

**UNIVERSITA' DEGLI STUDI DI  
PADOVA**



**FACOLTA' DI INGEGNERIA**

**CORSO DI LAUREA IN INGEGNERIA  
DELL'AUTOMAZIONE**

**A.A. 2010/2011**

**Tesi di laurea**

**ALGORITMI DI MOTION CUIENG  
PER SIMULATORI DI VEICOLO**

**RELATORE**

Alessandro Beghi

**CANDIDATO**

Daniele D'Ambrosio

**CORRELATORE**

Mattia Bruschetta



*ai miei genitori*



# Indice

<b>Introduzione</b>	<b>vii</b>
<b>1 Storia dei Simulatori</b>	<b>1</b>
1.1 Strutture esapodali . . . . .	1
1.1.1 Primi simulatori . . . . .	2
1.1.2 Motion Cueing System . . . . .	3
1.2 Desdemona . . . . .	6
1.3 Piattaforma Vi-Grade . . . . .	7
<b>2 Model Predictive Control</b>	<b>11</b>
2.1 Vantaggi e svataggi del controllo MPC . . . . .	11
2.2 Algoritmo di controllo . . . . .	13
2.3 Architettura del controllo . . . . .	14
2.3.1 Modello di predizione . . . . .	14
2.3.2 Funzione obiettivo . . . . .	15
2.3.3 Calcolo della legge di controllo . . . . .	16
2.4 Formulazione del problema MPC . . . . .	17
2.4.1 Caso non vincolato . . . . .	20
2.4.2 Vincoli in ingresso e uscita . . . . .	21
2.4.3 Vincoli sullo stato . . . . .	23
2.5 MPC Toolbox di Matlab . . . . .	24
2.5.1 Modello di Predizione . . . . .	24
2.5.2 Ottimizzazione . . . . .	25
2.5.3 Object Reference . . . . .	26
<b>3 Programmazione Quadratica</b>	<b>29</b>
3.1 Active Set . . . . .	30

<b>4</b>	<b>Applicazione MPC</b>	<b>35</b>
4.1	Modello . . . . .	35
4.1.1	Risposta al gradino . . . . .	36
4.2	Controllo MPC con il ToolBox di MATLAB . . . . .	37
4.3	Controllo MPC . . . . .	39
4.3.1	Caso non vincolato, variazione dei pesi d'uscita . . . . .	41
4.3.2	Caso vincolato, limite in ingresso . . . . .	42
4.3.3	Caso vincolato, limiti in posizione . . . . .	44
4.3.4	Caso vincolato, limiti in posizione e variazione dell'orizzonte di predizione . . . . .	45
4.3.5	Caso vincolato, limiti in posizione e look-ahead . . . . .	46
4.3.6	Caso vincolato, limiti in posizione, look-ahead e vincolo in soglia di percezione . . . . .	48
<b>5</b>	<b>Simulazioni e risultati</b>	<b>51</b>
5.1	Singolo asse disaccoppiato . . . . .	51
5.2	Due assi accoppiati e tilt-coordination . . . . .	54
<b>6</b>	<b>Conclusioni</b>	<b>65</b>

# Introduzione

Simulare significa prendere un modello della realtà che consenta di valutare e prevedere lo svolgersi di una serie di eventi possibili causati dall'imposizione di alcune condizioni da parte di un utente. Un simulatore di guida, ad esempio, consente di prevedere il comportamento del veicolo conoscendo le sue caratteristiche e i comandi del pilota.

Le simulazioni, grazie alla grandi possibilità di calcolo offerte dall'informatica, sono uno strumento sperimentale di analisi molto potente, utilizzato sia per ambiti scientifici che tecnologici. La simulazione infatti, non è altro che la trasposizione logico-matematica della realtà; tale modello concettuale può essere definito come l'insieme dei processi che hanno luogo nel sistema valutato e il cui insieme permette di comprendere le logiche di funzionamento del sistema stesso. Molto diffuso quindi può essere il suo uso nell'analisi di sistemi dinamici, sia naturali che artificiali, ma anche economici.

Più specificatamente, i simulatori di guida automobilistica, forniscono l'opportunità di riprodurre le caratteristiche dinamiche del veicolo reale in un contesto virtuale. Essi riproducono condizioni e fattori esterni all'abitacolo con il quale il veicolo interagisce; inoltre paesaggi ed eventi sono virtualizzati in modo realistico tale da far sentire il pilota completamente immerso nella sua seduta di guida.

Negli ultimi anni i simulatori si stanno evolvendo con notevole successo grazie alla loro utilità e molti studi vengono fatti per migliorare la percezione della realtà; possono fornire un'esperienza costruttiva ai piloti novizi e hanno la capacità di sottoporre in situazioni complicate anche quelli più competenti. Ad esempio, per imparare la guida di autotrasporti offrono l'opportunità di iniziare la pratica senza dover essere a bordo di un veicolo così ingombrante e costoso. Dall'altra parte, per i piloti più esperti, la simulazione porta ad analizzare le proprie capacità, individuando le parti in cui la guida risulta insufficiente e consigliando i passi necessari per migliorarla.

Per le compagnie automobilistiche, inoltre, la simulazione offre l'opportunità di far imparare ai propri piloti e al suo staff i veicoli a disposizione, provare eventuali setup diversi e comunque imparare i circuiti che andranno a correre nelle gare ufficiali. Questo perchè solo con la pratica e un'analisi del tipo di guida si può arrivare a migliorare i propri difetti e ad inseguire la traiettoria che ci porterà ad avere il minor tempo sul giro. Questo per i team automobilistici significa *tempo* e *denaro*, sia per la dislocazione dei circuiti per il mondo, sia per la manutenzione dei veicoli ed inoltre, cosa più importante, salvaguardare la sicurezza del proprio personale.

Ovviamente perchè un simulatore possa far provare al pilota le sensazioni che avrebbe se stesse guidando nella realtà, percezioni visive e posturali che la piattaforma virtuale dà all'utente devono essere coerenti tra di loro, sia per quanto riguarda la fase, sia l'intensità degli stimoli forniti. Per ovvie ragioni, la riproduzione fedele del sistema reale non sarà possibile. Continui studi vogliono portare queste percezioni almeno ad esserlo il più possibile. Ad esempio l'accelerazione di una macchina da corsa, avendo in gioco forze troppo grandi, non verrà mai riprodotta fedelmente a causa delle soglie meccaniche degli attuatori e soprattutto per i limiti spaziali che ha la piattaforma. Per questo, studi sempre maggiori sono fatti per capire il meccanismo che determina l'equilibrio nel corpo umano, in modo da poter *ingannare* il pilota e fargli percepire una sensazione di moto che più si avvicina a quella reale, nonostante il simulatore stia riproducendo accelerazioni di tutt'altra intensità.

Da studi fisiologici fatti a tal proposito, come descritto dal Dott. G. Chetta in [12] si è appreso che l'apparato vestibolare, situato in profondità nell'osso temporale dietro l'orecchio interno, fornisce al cervello informazioni sulla posizione della testa e del corpo nello spazio. Oltre all'orecchio però, anche l'apparato visivo trasmette all'encefalo una percezione diretta dell'ambiente esterno. Affinchè le informazioni prodotte dal sistema vestibolare possano essere interpretate dal sistema posturale, devono essere costantemente comparate e integrate con quelle derivanti dagli altri recettori periferici, tra i quali quelli visivi. Informazioni contrastanti dall'apparato vestibolare e quello visivo possono essere causa di disagi (è il caso della *cinetosi*, il "mal di mare", che può nascere quando si crea un contrasto tra l'informazione visiva che osserva un punto fisso e le oscillazioni della barca rilevate dal vestibolo).

L'apparato vestibolare è predisposto per rispondere al meglio ad accelerazioni rapide e di breve durata mentre si lascia facilmente ingannare da accelerazioni lunghe o inconsuete (ciò spiega i capogiri che si hanno quando si ruota più volte su se stessi e ci si ferma di colpo). Quindi per ottenere ad esempio una accelerazione costante, si vuole andare ad *ingannare* il sistema dell'equilibrio umano, mantenendo l'apparato visivo fermo e ruotando la sedia del pilota rispetto l'orizzonte, in modo tale che il pilota senta un'accelerazione gravitazionale coerente a quella che vorremmo fargli percepire senza dover muovere longitudinalmente l'abitacolo. Ovviamente ci dovrà essere una qualche strategia che, in modo del tutto automatico, mandi i segnali agli attuatori affinché la piattaforma si muova nel modo appena descritto.

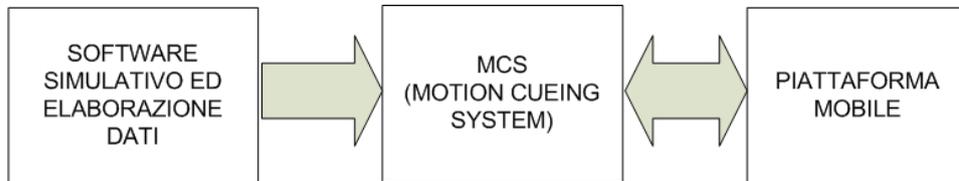


Figura 1: Apparato simulativo

Il sistema rappresentato in figura 1 vuole descrivere le componenti hardware e software di un simulatore: è strutturato da un programma simulativo che si occuperà di virtualizzare l'automobile e l'ambiente ed elaborare i dati provenienti dai comandi dei pilota e quindi in grado di generare i segnali del moto da fargli percepire. Inoltre avrà un secondo sistema, chiamato anche *motion cueing*, capace di prendere le accelerazioni sviluppate dall'elaboratore e trasformarle in comandi di attuazione della piattaforma; fondamentale quindi sarà lo scambio di informazioni con la struttura mobile per una corretta attuazione dei segnali. Il sistema di *motion cueing* o semplicemente *MCS* si occuperà di far muovere il simulatore nel miglior modo, ovvero dando al pilota una sensazione di moto che possa essere il più vicino possibile a quella reale soddisfacendo i limiti imposti per la piattaforma.

Operando sotto la soglia fisiologica di percezione dell'apparato vestibolare e sapendo che le accelerazioni sentite dall'operatore sull'abitacolo sono effetto della somma vettoriale tra l'accelerazione assoluta e di quella relativa alla gravità, si vuole presentare in questo lavoro un algoritmo di *motion cueing*, il quale in modo del tutto automatico gestisca le accelerazioni della macchina attraverso il *tilt-coordination*. Questa strategia vuole far percepire

al pilota accelerazioni a bassa frequenza o costanti attraverso l'inclinazione dell'abitacolo di un opportuno angolo. In questo modo i limiti longitudinali saranno rispettati e qualora il simulatore non sia nella posizione d'origine, si potrà agire sulla movimentazione lineare per portarla al tal punto non superando però l'accelerazione di soglia fisiologica di  $0.05 m/s^2$ . Per quanto riguarda le accelerazioni lineari ad alta frequenza, queste dovranno essere riprodotte dalla movimentazione longitudinale, stando attenti a non sbattere contro i margini della piattaforma con la conseguente generazione di una falsa percezione di moto al pilota. Altro compito non meno importante del *MCS* che però non verrà affrontato in questo progetto, sarà quello di sincronizzare la grafica con il feedback vestibolare, fondamentale per non creare le sensazioni di malessere descritte.

# Capitolo 1

## Storia dei Simulatori

### 1.1 Strutture esapodali

La piattaforma più utilizzata fin dall'inizio per i primi simulatori di volo è quella di Gough-Stewart. Studi su tale struttura risalgono ai primi anni del 1800 quando il matematico *Cauchy* studiò il moto del così detto *ottaedro articolato*. Ci volle più di un secolo, prima con *Gough* e poi con *Stewart*, affinché le grandi potenzialità di questa struttura potessero essere chiare alla comunità scientifica. La piattaforma di Gough-Stewart (o anche chiamata esapodo per la sua struttura a sei gambe) è un meccanismo parallelo a 6 gradi di libertà (figura 1.1).

Nella sua configurazione elementare è composto da una base mobile collegata al telaio fisso tramite un sistema di attuazione parallelo, in cui gli attuatori lineari, o gambe della piattaforma, sono disposti in modo da avere a due a due un punto in comune alternativamente con la base fissa e quella mobile mediante giunti sferici. I sei arti sono disposti simmetricamente per formare tre quadrilateri con i due lati opposti fissi sulla base mobile e sulla piattaforma fissa, ed i restati due costituiti da due coppie prismatiche. Il successo dell'esapodo, o in generale della meccanica parallela, è dovuto al fatto che a differenza della meccanica seriale fornisce una maggiore capacità di carico di lavoro ripartita tra tutti e 6 gli attuatori, un'ottima accuratezza nel posizionamento e consente di ottenere elevate accelerazioni per la base mobile.

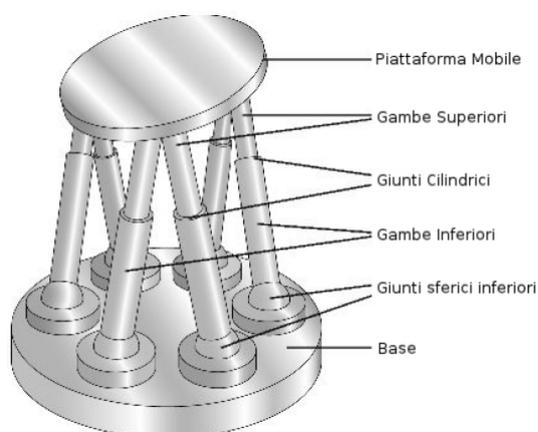


Figura 1.1: Struttura esapodale di Gough-Stewart

### 1.1.1 Primi simulatori

Il primo simulatore di guida automobilistica fu costruito dalla Volkswagen nei primi anni '70, con un sistema a 3 gradi di libertà. Il moto era dato facendo muovere la cabina di pilotaggio lungo i tre assi rotazionali yaw, roll e pitch. Un unico flat screen era montato davanti al pilota seduto nella postazione.

Per vedere, però, il primo simulatore costruito utilizzando un esapodo idraulico bisogna aspettare il 1985 quando Daimler-Benz riuscì a costruire una struttura che al tempo fu capace di sviluppare la più realistica sensazione di moto.

La cabina di un'auto o un camion erano montate dentro la struttura la quale era fornita di 6 proiettori CRT capaci di dare una visione di  $180^\circ$  al pilota. Nel 1993 il simulatore fu aggiornato in "Advanced Driving Simulator". La differenza principale con il precedente fu la maggiore estensione del moto lungo l'asse  $y$ ; attraverso un cilindro idraulico si riusciva ad avere un'escursione laterale di  $5.6\text{ m}$ .

Dagli inizi degli anni '90 in poi, anche altre case automobilistiche come Toyota, Mazda, BMW, Ford cominciarono ad utilizzare esapodi per i loro simulatori in quanto considerata la soluzione migliore. Inoltre, RENAULT con il simulatore ULTIMATE nel 2004, per incrementare ulteriormente lo spazio a disposizione per le accelerazioni longitudinali, fissò l'esapodo ad un sistema di rotaie capace di muovere l'abitacolo lungo gli assi  $xy$ . In questo modo, la piattaforma posizionata in una stanza di  $250\text{ m}^2$  è capace

di accelerare fino a  $\pm 7 \text{ m/s}^2$  lungo le rotaie con uno spostamento massimo di  $7 \text{ m}$  e fino a  $\pm 300 \text{ }^\circ/\text{s}$  per un massimo di  $30 \text{ }^\circ$  [2].

Nonostante la dimensione enorme della piattaforma, anche con questo tipo di simulatore non è stato possibile riprodurre con la stessa intensità le accelerazioni, a causa dei vincoli sugli attuatori, che il pilota avrebbe dovuto percepire. Per questo ricerche sempre più avanzate vengono effettuate sul tipo di struttura da utilizzare e per lo sviluppo di nuovi algoritmi di *motion cueing*.

### 1.1.2 Motion Cueing System

Per far muovere in modo intelligente un simulatore si utilizza un sistema chiamato *motion cueing* o *MCS*, il quale provvede a trasformare i riferimenti di accelerazione  $r(t)$  che gli arrivano in ingresso, in possibili segnali da dare agli attuatori. Gli algoritmi di *MCS* hanno dunque degli obiettivi contrastanti da soddisfare al meglio: far percepire al guidatore una sensazione di moto che si avvicini il più possibile alla realtà e mantenere la piattaforma dentro i suoi limiti.

Il primo algoritmo di questo tipo fu realizzato nei primi anni '70 da Schmidt e Conrad [9] i quali svilupparono la strategia “classica” per *motion cueing* che andava a calcolare le accelerazioni del simulatore lungo tutti i suoi gradi di libertà. Il classico algoritmo di *MCS*, formato da filtri in frequenza e descritto in [3], è strutturato nel seguente modo:

- si rimuovono le basse frequenze dalle accelerazioni longitudinali attraverso un filtro passa-alto, per poi integrare due volte il risultato in modo da avere in uscita il comando di posizione della piattaforma;
- si estraggono dalle accelerazioni le basse frequenze attraverso un filtro passa-basso e da queste si calcola l'angolo di *tilt coordination* che verrà aggiunto al comando di uscita per le posizioni angolari;
- si riporta la piattaforma alla sua posizione neutrale filtrando attraverso un passa-alto le risultanti dei comandi di posizionamento.

Quest'ultimo filtraggio, spesso chiamato anche *motion washout* (Reid and Nahon, 1985) [17], è necessario per evitare che si vada a saturare gli attuatori i quali poi porterebbero a far percepire al pilota false dinamiche del veicolo. Un altro tipo di filtraggio chiamato *anti-backlash filter* può essere

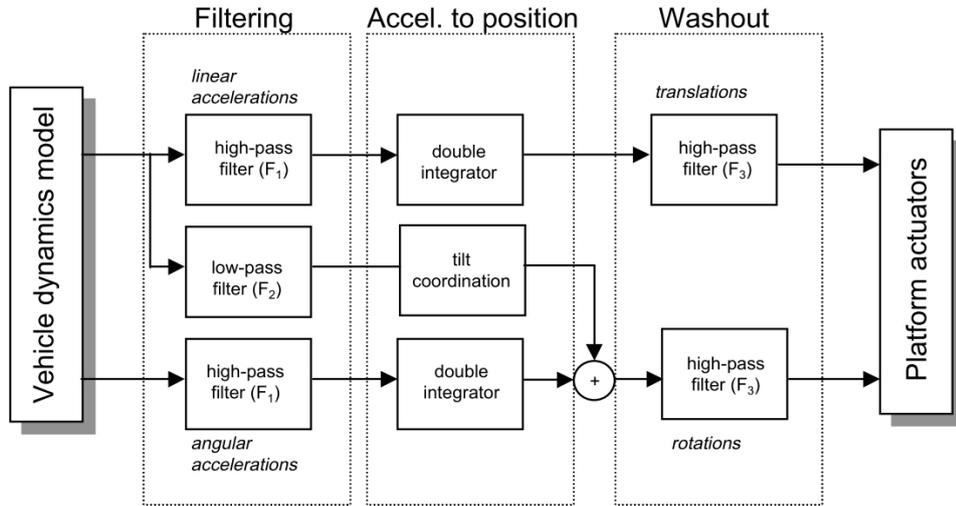


Figura 1.2: Struttura Motion Cueing classico

aggiunto per ridurre alcune risposte non ottimali in seguito al filtraggio passa-alto (Reymond, 2000) [3].

Utilizzando questo tipo di algoritmo, filtrando le basse frequenze del nostro riferimento le accelerazioni risultanti saranno estremamente ridotte. Questo perchè il settaggio dei parametri del filtro (come guadagno e frequenza di taglio) saranno impostati di sessione in sessione, in funzione del circuito e della guida del pilota, dato che il *cueing* dovrà fare in modo che non si vada mai a saturare gli attuatori. Qualora, ad esempio, succedesse che l'abitacolo fosse ai margini della posizione possibile, l'algoritmo ha due soluzioni: o restare con la cabina immobile riproducendo solo le accelerazioni angolari, aspettando quindi un'accelerazione in fase per poter tornar indietro alla sua posizione d'origine, oppure andar in controfase facendo percepire al pilota una accelerazione in contrasto a quella ricevuta dal sistema visivo, provocandogli quindi una sensazione di malessere.

In entrambi i casi, le soluzioni adottate non rispettano l'obiettivo dell'algoritmo di moto e quindi si cercherà di tarare il filtro tenendo conto del caso peggiore, facendo in modo che la massima accelerazione in ingresso possa restare entro le soglie degli attuatori della piattaforma.

Come logica conseguenza, in una sessione di guida dove prevalgono nettamente le accelerazioni "normali" verrà utilizzata solo una piccola parte della possibilità di moto del simulatore.

Per ovviare a questo tipo di problema, Parrish e Dieudonne (1975) [18]

proposero un metodo *adaptive* poi sviluppato ulteriormente negli anni successivi. Basandosi sul classico *MCS*, invece di usare la strategia solo nel dominio delle frequenze, si considera anche il dominio del tempo: il tipo di filtraggio presente nel precedente algoritmo rimane, ma in questo caso i parametri non sono costati e vengono cambiati ad ogni passo.

