4)$$

dove $o_m = \overbrace{[0 \ 0 \ \dots \ 0]}^{n_1}$. La tripletta di matrici (A,B,C) caratterizza il nostro modello, il quale verrà impiegato nella fase di progettazione del controllo predittivo.

Esempio 2.1 Consideriamo il seguente modello a tempo discreto:

$$\begin{cases} x_m(k+1) = A_m \cdot x_m(k) + B_m \cdot u(k) \\ y(k) = C_m \cdot x_m(k) \end{cases} \quad (2.5)$$

dove le matrici del sistema sono:

$$A_m = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}; \quad B_m = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}; \quad C_m = [1 \ 0]$$

Determiniamo la tripletta (A,B,C) del modello esteso e calcoliamo gli autovalori della matrice A

Soluzione. Dalla (2.4) si osserva che il sistema presenta due v.d.s e quindi $n_1 = 2$ e $o_m = [0 \ 0]$, pertanto otteniamo il seguente modello esteso:

$$\begin{cases} x(k+1) = A \cdot x(k) + B \cdot \Delta u(k) \\ y(k) = C \cdot x(k) \end{cases}$$

dove le matrici del modello esteso sono:

$$A = \begin{bmatrix} A_m & o_m^T \\ C_m \cdot A_m & 1 \end{bmatrix} = \left[\begin{array}{cc|c} 1 & 1 & 0 \\ 0 & 1 & 0 \\ \hline 1 & 1 & 1 \end{array} \right]; \quad B = \begin{bmatrix} B_m \\ C_m \cdot B_m \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1 \\ 0.5 \end{bmatrix};$$

$$C = [o_m \ 1] = [0 \ 0 \ 1]$$

Determiniamo il polinomio caratteristico e gli autovalori della matrice A:

$$\begin{aligned} \rho(\lambda) &= \det(\lambda I - A) &&= \det \begin{bmatrix} \lambda I - A_m & o_m^T \\ -C_m \cdot A_m & (\lambda - 1) \end{bmatrix} \\ &= (\lambda - 1) \det(\lambda I - A_m) \\ &= (\lambda - 1)^3 \end{aligned}$$

Pertanto, abbiamo tre autovalori in $\lambda = 1$. Tra questi, due appartengono al sistema reale che risultano in cascata ed uno disaccoppiato che fa parte del modello esteso.

Nell'appendice è possibile trovare il codice Matlab che consente di ottenere il Modello Esteso a partire da una rappresentazione del sistema nello spazio di stato a tempo-continuo

2.3 Predictive Control within One Optimization Window

Una volta definito il modello matematico, lo stadio successivo di progettazione di un sistema di controllo predittivo consiste nel determinare una procedura di calcolo che consenta di prevedere le uscite del processo ed il segnale di controllo che verrà applicato al sistema al variare delle matrici (A,B,C) che caratterizzano il nostro modello.

Nei sottoparagrafi seguenti indicheremo con k_i l'istante temporale corrente e con N_p la lunghezza della finestra di ottimizzazione ed il numero dei campioni considerati in suddetto intervallo (Ovvero stiamo suddividendo l'asse temporale con un'unità pari al passo di campionamento).

2.3.1 Prediction of State and Output Variables

Partendo dall'ipotesi che all'istante temporale k_i ($k_i > 0$) sia possibile misurare direttamente lo stato del sistema, abbiamo quindi a nostra disposizione il vettore $x(k_i)$ che fornisce l'informazione attuale sullo stato del processo. Il caso più generale, in cui tale informazione non sia direttamente accessibile verrà affrontato più avanti.

Indichiamo la traiettoria di controllo da applicare nella Time Window nel seguente modo:

$$\Delta u(k_i), \quad \Delta u(k_i + 1), \quad \dots, \quad \Delta u(k_i + N_c - 1).$$

Abbiamo quindi espresso l'ingresso come differenza tra l'azione di controllo tra due istanti consecutivi, tale scelta non è casuale, poichè tra breve vedremo che ci servirà per calcolare lo stato del sistema. In tale notazione, compare la lettera N_c che prende il nome di Control Horizon (Orizzonte di Controllo), il quale ha un'importanza rilevante poichè dalla sua scelta dipende l'impostazione del numero dei parametri utilizzati per caratterizzare la sequenza di controllo futura.

A partire dalla conoscenza di $x(k_i)$, la predizione delle variabili di stato verrà effettuata per un numero di campioni pari a N_p , dove N_p prende il nome di Prediction Horizon, il quale ricordo che rappresenta anche la lunghezza della finestra di ottimizzazione. Le variabili di stato negli istanti temporali futuri verranno indicate con la seguente notazione:

$$x(k_i + 1|k_i), \quad x(k_i + 2|k_i), \quad \dots, \quad x(k_i + m|k_i), \quad \dots, \quad x(k_i + N_p|k_i).$$

dove $x(k_i + m|k_i)$ è lo stato previsto per il sistema nell'istante temporale $(k_i + m)$ in base all'informazione corrente del processo $x(k_i)$.

L'orizzonte di controllo N_c viene scelto minore oppure uguale all'orizzonte di predizione N_p . Sulla base del modello nello spazio di stato (A,B,C), possiamo calcolare la successione degli stati del sistema facendo riferimento all'insieme dei parametri che caratterizzano la sequenza dei campioni futuri. In sostanza, a partire dalla scelta della traiettoria di controllo che verrà teoricamente impostata (poichè ricordo che solamente il primo campione della sequenza viene implementato), possiamo ricavare le informazioni relative agli stati successivi del sistema.

Ingresso-Stato:

$$\begin{aligned} x(k_i + 1|k_i) &= Ax(k_i) + B\Delta u(k_i) \\ x(k_i + 2|k_i) &= Ax(k_i + 1|k_i) + B\Delta u(k_i + 1) \\ &= A^2x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i + 1) \\ &\vdots \\ x(k_i + N_p|k_i) &= A^{N_p}x(k_i) + A^{N_p-1}B\Delta u(k_i) + A^{N_p-2}B\Delta u(k_i + 1) + \dots + A^{N_p-N_c}B\Delta u(k_i + N_c - 1). \end{aligned} \tag{2.6}$$

Come si può osservare abbiamo espresso il generico stato in termini dello stato corrente e della sequenza di ingresso.

Considerando ora i valori delle v.d.s, possiamo calcolare l'uscita prevista mediante una semplice sostituzione.

Stato-Uscita:

$$\begin{aligned}
y(k_i + 1|k_i) &= CAx(k_i) + CB\Delta u(k_i) \\
y(k_i + 2|k_i) &= CA^2x(k_i) + CAB\Delta u(k_i) + CB\Delta u(k_i + 1) \\
y(k_i + 3|k_i) &= CA^3x(k_i) + CA^2B\Delta u(k_i) + CAB\Delta u(k_i + 1) + CB\Delta u(k_i + 2) \\
&\vdots \\
y(k_i + N_p|k_i) &= CA^{N_p}x(k_i) + CA^{N_p-1}B\Delta u(k_i) + CA^{N_p-2}B\Delta u(k_i + 1) + \dots + CA^{N_p-N_c}B\Delta u(k_i + N_c - 1).
\end{aligned} \tag{2.7}$$

Anche in questo caso si osserva che data la natura ricorsiva delle equazioni precedenti, tutte le variabili di previsione dipendono dall'informazione dello stato corrente del processo e dall'andamento del controllo futuro $\Delta u(k_i + j)$, dove $j = 0, 1, \dots, N_c - 1$.

Quindi, al fine di poter esprimere l'uscita in maniera compatta, in una struttura dati che consenta di effettuare suddetto calcolo; Esprimiamo le due grandezze d'interesse (Output ed Input) mediante i seguenti vettori:

$$\begin{aligned}
Y &= [y(k_i + 1|k_i) \quad y(k_i + 2|k_i) \quad y(k_i + 3|k_i) \quad \dots \quad y(k_i + N_p|k_i)]^T \\
\Delta U &= [\Delta u(k_i) \quad \Delta u(k_i + 1) \quad \Delta u(k_i + 2) \quad \dots \quad \Delta u(k_i + N_c - 1)]^T
\end{aligned}$$

Dove nel caso di un singolo ingresso ed un'uscita, la dimensione di Y è pari a $N_p \times 1$ mentre quella di ΔU è $1 \times N_c$. Raggruppando le relazioni della (2.7) in una forma matriciale compatta otteniamo:

$$Y = Fx(k_i) + \Phi \Delta U \tag{2.8}$$

Dove le matrici F e Φ sono definite nella seguente maniera:

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

Abbiamo quindi completato lo stadio di progettazione che ci ha consentito di esprimere il modello in una forma di previsione opportuna per il calcolo dell'uscita. Si nota che in questa struttura compaiono le matrici del modello esteso (A,B,C) e le loro potenze; Tuttavia non è stato ancora definito il modo di determinare ΔU . Quindi nel sottoparagrafo successivo determineremo come avviene la scelta della traiettoria di controllo, che una volta nota ci consentirà di calcolare il vettore Y .

2.3.2 Optimization

Sulla base della traiettoria desiderata, impostata dal Planner, definiamo con $r(k_i)$ il segnale di set-point all'istante di campionamento k_i . L'obiettivo del sistema di controllo predittivo è quello di pilotare l'uscita prevista del sistema il più possibile vicina al segnale di set-point, entro l'orizzonte temporale di previsione. In seguito per semplicità, assumeremo che il segnale di set-point rimanga costante nella finestra di ottimizzazione (In qualche modo possiamo considerare come un pseudo-problema di regolazione, poichè questa ipotesi in realtà non può essere altrettanto valida se il nostro orizzonte temporale sia sufficientemente grande da consentire variazioni significative del riferimento). Questo stadio di progettazione si traduce quindi nel seguente obiettivo:

Trovare i valori ottimali dei parametri che caratterizzano il vettore ΔU che consentono di rendere minimo lo scarto definito dalla funzione di errore, data dalla differenza tra il segnale di set-point e l'uscita prevista.

Supposto che il vettore che contiene i dati relativi al segnale di set-point sia definito nel seguente modo:

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

Dove $r(k_i)$ è un generico scalare che rappresenta il nostro riferimento costante. (Quindi il segnale di controllo applicato all'istante k_i rimane lo stesso per tutto l'orizzonte temporale considerato e per tale motivo viene moltiplicato per un vettore composto da un numero di elementi unitari pari al numero di campioni della Time window).

Possiamo definire la funzione di costo J che rispecchia l'obiettivo del controllo, nella seguente maniera:

$$J = \overbrace{(R_s - Y)^T (R_s - Y)}^I + \overbrace{\Delta U^T \bar{R} \Delta U}^{II} \quad (2.9)$$

Per quanto riguarda il primo termine è direttamente collegato allo scopo principale della funzione obiettivo, ovvero quello di minimizzare gli errori tra l'uscita prevista ed il segnale di set-point. Si vuole in particolare far notare che avremmo potuto scegliere altri criteri rispetto allo scarto quadratico dell'errore per penalizzare lo scostamento dal riferimento desiderato, ma tale scelta presenta i seguenti vantaggi:

- In primo luogo, errori grandi incidono significativamente nella valutazione finale: un'errore che risulta pari al doppio di uno precedente incide con un peso 4 volte maggiore.
- L'altro vantaggio, è la semplice trattabilità dal punto di vista matematico. Infatti, vedremo che attraverso gli strumenti dell'Algebra Lineare il calcolo dell'ottimizzazione si riduce alla soluzione di un'insieme di equazioni lineari.
- Inoltre, se avessimo considerato semplicemente gli errori, non sarebbe stata una grande idea in quanto tali errori possono essere anche negativi e quindi richiederebbe di definire ulteriormente il loro significato all'interno dell'applicazione. Mentre in questa circostanza è facilmente interpretabile in quanto corrisponde alla sua norma e quindi è possibile valutare il comportamento qualitativo del sistema semplicemente analizzando il suo andamento temporale.

E' chiaro che esistono approcci alternativi per descrivere quanto una funzione risulti differente o simile ad un'altra scelta come riferimento, che spesso però richiedono conoscenze avanzate di teoria della misura, analisi funzionale e spazi funzionali. In genere questi metodi risultano più adatti alla realizzazione di sistemi di controllo ad anello aperto, dove in generale è richiesto un'approccio matematico più formale che consenta di garantire sotto opportune condizioni la validità della soluzione trovata.

Il secondo termine, rispecchia invece l'energia sprigionata dal segnale di controllo utilizzata per mantenere la traiettoria effettiva del sistema il più possibile vicina a quella desiderata. \bar{R} è una matrice diagonale definita nel seguente modo: $\bar{R} = r_w \cdot I$ dove la matrice identità ha dimensioni N_c e ($r_w \geq 0$) è un parametro di Tuning utilizzato per regolare la performance del sistema a ciclo chiuso. Nella circostanza in cui $r_w = 0$ la relazione (2.9) riflette la situazione in cui non si è interessati allo sforzo del controllo e l'obiettivo sarà unicamente quello di rendere la norma dell'errore il più piccolo possibile (ovvero il contributo del I termine che compare nel funzionale di costo).

Naturalmente se consideriamo la situazione opposta in cui i pesi r_w (la lettera w sta per weight) hanno valori elevati, stiamo penalizzando l'applicazione di segnali con piccole norme. Risulta evidente che norme sufficientemente elevate dei segnali di ingresso danno un contributo maggiore al secondo termine ma allo stesso tempo consentono di inseguire meglio la traiettoria di riferimento. Viceversa, segnali troppo deboli, sebbene incidano poco sul secondo termine, fanno allontanare il sistema dal comportamento desiderato e quindi penalizzano l'errore quadratico. Quindi l'obiettivo generale è quello di raggiungere il giusto Trade-Off (compromesso), tra non sollecitare troppo il sistema ed allo stesso tempo non degradare la sua performance (prestazione). Generalmente

in ambito industriale tale scelta è dettata dalla produttività del processo e quindi al tempo di ciclo delle operazioni che lo caratterizzano e ne definiscono la performance ed al costo di manutenzione dei macchinari legato in gran parte alla loro usura. Per definire l'efficienza del sistema viene quindi effettuata un'analisi preliminare adottando diversi metodi, quali ad esempio: L'analisi costo-benefici.

Comunque, una volta preimpostati i valori di R_s^T e r_w in base alle proprie esigenze; il problema di ottimizzazione è ben circoscritto. Quindi a partire dalla relazione (2.8) possiamo sostituire Y all'interno del funzionale di costo definito dalla (2.9) ed ottenere:

$$J = (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - 2\Delta U^T \Phi^T (R_s - Fx(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U. \quad (2.10)$$

ovviamente, al fine di poter minimizzare J ne calcoliamo la derivata per trovarne i punti stazionari:

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T (R_s - Fx(k_i)) + 2(\Phi^T \Phi + \bar{R}) \Delta U \quad (2.11)$$

Le condizioni necessarie di minimo si ottengono imponendo:

$$\frac{\partial J}{\partial \Delta U} = 0$$

A partire da questa equazione omogenea si ottiene la soluzione ottima per il segnale di controllo:

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i)) \quad (2.12)$$

Risulta chiaro che occorre fare l'ipotesi che la matrice $(\Phi^T \Phi + \bar{R})$ sia invertibile o comunque che non risulti malcondizionata. In ottimizzazione tale matrice prende il nome di Hessiana. Per quanto riguarda il secondo termine si osserva che R_s altro non è che il vettore contenente le informazioni relative al set-point espresso dalla relazione:

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

dove

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

Sebbene si considerino delle condizioni necessarie di minimo si fa notare che la struttura della funzione di costo è governata da due termini quadratici e quindi un minimo locale necessariamente è un minimo globale (in quanto è una combinazione di due funzioni convesse) naturalmente i problemi sorgono quando si considerano dei vincoli sul processo. La soluzione ottima del segnale di controllo è legata al segnale di set-point $r(k_i)$ e la variabile di stato $x(k_i)$ mediante la seguente equazione:

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (\bar{R}_s r(k_i) - Fx(k_i)) \quad (2.13)$$

Finalmente possiamo applicare i risultati ottenuti per predire l'uscita del sistema come vedremo tra breve nell'esempio (2.2).

Esempio 2.2 Consideriamo il seguente modello SISO del primo ordine definito dalla rappresentazione di stato a tempo-discreto :

$$\begin{cases} x_m(k+1) = a \cdot x_m(k) + b \cdot u(k) \\ y(k) = c \cdot x_m(k) \end{cases} \quad (2.14)$$

dove $a = 0.8$, $b = 0.1$ e $c = 1$ sono scalari. Determiniamo quindi la rappresentazione del modello esteso nello spazio di stato, considerando un orizzonte di predizione pari a $N_p = 10$ ed un orizzonte di controllo $N_c = 4$;

successivamente calcoleremo i componenti necessari per predire l'uscita del sistema Y e le quantità $\Phi^T \Phi$, $\Phi^T F$ e $\Phi^T \overline{R_s}$. Supposto che all'istante temporale k_i ($k_i = 10$ in questo esempio), $r(k_i) = 1$ e il vettore di stato $x(k_i) = [0.1 \ 0.2]^T$, (occorre fare attenzione che il vettore $x(k_i)$ fa riferimento rispettivamente alla variazione che ha subito la variabile di stato, nel passare dallo stato precedente a quello corrente ed al valore che assume nell'istante corrente) calcoliamo la soluzione ottima ΔU nella circostanza in cui $r_w = 0$ e $r_w = 10$, e confrontiamo le due soluzioni.

