

POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale e dell'Informazione

Corso di laurea Magistrale in Ingegneria dell'Automazione



**STUDIO DI TECNICHE DISTRIBUITE DI
CONTROLLO PREDITTIVO STOCASTICO PER
L'INSEGUIMENTO DI SEGNALI DI
RIFERIMENTO E APPLICAZIONE AL
COORDINAMENTO DI ROBOT MOBILI**

Relatore: Prof. Marcello FARINA

Tesi di Laurea di:

Simone MISIANO Matr. 797893

Anno Accademico 2013-2014

Ai miei genitori...

Sommario

La ricerca nell'ambito del controllo predittivo è negli ultimi anni in forte sviluppo. In primo luogo si stanno affermando metodologie innovative che garantiscono proprietà di robustezza rispetto a disturbi, di varia natura, agenti sul sistema. Molte di queste tecniche che, per poter garantire le proprietà ricercate, presuppongono che i disturbi agenti siano limitati, risultano molto conservative e sono caratterizzate da complessità computazionali elevate. D'altra parte, lo sviluppo tecnologico di quest'ultimo periodo sta portando alla costruzione di impianti sempre più grandi e complessi. In questo ambito, la ricerca relativa a innovative tecniche di controllo si sta orientando verso metodologie di tipo distribuito che permettono di controllare in modo separato ogni componente del sistema a larga scala e di gestirne una eventuale cooperazione.

In questa Tesi si propone uno schema di controllo in grado di gestire sistemi affetti da disturbi stocastici additivi. In particolare, dapprima sarà descritto un algoritmo di tipo centralizzato per la regolazione dello stato di un sistema soggetto a un disturbo stocastico. Dopodiché tale soluzione verrà estesa ad un caso di inseguimento di segnali di riferimento. Inoltre, si propone una implementazione di tipo distribuito per controllare sistemi indipendenti tra loro ma connessi tramite vincoli di accoppiamento. Lo schema ottenuto sarà, infine, applicato ad un impianto sperimentale, e in particolare al coordinamento di una piccola flotta di robot mobili di tipo unicycle.

Abstract

In the recent years, research on Model Predictive Control is increasing rapidly. Innovative methodologies are emerging that guarantee robustness properties with respect to disturbances of various types acting on the system. Many of these techniques, which assume that disturbances are bounded, are very conservative and are characterized by high computational complexity. Also, technological innovation is leading to the development of increasingly large and complex plants. In this field, the research is moving towards distributed methodologies, capable of controlling each component of a large scale system separately, and of managing interactions.

This Thesis discusses a control scheme able to manage systems affected by stochastic additive disturbances. First, focus is given to description of a centralized algorithm for regulation in case of system subject to a stochastic disturbance. The solution is then extended both to the case of tracking reference signals and to the distributed framework. In particular, we consider the case of independent systems, coupled through constraints. The proposed approach is applied to an experimental set-up, including a small fleet of unicycle robots to be coordinated.

Indice

1	Introduzione	1
1.1	Sistemi affetti da disturbi stocastici	2
1.2	Problema del coordinamento e del controllo di sistemi indipendenti	5
1.3	Obiettivo della Tesi e contributi originali	7
1.4	Struttura della Tesi	8
2	Controllo predittivo stocastico	11
2.1	Introduzione al controllo MPC	11
2.2	MPC per sistemi lineari	13
2.2.1	Scelta dei parametri per la stabilità	14
2.3	Controllo MPC Stocastico per la regolazione	16
2.3.1	Formalizzazione dei vincoli	17
2.3.2	Struttura del controllore	20
2.3.3	Riformulazione deterministica dei vincoli	22
2.3.4	Formulazione di MPC stocastico per la regolazione	30
3	Estensione al problema di inseguimento in un contesto distribuito	33
3.1	Controllo predittivo stocastico per l'inseguimento	34
3.1.1	Legge ausiliaria di controllo e calcolo del set invariante per il <i>tracking</i>	36
3.1.2	Formulazione di MPC stocastico per il <i>tracking</i>	37

3.2	Controllo predittivo distribuito stocastico per il <i>tracking</i> . . .	39
3.2.1	Vincoli Collettivi	40
3.2.2	Riformulazione del <i>set</i> invariante per il <i>tracking</i> . . .	42
4	Apparato Sperimentale	49
4.1	Modello del Sistema	50
4.2	Caratterizzazione del disturbo	55
5	Caso di studio: Formulazione ed implementazione di un problema di navigazione autonoma	65
5.1	Obstacle Avoidance	66
5.2	Collision Avoidance	71
5.3	Stima adattativa della varianza del disturbo	75
6	Simulazioni	77
6.1	Inseguimento di traiettoria	79
6.2	Obstacle e Collision Avoidance	81
6.3	Violazione dei vincoli	86
7	Risultati Sperimentali	91
7.1	Obstacle e Collision Avoidance	92
7.2	Prove sulla stima adattativa	100
8	Conclusioni e sviluppi futuri	105
A	Implementazione	109
	Bibliografia	123

Elenco delle figure

1.1	Controllo di temperatura di un edificio con MPC	3
1.2	Smart Grid	5
1.3	Cruise Control	7
2.1	MPC con Receding Horizon	12
2.2	Confronto tra $f(p)$ calcolata mediante la disuguaglianza di Cantelli e quella calcolata supponendo una distribuzione normale	25
3.1	Gestione dei vincoli di accoppiamento in contesto distribuito	43
3.2	Partizione arbitraria dello spazio	44
4.1	Robot <i>e-puck</i>	50
4.2	modello ad unicycle	51
4.3	Schema di controllo con <i>Feedback linearization</i>	53
4.4	Rumore agente sullo stato	59
4.5	Rumore agente sullo stato trascurando i campioni che generano i picchi	60
4.6	Funzioni di densità di probabilità	62
4.7	Test di Anderson	63
5.1	Selezione del vincolo maggiormente violato	69
5.2	Selezione del vincolo maggiormente violato	73

6.1	Raggiungimento di un obiettivo e inseguimento di traiettorie di riferimento	80
6.2	Accelerazioni e velocità lungo gli assi nel caso di inseguimento di un agente verso un goal in un piano libero da collisioni . .	81
6.3	<i>Obstacle Avoidance</i> : confronto sulla scelta dell'orizzonte di predizione	83
6.4	<i>Obstacle Avoidance</i> : confronto sulla scelta del numero di lati del politopo	83
6.5	Inseguimento di una traiettoria ad otto in presenza di ostacoli e vincoli sullo spazio di lavoro	84
6.6	<i>Collision Avoidance</i> tra due agenti	85
6.7	<i>Collision</i> e <i>Obstacle Avoidance</i> tra due agenti	85
6.8	<i>Collision Avoidance</i> : confronto sulla scelta di N nel caso di simmetria del sistema	86
6.9	violazioni del vincolo con probabilità del 10%	87
6.10	violazione del vincolo con probabilità del 20%	87
6.11	violazione del vincolo con probabilità del 30%	88
6.12	Traiettorie dello stato in 100 simulazioni	89
6.13	Distribuzione normale delle traiettorie dello stato	90
7.1	Inseguimento di una traiettoria sinusoidale in presenza di ostacoli	92
7.2	Navigazione con <i>Obstacle Avoidance</i> : traiettoria del robot <i>e-puck</i>	93
7.3	Navigazione di un <i>e-puck</i> verso un obiettivo con <i>obstacle avoidance</i>	94
7.4	Navigazione con <i>Obstacle Avoidance</i> : tempo di calcolo . . .	95
7.5	Navigazione con <i>Obstacle Avoidance</i> : velocità	95
7.6	Traiettorie: <i>Collision Avoidance</i> tra due robot <i>e-puck</i>	96
7.7	<i>Collision Avoidance</i> tra due robot <i>e-puck</i>	96
7.8	Traiettorie: <i>Collision Avoidance</i> tra 2 robot <i>e-puck</i>	97
7.9	<i>Collision Avoidance</i> tra 2 robot <i>e-puck</i>	97

7.10	Confronto tra le velocità dei robot con due orizzonti di previsione differenti	98
7.11	Confronto tra i tempi di campionamento con due orizzonti di calcolo differenti	99
7.12	Traiettorie: <i>Collision</i> e <i>Obstacle Avoidance</i> tra tre robot e-puck	100
7.13	<i>Collision</i> e <i>Obstacle Avoidance</i> tra tre robot e-puck	101
7.14	Traiettoria rettangolare seguita dall' <i>e-puck</i>	102
7.15	In alto: traiettoria della coordinata y . In basso: stima adattativa della varianza	103
7.16	In alto: ingrandimento della traiettoria di y nei tre aggiornamenti. In basso: distribuzione di y lungo il vincolo nei tre aggiornamenti della varianza	104

Capitolo 1

Introduzione

La ricerca nel settore dell'Ingegneria dell'Automazione e del controllo ha avuto un forte sviluppo negli ultimi decenni. In particolare, lo studio relativo a tecniche di controllo predittivo, particolarmente versatili in quanto in grado di gestire in modo ottimale sistemi multivariabili, non lineari ed in presenza di vincoli operativi, ha raggiunto un livello di sviluppo maturo e vede un numero di applicazioni sempre crescente. Tale considerazione, tuttavia, si applica principalmente a sistemi in cui non sono presenti incertezze e in cui il modello e l'implementazione dello schema di controllo dedicato siano di tipo centralizzato.

In questo capitolo verranno illustrati alcuni esempi che motivano, da un lato, la ricerca legata al controllo predittivo di sistemi affetti da disturbi stocastici e, dall'altro, lo sviluppo di sistemi di controllo distribuiti per il coordinamento di sistemi indipendenti. Verranno quindi illustrati i principali problemi, tuttora parzialmente aperti, che emergono in questi ambiti. Successivamente verranno delineati gli obiettivi, i contributi originali e la struttura della presente tesi.

1.1 Sistemi affetti da disturbi stocastici

La maggior parte delle tecniche di controllo predittivo proposte in letteratura fino alla fine degli anni '90 e diffuse anche in contesti applicativi industriali, sono sviluppate sotto l'ipotesi che sia disponibile una conoscenza ideale del sistema sotto controllo, cioè che esso operi in condizioni nominali. Tuttavia tale ipotesi risulta essere limitativa a causa della presenza di dinamiche parassite, disturbi esogeni non modellizzati, rumori e incertezze di modello. Si è reso perciò necessario, da un lato, studiare le proprietà di robustezza dei metodi di controllo predittivo disponibili rispetto a tali fenomeni e, dall'altro lato, proporre tecniche *ad-hoc* in grado di tenerli esplicitamente in considerazione. Da un lato diversi lavori, tra i quali [17], evidenziano molte criticità legate alla robustezza di tecniche predittive nominali rispetto alla presenza di disturbi (soprattutto a causa della presenza di vincoli operativi). Inoltre, diverse tecniche sono state sviluppate in modo da ottenere robustezza rispetto a errori di modello e/o disturbi additivi [18], [19].

La maggior parte degli schemi sviluppati sfrutta l'ipotesi che i disturbi agenti sul sistema siano limitati, ma presentano alcune limitazioni. Sono caratterizzati, infatti, da oneri computazionali elevati nella fase di progetto *off-line* (nel caso di tecniche *tube-based*, in cui devono essere calcolati opportuni insiemi robusti positivamente invarianti, [15] e [18]) o nella fase di applicazione *on-line* (nel caso di algoritmi *min-max*, [17]). Inoltre risultano essere molto conservativi, in quanto sono progettati per garantire stabilità nel “*worst case*”, e non sfruttano l'eventuale conoscenza più approfondita sui disturbi agenti sul sistema (ad esempio nel caso in cui siano caratterizzati come processi stocastici con una distribuzione di probabilità nota). Tuttavia è possibile trovare importanti applicazioni in cui è necessario caratterizzare i disturbi in maniera più accurata, ad esempio mediante variabili aleatorie. Di seguito vengono riportati alcuni esempi di sistemi affetti da disturbi stocastici, in cui si è reso necessario cercare soluzioni adeguate e innovative.

Controllo della temperatura negli edifici

Dati statistici riportano che il 40% dell'uso energetico globale è sfruttato dagli impianti di climatizzazione degli edifici. Un utilizzo energetico ottimale risulta quindi fondamentale al fine di minimizzare i costi e massimizzare lo sfruttamento energetico senza generare sprechi.

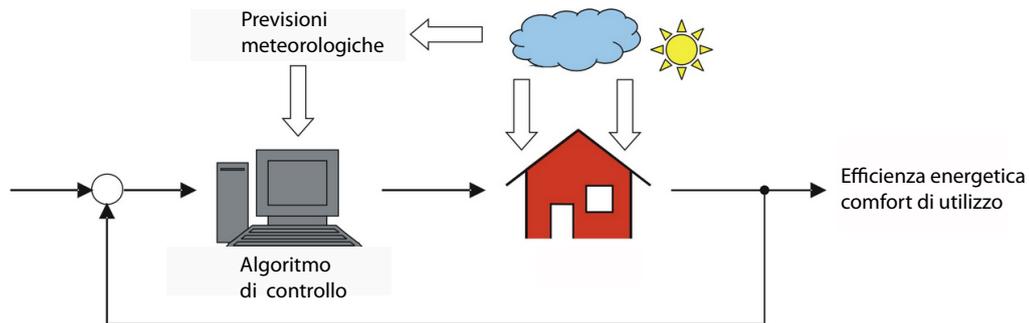


Figura 1.1: Controllo di temperatura di un edificio con MPC

In questo contesto, l'utilizzo di previsioni climatiche e meteorologiche, nonché relative al livello di occupazione degli edifici stessi e dall'utilizzo delle risorse in essi presenti risulta di fondamentale utilità per minimizzare gli sprechi energetici. È tuttavia noto che le previsioni meteorologiche portano con sé incertezze intrinseche relative alla natura stocastica dei processi atmosferici, e alla imperfezione dei modelli usati ed errori derivati dalla conoscenza incerta delle condizioni iniziali. In questo ambito, la caratterizzazione delle incertezze risulta fondamentale in fase di progetto e implementazione di un controllore dedicato. Ad esempio, in [8] una strategia di controllo di tipo *Stochastic Model Predictive Control* è utilizzata per controllare la temperatura ed ottimizzare l'efficienza energetica di un edificio rispetto a disturbi stocastici derivanti dalle previsioni meteorologiche. Un secondo esempio di applicazione di un algoritmo di tipo *Stochastic Model Predictive Control* è descritto in [9] dove le variabili stocastiche in gioco sono le previsioni dei carichi termici e della temperatura dell'ambiente. Un

ultimo esempio è descritto in [10], in cui un algoritmo MPC di tipo stocastico è adottato per regolare il comfort di un edificio soggetto alle variabili stocastiche presenti, in particolare la temperatura esterna, la radiazione solare, e aumenti di temperatura interna dovuti alla presenza di persone e all'utilizzo di elettrodomestici.

Smart Grid e Microgrid

Nell'ambito della trasmissione dell'energia elettrica le reti intelligenti o Smart Grid si stanno affermando sempre più. L'obiettivo primario delle Smart Grid si riassume in uno sfruttamento più efficiente dell'energia, con l'implementazione di tecnologie orientate al risparmio energetico in ogni fase, dalla produzione, all'uso finale in ambito commerciale e domestico. Le reti intelligenti si caratterizzano per le dimensioni ridotte e per la presenza significativa di dispositivi in grado di produrre energia da risorse rinnovabili. Il problema principale in questo ambito risiede nel fatto che la disponibilità di risorse rinnovabili è fortemente aleatoria. Dunque fondamentale è la disponibilità di modelli che descrivano le dinamiche dei disturbi e delle incertezze agenti sui vari sistemi. In [14] viene elaborato un algoritmo per la gestione distribuita di una rete di MicroGrids. In particolare, in virtù della presenza di incertezze di natura stocastica, si propone un algoritmo di compensazione dei disturbi al fine di ottenere una corretta ottimizzazione energetica. In Figura 1.2 è sintetizzato un esempio di rete distribuita per la gestione di una Smart Grid. Sono evidenziati differenti disturbi che agiscono sulla rete elettrica, tra cui la luce solare, la temperatura esterna e agenti atmosferici indesiderati.

Gestione dell'energia di un veicolo elettrico

È noto che il comportamento del conducente alla guida influisce notevolmente sul consumo di carburante del veicolo. In [11] vengono registrate informazioni statistiche sulla guida del conducente in differenti situazioni, le quali vengono continuamente aggiornate nel momento in cui il condu-

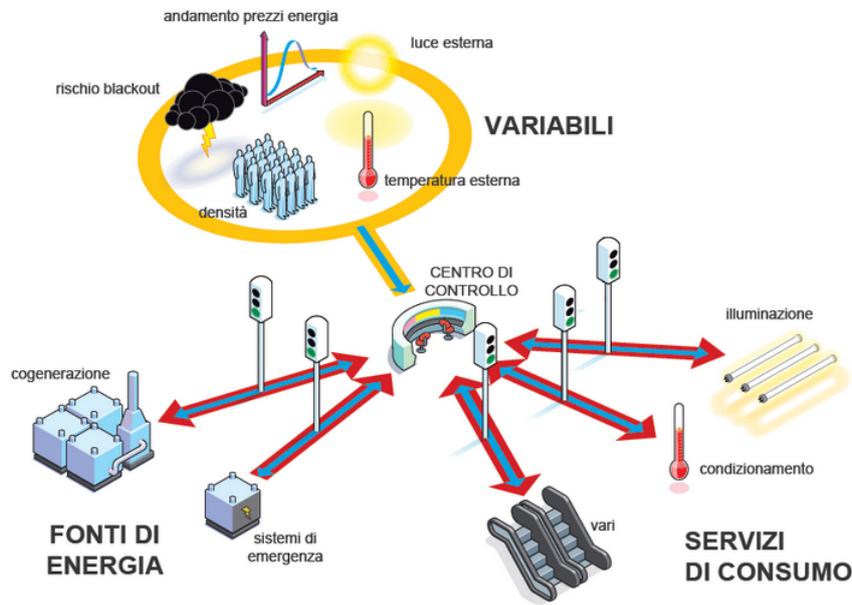


Figura 1.2: Smart Grid

cente modifica il suo stile di guida rispetto ai dati pregressi. Tale analisi ha permesso di identificare un modello dell'incertezza della guida come un modello lineare su cui agisce un rumore stocastico. Anche in quest'ambito viene sfruttato in modo completo l'informazione disponibile sull'incertezza attraverso il progetto di uno schema di controllo *ad-hoc* basato sul controllo predittivo stocastico. Per maggiori dettagli si rimanda a [11].

1.2 Problema del coordinamento e del controllo di sistemi indipendenti

Come discusso, tra le varie tecniche presenti in letteratura nell'ambito dei controlli automatici, il controllo predittivo è sicuramente uno dei più utilizzati in svariate applicazioni, grazie alla sua versatilità. Originariamente, data la complessità computazionale intrinseca, le applicazioni tipiche erano legate alla regolazione di impianti industriali che presentavano dinamiche sufficientemente lente e la cui dimensione fosse relativamente ridotta. Negli

ultimi anni, sia lo sviluppo tecnologico, che ha portato a migliorare le velocità computazionali dei calcolatori, sia la ricerca di algoritmi sempre più efficienti, ha permesso di sfruttare maggiormente il controllo predittivo per regolare sistemi caratterizzati da dinamiche più veloci e complesse.

In aggiunta, lo studio e la ricerca si stanno orientando verso sistemi complessi e di dimensioni considerevoli, si pensi ad esempio a reti di irrigazione o reti per la distribuzione dell'energia. Per questi sistemi alcune delle soluzioni proposte negli ultimi anni sono classificate come metodi distribuiti e propongono di suddividere il problema di controllo complesso e di vasta scala in sotto-problemi risolti in parallelo da sistemi di controllo integrati nelle varie componenti dell'impianto. Gli algoritmi di tipo distribuito differiscono a seconda del livello di interazione che sussiste tra i sotto-sistemi, i quali possono essere dinamicamente vincolati oppure semplicemente interagenti poiché sfruttano medesime risorse. Per realizzare un sistema di controllo distribuito, i sotto-sistemi devono scambiarsi informazioni, a vari livelli e con differenti modalità. L'informazione è un fattore chiave del sistema automatizzato, la quale una volta raccolta viene elaborata e successivamente trasmessa tra sotto-sistema e sotto-sistema. Nell'ambito del controllo distribuito di sistemi interagenti, la tecnica di controllo predittivo risulta essere particolarmente adatta e gli algoritmi che integrano soluzioni distribuite in un contesto di controllo predittivo sono chiamati *Distributed Model Predictive Control*. Per maggiori approfondimenti su questi approcci si rimanda la trattazione a [16], dove è presentata una notevole raccolta di metodi e applicazioni.

Svariate sono le applicazioni che hanno motivato lo studio e la ricerca in questo ambito. Ad esempio forte impulso è stato dato da problemi di controllo e coordinamento di agenti autonomi robotizzati. Questi robot autonomi, detti *Unmanned Autonomous Vehicles*, sono sempre più utilizzati in differenti applicazioni, dal campo militare a quello navale e spaziale, ad esempio per fini esplorativi di ambienti ostili per l'uomo, oppure in ambito domotico come sostegno all'uomo nello svolgere operazioni di vita quotidiana-

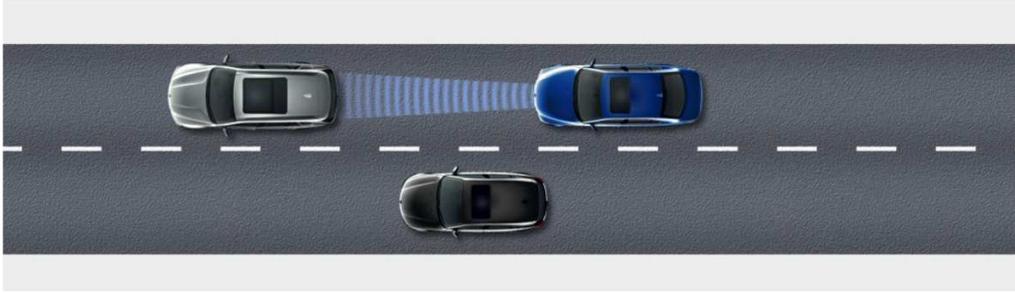


Figura 1.3: Cruise Control

na. Nell'ambito della robotica mobile sono stati sviluppati diversi algoritmi distribuiti basati su MPC. In tutti questi, gli obiettivi riguardano il coordinamento di flotte di robot che devono generare percorsi ottimi rispettando vincoli di *obstacle* e *collision avoidance*, come ad esempio discusso in [13]. Nell'ambito della *collision avoidance* un applicazione nota in campo automobilistico è relativa all *cruise control*, in grado di regolare automaticamente la velocità da mantenere per rispettare i vincoli di distanza rispetto a altri veicoli, si veda la Figura 1.3.

1.3 Obiettivo della Tesi e contributi originali

L'obiettivo di questa tesi è sviluppare un algoritmo di controllo predittivo stocastico di tipo distribuito per l'inseguimento di traiettorie di riferimento di sistemi indipendenti ma accoppiati tramite vincoli. La formulazione sarà ricavata a partire da algoritmi già parzialmente noti in letteratura (sviluppati prevalentemente per la regolazione di sistemi nel caso centralizzato) e ve ne sarà proposta una estensione innovativa. Un vantaggio di questi algoritmi rispetto a metodi analoghi presenti in letteratura, riguarda la possibilità di includere disturbi anche con supporto illimitato nel progetto del controllore; inoltre la formulazione presentata permette di garantire le proprietà di stabilità e *recursive feasibility* mediante l'utilizzo di una strategia opportu-

na di inizializzazione e di una scelta degli insiemi terminali.