Ad ogni istante di campionamento, il guadagno e la frequenza di taglio del classico filtro *MCS* sono derivati dalla minimizzazione di una funzione costo quadratica

$$V_k = (r_k - a_k)^2 + w_1 \cdot v_k^2 + w_2 \cdot p_k^2. \quad (1.1)$$

Il primo termine di  $V_k$  è l'errore quadratico tra l'accelerazione del veicolo reale  $r_k$  e quella della piattaforma  $a_k$ . I restanti due termini sono relativi alla velocità  $v_k$  e alla posizione  $p_k$  della piattaforma.

Rispetto alla versione precedente, questa *adaptive strategy* risulta sfruttare maggiormente lo spazio di lavoro in condizioni di guida “regolare”, ossia con accelerazioni inferiori rispetto i valori con cui vengono tarati i filtri. I pesi  $w_1$  e  $w_2$  definiscono un “trade-off” tra la sensazione di moto che si vuole sviluppare e i limiti imposti dagli attuatori: un aumento di  $w_1$  e  $w_2$  penalizzerà spostamenti e velocità elevate mentre diminuendoli si andrà a favorire la minimizzazione dell'errore quadratico tra le accelerazioni. In ogni caso la taratura andrà effettuata off-line considerando la specifica guida del pilota o del tipo di circuito su cui si andrà ad effettuare la sessione simulativa.

Nel 1982 Sivan e Ish Shalom [19] svilupparono l'algoritmo *ottimo* il quale, in combinazione con il filtraggio lineare passa-basso e passa-alto come per il classico *MCS*, calcola i comandi da dare alla piattaforma minimizzando una funzione costo globale come quella espressa in (1.2):

$$V[u(t)] = \int_0^{\infty} [(\hat{r}(t) - \hat{a}(t))^2 + w_1 \cdot v^2(t) + w_2 \cdot p^2(t)] dt \quad (1.2)$$

soggetto a

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1.3)$$

con  $x = [p \ v \ a]^T$

dove  $(A, B)$  sono le matrici del modello di percezione del moto,  $\hat{a}$  rappresenta l'accelerazione percepita dal pilota all'interno dell'abitacolo e  $u(t)$  l'ingresso di controllo del sistema. Per calcolare il minimo della funzione

(1.2) occorre conoscere il riferimento lungo un'intera sessione di guida per il tempo  $[0, \infty]$ .

In questa ultima strategia si ha oltre al modello del sistema considerato, anche il primo utilizzo di una formulazione del modello di percezione del moto proposta negli anni prima da Zacharias (1978); questo perchè dovendo riprodurre la sensazione di moto in una struttura con dei vincoli ben noti, si cerca di andare ad “ingannare” la percezione del pilota in modo da inseguire la traiettoria non facendo uso solo di accelerazioni longitudinali ma anche attraverso l'utilizzo del *tilt-coordination*.

## 1.2 Desdemona

Parallelamente allo sviluppo di simulatori attraverso la meccanica di Gough-Stewart, sono stati fatti svariati studi per staccarsi dalle classiche strutture con lo scopo di creare una nuova piattaforma più performante, con nuove capacità. Dunque nei primi anni 2000 nasce il progetto *Desdemona*.

Desdemona è un simulatore a base mobile situato presso il TNO (Soesterberg, Olanda), costruito con l'intenzione di dare un voluto disorientamento spaziale, ovvero facendo in modo che il pilota perda il senso “del sopra e del sotto”, fondamentale per sessioni simulate sia nel campo aeronautico sia nel campo automobilistico.

Progettato in cooperazione con l'AMST (Ranhofen, Austria) il simulatore si muove lungo i 6 gradi di libertà. La cabina di pilotaggio è montata su una sospensione cardanica, ovvero un insieme di tre anelli, ciascun anello con un paio di cuscinetti, capace di far ruotare la piattaforma attorno a qualsiasi asse rotazionale ( $\varphi_c, \psi_c, \theta_c$ , 3 gradi di libertà,  $> 2\pi$ ). Questa sospensione può essere mossa lungo l'asse verticale ( $H$ , grado di libertà,  $\pm 1m$ ) e orizzontalmente lungo una rotaia ( $R$ , 1 grado di libertà,  $\pm 4m$ ). Tutta la struttura, inoltre, può essere ruotata attorno il proprio asse centrale ( $\psi_s$ ) in modo da facilitare il moto centrifugo (1 grado di libertà,  $< 3g$ ).

Si riportano, quindi, le specifiche della nuova piattaforma in tabella 1.1.

Ovviamente per usufruire della performance di questa particolare struttura (raggio 4m) doveva essere realizzato un nuovo *motion cueing*. L'utilizzo di un classico *MCS* o una sua versione estesa non avrebbe portato all'utilizzo dell'intero spazio a disposizione; questo perchè quando viene usata una

Tabella 1.1: Desdemona

	$\psi_s$	$R$	$H$	$\varphi_c$	$\psi_c$	$\theta_c$
posizione	illimitata	$\pm 4m$	$\pm 1m$	illimitata	illimitata	illimitata
velocità	$\pm 151^\circ/s$	$\pm 3.2m/s$	$\pm 2m/s$	$\pm 180^\circ/s$	$\pm 180^\circ/s$	$\pm 180^\circ/s$
accelerazione	$\pm 45^\circ/s^2$	$\pm 4.9m/s^2$	$\pm 4.9m/s^2$	$\pm 90^\circ/s^2$	$\pm 90^\circ/s^2$	$\pm 90^\circ/s^2$

struttura esapodale la cinematica risultante è cartesiana. Per Desdemona, invece, avendone una polare, un apposito *washout filter* è stato progettato chiamato *Spherical Washout Filter* (Wentink, 1985) [11]; caratteristica principale di quest'ultimo è la capacità di sfruttare l'enorme spazio circolare a disposizione di Desdemona, il quale anche in presenza di forti accelerazioni riesce a riprodurle in modo migliore rispetto un esapodo utilizzando sia la *tilt coordination* sia l'accelerazione centrifuga.

### 1.3 Piattaforma Vi-Grade

Vi-Grade è un'azienda fondata nel 2005, che offre e sviluppa software per simulazioni per tutte le compagnie che vogliono aggiungere un collegamento agli studi ingegneristici teorici e la prova nella realtà di quanto si è studiato.

VI-DriveSim è una nuova linea di prodotti creata da VI-Grade la quale cerca di dare alle compagnie automobilistiche un mezzo di integrazione tra sviluppo di nuovi prototipi o sistemi di controllo e test su una piattaforma che rispecchia l'ambiente reale.

VI-DriveSim si differenzia in due possibili configurazioni: statica e dinamica. Quest'ultima è composta dai seguenti componenti hardware e software:

- piattaforma di moto rivoluzionaria;
- modelli e dati dei veicoli allo stato dell'arte basati sul software già esistente VI-CarRealTime;
- algoritmo di motion cueing;
- provvista di grafica e suoni;
- attuatori che riescono a dare un feedback al controllo pilota.

La piattaforma mobile è basata su una nuova architettura a 6 DOF (*Degree of freedom*) la quale lavora con delle escursioni relativamente ridotte rispetto

ai sistemi esapodali descritti precedentemente. In figura 1.3 è rappresentata la piattaforma mobile di VI-DriveSim. In tabella 1.2, inoltre, i valori delle

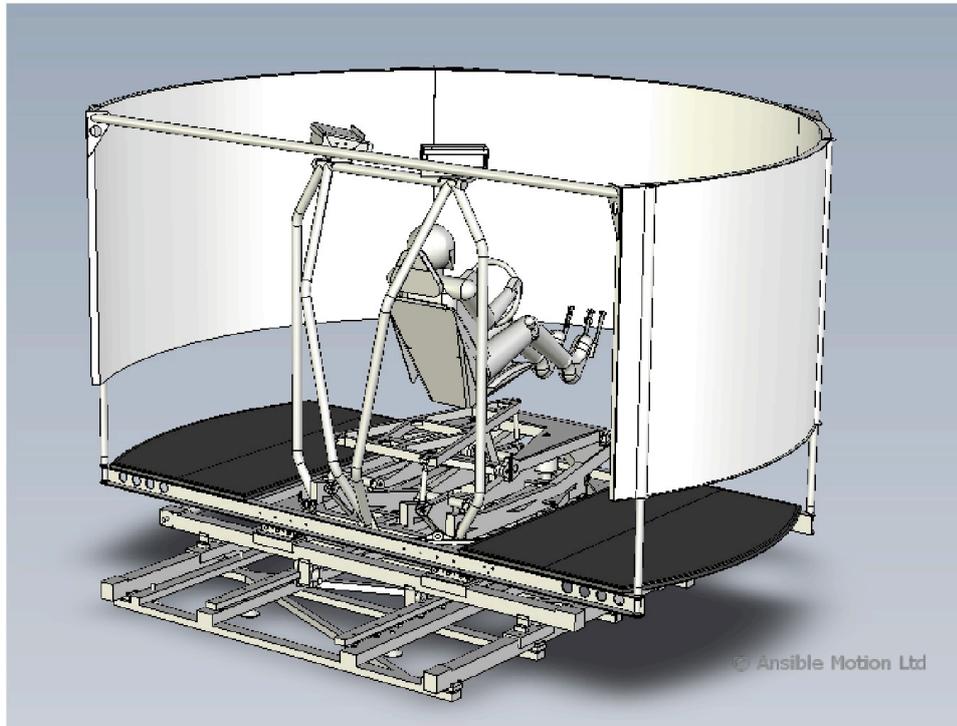


Figura 1.3: Piattaforma VI-Grade

performance del simulatore.

I movimenti longitudinale  $x$  e  $y$  insieme alla rotazione dello yaw sono disaccoppiati mentre roll, pitch e movimenti lungo l'asse  $z$  sono accoppiati tra di loro. VI-DriveSim vuole dunque realizzare un nuovo tipo di approccio per lo sviluppo di una piattaforma di guida, la quale oltre allo scopo principale di test per nuovi sistemi elettronici e training di guida, visto le dimensioni ridotte potrà essere utilizzata per studiare la percezione multisensoriale umana, oppure presso scuole di guida per un primo e sicuro approccio alla guida sulle strade o semplicemente per divertimento.

A causa del poco spazio a disposizione per gli spostamenti, un adeguato *motion cueing* sarà fondamentale per riuscir a riprodurre al meglio le accelerazioni; guardando le escursioni massime elencate nella tabella (1.2), per accelerazioni ad esempio, costanti e a lunga durata, l'utilizzo esclusivo dell'asse  $x$  per riprodurre tali valori porterà la cabina in pochi attimi

Tabella 1.2: Performance Piattaforma VI-Grade

	Escursione	Velocità	Accelerazione
X	1m	1.3m/s	3.3m/s <sup>2</sup>
Y	1m	1.3m/s	3.6m/s <sup>2</sup>
Z	0.3m	0.9m/s	4.9m/s <sup>2</sup>
Roll	40 deg	112 deg /s	600 deg /s <sup>2</sup>
Pitch	24 deg	61 deg /s	600 deg /s <sup>2</sup>
Yaw	40 deg	61 deg /s	240 deg /s <sup>2</sup>

verso i suoi limiti di funzionamento. Per questo motivo si vuole sviluppare un algoritmo che, guardando in avanti di qualche secondo (conoscendo dunque la planimetria della pista), sarà capace di prevedere che non riuscirà a riprodurre fedelmente l'accelerazione utilizzando solamente un asse ( $x$ ) ma potrà, *ingannando la percezione umana*, anche andare ad agire sul pitch della piattaforma per la riproduzione di segnali a bassa frequenza lasciando che i movimenti di breve durata e improvvisi vengano riprodotti attraverso spostamenti longitudinali.



## Capitolo 2

# Model Predictive Control

Una delle strategie che ha avuto origine intorno alla fine degli anni '70 ed è tuttora applicata nei processi industriali è il controllo predittivo basato sul modello *MPC* (Model Predictive Control).

E' un metodo molto generale che può essere usato sia per problemi di controllo a singola variabile d'ingresso e d'uscita (SISO) ma anche ben adatto ad essere applicato nel caso di problemi di controllo multivariabile (MIMO) con un elevato grado di interazione fra le variabili d'ingresso e quelle di uscita. E' una strategia che tratta vincoli di disuguaglianza o uguaglianza sulle variabili d'ingresso-uscita del sistema in modo del tutto naturale.

Facendo parte della famiglia dei controlli predittivi, l'utilizzo di un modello dinamico del processo per la predizione dell'evoluzione del sistema su un opportuno orizzonte temporale sarà l'aspetto caratterizzante della strategia. Essendo un controllo molto generico, diverse tecniche di implementazione sono state realizzate fino ai giorni d'oggi; per ognuno di esse bisogna necessariamente conoscere il modello del sistema, ovvero disporre delle equazioni e dei parametri che, noti gli ingressi, permetteranno di determinare gli andamenti futuri delle uscite.

Gli algoritmi MPC determinano il segnale di controllo da fornire in ingresso al sistema *minimizzando una funzione obiettivo*. Il modello utilizzato per rappresentare il processo e la funzione costo da minimizzare stabiliscono le differenze tra i diversi metodi di implementazione.

### 2.1 Vantaggi e svataggi del controllo MPC

Il successo del controllo dagli anni '70 ad oggi è dovuto ai seguenti aspetti:

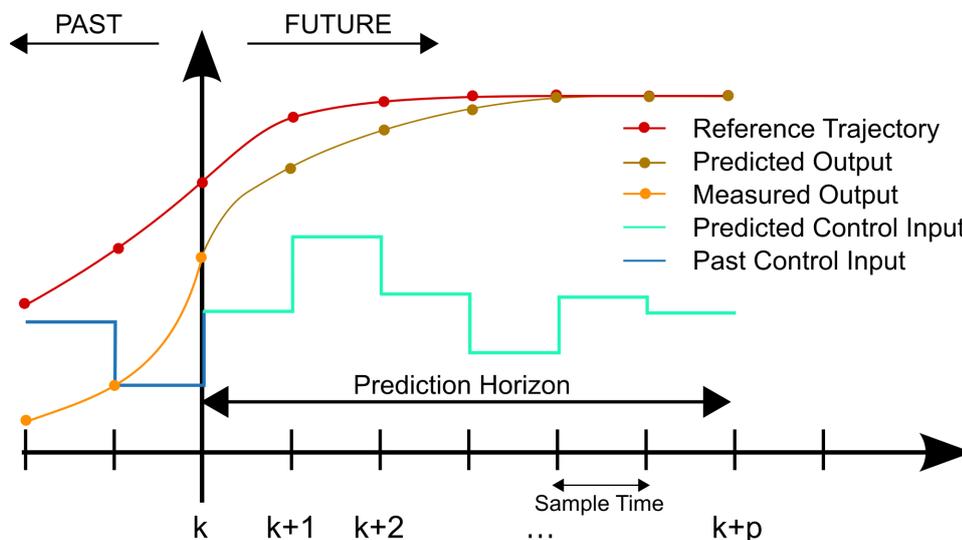


Figura 2.1: Strategia MPC

- MPC è una strategia generale utilizzata senza troppe differenze sia per sistemi SISO sia per sistemi MIMO, con la possibilità di inserire o meno limiti sulle variabili che caratterizzano il sistema;
- è in grado di effettuare il controllo dei sistemi più difficili, caratterizzati da elevati ritardi o risposte a fase non minima;
- poichè gli ingressi di controllo del sistema vengono calcolati con procedure di minimizzazione, è naturale poter pensare di inserire strategie per l'ottimizzazione delle prestazioni dell'impianto;
- essendo la strategia di controllo calcolata ad ogni passo di campionamento, questa può essere aggiornata in modo da tener conto delle variazioni delle condizioni nella quale il processo lavora, oppure cambiare vincoli o obiettivi nel corso della sessione di controllo.

Ovviamente, nonostante questi vantaggi, presenta delle difficoltà che possono renderne difficile l'impiego:

- MPC ha un approccio completamente differente dalla tipologie di controllo più convenzionali e pertanto poco familiare al personale tecnico;
- i calcoli da eseguire, specie con molte variabile d'ingresso-uscita o con vincoli molto stretti, sono piuttosto complessi e la risoluzione dell'ot-

timizzazione ad ogni istante di campionamento richiede un elevato impegno alle risorse di calcolo;

- l'obbligo di dover utilizzare un modello dinamico comporta una fase iniziale di identificazione del processo che può portare a diverse settimane di lavoro;
- la non disponibilità di modelli completi del processo spesso comporta l'uso di modelli empirici, o modelli validati solo in riferimento alle condizioni operative considerate durante le relative prove di identificazione.

## 2.2 Algoritmo di controllo

La strategia che contraddistingue tutti gli algoritmi di controllo predittivo basato sul modello dinamico è la seguente:

1. si fissa un orizzonte temporale per la predizione che indichiamo con  $P$ ; utilizzando quindi il modello del processo, per ogni istante  $t$ , si determina quale sarà l'evoluzione dell'uscita del sistema lungo tale orizzonte:  $y(t+k|t)$   $k = 1 \dots P - 1$ . Il valore previsto per  $y$  dipende dai valori assunti nel passato dall'ingresso e dall'uscita e dai futuri valori dell'ingresso  $u(t+k|t)$  con  $k = 0 \dots P - 1$  che saranno inviati al sistema;
2. l'insieme dei futuri ingressi di controllo viene calcolato ottimizzando un determinato indice in modo da mantenere il processo il più vicino possibile ad una traiettoria di riferimento  $r(t+k)$ . Questo indice è di tipo *quadratico* ed è funzione dell'errore tra il valore previsto per l'uscita ed il riferimento e del segnale di controllo. Per un problema lineare senza vincoli, la soluzione può essere trovata in maniera esplicita; se invece i segnali e il processo sono soggetti a vincoli la soluzione viene raggiunta in maniera *iterativa*;
3. il segnale di controllo  $u(t|t)$  viene inviato al processo, il segnale di controllo seguente  $u(t+1|t)$  viene scartato perchè all'istante di campionamento successivo si avrà il valore dell'uscita  $y(t+1)$  aggiornata, dunque una nuova sequenza d'ingresso sarà calcolata in funzione delle

nuove informazioni ricevute e così facendo  $u(t+1|t+1)$  sarà diverso da  $u(t+1|t)$ . Questa tecnica è denominata *receding horizon*.

## 2.3 Architettura del controllo

Come detto nell'introduzione, esistono in commercio molti controllori MPC, in base al tipo di implementazione scelta. In generale un algoritmo di questo

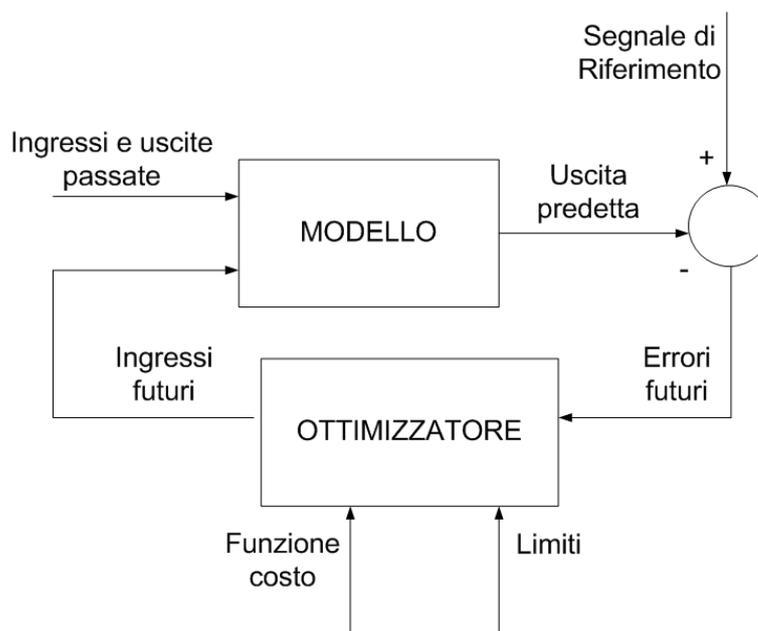


Figura 2.2: Schema Controllo MPC

tipo è costituito da tre elementi chiave:

- modello di predizione;
- funzione obiettivo;
- calcolo della legge di controllo.

### 2.3.1 Modello di predizione

E' la parte più importante del controllore che descrive la dinamica del processo e permette di calcolare la predizione dell'uscita  $y(t+k|t)$ ; contemporaneamente deve essere intuitivo per facilitare l'analisi su ciò che si sta

facendo. Esso è composto dal modello del processo ed un eventuale modello del disturbo presente.

Il processo può essere modellato attraverso diverse rappresentazioni:

- risposta al gradino;
- risposta all'impulso;
- spazio di stato.