Soluzione.: Il modello esteso nello spazio di stato è:

$$\begin{cases} \begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} a & 0 \\ a & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} b \\ b \end{bmatrix} \cdot \Delta u(k) \\ y(k) = [0 \ 1] \cdot \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \end{cases} \quad (2.15)$$

A partire dalla seguente relazione:

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

dove le matrici F e Φ sono rispettivamente:

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

Quindi considerando la definizione delle due matrici è possibile calcolare i loro elementi singolarmente. Tuttavia, dato che da un punto di vista pratico tale algoritmo deve girare su un calcolatore; risulta più efficiente definire gli elementi che costituiscono le singole colonne in modo ricorsivo, (ovvero in funzione di quelli calcolati precedentemente). Pertanto i coefficienti nella F e nella Φ sono calcolati nel seguente modo:

$$\begin{aligned} CA &= [s_1 \ 1] \\ CA^2 &= [s_2 \ 1] \\ CA^3 &= [s_3 \ 1] \\ &\vdots \\ CA^k &= [s_k \ 1] \end{aligned} \quad (2.16)$$

dove $s_1 = a$, $s_2 = a^2 + s_1$, \dots , $s_k = a^k + s_{k-1}$, e per quanto riguarda la matrice Φ si osserva che presenta una struttura algebrica particolare, ovvero è caratterizzata dal fatto che gli elementi presenti sulle diagonali principali sono uguali tra loro. Questo tipo di matrice prende il nome di matrice *Toeplitz*. Pertanto è sufficiente calcolare gli elementi della prima colonna per ricavare quelli delle successive. Nell'Appendice ho riportato il codice che implementa la procedura che consente di calcolare tale matrice:

$$\begin{aligned}
CB &= g_0 = b \\
CAB &= g_1 = ab + g_0 \\
CA^2B &= g_2 = a^2b + g_1 \\
&\vdots \\
CA^{k-1}B &= g_{k-1} = a^{k-1}b + g_{k-2} \\
CA^k B &= g_k = a^k b + g_{k-1}.
\end{aligned} \tag{2.17}$$

Quindi, a partire dai parametri che definiscono il sistema ($a = 0.8$ e $b = 0.1$) e dai parametri di controllo ($N_p = 10$, $eN_c = 4$), possiamo ora calcolare le matrici costitutive della legge di controllo predittivo:

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

$$\Phi^T F = \begin{bmatrix} 9.2325 & 3.2147 \\ 8.3259 & 2.7684 \\ 7.2927 & 2.3355 \\ 6.1811 & 1.9194 \end{bmatrix}; \Phi^T \overline{R_s} = \begin{bmatrix} 3.2147 \\ 2.7684 \\ 2.3355 \\ 1.9194 \end{bmatrix}$$

Si osserva che il vettore $\Phi^T \overline{R_s}$ risulta identico all'ultima colonna della matrice $\Phi^T F$. Il motivo è legato al fatto che l'ultima colonna di F è identica a $\overline{R_s}$, questa osservazione è stata evidenziata appositamente per coloro che desiderino verificare la correttezza dell'algoritmo implementato. Comunque più avanti presenterò una possibile implementazione basata sul calcolo ricorsivo definito in precedenza.

All'istante temporale $k_i = 10$, il vettore di stato $x(k_i) = [0.1 \quad 0.2]^T$. Nel caso in cui $r_w = 0$, l'errore che si ottiene confrontando l'uscita prevista Y con il riferimento desiderato R_s viene ridotto senza tener conto della variazione dell'ingresso. Allora, la sequenza di campioni ottima sull'orizzonte di controllo è data dal seguente calcolo:

$$\Delta U = (\Phi^T \Phi)^{-1} (\Phi^T R_s - \Phi^T F x(k_i)) = [7.2 \quad -6.4 \quad 0 \quad 0]^T.$$

Possiamo notare che senza ponedrare la variazione dell'ingresso, gli ultimi due elementi $\Delta u(k_i + 2) = 0$ e $\Delta u(k_i + 3) = 0$, mentre i primi due elementi che esprimono la variazione del segnale di controllo nei due istanti iniziali dell'horizon control presentano un valore piuttosto elevato.

Questo si può constatare dal grafico riportato nella figura (2.1); che mostra la variazione che subisce la variabile di stato in seguito all'applicazione del segnale di controllo ottimo. In tale andamento si nota che nella fase iniziale il sistema presenta una variazione significativa di x_m , legato alla derivata del segnale di controllo; In fine, una volta che si è stabilizzato il valore dell'uscita sul segnale di set-point, si osserva che il valore di Δx_m e Δu decade a zero e quindi il processo ha raggiunto la condizione operativa desiderata. Questo risulta evidente analizzando l'andamento dell'errore quadratico in figura (2.2), che nel nostro caso rappresenta la funzione obiettivo da minimizzare.

Esaminiamo ora l'effetto che subisce la soluzione ottima di controllo, quando $r_w = 10$. La soluzione ottima ΔU si ottiene a partire dalla seguente relazione:

$$\begin{aligned}
\Delta U &= (\Phi^T \Phi + 10 \times I)^{-1} (\Phi^T R_s - \Phi^T F x(k_i)) \\
&= [0.1269 \quad 0.1034 \quad 0.0829 \quad 0.065]^T.
\end{aligned} \tag{2.18}$$

Confrontando tale soluzione con quella precedente, osserviamo che la variazione del segnale di controllo presenta una riduzione significativa per quanto riguarda i primi due elementi, inoltre le ultime due componenti non sono più pari a zero. Questo comportamento è riportato in figura (2.3), la quale mostra un'andamento più smooth (ovvero addolcito o se vogliamo definirlo in termini analitici: una continuità regolare caratterizzata dall'assenza di brusche variazioni) dell'ingresso che appunto influisce sull'uscita del sistema in modo meno

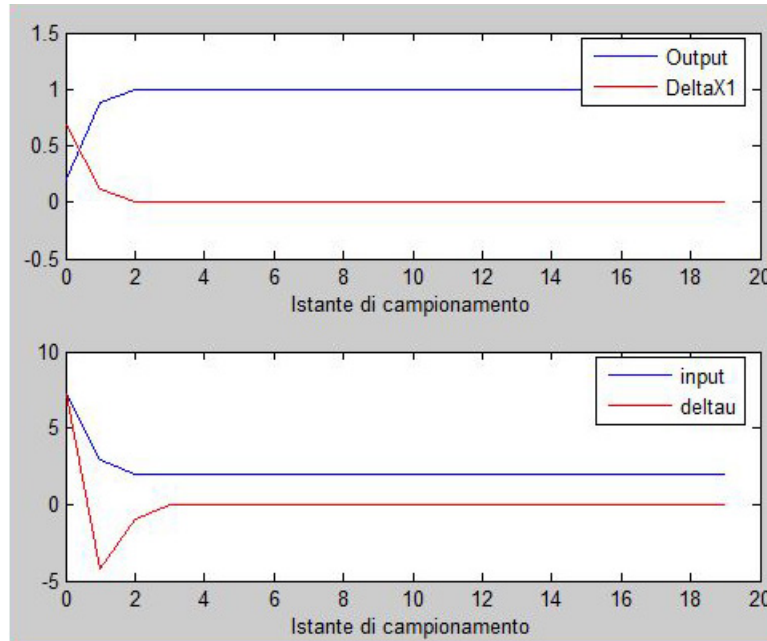


Figura 2.1: Mpc with no weigth on Δu

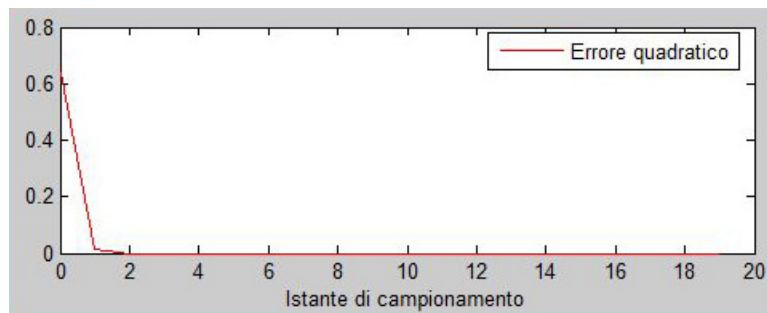


Figura 2.2: e^2 Mpc with no weigth on Δu

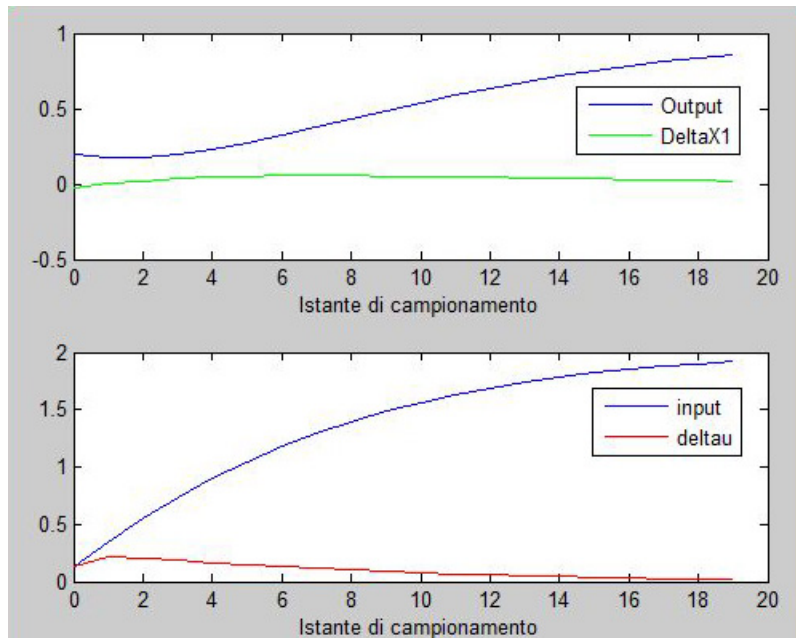


Figura 2.3: Mpc with weigth on Δu

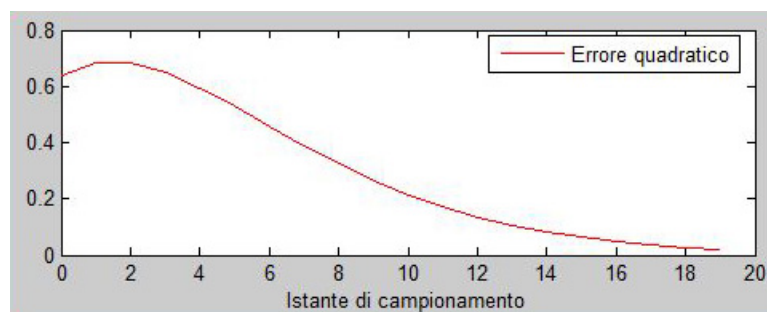


Figura 2.4: e^2 Mpc with weigth on Δu

aggressivo da come si può constatare dalla variazione della variabile di stato. Risulta chiaro che il sistema impiegherà più tempo per raggiungere il valore di set-point, inoltre come è stato detto in precedenza, l'azione del controllo si riflette anche sull'andamento della variabile di stato. Infatti, sebbene la variazione non si sia ancora definitivamente assestata è possibile osservare che rispetto al caso precedente, abbiamo una sollecitazione meno intensa e questo si riflette ovviamente in una minore sovraelongazione del sistema a discapito però di un maggiore tempo di salita.

In sintesi, da questo confronto emerge che:

Se desideriamo muoverci con cautela e quindi esercitare un'azione di controllo più delicata, allora l'ingresso impiegherà più tempo per far raggiungere al sistema la situazione di regime (*i.e.*, i valori di ΔU decrescono più lentamente), poichè l'energia considerata per ottimizzare l'azione di controllo viene distribuita su un'orizzonte temporale maggiore. Possiamo verificare questa osservazione aumentando il valore di N_c fino a 9, mantenendo lo stesso valore di $r_w = 10$. (E' importante ricordare che non possiamo arbitrariamente scegliere il valore di N_c in quanto ovviamente è limitato dall valore N_p che rappresenta l'orizzonte di predizione).

$$\Delta U^T = [0.1227 \quad 0.0993 \quad 0.0790 \quad 0.0614 \quad 0.0463 \quad 0.0334 \quad 0.0227 \quad 0.0139 \quad 0.0072] \quad (2.19)$$

Confrontando quindi la (2.18), ovvero il caso con $N_c = 4$, con la (2.19) ($N_c = 9$), si osserva che i primi quattro parametri di controllo di ΔU risultano leggermente differenti tra di loro.

Matlab Computation of MPC Gains

Sebbene il codice relativo al calcolo dei guadagni (per intenderci: i numeretti che andiamo a mettere all'interno della legge di controllo per ottenere il vettore del segnale da applicare al sistema) sia riportato nell'Appendice, ho volutamente descritto i passi che commentano dettagliatamente suddetto codice:

Tutorial 1 : Nel file `mpcgain.m` ho realizzato una funzione Matlab per il calcolo delle matrici $\Phi^T \Phi$, $\Phi^T F$, $\Phi^T \overline{R_s}$. Il passo fondamentale è quello di creare le due matrici F e Φ . La struttura algebrica della matrice Φ è quella di una matrice *Toeplitz*, ovvero vuol dire semplicemente che gli elementi presenti sulle diverse diagonali sono uguali tra loro. Quindi è possibile costruire suddetta matrice a partire semplicemente dalla conoscenza della sua prima colonna e le restanti sono ottenute direttamente shiftando quella precedente. (Semplice a dirsi...una volta implementata...semplice a calcolarla!).

Step by Step

1. Il primo passo consiste nel creare il modello esteso necessario per progettare l'MPC. I parametri che caratterizzano i dati di ingresso della funzione sono ovviamente le matrici della rappresentazione nello spazio di stato che modellano il processo (A_p , B_p , C_p), il pedice p sta ad indicare appunto il processo effettivo, e l'orizzonte di predizione N_p e quello di controllo N_c . Vedere codice in appendice relativo allo step 1.
2. il secondo step consiste nel calcolare le matrici F e Φ che, come abbiamo già accennato, presentano una struttura particolare. Sfruttando tale caratteristica è possibile ottenere i restanti elementi senza procedere ad un loro calcolo diretto. Vedere codice in appendice relativo allo step 2.
3. il terzo ed ultimo step consiste semplicemente nel calcolare i guadagni definiti dalla legge di controllo. Vedere codice in appendice relativo allo step 3.

A partire dalla funzione `mpcgain.m` è possibile realizzare il nostro 'primo' sistema di controllo predittivo. In appendice è riportato il codice relativo all'esempio (2.2); Comunque (sebbene il codice sia ampiamente commentato) suggerisco di continuare la lettura del paragrafo successivo per avere una maggiore percezione dei parametri di controllo utilizzati. Il codice fa riferimento al caso non ponderato, comunque si può ottenere il caso pesato semplicemente impostando nella legge di controllo un valore differente del peso.

2.4 Receding Horizon Control state space model

Sebbene i parametri ottimali presenti nel vettore ΔU rappresentino la variazione del segnale di controllo lungo l'intera Time Window $\Delta u(k_i), \Delta u(k_i + 1), \Delta u(k_i + 2), \dots, \Delta u(k_i + N_c - 1)$, (dove è ormai chiaro che è possibile avere la variazione fino all'istante temporale $N_c - 1$ in quanto ad N_c non ho calcolato il segnale da applicare) rispettando il principio del RHC, noi implementiamo solo il primo campione della sequenza calcolata, *i.e.*, $\Delta u(k_i)$, ignorando i restanti valori. Non appena giunge il periodo di campionamento successivo, le misure più recenti vengono acquisite per ricavare il vettore di stato $x(k_i + 1)$ che verrà impiegato per calcolare la nuova successione degli ingressi. Questa routine di calcolo viene ripetuta in real time in modo da determinare la legge di controllo su un'orizzonte temporale infinito.

Esempio 2.3 Per avere una comprensione di come questa estensione temporale venga realizzata, riprendiamo l'esempio (2.2), dove il sistema del primo ordine è definito dalla seguente rappresentazione di stato:

$$x_m(k+1) = 0.8x_m(k) + 0.1u(k)$$

Consideriamo il caso in cui la matrice dei pesi sia nulla, ovvero $r_w = 0$. Le condizioni iniziali sono rispettivamente $x(10) = [0.1 \ 0.2]^T$ e $u(9) = 0$. Dove ricordo che il vettore $x(k_i)$ fa riferimento rispettivamente alla variazione che ha subito la variabile di stato, nel passare dallo stato precedente a quello corrente ($\Delta x_m(k)$) ed al valore che assume l'uscita nell'istante corrente ($y(k)$).

Soluzione: All'istante di campionamento iniziale $k_i = 10$, avevamo calcolato il segnale di controllo ottimo pari a $\Delta u(10) = 7.2$, che appunto esprime la variazione subita dall'ingresso nel passare dall'istante precedente a quello attuale ($u(10) - u(9)$). Pertanto, a partire dalla conoscenza dell'ingresso all'istante precedente $u(9) = 0$, possiamo quindi calcolare il segnale di controllo da applicare al sistema, $u(10) = u(9) + \Delta u(10) = 7.2$ e considerando $x_m(10) = y(10) = 0.2$, possiamo calcolare il valore che assume la variabile di stato del sistema nel successivo step della simulazione:

$$\begin{aligned} x_m(11) &= 0.8x_m(10) + 0.1u(10) = 0.88 \\ &= 0.8 \cdot 0.2 + 0.1 \cdot 7.2 \end{aligned} \quad (2.20)$$

Ora l'istante corrente è pari a $k_i = 11$, quindi le **informazioni a nostra disposizione sono:**

$$\begin{aligned} u(10) &= 7.2 \\ x_m(10) &= 0.2 \\ x_m(11) &= 0.88 \\ \Delta x_m(11) &= x_m(11) - x_m(10) = 0.68 \\ y(11) &= 0.88 \end{aligned} \quad (2.21)$$

Pertanto il vettore che tiene conto di tali informazioni è $\mathbf{x}(11) = [0.68 \ 0.88]^T$.