Le tecniche descritte sono applicate ad un caso di studio: il coordinamento di una piccola flotta di robot a unicycle. Mediante prove sperimentali sarà caratterizzato il disturbo agente sul sistema reale come un rumore bianco con distribuzione gaussiana. Tale caratterizzazione risulterà necessaria per il progetto del controllore locale. Una volta definito opportunamente l'algoritmo, numerose prove sperimentali hanno permesso di validare le proprietà dello schema risultante.

Infine, per rendere l'algoritmo sensibile a variazioni sull'ambiente sarà proposta una metodologia di stima adattativa del disturbo che permetterà al controllore di aggiornare la conoscenza del disturbo agente.

1.4 Struttura della Tesi

La tesi è strutturata nel modo seguente:

- Nel capitolo 2 si presenta dapprima la tecnica classica di controllo MPC (per la regolazione) per sistemi lineari non affetti da disturbi. Dopodiché si introduce un disturbo stocastico (rumore bianco a media nulla) sul sistema e si riformula il problema di regolazione in termini probabilistici. Particolare enfasi viene messa sulla riformulazione dei vincoli.
- Nel capitolo 3 si estende la soluzione ad un problema di inseguimento di traiettorie di riferimento. Inizialmente si presenta una soluzione in un contesto centralizzato; dopodiché la metodologia è estesa ad un caso distribuito per il controllo di sotto-sistemi interagenti mediante vincoli ma non dinamicamente accoppiati. Da ultimo si riformula l'algoritmo distribuito sfruttando una soluzione innovativa.

- Nel capitolo 4 si descrive l'apparato sperimentale scelto su cui testare e validare l'algoritmo descritto nel capitolo 3. Inoltre, mediante prove sperimentali si dà una caratterizzazione del disturbo agente, verificando che le ipotesi considerate, riguardanti la natura del disturbo, siano accettabili.
- Nel capitolo 5 si sfrutta l'algoritmo generale descritto nel capitolo 3 per risolvere un problema di navigazione autonoma con proprietà di *Obstacle* e *Collision Avoidance*. In particolare si spiega come includere nella funzione obiettivo e nei vincoli tali proprietà. Si descrive, quindi, una soluzione *on-line* per stimare i disturbi agenti, in modo da rendere l'algoritmo adattativo.
- Nel capitolo 6 si riportano i risultati ottenuti da prove in simulazione del problema descritto nel capitolo 5. Dapprima vengono mostrate prove riguardo l'inseguimento di traiettoria in uno spazio libero di collisioni; dopodiché si mostrano prove in cui sono presenti ostacoli; si mostrano quindi prove che includono più agenti nello spazio operativo.
- Nel capitolo 7 si svolgono prove, simili a quelle effettuate in simulazione, sul sistema reale dimostrando la validità delle ipotesi considerate e dell'algoritmo descritto.
- Nel capitolo 8 si riassumono i risultati ottenuti e si propongono alcuni sviluppi futuri.
- Nell'Appendice A vengono discussi gli aspetti implementativi dell'algoritmo presentato.

Capitolo 2

Controllo predittivo stocastico centralizzato per la regolazione

2.1 Introduzione al controllo MPC

Una strategia di controllo ampiamente utilizzata nelle industrie di processo, sviluppatasi intorno agli anni '80, è il controllo predittivo (MPC, *Model Predictive Control*). È una tecnica di controllo molto versatile, che trova applicazione in diversi ambiti. In particolare, sistemi di controllo basati su MPC risultano essere molto utili per gestire sistemi multivariabili, di larga scala e/o caratterizzati da non linearità. Inoltre il progetto di sistemi di controllo basati su MPC non presenta le limitazioni che possono presentarsi nel caso di progetto con tecniche classiche (nel dominio della frequenza) riscontrate, ad esempio, nel caso di sistemi multivariabili, a fase non minima o instabili. Infine presenta numerosi vantaggi rispetto a schemi di controllo tradizionali, poiché è in grado di gestire in modo naturale vincoli e saturazioni, e conferisce caratteristiche di ottimalità al sistema.

Il controllo predittivo viene formulato come un opportuno problema di ottimizzazione soggetto a vincoli sulle variabili di stato e di ingresso. In particolare la tecnica MPC consiste nel determinare la sequenza di ingressi ottimi da fornire al sistema lungo un orizzonte futuro, detto orizzonte di

predizione, $\mathcal{U}_t^o = [u_t, \dots, u_{t+N-1}]$, mediante la minimizzazione *on line* di una cifra di merito, costruita sulla predizione del comportamento del sistema in un orizzonte futuro fissato. Tale metodo è utilizzato congiuntamente alla tecnica *Receding Horizon* (RH), la quale consiste nel risolvere il problema di ottimizzazione ad ogni istante di campionamento ed applicare al sistema solo il primo ingresso della sequenza ottima di ingressi precedentemente calcolata. Al successivo istante si risolve nuovamente il problema, muovendo di un passo in avanti l'orizzonte di predizione.

In particolare, all'istante t si risolve il problema di ottimizzazione vincolata e si seleziona il primo valore u_t del vettore degli ingressi. All'istante successivo $t + 1$, l'orizzonte di predizione viene fatto scorrere in avanti e l'ottimizzazione viene nuovamente risolta. In Figura 2.1 è riportata una illustrazione della tecnica appena descritta per un sistema SISO.

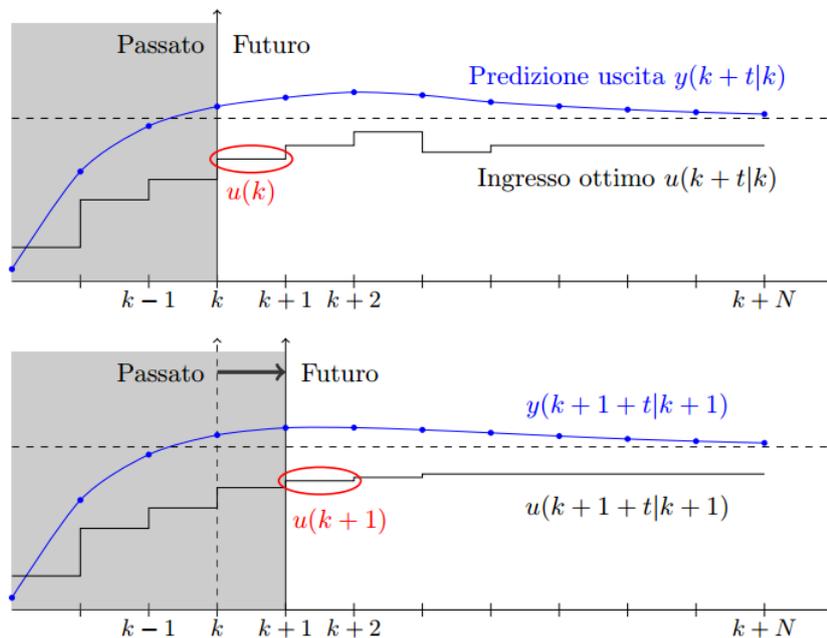


Figura 2.1: MPC con Receding Horizon

In questo capitolo verrà descritto un algoritmo di controllo predittivo stocastico per la regolazione nel caso centralizzato di un sistema soggetto

a disturbi stocastici. Inizialmente sarà descritta la tecnica classica MPC per sistemi non affetti da disturbi; dopodiché sarà aggiunto un disturbo stocastico sul sistema e sarà formulato l'algoritmo, analizzando le proprietà di stabilità e la scelta dei parametri di progetto.

2.2 MPC per sistemi lineari

Si consideri un generico sistema lineare

$$\begin{cases} x_{t+1} = Ax_t + Bu_t \\ y_t = Cx_t \end{cases} \quad (2.1)$$

dove $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$ e $y_t \in \mathbb{R}^p$ sono, rispettivamente, i vettori di stato, di ingresso e di uscita del sistema. Fissato un orizzonte di predizione N si vuole calcolare il vettore di ingressi ottimo $\mathcal{U}_t^\circ = [u_{t+1}, \dots, u_{t+N-1}]$ che minimizzi la cifra di merito:

$$V(x_t, \mathcal{U}_t^\circ) = \sum_{k=0}^{N-1} l(x_{t+k}, u_{t+k}) + V^f(x_{t+N}) \quad (2.2)$$

dove $l(\cdot)$ è una funzione definita positiva di stato ed ingresso, detto *stage cost*, mentre V^f è una funzione, anch'essa definita positiva dipendente dallo stato alla fine dell'orizzonte di predizione, detta costo finale (o *terminal cost*). Una scelta comune per tali funzioni ricade su funzioni quadratiche, in particolare:

$$l(x_t, u_t) = \|x_t\|_Q^2 + \|u_t\|_R^2 = x_t^T Q x_t + u_t^T R u_t \quad (2.3)$$

$$V^f(x_{t+N}) = \|x_{t+N}\|_P^2 = x_{t+N}^T P x_{t+N} \quad (2.4)$$

dove Q è una matrice simmetrica semidefinita positiva ($Q = Q^T \succeq 0$), mentre R è anch'essa una matrice simmetrica ma definita positiva ($R = R^T \succ 0$). La scelta della matrice P verrà, invece, specificata in seguito.

Si suppone che lo stato e l'ingresso siano vincolati ad assumere valori entro insiemi definiti a priori. Tali vincoli sono formalizzati come segue:

$$x_t \in \mathbb{X} \quad (2.5)$$

$$u_t \in \mathbb{U} \quad (2.6)$$

dove si suppone che \mathbb{X} e \mathbb{U} siano insiemi convessi, contenenti l'origine e politopici. In particolare questa ultima scelta verrà meglio specificata in seguito.

Si richiede inoltre che lo stato alla fine dell'orizzonte di predizione sia compreso in un insieme, detto *terminal set*:

$$x_{t+N} \in \mathbb{X}^f \quad (2.7)$$

Per quanto detto sopra, con la tecnica MPC con RH, una volta risolta l'ottimizzazione all'istante t , viene applicato al sistema solo il primo ingresso $u_{t|t}$. Alla successiva iterazione $t + 1$ verrà applicato l'ingresso $u_{t+1|t+1}$. In questo modo la legge di controllo *state-feedback* in anello chiuso ottenuta risulta essere invariante nel tempo, ed è la seguente:

$$u_{t|t} = K_{MPC}(x_t) \quad (2.8)$$

2.2.1 Scelta dei parametri per la stabilità

Nella formulazione del problema discussa in questo capitolo compaiono due parametri di progetto fondamentali per garantire le proprietà di *recursive feasibility* e di stabilità: il *terminal set* \mathbb{X}^f e il *terminal cost* V^f .

Nel contesto dell'MPC, ad ogni istante temporale, è fondamentale assicurare l'esistenza di una soluzione e garantire, inoltre, che all'istante seguente il problema di ottimizzazione vincolata sia ancora ben posto. In particolare, la ricerca di una soluzione ammissibile ad ogni passo rientra nel problema della *recursive feasibility*.

In particolare la proprietà fondamentale che il *terminal set* deve possedere,

per garantire *recursive feasibility*, è la positiva invarianza.

Definendo ora una opportuna legge di controllo ausiliaria stabilizzante

$$u_t = -K_a x_t \quad (2.9)$$

(cioè tale che $A - BK_a$ sia stabile), il set \mathbb{X}^f si dice positivamente invariante rispetto al sistema controllato $x_{t+1} = (A - BK_a)x_t$ se:

$$x_t \in \mathbb{X}^f \Rightarrow x_{t+1} \in \mathbb{X}^f \quad (2.10)$$

Tale set è definito, inoltre, in modo tale che:

$$\mathbb{X}^f \subseteq \mathbb{X} \quad (2.11)$$

$$-K_a \mathbb{X}^f \subseteq \mathbb{U} \quad (2.12)$$

Si noti che la proprietà definita in (2.12) garantisce che $-K_a x \in \mathbb{U}, \forall x \in \mathbb{X}^f$. Come detto, quindi, una definizione opportuna di \mathbb{X}^f permette di garantire che, se il problema MPC ammette soluzione all'istante t , allora ammette soluzione anche ad ogni istante successivo.

Le proprietà di stabilità e convergenza dello stato all'origine vengono garantite scegliendo opportunamente il *terminal cost* e, nel caso in esame, attraverso la scelta della matrice di peso P dell'equazione (2.4).

In particolare tale matrice si ricava come soluzione della seguente equazione di Lyapunov:

$$(A - BK_a)^T P (A - BK_a) - P = -(Q + K_a^T R K_a) \quad (2.13)$$

La matrice P permette, quindi, di definire in modo molto semplice il set \mathbb{X}^f , nel caso esso sia definito come ellissoidale). In particolare si definisce come:

$$\mathbb{X}^f = \{x : x^T P x \leq \alpha\} \quad (2.14)$$

dove α è definito opportunamente in modo tale che (2.11) e (2.12) siano verificate. Per ulteriori dettagli si rimanda la trattazione a [1].

È opportuno notare infine che, nonostante la scelta della legge di controllo

ausiliaria K_a sia totalmente arbitraria, (a patto che, ovviamente, $A - BK_a$ sia stabile), tale scelta ricada di norma, per ovvie ragioni, sul guadagno del classico controllore LQ.

2.3 Controllo MPC Stocastico per la regolazione

La tecnica di controllo descritta fino ad ora si adatta bene a sistemi di cui è perfettamente nota l'evoluzione temporale dello stato, ovvero senza considerare disturbi di alcun genere. Tuttavia, nelle applicazioni reali tale ipotesi non può essere fatta a priori, proprio poiché errori derivanti da incertezze di modello, dalle misure o dal sistema di attuazione, sono in generale presenti. Quindi non si è in grado di fornire conclusioni certe riguardo la stabilità del sistema. Risulta a questo punto indispensabile rendere robusto il sistema rispetto ai vari disturbi presenti, al fine di garantire che le proprietà desiderate vengano mantenute.

Negli ultimi anni molti metodi sono stati proposti per sistemi affetti da disturbi. Nella maggior parte dei casi i metodi proposti si basano sull'ipotesi che il disturbo agente sul sistema sia incognito, ma limitato. Tali metodi vengono classificati come metodi deterministici, in quanto non considerano l'ipotesi che il disturbo sul sistema sia un processo stocastico. In particolare, sfruttando tali approcci, si è dimostrato che, adottando una particolare strategia di restringimento dei vincoli operativi e introducendo un grado di libertà supplementare relativo allo stato iniziale del sistema "nominale", è possibile ottenere garanzie di stabilità robusta. Tuttavia, tutti questi algoritmi, nonostante garantiscano ottimalità e *recursive feasibility*, assumono che il rumore sia limitato e si basano su approcci di tipo *worst case*, risultando quindi molto conservativi. Inoltre risultano essere computazionalmente onerosi in fase di progetto, a causa del calcolo dei set invarianti, calcolati off-line.

Altro aspetto molto importante riguarda il fatto che non vengono sfruttate le informazioni disponibili derivanti dai disturbi, quali ad esempio la loro distribuzione. Per queste ragioni sono stati sviluppati più recentemente algoritmi di tipo probabilistico. Alcuni di essi sono basati su approcci randomizzati o approcci detti “a scenario”. Questi sfruttano una metodologia che permette di considerare sistemi sia lineari che non lineari, affetti da disturbi che presentano supporto illimitato e non convesso. Per contro, oltre ad essere computazionalmente esigenti per implementazioni pratiche, le proprietà di convergenza e *recursive feasibility* sono difficili da provare. Fare riferimento a [4] per maggiori informazioni.

Altri approcci si basano sulla riformulazione deterministica del problema, sfruttando la conoscenza statistica del rumore o dell’incertezza del modello. Alcuni di essi si basano sull’ipotesi di disturbi limitati e in tal caso le proprietà di convergenza e *recursive feasibility* sono facilmente rispettate. Se invece il disturbo agente sul sistema ha supporto illimitato, sono necessarie analisi molto più accurate.

In questo capitolo verrà descritto un algoritmo che garantisca le proprietà ricercate anche in presenza di disturbi con supporto illimitato.

2.3.1 Formalizzazione dei vincoli

Si consideri il seguente sistema lineare Σ :

$$x_{t+1} = Ax_t + Bu_t + Fw_t \quad (2.15)$$

dove $w_t \in \mathbb{R}^p$ è un rumore bianco a media nulla e varianza W e con supporto, in generale, illimitato. Tale disturbo, come già precedentemente detto, può essere introdotto per considerare ad esempio errori di modello e di attuazione.

Per il sistema in questione si suppone che valgano le seguenti ipotesi:

- perfetta conoscenza dello stato;

- la coppia (A, B) sia stabilizzabile.

L'obiettivo di questa trattazione è trovare una legge di controllo che garantisca le proprietà di ottimalità, stabilità e *recursive feasibility* nel rispetto dei vincoli. In particolare, ci si propone di risolvere un problema MPC per la regolazione a zero dello stato del sistema soggetto a vincoli.

Prima di addentrarsi nella trattazione vera e propria, è opportuno comprendere come poter affrontare una ottimizzazione vincolata in un contesto probabilistico.

Si è detto che sul sistema agisce un disturbo, il quale si suppone possa avere supporto illimitato. Ciò significa che le variabili di stato ed ingresso, durante la loro evoluzione, risultano essere disturbate, risultando anch'esse variabili stocastiche con varianza opportuna rispetto al valore atteso che assumerebbero se il disturbo non fosse presente. È quindi chiaro che predire in modo esatto l'andamento dello stato lungo l'orizzonte futuro non è possibile. La soluzione migliore è, quindi, predire il valore atteso dello stato e l'incertezza rispetto all'evoluzione di quest'ultimo (varianza). In vista di ciò il funzionale di costo descritto in (2.2) verrà modificato, considerando il suo valore atteso. In particolare, il funzionale di costo è:

$$J_t = \mathbb{E} \left\{ \sum_{k=0}^{N-1} (\|x_{t+k}\|_Q^2 + \|u_{t+k}\|_R^2) + \|x_{t+N}\|_P^2 \right\} \quad (2.16)$$

in cui \mathbb{E} è l'operatore di valore atteso.

Come discusso, l'ottimizzazione che ci si propone di risolvere è soggetta a vincoli. A fronte di quanto detto, anche i vincoli devono essere espressi in termini probabilistici. Non è possibile infatti, a causa del disturbo, garantire vincoli *hard* sulle variabili vere dello stato e dell'ingresso. Risulta quindi chiaro che vincoli del tipo (2.5) e (2.6) non possono essere garantiti con probabilità 1.

Ciò che si fa, per ovviare a tale problema, è introdurre una probabilità di violazione dei vincoli.

Si ridefiniscono quindi i vincoli (2.5) e (2.6) in termini probabilistici come

segue:

$$\mathcal{P} \left\{ (b_r)^T x_t \geq x_r^{max} \right\} \leq p_r^x \quad r = 1, \dots, n_x \quad (2.17)$$

$$\mathcal{P} \left\{ (c_s)^T u_t \geq u_s^{max} \right\} \leq p_s^u \quad s = 1, \dots, n_u \quad (2.18)$$

dove:

- $\mathcal{P}\{\phi\}$ denota la probabilità di ϕ ;
- p_r^x, p_s^u sono parametri di progetto e rappresentano le probabilità massime di violazione dei vincoli;
- x_r^{max}, u_s^{max} sono i limiti per x e u , anch'essi scelti in fase di progetto e legati alla definizione dei set \mathbb{X} e \mathbb{U} in (2.5) e (2.6);
- b_r, c_s sono opportuni vettori costanti, anch'essi legati alla definizione di \mathbb{X} e \mathbb{U} .

Si assume inoltre che i vincoli lineari $(b_r)^T x_t \leq x_r^{max}$ e $(c_s)^T u_t \leq u_s^{max}$ definiscano insieme convessi, contenenti l'origine e politopici:

$$\mathbb{X} = \{x \in \mathbb{R}^n : b_r x \leq 1, r = 1, \dots, n_x\} \quad (2.19)$$

$$\mathbb{U} = \{u \in \mathbb{R}^m : c_s u \leq 1, s = 1, \dots, n_u\} \quad (2.20)$$

Ci si propone a questo punto di definire in modo rigoroso la struttura dell'algoritmo che verrà utilizzato. Dapprima si analizzerà la natura del controllo e verrà fatta una analisi delle variabili in merito. Dopodiché il problema verrà riformulato in un ambito deterministico, al fine di definire in modo efficiente un algoritmo di MPC risolvibile *on line*. In particolare si sfrutterà una tecnica, nota in letteratura, per affrontare tale problema.

L'idea che sta alla base è quella di utilizzare un sistema, che definisce l'evoluzione del valore atteso e corrispondente al sistema (2.15) in cui non agisce alcun disturbo, per risolvere il problema di ottimizzazione vincolata.

In particolare i vincoli saranno definiti rispetto al valore atteso e alla varianza dello stato e dell'ingresso. Infine, una volta risolta l'ottimizzazione, il controllore permetterà di ricavare i valori ottimi degli ingressi reali, da applicare al sistema.

2.3.2 Struttura del controllore

In modo del tutto analogo al noto algoritmo robusto *tube-based*, per il quale si rimanda la trattazione a [5] e [6], si definisce la legge di controllo in anello chiuso per il sistema (2.15)

$$u_t = \bar{u}_t - K_t(x_t - \bar{x}_t) \quad (2.21)$$

dove x_t e u_t sono, rispettivamente, lo stato e l'ingresso del sistema (2.15), mentre \bar{x}_t e \bar{u}_t sono, rispettivamente lo stato e l'ingresso del sistema "nominale" $\bar{\Sigma}$:

$$\bar{x}_{t+1} = A\bar{x}_t + B\bar{u}_t \quad (2.22)$$

in cui \bar{u}_t e il guadagno K_t , generalmente tempo-variante, rappresentano le variabili di ottimizzazione, calcolate ad ogni istante.

Per quanto detto, si applica il classico controllo MPC al sistema nominale $\bar{\Sigma}$ e, una volta risolta l'ottimizzazione, si utilizza la legge di controllo in *feedback* definita in (2.21) per ottenere l'ingresso da applicare al sistema reale.

Si considerino ora i due sistemi (2.15) e (2.22) e la legge di controllo (2.21). Si definisca la variabile errore dello stato $\delta x_t = x_t - \bar{x}_t$ e si calcoli la sua dinamica. Si ottiene:

$$x_{t+1} - \bar{x}_{t+1} = A(x_t - \bar{x}_t) + B(u_t - \bar{u}_t) + Fw_t \quad (2.23)$$

↓

$$\delta x_{t+1} = (A - BK_t)\delta x_t + Fw_t \quad (2.24)$$

che descrive l'evoluzione temporale dell'errore dello stato. Si noti che δx_t è una variabile stocastica.

Se si inizializza il valore dello stato nominale \bar{x}_t in modo che $\mathbb{E}\{\delta x_0\} = 0$, poiché $\mathbb{E}\{w_t\} = 0, \forall t \geq 0$, allora $\bar{x}_t = \mathbb{E}\{x_t\}$ e $\mathbb{E}\{\delta x_t\} = 0, \forall t \geq 0$.

Detta X_t la varianza dell'errore all'istante t , si ha che $X_t = \mathbb{E}\{\delta x_t \delta x_t^T\}$.