Per la realizzazione del nostro controllore *MPC* si è scelta quest'ultima forma. Infatti, come descritto dal Maciejowsky [16], utilizzando il modello in spazio di stato sarà possibile utilizzare tutta la teoria dei sistemi lineari e del controllo. Inoltre al contrario dei modelli basati su risposta all'impulso o al gradino, sistemi multivariabili sono maggiormente utilizzabili se rappresentati attraverso questo tipo di modello.

Si andrà a vedere come attraverso questo tipo di formulazione il calcolo della legge di controllo verrà effettuata attraverso la risoluzione di un problema di *Programmazione Quadratica*, per il quale svariati algoritmi sono stati implementati in letteratura.

La modellizzazione di rumori o disturbi, verrà inserita direttamente nel modello del sistema ed inoltre, la possibilità di usare un'osservatore per la stima dello stato è una prerogativa che utilizzando il modello in spazio di stato ci è garantita.

L'osservatore di stato è un sistema dinamico che ha lo scopo di stimare l'evoluzione di un sistema. Nell'implementazione di leggi di controllo con feedback è necessario quando non si può misurare direttamente lo stato o quando l'errore nella misurazione è più grande di quello che si commette andando a stimarlo. Affinchè la stima sia possibile occorre che il sistema sia osservabile e in tal modo a partire dalla misura delle uscite si ricostruisce lo stato. La mancanza dell'osservatore nella strategia MPC farebbe in modo che il controllo lavorasse come se fosse in catena aperta.

### 2.3.2 Funzione obiettivo

Per trovare gli ingressi di controllo del nostro sistema andremo a minimizzare una certa funzione costo la quale caratterizzerà il tipo di risposta. Tale minimizzazione avrà sicuramente lo scopo di far seguire all'uscita futura  $y$ , nell'orizzonte di predizione considerato, un determinato segnale di riferimento  $r$ , tenendo sotto controllo l'ingresso a disposizione.

Una possibile funzione costo descritta dal Maciejowsky che tiene conto di tutto ciò e che noi andremo ad usare è la seguente :

$$J = \sum_{j=0}^P \delta(j) [\hat{y}(t+j|t) - r(t+j)]^2 + \sum_{j=1}^N \lambda(j) [\Delta u(t+j-1)]^2 \quad (2.1)$$

con

- $P$  orizzonte di predizione;
- $N$  orizzonte di controllo;
- $\delta(j)$ ,  $\lambda(j)$  pesi rispettivamente sull'errore dell'uscita e sulla variazione dell'ingresso. Danno la possibilità di variare le performance della risposta del sistema in funzione delle proprie esigenze.

Da notare che tale indice non è l'unico utilizzabile ma svariate rappresentazione sono possibili, in base delle esigenze del nostro sistema. Ovviamente, l'utilizzo di funzioni obiettivo più complesse ci aiuterà ad avere maggior controllo sull'andamento dei nostri ingressi-uscite ma la minimizzazione, specie con vincoli, diventerà sempre più onerosa in termini di risorse di calcolo.

### 2.3.3 Calcolo della legge di controllo

Per il calcolo della sequenza di controllo deve essere fatta la minimizzazione della funzione costo. Come detto precedentemente nel caso in cui si dovesse minimizzare un indice senza vincoli, l'ingresso di controllo potrà essere calcolato in maniera esplicita. Non sempre, e soprattutto non nel caso reale, sarà possibile lavorare senza limiti per ragioni costruttive o per motivi di sicurezza: l'ambiente su cui andremo ad operare sarà normalmente caratterizzato dall'intersezione di soglie e quindi viene naturale pensare di inserirli nella funzione costo da minimizzare.

In generale i vincoli posso riguardare le variabili d'ingresso e d'uscita oppure semplicemente le variazioni del segnale di controllo. Quindi:

$$u_{min} \leq u(t) \leq u_{max} \quad \forall t;$$

$$du_{min} \leq u(t) - u(t-1) \leq du_{max} \quad \forall t;$$

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

I valori futuri dell'uscita  $y(t + k|t)$  lungo l'orizzonte di predizione  $P$ , noto il modello del sistema, sono funzione dei valori assunti nel passato dall'ingresso e dall'uscita e dai futuri segnali di controllo. Se sono presenti anche dei limiti, la risoluzione dell'ingresso di controllo non è così immediata dato che andiamo a minimizzare almeno  $P + 1$  variabili indipendenti. Per ridurre questo grado di libertà, si ha la possibilità di aggiungere un'ulteriore funzionalità all'algoritmo, il *control horizon concept*; il controllore dopo un certo numero di passi  $N < P$  assume non ci siano variazioni del segnale d'ingresso:

$$\Delta u(t + j - 1) = 0 \quad j > N_u.$$

In questo modo i gradi di libertà (e il peso computazionale dell'algoritmo di risoluzione) diminuiscono.

## 2.4 Formulazione del problema MPC

Si procede ora all'implementazione del controllo MPC per un modello espresso in spazio di stato, come descritto dal Wang [10], partendo da un sistema dinamico SISO descritto da:

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

$$y(k) = C_m x_m(k) \quad (2.3)$$

dove  $u$  è la variabile di ingresso,  $y$  è l'uscita del processo e  $x_m$  è la variabile di stato assunta di dimensione  $n_1$ . Si prende ora la differenza

$$x_m(k + 1) - x_m(k) = A_m(x_m(k) - x_m(k - 1)) + B_m(u(k) - u(k - 1)) \quad (2.4)$$

e definendo la differenza dello stato

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

e la variazione d'ingresso

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

la formulazione (2.4) prima descritta risulta

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

Si nota che in questo caso il modello non ha più in ingresso  $u(k)$  ma  $\Delta u(k)$ ; il prossimo passo sarà quindi quello di collegare  $\Delta x_m(k)$  all'uscita  $y(k)$ .

Per fare questo si definisce la nuova variabile di stato

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

Inoltre considerando

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

risulta:

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \overbrace{\begin{bmatrix} A_m & 0_m^T \\ C_m A_m & 1 \end{bmatrix}}^A \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \overbrace{\begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}}^B \Delta u(k) \quad (2.5)$$

$$y(k) = \underbrace{\begin{bmatrix} 0_m & 1 \end{bmatrix}}_C \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$$

con  $0_m = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}$  un vettore di  $n_1$  colonne.

Assumendo che all'istante di campionamento  $k_i$ , la variabile di stato  $x(k_i)$  sia misurabile, la futura traiettoria di controllo è data da  $\Delta u(k_i)$ ,  $\Delta u(k_i+1)$  ...  $\Delta u(k_i + N_c - 1)$  dove  $N_c$  è chiamato orizzonte di controllo, ovvero il numero di ingressi usati per calcolare la futura traiettoria del sistema.

Basandosi sul modello a spazio di stato aumentato descritto da  $(A, B, C)$ , i futuri stati sono calcolati sequenzialmente nel seguente modo:

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

mentre le future variabili d'uscita sono definite da:

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

o scrivendole in forma vettoriale

$$Y = Fx(k_i) + \Phi\Delta u \quad (2.6)$$

dove

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \dots \\ CA^{N_p} \end{bmatrix}, \Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix}. \quad (2.7)$$

Una volta definito il sistema in spazio di stato, come in ogni formulazione del problema MPC, sarà da definire una funzione obiettivo nella quale saranno presenti le uscite, il riferimento e gli ingressi adottati. L'obiettivo principale è quello di inseguire una traiettoria  $r(k_i)$  il più vicino possibile. Si vuole quindi trovare il migliore parametro di controllo  $\Delta u$  per il quale il nostro indice (2.8) sia minimizzato:

$$J = (R_s - Y)^T Q (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (2.8)$$

dove si assume che

$$R_s^T = [1 \ 1 \ \dots \ 1] \cdot r(k_i) \quad (2.9)$$

sia il vettore che contiene il riferimento (in questo caso costante per ogni passo di campionamento) e

$$\bar{R} = r_i I_{N_c}$$

$$Q = r_o I_{N_p}$$

siano le matrici diagonali con i pesi rispettivamente per gli ingressi e le uscite. Ovviamente il 1° elemento della matrice  $Q$  sarà riferito al 1° passo di predizione mentre il 1° elemento della matrice  $\bar{R}$  al 1° passo dell'orizzonte di controllo e così via. La variazione di questi parametri è utilizzata per ottenere performance differenti nel controllo in catena chiusa. Ad esempio con  $\bar{R} = 0$  la funzione costo è interpretata come la situazione in cui non si vuole dare nessuna importanza a quanto grande diverrà  $\Delta U$ , ma solo minimizzare l'errore quadratico

$$(R_s - Y)^T Q (R_s - Y). \quad (2.10)$$

Attraverso questo tipo di implementazione oltre che a variare il peso per ogni sessione di controllo, sarà possibile variare sia i pesi da un passo di predizione all'altro lungo l'orizzonte di previsione ma anche variare i pesi da un istante di campionamento all'altro, nel caso, ad esempio, in cui un'uscita si sta avvicinando ai suoi limiti e si vorrebbe che stesse nei pressi della sua posizione neutrale (riferimento a 0); andremo quindi a pesare maggiormente il peso su tale variabile.

Per quanto riguarda il riferimento, dall'equazione (2.9) si nota che per ogni passo di campionamento, passiamo al controllore un vettore costante pari al valore di  $r(k_i + 1)$ . Se abbiamo a disposizione l'intero riferimento per i prossimi  $N_p$  passi, possiamo cambiare tale struttura passando al controllo un riferimento variabile lungo tutto l'orizzonte di previsione (*tecnica look-ahead*).

#### 2.4.1 Caso non vincolato

Per capire come trovare l'ottimo  $\Delta U$  che minimizza l'indice  $J$  nel caso senza limiti, riscriviamo quest'ultimo utilizzando (2.6)

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

Si calcola ora la derivata prima della funzione costo (2.11)

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

e si pone la condizione necessaria per trovare il minimo è

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

dalla quale ricaviamo il valore dell'ingresso ottimo per il segnale di controllo:

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

Per il principio del *receding control horizon* ad ogni singolo passo di campionamento andremo a calcolarci un set d'ingresso  $\Delta U$  per i prossimi  $N_c$  passi, dove  $N_c$  è il nostro orizzonte di controllo. Di questo vettore si andrà ad usare soltanto il primo elemento ( $\Delta u(1)$ ) e in seguito il minimo verrà ricalcolato considerando i cambiamenti avuti nel sistema al passo successivo. Ad ogni

istante quindi andremo a definire la variazione dell'ingresso come

$$\begin{aligned}\Delta u(k_i) &= \overbrace{[1 \ 0 \ \dots \ 0]}^{N_c} (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \bar{R}_s r(k_i) - \Phi^T F x(k_i)) = \\ &= K_y r(k_i) - K_{mpc} x(k_i)\end{aligned}\quad (2.12)$$

Tornando quindi alla classica equazione di aggiornamento dello stato si avrà

$$x(k+1) = Ax(k) + B\Delta u(k) \quad (2.13)$$

e sostituendo all'interno di quest'ultima (2.12) risulta

$$x(k+1) = Ax(k) - BK_{mpc}x(k) + BK_y r(k) \quad (2.14)$$

$$= (A - BK_{mpc})x(k) + BK_y r(k) \quad (2.15)$$

dove gli autovalori in catena chiusa possono essere calcolati dal polinomio caratteristico

$$\det[\lambda I - A - BK_{mpc}] = 0. \quad (2.16)$$

### 2.4.2 Vincoli in ingresso e uscita

Si immagini ora di aver dei limiti per le variazioni d'ingresso possibili  $\Delta U^{min} \leq \Delta u(k_i) \leq \Delta U^{max}$  e un orizzonte di controllo  $N_c = 3$ ; dunque le variazioni d'ingresso varranno solo per gli istanti  $\Delta u(k_i)$ ,  $\Delta u(k_i + 1)$  e  $\Delta u(k_i + 2)$ , espressi in forma matriciale

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i + 1) \\ \Delta u(k_i + 2) \end{bmatrix} \leq \begin{bmatrix} \Delta U^{max} \\ \Delta U^{max} \\ \Delta U^{max} \\ -\Delta U^{min} \\ -\Delta U^{min} \\ -\Delta U^{min} \end{bmatrix}. \quad (2.17)$$

Per quanto riguarda l'ingresso ricordando che

$$u(k_i + 1) = u(k_i) + \Delta u(k_i + 1) = \Delta u(k_i + 1) + \Delta u(k_i) + u(k_i - 1)$$

e ponendo ad esempio dei limiti  $-L_1 \leq u(k_i) \leq L_2$ , si ottiene:

$$\begin{bmatrix} u(k_i) \\ u(k_i + 1) \\ u(k_i + 2) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i + 1) \\ \Delta u(k_i + 2) \end{bmatrix} + \begin{bmatrix} u(k_i - 1) \\ u(k_i - 1) \\ u(k_i - 1) \end{bmatrix}$$

la quale esplicitata rispetto a  $\Delta U$  risulta

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ -1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i + 1) \\ \Delta u(k_i + 2) \end{bmatrix} \leq \begin{bmatrix} L_2 - u(k_i - 1) \\ L_2 - u(k_i - 1) \\ L_2 - u(k_i - 1) \\ L_1 + u(k_i - 1) \\ L_1 + u(k_i - 1) \\ L_2 + u(k_i - 1) \end{bmatrix} \quad (2.18)$$

Infine per i limiti sulle variabili di uscita si ricorda che  $Y = Fx(k_i) + \Phi\Delta U$  e considerando soglie definite come  $Y^{min} \leq Y \leq Y^{max}$  si ottiene:

$$\begin{bmatrix} \Phi \\ -\Phi \end{bmatrix} \Delta U \leq \begin{bmatrix} Y^{max} - Fx(k_i) \\ -Y^{min} + Fx(k_i) \end{bmatrix} \quad (2.19)$$

Dunque, calcolare l'ingresso ottimo in un controllo MPC in presenza di vincoli è analogo a trovare il parametro  $\Delta U$  che minimizza la funzione  $J$

$$J = (R_s - Fx(k_i))^T Q (R_s - Fx(k_i)) - 2\Delta U^T \Phi^T Q (R_s - Fx(k_i)) + \Delta U^T (\Phi^T Q \Phi + \bar{R}) \Delta U \quad (2.20)$$

soggetto ai vincoli di disuguaglianza

$$\begin{bmatrix} M1 \\ M2 \\ M3 \end{bmatrix} \Delta U \leq \begin{bmatrix} N1 \\ N2 \\ N3 \end{bmatrix} \quad (2.21)$$

dove le matrici  $M1$ ,  $M2$ ,  $M3$  sono quelle che moltiplicano a sinistra  $\Delta U$  rispettivamente nelle formule (2.17), (2.18) e (2.19), mentre  $N1$ ,  $N2$  e  $N3$  sono i vettori presenti a destra del minore uguale nelle formule citate.

Si nota come l'indice  $J$  sia quadratico e i vincoli siano disuguaglianze lineari, quindi il calcolo dell'ingresso  $\Delta U$  equivale a trovare la soluzione ottima per un problema di programmazione quadratica.

Data la formulazione classica per un problema QP

$$\begin{aligned} & \frac{1}{2} \eta^T H \eta + \eta^T f \\ & A \eta \leq b \end{aligned} \quad (2.22)$$

si va a definire le matrici  $H$  e  $f$  nel caso di controllo MPC

$$\begin{aligned} H &= (\Phi^T Q \Phi + \bar{R}) \\ f &= -\Phi^T Q (R_s - Fx(k_i)) \end{aligned}$$

e le matrici di vincolo

$$A = \begin{bmatrix} M1 \\ M2 \\ M3 \end{bmatrix}, b = \begin{bmatrix} N1 \\ N2 \\ N3 \end{bmatrix}.$$

In questo modo andremo a calcolare la sequenza d'ingresso risolvendo un problema di *programmazione lineare vincolata*.

### 2.4.3 Vincoli sullo stato

Si è visto come nella dichiarazione dei parametri dell'MPC sia possibile inserire vincoli su ingressi e uscite da soddisfare nell'ottimizzazione della funzione obiettivo. La possibilità di inserire un vincolo anche sulle variabili di stato stimate può essere una peculiarità utile, che spesso, come nell'MPC sviluppato nel toolbox di MATLAB, non viene implementata. Tale possibilità ci garantisce di inserire dei limiti senza doverli pesare necessariamente nella nostra funzione obiettivo e non inserire obbligatoriamente un riferimento per queste variabili.

Analizzando un caso pratico, supponiamo di utilizzare un sistema che in uscita restituisca l'accelerazione e che abbia come stato velocità e posizione, e si abbia la necessità di imporre dei limiti su quest'ultima.

Partendo dall'equazione (2.5) andiamo a dichiarare le nuove matrici di spazio aumentato  $A_{pos}$ ,  $B_{pos}$  inserendo nelle ultime righe della definizione di  $A$ , invece di  $C_m \cdot A_m$ ,  $C_m^{pos} \cdot A_m$  dove  $C_m^{pos}$  è la matrice che restituisce il valore dello stato da vincolare. Allo stesso modo andiamo a dichiarare la nuova matrice  $B_{pos}$ .

$$A_{pos} = \begin{bmatrix} A_m & 0_m^T \\ C_m^{pos} A_m & 1 \end{bmatrix} \quad B_{pos} = \begin{bmatrix} B_m \\ C_m^{pos} B_m \end{bmatrix} \quad (2.23)$$

Si va ora a dichiarare le nuove variabili  $F_{pos}$ ,  $\Phi_{pos}$  così come erano state dichiarate in (2.7) inserendo invece di  $A$  e  $B$  le matrici (2.23). Infine quando si andrà ad impostare i limiti per le uscite, oltre ad avere la disuguaglianza (2.19) si avrà:

$$\begin{bmatrix} \Phi_{pos} \\ -\Phi_{pos} \end{bmatrix} \Delta U \leq \begin{bmatrix} x^{max} - F_{pos} x_{pos}(k_i) \\ -x^{min} + F_{pos} x_{pos}(k_i) \end{bmatrix}$$

dove  $x^{max}$ ,  $x^{min}$  sono i valori limite per la variabile di stato e  $x_{pos}$  è l'evoluzione dello stato aumentato

$$x_{pos}(k_i + 1) = A_{pos} \cdot x_{pos}(k_i) + B_{pos} \cdot \Delta U$$

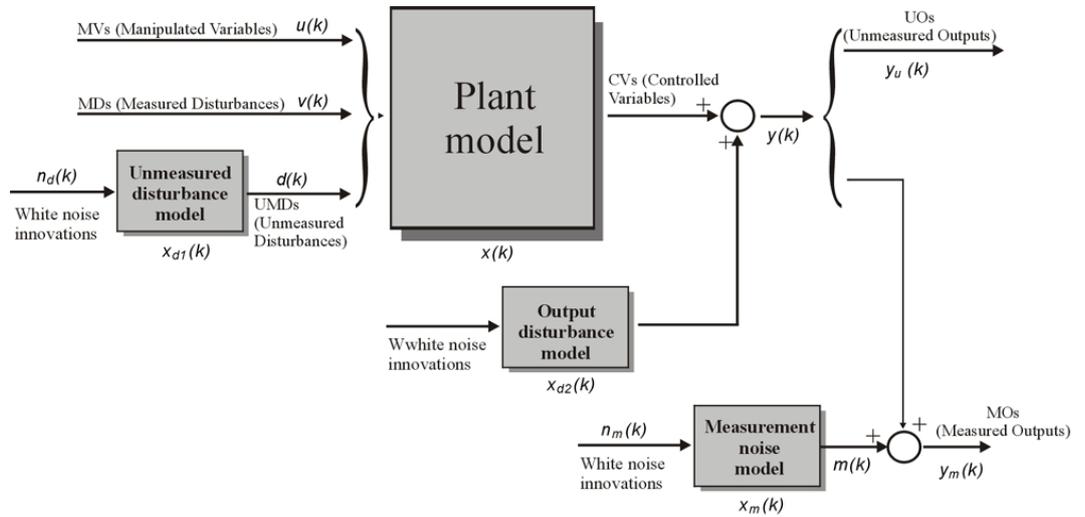


Figura 2.3: Modello nel MPC Toolbox

fatto evolvere contemporaneamente e allo stesso modo di

$$x(k_i + 1) = A \cdot x(k_i) + B \cdot \Delta U.$$

## 2.5 MPC Toolbox di Matlab

Per le prime prove e per imparare ad usare l'ambiente del controllo predittivo si è usato l'MPC toolbox di Matlab. Dato il modello da simulare ed eventualmente inserendo l'errore elemento di disturbo (misurato o non misurato) si è andati a vedere come la risposta del sistema controllato insegue in modo ottimo una traiettoria da noi prefissata utilizzando eventualmente dei vincoli.

### 2.5.1 Modello di Predizione

In figura 2.3 è rappresentato lo schema a blocchi del controllo MPC sviluppato nel Toolbox, il quale evidenzia la struttura di progettazione e definisce i modelli e le variabili che lo caratterizzano.