Possiamo quindi calcolare la sequenza di controllo ottima a partire dai guadagni definiti dalla legge di controllo:

$$\Delta U = (\Phi^T \Phi)^{-1} (\Phi^T R_s - \Phi^T F x(11)) = [-4.24 \ -0.96 \ 0 \ 0]^T \quad (2.22)$$

Questo ci consente quindi di calcolare il segnale di controllo ottimo da applicare all'istante corrente (o più correttamente nel periodo di campionamento corrente) che è pari a:

$$\begin{aligned} u(11) &= u(10) + \Delta u(11) = 2.96 \\ &= 7.2 - 4.24 = 2.96 \end{aligned} \quad (2.23)$$

Una volta implementato suddetto controllo possiamo determinare lo stato del sistema del periodo di campionamento successivo:

$$\begin{aligned} x_m(12) &= 0.8x_m(11) + 0.1u(11) = 1 \\ &= 0.8 \cdot 0.88 + 0.1 \cdot 2.96 = 1 \end{aligned} \quad (2.24)$$

All'istante $\mathbf{k}_i = \mathbf{12}$, la nuova informazione relativa al processo è quindi: le **informazioni a nostra disposizione sono**:

$$\begin{aligned} u(11) &= 2.96 \\ x_m(11) &= 0.88 \\ x_m(12) &= 1 \\ \Delta x_m(12) &= x_m(12) - x_m(11) = 0.12 \\ y(12) &= 1 \end{aligned} \tag{2.25}$$

Pertanto il vettore che tiene conto di tali informazioni è $\mathbf{x}(11) = [0.12 \ 1]^T$.

Possiamo quindi calcolare la sequenza di controllo ottima a partire dai guadagni definiti dalla legge di controllo:

$$\Delta U = (\Phi^T \Phi)^{-1} (\Phi^T R_s - \Phi^T F x(12)) = [-0.96 \ 0 \ 0 \ 0]^T \tag{2.26}$$

Questo ci consente quindi di calcolare il segnale di controllo ottimo da applicare all'istante corrente (o più correttamente nel periodo di campionamento corrente) che è pari a:

$$\begin{aligned} u(12) &= u(11) + \Delta u(12) = 2 \\ &= 2.96 - 0.96 = 2 \end{aligned} \tag{2.27}$$

Una volta implementato suddetto controllo possiamo determinare lo stato del sistema del periodo di campionamento successivo:

$$\begin{aligned} x_m(13) &= a \cdot x_m(12) + b \cdot u(12) = 1 \\ &= 0.8 \cdot 1 + 0.1 \cdot 2 = 1 \end{aligned} \tag{2.28}$$

All'istante $\mathbf{k}_i = \mathbf{13}$, la nuova informazione relativa al processo è quindi: le **informazioni a nostra disposizione sono**:

$$\begin{aligned} u(12) &= 2 \\ x_m(12) &= 1 \\ x_m(13) &= 1 \\ \Delta x_m(13) &= x_m(13) - x_m(12) = 0.0 \\ y(13) &= 1 \end{aligned} \tag{2.29}$$

Pertanto il vettore che tiene conto di tali informazioni è $\mathbf{x}(11) = [0 \ 1]^T$.

Possiamo quindi calcolare la sequenza di controllo ottima a partire dai guadagni definiti dalla legge di controllo:

$$\Delta U = (\Phi^T \Phi)^{-1} (\Phi^T R_s - \Phi^T F x(13)) = [0 \ 0 \ 0 \ 0]^T \tag{2.30}$$

Questo esempio mostra le differenze tra i parametri dei vettori ΔU calcolati ai differenti istanti temporali, in base alla filosofia del RHC. Si nota inoltre, come piano piano che l'uscita si avvicina al valore desiderato il vettore ΔU tenda a zero.

2.4.1 Closed-loop Control System

Riprendendo l'esempio (2.3) possiamo mettere in evidenza un'altro aspetto importante relativo alla legge di controllo. Se esaminiamo attentamente tale esempio, ci accorgiamo che al generico istante k_i , il vettore che contiene i parametri ottimali ΔU è caratterizzato dal contributo di due termini:

$$\begin{aligned}\Delta U &= (\Phi^T \Phi)^{-1} (\Phi^T R_s - \Phi^T F x(k_i)) \\ &= \overbrace{(\Phi^T \Phi)^{-1} (\Phi^T R_s)}^I - \overbrace{(\Phi^T \Phi)^{-1} \Phi^T F x(k_i)}^II\end{aligned}\quad (2.31)$$

Dove il primo contributo $(\Phi^T \Phi)^{-1} (\Phi^T R_s)$ rappresenta l'andamento della traiettoria impostata per il segnale di set-point (che rimane appunto invariato); Mentre il secondo termine $(\Phi^T \Phi)^{-1} \Phi^T F x(k_i)$ corrisponde al controllo con reazione dallo stato, che fa riferimento al framework di progettazione del controllo predittivo. Ovviamente questo termine dipende dallo stato che assume il sistema all'istante k_i e quindi è quello che influisce sul cambiamento dei valori del vettore ΔU , come si è potuto constatare dall'esempio precedente. Un'altra osservazione che può sembrare banale, ma che a mio avviso è importante sottolineare e che tale legge di controllo, oltre a basarsi sulla conoscenza dello stato corrente del sistema è fortemente legata dal modello matematico che descrive il processo, infatti la struttura della matrice Hessiana e della matrice F dipende principalmente dalla matrice di transizione dello stato e da quella di uscita, questa riflessione ci consente di fare la seguente considerazione:

Se per il momento tralasciamo il vettore di stato nel secondo termine, ovvero abbiamo $(\Phi^T \Phi)^{-1} \Phi^T F$, osserviamo che entrambe i contributi dipendono dai parametri del sistema, pertanto sono delle matrici costanti se ovviamente il processo da controllare risulta stazionario. Quindi innanzitutto è importante stabilire al momento della formulazione del modello, se la matrice di transizione dello stato è a tempo-invariante o meno (E questa è una scelta del progettista che deve valutare in base alla natura del processo). Questa scelta influenza notevolmente la progettazione e la performance del sistema, in quanto caratterizza lo schema a blocchi del nostro sistema di controllo. Pertanto il nostro obiettivo sarà quello di esprimere la legge di controllo nella forma standard di un controllore con reazione dallo stato di un sistema lineare stazionario (linear time-invariant state feedback control), in modo da sviluppare una sua possibile implementazione.

Sulla base del principio che caratterizza i controllori governati dalla filosofia del RHC, consideriamo solo il primo elemento del vettore ΔU all'istante k_i come variazione che deve subire il segnale di controllo per ottenere l'ingresso ottimo da applicare al sistema, quindi si ottiene:

$$\begin{aligned}\Delta u(k_i) &= \overbrace{\begin{bmatrix} 1 & 0 & \dots \end{bmatrix}}^{N_c} (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \bar{R}_s r(k_i) - \Phi^T F x(k_i)) \\ &= K_y r(k_i) - K_{mpc} x(k_i)\end{aligned}\quad (2.32)$$

La relazione (2.32) è la forma di un linear time-invariant state feedback control.

Dove K_y è il primo elemento di:

$$(\Phi^T \Phi + \bar{R})^{-1} \Phi^T \bar{R}_s$$

e K_{mpc} rappresenta la matrice dei guadagni dell'anello in retro-azione dato dalla prima riga di:

$$(\Phi^T \Phi + \bar{R})^{-1} \Phi^T F$$

Perciò, a partire dalla definizione del modello esteso:

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

Il sistema ad anello chiuso si ottiene sostituendo la relazione (2.32) nel modello esteso; cambiando l'indice k_i con la lettera k , si ottiene l'equazione della legge di controllo ad anello-chiuso, (closed-loop control system):

$$\begin{aligned} x(k+1) &= Ax(k) - BK_{mpc}x(k) + BK_y r(k) \\ &= (A - BK_{mpc})x(k) + BK_y r(k) \end{aligned} \quad (2.33)$$

Quindi, ora ('per la felicità dei controllisti') è possibile calcolare gli autovalori del sistema ad anello chiuso determinati dalla seguente equazione caratteristica:

$$\det[\lambda I - (A - BK_{mpc})] = 0 \quad (2.34)$$

Ora il secondo passo fondamentale che ci consente di abbozzare lo schema a blocchi del sistema, è quello di definire i guadagni di feedback legati allo stato ed all'uscita del sistema.

Data la particolare struttura delle matrici C e A , l'ultima colonna della matrice F risulta identica a \overline{R}_s , ovvero $[1 \ 1 \ \dots \ 1]^T$, pertanto K_y risulta identico all'ultimo elemento di K_{mpc} . In particolare si osserva che il vettore della variabile di stato $x(k_i) = [\Delta x_m(k)^T \ y(k)^T]^T$, e a partire dalla definizione di K_y possiamo esprimere il guadagno $K_{mpc} = [K_x \ K_y]$, dove K_x equivale al vettore del guadagno di feedback legato alla variazione dello stato del sistema $\Delta x_m(k)$, e K_y al guadagno di feedback legato all'uscita $y(k)$. Finalmente possiamo caratterizzare lo schema a blocchi di un sistema di controllo predittivo ad anello chiuso.

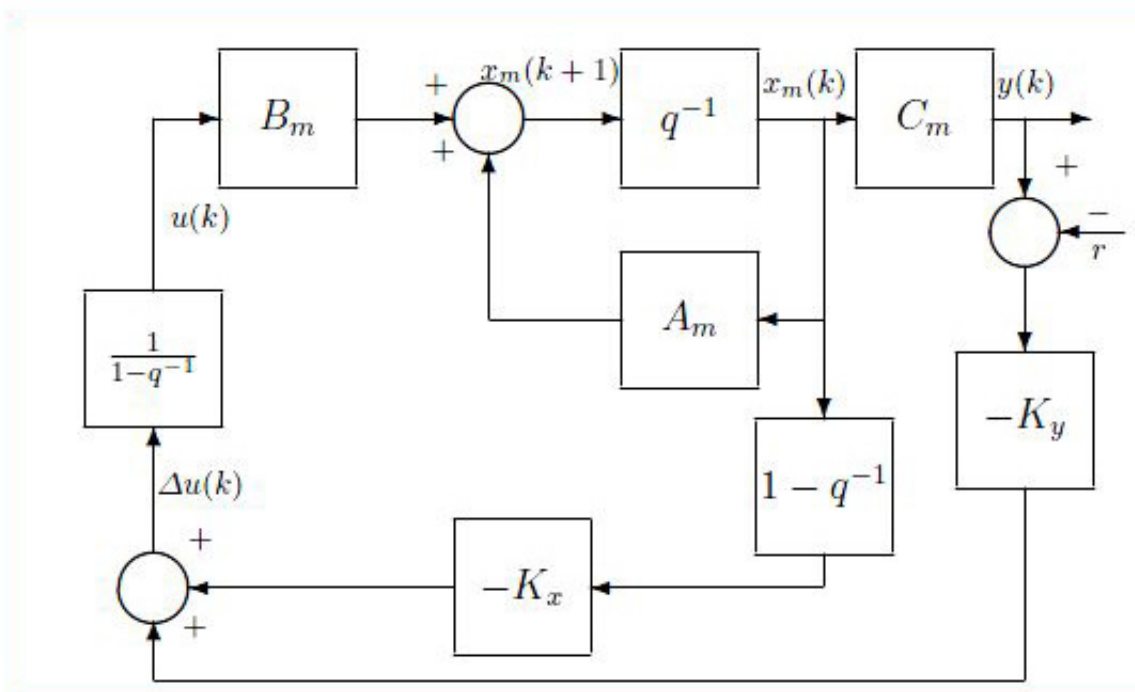


Figura 2.5: Block diagram of DMPC

In questo schema q^{-1} rappresenta l'operatore di ritardo (backward shift operator). Il diagramma mostra la struttura di retroazione dallo stato di un discrete model prediction control (DMPC) con un'azione integrale, nella quale il blocchetto $\frac{1}{1-q^{-1}}$ definisce l'integratore a tempo discreto (equivalente al classico integratore a tempo continuo $\frac{1}{s}$).

Esempio 2.4: Riprendendo l'esempio (2.2) del modello Siso, analizzeremo le matrici di guadagno ad anello chiuso ed i corrispondenti autovalori del sistema; nel caso in cui i pesi del sistema siano pari a $r_w = 0$ e $r_w = 10$.

Soluzione: Partiamo dal caso in cui $r_w = 0$:

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

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

Pertanto, gli autovalori del sistema ad anello chiuso sono calcolati valutando il polinomio caratteristico e quindi la closed-loop matrix $A - BK_{mpc}$, dove nel nostro caso le matrici A e B del modello esteso sono date da:

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

Quindi gli autovalori sono $\lambda_1 = -6.409 \times 10^{-7}$ e $\lambda_2 = 6.409 \times 10^{-7}$, che sono situati approssimativamente in corrispondenza dell'origine del piano complesso.

Studiamo adesso il caso in cui $r_w = 10$:

$$K_y = [1 \ 0 \ 0 \ 0] (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}) = 0.2453$$

$$K_{mpc} = [1 \ 0 \ 0 \ 0] (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T F) = [0.6939 \ 0.2453].$$

Con questo nuovo vettore dei guadagni, gli autovalori del sistema ad anello chiuso sono $\lambda_{1,2} = 0.8530 \pm j0.0542$, mostrando che le dinamiche del sistema hanno una risposta più lenta rispetto a quella del caso in cui $r_w = 0$. (Ricordo che la velocità delle dinamiche del sistema a tempo discreto è caratterizzata dalla parte reale degli autovalori che deve rimanere confinata nella regione di convergenza (ROC Region of Convergence) descritta dalla circonferenza di raggio unitario. Ovviamente più si è vicini all'origine, maggiore sarà la velocità di risposta del sistema).

Esempio 2.5: Si consideri il sistema a tempo continuo descritto dalla seguente funzione di trasferimento nel dominio di Laplace:

$$G(s) = \frac{100}{s^2 + s + \frac{100}{w^2}}$$

$$= \frac{100}{s^2 + \xi w s + w^2}$$

Dove abbiamo indicato con $w = 10$ la pulsazione naturale del sistema e con $\xi = 0.1$ il coefficiente di smorzamento. In base al teorema del campionamento dovremmo scegliere una frequenza di campionamento che sia almeno il doppio della banda passante del sistema, comunque scegliamo un'intervallo di campionamento pari a $\Delta t = 0.01$ che nel nostro caso equivale a dieci volte la banda passante del sistema. Naturalmente il sistema che stiamo considerando è stabile ed è caratterizzato da due poli complessi coniugati $\lambda_{1,2} = -0.5 \pm j10$. Analizzeremo ora il problema della sensibilità relativa alla selezione dei parametri di progetto nel caso in cui vengano scelti $N_c = 3$, $N_p = 20$ oppure $N_p = 200$; con un segnale di set-point $\bar{R} = 0.5I$. In sintesi dobbiamo affrontare un problema di regolazione, in cui il sistema è caratterizzato da due poli complessi coniugati.

Soluzione: Innanzitutto è opportuno passare ad una rappresentazione a tempo continuo nello spazio di stato, per procedere alla discretizzazione del processo ed ottenere quindi un modello a tempo discreto nello spazio di stato (riporto le istruzioni matlab):

```
omega = 10;
numc = omega*omega;
denc = [1 0.1*omega omega*omega];
[Ac, Bc, Cc, Dc] = tfss(numc,denc);
```

Per quanto riguarda le altre istruzioni si può far riferimento al codice extmodel.m. Comunque nel passaggio intermedio nello spazio di stato a tempo discreto si ottengo gli autovalori del sistema che sono $\lambda_{1,2} = 0.9901 \pm 0.0992i$ che come si può facilmente evincere si trovano all'interno del cerchio di raggio unitario. (D'altronde abbiamo scelto correttamente il tempo di campionamento). Comunque alla fine otteniamo il seguente modello esteso:

$$\begin{cases} x(k+1) = A \cdot x(k) + B \cdot \Delta u(k) \\ y(k) = C \cdot x(k) \end{cases}$$

dove:

$$A = \begin{bmatrix} 0.9851 & -0.9934 & 0 \\ 0.0099 & 0.9950 & 0 \\ 0.9934 & 99.5021 & 1 \end{bmatrix}; \quad B = [0.0099 \quad 0.0000 \quad 0.0050]$$

$$C = [0 \quad 0 \quad 1]$$

Innanzitutto studiamo l'effetto che ha la scelta dell'orizzonte di predizione sulla legge di controllo ΔU . Supponiamo che all'istante di campionamento $k = 10$, le condizioni iniziali del sistema esteso sono $x(10) = [0.1 \quad 0.2 \quad 0.3]^T$.

Nel caso in cui $N_p = 20$ il segnale di controllo:

$$\Delta U = [-144.9984 \quad -65.4710 \quad 1.2037]^T$$

per quanto riguarda lo state feedback control gain ed i corrispondenti autovalori abbiamo:

$$K_{mpc} = [45.4168 \quad 705.6123 \quad 0.9513]$$

$$\lambda_1 = 0.6974; \lambda_{2,3} = 0.8959 \pm 0.1429j.$$

Mentre nel caso in cui $N_p = 200$ il segnale di controllo:

$$\Delta U = [-645.5885 \quad -0.4664 \quad 629.0276]^T$$

per quanto riguarda lo state feedback control gain ed i corrispondenti autovalori abbiamo:

$$K_{mpc} = [80.6 \quad 3190 \quad 0.79]$$

$$\lambda_1 = 0.9749; \lambda_{2,3} = 0.5207 \pm 0.2919j.$$

Da questa prima analisi possiamo evincere che estendendo l'orizzonte di predizione il vettore degli ingressi subisce delle variazioni significative. (Basta confrontare i due vettori ΔU). Inoltre esaminando attentamente la matrice *Hessiana*:

$$\Phi^T \Phi + \bar{R}$$

si scopre che è i suoi valori dipendono dall'orizzonte di predizione. Infatti risulta rispettivamente:

Nel caso in cui $N_p = 20$:

$$\Phi^T \Phi = \begin{bmatrix} 9.8796 & 8.9387 & 8.0099 \\ 8.9387 & 8.1020 & 7.2737 \\ 8.0099 & 7.2737 & 6.5425 \end{bmatrix}$$

con condizionamento relativo all'invertibilità della matrice (condition number):

$$\kappa(\Phi^T \Phi + 0.5I) = 49.98$$

Nel caso in cui $N_p = 200$:

$$\Phi^T \Phi = \begin{bmatrix} 236.0557 & 235.5010 & 234.5466 \\ 235.5010 & 235.3753 & 234.8473 \\ 234.5466 & 234.8473 & 234.7473 \end{bmatrix}$$

con condizionamento relativo all'invertibilità della matrice (condition number):

$$\kappa(\Phi^T \Phi + 0.5I) = 1410$$

Quindi la stabilità di calcolo nell'invertire la matrice Hessiana diventa precaria aumentando l'orizzonte di predizione N_p . E' possibile inoltre riscontrare dei problemi nel calcolo numerico della matrice *Hessiana*, quando si attribuiscono dei pesi nulli alla matrice \bar{R} ; infatti in tale circostanza l'escursione del segnale d'ingresso è notevole e può dar luogo a valori elevati di $\Phi^T \Phi$.