La sua evoluzione temporale si ottiene come segue:

$$\begin{aligned} X_{t+1} &= \mathbb{E}\left\{\delta x_{t+1} \delta x_{t+1}^T\right\} = \\ &= \mathbb{E}\left\{(x_{t+1} - \bar{x}_{t+1})(x_{t+1} - \bar{x}_{t+1})^T\right\} = \\ &= \mathbb{E}\left\{(Ax_t + Bu_t + Fw_t - A\bar{x}_t - B\bar{u}_t)(Ax_t + Bu_t + \right. \\ &\quad \left. + Fw_t - A\bar{x}_t - B\bar{u}_t)^T\right\} = \\ &= \mathbb{E}\left\{((A - BK_t)\delta x_t + Fw_t)((A - BK_t)\delta x_t + Fw_t)^T\right\} = \\ &= \mathbb{E}\left\{(A - BK_t)\delta x_t \delta x_t^T (A - BK_t)^T + Fw_t w_t^T F^T + \right. \\ &\quad \left. + (A - BK_t)\delta x_t w_t^T F^T + Fw_t \delta x_t^T (A - BK_t)^T\right\} \end{aligned}$$

Ricordando che $\mathbb{E}\{\delta x_t w_t^T\} = 0$ si ottiene la seguente equazione:

$$X_{t+1} = (A - BK_t)X_t(A - BK_t)^T + FWF^T \quad (2.25)$$

L'equazione (2.25) rappresenta l'evoluzione temporale della matrice di covarianza dell'errore.

Allo stesso modo, ponendo $\delta u_t = u_t - \bar{u}_t$, per la legge di controllo (2.21), risulta $\delta u_t = -K_t \delta x_t$. Quindi $\mathbb{E}\{\delta u_t\} = 0$ e $U_t = \mathbb{E}\{\delta u_t \delta u_t^T\} = K_t X_t (K_t)^T, \forall t \geq 0$.

Le variabili di stato ed ingresso del sistema (2.15) possono essere considerate quindi come variabili stocastiche a media nulla e varianza opportuna,

in particolare:

$$\begin{aligned}x_t &= \bar{x}_t + \delta x_t \Rightarrow x_t \sim vs(\bar{x}_t, X_t) \\u_t &= \bar{u}_t + \delta u_t \Rightarrow u_t \sim vs(\bar{u}_t, U_t),\end{aligned}$$

dove vs indica una generica variabile stocastica.

2.3.3 Riformulazione deterministica dei vincoli

Come già detto all'inizio di questo capitolo i vincoli sullo stato e sull'ingresso sono formalizzati in senso probabilistico.

Ci si propone ora di riscrivere i vincoli (2.17) e (2.18) in “modo deterministico”, ossia in termini di valore atteso e varianza delle variabili in gioco, la cui evoluzione temporale è descritta dalle equazioni deterministiche (2.22) e (2.25). A tal scopo si fa riferimento alla disuguaglianza di Cantelli-Chebyshev.

Disuguaglianza di Cantelli-Chebyshev: sia y una variabile aleatoria con media \bar{y} e varianza Y . Allora per ogni $\alpha \geq 0 \in \mathbb{R}$ vale:

$$\mathcal{P}(y \geq \bar{y} + \alpha) \leq \frac{Y}{Y + \alpha^2} \quad (2.26)$$

□

Si consideri ora il vincolo in (2.17) e si definisca

$$\delta x_t = x_r^{max} - b_r^T \bar{x}_t \quad (2.27)$$

Si mostra ora come utilizzare (2.26) per garantire (2.17) e (2.18). Combinando l'equazione (2.27) con la disuguaglianza (2.26) risulta:

$$\mathcal{P}\{b_r^T x_t \geq x_r^{max}\} = \mathcal{P}\{b_r^T x_t \geq b_r^T \bar{x}_t + \delta x_t\} \leq \frac{b_r^T X b_r}{b_r^T X b_r + \delta x_t^2} \quad (2.28)$$

La disuguaglianza in probabilità è soddisfatta imponendo:

$$\frac{b_r^T X b_r}{b_r^T X b_r + \delta x_t^2} \leq p_x \rightarrow b_r^T X b_r \left(\frac{1 - p_x}{p_x} \right) \leq \delta x_t^2 \rightarrow \sqrt{b_r^T X b_r} \sqrt{\frac{1 - p_x}{p_x}} \leq \delta x_t$$

È , quindi, possibile riscrivere i vincoli nel modo seguente:

$$(b_r)^T \bar{x}_t \leq x_t^{max} - \sqrt{(b_r)^T X_t b_r} f(p_r^x) \quad (2.29)$$

$$(c_s)^T \bar{u}_t \leq u_t^{max} - \sqrt{(c_s)^T U_t c_s} f(p_s^u) \quad (2.30)$$

dove:

$$f(p) = \sqrt{\frac{1-p}{p}}, \text{ con } p \in (0, 0.5] \quad (2.31)$$

La riformulazione dei vincoli in senso deterministico ha permesso, a partire da una formulazione sulle variabili vere dello stato e dell'ingresso, (x_t, u_t) di ricavare una formulazione in termini del valore atteso (\bar{x}_t, \bar{u}_t) e della varianza (X_t, U_t) delle suddette variabili. In particolare a partire da vincoli *soft* sulle variabili vere, si ottengono vincoli di tipo *hard* sui valori attesi e sulle varianze.

Si noti che la riformulazione dei vincoli da probabilistici a deterministici è stata ricavata senza considerare nessuna informazione eventualmente disponibile sulla distribuzione del disturbo agente. In particolare tali disuguaglianze (2.29) e (2.30) possono essere rese meno conservative nel caso in cui sia nota la distribuzione di w_t . Si consideri ora il caso in cui sia x sia una variabile gaussiana.

Si consideri una generica variabile aleatoria con distribuzione gaussiana:

$$b^T x \sim \mathcal{G} \left(\mathbb{E}[b^T x], \text{Var} [b^T x] \right),$$

con media $\mathbb{E}[b^T x] = b^T \bar{x}$ e varianza $\text{Var} [b^T x] = \mathbb{E}[b^T (x - \bar{x})(x - \bar{x})^T b]$, cioè $b^T x \sim \mathcal{G}(b^T \bar{x}, b^T X b)$. Si definisca la seguente variabile a media nulla e varianza unitaria

$$\delta = \frac{b^T (x - \bar{x})}{\sqrt{b^T X b}} \sim \mathcal{G}(0, 1)$$

Ora, è noto che per una variabile gaussiana a media nulla e varianza unitaria la funzione di probabilità cumulativa è univocamente definita. In particolare la probabilità che δ sia minore di un valore $\bar{\delta}$ rappresenta il valore puntuale

della distribuzione cumulativa in corrispondenza di $\bar{\delta}$. Sia quindi \mathcal{N} la funzione di probabilità cumulativa, allora risulta

$$\begin{aligned} \mathcal{P}(\delta \leq \bar{\delta}) &= \mathcal{N}(\bar{\delta}) \Rightarrow \mathcal{P}(\delta > \bar{\delta}) = 1 - \mathcal{N}(\bar{\delta}) \Rightarrow \\ \Rightarrow \mathcal{P}\left(\frac{b^T(x - \bar{x})}{\sqrt{b^T X b}} > \bar{\delta}\right) &= 1 - \mathcal{N}(\bar{\delta}) \Rightarrow \mathcal{P}\left(b^T x \geq b^T \bar{x} + \bar{\delta} \sqrt{b^T X b}\right) = 1 - \mathcal{N}(\bar{\delta}) \end{aligned}$$

Ponendo $p = 1 - \mathcal{N}(\bar{\delta})$ si ricava $\bar{\delta} = \mathcal{N}^{-1}(1 - p)$. A questo punto, sostituendo il valore di $\bar{\delta}$ si ottiene:

$$\mathcal{P}\left(b^T x > b^T \bar{x} + \sqrt{b^T X b} \mathcal{N}^{-1}(1 - p)\right) = p$$

si ottiene la seguente implicazione biunivoca

$$\mathcal{P}(\delta > \bar{\delta}) = p \Leftrightarrow b^T x > b^T \bar{x} + \sqrt{b^T X b} \mathcal{N}^{-1}(1 - p)$$

Per garantire le disequazioni (2.17) e (2.18) basta porre:

$$b^T \bar{x} + \sqrt{b^T X b} \mathcal{N}^{-1}(1 - p) \leq \bar{x}_t^{max} \quad (2.32)$$

Dalla (2.32) si cavano le espressioni (2.29) e (2.30), dove

$$f(p) = \mathcal{N}^{-1}(1 - p) \quad (2.33)$$

Se si confronta l'andamento della funzione $f(p)$ appena calcolata con quello della funzione calcolata mediante la disuguaglianza di Cantelli si verifica quanto segue

$$\mathcal{N}^{-1}(1 - p) \leq \sqrt{\frac{1 - p}{p}} \quad (2.34)$$

In Figura 2.2 si riporta l'andamento, su scala logaritmica, delle due funzioni, al variare della probabilità p .

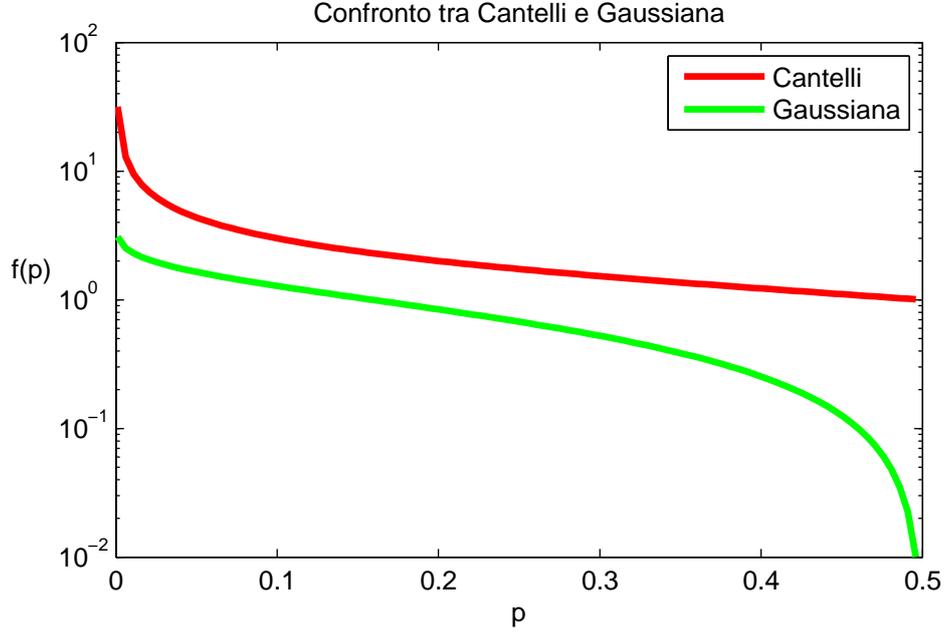


Figura 2.2: Confronto tra $f(p)$ calcolata mediante la disuguaglianza di Cantelli e quella calcolata supponendo una distribuzione normale

Risulta quindi chiaro che l'utilizzo delle informazioni riguardo la gaussianità del rumore permette di rilassare ulteriormente i vincoli, ottenendo soluzioni migliori in termini di prestazioni.

Infine, si può notare in (2.29), (2.30) come l'operazione di restrizione dei vincoli abbia generato dei vincoli non lineari rispetto a X . Ciò può comportare problemi di tipo computazionale e può risultare necessario un passaggio ulteriore per linearizzare tali vincoli e includerli nel problema di ottimizzazione. È possibile verificare che (2.29) e (2.30) sono implicati dai seguenti:

$$b_r^T \bar{x}_t \leq (1 - 0.5\epsilon)x_t^{max} - \frac{1 - p_x}{2\epsilon x_t^{max} p_x} b_r^T X b_r \quad (2.35)$$

$$c_s^T \bar{u}_t \leq (1 - 0.5\epsilon)u_t^{max} - \frac{1 - p_u}{2\epsilon u_t^{max} p_u} c_s^T U c_s \quad (2.36)$$

dove $\epsilon \in (0, 1]$. Ovviamente, tale processo di linearizzazione comporta un restringimento ulteriore dei vincoli, che a sua volta si traduce in una mag-

giore conservatività dei risultati. Per i passaggi relativi alla linearizzazione si rimanda la trattazione a [3].

Fino ad ora è stata presentata una riformulazione dei vincoli da probabilistici a deterministici, linearizzando poi questi ultimi. In realtà, in questo contesto verrà implementata una soluzione in cui il guadagno è costante, $K_t = \bar{K}$, $\forall t \geq 0$, quindi non sarà richiesta l'operazione di linearizzazione appena presentata. Questa strategia, che verrà poi utilizzata, genererà una soluzione subottima, dovuta al fatto che l'ottimizzazione verrà eseguita solamente rispetto alla sequenza dei valori attesi dell'ingresso e non alla sequenza dei guadagni, essendo essi posti costanti, ma risulterà essere una buona scelta in termini di performance e di complessità implementativa e quindi risulterà essere un buon *trade off*. Nel proseguo della tesi questo aspetto verrà maggiormente sottolineato.

In questa trattazione, inoltre, per motivi puramente semplificativi, i vincoli (2.29) e (2.30) vengono riscritti in termini normalizzati rispetto ai valori massimi x_t^{max} e u_t^{max} come segue:

$$(b_r)^T \bar{x}_t \leq 1 - \sqrt{(b_r)^T X_t b_r} f(p_r^x) \quad (2.37)$$

$$(c_s)^T \bar{u}_t \leq 1 - \sqrt{(c_s)^T U_t c_s} f(p_s^u) \quad (2.38)$$

dove le matrici $(b_r)^T$ e $(c_s)^T$ sono ricalcolate rispetto alla normalizzazione. D'ora in avanti sarà questa la forma utilizzata per formulare i vincoli su stato ed ingresso del sistema.

Funzione di costo

Prima di formulare correttamente il problema MPC è necessario analizzare in modo dettagliato il funzionale di costo descritto in (2.16). Si consideri inizialmente solo il termine $\mathbb{E} \left\{ \|x_t\|_Q^2 \right\}$, dal quale è possibile fare facili estensioni. In particolare si vuole comprendere il significato dell'applicazione del

valore atteso alla norma euclidea di una variabile aleatoria

$$\begin{aligned}\mathbb{E} \{ \|x_t\|_Q^2 \} &= \mathbb{E} \{ \|\bar{x}_t + \delta x_t\|_Q^2 \} = \mathbb{E} \left\{ (\bar{x}_t + \delta x_t)^T Q (\bar{x}_t + \delta x_t) \right\} = \\ &= \mathbb{E} \left\{ \bar{x}_t^T Q \bar{x}_t + \delta x_t^T Q \delta x_t + \underbrace{2\delta x_t^T Q \bar{x}_t}_{\mathbb{E}\{\delta x_t\}=0} \right\} = \mathbb{E} \left\{ \bar{x}_t^T Q \bar{x}_t + \delta x_t^T Q \delta x_t \right\} = \\ &= \bar{x}_t^T Q \bar{x}_t + \mathbb{E} \left\{ \delta x_t^T Q \delta x_t \right\} = \bar{x}_t^T Q \bar{x}_t + \text{tr} \left\{ \mathbb{E} \left\{ \delta x_t^T Q \delta x_t \right\} \right\}\end{aligned}$$

si noti che il primo termine è la norma euclidea di \bar{x}_t pesata da Q ; mentre al secondo termine si è inserito l'operatore di traccia, poiché la traccia di uno scalare è lo scalare stesso. Si consideri, quindi, il secondo termine e si porti all'interno del valore atteso l'operatore di traccia

$$\begin{aligned}\text{tr} \left\{ \mathbb{E} \left\{ \delta x_t^T Q \delta x_t \right\} \right\} &= \mathbb{E} \left\{ \text{tr} \left\{ \delta x_t^T Q \delta x_t \right\} \right\} = \\ \text{si ricorda che } \text{tr}(AB) &= \text{tr}(BA) \quad \rightarrow \quad = \text{tr} \left\{ \mathbb{E} \left\{ Q \delta x_t^T \delta x_t \right\} \right\} = \\ &= \text{tr} \left\{ Q \underbrace{\mathbb{E} \left\{ \delta x_t^T \delta x_t \right\}}_{X_t} \right\} = \text{tr} \{ Q X_t \} \\ \Rightarrow \mathbb{E} \{ \|x_t\|_Q^2 \} &= \mathbb{E} \{ \|\bar{x}_t\|_Q^2 \} + \text{tr} \{ Q X_t \}\end{aligned}$$

Allo stesso modo si ricava

$$\begin{aligned}\mathbb{E} \{ \|u_t\|_R^2 \} &= \mathbb{E} \{ \|\bar{u}_t\|_R^2 \} + \text{tr} \left\{ (K_t)^T R K_t X_t \right\} \\ \mathbb{E} \{ \|x_{t+N}\|_P^2 \} &= \mathbb{E} \{ \|\bar{x}_{t+N}\|_P^2 \} + \text{tr} \{ P X_{t+N} \}\end{aligned}$$

È quindi possibile riscrivere la funzione di costo come somma di due termini: uno dipendente dalla sequenza dei valori attesi della variabile d'ingresso lungo l'orizzonte, $\bar{u}_{t\dots t+N-1}$ e dal valore atteso dello stato all'istante iniziale, \bar{x}_t ; l'altro dalla sequenza dei guadagni $K_{t\dots t+N-1}$ lungo l'orizzonte e dal valore iniziale della varianza dello stato X_t . Si noti come in base alle legge di controllo (2.21) l'ingresso \bar{u}_t può essere usato per controllare il valor medio dello stato, mentre il guadagno K_t è scelto per ridurre il più possibile la varianza dello stato. Si ottiene quindi:

$$J(t) = J_m(\bar{x}_t, \bar{u}_{t\dots t+N-1}) + J_v(X_t, K_{t\dots t+N-1}) \quad (2.39)$$

dove in particolare:

$$J_m = \sum_{k=t}^{t+N-1} \|\bar{x}_k\|_Q^2 + \|\bar{u}_k\|_R^2 + \|\bar{x}_{t+N}\|_P^2 \quad (2.40)$$

$$J_v = \sum_{k=t}^{t+N-1} tr \left\{ (Q + (K_k)^T R K_k) X_k \right\} + tr \{ P X_{t+N} \} \quad (2.41)$$

dove tr è l'operatore traccia.

Vincolo terminale

Un ruolo importante nell'ambito del controllo predittivo è assunto dal vincolo terminale il quale, come già detto, è necessario al fine di garantire proprietà di *recursive feasibility* del sistema. Anche in questo caso i vincoli vengono formalizzati rispetto al valore atteso e alla varianza dello stato. In particolare si richiede che, alla fine dell'orizzonte di predizione $t + N$, il valore atteso dello stato appartenga ad uno specifico set e la varianza sia contenuta. I vincoli sono quindi:

$$\bar{x}_{t+N} \in \bar{\mathbb{X}}^f \quad (2.42)$$

$$X_{t+N} \leq \bar{X} \quad (2.43)$$

Si introduce quindi una legge ausiliaria $\bar{u}_t = -\bar{K}\bar{x}_t$ per il sistema (2.22) che garantisca che $(A - B\bar{K})$ sia stabile. Il set $\bar{\mathbb{X}}^f$ è scelto essere positivamente invariante per il sistema ad anello chiuso $\bar{x}_{t+1} = (A - B\bar{K})\bar{x}_t$. Quindi deve valere:

$$(A - B\bar{K})\bar{x} \in \bar{\mathbb{X}}^f, \quad \forall \bar{x} \in \bar{\mathbb{X}}^f \quad (2.44)$$

La matrice di covarianza \bar{X} viene definita come soluzione di regime dell'equazione di *Lyapunov* e calcolata come segue:

$$(A - B\bar{K})\bar{X}(A - B\bar{K})^T - \bar{X} = -F\bar{W}F^T \quad (2.45)$$

dove $\bar{W} \geq W$.

Inoltre devono essere soddisfatti anche i vincoli definiti in (2.37) e (2.38).

In particolare, ponendo \bar{X} e $\bar{U} = \bar{K}\bar{X}\bar{K}^T$ al posto di X, U :

$$(b_r)^T \bar{x}_t \leq 1 - \sqrt{(b_r)^T \bar{X} b_r} f(p_r^x) \quad (2.46)$$

$$(c_s)^T \bar{u}_t \leq 1 - \sqrt{(c_s)^T \bar{U} c_s} f(p_s^u) \quad (2.47)$$

per ogni $\bar{x}_t \in \mathbb{X}^f$ e $\bar{u}_t \in -\bar{K}\mathbb{X}^f$.

Condizioni iniziali dell'algorithm

L'algorithm MPC stocastico illustrato, come già detto, ha lo scopo di risolvere l'ottimizzazione vincolata rispetto non solo alla sequenza degli ingressi e dei guadagni, ma anche rispetto alle condizioni iniziali dello stato (\bar{x}_t, \bar{X}_t) . È quindi necessario inizializzare lo stato del sistema ogni volta prima di risolvere l'ottimizzazione.

Al fine di utilizzare tutta l'informazione disponibile sullo stato, ad ogni istante di tempo, la scelta migliore sarebbe quella di inizializzare il valore corrente dello stato nominale \bar{x}_t con il valore dello stato vero x_t , inizializzando di conseguenza la varianza X_t a zero.

Tuttavia si è supposto che sul sistema possa agire un disturbo illimitato. In particolare, nel caso in cui agisca un tale disturbo, l'inizializzazione potrebbe portare ad un problema di ottimizzazione non *feasible*. Ciononostante, i vincoli terminali (2.42) e (2.43) garantiscono *recursive feasibility* nel caso in cui si utilizzino come inizializzazioni, per il valore atteso dello stato $\bar{x}_{t|t}$ e la sua varianza $X_{t|t}$, i valori ottenuti al passo precedente $\bar{x}_{t|t-1}, X_{t|t-1}$, aggiornati secondo le equazioni dinamiche (2.22) e (2.25).

A fronte di queste considerazioni, si definiscono quindi le due strategie alternative di inizializzazione:

- **Strategia 1** - *Reset dello stato*: $(\bar{x}_{t|t}, X_{t|t}) = (x_t, 0)$
- **Strategia 2** - *Predizione dello stato*: $(\bar{x}_{t|t}, X_{t|t}) = (\bar{x}_{t|t-1}, X_{t|t-1})$

e attraverso una selezione binaria, verrà opportunamente scelta la strategia di inizializzazione.

2.3.4 Formulazione di MPC stocastico per la regolazione

A questo punto è possibile formulare il problema di controllo predittivo di tipo stocastico (s-MPC) come segue:

ad ogni istante di tempo t si risolve il seguente problema

$$\begin{aligned}
& \min_{\bar{u}_{t\dots t+N-1}, K_{t\dots t+N-1}, \bar{x}_t, X_t} J_m(\bar{x}_t, \bar{u}_{t\dots t+N-1}) + J_v(X_t, K_{t\dots t+N-1}) \\
& \bar{x}_{t+1} = A\bar{x}_t + B\bar{u}_t \\
& X_{t+1} = (A - BK_t)X_t(A - BK_t)^T + FWF^T \\
& (b_r)^T \bar{x}_{t+k} \leq 1 - \sqrt{(b_r)^T X_{t+k} b_r} f(p_r^x), \text{ con } k = 0, \dots, N-1 \\
& (c_s)^T \bar{u}_{t+k} \leq 1 - \sqrt{(c_s)^T U_{t+k} c_s} f(p_s^u), \text{ con } k = 0, \dots, N-1 \\
& \bar{x}_{t+N} \in \bar{X}_f \\
& X_{t+N} \leq \bar{X} \\
& (\bar{x}_{t|t}, X_{t|t}) \in \{(x_t, 0), (\bar{x}_{t|t-1}, X_{t|t-1})\}
\end{aligned}$$

Il risultato dell'ottimizzazione è la sequenza ottima degli ingressi $\bar{u}_{t\dots t+N-1|t}$, dei guadagni $K_{t\dots t+N-1|t}$, il valore atteso $\bar{x}_{t|t}$ e la varianza $X_{t|t}$ dello stato. A questo punto, in accordo con il principio Receding Horizon, solo il primo ingresso, \bar{u}_t , viene utilizzato nella legge di controllo (2.21), da cui si ricava l'ingresso reale per il sistema (2.15), come $\bar{u}_t = \bar{u}_{t|t} - \bar{K}_{t|t}(x_t - \bar{x}_{t|t})$.