Il sistema consiste in:

- un modello del *processo* da controllare, in cui gli ingressi sono le *manipulated variables*, *measured disturbances* e *unmeasured disturbances*;
- un modello che genera gli *unmeasured disturbances*.

Se non si ha bisogno di specificare il modello che genera i disturbi, il controllore assume di default che questi sono generati da integratori pilotati da rumore bianco.

Il modello del processo è una sistema tempo-invariante lineare descritto dalle equazioni:

$$\begin{aligned}x(k+1) &= Ax(k) + B_u(k)u(k) + B_v(k)v(k) + B_d(k)d(k) \\y_m(k) &= C_mx(k) + D_{vm}v(k) + D_{dm}d(k) \\y_u(k) &= C_u x(k) + D_v uv(k) + D_{du}d(k)\end{aligned}$$

dove  $x(k)$  è il vettore  $n_x$ -dimensionale dello stato del processo,  $u(k)$  è il vettore  $n_u$ -dimensionale degli ingressi (MV),  $v(k)$  è il vettore  $n_v$ -dimensionale dei disturbi misurati (MD),  $d(k)$  è il vettore  $n_d$ -dimensionale dei disturbi non misurati (UD) che entrano nel processo,  $y_m(k)$  è il vettore delle uscite misurate (MO) e  $y_u$  è il vettore degli uscite non misurate (UO). La dimensione totale  $n_y$  del vettore  $y(k)$  comprende sia  $y_m(k)$  sia  $y_u(k)$ .

L'MPC Toolbox accetta sia processi espressi come sistemi LTI sia modelli ottenuti tramite dati di input/output usando l'*Identification Toolbox*. Unica nota che da l'help del Toolbox è quella di usare un modello del processo in cui gli ingressi non abbiano un effetto diretto sulle uscite.

Se si usa un modello continuo, questo verrà campionato con lo stesso tempo di campionamento del controllore.

## 2.5.2 Ottimizzazione

### Standard Form

Dato il modello di predizione del sistema e assumendo che le stime di  $x(k)$ ,  $x_d(k)$  sono disponibili al tempo  $k$ , l'ingresso di controllo del MPC al passo  $k$  è ottenuta risolvendo il problema di minimizzazione dell'indice (2.24)

$$\begin{aligned}& \min_{\Delta u(k|k), \dots, \Delta u(m-1+k|k), \varepsilon} \sum_{i=0}^{p-1} \left( \sum_{j=1}^{n_y} \left| w_{i+1,j}^y (y_j(k+1+i|k) - r_j(k+i+1)) \right|^2 \right. \\ & \left. + \sum_{j=1}^{n_u} \left| w_{i,j}^{\Delta u} \Delta u_j(k+1+i|k) \right|^2 + \sum_{j=1}^{n_u} \left| w_{i,j}^u (u_j(k+i|k) - u_{jtarget}(k+i)) \right|^2 + \rho_\varepsilon \varepsilon^2 \right) \quad (2.24)\end{aligned}$$

dove  $u(k) = u(k-1) + \Delta u(k|k)$  e  $\Delta u(k|k)$  è il primo elemento della sequenza d'ingresso calcolata dall'ottimizzatore.

Quando il vettore di riferimento  $r$  non è noto in avanti, il riferimento attuale  $r(k)$  viene usato come riferimento per l'intera predizione, ovvero  $r(k+i+1) = r(k)$ . Nel model predictive control la conoscenza della traiettoria futura è chiamata anche *look-ahead* o *anticipative action*. Una simile azione può essere usata anche per i disturbi misurati  $v(k)$ .

$w_{i,j}^{\Delta u}$ ,  $w_{i,j}^u$ ,  $w_{i,j}^y$  sono i pesi non negativi per le variabili rispettivamente  $\Delta u$ ,  $u$  e  $y$ ; più è piccolo  $w$  meno importanza si dà al termine corrispondente nella funzione obiettivo durante la minimizzazione. Nell'equazione 2.24 i vincoli su  $u$ ,  $\Delta u$  e  $y$  sono resi "soft" introducendo la variabile di slack  $\varepsilon \geq 0$  con il suo relativo peso  $\rho_\varepsilon$ .

Dato che i vincoli grazie alla variabile di slack saranno spesso sforabili, il problema QP sarà quasi sempre ammissibile; però se per qualche ragione numerica non dovesse esserlo si andrà ad utilizzare come ingresso il secondo elemento della sequenza ottima calcolata all'istante precedente.

### Alternative Cost Function

Il Toolbox di Matlab, oltre ad offrire una funzione costo "Standard", dà la possibilità all'utente di utilizzarne una alternativa la quale assomiglia molto a quella presente all'indice definito utilizzando la procedura descritta del Wang nella precedente sezione.

$$J(\Delta u, \varepsilon) = \sum_{i=0}^{p-1} \left( [y(k+i+1|k) - r(k+i+1)]^T Q [y(k+i+1|k) - r(k+i+1)] + \Delta u(k+i|k)^T R_{\Delta u} \Delta u(k+i|k) + [u(k+i|k) - u_{target}(k+i)]^T R_u [u(k+i|k) - u_{target}(k+i)] + \rho_\varepsilon \varepsilon^2 \right)$$

dove  $Q$  è una matrice quadrata di dimensione  $n_y$  e  $R_{\Delta u}$  e  $R_u$  sono matrici  $n_u \times n_u$ , tutte semidefinite positive.

Le due funzioni costo presenti nel Toolbox sono equivalenti quando i pesi  $w_{i,j}^y$ ,  $w_{i,j}^{\Delta u}$ ,  $w_{i,j}^u$  sono costanti per tutti i passi di predizione  $1, \dots, P$ , e quando le matrici  $Q$ ,  $R_{\Delta u}$  e  $R_u$  diagonali hanno come elementi rispettivamente i quadrati dei pesi  $w_{i,j}^y$ ,  $w_{i,j}^{\Delta u}$ ,  $w_{i,j}^u$ .

### 2.5.3 Object Reference

Tutti i parametri definiti nella legge del controllo MPC (tra i quali orizzonte di predizione, pesi, limiti...etc) sono memorizzati in un oggetto *MPC* il quale

in fase di programmazione dovrà essere impostato “ad hoc” per far inseguire correttamente il riferimento al sistema in *closed loop*. Tali proprietà sono elencate qui di seguito.

Tabella 2.1: Oggetto MPC Controller

Proprietà	Descrizione
Manipulated variables	Limiti inferiori e superiori di ingressi e variazione d’ingresso, valori ECR, nome, unità e riferimento d’ingresso
Output variables	Limiti inferiori e superiori d’uscita, valori ECR, nome, unità
Disturbances variables	Nome dei disturbi e unità
Weights	Definizione dei pesi che influiscono sulle performance del controllore
Model	Modello del processo, del disturbo d’ingresso e del rumore d’uscita
Ts	Tempo di campionamento
Optimizer	Parametri per il QP solver
PredictionHorizon	Orizzonte di predizione
ControlHorizon	Orizzonte di controllo



## Capitolo 3

# Programmazione Quadratica

Si è visto nella formulazione del problema MPC che per calcolare l'ingresso ottimo per il nostro sistema si andrà a risolvere un problema di programmazione quadratica.

La programmazione quadratica fa parte dei problemi di *ottimizzazione matematica* che si occupa di minimizzare (o massimizzare) una funzione costo quadratica con diverse variabili, soggetta a vincoli lineari. Assumendo che  $x \in \mathbb{R}^n$ , la matrice  $H^{n \times n}$  sia simmetrica e che  $c$  sia un vettore  $n \times 1$ , il problema può essere così formulato:

$$\min f(x) = \frac{1}{2}x^T Hx + c^T x, \quad (3.1)$$

soggetto a uno o più vincoli espressi nella forma :

$$Ax \leq b \quad Ex = d.$$

Se la matrice  $H$  è semidefinita positiva, allora  $f(x)$  è una funzione convessa: in questo caso la programmazione quadratica avrà un minimizzatore globale se esiste un vettore  $x$  che soddisfa i vincoli. Se invece  $H$  è definita positiva, allora il problema ha una soluzione ammissibile e il minimizzatore globale è unico. Se  $H$  è 0 allora il problema diventa di programmazione lineare.

Tra i vari metodi possibili per risolvere un problema di programmazione quadratica citiamo i più famosi:

- Active set
- Interior point.

Per i nostri scopi si utilizzerà la funzione *quadprog* di *MATLAB*, funzione inclusa nel *optimization toolbox*.

Tale metodo risolverà l'ottimizzazione in funzione del tipo di problema affrontato utilizzando fondamentalmente due algoritmi: *Medium-scale* e *Large-scale*. Un algoritmo si dice *Large-scale* quando usa algebra lineare che non richiede di memorizzare e operare su matrici piene. Il metodo *Medium-scale* crea internamente delle matrici piene e usa algebra lineare densa. Metodi *large-scale* possono essere usati solo per problemi piccoli, ovvero quando nel problema QP compaiono solo vincoli di uguaglianza e quando la matrice  $H$  della funzione costo quadratica e la matrice  $A$  dei vincoli di uguaglianza sono sparse.

Nel nostro caso si userà quasi sempre algoritmi *Medium-scale* per cui è implementata una strategia *Active-set*.

### 3.1 Active Set

La procedura per arrivare alla soluzione del problema si articola in due fasi:

- la prima va alla ricerca di una soluzione ammissibile;
- la seconda procede nel creare una sequenza interattiva di punti ammissibili che convergono alla soluzione finale.

Consideriamo il generico problema di programmazione quadratica nella forma:

$$\min f(x) = c_0 + c^T x + x^T Q x \quad (3.2)$$

soggetto ai vincoli

$$a_i^T x \geq b_i \quad a_i^T x = b_i \quad a_i^T x \leq b_i.$$

Inizialmente, poniamo l'attenzione sui problemi di QP con soli vincoli di uguaglianza

$$\min f(x) = c_0 + c^T x + x^T Q x \quad (3.3)$$

$$a_i^T x = b_i \quad (3.4)$$

e supponiamo di scegliere un punto  $x^0$  ammissibile e utilizzare l'approssimazione in serie di Taylor del secondo ordine per la funzione obiettivo (3.2).

Essendo quadratica, la rappresentazione del secondo ordine della funzione obiettivo è esatta. Il problema quindi può essere così riscritto:

$$\min f(x^0 + \Delta x) = f(x^0) + (\nabla_x f(x^0))^T \cdot \Delta x + (\Delta x)^T Q \Delta x \quad (3.5)$$

$$A \cdot \Delta x = 0 \quad (3.6)$$

dove  $\Delta x$  è il passo di ottimizzazione che permette di raggiungere il minimo (globale) del problema partendo dal punto  $x^0$ , ed i vincoli garantiscono che la soluzione resti ammissibile.

Possiamo riportare la funzione Lagrangiana per il problema:

$$L(\Delta x, v) = f(x^0) + (\nabla_x f(x^0))^T \cdot \Delta x + (\Delta x)^T \cdot Q \cdot \Delta x + v^T \cdot (A \cdot \Delta x)$$

e le condizioni di ottimalità

$$\nabla_{\Delta x} L(\Delta x, v) = 0 \quad A \cdot \Delta x = 0$$

Trovare il passo  $\Delta x$  che soddisfa le condizioni di ottimalità corrisponde a risolvere il seguente sistema lineare:

$$\begin{bmatrix} -2Q & A^T \\ A & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ v \end{bmatrix} = \begin{bmatrix} \nabla_x f(x^0) \\ 0 \end{bmatrix}. \quad (3.7)$$

Nei metodi Active set si determina la soluzione della minimizzazione, risolvendo una sequenza di problemi di QP con i soli vincoli di uguaglianza. Tali metodi si basano sull'osservazione che, per qualsiasi soluzione (ed in particolare per quella ottima), i vincoli di disuguaglianza non stretti possono essere scartati ed i vincoli di disuguaglianza stretti possono essere riscritti come vincoli di uguaglianza, senza influenzare la soluzione.

Di seguito sono discussi i passaggi fondamentali del metodo.

Si sceglie un punto iniziale  $x^0$  e si determina l'insieme dei vincoli soddisfatti all'uguaglianza in  $x^0$ . Questi formano l'Active Set iniziale  $A^0$ . Poi, iterativamente ( $k > 0$ ) si utilizza il seguente procedimento:

1. si cerca l'incremento  $(\Delta x)^k$  ed il vettore di moltiplicatori  $v^k$  che porterebbe all'ottimo del problema di QP con i soli vincoli in  $A^k$  soddisfatti all'uguaglianza. L'incremento ottimo ed il relativo vettore dei moltiplicatori possono essere ottenuti risolvendo il sistema lineare (3.7).
2. Se  $(\Delta x)^k \neq 0$  (oppure se  $\|(\Delta x)^k\|$  non è sufficiente piccolo), un ottimo del problema con i vincoli in  $A^k$  soddisfatti all'uguaglianza è in  $x^k +$

$(\Delta x)^k$ . Tuttavia il punto  $x^k + (\Delta x)^k$  potrebbe essere inammissibile per alcuni dei vincoli fuori dall'Active Set. Pertanto, per ogni  $i \in G$  non incluso nell'Active Set, per cui si ha  $a_i^T \cdot \Delta x < 0$  (l'incremento  $\Delta x$  riduce lo slack del vincolo), si valuta il seguente coefficiente:

$$\lambda_i^G := \frac{a_i^T \cdot x^T - b_i}{-a_i^T \cdot (\Delta x)}.$$

Il numeratore rappresenta lo slack del vincolo  $i$  nel punto  $x^k$ , mentre il denominatore è il decremento nello slack del vincolo  $i$  a seguito dello spostamento indicato da  $(\Delta x)^k$ . Intuitivamente,  $\lambda_i^G$  è la massima frazione di incremento  $(\Delta x)^k$  che può essere effettuata senza violare il vincolo  $i$ . In modo analogo, per ogni  $i \in L$  non incluso nell'Active Set, per cui  $a_i^T \cdot \Delta x > 0$ , si valuta il coefficiente:

$$\lambda_i^L := \frac{-a_i^T \cdot x^T - b_i}{a_i^T \cdot (\Delta x)}.$$

La massima frazione di incremento  $(\Delta x)^k$  che può essere effettuata senza violare nessun vincolo è

$$\lambda := \min(\{\lambda_i^G\} \cup \{\lambda_i^L\} \cup \{1.0\}) \quad (3.8)$$

ed il punto candidato a soluzione è

$$x^{k+1} := x^k + \lambda(\Delta x)^k.$$

Se  $\lambda \neq 1$ , il vincolo  $i^*$  che determina la frazione di incremento ( $i^* = \arg \min(\{\lambda_i^G\} \cup \{\lambda_i^L\})$ ) diventa attivo e inserito nell'Active Set ( $A^{k+1} := A^k \cup i^*$ ).

3. Se  $(\Delta x)^k = 0$  (oppure se  $\|(\Delta x)^k\|$  è sufficientemente piccolo), il punto  $x^k$  è un ottimo del problema in cui i soli vincoli in  $A^k$  sono soddisfatti all'uguaglianza. Tuttavia, le condizioni imposte all'Active Set potrebbero essere troppo restrittive. Un'incogruenza nei vincoli di segno sui moltiplicatori relativi ai limiti nell'Active Set indica la possibilità di migliorare la soluzione corrente escludendo vincoli. Quindi, per ogni  $i \in L$  incluso nell'Active Set, se  $v_i > 0$   $i$  deve essere rimosso dall'Active Set ( $A^{k+1} := A^k \setminus \{i\}$ ); in modo analogo, per ogni  $i \in G$  incluso nell'Active Set, se  $v_i < 0$   $i$  deve essere rimosso dall'Active Set.

4. Se il candidato a soluzione ottima è cambiato (passo 2), oppure dei vincoli sono stati rimossi dall'Active Set (passo 3), si pone  $k := k + 1$  e si itera il procedimento (passi 1-4). Altrimenti,  $x^k$  è una soluzione ottima per il problema di partenza.



## Capitolo 4

# Applicazione MPC

In questo capitolo si vuole analizzare il controllo MPC per un semplice sistema lineare del secondo ordine ad un grado di libertà *Massa-Molla-Smorzatore*. Non conoscendo infatti, le equazioni che descrivono la piattaforma, l'utilizzo di un sistema del secondo ordine è una buona approssimazione del ritardo meccanico nel caso si utilizzi una *feedback linearization*. Si andrà a mettere in evidenza le caratteristiche e i vantaggi che offre il controllo predittivo, con le diverse parametrizzazioni possibili che sono state implementate per questa strategia.

### 4.1 Modello

Per il sistema rappresentato in figura 4.1, assumendo che eventuali dispersioni siano trascurabili, la forza applicata alla molla è proporzionale all'allungamento  $x$  con una costante di proporzionalità  $k$  e la forza generata dalla massa è proporzionale all'accelerazione di questa come descritto dalla seconda legge della dinamica di Newton. Inoltre, la presenza dell'ammortizzazione  $B$ , inserisce una ulteriore forza proporzionale alla velocità della massa come descritto da (4.1).

$$m \frac{\partial^2}{\partial t^2} x(t) + \underbrace{B \frac{\partial}{\partial t} x(t)}_{\text{attrito viscoso}} + \underbrace{kx(t)}_{\text{richiamo della molla}} = \underbrace{F(t)}_{\text{forza}} \quad (4.1)$$

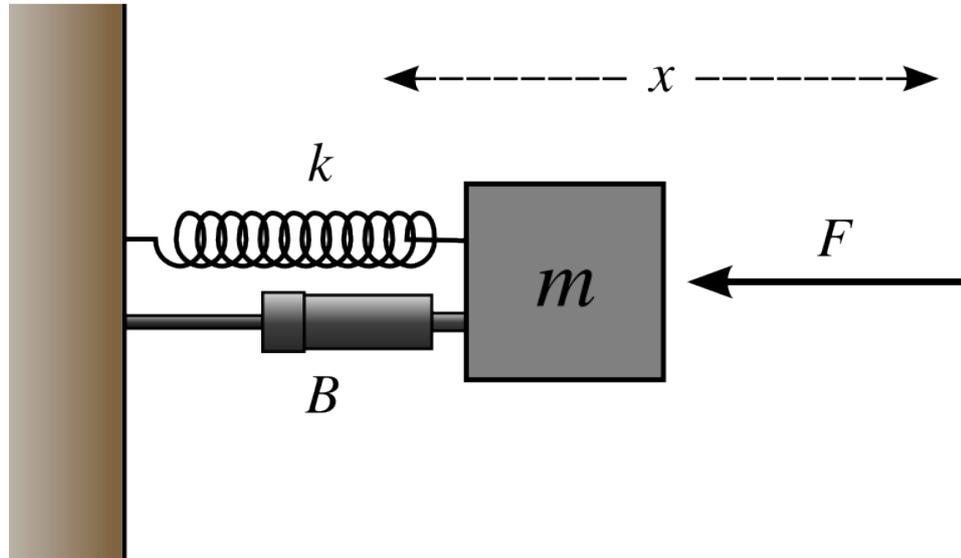


Figura 4.1: Sistema massa-molla-smorzatore

Tale equazione può essere riformulata attraverso la classica espressione che descrive un sistema del secondo ordine

$$\frac{\partial^2}{\partial t^2}x(t) + 2\zeta\omega_0\frac{\partial}{\partial t}x(t) + \omega_0^2x(t) = \tilde{u}(t) \quad (4.2)$$

dove

- $\omega_0$ =pulsazione naturale (velocità del sistema);
- $\zeta$ =coefficiente di smorzamento (tipo di risposta del sistema).

Dunque per il sistema *MMS* (*Massa-Molla-Smorzatore*)

$$\omega_0 = \sqrt{\frac{k}{m}} \quad \zeta = \frac{b}{2\sqrt{km}} \quad \tilde{u}(t) = \frac{F(t)}{m}$$

#### 4.1.1 Risposta al gradino

Si va ora ad analizzare la risposta al gradino per un sistema del secondo ordine avendo in ingresso un segnale del tipo  $\tilde{u}(t) = \omega_0^2\delta_{-1}(t)$ .

Per il nostro lavoro si andrà ad utilizzare la funzione di trasferimento derivata dall'equazione (4.2) e parametrizzata con valori  $\zeta = 0.25$  e  $\omega_0 = 25$ . La funzione presenta la risposta al gradino in figura, 4.2 la quale è caratterizzata da un tempo di salita  $t_s = 0.1s$  e un tempo di assestamento  $t_a = 0.4$ . Tali parametri sembrano essere adatti per modellare la movimentazione di un singolo asse della piattaforma.