Un'altro aspetto importante da considerare è legato alla stabilità del sistema di controllo predittivo ad anello chiuso; infatti non necessariamente la legge di controllo in un sistema caratterizzato da un'orizzonte di predizione e di controllo brevi risulta stabile. Infatti in un sistema che presenta delle dinamiche che si trovano al limite di stabilità, è importante valutare quali valori attribuire alla matrice dei pesi. Questo si può notare osservando i grafici delle figure (2.15) e (2.16), che mostrano come l'assegnazione di un peso differente (nel caso in esame $w_{11} = 0.5$ ed $w_{22} = w_{33} = 0$) possa dar luogo ad un comportamento instabile.

Quindi ci troviamo di fronte ad un bivio caratterizzato: da una parte dalla scelta di un N_p non troppo elevato per problemi di malcondizionamento della matrice *Hessiana*, e dall'altra una scelta non troppo piccola dell'orizzonte di controllo il quale ricordo è limitato da N_p . A questi problemi si aggiunge inoltre la scelta dei pesi da attribuire agli ingressi di controllo. Infatti, se da un lato occorre avere l'accortezza di non esercitare un'azione di controllo troppo spinta, dall'altra bisogna fare attenzione a penalizzare maggiormente lo spostamento di dinamiche critiche, in modo da ponderare l'energia del segnale d'ingresso solo sulle dinamiche più stabili del processo al fine di raggiungere la performance desiderata.

Questi sono i parametri sui quali generalmente viene effettuato il tuning per garantire la stabilità e rispettare le specifiche di progetto. Comunque è possibile superare questa difficoltà proponendo dei metodi numerici che consentano di 'alleviare' il problema del mal-condizionamento e quindi garantire una maggiore stabilità al controllo adoperando un'orizzonte di predizione e controllo maggiori. Una possibile soluzione è quella di alleggerire il calcolo utilizzando il metodo 'exponential data weight' il quale durante l'evoluzione del sistema modifica i valori della matrice in modo da garantire una stabilità numerica. Generalmente questa operazione viene effettuata valutando un limite superiore sulla norma della matrice da invertire.

2.5 Predictive Control of MIMO Systems

Fino ad ora abbiamo considerato il caso di sistemi SISO; Adesso, grazie alla formulazione del modello in forma di spazio di stato, è possibile estendere con facilità i risultati ottenuti precedentemente nel caso MIMO, affrontando il problema con la stessa metodologia di progettazione.

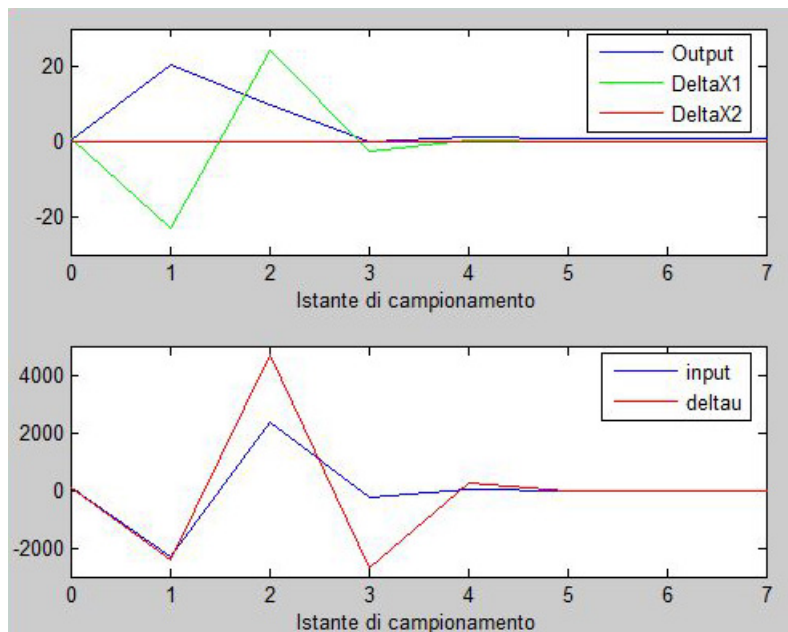


Figura 2.6: MPC short $N_p=20$ no weight

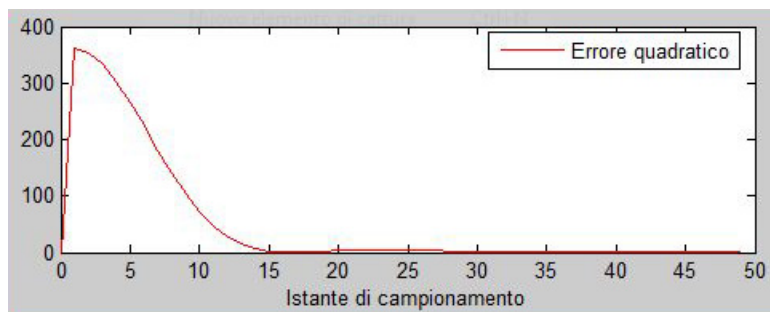


Figura 2.7: MPC error short $N_p=20$ no weight

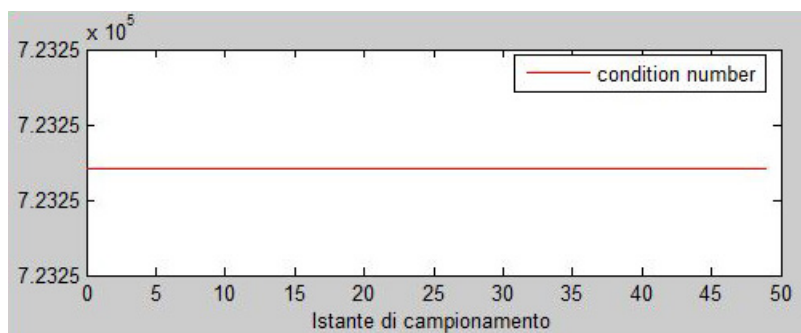


Figura 2.8: MPC short $N_p=20$ ill-conditioning

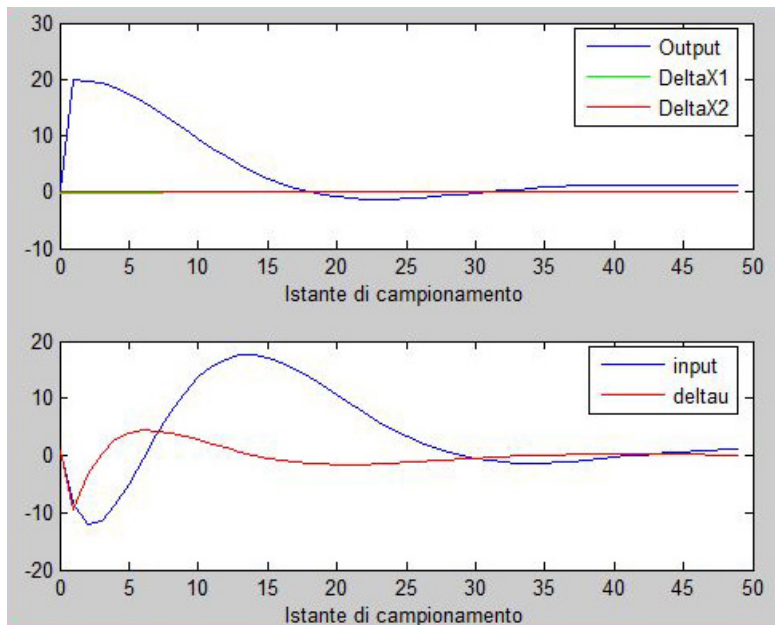


Figura 2.9: MPC short $N_p=20$ weight=0.5

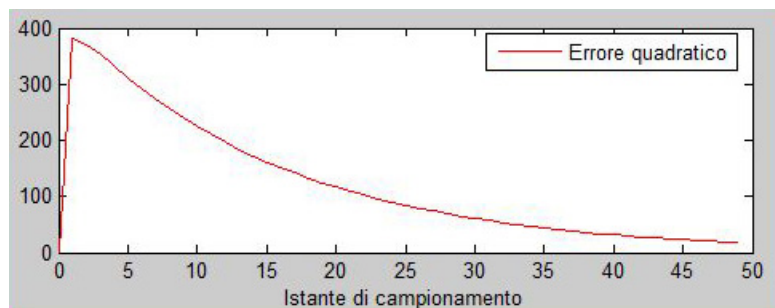


Figura 2.10: MPC error short $N_p=20$ weight=0.5

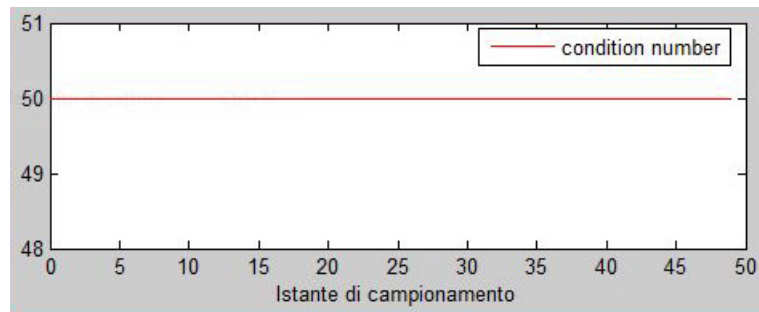


Figura 2.11: MPC short $N_p=20$ ill-conditioning weight=0.5

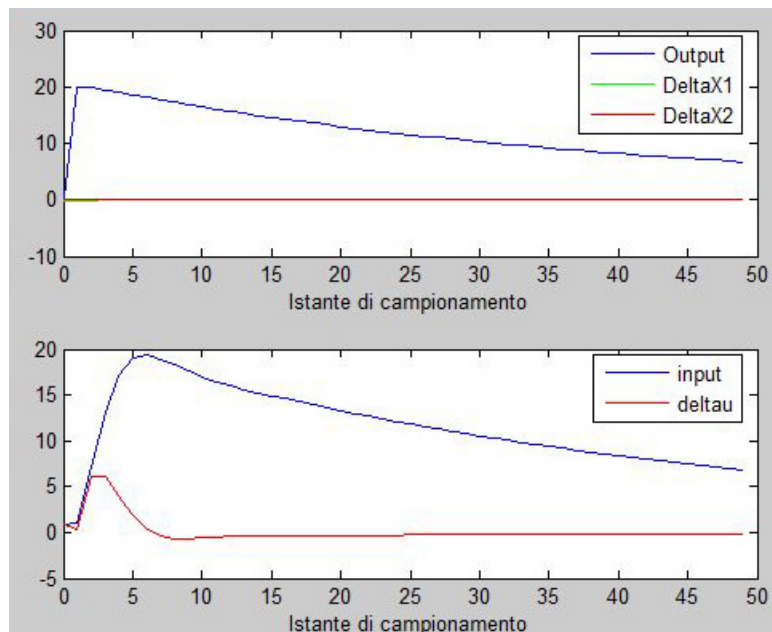


Figura 2.12: MPC long $N_p=200$ weight=0.5

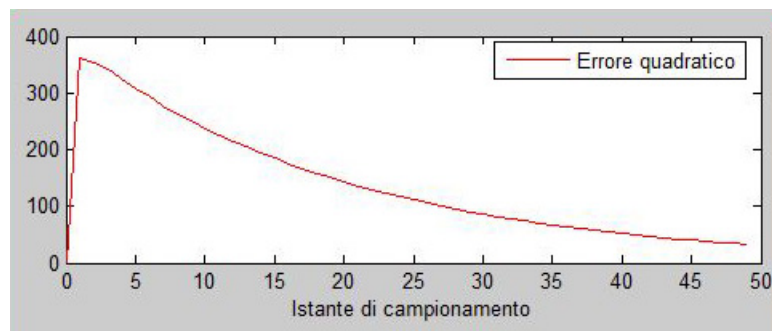


Figura 2.13: MPC error long $N_p=200$ weight=0.5

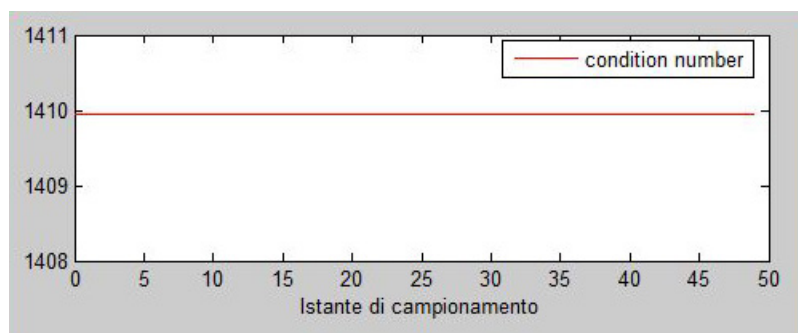


Figura 2.14: MPC error long $N_p=200$ ill-conditioning weight=0.5

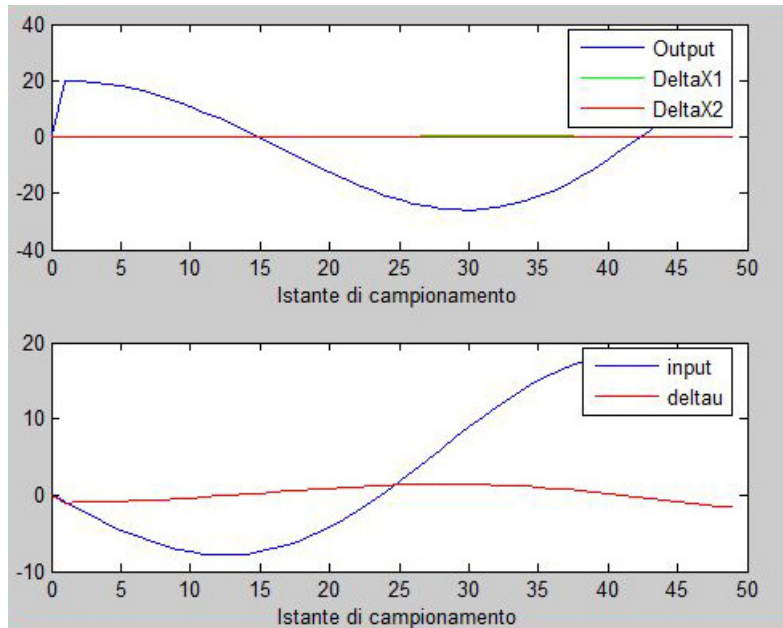


Figura 2.15: MPC short $N_p=20$ first weight=0.5

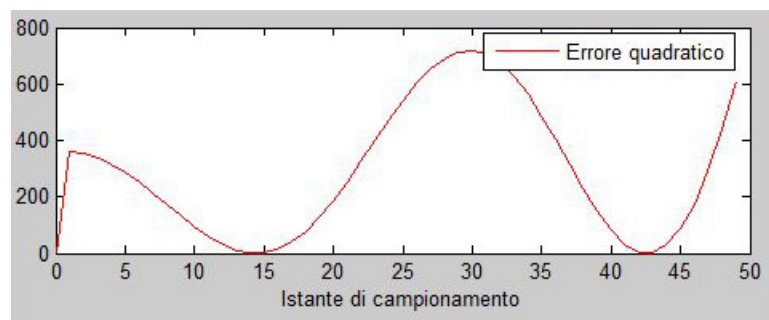


Figura 2.16: MPC error short $N_p=20$ first weight=0.5

2.5.1 General formulation of the model

Consideriamo un processo caratterizzato da \mathbf{m} ingressi, \mathbf{q} uscite e n_1 stati. Inoltre, facciamo l'ipotesi che il numero delle uscite sia inferiore od uguale al numero degli ingressi (*i.e.* $\mathbf{q} \leq \mathbf{m}$). Se il numero degli outputs risultasse maggiore rispetto a quello degli inputs, ovviamente risulterebbe difficile sperare di controllare in modo indipendente le varie uscite del sistema (ovvero le grandezze misurate a valle del processo, che desideriamo governare), al fine da poter annullare l'andamento degli errori delle rispettive uscite a regime.

Nell'affrontare il problema del controllo predittivo, prenderemo in considerazione nella formulazione generale del modello anche la presenza del rumore e dei disturbi.

$$\begin{cases} x_m(k+1) = A_m x_m(k) + B_m u(k) + B_d \omega(k) \\ y(k) = C_m x_m(k) \end{cases} \quad (2.35)$$

Dove $\omega(k)$ è un disturbo che agisce sul processo, assimilabile ad una sequenza di campioni di rumore bianco filtrato da un'integratore. Pertanto avremo una sequenza di campioni scorrelati tra loro, caratterizzata da una media nulla e da uno spettro di potenza costante; ovvero consideriamo la situazione limite in cui la funzione di autocorrelazione è impulsiva. In tale circostanza avremo quindi che le statistiche del primo ordine sono caratterizzate da una covarianza nulla e da una funzione di autocorrelazione che tende a coincidere con il quadrato del valor medio. In sintesi stiamo idealmente considerando un processo stazionario ergodico, che comunque nella maggior parte dei casi pratici può ritenersi accettabile se la durata delle nostre osservazioni risulti limitata, in modo tale che le grandezze statistiche del segnale aleatorio rimangano invariate nell'arco temporale del fenomeno considerato. Risulta chiaro quindi che suddetto aspetto è legato alle condizioni operative in cui si trova l'impianto e quindi se si considerano processi continui è opportuno valutare l'effetto di agenti esogeni legati a fattori ambientali.