Come detto precedentemente, in questa trattazione verrà implementato un algoritmo in cui l'ottimizzazione è risolta rispetto alla sequenza degli ingressi $\bar{u}_{t\dots t+N-1}$, al valore atteso \bar{x}_t e alla varianza dello stato X_t all'istante iniziale e non rispetto al guadagno, in quanto quest'ultimo è definito costante e pari al guadagno della legge ausiliaria. In particolare il problema di ottimizzazione risulta essere:

$$\min_{\bar{u}_{t\dots t+N-1}, \bar{x}_t, X_t} J_m(\bar{x}_t, \bar{u}_{t\dots t+N-1}) + J_v(X_t)$$

$$\begin{aligned}\bar{x}_{t+1} &= A\bar{x}_t + B\bar{u}_t \\ X_{t+1} &= (A - B\bar{K})X_t(A - B\bar{K})^T + FWF^T \\ (b_r)^T \bar{x}_{t+k} &\leq 1 - \sqrt{(b_r)^T X_{t+k} b_r} f(p_r^x), \text{ con } k = 0, \dots, N-1 \\ (c_s)^T \bar{u}_{t+k} &\leq 1 - \sqrt{(c_s)^T U_{t+k} c_s} f(p_s^u), \text{ con } k = 0, \dots, N-1 \\ \bar{x}_{t+N} &\in \bar{X}_f \\ X_{t+N} &\leq \bar{X} \\ (\bar{x}_{t|t}, X_{t|t}) &\in \{(x_t, 0), (\bar{x}_{t|t-1}, X_{t|t-1})\}\end{aligned}$$

Si noti che, in questo caso, il termine J_v , anche se non dipende più dalla sequenza dei guadagni $K_{t\dots t+N-1}$, non può essere tralasciato, poiché dipende ancora dal valore iniziale della varianza. Inoltre, tale termine, se si usa la **Strategia 1** è calcolabile *off line*. D'altra parte con la **Strategia 2**, è necessario ricalcolarlo *on line* ad ogni iterazione.

Capitolo 3

Controllo predittivo stocastico: estensione al problema di inseguimento in un contesto distribuito

Nel capitolo 2 è stato presentato un algoritmo di controllo predittivo stocastico per la regolazione a zero dello stato di un sistema in un contesto centralizzato. Al fine di proporre un algoritmo che possa gestire sistemi di larga scala e che possa evolvere verso stati differenti dall'origine, nel presente capitolo si estende tale procedura ad un controllo distribuito di sistemi interagenti per l'inseguimento di segnali di riferimento costanti.

Inizialmente si proporrà una soluzione di un problema di *tracking* per il controllo di un sistema in un contesto centralizzato. Dopodiché, l'algoritmo ottenuto verrà esteso ad un caso distribuito, per la gestione coordinata di più sistemi indipendenti ma connessi tramite vincoli di accoppiamento.

3.1 Controllo predittivo stocastico per l'inseguimento

In molti casi applicativi ci si trova di fronte a problemi in cui il sistema debba evolvere verso stati differenti, non corrispondenti all'origine del sistema di riferimento o ancor più in generale debba inseguire delle traiettorie opportune. In tutti questi casi il problema che ci si propone di affrontare riguarda il ben noto controllo per il *tracking*.

La tecnica standard per risolvere un problema di *tracking* in un contesto MPC prevede di ridefinire il problema in un sistema di riferimento traslato rispetto a quello di partenza, dove l'origine corrisponde al punto di arrivo. Inoltre, in generale, si richiede che lo stato, in N passi, si possa trovare nell'insieme positivamente invariante opportunamente definito nell'intorno dello stato di arrivo. Tuttavia un orizzonte di controllo limitato (per ragioni computazionali) e vincoli di tipo *hard* imposti sullo stato terminale, possono causare il non raggiungimento del target, limitando la regione di *feasibility* del problema in modo, spesso, inaccettabile. Questo problema può essere risolto rendendo flessibile la scelta dell'orizzonte di controllo N e ricalcolando *on line* il set terminale per il nuovo target di regime. Tuttavia un tale algoritmo porta con sé una elevata complessità e un ricalcolo *on-line* potrebbe richiedere tempi eccessivamente lunghi, facendo diventare il controllo del sistema insostenibile. Una soluzione a tale problema è discussa in [2] per l'inseguimento di riferimenti costanti. L'idea ivi proposta consiste nel considerare ad ogni passo di campionamento un target artificiale che garantisca *recursive feasibility*. In particolare si introduce una nuova variabile decisionale nel problema di ottimizzazione, rappresentante il suddetto target artificiale. Nel funzionale di costo si minimizza l'errore quadratico tra i valori artificiali e quelli desiderati di stato e ingresso. Si considera infine un vincolo terminale esteso basato sulla definizione di set invariante per il *tracking*.

Il controllore permette di portare il sistema verso la soluzione di regime am-

missibile più vicina rispetto all'obiettivo. Teoricamente l'aggiunta di una variabile decisionale, e quindi di un grado di libertà in più, potrebbe far perdere ottimalità all'algoritmo. Ciononostante, la perdita di ottimalità può essere ridotta penalizzando opportunamente la differenza tra il target artificiale e quello desiderato.

In questa tesi viene presentato tale algoritmo, quindi rivisitato e adattato ad un caso stocastico.

Si consideri il sistema nominale (2.22), dove viene definita la variabile nominale di uscita \bar{z}_t tale che:

$$\bar{z}_t = C\bar{x}_t \quad (3.1)$$

dove $\bar{z}_t \in \mathbb{R}^m$.

Ci si propone quindi di progettare uno schema di controllo in retroazione, basato su MPC, per l'inseguimento di un segnale riferimento $z_{target} \in \mathbb{R}^m$. Si vuole, quindi, portare l'uscita del sistema \bar{z}_t , asintoticamente, al valore desiderato z_{target} rispettando i vincoli.

Si consideri ora una generica variabile di uscita di target z_{target} . Ogni coppia di variabili stato e ingresso $(\bar{x}_{target}, \bar{u}_{target})$ associate a questo target deve soddisfare l'equazione

$$\underbrace{\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix}}_S \begin{bmatrix} \bar{x}_{target} \\ \bar{u}_{target} \end{bmatrix} = \begin{bmatrix} 0 \\ z_{target} \end{bmatrix} \quad (3.2)$$

L'equazione scritta richiede ovviamente che $rank(S) = n+m$. Questa ipotesi garantisce che l'equazione (3.2) abbia una soluzione non banale, cioè:

$$\begin{bmatrix} \bar{x}_{target} \\ \bar{u}_{target} \end{bmatrix} = M z_{target} \quad (3.3)$$

dove

$$M = S^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (3.4)$$

3.1.1 Legge ausiliaria di controllo e calcolo del set invariante per il *tracking*

Dato un generico *set-point* \tilde{z} per la variabile di uscita, i *set-point* per gli stati e gli ingressi sono calcolati come:

$$\begin{bmatrix} \tilde{x} \\ \tilde{u} \end{bmatrix} = M\tilde{z} \quad (3.5)$$

Si definisce la legge di controllo ausiliaria in *feedback* per il sistema (2.22):

$$\bar{u}_t = -\bar{K}(\bar{x}_t - \tilde{x}) + \tilde{u} = -\bar{K}\bar{x}_t + L\tilde{z} \quad (3.6)$$

dove $L = [\bar{K} \quad I]M$.

Ora, se la matrice $A - B\bar{K}$ è Hurwitz, allora il sistema in anello chiuso evolve verso lo stato di riferimento (\tilde{x}, \tilde{u}) . Un *admissible invariant set* per il *tracking* corrisponde all'insieme di stati iniziali e stati e ingressi di regime (\tilde{x}, \tilde{u}) che possono essere stabilizzati dalla legge di controllo in cui le variabili di stato e ingresso rispettano vincoli operativi.

Per quanto già detto, la variabile \tilde{z} è un grado di libertà aggiuntivo del sistema. Se si definisce quindi lo stato allargato come $\xi = (\bar{x}, \tilde{z})$, combinando le equazioni (2.22) e (3.6), il sistema in anello chiuso evolve secondo l'equazione:

$$\xi_{t+1} = \bar{A}\xi_t \quad (3.7)$$

dove

$$\bar{A} = \begin{bmatrix} A - B\bar{K} & BL \\ 0 & I \end{bmatrix} \quad (3.8)$$

Sia W_λ un poliedro convesso, definito come:

$$W_\lambda = \{\xi = (\bar{x}, \tilde{z}) : \bar{x} \in \bar{\mathbb{X}}, \bar{u} = -\bar{K}\bar{x} + L\tilde{z} \in \bar{\mathbb{U}}, M\tilde{z} \in \lambda\bar{\mathbb{X}} \times \lambda\bar{\mathbb{U}}\} \quad (3.9)$$

dove $\lambda \in (0, 1)$ è un parametro scelto in fase di progetto, prossimo a 1 e $\bar{\mathbb{X}}$ e $\bar{\mathbb{U}}$ definiscono insiemi convessi, contenenti l'origine e politopici, definiti

come segue:

$$\bar{\mathbb{X}} = \left\{ \bar{x} : (b_r)^T \bar{x} \leq 1 - \sqrt{(b_r)^T \bar{X} b_r} f(p_r^x), r = 1, \dots, n_x \right\} \quad (3.10)$$

$$\bar{\mathbb{U}} = \left\{ \bar{u} : (c_s)^T \bar{u} \leq 1 - \sqrt{(c_s)^T \bar{U} c_s} f(p_s^u), s = 1, \dots, n_u \right\} \quad (3.11)$$

Si dice, quindi, che un insieme Ω^ξ è un *admissible invariant set* se:

$$\forall \xi \in \Omega^\xi \Rightarrow \bar{A}\xi \in \Omega^\xi, \Omega^\xi \subseteq W_\lambda \quad (3.12)$$

È quindi possibile definire il *maximal admissible invariant set* per il *tracking* come segue:

$$\mathbb{O}_{\infty,1} = \{ \xi = (\bar{x}, \tilde{z}) : \bar{A}^k \xi \in \mathbb{W}_1, \forall k \geq 0 \} \quad (3.13)$$

Tuttavia, la matrice \bar{A} non è Hurwitz, a causa della presenza di autovalori in 1. Questo potrebbe rendere il *maximal admissible invariant set* non costituito da un insieme finito di vincoli, non risultando calcolabile in un numero finito di passi. Ciononostante è possibile scegliere il seguente set:

$$\mathbb{O}_{\infty,\lambda}^\xi = \left\{ \xi : \bar{A}^k \xi \in \mathbb{W}_\lambda, \forall k \geq 0 \right\} \quad (3.14)$$

che risulta essere ben determinato per ogni valore di $\lambda \in (0, 1)$.

Per maggiori chiarimenti sul calcolo del *admissible invariant set* si rimanda la trattazione a [7].

3.1.2 Formulazione di MPC stocastico per il *tracking*

In questa sezione ci si propone di definire il problema di MPC per il *tracking*. Come già detto, il controllo si basa sull'aggiunta di una nuova variabile decisionale. Inoltre, tramite una funzione di costo modificata, rispetto a quella del tradizionale MPC, si minimizza la deviazione tra i valori di stato e ingresso veri con quelli di riferimento calcolati in (3.5). Infine si considera un vincolo terminale esteso, per la nuova variabile (x, \tilde{z}) .

In particolare, la funzione di costo viene riformulata in termini del valore atteso dello stato come segue:

$$J_t = \mathbb{E} \left\{ \sum_{k=0}^{N-1} (\|x_{t+k} - \tilde{x}_t\|_Q^2 + \|u_{t+k} - \tilde{u}_t\|_R^2) + \|x_{t+N} - \tilde{x}_t\|_P^2 \right\} + \|\tilde{z}_t - \tilde{z}_G\|_T \quad (3.15)$$

dove z_G rappresenta il target, e \tilde{z}_t rappresenta il target “virtuale” migliore che può essere definito al tempo t . Inoltre come in (3.1) si calcolano:

$$\begin{bmatrix} \tilde{x}_t \\ \tilde{u}_t \end{bmatrix} = M\tilde{z}_t \quad (3.16)$$

A questo punto, come già fatto nella Sezione 2.3.3, si riscrive tale funzione di costo come somma dei seguenti termini:

$$J_t = J_m(\bar{x}_t, \bar{u}_{t\dots t+N-1}, \tilde{z}_t) + J_v(X_t) \quad (3.17)$$

dove:

$$J_m = \sum_{k=t}^{t+N-1} (\|\bar{x}_k - \tilde{x}_t\|_Q^2 + \|\bar{u}_k - \tilde{u}_t\|_R^2) + \|\bar{x}_{t+N} - \tilde{x}_t\|_P^2 + \|\tilde{z}_t - \tilde{z}_G\|_T \quad (3.18)$$

$$J_v = \sum_{k=t}^{k+N-1} tr \left\{ (Q + (\bar{K})^T R \bar{K}) X_t \right\} + tr \{ P X_{t+N} \} \quad (3.19)$$

Sia $\bar{\xi} = (\bar{x}, \tilde{z})$, il nuovo vettore di stato. Il problema è quindi riformulato nel modo seguente:

$$\begin{aligned} & \min_{\bar{u}_{t\dots t+N-1}, \bar{x}_t, X_t, \tilde{x}_t} J_m(\bar{x}_t, \bar{u}_{t\dots t+N-1}, \tilde{z}_t) + J_v(X_t) \\ & \bar{x}_{t+1} = A\bar{x}_t + B\bar{u}_t \\ & X_{t+1} = (A - BK)X_t(A - BK)^T + FWF^T \\ & (b_r)^T \bar{x}_{t+k} \leq 1 - \sqrt{(b_r)^T X_{t+k} b_r} f(p_r^x), \quad \text{con } k = 0, \dots, N-1 \\ & (c_s)^T \bar{u}_{t+k} \leq 1 - \sqrt{(c_s)^T U_{t+k} c_s} f(p_s^u), \quad \text{con } k = 0, \dots, N-1 \end{aligned}$$

$$\begin{aligned}
(\bar{x}_{t+N}, \tilde{z}_t) &\in \mathbb{O}_{\infty, \lambda}^{(\bar{x}_{t+N}, \tilde{z}_t)} \\
X_{t+N} &\leq \bar{X} \\
(\bar{x}_{t|t}, X_{t|t}) &\in \{(x_t, 0), (\bar{x}_{t|t-1}, X_{t|t-1})\}
\end{aligned}$$

3.2 Controllo predittivo distribuito stocastico per il *tracking*

Al giorno d'oggi specifiche richieste di mercato richiedono la realizzazione di impianti complessi su larga scala. In questo caso il progetto di un sistema di controllo, fondamentale per il corretto funzionamento dell'impianto stesso, risulta essere decisamente più complesso. Sono state sviluppate negli ultimi decenni differenti teorie per la gestione ed il controllo di impianti di larga scala in modo distribuito. L'idea che sta alla base di tutte queste si basa sulla suddivisione del sistema complesso in un insieme di sotto-sistemi più semplici, ognuno dei quali è controllato da un opportuno regolatore, che possono scambiarsi informazioni. Sebbene la scelta ottimale sarebbe quella di ricorrere a sistemi di controllo di tipo centralizzato (in cui un unico controllore regola e coordina tutti i sotto-sistemi, in grado di considerare le interazioni tra i sotto-sistemi stessi, i quali possono avere dinamiche fortemente accoppiate o essere connessi da vincoli opportuni), tuttavia, tale scelta può comportare problemi di progetto troppo complessi e può richiedere sforzi computazionali notevoli, rendendone impossibile l'implementazione. Per questo motivo si possono intraprendere differenti strade. Una prima riguarda il controllo decentralizzato dei sotto-sistemi, dove ogni singolo sistema è dotato di un proprio controllore, il quale non prevede, però, la comunicazione con gli altri sotto-sistemi. In questo modo si riduce un problema complesso in un insieme di problemi più semplici. Questo approccio

si presta all' applicazione di sistemi facilmente disaccoppiabili, dove si trascurano le interazioni fra i sotto-sistemi stessi. Sono però pochi i sistemi perfettamente disaccoppiabili, perché un minimo di interazione è sempre presente e ignorarla spesso non è possibile.

Una seconda strada si basa su tecniche di tipo distribuito, dove ogni sotto-sistema è dotato di un proprio controllore, il quale però, a differenza del semplice controllo decentralizzato, può scambiare informazioni con gli altri. Questa metodologia di controllo si pone a metà strada tra i controlli centralizzati e decentralizzati.

Gli approcci di tipo distribuito sono spesso utilizzati in contesti di controllo MPC. In particolare ogni sottosistema risolve un problema di ottimizzazione vincolato, come quello già presentato precedentemente, e scambia informazioni con gli altri.

È importante notare che tale approccio può comportare soluzioni subottime rispetto alla soluzione del problema in un contesto centralizzato.

Si consideri un sistema composto da M sotto-sistemi indipendenti Σ_i , con $i = 1, \dots, M$, ognuno dei quali descritto dalla equazione:

$$\begin{cases} x_{t+1}^{[i]} = A_i x_t^{[i]} + B_i u_t^{[i]} + F_i w_t^{[i]} \\ z_t^{[i]} = C_i x_t^{[i]} \end{cases} \quad (3.20)$$

Per ognuno dei sotto-sistemi si suppone valgano tutte le ipotesi fatte per (2.15). Si suppone, inoltre, che $w_t^{[i]}$ sia incorrelato con $w_t^{[j]}$ per tutti $i \neq j$. Scopo di questa ultima sezione è sviluppare uno schema di controllo MPC stocastico per il *tracking* di tipo distribuito che garantisca il soddisfacimento di vincoli sia locali che collettivi.

3.2.1 Vincoli Collettivi

Rispetto all' algoritmo MPC stocastico già discusso, l'estensione ad un caso distribuito la si ottiene inserendo nel problema di ottimizzazione un vincolo

aggiuntivo relativo allo stato di più sotto-sistemi contemporaneamente. Il vincolo aggiuntivo ha la stessa forma dei vincoli (2.17):

$$\mathcal{P} \left\{ \sum_{i=1}^M \left(d_l^{[i]} \right)^T x_t^{[i]} \geq 1 \right\} \leq p_l^c, \text{ con } l = 1, \dots, n_c \quad (3.21)$$

Anche in questo caso, mediante la disuguaglianza di Cantelli-Chebyshev, il vincolo viene riformulato in termini deterministici. In particolare tale vincolo, *soft* rispetto alla variabile vera dello stato, diviene *hard* rispetto al valore atteso e alla varianza dello stato:

$$\sum_{i=1}^M \left(d_l^{[i]} \right)^T \bar{x}_t^{[i]} \leq 1 - \sqrt{\sum_{i=1}^M \left(d_l^{[i]} \right)^T X_t^{[i]} d_l^{[i]} f(p_l^c)} \quad (3.22)$$

Come si può facilmente notare, il vincolo (3.22) include le variabili decisionali di più di un sottosistema e così formulato può essere incluso solamente in un problema MPC di tipo centralizzato. Come detto, però, si è optato per un approccio di tipo distribuito e quindi questo vincolo non può essere incluso nel problema di ottimizzazione in questo modo ma risulta necessario riformularlo.

In particolare ogni sotto-sistema, nel proprio problema di ottimizzazione, deve includere un vincolo relativo allo stato del proprio sotto-sistema, tenendo eventualmente conto dello stato dei sotto-sistemi vicini.

Si supponga che, per ogni $i = 1, \dots, M$, al tempo t , sia nota la traiettoria dello stato generata dalla soluzione del problema MPC all'istante precedente relativa agli altri sotto-sistemi con cui lui stesso è accoppiato, tramite un vincolo del tipo (3.22). In questo modo, ogni sottosistema implicato in (3.22) può calcolare

$$\delta_l(t+k|t-1) = 1 - \sum_{i=1}^M \left(d_l^{[i]} \right)^T \bar{x}_{t+k|t-1}^{[i]} - \sqrt{\sum_{i=1}^M \left(d_l^{[i]} \right)^T X_{t+k|t-1}^{[i]} d_l^{[i]} f(p_l^c)} \quad (3.23)$$

che rappresenta, se positivo, la “distanza” dalla violazione del vincolo, che si avrebbe se tutti i sotto-sistemi adottassero le azioni di controllo calcolate

al tempo $t - 1$, cioè $u_{t+k|t-1}^{[i]}$, con $k = t, \dots, t + N - 2$.

L'idea sfruttata è quella di permettere, alla nuova traiettoria predetta all'istante corrente, da ogni sotto-sistema, di "avvicinarsi" alla violazione del vincolo di un valore corrispondente a una frazione di $\delta_l(t + k|t - 1)$, ed in particolare di $\frac{1}{\nu_l} \delta_l(t + k|t - 1)$, dove ν_l è il numero di sotto-sistemi coinvolti nell' l -esimo vincolo di accoppiamento. In tal modo, se anche tutti i sotto-sistemi si avvicinassero alla "soglia" del vincolo di quanto è concesso, il vincolo non risulterebbe violato. Allo stesso tempo si richiede che la varianza dello stato predetta all'istante corrente non sia maggiore di quella calcolata (e trasmessa) all'istante $t - 1$. Tutto questo si traduce matematicamente nei seguenti vincoli:

$$\left(d_l^{[i]}\right)^T \left(\bar{x}_{t+k}^{[i]} - \bar{x}_{t+k|t-1}^{[i]}\right) \leq \frac{1}{\nu_l} \delta_l(t + k|t - 1) \quad (3.24)$$

$$X_{t+k}^{[i]} \leq X_{t+k|t-1}^{[i]} \quad (3.25)$$

Relativamente a (3.25), si osserva che risulta

$$\sqrt{\sum_{i=1}^M \left(d_l^{[i]}\right)^T X_{t+k}^{[i]} d_l^{[i]} f(p_l^c)} \leq \sqrt{\sum_{i=1}^M \left(d_l^{[i]}\right)^T X_{t+k|t-1}^{[i]} d_l^{[i]} f(p_l^c)}, \quad k = 0, \dots, N-1 \quad (3.26)$$

In Figura 3.1 è proposta una illustrazione relativa all'algoritmo distribuito appena descritto. In particolare, i due sotto-sistemi sono due agenti liberi di muoversi nel piano cartesiano. I due agenti sono stati approssimati a politopi con numero di lati pari a otto. All'istante $t - 1$ viene calcolata la distanza δ tra i due agenti, lungo tutto l'orizzonte di predizione. All'istante successivo, ogni agente, al fine di evitare collisione, potrà al più muoversi di $\frac{\delta}{2}$.