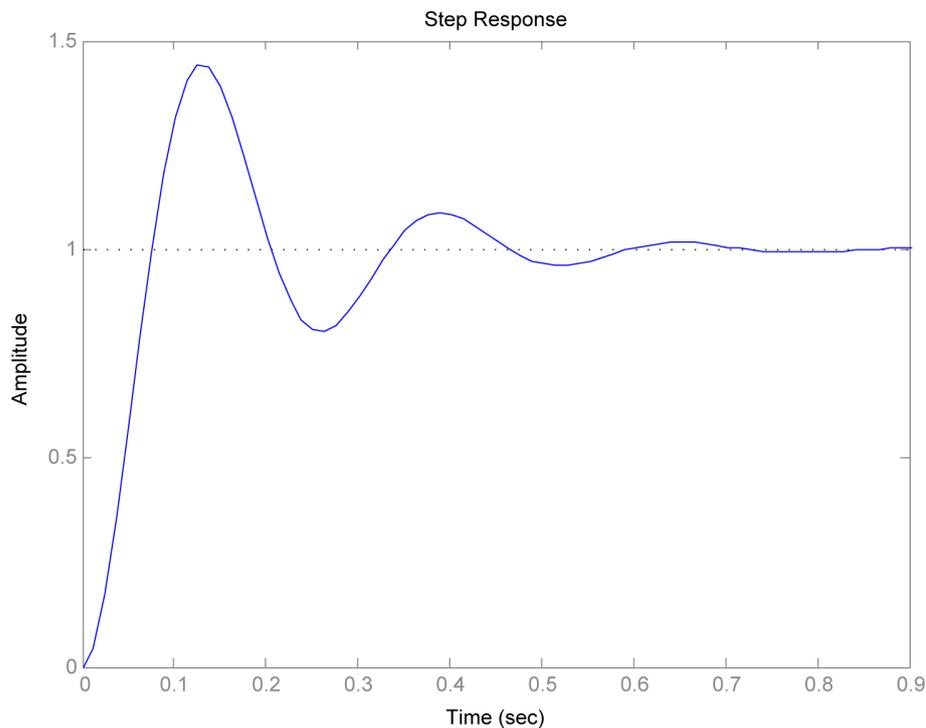


Figura 4.2: Risposta al gradino del sistema MMS scelto

## 4.2 Controllo MPC con il ToolBox di MATLAB

Supponiamo di voler controllare utilizzando il Model Predictive Control lo spostamento  $x(t)$  della massa  $m$  agendo sulla forza  $F(t)$  nel sistema descritto precedentemente. Per il controllo di questo primo sistema si è scelto l'uso dell'MPC ToolBox implementato da Bemporad presente in MATLAB perchè, oltre ad essere ben collaudato, presenta un'interfaccia semplificata e metodi già sviluppati, ed in poco tempo si riesce a configurare l'oggetto "MPC" in modo da avere il sistema controllato in catena chiusa.

Per far ciò si è andrà ad impostare i campi del controllore presenti nell'*Object reference* con i seguenti valori:

- peso nullo per le variabili d'ingresso in modo tale da non essere presenti nella minimizzazione della funzione obiettivo. Dunque solo le variabili *input rate* saranno valorizzate; questo perchè non si ha al momento nessun interesse nel far inseguire un riferimento all' ingresso (o *MV*);
- nessun tipo di vincolo, sia per le variabili d'uscita (*OV*), sia per le

variabili d'ingresso ( $MV$ ) e sia per la variazione d'ingresso ( $Ratemin$  e  $Ratemax$ );

- riferimento periodico;
- orizzonte di predizione e di controllo impostati a 25 passi;
- nessun tipo di disturbo e modello del sistema identico a quello del processo (caso ideale);
- peso unitario sull'uscita e dell'ordine del  $10^{-5}$  per la variazione d'ingresso ( $\Delta U$ );
- tempo di campionamento uguale a quello del modello pari a  $T_s = 0.04$ .

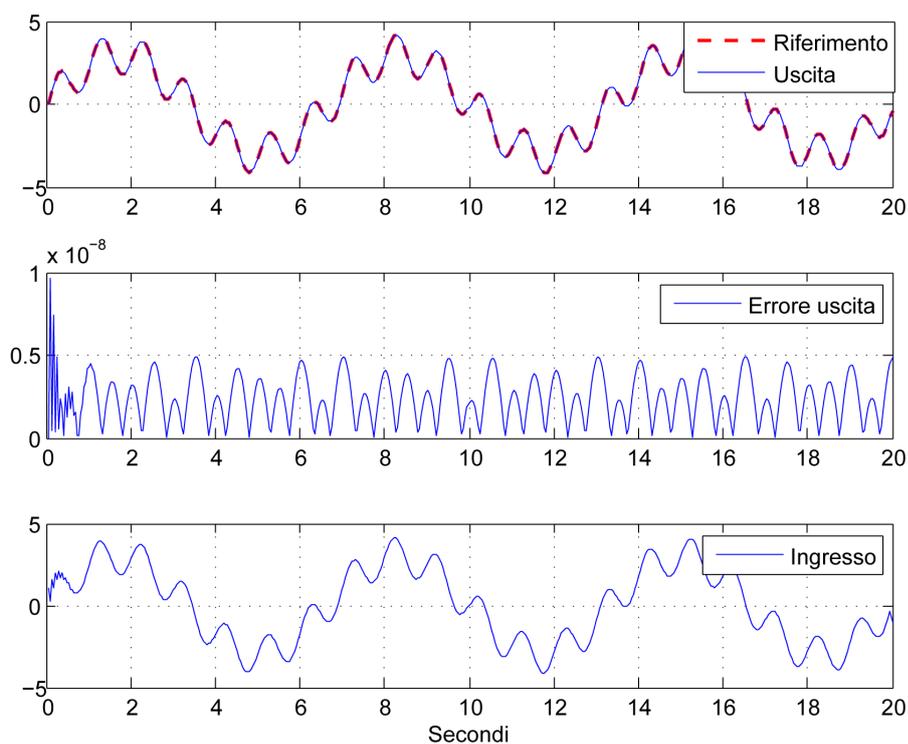


Figura 4.3: Controllo MPC del sistema senza nessun vincolo utilizzando l'MPC Toolbox di MATLAB

Nel grafico 4.3 viene rappresentato 20s di simulazione per il sistema controllato attraverso l'MPC sviluppato da Bemporad. Il controllore avendo a disposizione il modello del sistema (che in questo caso banale è identico a

quello del processo), non avendo disturbi e potendo lavorare senza limiti, riesce ad inseguire perfettamente il riferimento. Inoltre il calcolo dell'ingresso ottimo avviene in maniera analitica e ciò garantisce una soluzione sempre ammissibile e un tempo di calcolo estremamente inferiore al caso vincolato. Utilizzando ora gli stessi valori definiti nel Toolbox, si imposta i

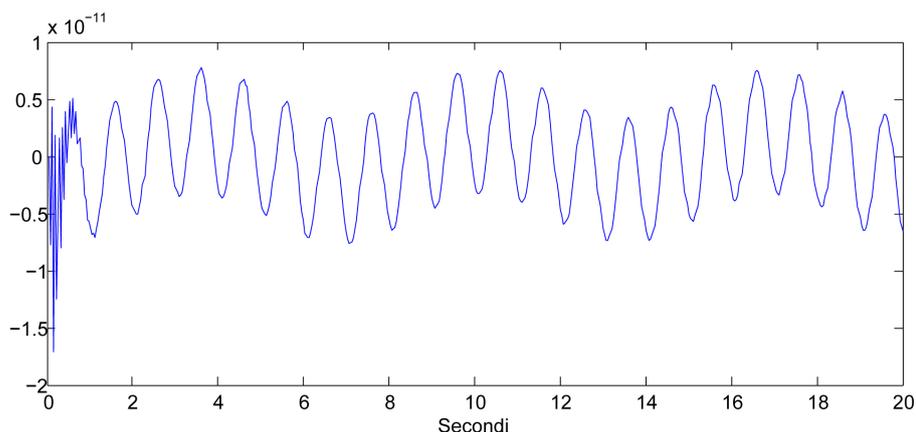


Figura 4.4: Errore tra il controllo MPC di Bemporad e quello sviluppato seguendo l'approccio del Wang

parametri del controllo MPC implementato e si va a valutare la risposta confrontandola con quella precedente (figura 4.4). Si evidenzia come l'MPC di Bemporad con pesi nulli in ingresso restituisca le stesse performance del controllo sviluppato nel capitolo 3. Questo ci garantisce la possibilità di usare una strategia che abbia le stesse funzionalità di un controllo famoso e collaudato come quello presente nel Toolbox. Inoltre, avendo effettuato l'implementazione partendo dal modello in spazio di stato, si ha avuto modo di capire nello specifico la strategia MPC con la possibilità di inserire nuove funzionalità in caso di bisogno, prerogativa che sarebbe stata negata utilizzando l'MPC Toolbox, non permettendo implementazioni supplementari essendo un metodo a "scatola chiusa".

### 4.3 Controllo MPC

Un sistema di *motion cueing* riceve in ingresso le accelerazioni che la piattaforma dovrebbe riprodurre e restituisce le accelerazioni elaborate tenendo conto dei limiti fisici del simulatore.

Per la presentazione di un esempio pratico dell'MPC si va a definire un modello del secondo ordine che rappresenti la movimentazione lungo un singolo asse, in questo caso  $x$ , così come si è descritto il sistema massa molla smorzatore.

Partendo dalla funzione di trasferimento espressa attraverso la trasformata di Laplace (4.3)

$$\frac{1}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad (4.3)$$

con parametri  $\zeta = 0.25$  e  $\omega_0 = 25$ , questa viene definita in spazio di stato usando il metodo *tf2ss* di MATLAB che restituirà le seguenti matrici:

$$A = \begin{bmatrix} -12.5 & -625 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (4.4)$$

$$C = \begin{bmatrix} 0 & 625 \end{bmatrix}. \quad (4.5)$$

Nel sistema (4.5) non viene tenuto conto dell'evoluzione della posizione e della velocità, ma solo dell'accelerazione

Nei parametri che andremo ad impostare per il motion cueing utilizzando *MPC*, oltre al modello del sistema, saranno presenti anche i vincoli della piattaforma (posizione, velocità e accelerazione) che garantiranno che l'algoritmo di ottimizzazione calcoli il miglior ingresso. Dunque si andrà ad aumentare lo stato del sistema per tenere conto dell'evoluzione di velocità e posizione definendo le matrici in (4.6)-(4.7)

$$A = \begin{bmatrix} -12.5 & -625 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.6)$$

$$C = \begin{bmatrix} 0 & 625 & 0 & 0 \end{bmatrix}. \quad (4.7)$$

Nella formulazione base del controllo predittivo i vincoli sono espressi per variabili d'ingresso e d'uscita. Sarà necessario riscrivere la matrice  $C$  in modo che tra le uscite ci sia anche la posizione; discorso analogo potrà essere fatto per la velocità. La matrice  $C$  del sistema (4.7) viene riscritta come

$$C = \begin{bmatrix} 0 & 625 & 0 & 0 \\ 0 & 0 & 0 & 625 \end{bmatrix}$$

Non sapendo se nella piattaforma si riuscirà avere a disposizione trasduttori di velocità o posizione, queste grandezze dovranno essere stimate.

Si va ora discretizzare il sistema attraverso il comando MATLAB *c2dm*; utilizzando un sistema di conversione “zoh” e un tempo di campionamento  $T_s = 0.04$  risulterà il sistema discreto da inserire nel modello del controllo predittivo. Il valore di  $T_s$  sarà anche utilizzato dall’MPC ed è sembrato essere accettabile considerando i tempi con cui lavora il software già esistente VI-DriveSim senza appesantire l’algoritmo di ottimizzazione. Se nella modellizzazione del sistema si vuole inserire un ritardo si hanno due possibilità per farlo:

- ritardare l’uscita aumentando lo stato del sistema inserendo nella matrice  $A$  la matrice  $C$  e costruendone una nuova la quale manderà in un uscita la nuova variabile di stato, in modo tale da preservare il significato degli stati;
- ritardare direttamente la funzione di trasferimento continua attraverso il comando MATLAB *outpudelay* e trasformarla in spazio di stato come descritto prima.

#### 4.3.1 Caso non vincolato, variazione dei pesi d’uscita

Si va ora ad applicare il controllo *MPC* al sistema appena descritto impostando i seguenti parametri:

- orizzonte di controllo e di predizione pari a 25 passi in avanti ovvero si prenderà in considerazione 1s del riferimento;
- pesi variabili per le uscite in modo da valutare diversi tipi di risposta;
- peso in ingresso quasi nullo ( $10^{-10}$ ) non volendo dare importanza a quanto grande diverrà il segnale di controllo;
- nessun limite in ingresso e uscita;
- controllo senza look-ahead, ovvero il riferimento per i prossimi  $n$  passi sarà preso costante con valore l’istante successivo.

Dalle figure 4.5 e 4.6 si nota che variando solamente i pesi delle uscite nella funzione costo si hanno due risposte completamente differenti. Nel primo caso (figura 4.5) l’errore tra il riferimento e l’accelerazione è piuttosto elevato ma si ha la posizione più contenuta, sempre sotto i 10m. Nel secondo caso

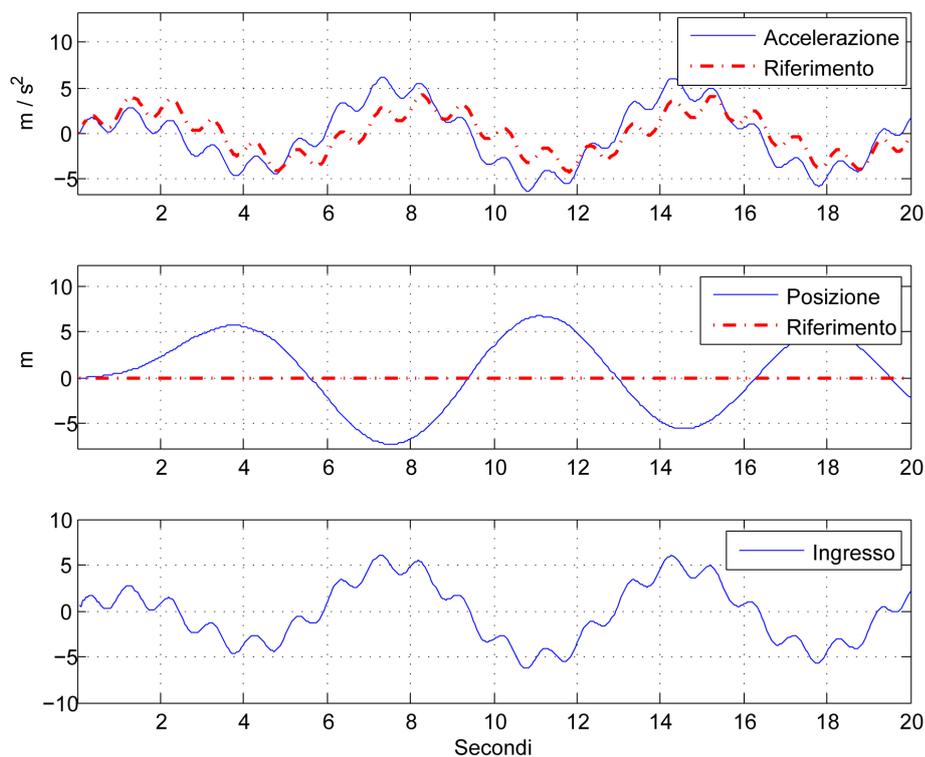


Figura 4.5: Controllo MPC del sistema senza nessun vincolo, peso in uscita unitario

invece (figura 4.6) il controllo segue maggiormente il riferimento in accelerazione e permette che la posizione lungo  $x$  assuma valori molto alti ( $50m$ ). Questo perchè, al contrario del caso precedente, l'errore tra l'accelerazione e la sua traiettoria è pesato maggiormente rispetto a quello della posizione.

Nel caso pratico, inserire un riferimento nullo e andare a pesare l'errore in posizione, significa voler tenere la postazione pilota il più vicino possibile al centro della piattaforma, soprattutto quando si è distanziata parecchio per inseguire il riferimento in accelerazione.

### 4.3.2 Caso vincolato, limite in ingresso

Come detto, il caso non vincolato è poco utilizzato nel controllo MPC, perchè in qualsiasi sistema reale ci sono dei vincoli da rispettare. Il controllo predittivo nella sua formulazione prevede la dichiarazione implicita dei limiti, che verranno presi in considerazione durante il calcolo dell'ingresso attraverso la risoluzione di un problema di programmazione quadratica.

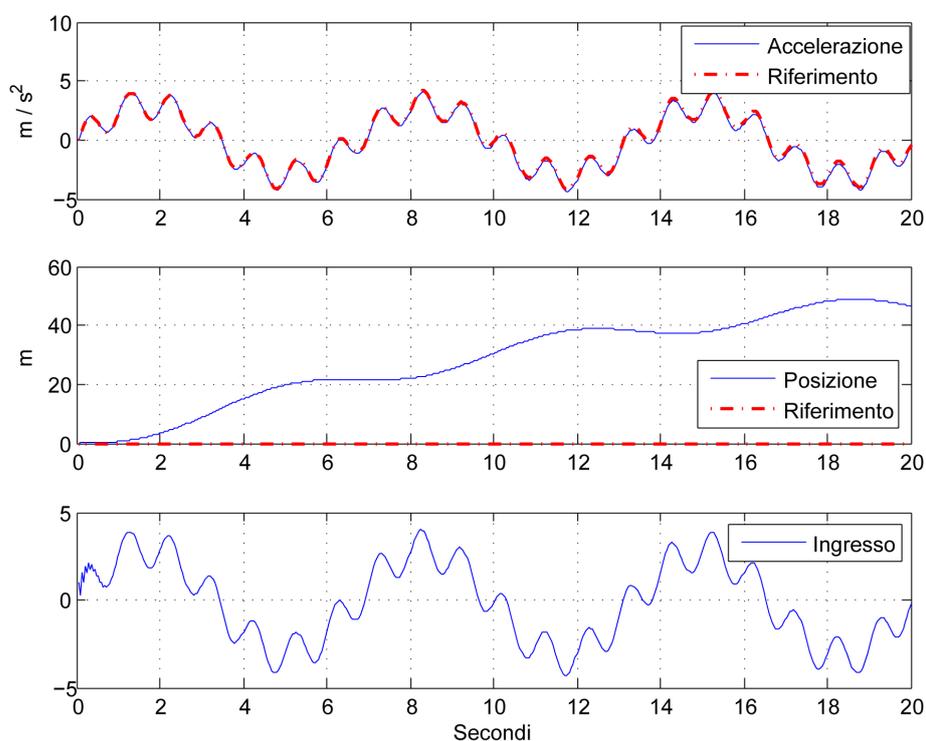


Figura 4.6: Controllo MPC del sistema senza nessun vincolo, peso unitario per l'accelerazione e peso 0.01 sulla posizione

In figura 4.7 è rappresentata la risposta del sistema controllato con una soglia in ingresso pari a 2.5.

Fino a quando l'ingresso rimane dentro i margini imposti, il sistema insegue il riferimento quasi perfettamente, grazie al piccolo peso impostato per la posizione nell'indice  $J$ . In prossimità del vincolo, l'ingresso e l'uscita vanno in saturazione; non potendo fornire un controllo più elevato, resta a tal valore finché il riferimento non risulta inferiore al limite. La variazione dell'orizzonte di predizione fissato a 25 in questo caso non incide molto sulle performance del controllo; infatti anche con un valore inferiore alla decina di passi si avrebbe lo stesso risultato di figura 4.7.

Un comportamento simile si ha fissando un vincolo per l'uscita di accelerazione. Il controllo permette l'inseguimento della traiettoria finché essa resta sotto il margine imposto; quando ci arriva satura fino a quando il riferimento scende sotto la soglia. Un diverso tipo di risposta può essere realizzata inserendo pesi diversi, cambiando in questo modo la funzione obiettivo.