Quindi possiamo definire la sequenza di campioni di rumore bianco nel modo seguente:

$$\omega(k) - \omega(k-1) = \epsilon(k) \quad (2.36)$$

Analogamente a quanto è stato fatto nella formulazione del modello di un sistema SISO, esprimiamo i due passaggi intermedi che ci consentiranno di caratterizzare la transizione del sistema da un'istante di campionamento a quello successivo. Quindi a partire dalle seguenti equazioni:

$$\begin{aligned} \Delta x_m(k) &= x_m(k) - x_m(k+1) \\ \Delta u(k) &= u(k) - u(k-1) \end{aligned} \quad (2.37)$$

Otteniamo:

Ingresso-Stato:

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) + B_d \epsilon(k). \quad (2.38)$$

Stato-Uscita:

Per legare l'uscita corrente $y(k)$ alla variabile di stato $\Delta x_m(k)$ abbiamo:

$$\begin{aligned} y(k+1) - y(k) &= \Delta y(k+1) \\ &= C_m \Delta x_m(k+1) \\ &= C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) + C_m B_d \epsilon(k) \end{aligned} \quad (2.39)$$

Quindi, indicando la nuova v.d.s del modello esteso con $x(k) = [\Delta x_m(k)^T y(k)^T]^T$, abbiamo:

Modello-Esteso:

$$\begin{cases} \begin{aligned} \overbrace{\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix}}^{x(k+1)} &= \overbrace{\begin{bmatrix} A_m & o_m^T \\ C_m \cdot A_m & I_{q \times q} \end{bmatrix}}^A \cdot \overbrace{\begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}}^{x(k)} + \overbrace{\begin{bmatrix} B_m \\ C_m \cdot B_m \end{bmatrix}}^B \cdot \Delta u(k) + \overbrace{\begin{bmatrix} B_d \\ C_m B_d \end{bmatrix}}^{B_\epsilon} \cdot \epsilon(k) \\ \\ y(k) &= \overbrace{\begin{bmatrix} o_m & I_{q \times q} \end{bmatrix}}^C \cdot \overbrace{\begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}}^{x(k)} \end{aligned} \end{cases} \quad (2.40)$$

Dove:

- $I_{q \times q}$ è la matrice identità che ha dimensioni $q \times q$, che corrisponde al numero degli outputs.
- o_m è una matrice composta da elementi nulli di dimensione $q \times n_1$.
- A_m è una matrice di dimensione $n_1 \times n_1$.
- B_m è una matrice di dimensione $n_1 \times m$.
- C_m è una matrice di dimensione $q \times n_1$.

In seguito indicheremo la dimensione complessiva del modello esteso con la lettera $n (= n_1 + q)$. Vi sono due importanti punti da analizzare in questo modello:

1. Il primo riguarda lo studio dei suoi autovalori.
2. Il secondo si concentra sulla realizzazione del modello nello spazio di stato.

Al termine di questa analisi potremmo avere una maggiore comprensione del modello in esame.

Autovalori del Modello Esteso

Il polinomio caratteristico del sistema esteso è:

$$\begin{aligned} \rho(\lambda) &= \det \begin{bmatrix} \lambda I - A_m & o_m^T \\ -C_m A_m & (\lambda - I) I_{q \times q} \end{bmatrix} \\ &= (\lambda - 1)^q \det(\lambda I - A_m) = 0 \end{aligned} \quad (2.41)$$

dove è stata utilizzata la proprietà algebrica per cui il determinante di una matrice triangolare inferiore è pari al prodotto dei determinanti delle matrici presenti sulla diagonale principale. Quindi, gli autovalori del modello esteso sono dati dall'unione degli autovalori del processo e gli q autovalori in $\lambda = 1$. Questo significa che vi sono q integratori incorporati nella progettazione del modello esteso, ognuno per una differente uscita del sistema. Pertanto, questo è il motivo per cui generalmente è presente un'azione integrale all'interno di un sistema MPC.

Controllabilità ed Osservabilità del Modello Esteso

Dato che la base di partenza di una progettazione di un sistema MPC è legata alla rappresentazione di stato del modello esteso (il quale è caratterizzato dalla presenza di q integratori aggiuntivi nel processo originario), risulta fondamentale che tale modello non perda le proprietà di Controllabilità ed Osservabilità del sistema di partenza con particolare riferimento alle dinamiche instabili del processo.

La Controllabilità e l'Osservabilità costituiscono due pre-requisiti essenziali per i sistemi di controllo predittivi, se si desidera ottenere una determinata performance per il sistema di controllo ad anello chiuso e se si vuole progettare un buon osservatore dello stato.

Comunque tali pre-requisiti possono essere meno stringenti, richiedendo semplicemente che il sistema risulti Stabilizzabile e Rilevabile ¹. In questo lavoro ho considerato che il sistema sia completamente controllabile ed osservabile in modo da raggiungere la performance desiderata del sistema ad anello chiuso. Più avanti analizzeremo un'esempio che metterà in luce l'importanza che riveste l'osservabilità per la progettazione di un'osservatore.

Poichè l'elemento sostanziale che differenzia il modello in forma di predizione, rispetto a quello originario, è legato alla presenza dei modi integrali aggiuntivi, risulta necessario studiare sotto quali opportune condizioni questi modi risultino controllabili. Il modo più semplice per affrontare tale analisi è basata sulla considerazione che il modello del processo sia in forma minima.

¹Un sistema è Stabilizzabile, se l'eventuale presenza di modi non controllabili diano comunque luogo ad evoluzioni stabili del processo. Mentre i suoi modi controllabili potrebbero essere stabili oppure instabili. Un sistema si definisce Rilevabile, se l'eventuale presenza di modi non osservabili del sistema diano luogo ad evoluzioni stabili. Anche qui i modi osservabili possono essere sia stabili che instabili. Risulta chiaro che in questo contesto, per stabilità si intende che i corrispondenti autovalori devono essere strettamente confinati all'interno del cerchio di raggio unitario.

Sebbene è possibile rimandare a qualsiasi testo di controlli automatici, reputo importante richiamare brevemente i concetti principali che definiscono tale forma canonica; in quanto a partire da questi imposteremo la nostra analisi.

La Forma di Controllo (o anche detta Forma Compagna) è una particolare rappresentazione nello spazio di stato caratterizzata dal fatto che le matrici in gioco (A_c, B_c, C_c, D_c) hanno la peculiarità dall'aver tutti gli elementi indipendenti dai valori dei coefficienti della F.d.T, salvo $2n+1$ termini (dove n è la dimensione del nostro sistema). Quindi $2n+1$ rappresenta il minimo numero di parametri necessari per descrivere la F.d.T in termini di v.d.s. In particolare l'ultima riga della matrice dinamica A_c ha per elementi i coefficienti del polinomio caratteristico, ossia del denominatore della $F(s)$, le cui radici sono i poli della F.d.T; mentre C_c ha per elementi i coefficienti del numeratore della $F(s)$, ossia del polinomio le cui radici sono gli zeri della F.d.T. Il nome di questa forma canonica è dovuto al fatto che, indipendentemente dal valore dei parametri (i coefficienti a_i della F.d.T) il sistema che essa definisce è sempre completamente controllabile. La proprietà di osservabilità scaturisce dall'ipotesi che i polinomi della F.d.T siano polinomi primi tra loro (ovvero non abbiano radici in comune e quindi non si verifichino cancellazioni). A questo punto risulta essenziale studiare sotto quali condizioni una particolare F.d.T risulti Realizzabile. Formalizziamo quindi quanto esposto.

Definizione 2.1 Una realizzazione di una funzione di trasferimento a tempo discreto $G(z)$ è una generica terna di matrici (A, B, C) (che modellano una possibile rappresentazione nello spazio di stato) tale che sia verificata la seguente relazione: $G(z) = C(zI - A)^{-1}B$. Se suddetto insieme (A, B, C) esiste, allora $G(z)$ si dice Realizzabile.

Definizione 2.2 Una particolare realizzazione (A, B, C) si definisce minima per una data $G(z)$, se non esistono altre realizzazioni che abbiano una dimensione inferiore alla terna considerata per rappresentare il sistema.

Una realizzazione in forma minima, come è stato detto in precedenza, ha la caratteristica essenziale sintetizzata dal seguente teorema:

Teorema 2.1 Una realizzazione minima risulta Controllabile ed Osservabile.

A partire da queste definizioni e teoremi, ci proponiamo di determinare le condizioni sotto le quali il nostro modello esteso risulti controllabile ed osservabile.

Teorema 2.2 Sotto l'ipotesi che il modello che descrive il processo (A_m, B_m, C_m) , risulti controllabile ed osservabile, possiamo ottenere la funzione di trasferimento $G(z)$ a partire dalla definizione di Realizzazione in forma minima:

$$G_m(z) = C_m(zI - A_m)^{-1}B_m$$

Allora, la funzione di trasferimento del modello esteso adottato per la progettazione del sistema MPC (equaz.2.40) ha la seguente forma:

$$\overbrace{G(z)}^{F.d.Text-model} = \frac{z}{z-1} \cdot \overbrace{G_m(z)}^{F.d.Tprocess} \quad (2.42)$$

la quale risulta controllabile ed osservabile se e solo se la F.d.T del processo $G_m(z)$ non ha zeri in $z = 1$ ².

Dimostrazione 2.1 Per mostrare che il modello esteso sia controllabile ed osservabile, dobbiamo dimostrare che risulta verificata la relazione (2.42). Dopo di che le proprietà di controllabilità ed osservabilità seguono dalla particolare forma minima del modello esteso senza la cancellazione polo-zero.

Si ricorda che data una matrice quadrata M con la seguente struttura a blocchi:

$$M = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} \quad (2.43)$$

²Gli zeri di una funzione di trasferimento di un sistema MIMO sono quei valori z per cui la matrice $G_m(z)$ cade di rango.

Se A_{11}^{-1} e A_{22}^{-1} esistono, allora:

$$M^{-1} = \begin{bmatrix} A_{11}^{-1} & 0 \\ -A_{22}^{-1}A_{21}A_{11}^{-1} & A_{22}^{-1} \end{bmatrix} \quad (2.44)$$

Quindi a partire da tale proprietà algebrica risulta:

$$G(z) = C(zI - A)^{-1}B. \quad (2.45)$$

dove:

$$C(zI - A)^{-1} = \begin{bmatrix} (zI_m - A_m)^{-1} & 0 \\ (1 - z^{-1})C_m A_m (zI_m - A_m)^{-1} & (1 - z^{-1})I_q \end{bmatrix}. \quad (2.46)$$

Andando a sostituire le matrici B e C del modello esteso nella relazione (2.45) si ottiene la relazione (2.42). Quindi sotto l'ipotesi che la F.d.T del processo $G_m(z)$ non abbia zeri in $z = 1$ e sia in forma minima, allora la F.d.T del modello esteso può essere definita a partire dalla relazione (2.42), e pertanto risulta osservabile e controllabile.

Vediamo ora di applicare le proprietà appena analizzate, considerando un sistema SISO. Mettiamoci nella situazione 'sfortunata' in cui uno degli zeri della funzione di trasferimento sia posizionato in $z = 1$, allora per quanto detto in precedenza il modello esteso non è controllabile. Ad esempio, se:

$$G_m(z) = \frac{(z - 1)}{(z - 0.6)(z - 0.8)},$$

allora si verificherà una cancellazione polo-zero in $G(z)$, che appunto risulta da:

$$G(z) = \frac{z}{z - 1} \cdot \frac{(z - 1)}{(z - 0.6)(z - 0.8)}$$

Nel caso di un sistema con un solo ingresso ed una sola uscita il guadagno a regime del modello del processo è nullo e non permette un'azione integrale. Sottolineiamo inoltre che il numero degli ingressi deve essere maggiore od uguale al numero delle uscite (*i.e.* $m \geq q$). Quando il numero degli inputs, risulta inferiore a quello degli outputs, i modi del modello esteso possono diventare non controllabili.

Quando si utilizza MATLAB, per ottenere una realizzazione minima di un modello nello spazio di stato viene effettuato una riduzione dell'ordine del sistema. Ad esempio, si consideri la seguente funzione di trasferimento a tempo discreto:

$$G_m(z) = \frac{(z - 0.1)}{(z - 0.1)(z - 0.9)},$$

Si nota una cancellazione polo-zero in $z = 0.1$. Utilizzando la funzione (tf2ss.m) di MATLAB, si può ottenere una possibile realizzazione nello spazio di stato di $G_m(z)$ la quale è costituita da due variabili di stato:

$$A_m = \begin{bmatrix} 1 & -0.09 \\ 1 & 0 \end{bmatrix}; \quad B_m = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \quad C_m = [1 \quad -0.1].$$

Questa non rappresenta una realizzazione in forma minima dato che la funzione di trasferimento ha una cancellazione polo-zero. Allora è possibile ottenere una realizzazione in forma minima nello spazio di stato, implementando il seguente script relativo a questo semplice esempio:

```
numd = [1 -0.1];
dend = conv([1 -0.1],[1 -0.9]);
sys1 = tf(numd, dend);
sys = ss(sys1,'min');
[Am, Bm, Cm, Dm] = ssdata(sys);
```

Quindi la realizzazione minima attraverso la riduzione dell'ordine del modello è data da:

$$A_m = 0.9; \quad B_m = -0.9806; \quad C_m = -1.0198,$$

La quale presenta un'unica variabile di stato come appunto era da aspettarselo.

2.5.2 Solution of Predictive Control for MIMO Systems

I risultati che abbiamo ottenuto nel caso SISO sono facilmente estendibili nel caso dei sistemi MIMO; bisogna solo prestare attenzione alle dimensioni delle matrici dello stato, degli ingressi e delle uscite in un ambiente caratterizzato da più uscite e più ingressi. Quindi indichiamo con:

$$\begin{aligned} \Delta U &= [\Delta u(k_i)^T \quad \Delta u(k_i + 1)^T \quad \dots \quad \Delta u(k_i + N_c - 1)^T]^T \\ Y &= [y(k_i + 1|k_i)^T \quad y(k_i + 2|k_i)^T \quad y(k_i + 3|k_i)^T \quad \dots \quad y(k_i + N_p|k_i)^T]^T. \end{aligned}$$

A partire dalla terna di matrici (A,B,C) della rappresentazione nello spazio di stato, il valore che assumeranno le variabili di stato negli istanti temporali successivi sono ottenuti calcolando in modo sequenziale (a partire dall'insieme dei valori ottimi dei parametri di controllo che verranno implementati negli istanti successivi)

$$\begin{aligned} x(k_i + 1|k_i) &= Ax(k_i) + B\Delta u(k_i) + B_d\epsilon(k_i) \\ x(k_i + 2|k_i) &= Ax(k_i + 1) + B\Delta u(k_i + 1) + B_d\epsilon(k_i + 1|k_i) \\ &= A^2x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i + 1) + AB_d\epsilon(k_i) + B_d\epsilon(k_i + 1|k_i) \\ &\vdots \\ x(k_i + N_p|k_i) &= A^{N_p}x(k_i) + A^{N_p-1}B\Delta u(k_i) + A^{N_p-2}B\Delta u(k_i + 1) + A^{N_p-N_c}B\Delta u(k_i + N_c - 1) + A^{N_p-1}B_d\epsilon(k_i) \\ &\quad + A^{N_p-2}B_d\epsilon(k_i + 1|k_i) + \dots + B_d\epsilon(k_i + N_p - 1|k_i). \end{aligned}$$

A partire dall'ipotesi che $\epsilon(k)$ sia una sequenza di campioni generata da un generatore di rumore bianco caratterizzato da un valor medio nullo, il valore previsto per $\epsilon(k_i + i|k_i)$ all'istante di campionamento i si può considerare nullo. Quindi poichè l'effetto del rumore sui valori previsti è nullo, la predizione della variabile di stato e di uscita del sistema viene calcolata come il valore atteso delle rispettive variabili. Per semplicità di notazione l'operatore di valore atteso verrà omesso.³ Quindi si ottiene:

$$Y = Fx(k_i) + \Phi\Delta U. \quad (2.47)$$

Dove le matrici F e Φ sono definite nella seguente maniera:

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

Pertanto la variazione che subisce il segnale di controllo ottimo nella finestra temporale è:

$$\Delta U = (\Phi^T\Phi + \bar{R})^{-1}(\Phi^T\bar{R}_s r(k_i) - \Phi^T Fx(k_i)), \quad (2.48)$$

Dove però in tale circostanza la matrice $\Phi^T\Phi$ ha dimensione $mN_c \times mN_c$ e la matrice $\Phi^T F$ ha dimensione $mN_c \times n$ ed in fine $\Phi^T\bar{R}_s$ risulta uguale alle ultime q colonne della matrice $\Phi^T F$. La matrice dei pesi \bar{R} è una matrice a blocchi con m blocchi e ha dimensione pari a quella della matrice $\Phi^T\Phi$. Il segnale di set-point è $r(k_i) = [r_1(k_i) \quad r_2(k_i) \quad \dots \quad r_q(k_i)]^T$ definito come l'insieme dei q segnali di set-point per un sistema MIMO.