3.2.2 Riformulazione del *set* invariante per il *tracking*

A questo punto è necessario riformulare i *set* invarianti per il *tracking* in un caso distribuito. In particolare il *set* definito in (3.9) deve essere riformulato per ogni sottosistema, in cui si tenga conto sia di vincoli locali che vincoli

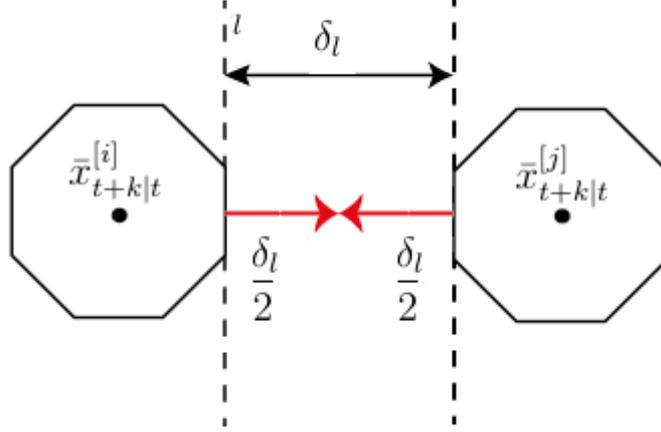


Figura 3.1: Gestione dei vincoli di accoppiamento in contesto distribuito

collettivi.

Si definisce, quindi, per $i = 1, \dots, M$

$$\bar{\mathbb{X}}^{[i]} := \left\{ \bar{x}^{[i]} : (b_r)^T \bar{x}^{[i]} \leq 1 - \sqrt{(b_r)^T \bar{X}^{[i]} b_r} f(p_r^x), r = 1, \dots, n_x \right\} \quad (3.27)$$

$$\bar{\mathbb{U}}^{[i]} := \left\{ \bar{u}^{[i]} : (c_s)^T \bar{u}^{[i]} \leq 1 - \sqrt{(c_s)^T \bar{U}^{[i]} c_s} f(p_s^u), s = 1, \dots, n_u \right\} \quad (3.28)$$

I *set* (3.27) e (3.28) sono relativi ai vincoli locali e sono definiti in modo “decentralizzato”. Per ogni agente, infatti, tali *set* sono automaticamente definiti senza l’ausilio di informazioni relative ad altri agenti. In aggiunta è necessario definire un *set* che tenga conto di tutti i sotto-sistemi, il quale viene formulato in maniera “centralizzata”. Si definisce quindi un insieme di *set* del tipo:

$$\left\{ \hat{\mathbb{X}}^{[1]}, \dots, \hat{\mathbb{X}}^{[M]} \right\}$$

dove tutti i *set* $\hat{\mathbb{X}}^{[i]}$ devono essere definiti contemporaneamente in modo tale che:

$$\sum_{i=1}^M \left(d_i^{[i]} \right)^T \bar{x}^{[i]} \leq 1 - \sqrt{\sum_{i=1}^M \left(d_i^{[i]} \right)^T \bar{X}^{[i]} d_i^{[i]} f(p_i^c)}, \quad \forall l, \forall \bar{x}^{[i]} \in \hat{\mathbb{X}}^{[i]}, \forall i = 1, \dots, M \quad (3.29)$$

Si noti che esistono infinite possibili scelte del set $\{\hat{\mathbb{X}}^{[1]}, \dots, \hat{\mathbb{X}}^{[M]}\}$ tale che valga la (3.29).

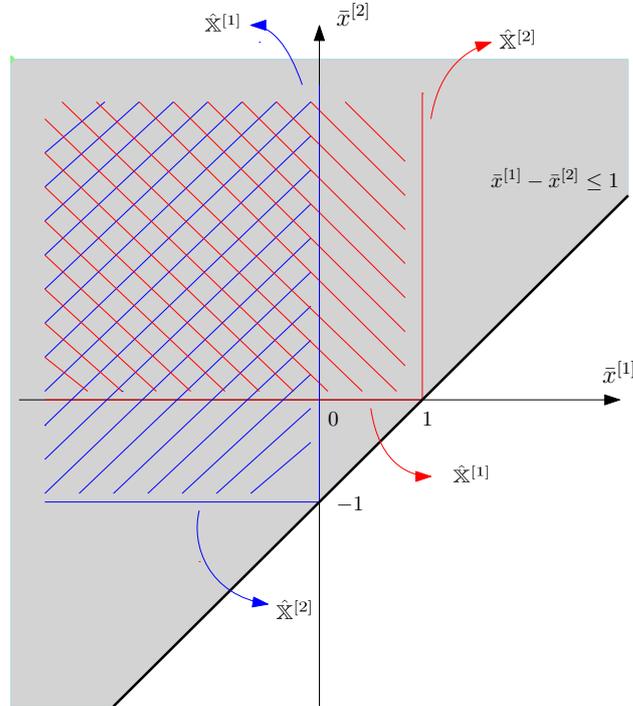


Figura 3.2: Partizione arbitraria dello spazio

Viene mostrato ora un esempio (Figura 3.2) che mostra come sia arbitraria la scelta del set che verifica il vincolo collettivo. In particolare si illustra il caso in cui il vincolo collettivo da rispettare sia $\bar{x}^{[1]} - \bar{x}^{[2]} \leq 1$ (semipiano in grigio). Relativamente a questo caso in due dimensioni, si devono scegliere arbitrariamente i set $\hat{\mathbb{X}}^{[1]}$ e $\hat{\mathbb{X}}^{[2]}$ in modo tale che il set $\hat{\mathbb{X}}^{[1]} \times \hat{\mathbb{X}}^{[2]}$ risulti essere contenuto all'interno della regione definita dal vincolo $\bar{x}^{[1]} - \bar{x}^{[2]} \leq 1$. Una possibile scelta è $\hat{\mathbb{X}}^{[1]} = \{\bar{x}^{[1]} : \bar{x}^{[1]} \leq 1\}$ e $\hat{\mathbb{X}}^{[2]} = \{\bar{x}^{[2]} : \bar{x}^{[2]} \geq 0\}$. Si verifica facilmente che il prodotto cartesiano $\hat{\mathbb{X}}^{[1]} \times \hat{\mathbb{X}}^{[2]}$ (regione di piano in rosso) soddisfa il vincolo.

D'altra parte, una seconda possibile scelta corrisponde a $\hat{\mathbb{X}}^{[1]} = \{\bar{x}^{[1]} : \bar{x}^{[1]} \leq 0\}$ e $\hat{\mathbb{X}}^{[2]} = \{\bar{x}^{[2]} : \bar{x}^{[2]} \geq -1\}$. Anche in questo caso il prodotto cartesiano $\hat{\mathbb{X}}^{[1]} \times \hat{\mathbb{X}}^{[2]}$ (regione di piano in blu) soddisfa il vincolo collettivo.

A questo punto in modo naturale si definiscono i set $W_\lambda^{[i]}$ e $\mathbb{O}_{\infty,\lambda}^{[\xi^{[i]}]}$

$$W_\lambda^{[i]} = \left\{ (\bar{x}^{[i]}, \tilde{z}^{[i]}) : \bar{x}^{[i]} \in \left(\bar{\mathbb{X}}^{[i]} \cap \hat{\mathbb{X}}^{[i]} \right), \bar{u}^{[i]} = -\bar{K}\bar{x}^{[i]} + L\tilde{z}^{[i]} \in \bar{\mathbb{U}}^{[i]} \right. \\ \left. M\tilde{z}^{[i]} \in \lambda \left(\bar{\mathbb{X}}^{[i]} \cap \hat{\mathbb{X}}^{[i]} \right) \times \lambda \bar{\mathbb{U}}^{[i]} \right\}$$

e l'*admissible invariant set* è definito come

$$\mathbb{O}_{\infty,\lambda}^{[\xi^{[i]}]} = \left\{ \xi^{[i]} : \bar{A}^{[k]}\xi^{[i]} \in W_\lambda^{[i]}, \forall k \geq 0 \right\} \quad (3.30)$$

Il vincolo terminale da inserire nel problema di ottimizzazione è quindi

$$\left(\bar{x}_{t+N}^{[i]}, \tilde{z}_t^{[i]} \right) \in \mathbb{O}_{\infty,\lambda}^{[\xi^{[i]}]} \quad (3.31)$$

$$X_{t+N}^{[i]} \leq \bar{X}^{[i]} \quad (3.32)$$

Ad ogni istante di tempo t , ogni sotto-sistema i -esimo risolve il seguente problema di ottimizzazione soggetto a vincoli:

$$\min_{\bar{u}_{t \dots t+N-1}^{[i]}, \bar{x}_t^{[i]}, X_t^{[i]}, \tilde{x}_t^{[i]}} J_m(\bar{x}_t^{[i]}, \bar{u}_{t \dots t+N-1}^{[i]}, \tilde{z}_t^{[i]}) + J_v(X_t^{[i]})$$

$$\bar{x}_{t+1}^{[i]} = A\bar{x}_t^{[i]} + B\bar{u}_t^{[i]}$$

$$X_{t+1}^{[i]} = (A_i - B_i K)X_t^{[i]}(A_i - B_i K)^T + F_i W_i F_i^T$$

$$(b_r^{[i]})^T \bar{x}_{t+k}^{[i]} \leq 1 - \sqrt{(b_r^{[i]})^T X_{t+k}^{[i]} b_r^{[i]}} f(p_r^{x^{[i]}}), \quad \text{con } k = 0, \dots, N-1$$

$$(c_s^{[i]})^T \bar{u}_{t+k}^{[i]} \leq 1 - \sqrt{(c_s^{[i]})^T U_{t+k}^{[i]} c_s^{[i]}} f(p_s^{u^{[i]}}), \quad \text{con } k = 0, \dots, N-1$$

$$(d_t^{[i]})^T (\bar{x}_{t+k|t}^{[i]} - \bar{x}_{t+k|t-1}^{[i]}) \leq \frac{1}{\nu_t} \delta_l(t+k|t-1)$$

$$X_{t+k|t}^{[i]} \leq X_{t+k|t-1}^{[i]}$$

$$\left(\bar{x}_{t+N}^{[i]}, \tilde{z}_t^{[i]} \right) \in \mathbb{O}_{\infty,\lambda}^{(\bar{x}_{t+N}^{[i]}, \tilde{z}_t^{[i]})}$$

$$X_{t+N}^{[i]} \leq \bar{X}^{[i]}$$

$$\left(\bar{x}_{t|t}^{[i]}, X_{t|t}^{[i]} \right) \in \left\{ (x_t^{[i]}, 0), (\bar{x}_{t|t-1}^{[i]}, X_{t|t-1}^{[i]}) \right\}$$

Nell'ambito del controllo di una flotta di robot mobili il vincolo terminale definito in (3.31) potrebbe, tuttavia, risultare complesso da calcolare. Infatti, questa soluzione prevede di calcolare in modo predefinito lo spazio di arrivo degli agenti, e quindi richiede di partizionare lo spazio a priori in modo tale da garantire che ogni agente, dopo N passi, raggiunga una regione dello spazio precedentemente calcolata e che vi possa rimanere senza violare vincoli operativi. Questo metodo, quindi, richiede che, dopo N passi, sia già definita una regione invariante. Tuttavia questa definizione a priori dei *set* rende molto complesso il problema e inoltre tale soluzione di controllo non risulta essere particolarmente flessibile rispetto ad ambienti che variano nel tempo. Per questi motivi si propone una differente soluzione che non prevede la definizione a priori delle regioni invarianti. Si adotta una strategia del tipo *zero terminal constraint* per il il valore atteso dello stato al termine dell'orizzonte di predizione, quindi:

$$\bar{x}_{t+N}^{[i]} = [I \quad 0] M_i \tilde{z}_t^{[i]} \quad (3.33)$$

Il termine $\tilde{z}_t^{[i]}$ deve essere definito in modo da soddisfare i vincoli locali come definiti in (3.27) e (3.28). Si impone quindi

$$M_i \tilde{z}_t^{[i]} \in \lambda \left(\bar{\mathbb{X}}^{[i]} \times \bar{\mathbb{U}}^{[i]} \right) \quad (3.34)$$

dove $\lambda \in (0, 1)$. Il valore \tilde{z}_t deve anche essere definito in modo tale da garantire i vincoli collettivi. A tal fine si suppone sia nota la soluzione del problema di ottimizzazione al passo precedente, in particolare $\tilde{z}_{t-1|t-1}^{[i]}$ e in modo analogo a quanto espresso in (3.23) si definisce

$$\delta_l^{[z]}(t-1) = 1 - \sum_{i=1}^M \left(d_l^{[i]} \right)^T [I \quad 0] M_i \tilde{z}_{t-1|t-1}^{[i]} - \sqrt{\sum_{i=1}^M \left(d_l^{[i]} \right)^T \bar{X}^{[i]} d_l^{[i]} f(p_l^c)} \quad (3.35)$$

e si ricava il vincolo terminale

$$\left(d_l^{[i]} \right)^T [I \quad 0] M_1 \left(\tilde{z}_t^{[i]} - \tilde{z}_{t-1|t-1}^{[i]} \right) \leq \frac{1}{\nu_l} \delta_l^{[z]}(t-1) \quad (3.36)$$

In aggiunta il vincolo da imporre sulla varianza dello stato risulta essere

$$X_{t+N}^{[i]} \leq \bar{X}^{[i]} \quad (3.37)$$

In definitiva il problema di ottimizzazione da risolvere è

$$\begin{aligned} & \min_{\bar{u}_{t \dots t+N-1}^{[i]}, \bar{x}_t^{[i]}, X_t^{[i]}, \tilde{z}_t^{[i]}} J_m(\bar{x}_t^{[i]}, \bar{u}_{t \dots t+N-1}^{[i]}, \tilde{z}_t^{[i]}) + J_v(X_t^{[i]}) \\ & \bar{x}_{t+1}^{[i]} = A\bar{x}_t^{[i]} + B\bar{u}_t^{[i]} \\ & X_{t+1}^{[i]} = (A_i - B_i K)X_t^{[i]}(A_i - B_i K)^T + F_i W_i F_i^T \\ & (b_r^{[i]})^T \bar{x}_{t+k}^{[i]} \leq 1 - \sqrt{(b_r^{[i]})^T X_{t+k}^{[i]} b_r^{[i]}} f(p_r^{x[i]}), \quad \text{con } k = 0, \dots, N-1 \\ & (c_s^{[i]})^T \bar{u}_{t+k}^{[i]} \leq 1 - \sqrt{(c_s^{[i]})^T U_{t+k}^{[i]} c_s^{[i]}} f(p_s^{u[i]}), \quad \text{con } k = 0, \dots, N-1 \\ & (d_t^{[i]})^T (\bar{x}_{t+k|t}^{[i]} - \bar{x}_{t+k|t-1}^{[i]}) \leq \frac{1}{\nu_l} \delta_l(t+k|t-1) \\ & X_{t+k|t}^{[i]} \leq X_{t+k|t-1}^{[i]} \\ & \bar{x}_{t+N}^{[i]} = [I \quad 0] M_i \tilde{z}_t^{[i]} \\ & M_i \tilde{z}_t^{[i]} \in \lambda \left(\bar{\mathbb{X}}^{[i]} \times \bar{\mathbb{U}}^{[i]} \right) \\ & (d_t^{[i]})^T [I \quad 0] M_i \left(\tilde{z}_t^{[i]} - \tilde{z}_{t-1|t-1}^{[i]} \right) \leq \frac{1}{\nu_l} \delta_l^{[z]}(t-1) \\ & X_{t+N}^{[i]} \leq \bar{X}^{[i]} \\ & \left(\bar{x}_{t|t}^{[i]}, X_{t|t}^{[i]} \right) \in \left\{ (x_t^{[i]}, 0), (\bar{x}_{t|t-1}^{[i]}, X_{t|t-1}^{[i]}) \right\} \end{aligned}$$

Capitolo 4

Apparato Sperimentale

L'apparato sperimentale utilizzato per validare lo schema di controllo presentato in questa tesi è costituito da tre robot mobili liberi di muoversi in un piano di lavoro, di dimensioni 115cm × 66cm. Sopra tale piano è presente una telecamera (*webcam Microsoft LifeCam HD-6000 VX-800*), con lo scopo di osservare le posizioni e gli orientamenti assunti dai robot.

I robot utilizzati sono *e-puck*, si veda la Figura 4.1, nati da un progetto dell'*Ecole Polytechnique Fédérale de Lausanne* con lo scopo di svolgere esperimenti e ricerca in ambito di robotica autonoma, cooperativa ed interazione uomo-macchina. Gli *e-puck* sono robot mobili, di tipo “uniciclo”, di piccole dimensioni le cui caratteristiche principali sono semplicità d'uso e basso costo. Sono costituiti da una meccanica relativamente semplice e presentano una elettronica non banale e versatile ma allo stesso tempo non eccessivamente complessa. Infine, gli *e-puck* sono venduti con licenza *open hardware*, che significa che chiunque può costruirli o apportare modifiche ai codici sorgente, senza però averne l'esclusiva.

Ciascun robot è dotato di un dispositivo *bluetooth* che permette sia la programmazione che lo scambio di dati con altri dispositivi. L'attuazione è trasmessa al robot da due motori passo-passo, che consentono un controllo indipendente delle velocità di ogni singola ruota. Il robot dispone anche di

diversi sensori, tra cui una telecamera, sensori di prossimità ed accelerometri, che tuttavia non sono stati utilizzati in questa tesi. Per una maggiore descrizione dei robot *e-puck* si rimanda la trattazione a [6].



Figura 4.1: Robot *e-puck*

4.1 Modello del Sistema

Il modello utilizzato per descrivere il comportamento dinamico di questi robot mobili è chiamato *unicycle model* (modello ad uniciclo), molto noto in letteratura. Tale modello è descritto dalle seguenti equazioni

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (4.1)$$

Dal modello (4.1) si ricava facilmente che ad ogni istante sussiste un vincolo, detto anolonomo, descritto dall'equazione

$$\dot{x} \sin \theta = \dot{y} \cos \theta \quad (4.2)$$

L'equazione (4.2) vincola la velocità del robot ad avere direzione, in ogni istante, perpendicolare all'asse che congiunge le ruote.

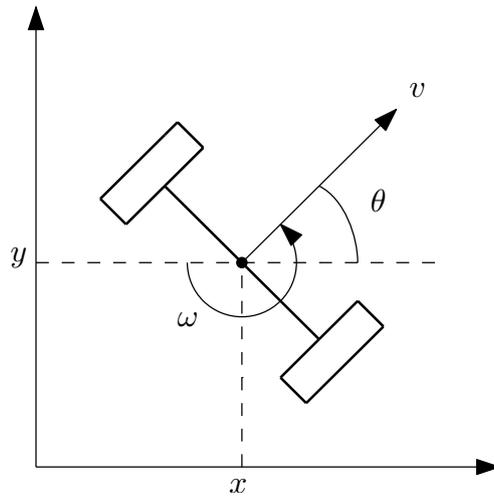


Figura 4.2: modello ad unicycle

Si noti che per costruire il modello definito in (4.1) sono state fatte delle approssimazioni, prima tra tutte sugli ingressi, che nel modello utilizzato sono le velocità v e ω ; mentre i veri ingressi ai robot sono le coppie applicate alle ruote dai motori. In questo modo viene trascurata la dinamica delle velocità e l'inerzia del robot durante il movimento.

L'algoritmo di controllo descritto nel capitolo precedente si basa sull'ipotesi che il sistema sotto controllo sia lineare. Tuttavia il modello del sistema definito (4.1) presenta delle non linearità che non possono essere trascurate, in quanto determinanti nella dinamica dell'agente. D'altra parte però un sistema non lineare risulta essere più complesso da gestire mediante un approccio di tipo MPC. Quindi, al fine di ricavare una implementazione che si presti all'uso *on-line*, è necessario che il sistema sotto controllo sia lineare. Come ampiamente descritto in [5] e [6], si ottiene un modello lineare del sistema mediante l'applicazione di un controllo in retroazione, noto in letteratura come *Feedback Linearization*.

Si considerino le accelerazioni lungo gli assi:

$$\begin{cases} a_x = \ddot{x} = \dot{v} \cos \theta - v \dot{\theta} \sin \theta = \dot{v} \cos \theta - v \omega \sin \theta \\ a_y = \ddot{y} = \dot{v} \sin \theta + v \dot{\theta} \cos \theta = \dot{v} \sin \theta + v \omega \cos \theta \end{cases}$$

riscrivendo in forma matriciale, si ha:

$$\begin{pmatrix} a_x \\ a_y \end{pmatrix} = \underbrace{\begin{bmatrix} \cos \theta & -v \sin \theta \\ \sin \theta & v \cos \theta \end{bmatrix}}_T \begin{pmatrix} \dot{v} \\ \omega \end{pmatrix}$$

Ora se $v \neq 0$ allora la matrice T è invertibile e si ottiene il seguente controllore dinamico del primo ordine

$$\begin{cases} \dot{v} = a_x \cos \theta + a_y \sin \theta \\ \omega = \frac{-a_x \sin \theta + a_y \cos \theta}{v} \end{cases}$$

Applicando tale controllore al sistema si ricava il seguente sistema lineare

$$\begin{cases} \ddot{x} = a_x \\ \ddot{y} = a_y \end{cases} \quad (4.3)$$

che rappresenta le dinamiche del sistema sotto *Feedback Linearization*, rappresentato in Figura 4.3, in cui a_x e a_y sono le nuove variabili di ingresso

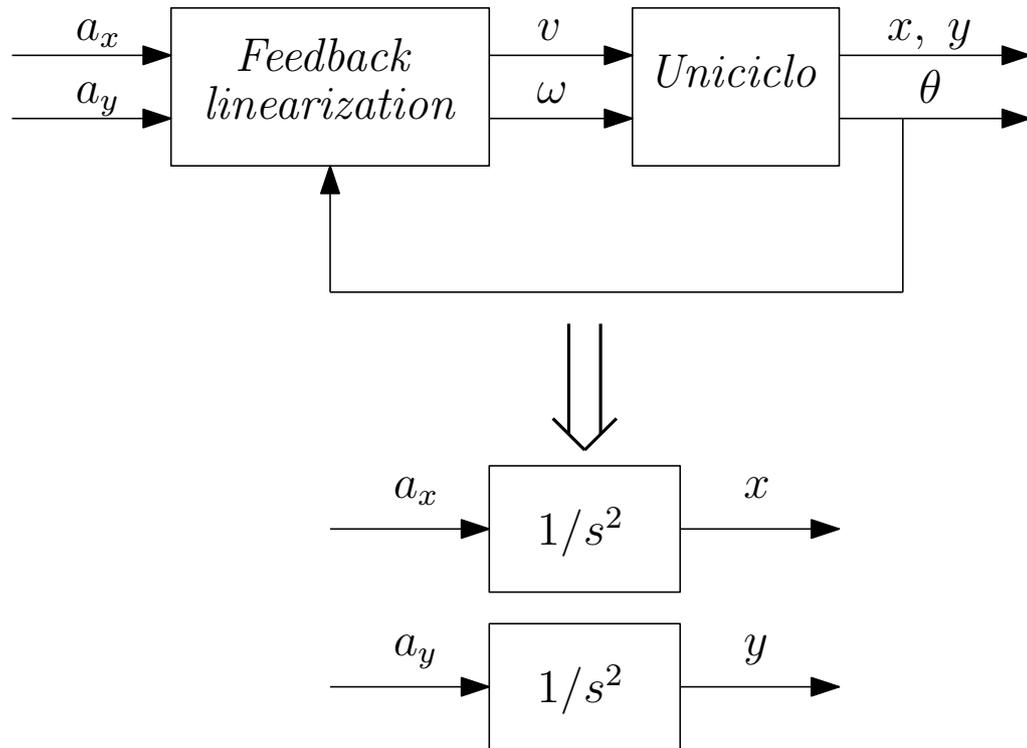


Figura 4.3: Schema di controllo con *Feedback linearization*

Si noti come, mediante la tecnica di *Feedback Linearization*, il sistema sotto controllo si riduce ad un sistema costituito da una coppia di doppi integratori, scomposti lungo gli assi, perfettamente disaccoppiati. Si sottolinea inoltre che la linearizzazione utilizzata è di tipo dinamico. D'altra parte una linearizzazione statica non sarebbe stata possibile per la presenza del vincolo di anomalia (4.2), che porterebbe ad un sistema non controllabile. L'implementazione pratica dell'anello linearizzante è fatta mediante integrazione delle equazioni del sistema con il metodo di discretizzazione di Eulero in avanti. Per maggiori dettagli si rimanda la trattazione a [6] e [5]. Tuttavia è importante considerare che un approccio di tipo *Feedback Linearization* porta con sé una notevole limitazione. In particolare, come ampiamente spiegato in [6], il fatto che la velocità dei robot è ricavata in *open loop*, è una limitazione poiché non si hanno garanzie che i valori ottenuti siano

effettivamente quelli corretti. Per risolvere tale problema, in [6] è stata proposta una soluzione che consiste nell'utilizzare un predittore di Kalman per la stima delle velocità. Inoltre, sempre in [6], si è deciso di trasferire l'anello linearizzante unito al predittore di Kalman, direttamente all'interno dell'agente. Questa scelta ha apportato notevoli miglioramenti in termini di prestazioni poiché è possibile far svolgere parallelamente i due cicli, MPC (esterno) e *Feedback Linearization* (interno), rispetto alle soluzioni impiegate in [5] in cui si svolgeva un ciclo MPC in serie alla integrazione delle equazioni e alla *Feedback Linearization*.