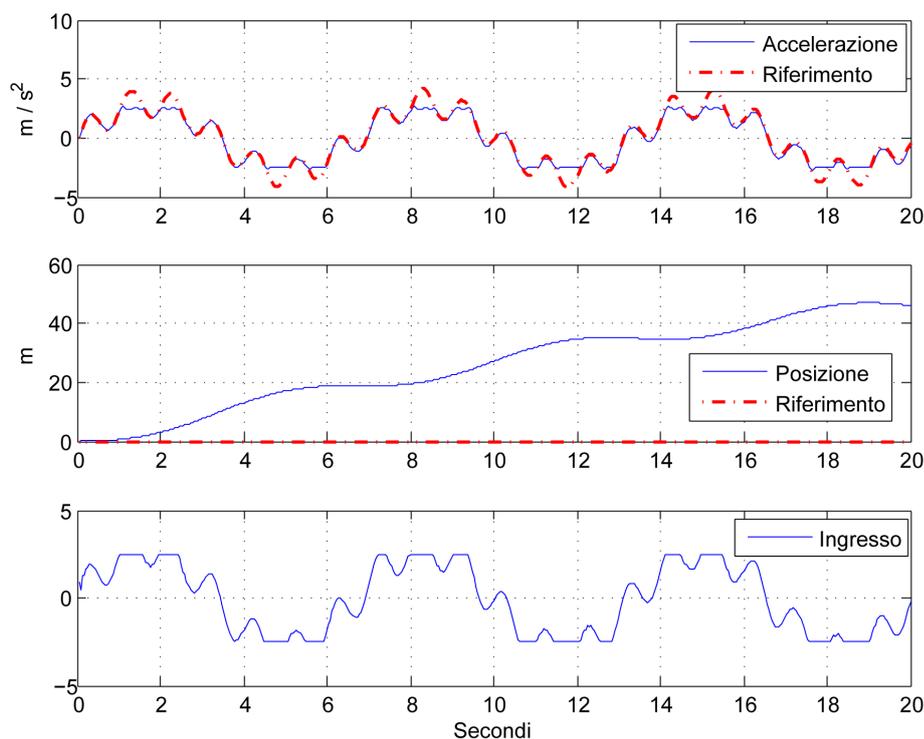


Figura 4.7: Controllo MPC del sistema con vincolo in ingresso a 2.5, peso sulla posizione 0.01

### 4.3.3 Caso vincolato, limiti in posizione

Un caso interessante si presenta quando si vuole limitare la posizione della piattaforma e inseguire una traiettoria ben precisa in accelerazione, come in figura 4.8. In questo specifico caso il controllo cerca di seguire il più possibile l'accelerazione desiderata finchè il limite in posizione lo permette. Rispetto al caso precedente, vincolo in ingresso o in accelerazione, la saturazione presente in figura 4.7 nell'inseguimento della traiettoria non potrà verificarsi perchè al raggiungimento della soglia in posizione, accelerazione e velocità dovranno essere di verso opposto per non superare il vincolo e dunque il sistema già prima del limite dovrà decelerare (caso di soglia limite positiva). La decelerazione in questo specifico caso non dovrà essere percepibile per il pilota, che avrebbe una sensazione di moto contrastante con quella che il sistema visivo sta trasmettendo. Evitare questa situazione è possibile restando entro alcuni limiti di soglia in accelerazione come descritto da Soyka [15], sotto i quali il sistema vestibolare non riesce percepire i movimenti

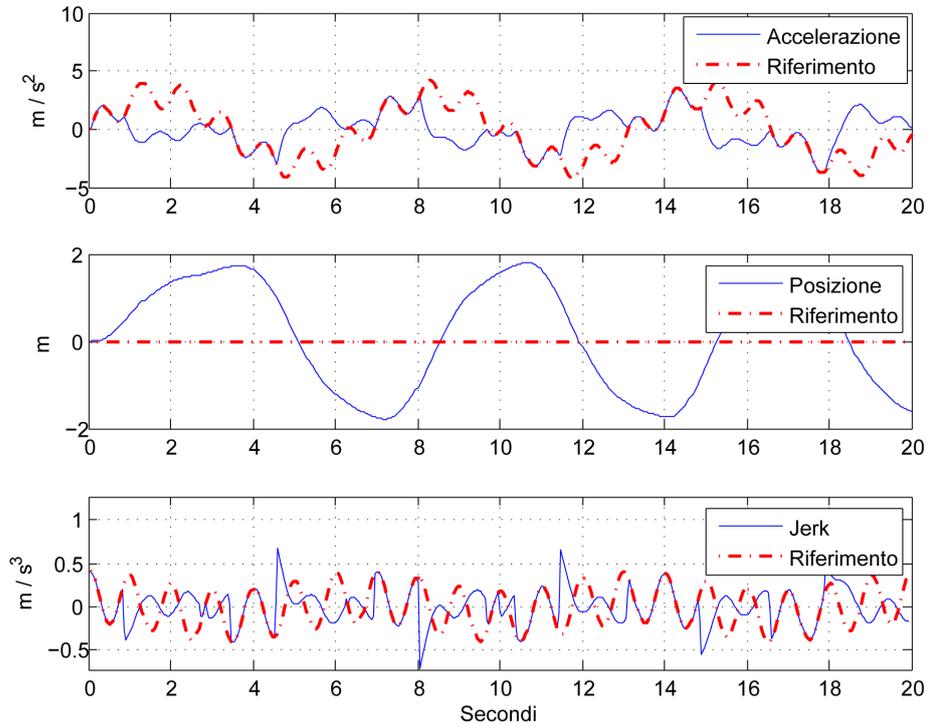


Figura 4.8: Controllo MPC del sistema con vincolo in posizione a  $2m$ , peso sulla posizione  $0.01$  e  $25$  passi di orizzonte di predizione

lineari che la piattaforma sta effettuando.

In figura 4.8 è anche presente l'andamento del jerk, altro parametro importante per valutare la bontà del *motion cueing* [15] analizzando le variazioni in accelerazione delle nostre risposte, evidenziando se il riferimento è in fase o meno con l'uscita. Nello specifico caso analizzato, il jerk spesso è di segno opposto al riferimento eccetto alcuni istanti in cui i limiti non ostacolano l'inseguimento.

#### 4.3.4 Caso vincolato, limiti in posizione e variazione dell'orizzonte di predizione

Al contrario delle simulazioni effettuate usando vincoli in ingresso e accelerazione, come evidenziato in figura 4.9, la variazione dell'orizzonte di predizione con limiti in posizione è un parametro del controllo che si è rivelato fondamentale. Con  $5$  passi di predizione ( $0.2s$  in avanti con tempo di campionamento pari a  $0.04s$ ), rispetto al grafico 4.8 il controllo risulta

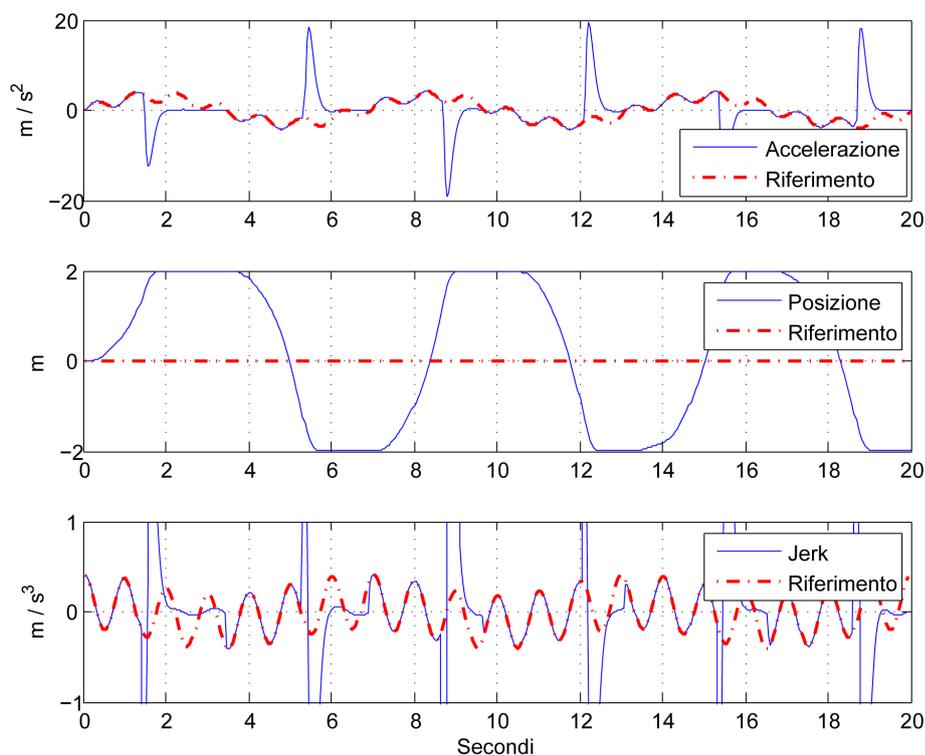


Figura 4.9: Controllo MPC del sistema con vincolo in posizione a  $2m$ , peso sulla posizione  $0.01$  e  $5$  passi di orizzonte di predizione

completamente inefficiente presentando dei picchi elevati in controfase nell'inseguimento dell'accelerazione, che porterebbero ad avere un *motion cueing* pessimo. Per avere piccoli errori in uscita, quindi, è fondamentale avere a disposizione una parte significativa del riferimento in avanti. Con una variazione dei pesi si sarebbe potuto tenere più limitata la posizione ed in questo modo il controllo non sarebbe costretto ad una manovra improvvisa come quella presente nel grafico 4.9.

#### 4.3.5 Caso vincolato, limiti in posizione e look-ahead

Fino ad ora, il controllo effettua la predizione utilizzando un riferimento futuro che è costante lungo tutto l'orizzonte di predizione, pari al valore della traiettoria al passo  $k+1$ -esimo. Questo perchè non sempre in una sessione simulativa sarà possibile conoscere le future manovre. Nel caso in cui però si volesse effettuare una simulazione con una traiettoria ben

determinata, quindi senza che il pilota possa comandare la vettura, sarà a disposizione l'intero riferimento e in questo caso è riduttivo pensare di utilizzare solamente il valore del passo successivo.

Per questo si è implementata una tecnica look-ahead la quale varia il riferimento lungo tutto l'orizzonte di predizione facendo in modo che il controllo sia più efficiente come rappresentato in figura 4.9. L'MPC, avendo a

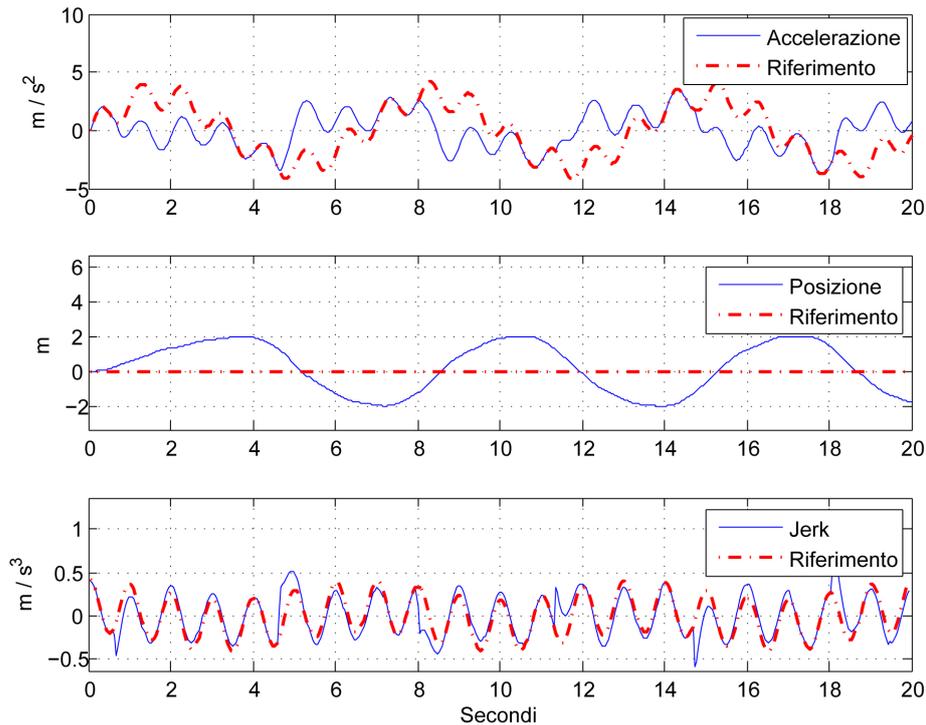


Figura 4.10: Controllo MPC del sistema con vincolo in posizione a  $2m$ , peso sulla posizione  $0.01$ , 25 passi di orizzonte di predizione e look-ahead

disposizione la traiettoria per il prossimo secondo, riesce ad effettuare l'ottimizzazione tenendo conto delle variazioni del riferimento lungo la predizione. In questo modo prevede che si andrà incontro ai limiti di posizione, e quindi la miglior cosa da fare sarà seguire il riferimento in alta frequenza come mostra il grafico dell'accelerazione. L'ipotesi formulata è confermata guardando l'andamento del 3° grafico: rispetto al caso senza look-ahead 4.8 il jerk riesce a restare nella maggior parte dei casi in fase con il riferimento, non facendo percepire al pilota falsi stimoli. Un ulteriore miglioramento viene effettuato impostando il controllo con orizzonte di predizione di  $2s$ .

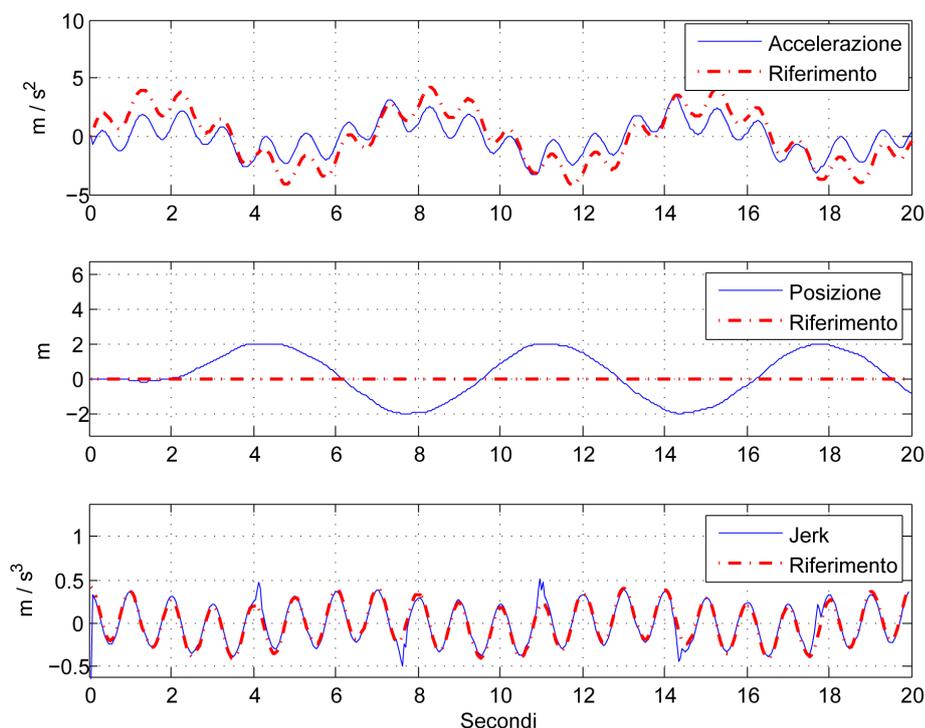


Figura 4.11: Controllo MPC del sistema con vincolo in posizione a  $2m$ , peso sulla posizione  $0.01$ ,  $50$  passi di orizzonte di predizione e look-ahead

L'azione predittiva raddoppiata rispetto al caso precedente fa in modo che l'errore per il jerk sia quasi sempre nullo e l'accelerazione sia inseguita in modo ottimo come rappresentato in figura 4.11.

#### 4.3.6 Caso vincolato, limiti in posizione, look-ahead e vincolo in soglia di percezione

Nonostante il jerk sia quasi perfettamente inseguito nel controllo con look-ahead con  $2s$  di predizione, l'accelerazione in uscita nel grafico 4.11 durante i primi  $2s$ , presenta alcuni punti in cui è di segno opposto rispetto il riferimento. Non è ancora chiaro dalla letteratura esistente se ciò, nonostante la variazione di accelerazione sia coerente con la traiettoria, possa produrre percezioni contrastanti con l'informazione trasmessa dall'apparato visivo. Ad ogni modo, il controllo MPC noto il modello percettivo, dà la possibilità di inserire un vincolo di soglia in accelerazione, in modo tale che accelerazioni di segno opposto al riferimento non vengano riprodotte ma restino

dentro certi limiti entro i quali il pilota non riesce a sentire il movimento lineare. In pratica quando il riferimento è positivo viene inserito un limite

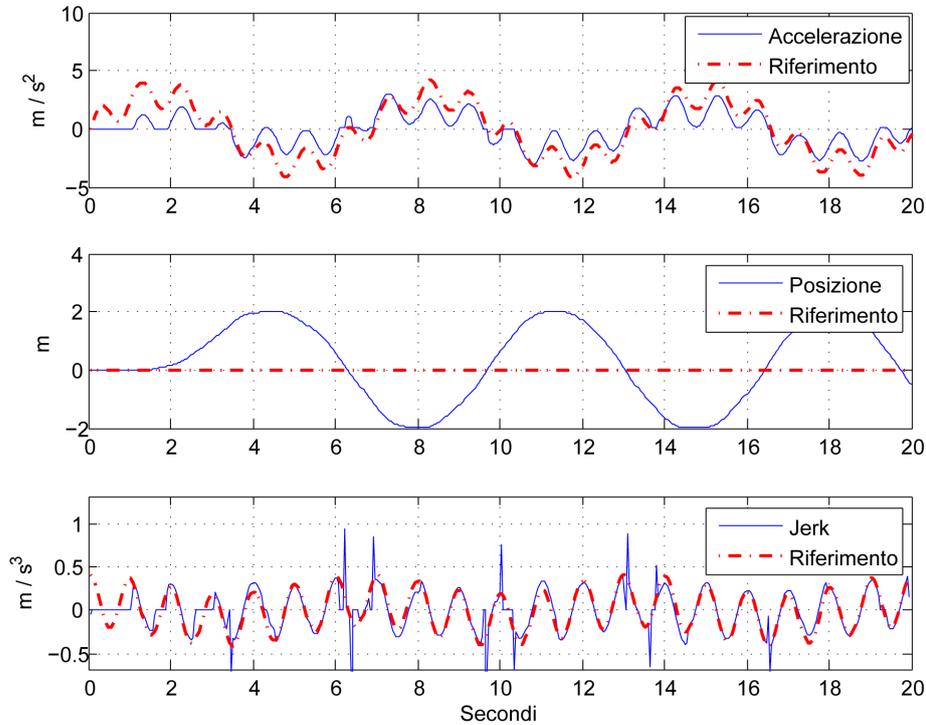


Figura 4.12: Controllo MPC del sistema con vincolo in posizione a 2, peso sulla posizione 0.01, 50 passi di orizzonte di predizione, look-ahead e soglia di percezione a  $0.05m/s^2$

in accelerazione a  $-0.05m/s^2$  come evidenziato dal grafico 4.12.

Quindi si avrà un inserimento nell’algoritmo di controllo di vincoli “riferimento-varianti”, opzione che il Toolbox di MATLAB non ci avrebbe permesso di fare. In funzione del riferimento positivo o negativo, si imposta il limite di soglia rispettivamente con segno opposto pari a  $0.05m/s^2$ . Ovviamente l’ulteriore vincolo porta il sistema ad essere sempre più complesso durante l’ottimizzazione e per questo estendere l’orizzonte di predizione a 50 passi è una scelta necessaria per non avere risposte inadeguate come nel caso di 25 passi.



## Capitolo 5

# Simulazioni e risultati

Nel lavoro di tesi non avendo a disposizione un modello che rappresenti il ritardo meccanico della piattaforma VI-Grade, il quale dovrà essere ricavato in futuro facendo uso di tecniche di identificazione di modelli per sistemi lineari, utilizzeremo un sistema molto più semplice massa-molla-smorzatore, simile a quello usato nel capitolo precedente .

Il modello in questione dovrà comunque far evidenziare come il controllo MPC utilizzi in modo del tutto naturale la *tilt-coordination* come strategia per il *motion cueing*, ovvero dovrà essere implicito che la piattaforma in caso di vincoli longitudinali, accelerazioni lineari verranno fatte percepire attraverso un'inclinazione opportuna della base.

### 5.1 Singolo asse disaccoppiato

Partendo da una singola movimentazione, ad esempio quella longitudinale lungo l'asse  $x$ , si utilizza il modello visto nel capitolo precedente descritto dalla trasformata di Laplace (5.1)

$$\frac{1}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad (5.1)$$

con valori  $\zeta = 0.25$  e  $\omega_0 = 25$ .

Per quanto riguarda il riferimento, si ha avuto modo di utilizzare, grazie ai dati forniti da VI-Grade, una telemetria acquisita attraverso una simulazione con un driver virtuale lungo il circuito di *Silverstone*. Si è scelto quindi di estrapolarne 30s significativi (1500 passi di campionamento con  $T_s = 0.02$ ) in modo tale da analizzare le prestazioni del controllo su una parte di circuito.