Quindi considerando il principio su cui si basa il RHC, i primi m elementi di ΔU vengono considerati per implementare l'azione di controllo ottima:

$$\begin{aligned} \Delta u(k_i) &= \overbrace{[I_m \quad o_m \quad \dots \quad o_m]}^{N_c} (\Phi^T\Phi + \bar{R})^{-1} (\Phi^T\bar{R}_s r(k_i) - \Phi^T Fx(k_i)) \\ &= K_y r(k_i) - K_m p c x(k_i). \end{aligned} \quad (2.49)$$

³Il valore atteso di un segnale di potenza è definito come $\lim_{N \rightarrow \infty} \frac{1}{2N+1} \cdot \sum_{n=-N}^N x[n]$

dove I_m e o_m sono rispettivamente la matrice identità e la matrice nulla, di dimensione $m \times m$.

2.6 State Estimation

Fino a questo momento abbiamo considerato la progettazione di controllori per sistemi in cui l'informazione $x(k_i)$ risultava disponibile all'istante temporale corrente k_i . Questo naturalmente implica che è possibile misurare tutte le variabili di stato del processo attraverso opportuni sensori. In verità, nella maggior parte delle applicazioni, non sempre è possibile realizzare sistemi di controllo con un feedback completo dello stato (full-state feedback design), dato che è piuttosto raro accedere direttamente a tutte le variabili di stato del processo. Anche per il semplice fatto che alcune delle variabili di stato non modellano quantità fisiche. Comunque, anche nelle situazioni in cui tutte le variabili di stato abbiano una corrispondenza con le grandezze fisiche del processo, la possibilità di non avere a disposizione dei sensori che forniscano misure che abbiano una certa accuratezza risulta abbastanza frequente sia perchè non sono presenti sul mercato e sia perchè il loro costo di realizzazione potrebbe risultare proibitivo. Inoltre la misura di alcune di queste variabili potrebbe essere corrotta dalla presenza di rumore significativo da pregiudicare le prestazioni del sistema di controllo basato sulla affidabilità dei sensori in gioco. Pertanto, risulta inevitabile la scelta di stimare il vettore di stato, piuttosto che effettuarne una misura. A questo punto la domanda sorge spontanea: Come è possibile stimare il vettore di stato, se le grandezze in gioco non possono essere misurate?

La risposta è abbastanza intuitiva, ovvero osservando l'uscita del sistema a partire dalla conoscenza dell'andamento del segnale d'ingresso per un'intervallo di tempo finito e ricostruendo il vettore di stato, attraverso i valori registrati dell'output (che sono stati osservati precedentemente). Il modello matematico del processo attraverso il quale il vettore di stato viene stimato, a partire dalla misurazione dell'output viene denominato *osservatore* o stimatore dello stato. La presenza di un'osservatore è essenziale nei moderni sistemi di controllo, soprattutto per svolgere la funzione di filtrare il rumore e ridurre gli effetti indesiderati sulle acquisizioni dei sensori a nostra disposizione. Quando un'osservatore stima l'intero vettore di stato, prende il nome di *full-order observer* ovvero un'osservatore completo dello stato. Analogamente, un'osservatore che stima le variabili che non possono essere misurate si chiama ridotto *reduced-order observer*. Sebbene un'osservatore ridotto dia luogo ad un sistema di controllo di ordine più piccolo rispetto ad un'osservatore completo, si preferisce progettare un sistema basato su un'osservatore completo, soprattutto se la presenza del rumore risulta significativi sulle variabili di stato misurate. Un controllore basato sulla stima di un'osservatore, prende il nome di *compensatore*.

Compresa quindi l'importanza di considerare la presenza di un'osservatore nel nostro schema di controllo, vedremo di seguito come integrare questo nuovo elemento all'interno di un sistema di controllo predittivo.

2.6.1 Basic Ideas about an Observer

Risulta abbastanza difficile immaginare come sia possibile costruire un'osservatore in un sistema di controllo predittivo, dove la conoscenza del modello matematico che descrive il processo risulta essenziale per la progettazione della legge di controllo. Ad esempio, si consideri la seguente rappresentazione nello spazio di stato:

$$x_m(k+1) = A_m x_m(k) + B_m u(k). \quad (2.50)$$

Allora possiamo utilizzare tale modello per calcolare il valore delle variabili di stato $\widehat{x}_m(k)$, $k = 1, 2, \dots$, con condizione iniziale $\widehat{x}_m(0)$ ed un segnale di ingresso $u(k)$ nel seguente modo:

$$\widehat{x}_m(k+1) = A_m \widehat{x}_m(k) + B_m u(k). \quad (2.51)$$

In effetti, questo approccio potrebbe funzionare dopo che sia passato una fase transitoria, se ovviamente il processo risulta stabile e la nostra *stima dello stato iniziale del sistema* sia sufficientemente corretta. Tuttavia, quali problemi potrebbero nascere da questo tipo di approccio?

In sostanza, quello che abbiamo appena descritto è una predizione ad anello aperto (ovvero una volta valutato lo stato iniziale del sistema sulla base di alcune considerazioni preliminari, sulla base della conoscenza del modello calcoliamo l'uscita del sistema negli istanti successivi). Infatti, considerando l'errore, definito come:

$$\widehat{e}_m(k) = x_m(k) - \widehat{x}_m(k)$$

Quindi considerando l'evoluzione dell'errore nell'istante di campionamento successivo si ha:

$$\begin{aligned}
 \widehat{e}_m(k+1) &= x_m(k+1) - \widehat{x}_m(k+1) \\
 &= A_m x_m(k) + B_m u(k) - A_m \widehat{x}_m(k) - B_m u(k) \\
 &= A_m (x_m(k) - \widehat{x}_m(k)) \\
 &= A_m \widehat{e}_m(k).
 \end{aligned} \tag{2.52}$$

Quindi considerando un'errore iniziale differente da zero $\widehat{x}_m(0) \neq 0$, si ottiene che:

$$\widehat{e}_m(k) = A_m^k \widehat{e}_m(0). \tag{2.53}$$

A questo punto occorre considerare due punti essenziali che limitano questo tipo di soluzione.

1. In primo luogo se A_m ha tutti gli autovalori confinati all'interno della circonferenza di raggio unitario, allora l'errore di sistema risulta stabile e la sua norma converge a zero ovvero risulta che $\|\widehat{e}_m(k)\| \rightarrow 0$ per $k \rightarrow \infty$, il che significa che il valore della variabile stimata $\widehat{x}_m(k)$ converge a quello effettivo $x_m(k)$.

Tuttavia, se A_m ha uno o più autovalori situati all'esterno della circonferenza di raggio unitario, allora l'errore di sistema $\widehat{e}_m(k)$ risulta instabile e la sua norma diverge, ovvero $\|\widehat{e}_m(k)\| \rightarrow \infty$ per $k \rightarrow \infty$, il che significa che la nostra predizione $\widehat{x}_m(k)$ non converge al valore effettivo di $x_m(k)$.

In fine, se A_m ha uno o più autovalori sulla frontiera della regione di convergenza, gli errori relativi alle variabili di stato $\widehat{e}_m(k)$ non convergeranno a zero.

2. In secondo luogo, se il modello A_m del processo risulta stabile, non abbiamo controllo sulla velocità di convergenza dell'errore $\|\widehat{e}_m(k)\| \rightarrow 0$; che dipende dalla posizione dei poli del processo. In particolare, se i poli del processo sono prossimi all'origine del piano complesso, allora l'errore converge velocemente a zero; altrimenti la velocità di convergenza potrebbe essere più lenta.

A questo punto ci si domanda come sia possibile migliorare la nostra stima di $x_m(k)$. La soluzione va ricercata nel principio di retro-azione dove il segnale di errore viene utilizzato per migliorare la stima. L'osservatore è costruito a partire dalla seguente equazione:

$$\widehat{x}(k+1) = \overbrace{A_m \widehat{x}_m(k) + B_m u(k)}^{\text{model}} + \overbrace{K_{ob}(y(k) - C_m \widehat{x}_m(k))}^{\text{correction-term}}, \tag{2.54}$$

dove K_{ob} è la matrice dei guadagni dell'osservatore. Nella struttura dell'osservatore, la variabile di stato stimata $\widehat{x}_m(k+1)$ è composta da due termini.

Il primo termine fa riferimento al modello di partenza adottato per descrivere il processo; mentre il secondo è un termine di correzione basato sull'errore di previsione. Tale errore è definito dalla differenza tra l'uscita misurata ed il valore di output che era stato previsto per l'istante di campionamento corrente, in base alla stima di $\widehat{x}_m(k)$. Quindi, l'Osservatore sulla base dell'errore commesso nello step precedente cerca di migliorare la stima successiva. Il miglioramento viene realizzato scegliendo opportunamente la matrice dei guadagni K_{ob} .

Al fine di determinare la matrice dei guadagni K_{ob} , esamineremo l'equazione che definisce l'errore ad anello chiuso, in modo da definirla in funzione dell'errore di stima precedente per costruire un'algoritmo di calcolo ricorsivo. Innanzitutto, esprimiamo l'uscita misurata, in funzione delle variabili di stato del sistema. Ovvero, sostuiamo :

$$y(k) = C_m x_m(k) \tag{2.55}$$

Nell'equazione dell'osservatore (2.54).

$$\begin{aligned}
 \widehat{x}(k+1) &= A_m \widehat{x}_m(k) + B_m u(k) + K_{ob}(C_m x_m(k) - C_m \widehat{x}_m(k)) \\
 &= A_m \widehat{x}_m(k) + B_m u(k) + K_{ob} C_m (x_m(k) - \widehat{x}_m(k)) \\
 &= A_m \widehat{x}_m(k) + B_m u(k) + K_{ob} C_m \widehat{e}_m(k)
 \end{aligned} \tag{2.56}$$

Consideriamo ora l'equazione che definisce l'errore:

$$\begin{aligned}
 \widehat{e}_m(k+1) &= x_m(k+1) - \widehat{x}_m(k+1) \\
 &= A_m x_m(k) + B_m u(k) - A_m \widehat{x}_m(k) - B_m u(k) - K_{ob} C_m \widehat{e}_m(k) \\
 &= A_m (x_m(k) - \widehat{x}_m(k)) - K_{ob} C_m \widehat{e}_m(k) \\
 &= A_m \widehat{e}_m(k) - K_{ob} C_m \widehat{e}_m(k) \\
 &= (A_m - K_{ob} C_m) \widehat{e}_m(k)
 \end{aligned} \tag{2.57}$$

Quindi, considerando le condizioni iniziali relative all'errore di partenza $\widehat{e}_m(0)$ si ottiene la relazione desiderata:

$$\widehat{e}_m(k) = (A_m - K_{ob} C_m)^k \widehat{e}_m(0). \tag{2.58}$$

Confrontando la relazione che descrive l'evoluzione temporale dell'errore in funzione dell'errore commesso all'istante precedente (2.58) (closed-loop error equation), con quella ad anello aperto che si basava unicamente sul guess iniziale dello stato del sistema, e quindi dell'errore iniziale (2.53), risulta chiaro che la matrice di guadagno K_{ob} può essere utilizzata per regolare la velocità di convergenza dell'errore. Se il sistema presenta una sola uscita, una tecnica piuttosto comune è quella di assegnare gli autovalori della legge ad anello chiuso della matrice dinamica che descrive l'andamento dell'errore di sistema ($A_m - K_{ob} C_m$) in una zona desiderata del piano complesso a seconda della velocità di convergenza specificata. Il seguente esempio mostra come selezionare i guadagni della matrice K_{ob} .

Esempio 2.6: Consideriamo l'equazione linearizzata del moto del pendolo:

$$\frac{d^2\theta}{dt^2} + \omega^2\theta = u. \tag{2.59}$$

dove θ è la posizione angolare del pendolo. Si desidera progettare un'osservatore che consenta di ricostruire la posizione angolare del pendolo, a partire dalle misure della velocità angolare $\frac{d\theta}{dt}$. Prendiamo in considerazione una velocità angolare pari a $\omega = 2rad/sec$ ed un'intervallo di campionamento pari a $\Delta t = 0.1(sec)$. Si desidera piazzare i poli dell'osservatore, rispettivamente in 0.1 e 0.2. Di seguito confronteremo la situazione in cui la nostra stima venga effettuata in open-loop ed in closed-loop.

Soluzione: Indichiamo con la variabile di stato $x_1(t) = \theta$ la posizione angolare del pendolo e con $x_2(t) = \dot{\theta}$ la sua velocità, a partire dall'equazione che descrive il moto del pendolo si ottiene la seguente rappresentazione nello spazio di stato:

$$\begin{aligned}
 \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\
 y(t) &= [0 \quad 1] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}.
 \end{aligned} \tag{2.60}$$

Quindi a partire dalle informazioni relative a $\omega = 2rad/sec$ ed un'intervallo di campionamento pari a $\Delta t = 0.1(sec)$, otteniamo il corrispondente modello nello spazio di stato a tempo discreto:

$$\begin{aligned}
 \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} 0.9801 & 0.0993 \\ -0.3973 & 0.9801 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0.0050 \\ 0.09930 \end{bmatrix} u(k) \\
 y(k) &= [0 \quad 1] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}.
 \end{aligned} \tag{2.61}$$

- **open-loop:** Nell'effettuare la nostra analisi, incominceremo a studiare il comportamento del nostro sistema utilizzando solamente il modello matematico che descrive il moto del pendolo, per prevedere la posizione angolare $\theta(x_1)$. A partire dall'ipotesi che il segnale di ingresso $u(k) = 0$ e le condizioni iniziali relative alle variabili di stato $\theta(0) = x_1(0) = 1$ e $\dot{\theta}(0) = x_2(0) = 0$. Le traiettorie che descrivono l'andamento sia di θ che di $\dot{\theta}$ sono riportate in figura (2.17) Si evince che θ e $\dot{\theta}$ sono segnali sinusoidali. Ora, supponendo di conoscere le condizioni iniziali relative alle variabili di stato del processo $\widehat{x}_1(0) = 0.3$ e $\widehat{x}_2(0) = 0$, utilizzando la seguente relazione:

$$\widehat{x}_m(k+1) = A_m \widehat{x}_m(k) + B_m u(k).$$

otteniamo gli andamenti delle stime delle variabili θ e $\dot{\theta}$. (figura (2.17)) Se confrontiamo la traiettoria stimata con quella reale si può constatare che la stima di θ , denotata con \widehat{x}_1 , risulta lontana da quella effettiva del sistema. Questo esempio mostra come non sia sufficiente l'utilizzo del modello matematico del processo per prevedere la posizione angolare del pendolo.

- **closed-loop:** Ora, sulla base dei risultati ottenuti precedentemente, ci accingiamo a progettare un'Osservatore che preveda la posizione angolare del pendolo. Quindi, indicando in forma parametrica la matrice dei guadagni dell'osservatore:

$$K_{ob} = [j_1 \quad j_2]^T.$$

il polinomio caratteristico dell'osservatore risulta quindi:

$$\begin{aligned} \rho(\lambda) &= \det(\lambda I - \begin{bmatrix} 0.9801 & 0.0993 - j_1 \\ -0.3973 & 0.9801 - j_2 \end{bmatrix}) \\ &= (\lambda - 0.9801)(\lambda + j_2 - 0.9801) - 0.3973 \times (j_1 - 0.0993), \end{aligned}$$

Eguagliamo quindi il polinomio caratteristico con la configurazione dei poli assegnata $(\lambda - 0.1)(\lambda - 0.2)$:

$$(\lambda - 0.9801)(\lambda + j_2 - 0.9801) - 0.3973 \times (j_1 - 0.0993) = (\lambda - 0.1)(\lambda - 0.2).$$

La soluzione dell'equazione polinomiale rispetto alle variabili j_1 e j_2 che sono i guadagni ricercati, è:

$$K_{ob} = [j_1 \quad j_2]^T = [-1.6284 \quad 1.6601]^T.$$

Quindi utilizzando l'equazione che definisce il nostro osservatore asintotico dello stato, possiamo stimare la posizione angolare del pendolo:

$$\begin{bmatrix} \widehat{x}_1(k+1) \\ \widehat{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.9801 & 0.0993 \\ -0.3973 & 0.9801 \end{bmatrix} \begin{bmatrix} \widehat{x}_1(k) \\ \widehat{x}_2(k) \end{bmatrix} + K_{ob}(x_2(k) - \widehat{x}_2(k)),$$

con le condizioni iniziali $\widehat{x}_1(0) = 0.3$ e $\widehat{x}_2(0) = 0$. La figura(2.18) mostra come la nostra stima converga al valore effettivo dell'angolo in soli tre step.