Il sistema a tempo discreto ottenuto implementando l'anello linearizzante, equivalente a (4.3) è il seguente:

$$\left\{ \begin{array}{l} \begin{bmatrix} \xi_1(t+1) \\ \xi_2(t+1) \\ \xi_3(t+1) \\ \xi_4(t+1) \end{bmatrix} = \begin{bmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \\ \xi_3(t) \\ \xi_4(t) \end{bmatrix} + \begin{bmatrix} \tau^2/2 & 0 \\ \tau & 0 \\ 0 & \tau^2/2 \\ 0 & \tau \end{bmatrix} \begin{bmatrix} a_x(t) \\ a_y(t) \end{bmatrix} \\ \\ \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \\ \xi_3(t) \\ \xi_4(t) \end{bmatrix} \end{array} \right.$$

che in forma più compatta si scrive

$$\begin{cases} \xi_{t+1} &= A\xi_t + Bu_t \\ z_t &= C\xi_t \end{cases} \quad (4.4)$$

4.2 Caratterizzazione del disturbo

Si consideri il sistema (4.4) ottenuto mediante *Feedback Linearization* e, per semplicità di notazione, si indichi la variabile di stato con la lettera x . A questo punto, si faccia riferimento all'algoritmo descritto nel capitolo precedente, dove si è supposto che sul sistema agisca un rumore bianco con valor medio nullo, $w_t \sim WN(0, W)$

$$\begin{cases} x_{t+1} &= Ax_t + Bu_t + Fw_t \\ z_t &= Cx_t \end{cases} \quad (4.5)$$

In questo capitolo verranno illustrate le prove e le analisi svolte, al fine di ottenere una caratterizzazione del rumore w .

L'entità del disturbo sarà valutata mediante prove sperimentali, in anello aperto. Prima di effettuare le prove è necessario comprendere quali siano gli errori di cui l'apparato sperimentale di questa tesi è affetto. In particolare i rumori che agiscono sul sistema sono:

- errori derivanti dal modello e dalla linearizzazione di quest'ultimo. La modellizzazione del sistema, infatti, non considera né l'inerzia del robot né la dinamica delle velocità. Inoltre, con la linearizzazione si assumono i robot dei semplici doppi integratori, il che inevitabilmente porta con sé errori;
- errori derivanti dalla telecamera. La misura della posizione dei robot, infatti, è affetta da imprecisioni dovute sia alla distorsione della webcam che all'algoritmo di elaborazione dell'immagine;
- Filtro di Kalman. L'utilizzo di un filtro di Kalman per la stima della velocità del robot, genera una incertezza che non può essere trascurata.

Tutti questi errori costituiranno il termine w_t , che rappresenta il disturbo che influenza l'evoluzione delle stime e delle misure disponibili della variabile di stato x_t .

Il disturbo verrà stimato facendo delle prove in anello aperto sull'impianto.

In particolare saranno svolte tre prove a velocità costante. Nella prima il robot è orientato lungo l'asse x , nella seconda lungo l'asse y e nella terza verranno applicati al robot ingressi sinusoidali. Inoltre ognuna di queste tre prove sarà ripetuta tre volte, a velocità differenti, per un totale di nove prove. Un volta effettuate le prove, il disturbo w_t può essere calcolato con la seguente equazione:

$$w_t = x_{t+1} - \underbrace{\hat{x}_{t+1|t}}_{\text{predizione}} \quad (4.6)$$

dove il termine di predizione lo si calcola come

$$\hat{x}_{t+1|t} = Ax_t + Bu_t$$

che rappresenta lo stato all'istante successivo calcolato secondo l'equazione dinamica nominale.

Gli ingressi da dare al robot (a cui è applicata la *Feedback Linearization*) sono le accelerazioni, mentre la velocità sarà determinata dalle condizioni iniziali. Nella prima prova si orienta il robot lungo l'asse x e gli si impone la velocità desiderata congiuntamente alle accelerazioni nulle. Quindi per le tre prove si ha

$$(i) \begin{cases} v = 1 \text{ cm/s} \\ u_x = 0 \\ u_y = 0 \end{cases} \quad (ii) \begin{cases} v = 5 \text{ cm/s} \\ u_x = 0 \\ u_y = 0 \end{cases} \quad (iii) \begin{cases} v = 10 \text{ cm/s} \\ u_x = 0 \\ u_y = 0 \end{cases}$$

Nella seconda si orienta il robot lungo l'asse y e si ripete la prova con le stesse velocità e accelerazioni.

Per quanto riguarda la terza prova, oltre ad imporre il valore di velocità lineare v , è necessario imporre anche le accelerazioni. Si ricordi che

$$\begin{cases} a_x = \dot{v} \cos \theta - v \omega \sin \theta \\ a_y = \dot{v} \sin \theta + v \omega \cos \theta \end{cases} \quad \text{poiché } v = \text{cost}$$

Per generare un andamento pseudo-sinusoidale è necessario che la velocità angolare del robot rispetto al suo asse sia del tipo $\omega = -a \cos(bt)$. I parametri a e b sono stati scelti come segue, rispettivamente:

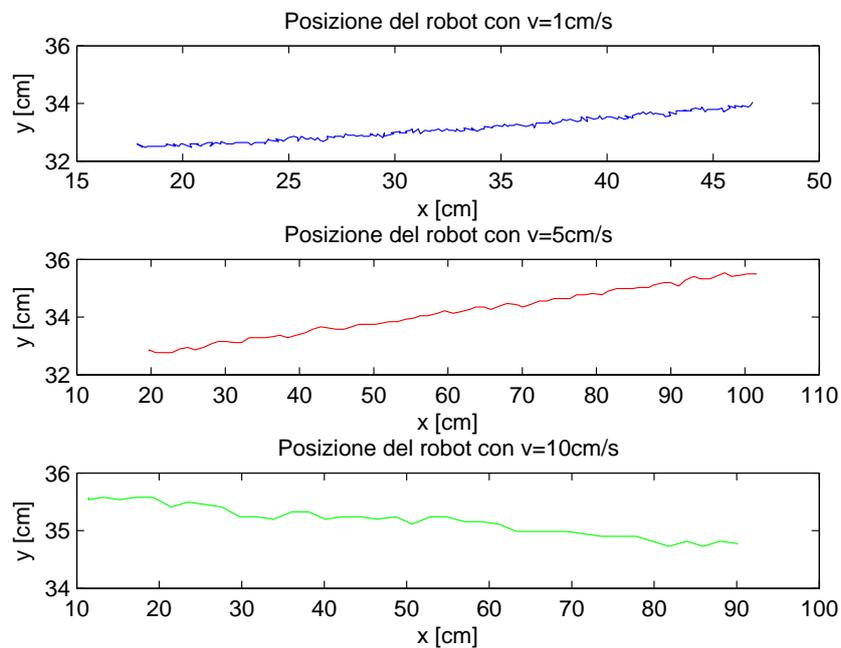
$$(i) \begin{cases} v = 1 \text{ cm/s} \\ \omega = -0.5 \cos(0.1t) \end{cases} \quad (ii) \begin{cases} v = 5 \text{ cm/s} \\ \omega = -0.5 \cos(0.1t) \end{cases}$$

$$(iii) \begin{cases} v = 10 \text{ cm/s} \\ \omega = -0.5 \cos(0.1t) \end{cases}$$

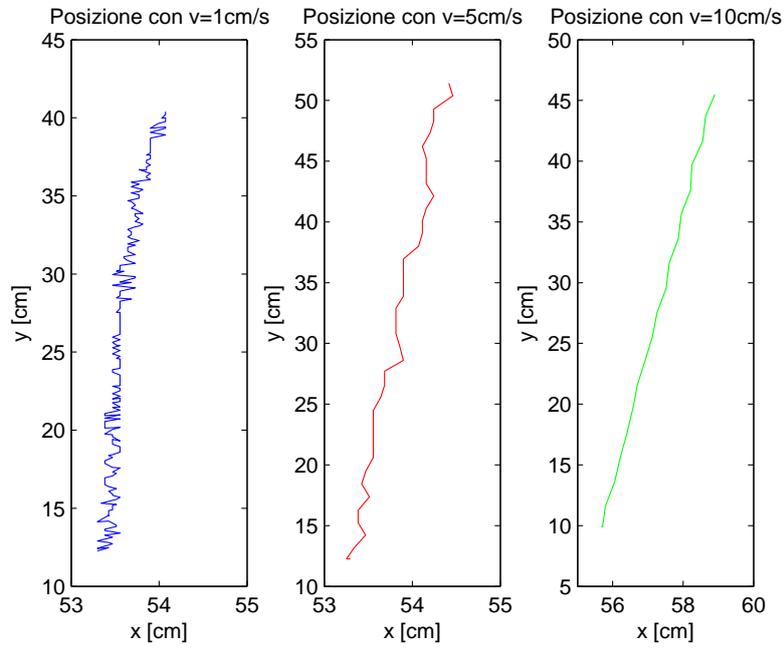
da quali è possibile ricavare le accelerazioni da inviare al robot.

Di seguito vengono riportati gli andamenti del robot nel piano di lavoro per le differenti prove.

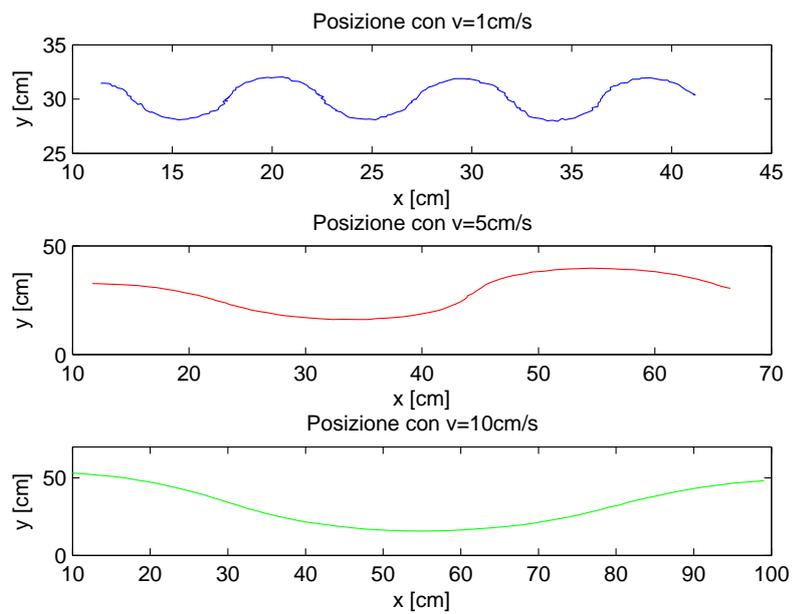
Prova lungo l'asse x



Prova lungo l'asse y



Prova seguendo una sinusoide



Per ottenere una caratterizzazione “media” rispetto alle varie prove, è stata analizzata la traiettoria \tilde{w}_t ottenuta aggregando le traiettorie $w_{t,k}$ ottenute nelle singole prove per $k = 1, \dots, 9$. Inoltre, sotto l’ipotesi che l’orientamento iniziale fosse arbitrario, sono state generate le traiettorie di w_t da analizzare come $w_t = [\tilde{w}_t, H\tilde{w}_t]$, dove

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

in modo da considerare analoghi i disturbi agenti sul disturbo in direzione x e in direzione y .

I grafici in Figura 4.4 riportano gli andamenti dei disturbi sullo stato, rispettivamente sulla posizione x , sulla posizione y e sulle velocità v_x e v_y .

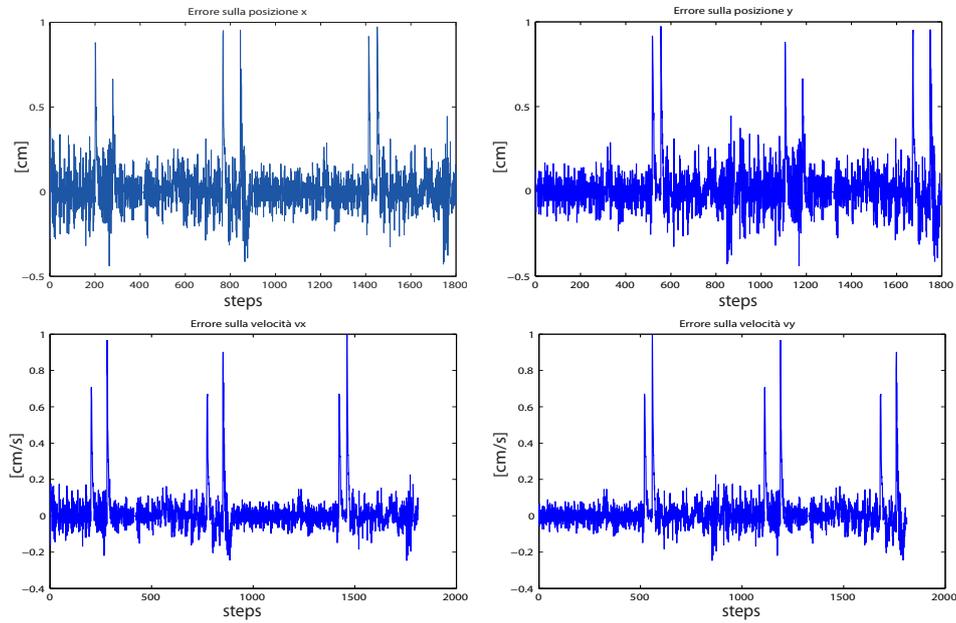


Figura 4.4: Rumore agente sullo stato

Si può notare come siano presenti dei picchi che, nonostante possano sembrare dei semplici *outliers*, sono in realtà errori relativi ai transitori di avviamento (ad esempio dovuti all'inizializzazione del filtro non corretta) delle dinamiche delle differenti prove. Come è stato detto, infatti, gli errori sono stati generati unendo tutte le prove compiute sull'impianto; tali picchi sono stati trascurati, dato che lo scopo è quello di concentrarci sulla caratterizzazione dell'errore in condizioni stazionarie. Eliminando i campioni relativi a tali picchi si ottengono i rumori in Figura 4.5. Inoltre si nota come i rumori siano effettivamente a media nulla.

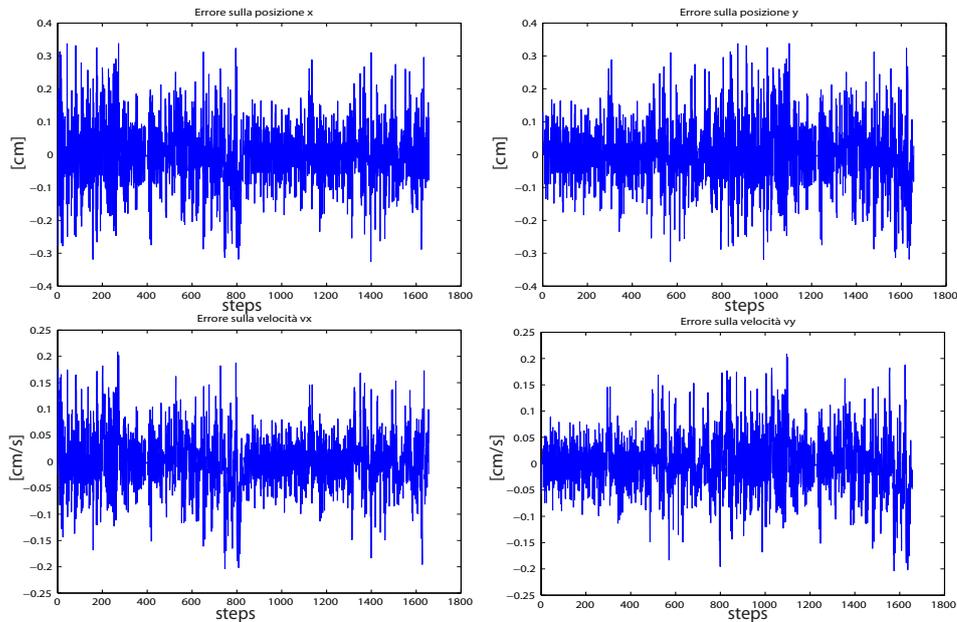


Figura 4.5: Rumore agente sullo stato trascurando i campioni che generano i picchi

La varianza campionaria è calcolata mediante la formula:

$$W = \frac{1}{n} \sum_{i=1}^n (w - \bar{w})(w - \bar{w})^T$$

dove $\bar{w} = 0$. Si ottiene:

$$W = \begin{bmatrix} 0.0110 & -0.0017 & -0.0007 & -0.0003 \\ -0.0017 & 0.0033 & 0.0003 & -0.0001 \\ -0.0007 & 0.0003 & 0.0110 & -0.0017 \\ 0.0003 & -0.0001 & -0.0017 & 0.0033 \end{bmatrix}$$

Si noti che W si può approssimare ad una matrice diagonale a blocchi. I valori assunti dagli elementi nei blocchi antidiagonali sono infatti molto più piccoli rispetto a quelli sulla diagonale principale. I valori sulla diagonale sono quindi le varianze che caratterizzano il rumore. Quindi la varianza del rumore sulle posizioni è $W_x = W_y = 0.0011$, mentre la varianza del rumore sulla velocità è $W_{vx} = W_{vy} = 0.0033$. A questo punto si riporta, in Figura 4.6, la funzione di densità di probabilità empirica dei rumori (in blu) e si confronta la loro distribuzione con quella di una gaussiana perfetta (in rosso) avente media nulla e varianza pari agli elementi sulla diagonale di W , la cui funzione è

$$y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}} \quad \text{se } x \sim \mathcal{G}(\bar{x}, \sigma^2)$$

Dalla Figura 4.6 si può concludere che i rumori possono essere caratterizzati come variabili aleatorie gaussiane a media nulla.

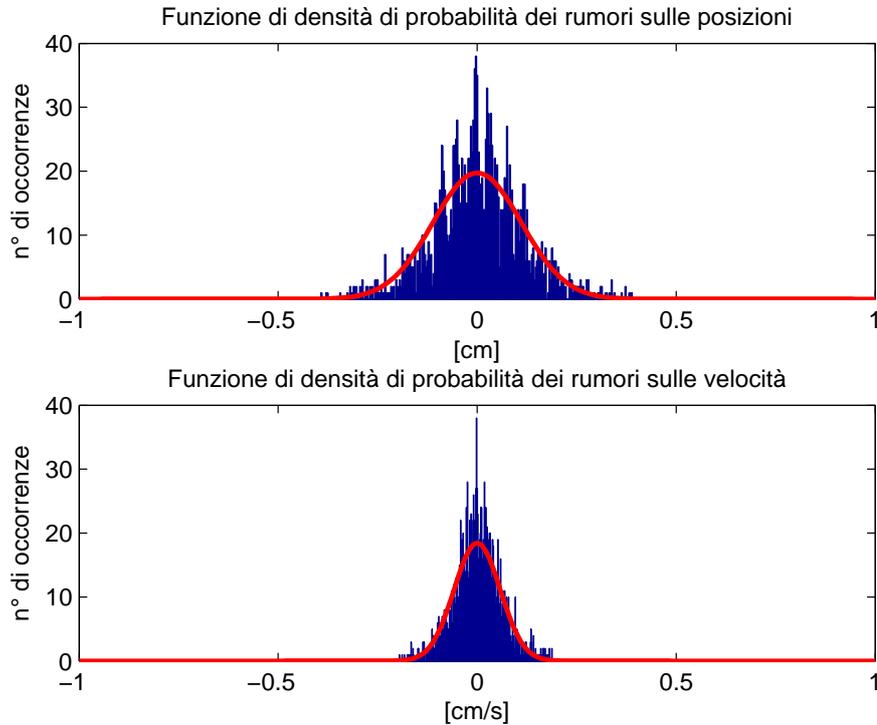


Figura 4.6: Funzioni di densità di probabilità

Si noti che nulla è stato finora detto sulla ipotesi considerata relativa alla bianchezza dei rumori. Allo scopo di analizzare le proprietà di bianchezza del rumore w_t è possibile ricorrere al classico Test di Anderson. Scegliendo un intervallo di confidenza pari a $\alpha = 0.05$ si ottengono i risultati mostrati in Figura 4.7.

In particolare, la percentuale di campioni fuori dalla fascia di confidenza al 5% per la componente di posizione è il 13%; mentre per la componente di velocità è il 12%. Nonostante ciò dimostri che l'ipotesi di bianchezza non è perfettamente accettabile, dalla Figura 4.7 è chiaramente visibile che i campioni (ossia i valori della funzione di covarianza campionaria dei dati) si concentrino in modo netto nella fascia di confidenza determinata. Que-

sto permette di concludere che l'ipotesi di rumore bianco possa essere, in prima approssimazione, accettabile in pratica. Ulteriori motivi di praticità portano inoltre ad accettare tale conclusione, in virtù del fatto che non è stato possibile ottenere un modello (dinamico) più raffinato del disturbo w_t . Ulteriori sviluppi del lavoro potranno essere dedicati a questo aspetto.