I dati in possesso, avendo un ordine di grandezza troppo elevato per le prestazioni che il simulatore VI-Grade andrà a riprodurre (decelerazioni longitudinali fino a  $25m/s^2$  e accelerazioni di  $12m/s^2$ ), si è scelto di ridurli applicando un filtro di *Butterworth* passa-alto del secondo ordine con frequenza di taglio  $0.001Hz$ . Il riferimento calcolato sarà ulteriormente scalato di un certo valore (nel nostro caso 7.5) come rappresentato in figura 5.1. Si va quindi ad effettuare la simulazione del sistema controllato attraverso

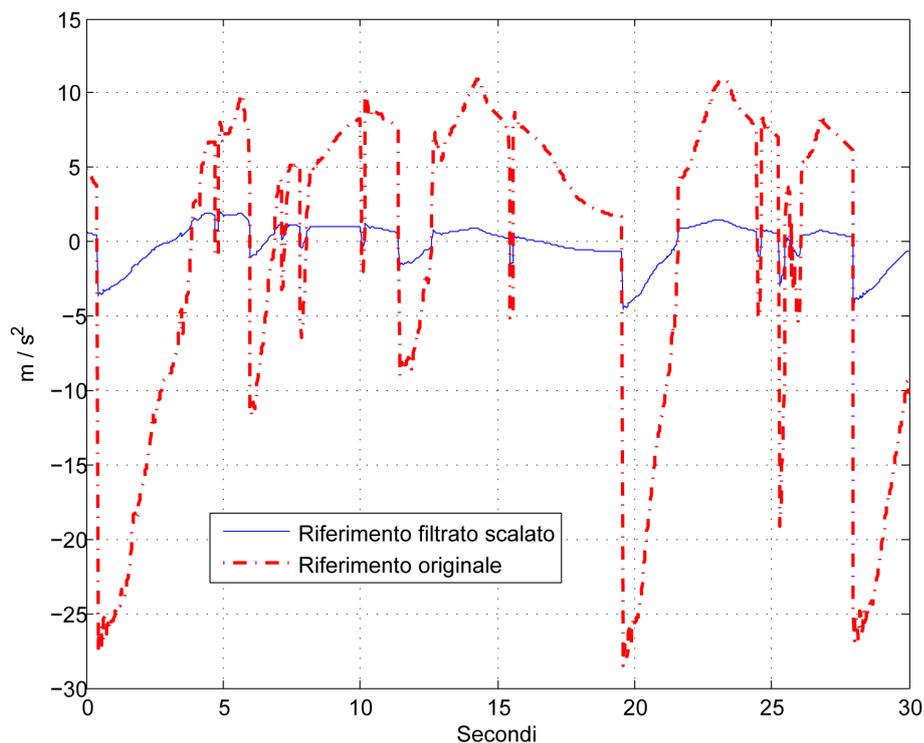


Figura 5.1: Riferimento longitudinale da telemetria originale e filtrato

l'MPC implementato, inseguendo una traiettoria filtrata e scalata con un solo vincolo in posizione corrispondente a  $0.5m$  per parte nella piattaforma, utilizzando un peso per l'accelerazione unitario e peso 0.01 per la posizione, una strategia look-ahead e 50 passi di predizione risultando quest'ultimi sufficienti per un corretto controllo.

In figura 5.2 l'uscita insegue il riferimento senza errori quando possibile avendo un limite in posizione piuttosto stretto; anche se spesso l'accelerazione è di segno opposto rispetto la traiettoria desiderata il jerk risultante è quasi sempre coerente. Per far sì che l'accelerazione non abbia questo

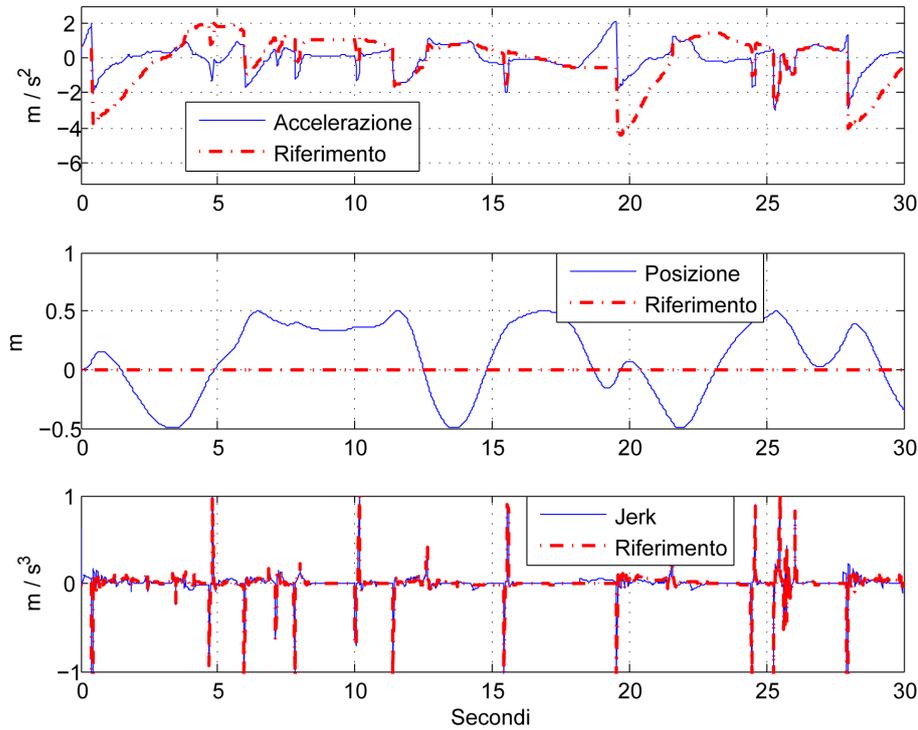


Figura 5.2: Controllo MPC del sistema con vincolo in posizione a 0.5, 50 passi di orizzonte di predizione, con look-ahead, senza soglia di percezione

genere di transitori si imposta nel controllo una soglia a  $\pm 0.05 m/s^2$  come appare in figura 5.3. Per l'inserimento del limite soglia, si è dovuto estendere l'orizzonte di predizione a 100 passi dato che con 1s di riferimento in avanti, non si riesce ad avere una risposta accettabile, presentando continui picchi in accelerazione portando la posizione dentro i limiti applicati.

Sempre in figura 5.3, il controllo fa inseguire il riferimento finchè sono soddisfatti i vincoli ma qualora il controllo voglia far assumere all'accelerazione un valore con segno opposto al riferimento, per compensare il limite in posizione, l'uscita satura al valore di soglia imposto. Il jerk in questa situazione, presenta dei picchi dovuti al cambiamento di accelerazione improvviso, non presenti nel riferimento.

Dunque per quanto la predizione possa essere estesa, il sistema non riuscirà mai a riprodurre l'intera accelerazione o frenata che il pilota dovrebbe percepire.

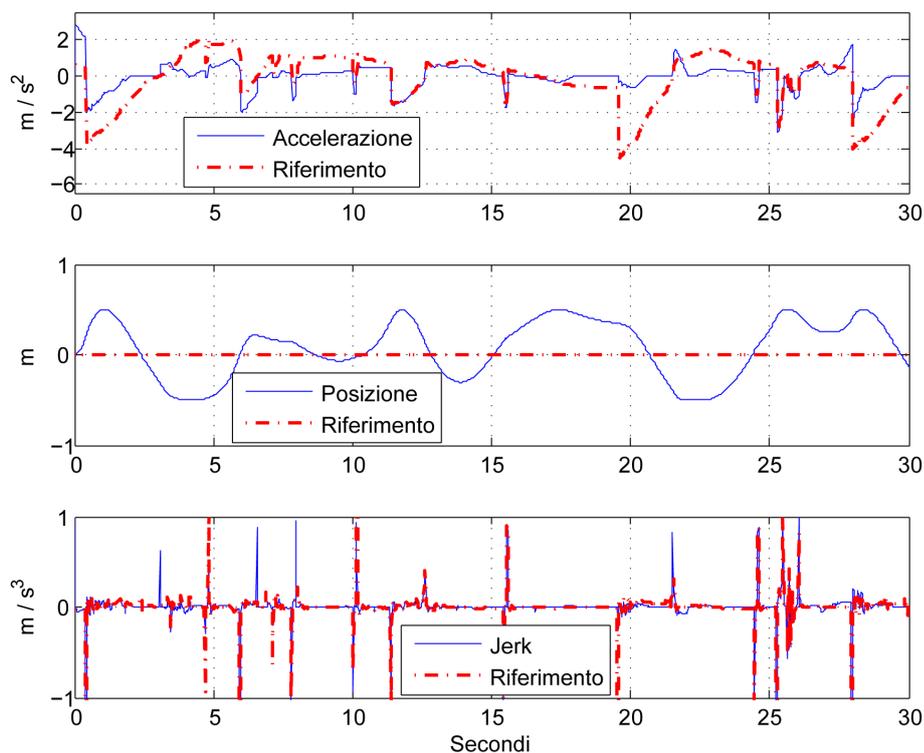


Figura 5.3: Controllo MPC del sistema con vincolo in posizione a  $0.5m$ , 100 passi di orizzonte di predizione, con look-ahead, con soglia in percezione

## 5.2 Due assi accoppiati e tilt-coordination

Si è visto che l'utilizzo di una singola movimentazione per riprodurre ad esempio un'accelerazione longitudinale avrà notevoli limiti, nonostante la predizione futura aiuti di molto il controllo. Attraverso l'inclinazione della postazione di un opportuno angolo, si trasmetterà al pilota la percezione di un'accelerazione che andrà a sommarsi con quella fornitagli longitudinalmente. Questa strategia chiamata *tilt-coordination* è un modo di compensare i vincoli spaziali della piattaforma, aggiungendo alla forza sviluppata dall'accelerazione longitudinale, ad esempio lungo l'asse  $x$ , una componente della forza gravitazionale data dal pilota ruotato di un angolo di *pitch* come descritto in figura 5.4.

Nello specifico si vede sulla sinistra come in condizioni di equilibrio, ovvero con la postazione pilota parallela all'asse orizzontale, le forze che intervengono lungo tale asse sono l'accelerazione longitudinale che il pilota

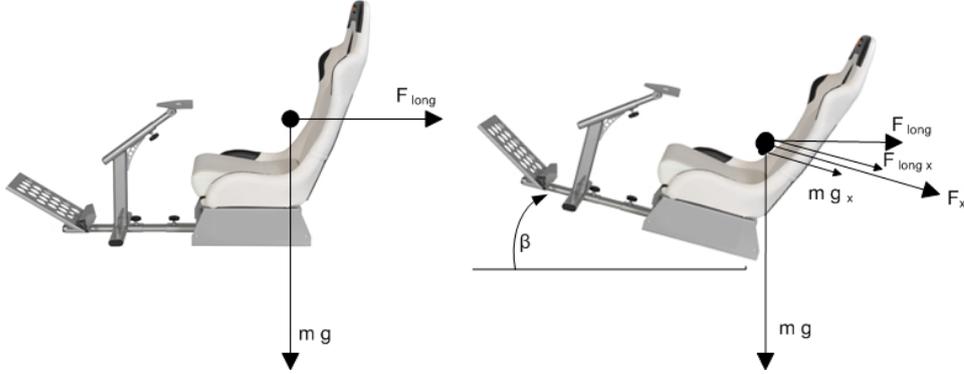


Figura 5.4: Forze incidenti nella rotazione del sedile

percepisce ( $F_{long}$ ) mentre la forza peso che la massa del pilota imprime sulla base del sedile, essendo perpendicolare a  $F_{long}$ , non si somma su di essa.

Quando la sedia, invece, viene ruotata con un angolo di pitch  $\beta$ , le forze  $m \cdot g$  e  $F_{long}$  che prima agivano nel nostro sistema, non essendo più parallele al riferimento cartesiano coerente con la postazione pilota, si suddividono nelle componenti  $x$  e  $y$ . Per il nostro scopo si va ad analizzare quelle lungo l'asse  $x$  per angoli  $\beta$  sufficientemente piccoli:

$$F_x = m \cdot g_x + F_{long\ x}$$

con

$$\begin{aligned} m \cdot g_x &= m \cdot g \cdot \sin \beta \approx m \cdot g \cdot \beta \\ F_{long\ x} &= m \cdot a \cdot \cos \beta \approx m \cdot a. \end{aligned}$$

Per usufruire di tale strategia si dovrà andare a modificare le matrici del modello (4.6), inserendo un ulteriore grado di libertà al sistema per comandare la rotazione del *pitch* e in modo che si restituisca in uscita:

- l'accelerazione longitudinale percepita  $\hat{a}_l$ , somma dell'accelerazione longitudinale e dell'angolo di rotazione  $\beta$  per l'accelerazione gravitazionale;
- l'accelerazione angolare percepita  $\hat{a}_a$ ;
- l'accelerazione longitudinale  $a_l$ ;
- l'accelerazione angolare  $a_a$ ;

- posizione longitudinale  $p_l$ ;
- posizione angolare  $p_a$ .

Le accelerazioni che saranno inseguite con il *motion cueing* sono quelle percepite. Nello specifico, non avendo introdotto nessun modello percettivo ma solamente descritto la struttura geometrica delle forze per introdurre un primo banale accoppiamento, l'accelerazione angolare percepita sarà uguale a quella effettiva. Come per il caso singolo asse, il riferimento per l'accelerazione longitudinale percepita ( $x$ ) sarà quello rappresentato in figura 5.1. Anche per l'accelerazione angolare si è scelto di far seguire una traiettoria estratta dalla telemetria (*pitch*) lungo il circuito di Silverstone, in modo tale che la rotazione della piattaforma non sia svincolata dall'accelerazione longitudinale. Per le posizioni e l'accelerazione lungo  $x$  effettiva invece il riferimento sarà nullo.

L'evoluzione dello stato per l'accelerazione longitudinale e angolare avviene in modo indipendente, ovvero si potrà scegliere per una o per l'altra una funzione di trasferimento che meglio caratterizza le movimentazioni lungo un DOF. Nel caso studiato, queste saranno rappresentate attraverso la funzione di trasferimento (5.1) con  $\omega_0 = 25$  e  $\eta = 0.25$ , e il sistema finale sarà dunque espresso in spazio di stato attraverso le matrici:

$$A = \begin{bmatrix} -12.5 & -625 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -12.5 & -625 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 625 & 0 & 0 & 0 & 0 & 0 & 6125 \\ 0 & 0 & 0 & 0 & 0 & 625 & 0 & 0 \\ 0 & 625 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 625 & 0 & 0 \\ 0 & 0 & 0 & 625 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 625 & 0 \end{bmatrix}$$

Il grafico 5.5 raffigura 30s di simulazione del sistema appena descritto

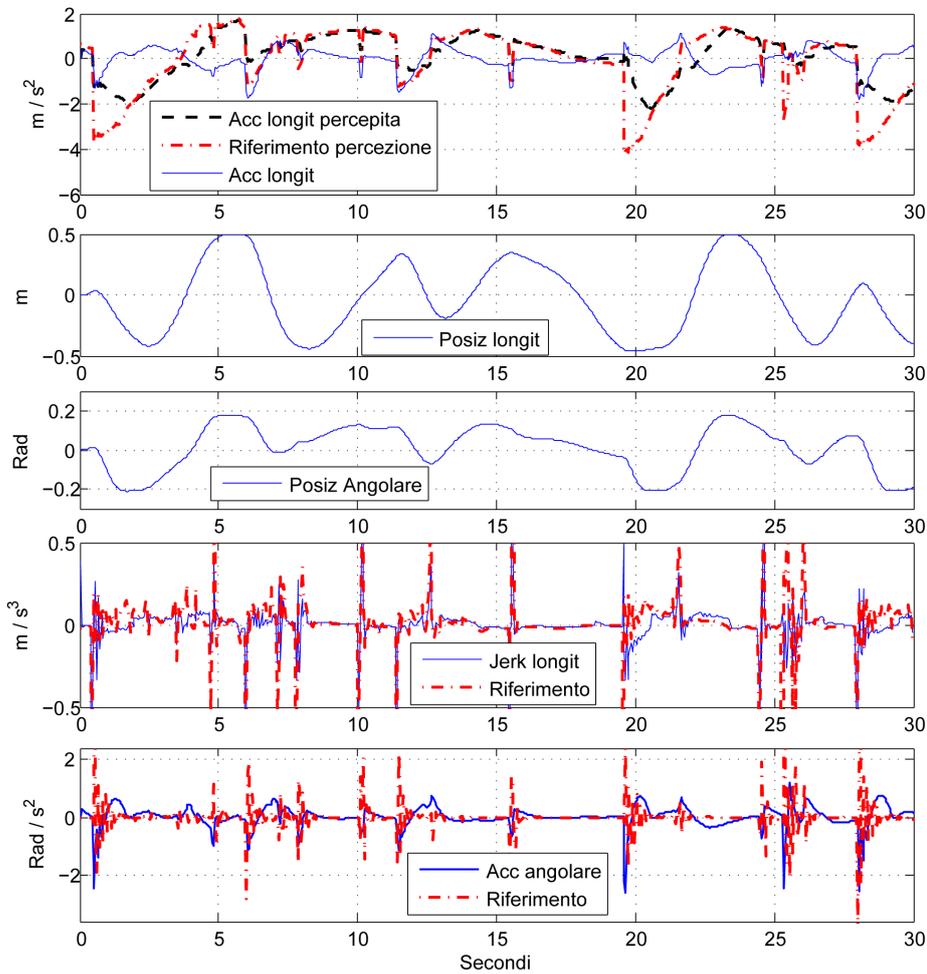


Figura 5.5: Accelerazioni, posizioni e jerk del sistema controllato dall'MPC con vincoli, 50 passi di orizzonte di predizione, senza look-ahead, senza soglia di percezione

controllato utilizzando MPC senza look-ahead e senza limite di soglia di percezione. Per un corretto inseguimento della traiettoria in accelerazione si è scelto di parametrizzare il controllo, in questa simulazione e nelle successive, con la seguente matrice  $Q$  dei pesi per le uscite nella funzione

costo

$$Q = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad (5.2)$$

dove il peso 5 sarà per  $\hat{a}_l$ , 4 per  $\hat{a}_a$  e così via per tutte le uscite. I pesi sono stati assegnati cercando di minimizzare l'errore rispetto il riferimento in accelerazione desiderata lasciando che le posizioni, sia lungo  $x$  che per il pitch, possano muoversi senza essere vincolate al riferimento. Inoltre per evidenziare le prestazioni del controllo MPC come algoritmo di *motion cueing* si impostano i seguenti limiti:

- limite in accelerazione longitudinale a  $\pm 3.6m/s^2$ ;
- limite in posizione longitudinale a  $\pm 0.5m$ ;
- limite in posizione angolare a  $\pm 0.21rad$ ;
- limite facoltativo per l'accelerazione di soglia di  $\pm 0.05m/s^2$  a seconda che il riferimento sia rispettivamente negativo o positivo.

Nel caso di figura 5.5 con i vincoli imposti, 50 passi di predizione (1s di predizione) e controllo senza look-ahead il sistema insegue in modo insoddisfacente la traiettoria del riferimento di percezione, evidenziato anche dal grafico del jerk che in corrispondenza dei primi 5s e tra i secondi 20 e 25, presenta valori che sono di segno opposto rispetto il riferimento; inoltre anche il grafico dell'accelerazione angolare presenta parecchi errori nell'inseguimento, avendo una risposta del tutto incoerente con la traiettoria impostata. In figura 5.6 è rappresentata invece la stessa simulazione utilizzando una strategia look-ahead per l'inseguimento. Con questa configurazione il sistema controllato in catena chiusa, entro i limiti imposti, risulta avere un'accelerazione longitudinale percepita che segue l'andamento del riferimento, eccetto negli istanti in cui non è possibile avere intensità così grandi. In generale il controllo è ottimo e guardando la risposta del jerk non appaiono quasi mai punti in cui si è distanti dal riferimento o in opposizione di segno. Alcuni errori si notano nell'accelerazione angolare, la quale rispetto il riferimento avrà una componente aggiuntiva dovuta al *tilt-coordination*; rispetto la simulazione

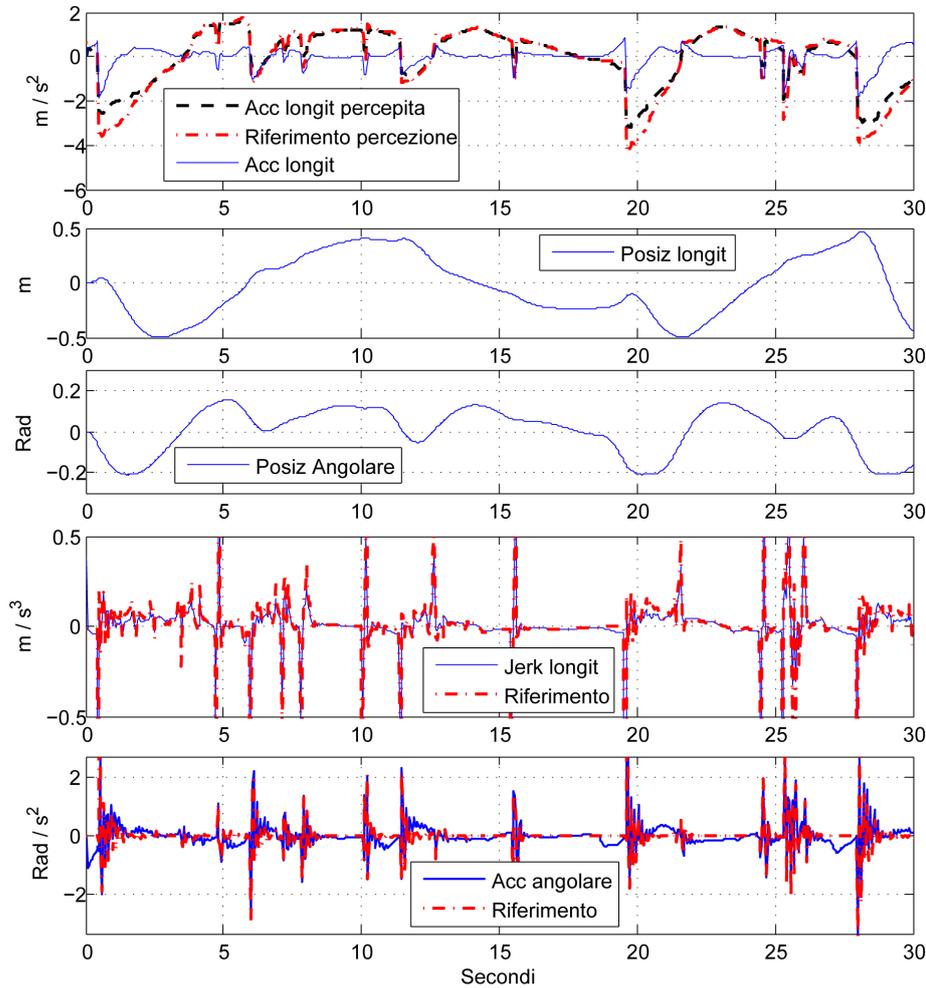


Figura 5.6: Accelerazioni, posizioni e jerk del sistema controllato dall'MPC con vincoli, 50 passi di orizzonte di predizione, con look-ahead, senza soglia di percezione

in figura 5.5, questa volta il controllo riesce a far inseguire la traiettoria prelevata dalla telemetria grazie all'uso di una strategia look-ahead che la quale si riesce a miglioroare la risposta.