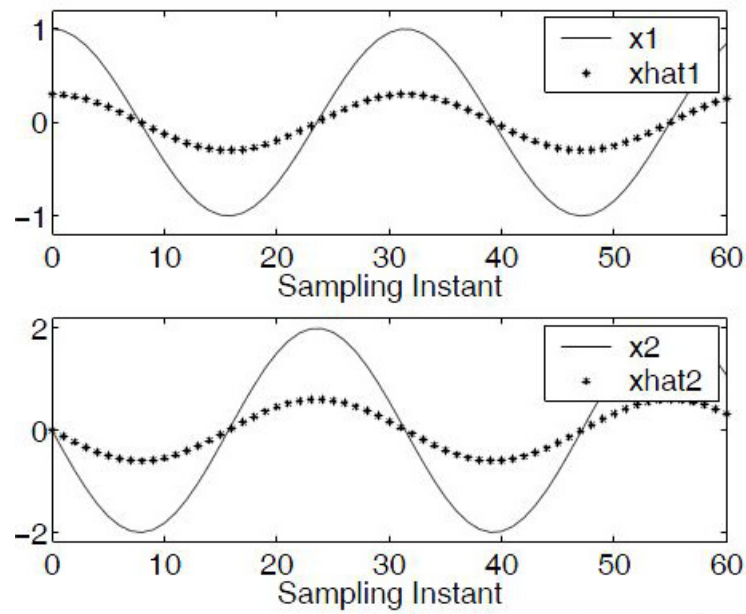


Figura 2.17: Estimation without observer

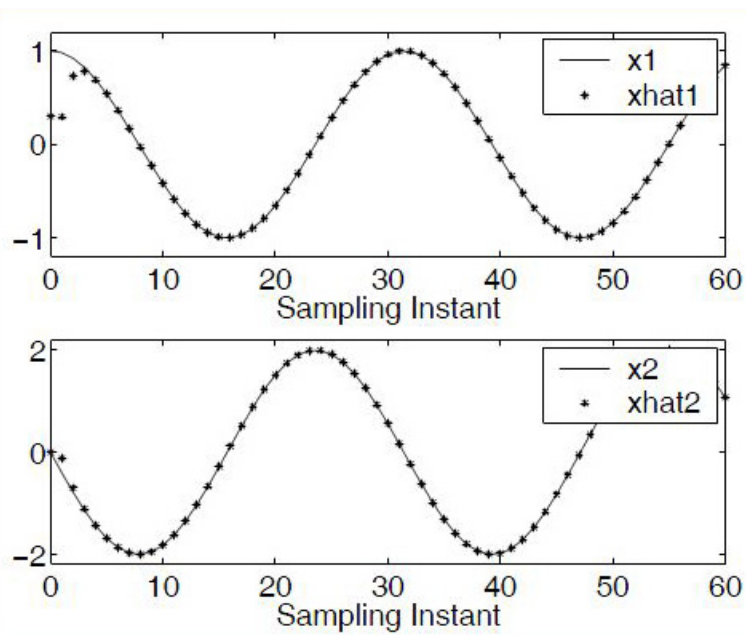


Figura 2.18: Estimation with observer

2.6.2 Basic Results About Observability

Abbiamo visto nella sezione precedente come sia stato possibile determinare i guadagni necessari per consentire all'osservatore di stimare lo stato del processo. Tuttavia, è importante ricordare che è possibile ricorrere a questa tecnica in fase di progettazione solo se il sistema risulti Osservabile. Pertanto è necessario stabilire se il sistema in oggetto abbia questa proprietà strutturale. A tale proposito, richiamiamo brevemente il concetto di osservabilità e stabiliamo le condizioni necessarie e sufficienti affinché un sistema a tempo discreto risulti osservabile.

Possiamo definire l'osservabilità, come quella proprietà che ci consente di determinare cosa 'stesse facendo' il sistema in un'istante temporale arbitrario, t_0 , dopo avere misurato l'andamento dell'uscita per un'arco temporale limitato, considerando ovviamente come istante d'inizio t_0 . È importante sottolineare il fatto che l'osservabilità è una proprietà globale del sistema, o se vogliamo è indipendente dalla scelta di t_0 ; infatti potrebbe essere possibile determinare alcuni stati iniziali del sistema registrando le uscite solo per particolari istanti temporali e quindi non rispettare la definizione 'per il momento qualitativa' di osservabilità. Risulta chiaro che l'Osservabilità richiede che tutti gli stati del sistema contribuiscano all'uscita, altrimenti non sarebbe possibile ricostruire tutte le possibili combinazioni delle variabili di stato (i.e. qualsiasi stato iniziale) misurando l'output del sistema. La relazione che lega l'Osservabilità all'uscita è Duale a quella esistente tra l'ingresso e la Controllabilità del sistema. (Si parla infatti di Controllability-Observability Duality).

In un sistema che risulta Controllabile, tutti gli ingressi devono influire sulle variabili di stato del sistema,; mentre per un sistema Osservabile tutte le variabili di stato devono influire sull'output. Ovviamente se qualche v.d.s non contribuisce all'output, allora il sistema si definisce non osservabile. Ora formalizziamo questi concetti, dando delle condizioni algebriche necessarie e sufficienti affinché un sistema sia Osservabile.

Un modello descritto in termini di v.d.s di un sistema dinamico, si definisce completamente osservabile se, per un'arbitrario istante di campionamento k_0 , esiste un'istante di campionamento successivo $k_1 > k_0$ tale che la conoscenza dell'output $y(k)$ e dell'input $u(k)$ nell'intervallo temporale $k_0 \leq k \leq k_1$ è sufficiente per determinare lo stato iniziale del sistema $x_m(k_0)$ e di conseguenza tutti gli stati successivi, ovvero $x_m(k)$ con $k_0 < k \leq k_1$.

Risulta chiaro che la definizione precedente ci fornisce una condizione sufficiente, tuttavia ai fini pratici noi siamo interessati ad una condizione necessaria e sufficiente. Quindi, la condizione algebrica necessaria e sufficiente affinché un sistema lineare descritto nello spazio di stato a tempo discreto risulti completamente osservabile è che la matrice di osservabilità:

$$L_o = \begin{bmatrix} C_m \\ C_m A_m \\ C_m A_m^2 \\ \vdots \\ C_m A_m^{n-1} \end{bmatrix} \quad (2.62)$$

abbia rango pari a n , dove n è la dimensione del vettore delle variabili di stato del modello. Per mostrare quanto risulti importante l'Osservabilità di un sistema consideriamo il seguente esempio.

Esempio 2.6: Il comportamento di un motore DC può essere descritto da un modello del secondo ordine composto da un'integratore ed una costante di tempo.

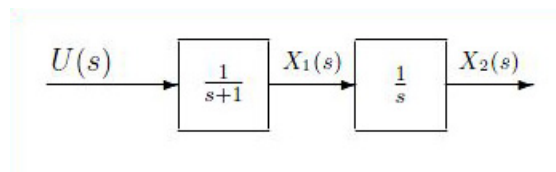


Figura 2.19: Motor model

La grandezza di input è la tensione di eccitazione del motore e quella di output è la posizione dell'albero di trasmissione. La costante di tempo è dovuta principalmente alle parti meccaniche del sistema, in quanto le dinamiche legate ai componenti elettrici hanno costanti di tempo molto piccole rispetto a quelle meccaniche e quindi sono trascurabili.

Indicando con x_1 la velocità angolare e con x_2 la posizione angolare dell'albero di trasmissione, otteniamo la seguente rappresentazione a tempo continuo nello spazio di stato:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t). \quad (2.63)$$

Supponiamo di misurare la velocità di rotazione, che ci porta dunque alla seguente equazione:

$$y(t) = [1 \ 0] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}. \quad (2.64)$$

Discretizziamo il processo utilizzando un'intervallo di campionamento pari a $\Delta t = 0.1$:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.9048 & 0 \\ 0.0952 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0.0952 \\ 0.0048 \end{bmatrix} u(k) \quad (2.65)$$

$$y(k) = [1 \ 0] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}.$$

Verifichiamo che il sistema a tempo discreto risulta non è osservabile e come conseguenza il sistema caratterizzato dall'osservatore ad anello chiuso presenta un polo in $z = 1$.

Soluzione: Il sistema ad anello aperto ha gli autovalori situati in $z_1 = 0.9048$ e $z_2 = 1$. La matrice di osservabilità è:

$$L_o = \begin{bmatrix} C_m \\ C_m A_m \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0.9048 & 0 \end{bmatrix},$$

dove $\det(L_o) = 0$. Quindi, la coppia (C_m, A_m) non è osservabile. Procediamo nella nostra analisi per valutare effettivamente cosa accade se il sistema che desideriamo controllare non risulta osservabile. Indicando come al solito con $K_{ob} = [j_1 \ j_2]^T$. Allora l'evoluzione dell'errore di predizione del sistema caratterizzato dall'osservatore è:

$$\hat{e}(k+1) = \left[\begin{bmatrix} 0.9048 & 0 \\ 0.0952 & 1 \end{bmatrix} - \begin{bmatrix} j_1 \\ j_2 \end{bmatrix} [1 \ 0] \right] \hat{e}(k). \quad (2.66)$$

Supponiamo di posizionare gli autovalori dell'osservatore in 0.1 e 0.2⁴. Allora il polinomio caratteristico ad anello chiuso è:

$$\det \begin{bmatrix} \lambda - 0.9048 + j_1 & 0 \\ -0.0952 & \lambda - 1 \end{bmatrix} = (\lambda - 0.9048 + j_1)(\lambda - 1). \quad (2.67)$$

Nel progettare l'osservatore, eguagliamo il polinomio caratteristico del sistema con il polinomio caratteristico che riflette la configurazione ad anello chiuso desiderata e risolviamo l'equazione rispetto al vettore dei guadagni dell'osservatore. In questo caso otteniamo:

$$(\lambda - 0.9048 + j_1)(\lambda - 1) = (\lambda - 0.1)(\lambda - 0.2).$$

Dal polinomio caratteristico che descrive il sistema possiamo osservare che comunque scegliamo il guadagno relativo a j_2 , il secondo polo in $\lambda = 1$ non può essere spostato. Questo semplicemente perchè tale polo risulta indipendente dal guadagno dell'osservatore. Questa è la conseguenza 'algebraica' legata al fatto che la coppia (C_m, A_m) non è osservabile. Comunque, se non è possibile cambiare la posizione del polo mediante un'osservatore ma tale configurazione è asintoticamente stabile allora il sistema non è osservabile ma risulta rilevabile. In particolare in questo esempio il polo non osservabile è situato sulla frontiera della regione di convergenza e quindi il sistema non risulta rilevabile. Pertanto, se misuriamo la velocità angolare del motore, allora la posizione angolare non può essere stimata in modo accurato a partire da questa unica misura.

⁴Tali autovalori prendono il nome di closed-loop poles of the observer

2.6.3 Kalman Filter

Abbiamo visto precedentemente come nel caso di un sistema costituito da una sola uscita, se la coppia (A_m, C_m) risulta osservabile, allora è possibile utilizzare la tecnica di assegnazione degli autovalori per determinare la matrice dei guadagni K_{ob} tale che gli autovalori dell'osservatore (*i.e.*, della matrice $A_m - K_{ob}C_m$) vengano posizionati nella configurazione desiderata. Nel caso di più uscite, K_{ob} può essere calcolata in modo ricorsivo utilizzando il filtro di Kalman. I filtri di Kalman vengono utilizzati in sistemi stocastici. Quindi il nostro modello è dato dalla seguente rappresentazione nello spazio di stato:

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k) + d(k) \\ y(k) &= C_m x_m(k) + \xi(k), \end{aligned} \quad (2.68)$$

dove il vettore dei disturbi che agiscono sul processo $d(k)$ e sull'uscita del sistema $\xi(k)$ sono generati da processi aleatori. Dato che i disturbi di diversa natura che possono agire sulle variabili di stato del processo e sulle misurazioni effettuate dai sensori utilizzati sono dei processi aleatori, la risposta del sistema, sia che venga considerata ad anello-aperto o con un sistema di controllo in retro-azione, anche essa risulta un processo aleatorio. Pertanto dobbiamo studiare il nostro sistema considerando 'in media' come si comporta. Detto in parole povere dobbiamo caratterizzare il comportamento complessivo del nostro sistema attraverso delle grandezze statistiche che classificano la natura aleatoria del processo. In particolare possiamo considerare le statistiche del primo ordine per caratterizzare le nostre variabili di stato che in questa circostanza sono delle variabili aleatorie. Da questo emerge che le matrici d e ξ altro non sono che matrici di covarianza definite nel seguente modo:

$$\begin{aligned} E d(k) d(\tau)^T &= \Theta \delta(k - \tau) \\ E \xi(k) \xi(\tau)^T &= \Gamma \delta(k - \tau) \end{aligned}$$

dove $\delta(k - \tau) = 1$ se $k = \tau$ e $\delta(k - \tau) = 0$ se $k \neq \tau$. Il guadagno ottimo dell'osservatore si ottiene risolvendo ricorsivamente la seguente equazione per $i = 0, 1, \dots$:

$$K_{ob}(i) = A_m P(i) C_m^T (\Gamma + C_m P(i) C_m^T)^{-1}, \quad (2.69)$$

dove definiamo:

$$P(i+1) = A_m P(i) - P(i) C_m^T (\Gamma + C_m P(i) C_m^T)^{-1} C_m P(i) A_m^T + \Theta. \quad (2.70)$$

Nello specifico, $P(0)$ soddisfa la seguente relazione:

$$E[x(0) - \hat{x}(0)][x(0) - \hat{x}(0)]^T = P(0).$$

Supponendo che il sistema (C_m, A_m) sia rilevabile dall'uscita $y(k)$ (*i.e.* non abbiamo stati instabili che non possono essere visti dall'uscita) e $(A_m, \Theta^{frac{12})}$ è stabilizzabile, allora per $k \rightarrow \infty$, la soluzione a regime dell'equazioni (2.69) e (2.70) soddisfano l'equazione algebrica di Riccati a tempo discreto:

$$P(\infty) = A_m P(\infty) - P(\infty) C_m^T (\Gamma + C_m P(\infty) C_m^T)^{-1} C_m P(\infty) A_m^T + \Theta, \quad (2.71)$$

e

$$K_{ob}(\infty) = A_m P(\infty) C_m^T (\Gamma + C_m P(\infty) C_m^T)^{-1}. \quad (2.72)$$

Anche, gli autovalori di $A_m - K_{ob}(\infty) C_m$ è garantito che risiedono all'interno della circonferenza di raggio unitario (*i.e.* sono stabili). Per non fare confusione, vorrei sottolineare che la soluzione iterativa dell'equazione di Riccati non è richiesta in real time. Il guadagno dell'osservatore può essere calcolato off-line per le applicazioni di controllo predittivo.

Tuning Observer Dynamics

In molte circostanze le matrici di covarianza Θ e Γ , corrispondenti alle caratteristiche dei disturbi, sono sconosciute. Quindi, in pratica si scelgono Θ , Γ e una condizione iniziale $P(0)$ per calcolare la matrice dei guadagni dell'osservatore K_{ob} attraverso la risoluzione iterativa dell'equazione di Riccati fino a quando la soluzione non converge ad una matrice costante. Allora, il sistema ad anello chiuso che si ottiene viene analizzato rispetto alla posizione degli autovalori contenuti nella matrice $(A_m - K_{ob}C_m)$, la risposta transitoria dell'osservatore, robustezza e l'effetto del rumore sulla risposta. Gli elementi della matrice di covarianza vengono modificati fino a quando non si ottiene il risultato desiderato. Questo procedimento di tipo trial-and-error può richiedere molto tempo ed ovviamente abbastanza frustrante, ed è una delle principali sfide con le quali ci dobbiamo confrontare al momento in cui ricorriamo all'utilizzo di un filtro di Kalman per la progettazione di un sistema con più variabili. Comunque, in alcuni casi è possibile specificare una regione nella quale i poli relativi all'errore di predizione ad anello chiuso dell'osservatore dovrebbero essere posizionati, in modo da vincolare la soluzione. Qui di seguito proponiamo una semplice tecnica nella quale la configurazione dei poli ad anello chiuso vengono assegnati all'interno di una circonferenza di raggio predefinito α ($0 < \alpha < 1$).

Come al solito definiamo l'errore di stima nel seguente modo:

$$\widehat{e}(k) = x(k) - \widehat{x}(k).$$

Allora l'errore di sistema dell'osservatore è definito dalla seguente relazione:

$$\widehat{e}(k+1) = (A_m - K_{ob}C_m)\widehat{e}(k) \quad (2.73)$$

Adesso effettuiamo la seguente trasformazione:

$$\widehat{A}_m = \frac{A_m}{\alpha}; \quad \widehat{C}_m = \frac{C_m}{\alpha}; \quad 0 < \alpha < 1.$$

Quindi otteniamo il sistema trasformato:

$$\widehat{e}_t(k+1) = \frac{1}{\alpha}(A_m - \widehat{K}_{ob}\widehat{C}_m)\widehat{e}_t(k) = (\widehat{A}_m - \widehat{K}_{ob}\widehat{C}_m)\widehat{e}_t(k). \quad (2.74)$$

Quindi risolvendo iterativamente le equazioni (2.69) e (2.70), oppure l'equazione di Riccati a regime (2.71) utilizzando \widehat{A}_m e \widehat{C}_m al posto delle matrici A_m e C_m , è possibile garantire che gli autovalori della matrice $(\widehat{A}_m - \widehat{K}_{ob}(\infty)\widehat{C}_m)$ siano confinati all'interno della circonferenza di raggio unitario, ovvero il nostro sistema di controllo risulta stabile. Dunque la risultante matrice dei guadagni dell'osservatore \widehat{K}_{ob} viene applicata al sistema originario dell'osservatore (2.78) che ci conduce alla seguente equazione caratteristica ad anello chiuso:

$$\det(zI - (A_m - \widehat{K}_{ob}\widehat{C}_m)) = \det(zI - (\widehat{A}_m - \widehat{K}_{ob}\widehat{C}_m) \times \alpha) = 0. \quad (2.75)$$

Pertanto, possiamo concludere che gli autovalori di $(A_m - \widehat{K}_{ob}\widehat{C}_m)$ risultano equivalenti agli autovalori della stessa matrice $(A_m - K_{ob}C_m)$ moltiplicati per un fattore α che garantisce che gli autovalori che determinano l'evoluzione temporale dell'errore di sistema dell'osservatore con \widehat{K}_{ob} risiedano all'interno della circonferenza di raggio α . Questo procedimento ci consente di avere un contatto diretto con le dinamiche dell'osservatore attraverso la scelta del parametro α . Quindi la tecnica trial-and-error si riduce effettivamente sulla scelta di un valore accettabile di α insieme a Θ ed Γ per ottenere la performance del sistema ad anello-chiuso desiderata.