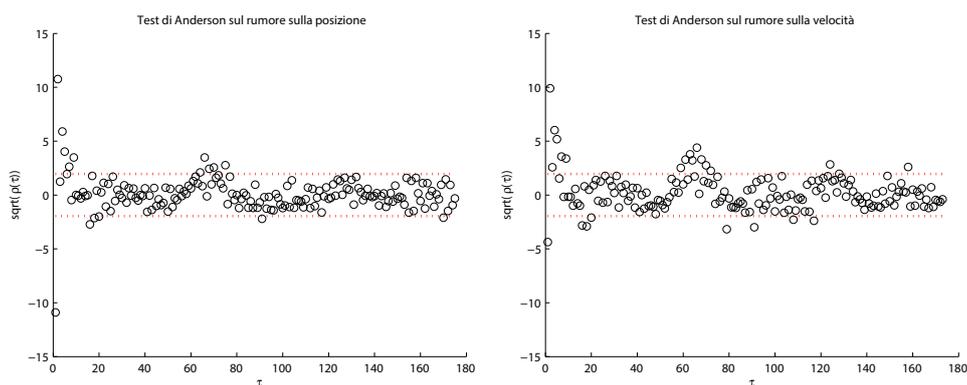


Figura 4.7: Test di Anderson

Capitolo 5

Caso di studio: Formulazione ed implementazione di un problema di navigazione autonoma

In questo capitolo verrà applicato l'algoritmo descritto nel capitolo 3 al caso di studio, descritto nel capitolo 4. Nello specifico, lo scopo applicativo di questa tesi consiste nel risolvere un problema di navigazione da parte di una flotta di robot autonomi verso un obiettivo in assenza di collisione reciproca o con ostacoli fissi. Ci si propone di guidare un insieme di agenti all'interno di uno spazio di lavoro opportuno, verso punti noti a priori chiamati *goal*. Si consideri dapprima il caso in cui sia presente un solo agente nello spazio di lavoro il quale, a partire da una posizione iniziale deve raggiungere un *goal* noto. All'interno dello spazio di lavoro potrebbero essere presenti ostacoli, anch'essi noti a priori, che rappresentano impedimenti nel raggiungimento del *goal* da parte dell'agente. In tal caso il problema da risolvere è noto come navigazione autonoma con *obstacle avoidance*. L'utilizzo di una strategia di controllo basata su MPC permette di risolvere il problema, nonostante l'obiettivo sia duplice: raggiungere il goal (*path following*) e evitare gli ostacoli

(*obstacle avoidance*). Tali problemi vengono risolti contemporaneamente, e a seconda della posizione dell'agente rispetto a ostacoli e *goal*, si dà una priorità maggiore ad un problema rispetto all'altro, grazie alla formulazione *ad hoc* del problema di ottimizzazione.

Nel caso specifico, in cui si utilizza MPC, il problema della navigazione autonoma di un agente con proprietà di *obstacle avoidance* può essere risolto inserendo un vincolo aggiuntivo nel problema di ottimizzazione. Questo, infine, può essere facilmente esteso ad un problema di navigazione autonoma con *collision avoidance*. Quest'ultimo richiede, nel caso in cui si considerano più agenti nello spazio di lavoro, che, senza collidere, questi raggiungano i rispettivi *goal*. Si affrontano nel dettaglio entrambi i problemi.

5.1 Obstacle Avoidance

Il problema dell'*obstacle avoidance*, come detto, può essere direttamente incluso nel problema di ottimizzazione mediante un vincolo opportuno. Ci si propone, quindi, di specificare i vincoli che ogni agente deve rispettare al fine di evitare gli ostacoli durante il percorso. In particolare si dice che un agente ha proprietà di *obstacle avoidance* se, per ogni ostacolo definito in posizione fissa, è possibile generare un percorso privo di collisioni. In tal modo l'agente, durante il moto, riesce a mantenersi ad una distanza opportuna dall'ostacolo.

Per semplicità di rappresentazione, gli ostacoli (indicati con l'indice h) verranno assunti circolari e centrati in posizioni note a priori. Per ogni ostacolo è quindi noto il raggio R_h^o ed il centro z_h^o . A questo punto, per ogni robot i , $i = 1, \dots, M$ si definisce un *set* di n_i^o ostacoli come $\mathcal{O} \subseteq \{1, \dots, n_i^o\}$, che generalmente rappresenta l'insieme di ostacoli che si trovano all'interno dello spazio di lavoro e che l'agente potrebbe incontrare durante il proprio moto. È importante notare che la presenza di ostacoli circolari, all'interno dello spazio operativo, genera una regione, priva di collisioni, non convessa, facendo sì che l'intero problema di ottimizzazione risulti non convesso. Si noti

che la soluzione di problemi di ottimizzazione di questo tipo non è semplice da ottenere, non essendo univocamente definito il minimo globale o essendo esso calcolabile in modo non convenzionale. Si comprende facilmente come nella pratica sia molto più semplice risolvere problemi di ottimizzazione convessi, essendo meno complessi sia da un punto di vista teorico che computazionale; inoltre la scelta comunemente usata ricade su funzionali di costo quadratici e con vincoli lineari, il che permette di semplificare ulteriormente il problema.

In questa tesi viene utilizzata una metodologia, che permette di ridefinire il problema in modo tale che, ad ogni passo, l'ottimizzazione risulti convessa. In particolare verrà inizialmente definito il vincolo di *obstacle avoidance* in termini di minima distanza che deve sussistere, al fine di evitare collisioni, tra robot e ostacolo. Questo vincolo risulterà essere non convesso. Dopodiché sarà riformulato sfruttando una rappresentazione attraverso vincoli lineari.

L'idea di base per generare vincoli convessi consiste nell'approssimare gli ostacoli, assunti essere circolari, a politopi regolari che circoscrivono l'ostacolo, definiti da vincoli lineari. Nella pratica, gli ostacoli saranno approssimati da politopi definiti su \mathbb{R}^2 e quindi poligoni. A questo punto, mediante una scelta che verrà spiegata in seguito, verrà selezionato uno solo tra i lati del poligono, che rappresenterà il vincolo da rispettare nel problema di ottimizzazione. In tal modo, ad ogni istante, la regione di spazio libera da collisioni è suddivisa in sottoregioni ognuna delle quali risulta essere convessa.

Si consideri il vincolo di minima distanza:

$$\|z - \bar{z}\|_2 \geq \bar{d} \quad (5.1)$$

dove \bar{z} e \bar{d} sono noti. Si consideri ora un politopo \mathbf{P} che circoscrive un cerchio generico, definito da un numero r di disuguaglianze lineari:

$$\mathbf{P} := \left\{ z : h_k^T(z - \bar{z}) \leq \bar{d}, \quad k = 1, \dots, r \right\} \quad (5.2)$$

Si osservi che la violazione di almeno una delle disuguaglianze:

$$h_k^T(z - \bar{z}) \leq \bar{d} \quad (5.3)$$

implica che (5.1) sia verificata.

Formulazione del vincolo di *Obstacle Avoidance*

Si consideri un robot i con raggio R_i e centro $z_t^{[i]}$. Siano, inoltre z_h^o e R_h^o , rispettivamente il centro e il raggio dell'ostacolo.

Si consideri il seguente il vincolo di minima distanza

$$\|z_t^{[i]} - z_h^o\|_2 \geq d_{hi}^o \quad (5.4)$$

dove $d_{hi}^o = R_i + R_h^o$. Come detto, il vincolo (5.4) non è convesso e quindi non è possibile includerlo in tale modo. Si definisce, allora il politopo \mathbf{P}_{hi}^o , che approssima il cerchio centrato in z_h^o e di raggio d_{hi}^o , mediante un set di vincoli lineari r_{hi}^o :

$$\mathbf{P}_{hi}^o := \left\{ z_t^{[i]} : (h_l^{[o,hi]})^T (z_t^{[i]} - z_h^o) \leq d_{hi}^o, \quad l = 1, \dots, r_{hi}^o \right\} \quad (5.5)$$

In particolare ogni vincolo del tipo

$$(h_l^{[o,hi]})^T (z_t^{[i]} - z_h^o) \leq d_{hi}^o$$

definisce un semipiano a cui appartiene il centro dell'ostacolo. Al fine di evitare collisione, ad ogni istante di tempo, l'agente deve trovarsi fuori da tale regione. Questa garanzia la si ottiene se almeno una delle disuguaglianze definite in (5.5) è violata.

La seguente condizione deve quindi sussistere:

$$\exists \bar{l} : \quad - \left(h_{\bar{l}}^{[o,hi]} \right)^T (z_t^{[i]} - z_h^o) \leq -d_{hi}^o \quad (5.6)$$

dove \bar{l} rappresenta il vincolo (lato del poligono) selezionato. Ci si propone ora di specificare come avviene la selezione di tale vincolo. Tale selezione viene compiuta attraverso la valutazione della traiettoria predetta, del robot

i , risolvendo il problema di ottimizzazione all'istante precedente a quello corrente. Tale traiettoria verrà indicata con $z_{k|t-1}^{[i]}$.

Si consideri da esempio la posizione $z_{k|t-1}^{[i]}$ per l'istante futuro k . L'idea sfruttata in questa tesi è quella di selezionare il vincolo che maggiormente viene violato da $z_{k|t-1}^{[i]}$. Si assume infatti che la traiettoria predetta al tempo precedente sia priva di collisioni. Esiste dunque un \bar{l} tale che

$$-(h_{\bar{l}[o,hi]})^T (z_{k|t-1}^{[i]} - z_h^o) \leq -d_{hi}^o$$

Dato che i vincoli "violati" da $z_{k|t-1}^{[i]}$ possono essere in numero superiore a uno, è necessario individuare il vincolo violato rispetto a quale il robot si trova a distanza maggiore. Formalmente viene selezionato \bar{l} in modo tale che

$$-(h_{\bar{l}}^{[o,hi]})^T (z_{k|t-1}^{[i]} - z_h^o) + d_{hi}^o < -(h_l^{[o,hi]})^T (z_{k|t-1}^{[i]} - z_h^o) + d_{hi}^o \quad \forall l \neq \bar{l}$$

Si consideri, a titolo di esempio, la Figura 5.1, in cui si descrive la scelta del vincolo da rispettare.

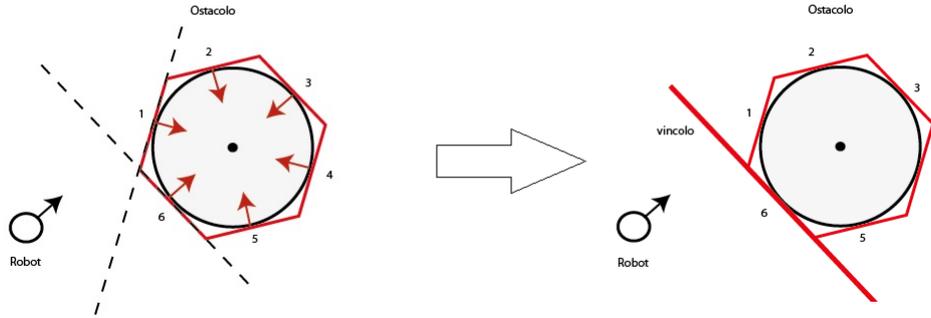


Figura 5.1: Selezione del vincolo maggiormente violato

L'ostacolo è approssimato con un pentagono, il che nonostante sia una scarsa approssimazione di un cerchio, risulta essere utile per comprendere la logica utilizzata. Nelle prove che verranno eseguite, sia in simulazione

che sull'apparato sperimentale, infatti, gli ostacoli saranno approssimati a poligoni con un numero di lati più elevato. Le frecce rosse indicano le regioni di piano definite dai vincoli in (5.5). È facile notare che il robot sta violando i vincoli indicati con 1 e 6. In questo caso è possibile notare che la distanza dalla retta che definisce il vincolo 6 è maggiore della distanza dalla retta che definisce il vincolo 1 e pertanto verrà selezionato il vincolo $\bar{l} = 6$. Il vincolo che verrà dunque imposto alla variabile $z_k^{[i]}$ nel problema di ottimizzazione (formulato all'istante t) corrisponde a

$$-\left(h_{\bar{l}}^{[o,hi]}\right)^T \left(z_k^{[i]} - z_h^o\right) \leq -d_{hi}^o \quad (5.7)$$

Prima di essere incluso nel problema di ottimizzazione, è necessario riformulare (5.7) come vincolo probabilistico, poiché, come già evidenziato, le variabili in gioco sono stocastiche e quindi non sono note le loro evoluzioni temporali in modo deterministico. Si introduce, quindi, una probabilità di violazione del vincolo (5.7)

$$\mathcal{P} \left\{ -\left(h_{\bar{l}}^{[o,hi]}\right)^T \left(z_k^{[i]} - z_h^o\right) \geq -d_{hi}^o \right\} \leq p_{hi}^o \quad (5.8)$$

In modo del tutto analogo a quanto fatto per i vincoli (2.29) e (2.30), mediante l'utilizzo della disuguaglianza di Cantelli-Chebyshev, il vincolo probabilistico di *obstacle avoidance*, definito in (5.8) è riformulato rispetto al valore atteso $\bar{z}_k^{[i]}$ e alla varianza $Z_k^{[i]}$ di $z_k^{[i]}$ di cui è nota l'evoluzione:

$$\left(h_{\bar{l}}^{[o,hi]}\right)^T \left(\bar{z}_k^{[i]} - z_h^o\right) \geq d_{hi}^o + \sqrt{\left(h_{\bar{l}}^{[o,hi]}\right)^T Z_k^{[i]} h_{\bar{l}}^{[o,hi]}} f(p_{hi}^o) \quad (5.9)$$

dove $\bar{z}_k^{[i]} = C\bar{x}_k^{[i]}$ e $Z_k^{[i]} = CX_k^{[i]}C^T$. Si ricorda, inoltre, che la funzione $f(p_{hi}^o)$ può essere calcolata con la formula derivata dalla disuguaglianza di Cantelli oppure con l'utilizzo della funzione di distribuzione cumulativa se il disturbo è gaussiano. A questo punto si noti che il vincolo definito in (5.9), garantisce che venga rispettato il vincolo in probabilità definito in (5.8).

Come visto, il vincolo deve essere garantito lungo tutto l'orizzonte di previsione. Operativamente la procedura da seguire è la seguente: si definisce,

per $k = t, \dots, t + N - 1$

$$\delta_{hi,l}^o(k|t-1) = \left(h_l^{[o,hi]} \right)^T \left(\bar{z}_{k|t-1}^{[i]} - z_h^o \right) - d_{hi}^o - \sqrt{\left(h_l^{[o,hi]} \right)^T Z_{k|t-1}^{[i]} h_l^{[o,hi]} f(p_{hi}^o)} \quad (5.10)$$

che rappresenta in termini quantitativi distanza fra $\bar{z}_{k|t-1}^{[i]}$ e la retta che definisce il vincolo l -esimo. Si noti che la funzione (5.10) è nota al tempo t , essendo funzione delle predizioni calcolate al passo precedente $t - 1$. Si noti, inoltre, che per le ipotesi adottate $\delta_{hi,l}^o$ assume un valore positivo, in corrispondenza di almeno un valore di l . Ora, per ogni valore di $\delta_{hi,l}^o$ si calcola il corrispondente vincolo l che viene maggiormente violato

$$\bar{l}_{hi}^o(k|t-1) = \operatorname{argmax}_{l \in \{1, \dots, r_{hi}^o\}} \delta_{hi,l}^o(k|t-1) \quad (5.11)$$

A questo punto può essere risolta l'ottimizzazione al passo t includendo il seguente vincolo:

$$\left(h_{\bar{l}_{hi}^o(k|t-1)}^{[o,hi]} \right)^T \left(\bar{z}_k^{[i]} - \bar{z}_{k|t-1}^{[i]} \right) \geq -\delta_{hi, \bar{l}_{hi}^o(k|t-1)}^o(k|t-1) \quad (5.12)$$

5.2 Collision Avoidance

Si suppone ora che sia presente più di un agente nell'area di lavoro. In questo caso, ogni agente, per poter inseguire correttamente la propria traiettoria di riferimento e, in modo congiunto, generare una traiettoria priva di collisioni, deve tener conto non soltanto della presenza degli ostacoli fissi, ma anche degli altri agenti mobili. Come già discusso la soluzione è ricavata mediante un algoritmo MPC di tipo distribuito. In particolare, grazie a tale approccio, ogni agente trasmette agli altri la previsione della propria traiettoria futura, in modo tale da permettergli di generare traiettorie che siano senza collisioni e che inseguano i riferimenti il meglio possibile. Ogni robot

verrà considerato dagli altri esattamente come un ostacolo che, durante il suo percorso, modifica la sua posizione. Per evitare collisione si applicherà la stessa logica utilizzata nella *obstacle avoidance*. Si ricorda che i robot sono di forma circolare, ed è possibile quindi definire un politopo che li circonda.

Si considerino due robot i e j con raggi, rispettivamente, R_i e R_j . Il vincolo è espresso mediante la minima distanza che tra loro deve sussistere per evitare collisione

$$\|z_t^{[i]} - z_t^{[j]}\|_2 \geq d_{ij} \quad (5.13)$$

dove $d_{ij} = R_i + R_j$. Anche in questo caso il vincolo non è convesso. Si definisce quindi un politopo \mathbf{P} che circonda un cerchio centrato in $z_t^{[j]}$ di raggio d_{ij} , definito da un set di r_{ij} disuguaglianze lineari

$$\mathbf{P} := \left\{ z_t^{[i]} : \left(h_l^{[i,j]} \right)^T \left(z_t^{[i]} - z_t^{[j]} \right) \leq d_{ij}, \quad l = 1, \dots, r_{ij} \right\} \quad (5.14)$$

Si consideri ancora il caso in cui siano note le traiettorie $\bar{z}_{k|t-1}^{[i]}$ e $\bar{z}_{k|t-1}^{[j]}$ per $k = 1, \dots, t + N - 1$. Affinché non vi siano collisioni tra i robot:

$$\exists \bar{l} : - \left(h_{\bar{l}}^{[i,j]} \right)^T \left(z_{k|t-1}^{[i]} - z_{k|t-1}^{[j]} \right) \geq -d_{ij} \quad \forall k = t, \dots, t + N - 1$$

A questo punto si seleziona il lato del politopo che circonda il robot i che è a maggiore distanza dal centro del robot j (e viceversa). In particolare il robot i -esimo individua il lato del politopo j -esimo che si trova a distanza maggiore, il quale risulta essere simmetrico a quello che il robot j -esimo individua per il politopo i -esimo. A titolo di esempio, si faccia riferimento alla Figura 5.2 in cui il robot i seleziona il vincolo 5 del robot j , mentre il robot j seleziona il vincolo 2 del robot i .

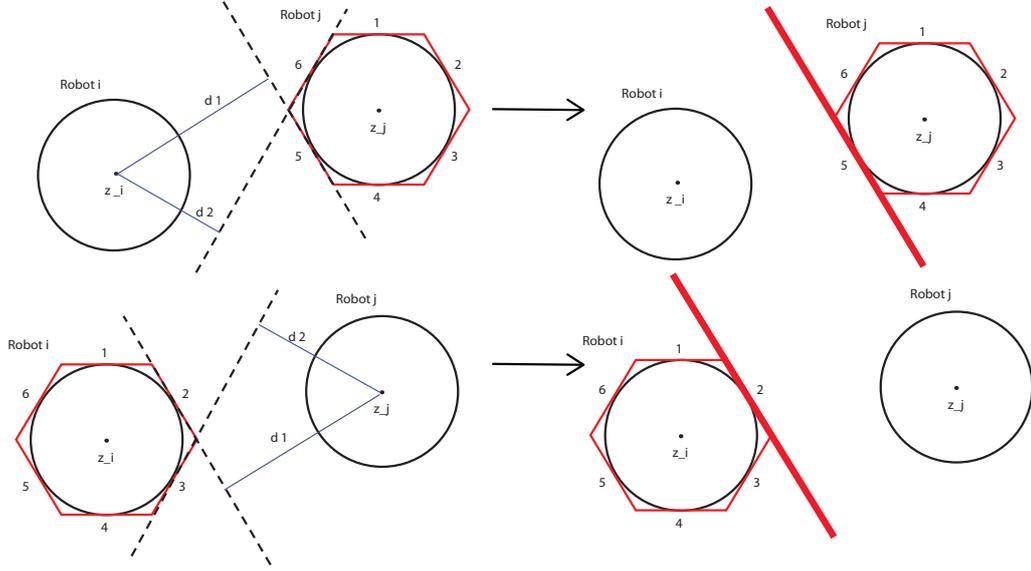


Figura 5.2: Selezione del vincolo maggiormente violato

Individuato il lato del politopo rispetto al quale il centro del robot si trova a distanza maggiore, si deve definire il vincolo di *collision avoidance* da imporre effettivamente. Anche in questo caso, è necessario introdurre una probabilità di violazione del vincolo:

$$\mathcal{P} \left\{ - \left(h_i^{[ij]} \right)^T \left(z_k^{[i]} - z_k^{[j]} \right) \geq -d_{ij} \right\} \leq p_{ij}$$

Riformulando il vincolo in termini deterministici mediante la disuguaglianza di Cantelli si ricava

$$\left(h_i^{[ij]} \right)^T \left(\bar{z}_k^{[i]} - \bar{z}_k^{[j]} \right) \geq d_{ij} + \sqrt{\left(h_i^{[hi]} \right)^T \left(Z_k^{[i]} + Z_k^{[j]} \right) h_i^{[ij]} f(p_{ij})} \quad (5.15)$$

Si definisca ora il vincolo che ogni robot deve includere per evitare collisioni con altri robot.

Similmente al caso della strategia sviluppata per evitare collisioni con ostacoli fissi, al passo t , sono note, *off line*, le predizioni lungo l'orizzonte dei

valori attesi e delle varianze calcolate al tempo $t - 1$. A partire da queste si calcola la seguente funzione per $k = t, \dots, t + N - 1$ e per ogni $l = 1, \dots, r_{ij}$

$$\begin{aligned} \delta_{ij,l}(k|t-1) &= \left(h_l^{[hi]} \right)^T \left(\bar{z}_{k|t-1}^{[i]} - \bar{z}_{k|t-1}^{[j]} \right) - d_{ij} + \\ &\quad - \sqrt{\left(h_l^{[ij]} \right)^T \left(Z_{k|t-1}^{[i]} + Z_{k|t-1}^{[j]} \right) h_l^{[ij]} f(p_{ij})} \end{aligned}$$

e l'indice relativo al vincolo maggiormente violato:

$$\bar{l}_{ij}(k|t-1) = \underset{l \in \{1, \dots, r_{ij}\}}{\operatorname{argmax}} \delta_{ij,l}(k|t-1) \quad (5.16)$$

A questo punto all'istante successivo t i vincoli che devono essere rispettati sono, coerentemente con quanto visto nel capitolo 4

$$\left(h_{\bar{l}_{ij}(k|t-1)} \right)^T \left(\bar{z}_k^{[i]} - \bar{z}_{k|t-1}^{[i]} \right) \geq \frac{1}{2} \delta_{ij, \bar{l}_{ij}(k|t-1)}(k|t-1) \quad (5.17)$$

$$\left(h_{\bar{l}_{ij}(k|t-1)} \right)^T \left(\bar{z}_{k|t-1}^{[j]} - \bar{z}_k^{[j]} \right) \geq \frac{1}{2} \delta_{ij, \bar{l}_{ij}(k|t-1)}(k|t-1) \quad (5.18)$$

$$Z_{k|t}^{[i]} \leq Z_{k|t-1}^{[i]} \quad (5.19)$$

$$Z_{k|t}^{[j]} \leq Z_{k|t-1}^{[j]} \quad (5.20)$$

Facendo riferimento alle disuguaglianze (5.17) e (5.18) si nota un disaccoppiamento, nel senso che il primo vincolo è relativo al robot i -esimo ed il secondo è relativo al j -esimo, con $i \neq j$. Inoltre si può anche notare una perfetta simmetria. Tale simmetria deriva dalla scelta di utilizzare un algoritmo distribuito, ed è proprio l'utilizzo di un tale algoritmo che permette di partizionare lo spazio in regioni simmetriche. Una volta calcolata la distanza tra due robot, l'idea di base descritta in (5.17) e (5.18) vincola entrambi i robot a muoversi al più della metà di tale distanza. In questo caso è garantita la *feasibility* ricorsiva dello schema di controllo.

Si faccia ora riferimento ai vincoli definiti in (5.19) e (5.20). Si noti che, poiché la matrice di covarianza Z , ad un generico istante, è unicamente definita dalla scelta del guadagno stabilizzante \bar{K} , per la proprietà di monotonia della equazione di Riccati, tali vincoli risultano essere sempre verificati.