Per ultimo si fa notare come con questa configurazione e con questa traiettoria il controllo non porta l'accelerazione ad essere di segno opposto al riferimento; l'utilizzo di un ulteriore vincolo nella soglia di percezione in accelerazione sarebbe totalmente inutile.

Per evidenziare l'utilità della soglia percettiva si va ad impostare una

scalatura nel riferimento, inferiore al caso precedente (2.5), in modo tale da avere accelerazioni più elevate e si andrà ad effettuare la simulazione senza limite in soglia adottando sempre una strategia look-ahead come mostrato figura 5.7. Si è scelto di effettuare uno zoom attorno ai 17 – 24s per evi-

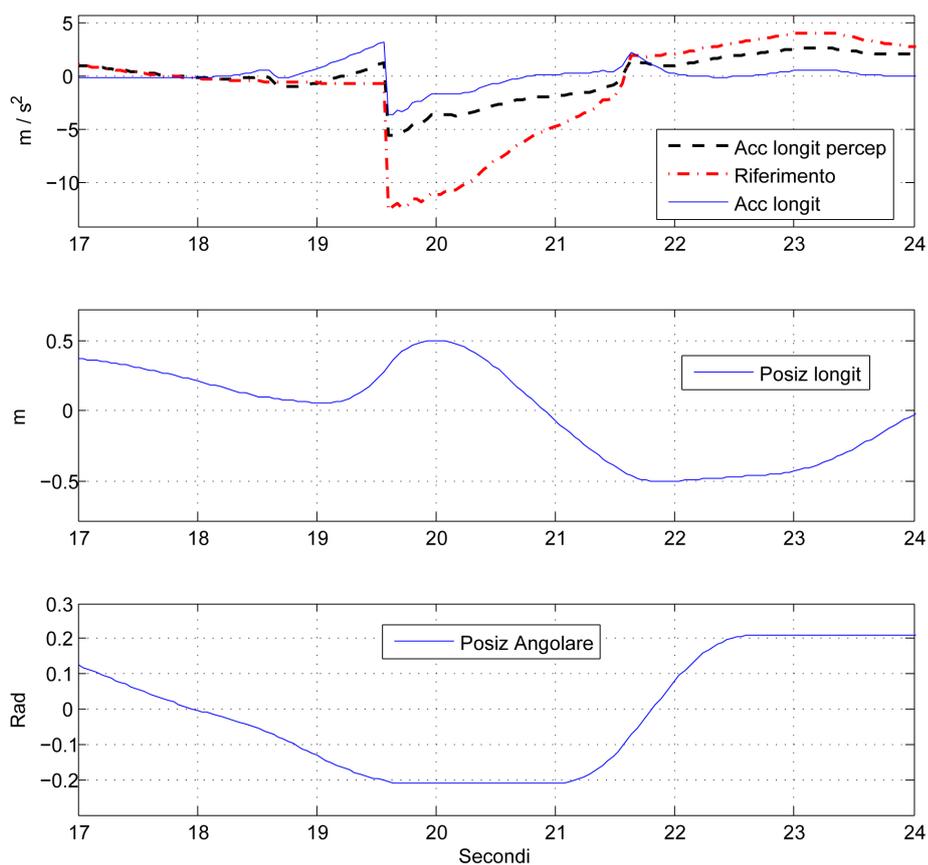


Figura 5.7: Accelerazioni e posizioni del sistema controllato dall'MPC con vincoli, 100 passi di orizzonte di predizione (2s), con look-ahead, senza soglia di percezione

denziare in corrispondenza dei 19.5s la presenza di una brusca frenata pari a  $17m/s^2$  che il controllo non riuscirà mai ad inseguire perfettamente; noti i suoi limiti andrà a gestire al meglio la situazione grazie alla predizione di 2s in avanti e all'ottimizzazione della funzione costo. Infatti dal grafico delle posizioni si vede come, prima della brusca decelerazione il sistema porti la posizione longitudinale al suo margine ( $0.5m$ ) per poi arrivare a quello opposto ( $-0.5m$ ) in poco meno di due secondi a causa del valore di acceler-

azione assunto pari al vincolo imposto di  $-3.6m/s^2$ . Per garantirci inoltre una percezione massima dell'accelerazione anche la posizione angolare ha assunto il suo valore limite di  $-0.21rad$  pari a circa  $12^\circ$ . Anche se il con-

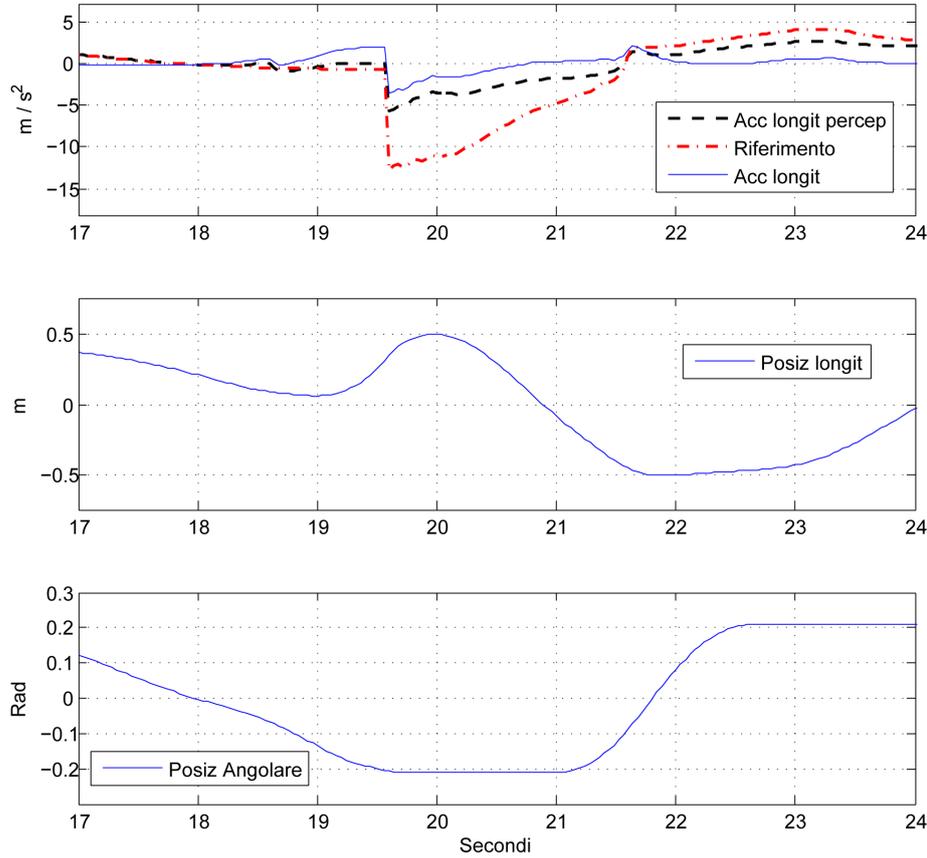


Figura 5.8: Accelerazioni e posizioni del sistema controllato dall'MPC con vincoli, 100 passi di orizzonte di predizione (2s), con look-ahead, con soglia di percezione

trollo MPC ci ha portato all'utilizzo in modo ottimo dello spazio di lavoro a disposizione (grafico 5.7), un sistema di *motion cueing* deve garantirci che l'accelerazione percepita non sia mai di segno opposto a quella reale. Per far ciò si è inserito un limite soglia nella accelerazione percepita in funzione del segno del riferimento. La situazione descritta precedentemente sussiste ancora, ma come rappresentato in figura 5.8 questa volta il sistema raggiunge la posizione massima spostandosi con un'accelerazione pari al valore di soglia inserito, in modo tale da non avere movimenti in contrasto con l'informazione visiva. Infine in figura 5.9 si ha il grafico lungo i 30s di simu-

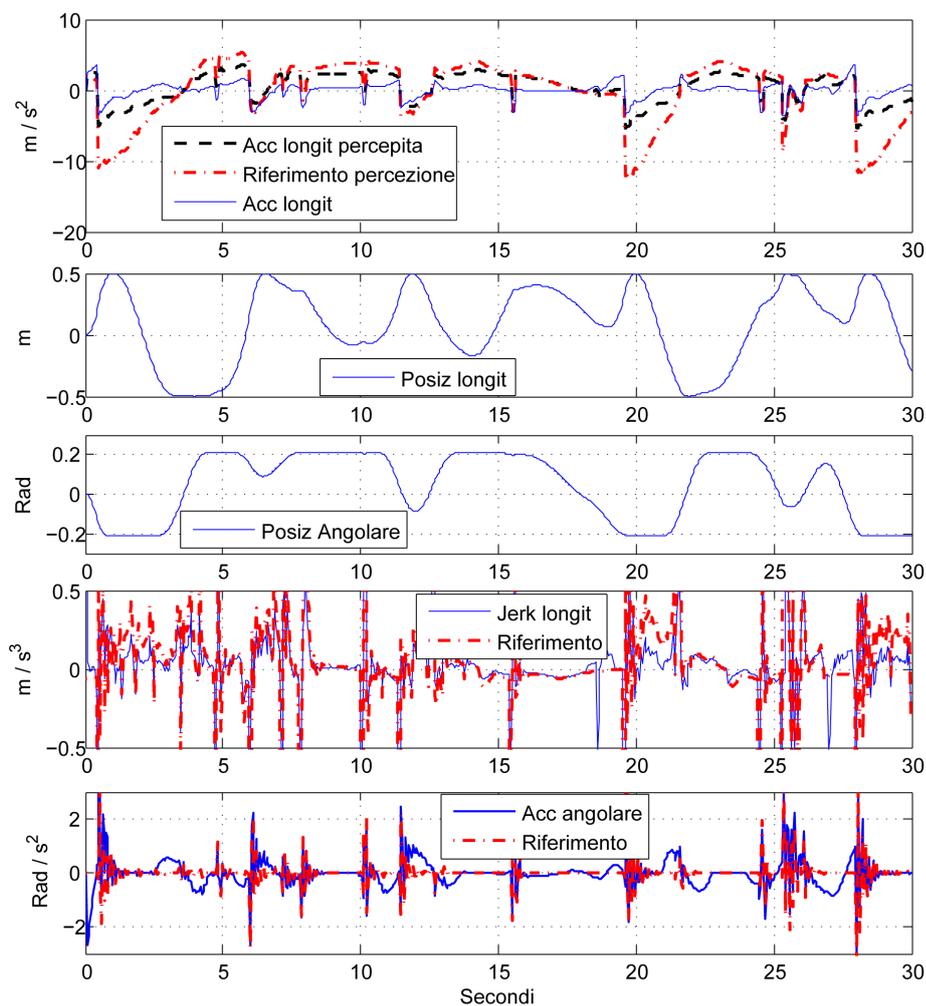


Figura 5.9: Accelerazioni, posizioni e jerk del sistema controllato dall'MPC con vincoli, 100 passi di orizzonte di predizione (2s), con look-ahead, con soglia di percezione

lazione con il riferimento scalato di 2.5, controllo con strategia look-ahead e vincolo in soglia. L'accelerazione percepita segue in alta-frequenza il riferimento, dovendo soddisfare i vincoli imposti, ma nonostante questo il jerk risulta inseguito correttamente eccetto casi in cui l'accelerazione deve distaccarsi dalla sua traiettoria per portarsi alla soglia di percezione prima di effettuare delle brusche manovre utilizzando l'intero spazio lungo  $x$  a disposizione. Inoltre il riferimento in accelerazione angolare, rispetto al caso con riferimento scalato di 7.5, presenta degli errori piuttosto grandi, anche di

qualche  $rad/s^2$ , dovuto al fatto che l'MPC cerca di compensare più che può la differenza di accelerazione longitudinale utilizzando la *tilt-coordination*.



## Capitolo 6

# Conclusioni

Scopo del lavoro di tesi era verificare se la possibilità di usare un controllo MPC per algoritmi di *motion cueing* è una scelta che porta a gestire in modo migliore i movimenti del simulatore. Prima dell'approccio al controllo predittivo gli algoritmi di MCS erano basati sul filtraggio del segnale di riferimento, impostando i parametri del filtro con metodi empirici fondati sulle sensazioni del pilota durante una sessione simulativa.

L'applicazione di un controllore di tipo MPC mira ad introdurre un approccio metodologico nel campo dei sistemi di motion cueing. La presenza di un modello percettivo mette poi le basi per uno sviluppo futuro nel quale il sistema uomo-piattaforma può essere separato ed analizzato distintamente nelle sue due componenti. La definizione però di un modello percettivo nello specifico non è stato sviluppato come non si ha avuto modo avere le equazioni che caratterizzano la piattaforma Vi-Grade.

Ciò nonostante, l'utilizzo di una funzione di trasferimento del secondo ordine a due gradi di libertà e un accoppiamento percettivo banale, ci ha portato a risultati che avremmo voluto trovare. Si ha dunque una base per valutare e impostare un algoritmo di *motion cueing* non più attraverso sensazioni soggettive, ma qualcosa di oggettivo che trova supporto grazie ad uno studio teorico di fondo.

Dai risultati ottenuti si ha avuto modo di vedere come il *tilt-coordination* effettivamente vada a riprodurre le basse frequenze della traiettoria e lascia alle accelerazioni longitudinali il compito di inseguire i comportamenti in alta frequenza. Una strategia come il *washout filter* è stata ottenuta in modo del tutto naturale con il controllo predittivo, facendo in modo che la posizione del pilota torni sempre al centro dello spazio di lavoro, a patto che

l'orizzonte di predizione riesca ad avere un riferimento di qualche secondo in avanti. Questo è possibile quando si sta effettuando una sessione simulativa offline, ma nel caso di guida "real-time" il riferimento non può essere predetto, eccetto qualche decina di millisecondi dovuti ai ritardi meccanici della piattaforma.

Lo studio di una strategia capace di andare ad *predirre* il riferimento futuro dall'esperienza di guida del specifico pilota sembra la via più corretta ma l'approfondimento di questo non è stato fatto durante la tesi. Anche la traiettoria che si è usata per la simulazione offline è stata filtrata e scalata per non addentrarsi troppo nello specifico ma è risaputo che entrambe le azioni hanno i propri svantaggi: da una parte il filtraggio di un segnale può portare un cambiamento di fase nel riferimento a causa dell'utilizzo di un filtro passa-alto dall'altra la scalatura della traiettoria agevola il controllo nella parti in cui si ha una frenata, ad esempio, di forte intensità ma porta a dar meno valore anche alle accelerazioni più piccole.

Molti aspetti devono essere affrontati ancora per avere un *motion cueing* completo funzionante per la piattaforma VI-Grade. Primo fra tutti deve essere sviluppato un modello che caratterizzi il ritardo meccanico della piattaforma attraverso identificazione di sistemi lineari; in questo modo la predizione delle uscite e l'ingresso di controllo ottimo possono fare la differenza nelle performance tra un MCS classico ed un MCS effettuato tramite MPC. Non meno importante sarà lo studio fisiologico di un modello di percezione che potrà aiutare notevolmente la gestione dei movimenti della piattaforma attraverso l'utilizzo di *tilt-coordination* e *washout filter*

# Bibliografia

- [1] Slob, J.J. (Jelmer), *State-of-the-Art Driving Simulators, a Literature Survey*. Eindhoven, 2008.
- [2] F. Colombet, M. Dagdelen, G. Reymond, *Motion Cueing: What is the impact on the driver's behavior?*. Monaco, 2008.
- [3] Gilles Reymond, Andras Kemeny, *Motion Cueing in the Renault Driving Simulator*. 2000.
- [4] P.J. Feenstra, M. Wentink, Z.C. Roza, *Desdemona, an alternative moving base design for driving simulation*. Soesterberg, 2007.
- [5] Bortoluzzo Ricky, *Studio e applicazione del Controllo Predittivo*.
- [6] M.M. Attallah, O. Rashwan, *Six DOF Hexapod - Challenge of Design and Innovation*. <http://biotsavart.tripod.com/hexapod.htm> Il Cairo.
- [7] Wiki, *Simulatori*. <http://it.wikipedia.org/wiki/Simulazione>.
- [8] Wiki, *Simulations*. <http://en.wikipedia.org/wiki/Simulation>.
- [9] M. Dagdelen, G. Reymond, A. Kemeny *Model-based predictive motion cueing strategy for vehicle driving simulators*. Guyancourt, 2009.
- [10] Liuping Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. Melbourne, 2009.
- [11] M. Wentink, W. Bles, R. Hosman, *Design & evaluation of spherical washout algorithm for Desdemona simulator*. Soesterberg, 2005.
- [12] Giovanni Chetta, *L'apparato vestibolare*. <http://www.giovannichetta.it/vestibolo.html>.

- [13] A. Ceselli, *Una nota sui metodi di tipo "Active Set" per programmazione quadratica.*
- [14] A. Bemporad, *Model Predictive Control Toolbox 3 - Users Guide.*
- [15] F. Soyka, H. J. Teufel, K.A. Beykirch, *Does jerk have to be considered in linear motion simulation?.* Tübingen, 2009
- [16] Jan Maciejowski, *Predictive Control with Constraints.*
- [17] L.D. Reid, M.A. Nahon, *Flight simulation motion-base drive algorithms: part 1 - developing and testing the equations.* Toronto, 1985
- [18] R.V. Parrish, J.E. Dieudonne, *Coordinated adaptive washout for motion simulators.* Journal of Aircraft, 1975
- [19] R. Sivan, J. Ish shalom, *An optimal control approach to the design of moving flight simulators.* IEEE International Conference on Systems, Man and Cybernetics, 1982

# Ringraziamenti

*Tantissime persone hanno contribuito nel sostenermi e aiutarmi in questi anni universitari.*

*Primi fra tutti i miei amici Teo, Ale (Sella), Ale (Rosso), Miglio e Anto (Non cito tutti gli altri nomi ma ovviamente ci siete anche voi!). Ringraziamenti doverosi dato che senza di loro e ai momenti passati felicemente probabilmente non sarei riuscito ad avere la forza morale per andar avanti.*

*I miei genitori, per avermi “sopportato” e sostenuto economicamente per tutti questi anni, per ogni capriccio che mi saltava in mente di fare.*

*I ragazzi/e della pizzeria Marco, Stefano, Paolo e Ale e tutti gli altri. Un gruppo speciale e affiatato il quale anch'esso ha contribuito alla crescita organizzativa ma anche nel creare diversificazioni nella vita di tutti i giorni.*

*Alla Giulia, per i bei momenti passati assieme nell'ultimo anno la quale nonostante gli alti e bassi mi tollera e spero lo farà ancora per molto tempo!*

*I compagni intramontabili di Ingegneria Berto, Gabri, Ponch, Giandu, Bovo e Damiano che fin dai primi anni mi han aiutato nel superamento degli esami grazie ai famosi “Gruppi studio”! Senza di loro non sarei qui.*

*Il mio correlatore Mattia Bruschetta che durante questo periodo di tesi mi ha sempre seguito togliendo tempo per le sue ricerche. Una splendida persona dal sapere immenso!! Grazie anche a Fabio Maran per le correzioni e consigli dati e il professor Beghi per il materiale fornitomi. Un ringraziamento anche a Diego Minen e David Bacchet per avermi offerto la possibilità di iniziare questo lavoro per VI-Grade.*

*Grazie a tutti.*

*Daniele D'Ambrosio*

*24/10/2010*