2.7 State Estimate Predictive Control

Nelle sezioni precedenti abbiamo visto come sia possibile ricorrere all'utilizzo di un'osservatore per implementare una legge di controllo predittivo nel caso in cui la variabile di stato $x(k_i)$ all'istante temporale k_i non sia misurabile. In sostanza tale variabile viene stimata mediante un'osservatore con la seguente forma:

$$\widehat{x}(k_i+1) = A\widehat{x}(k_i) + B\Delta u(k_i) + K_{ob}(y(k_i) - C\widehat{x}(k_i)). \quad (2.76)$$

In particolare si osserva che nell'implementazione del controllo predittivo attraverso l'utilizzo di un'osservatore, il segnale di controllo è $\Delta u(k_i)$ mentre la terna di matrici (A, B, C) derivano dal modello esteso utilizzato nella progettazione del controllo predittivo. In sostanza ciò che andiamo a sostituire è $x(k_i)$ con l'informazione acquisita dall'osservatore $\hat{x}(k_i)$, che in qualche modo possiamo considerarlo come una sorta di sensore virtuale. Quindi, la legge di controllo predittivo viene modificata per calcolare il segnale di controllo ΔU attraverso la minimizzazione del funzionale di costo J :

$$J = (R_s - F\hat{x}(k_i))^T (\bar{R}_s r(k_i) - F\hat{x}(k_i)) - 2\Delta U^T \Phi^T (R_s - F\hat{x}(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U. \quad (2.77)$$

dove le matrici \bar{R}, F, Φ, R e ΔU sono definiti dall'equazione (2.8). La soluzione ottima ΔU si ottiene nel modo seguente:

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - F\hat{x}(k_i)). \quad (2.78)$$

Inoltre, attraverso l'applicazione del principio RHC giungiamo all'implementazione del segnale di controllo ottimo all'istante temporale k_i :

$$\Delta u(k_i) = K_y r(k_i) - K_{mpc} \hat{x}(k_i), \quad (2.79)$$

la quale rappresenta la forma standard della legge di controllo in retro azione con il valore stimato del vettore di stato $x(k_i)$. Lo schema a blocchi della legge di controllo ad anello chiuso che tiene conto anche della presenza dell'osservatore è riportata di seguito.

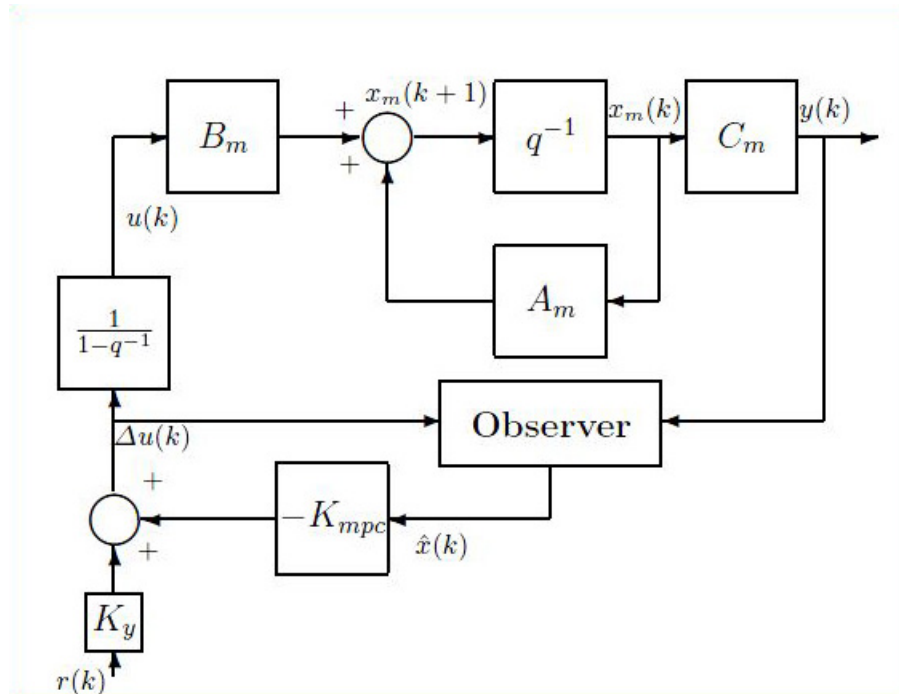


Figura 2.20: Block diagram of DMPC with observer

A questo punto ci si domanda come venga modificata l'equazione caratteristica del sistema ad anello chiuso e quindi gli autovalori del sistema. Andiamo quindi ad indagare su queste questioni. Consideriamo l'equazione di stato ad anello chiuso del sistema:

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) \\ &= Ax(k) + BK_y r(k) - BK_{mpc} \hat{x}(k). \end{aligned} \quad (2.80)$$

Nella quale abbiamo sostituito il segnale di controllo $\Delta u(k)$. Per quanto riguarda la dinamica dell'errore di predizione ad anello chiuso dell'osservatore è data dalla seguente equazione:

$$\widehat{e}(k+1) = (A - K_{ob}C)\widehat{e}(k). \quad (2.81)$$

dove ricordo che l'errore è definito come $\widehat{e}(k) = x(k) - \widehat{x}(k)$. Sostituendo $\widehat{x}(k)$ con $x(k) - \widehat{e}(k)$, possiamo riscrivere l'equazione (2.80) come:

$$x(k+1) = (A - BK_{mpc})x(k) - BK_{mpc}\widehat{e}(k) + BK_y r(k). \quad (2.82)$$

Combinando le due relazioni precedenti si ottiene:

$$\begin{bmatrix} \widehat{e}(k+1) \\ x(k+1) \end{bmatrix} = \begin{bmatrix} A - K_{ob}C & o_{n \times n} \\ -BK_{mpc} & A - BK_{mpc} \end{bmatrix} \begin{bmatrix} \widehat{e}(k) \\ x(k) \end{bmatrix} + \begin{bmatrix} o_{n \times m} \\ BK_y \end{bmatrix} r(k), \quad (2.83)$$

dove $o_{n \times n}$ è una matrice $n \times n$ composta da zeri e $o_{n \times m}$ è una matrice $n \times m$ composta da zeri. Quindi l'equazione caratteristica del sistema ad anello chiuso è determinata da:

$$\det \left[\lambda I - \begin{bmatrix} A - K_{ob}C & o_{n \times n} \\ -BK_{mpc} & A - BK_{mpc} \end{bmatrix} \right] = 0, \quad (2.84)$$

dalla quale si può osservare che è una matrice a blocchi triangolare inferiore e quindi si ha:

$$\det(\lambda I - (A - K_{ob}C)) \det(\lambda I - BK_{mpc}) = 0,$$

Dal polinomio caratteristico si può osservare che il sistema di controllo predittivo ad anello chiuso con la stima dello stato ha due equazioni caratteristiche indipendenti:

$$\det(\lambda I - (A - K_{ob}C)) = 0 \quad (2.85)$$

$$\det(\lambda I - (A - BK_{mpc})) = 0. \quad (2.86)$$

Dato che gli autovalori ad anello chiuso sono le radici dell'equazione caratteristica, le relazioni (2.85) e (2.86) rappresentano l'insieme degli autovalori del sistema composti da quelli del controllore predittivo e quelli dell'osservatore. Questo ha un'importanza notevole nella fase di progettazione, poichè la progettazione della legge di controllo e della struttura dell'osservatore possono essere condotte separatamente, in quanto risultano indipendenti tra di loro o detto in altri termini, i loro autovalori rimangono gli stessi.

Esempio: Consideriamo il modello studiato nell'esempio 2.1 costituito da due integratori:

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

Dove ricordo che le matrici sono:

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

Adesso progetteremo un sistema di controllo predittivo con stimatore dello stato e simuleremo la risposta ad anello chiuso per un segnale di set-point costante. Le specifiche di progetto sono $N_c = 5$, $N_p = 30$, il peso da attribuire all'azione del segnale di controllo è $r_w = 10$. L'osservatore verrà progettato utilizzando la tecnica dell'assegnazione degli autovalori, nella quale la configurazione specificata per i poli dell'osservatore è $0.01 \quad 0.0105 \quad 0.001$ la quale corrisponde ad una veloce risposta da parte dell'osservatore e quindi il tempo di convergenza dell'errore è relativamente basso.

Soluzione: Il sistema ad anello aperto ha tre autovalori situati in 1, dove due di questi rappresentano i due integratori presenti nel modello effettivo del processo ed uno alla struttura del controllore predittivo. Possiamo adoperare il comando in MATLAB 'place' per appunto 'piazzare' gli autovalori nei punti specificati, al fine di ottenere il vettore dei guadagni dell'osservatore K_{ob} .

Matlab code : Di seguito riportiamo le due istruzioni Matlab.

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

dove A' e C' sono le matrici A^T e C^T . Le trasposte sono necessarie poichè la funzione 'place.m' è stata scritta per la progettazione dei controllori. Utilizzando questo programma abbiamo la relazione duale tra controllore ed osservatore. Il risultante vettore dei guadagni è:

$$K_{ob} = [1.9865 \quad 0.9688 \quad 2.9685]^T.$$

Quindi a partire da questo insieme di parametri di performance specificati è possibile calcolare il vettore dei guadagni del controllore:

$$K_{mpc} = [0.8984 \quad 1.3521 \quad 0.4039]$$

Il quale effettivamente porta gli autovalori del sistema ad anello chiuso in $\lambda_{1,2} = 0.3172 \pm j0.4089$ e $\lambda_3 = 0.3624$. La figura che segue mostra la risposta del sistema ad anello chiuso per un riferimento a gradino.

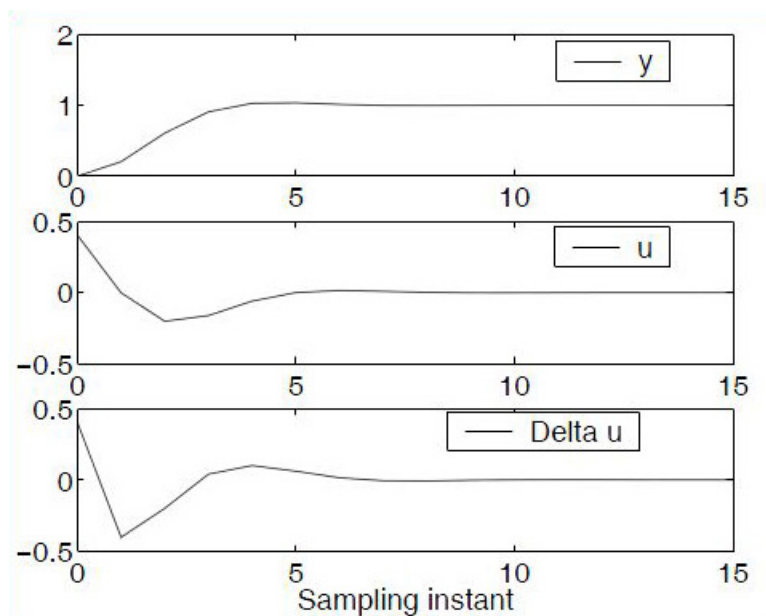


Figura 2.21: Predictive Control of double integrator plant

Capitolo 3

Appendice

3.1 Appendice

Listing 3.1: Modello esteso

```
1  %% l'obiettivo di questo file è di mostrare come ottenere un modello esteso
2  % a partire da una rappresentazione nello spazio di stato a tempo continuo.
3
4  % Processo a tempo continuo%
5
6  Ac = [0 1 0;
7        3 0 1;
8        0 1 0];
9  Bc = [1;
10        1;
11        3];
12 Cc = [0 1 0];
13 Dc = zeros(1,1);
14
15 % tempo di campionamento
16 Delta.t = 1;
17
18 % Discretizzazione del processo
19 [Ad,Bd,Cd,Dd]= c2dm(Ac,Bc,Cc,Dc,Delta.t);
20
21 % trasformazione nel modello esteso, vengono calcolate le dimensioni delle
22 % delle matrici del sistema per determinare il numero degli stati, ingressi
23 % ed uscite del processo in modo da costruire la struttura dell extmodel %
24
25 [m1,n1] = size(Cd);
26 [n1,n_in] = size(Bd);
27 A_e = eye(n1+m1, n1+m1);
28
29 % trasformazione nel modello esteso%
30
31 %struttura della matrice di transizione%
32 A_e(1:n1, 1:n1) = Ad;
33 A_e(n1+1:n1+m1, 1:n1) = Cd*Ad;
34
35 %struttura della matrice degli ingressi%
36 B_e = zeros(n1+m1,n_in);
37 B_e(1:n1,:) = Bd;
38 B_e(n1+1:n1+m1,:) = Cd*Bd;
39
40 %struttura della matrice degli output%
41 C_e = zeros(m1,n1+m1);
42 C_e(:,n1+1:n1+m1) = eye(m1,m1);
```

Listing 3.2: Mpc Gain

```

1  % L'obiettivo di questo file è di realizzare una funzione che consenta di
2  % calcolare i guadagni del sistema MPC GAINS. Per una descrizione
3  % dettagliata vedere il tutorial che ho scritto nel capitolo 2
4
5  %step 1%
6
7  function [Phi_Phi, Phi_F, Phi_R, A_e, B_e, C_e] = mpcgain(Ap, Bp, Cp, Nc, Np);
8  [m1,n1] = size(Cp);
9  [n1,n_in] = size(Bp);
10 A_e = eye(n1+m1, n1+m1);
11 A_e(1:n1, 1:n1) = Ap;
12 A_e(n1+1:n1+m1, 1:n1) = Cp*Ap;
13 B_e = zeros(n1+m1, n_in);
14 B_e(1:n1, :) = Bp;
15 B_e(n1+1:n1+m1, :) = Cp*Bp;
16 C_e = zeros(m1, n1+m1);
17 C_e(:,n1+1:n1+m1) = eye(m1,m1);
18
19 %step 2%
20
21 n = n1+m1;
22 h(1,:) = C_e;
23 F(1,:) = C_e*A_e;
24 for kk = 2:Np
25     h(kk,:) = h(kk-1,)*A_e;
26     F(kk,:) = F(kk-1,)*A_e;
27 end
28 v = h*B_e;
29 Phi = zeros(Np,Nc); %definisce le dimensioni di Phi
30 Phi(:,1)=v; %prima colonna di Phi
31 for i = 2:Nc
32     Phi(:,i) = [zeros(i-1,1);v(1:Np-i+1,1)]; %Toeplitz matrix
33 end
34
35 %step 3%
36
37 BarRs = ones(Np,1);
38 Phi_Phi = Phi'*Phi;
39 Phi_F = Phi'*F;
40 Phi_R = Phi'*BarRs;

```


Listing 3.3: Mpc no weight

```

1  % lo scopo di questo file è quello di realizzare un sistema di controllo
2  % predittivo utilizzando la tecnica del reeeding horizon control.
3  %
4
5  % Definiamo le matrici che definiscono il processo%
6
7  Ap = [0.8];
8  Bp = [0.1];
9  Cp = [1];
10 Dp = [0];
11
12
13 % Scelta dei parametri di progettazione:
14 % Np orizzonte di predizione
15 % Nc orizzonte di controllo
16
17 Np = 10;
18 Nc = 4;
19
20 % Richiamiamo la funzione mpc.gain per calcolare i guadagni delle matrici
21 % e specificare le condizioni iniziali per l'implementazione del receding
22 % horizon control. Lo stato iniziale per il processo è xm = 0, mentre
23 % quello relativo allo stato iniziale di feedback del modello esteso è Xf;
24 % il numero dei campioni del segnale di set-point è dato dal numero degli
25 % step simulativi N.sim mentre le variabili u ed y sono rispettivamente:
26 % l'ingresso all'istante u(k-1) e l'uscita all'istante k. Ed
27 % inizializziamo il valore dell'errore pari a zero.
28
29
30 [Phi_Phi, Phi_F, Phi_R, A_e, B_e, C_e] = mpcgain(Ap, Bp, Cp, Nc, Np);
31 [n, n_in] = size(B_e);
32 xm = 0.2;
33 %Xf = zeros(n,1);
34 Xf = [0.1;0.2];
35 N_sim = 20;
36 r = ones(N_sim,1);
37 u = 0; % u(k-1) = 0
38 y = 0.2;
39 e_2 = 0;
40
41 % Secondo il RHC, all'istante di campionamento kk, il vettore DeltaU è
42 % calcolato utilizzando il segnale di set-point r(kk) ed il vettore di
43 % stato Xf. Allora, Deltau(kk) è preso come primo elemento di DeltaU; e
44 % u(kk) = u(kk-1)+Deltau(kk). Impostiamo il peso pari a 0 (costante
45 % moltiplicativa che compare difronte alla matrice diagonale eye)
46 %
47
48 for kk = 1:N_sim;
49     DeltaU = inv(Phi_Phi+0*eye(Nc,Nc))*(Phi_R*r(kk)-Phi_F*Xf);
50     deltau = DeltaU(1,1);
51     u = u+deltau;
52
53 % andamento dei campioni dell'input,output e della sua derivata.
54
55     u1(kk) = u;
56     y1(kk) = y;
57     du(kk) = deltau;
58
59 % andamento dell'errore quadratico %
60
61     e_2(kk) = (r(kk)-y1(kk))^2;
62
63 % Lo stato del processo xm e l'output vengo fatti evolvere nella simulazione
64 % utilizzando il primo campione generato dalla sequenza ottima del segnale
65 % di controllo (u); le variabili di stato del processo vengono aggiornate
66 % nel meccanismo di retroazione del sistema reale con il valore dell'uscita
67 % corrente (u). Tali valori vengono poi utilizzati per aggiornare lo stato Xf

```

```
68 % del modello esteso.
69
70     xm_old = xm;
71     xm = Ap*xm+Bp*u;
72     y = Cp*xm;
73     Xf = [xm-xm_old;y];
74
75 % Andamento della variazione della v.d.s x1
76
77     Xf1(kk) = xm-xm_old;
78
79 end
80
81 % Il segnale di output viene plottato rispetto alla variazione delle v.d.s
82 %
83
84 k = 0:(N.sim-1);
85 figure
86 subplot(211)
87 plot(k,y1,'b')
88 hold on
89 plot(k,Xf1,'g')
90 xlabel('Istante di campionamento')
91 legend('Output','DeltaX1')
92
93 % andamento dell'input e della sua derivata
94
95 subplot(212)
96 plot(k,u1,'b')
97 hold on
98 plot(k,du,'r')
99 xlabel('Istante di campionamento')
100 legend('input','deltau')
101
102
103 % andamento dell'errore quadratico %
104
105 figure
106 subplot(211)
107 plot(k,e_2,'r')
108 xlabel('Istante di campionamento')
109 legend('Errore quadratico')
```