5.3 Stima adattativa della varianza del disturbo

Nel capitolo 4 si è data una caratterizzazione del disturbo di tipo *off-line*. Può capitare, tuttavia, che l'agente, durante il suo moto, non sia sempre soggetto allo stesso disturbo, nel senso che la varianza di quest'ultimo può non essere sempre costante ma, ad esempio può variare a seconda della velocità media dell'agente. Per risolvere tale problema si è pensato di implementare una soluzione di stima della varianza *on-line* in modo da rendere l'algoritmo adattativo. In particolare, sfruttando un approccio di stima ricorsiva con *forgetting factor*, ad ogni istante di campionamento si definisce una matrice di covarianza \hat{W}_t come segue:

$$\hat{W}_{t+1} = \rho \hat{W}_t + (1 - \rho) w_t w_t^T \quad (5.21)$$

dove $\rho \in (0, 1)$ è il *forgetting factor* e w_t è il rumore sullo stato.

In particolare più ρ è grande meno velocemente varia la stima rispetto a quella definita a priori. Nello schema di controllo proposto, detta W la stima della varianza di w_t effettivamente usata per il calcolo delle covarianze, essa non viene aggiornata al valore \hat{W}_t ad ogni passo, per evitare che i risultati ottenuti siano soggetti a fluttuazioni derivanti dai dati piuttosto che dalle loro proprietà statistiche. Si è scelto, inoltre di aggiornare la stima di W solamente quando il valore di W utilizzato per calcolare le restrizioni dei vincoli risulta essere sufficientemente differente da quello stimato. In tal caso si definisce un valore di soglia (*threshold*), scelto in fase di progetto, e si effettua l'aggiornamento se risulta

$$\|W - \hat{W}_t\|_2 > \text{threshold} \quad (5.22)$$

Cioè se si verifica (5.22), allora si pone $W = \bar{W}_t$ e si ricalcolano le restrizioni dei vincoli rispetto alla nuova varianza W .

Capitolo 6

Simulazioni

In questo capitolo saranno mostrati i risultati dei test svolti in simulazione dell'algoritmo descritto nel capitolo 5. In particolare, dopo aver brevemente descritto le scelte implementative adottate e i parametri di progetto, saranno illustrate differenti prove.

Si faccia riferimento al sistema definito in (4.5), dove le matrici sono

$$A = \begin{bmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \tau^2/2 & 0 \\ \tau & 0 \\ 0 & \tau^2/2 \\ 0 & \tau \end{bmatrix},$$
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

dove si è scelto come tempo di campionamento $\tau = 0.3$ s.

Il termine w_t , come visto nel capitolo 2, può essere approssimato con un rumore bianco gaussiano a media nulla e matrice di covarianza W (cioè $w \sim \text{WGN}(0, W)$),

dove si è calcolato

$$W = \begin{bmatrix} 0.011 & -0.0017 & 0 & 0 \\ -0.0017 & 0.0033 & 0 & 0 \\ 0 & 0 & 0.011 & -0.0017 \\ 0 & 0 & -0.0017 & 0.0033 \end{bmatrix}$$

L'area di lavoro è un rettangolo di vertici $(0, 0)$, $(115, 0)$, $(115, 66)$, $(0, 66)$; mentre la variabile di ingresso è soggetta al vincolo $\|u_k\|_\infty \leq 1$, $\forall k = t, \dots, t + N - 1$, dove le misure per il banco di lavoro sono in centimetri, mentre per gli ingressi in centimetri al secondo quadrato.

Le matrici di peso definite nella funzione di costo (2.16) sono scelte nel modo seguente: $Q = I_4$ e $R = I_2$, (dove I_n è la matrice identità di dimensione n , da cui risulta

$$P = \begin{bmatrix} 9.1890 & 5.0249 & 0 & 0 \\ 5.0249 & 9.2324 & 0 & 0 \\ 0 & 0 & 9.1890 & 5.0249 \\ 0 & 0 & 5.0249 & 9.2324 \end{bmatrix}$$

P , come detto, è ottenuta come soluzione dell'equazione di Lyapunov. Il guadagno \bar{K} , ottenuto tramite legge LQ è:

$$\bar{K} = \begin{bmatrix} 0.8412 & 1.5460 & 0 & 0 \\ 0 & 0 & 0.8412 & 1.5460 \end{bmatrix}$$

In questa sezione verranno presentati i risultati ottenuti in simulazione utilizzando il sistema definito in (4.4). Inizialmente verrà considerato un solo sistema con l'obiettivo di verificare i risultati teorici definiti nel capitolo 2; dopodiché si passerà al caso distribuito con più sistemi interagenti.

6.1 Raggiungimento di un obiettivo e inseguimento di una traiettoria di riferimento

Si consideri il problema di far raggiungere ad un agente un obiettivo in un ambiente privo di ostacoli. Verranno mostrate differenti prove, nelle quali all'agente è richiesto di portarsi in un punto specifico dello spazio di lavoro, oppure seguire le traiettorie di riferimento. In particolare la prima traiettoria di riferimento è rettilinea e di equazione $x - y - 20 = 0$; la seconda è una traiettoria a forma di otto con legge oraria: $x = 40 + 20 \cos(t)$ e $y = 30 + 20 \sin(t)$; la terza è una traiettoria circolare di legge oraria: $x = 70 + 20 \sin(20.1t)$ e $y = 30 + 20 \sin(0.1t)$; la quarta è rettangolare (vertici: $(30, 10)$, $(90, 10)$, $(90, 50)$, $(30, 50)$); l'ultima è sinusoidale di equazione: $y(t) = 35 + 15 \sin(0.3t)$.

In Figura 6.1 vengono mostrati i risultati ottenuti nelle prove. Si può notare come le traiettorie seguano fedelmente i riferimenti, nonostante gli andamenti delle suddette siano leggermente sporcati, il che è dovuto al rumore agente sul sistema. Si faccia riferimento alla Figura 6.1 e nello specifico alla figura in alto a sinistra. Apparentemente l'agente non segue la traiettoria ottimale per giungere al goal desiderato. Questo appare evidente dal fatto che, invece di percorrere la retta che congiunge la posizione di partenza e il *goal*, il robot si allinea con l'asse delle ascisse prima di percorrere il tratto finale orizzontale. Tale comportamento è il risultato dell'effetto congiunto dei vincoli sull'accelerazione imposti e la strategia adottata per definire il *terminal cost* (*zero terminal set*, si veda il capitolo 2).

Le accelerazioni nelle due direzioni, infatti, sono contemporaneamente vincolate ad assumere valore tra $[-1, 1]$. Tale vincolo, di carattere strutturale, si traduce in un vincolo di massima velocità raggiunta dal veicolo. Il robot si muove, dunque, lungo una retta con pendenza -1 .

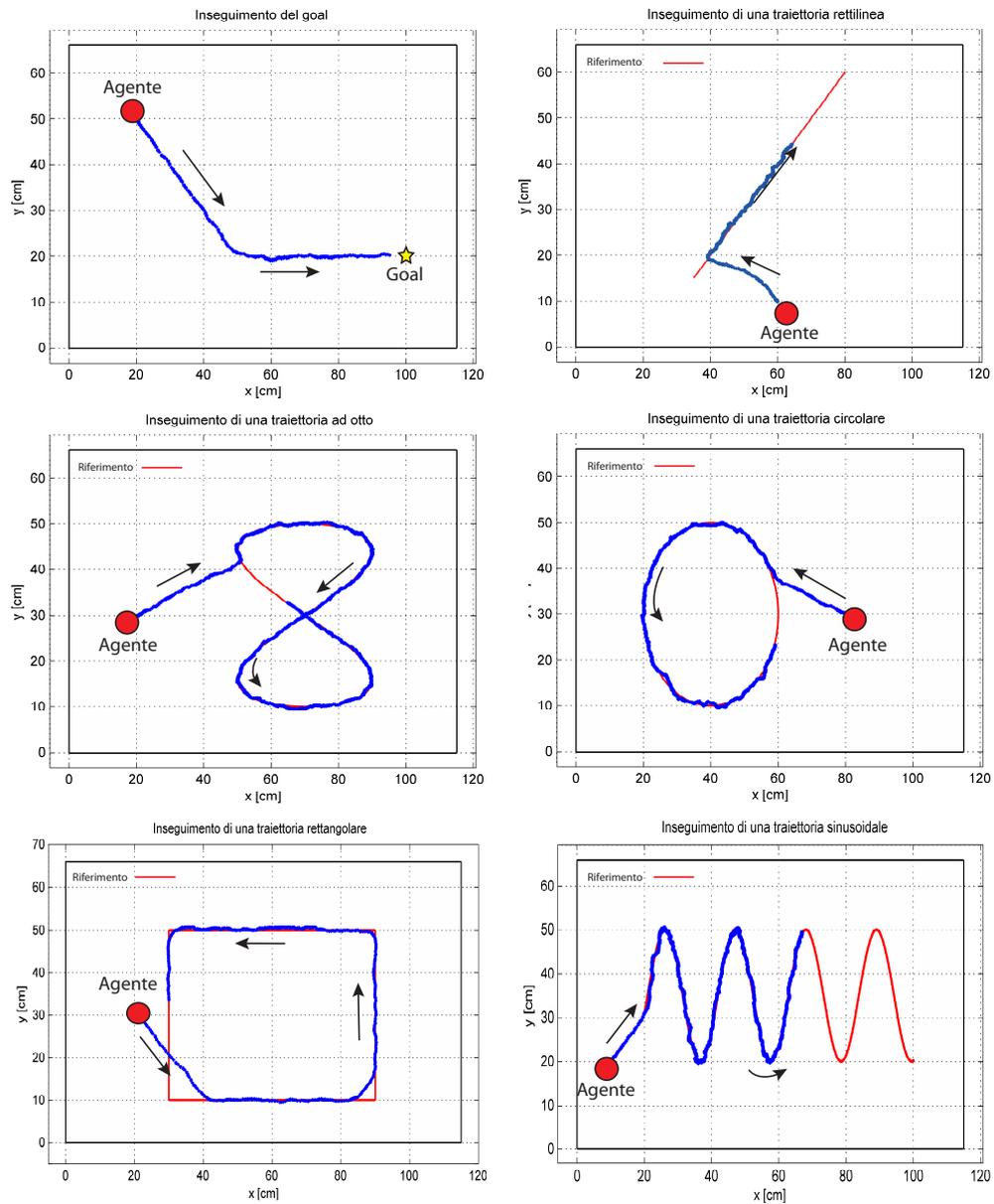


Figura 6.1: Raggiungimento di un obiettivo e inseguimento di traiettorie di riferimento

Per meglio comprendere quanto detto si faccia riferimento alla Figura 6.2 in cui sono riportati i grafici di accelerazione e velocità lungo gli assi.

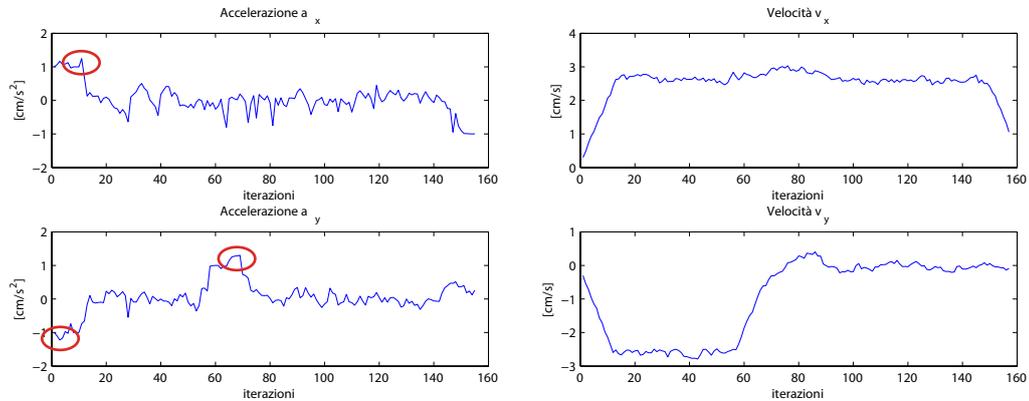


Figura 6.2: Accelerazioni e velocità lungo gli assi nel caso di inseguimento di un agente verso un goal in un piano libero da collisioni

Si ricordi, inoltre che i vincoli espressi rispetto alle variabili del sistema disturbato sono state poste in probabilità. Si noti, infatti, che in tre istanti differenti il vincolo viene violato (punti cerchiati in rosso). Un modo per ovviare a questo inconveniente è quello di permettere una definizione “asimmetrica” dei vincoli su a_x e a_y (che sono imposti per garantire il rispetto della saturazione di $a = \sqrt{a_x^2 + a_y^2}$) a seconda della posizione del *goal* rispetto alla posizione di partenza dell’agente.

6.2 Navigazione con Obstacle e Collision Avoidance

Nelle seguenti simulazioni uno o più agenti devono raggiungere un *goal* predefinito oppure seguire traiettorie di riferimento, senza collidere fra loro o con ostacoli presenti nel piano di lavoro.

La prima prova (si faccia riferimento alla Figura 6.3) prevede che un agente, per raggiungere il proprio *goal* debba aggirare l’ostacolo. In particolare sono messe a confronto due simulazioni identiche dal punto di vista della confi-

gurazione dello spazio di lavoro ma che sfruttano un orizzonte di predizione differente. Si può notare come, aumentando opportunamente l'orizzonte di predizione N (in particolare nel primo caso $N = 15$ e nel secondo caso $N = 35$), si ottenga una traiettoria più regolare. Ovviamente è bene tenere in considerazione che un aumento eccessivo dell'orizzonte potrebbe portare ad un problema troppo complesso computazionalmente. Si deve optare quindi per un corretto *trade-off* tra traiettoria ottenuta e complessità del problema.

In quasi tutte le simulazioni gli ostacoli sono stati approssimati a poligoni costituiti da 20 lati. Tale scelta rappresenta un buon compromesso tra accuratezza nell'approssimazione del poligono alla circonferenza e probabilità che il robot mobile non si “blocchi” di fronte all'ostacolo. Si faccia riferimento alla Figura 6.4 per comprendere quando detto. Si può notare che l'agente non riesce a superare l'ostacolo nel caso in cui quest'ultimo sia approssimato con un poligono definito da solo 4 lati, trovandosi di fronte ad un muro invalicabile. Si osservi che però questa situazione non porta ad una perdita di *feasibility*, ma solamente alla impossibilità di superare l'ostacolo. Ad ogni passo vi è, infatti, una soluzione al problema di ottimizzazione, che si traduce nel mantenere l'agente fermo di fronte all'ostacolo. Nel caso in esame è però necessaria una approssimazione del cerchio con un poligono di 40 lati affinché il robot possa aggirare l'ostacolo ed evitare situazioni di blocco. Questo è dovuto al fatto che la simulazione in Figura 6.4 rappresenta un caso di perfetta simmetria del sistema (l'agente, l'ostacolo e il goal sono perfettamente allineati tra loro). In una tale situazione è necessario intervenire sul corretto compromesso tra numero di lati del poligono ed orizzonte di predizione N . Nel caso specifico si è deciso di mantenere l'orizzonte di predizione costante ed aumentare il numero di lati del poligono.

Una soluzione alternativa al problema appena descritto può essere quella di rendere il sistema non simmetrico mediante l'utilizzo di un *goal* temporaneamente. In particolare si fissa un *goal* iniziale, e se si verifica una situazione

di stallo, che avviene nel momento in cui l'agente è fermo o quasi (velocità minore di 0.1 m/s) ma non si trova nel goal definito, la posizione del *goal* viene variata temporaneamente, permettendo all'agente di liberarsi dalla situazione di stallo e giungere così al *goal* originario.

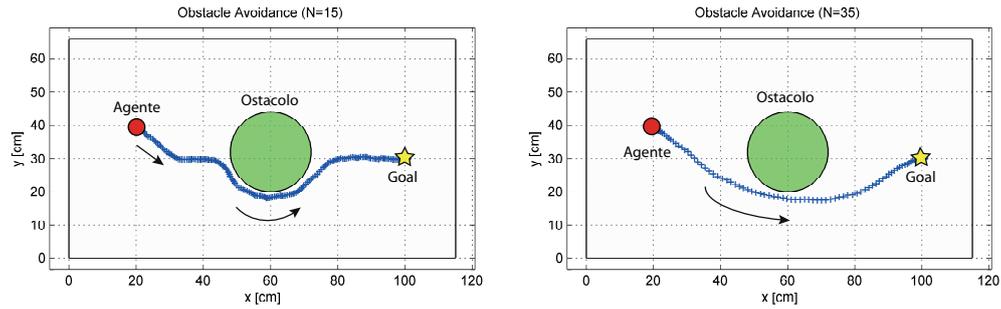


Figura 6.3: *Obstacle Avoidance*: confronto sulla scelta dell'orizzonte di predizione

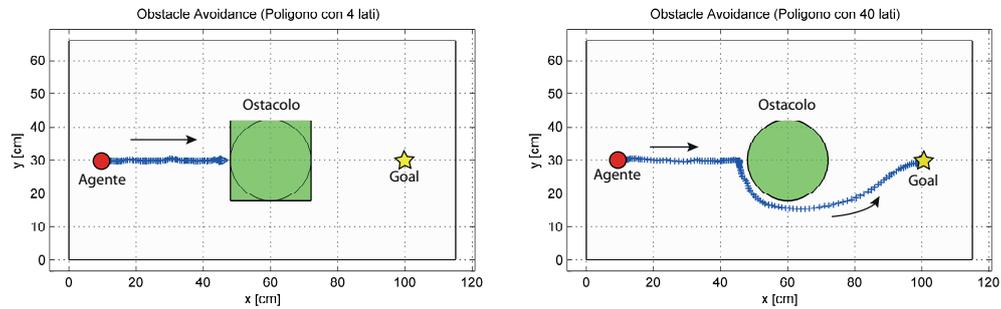


Figura 6.4: *Obstacle Avoidance*: confronto sulla scelta del numero di lati del poligono

La simulazione seguente descrive l'inseguimento di una traiettoria ad otto in presenza di ostacoli e vincoli sullo spazio operativo. In Figura 6.5 è mostrata la traiettoria simulata dell'agente. Si può notare come l'agente segua bene il riferimento e contemporaneamente eviti gli ostacoli. Come si nota dalla figura la traiettoria di riferimento esce dal piano di lavoro.

L'agente viola con una probabilità del 30% il vincolo definito dallo spazio di lavoro. Nelle prossime simulazioni si analizzerà maggiormente la violazione probabilistica dei vincoli.

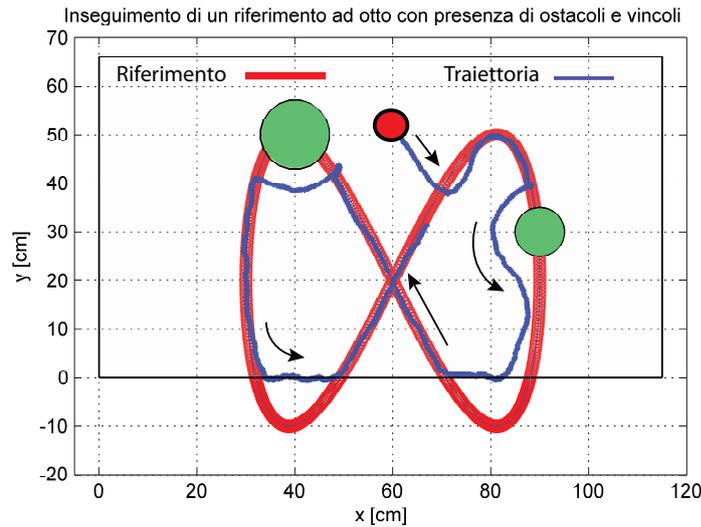


Figura 6.5: Inseguimento di una traiettoria ad otto in presenza di ostacoli e vincoli sullo spazio di lavoro

Nelle prossime simulazioni si considera il problema di *collision avoidance*, in cui ogni agente è circoscritto da un politopo definito da 20 lati. Nelle Figure 6.6 e 6.7 sono mostrati i risultati di due differenti simulazioni. In particolare in Figura 6.6 due agenti si portano in punti del piano simmetrici rispetto alla condizione iniziale; mentre in Figura 6.7 vi è l'utilizzo congiunto di *Obstacle* e *Collision Avoidance* in cui due agenti devono dapprima raggiungere due *goal* parziali, evitando l'ostacolo presente, ed una volta raggiunto invertire la loro posizione. È importante notare che non vi è collisione tra i due robot mobili, in quanto le apparenti sovrapposizioni tra le loro traiettorie rappresentano il passaggio dei robot in istanti differenti.

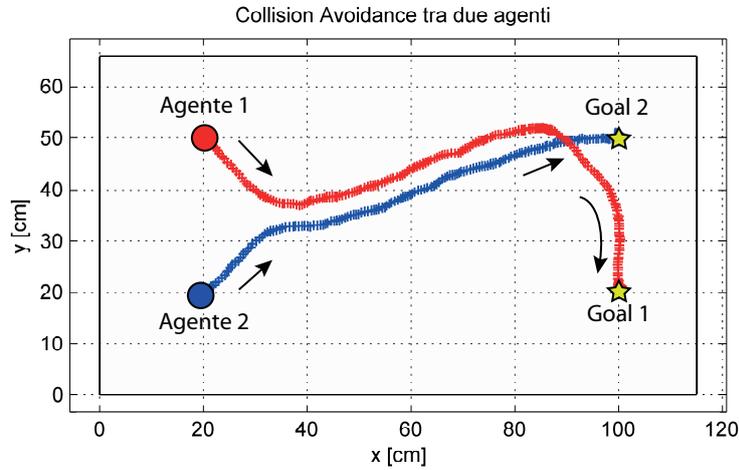


Figura 6.6: *Collision Avoidance* tra due agenti

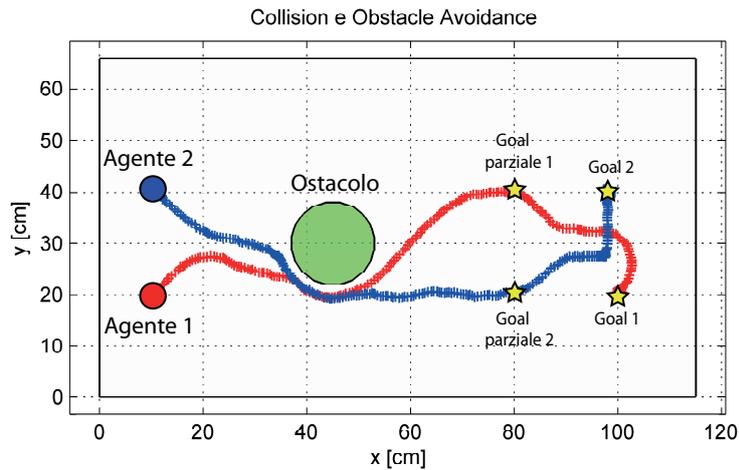


Figura 6.7: *Collision e Obstacle Avoidance* tra due agenti

Anche in questo caso possono verificarsi situazioni di simmetria che generano situazioni di stallo, non permettendo quindi agli agenti di procedere. In questo caso, aumentando in maniera opportuna l'orizzonte di predizione, è possibile risolvere il problema. Si faccia riferimento alla Figura 6.8 per comprendere quanto appena descritto.

In particolare nella figura a sinistra (dove $N = 15$) si verifica una situazione

di stallo, mentre nella figura a destra ($N = 35$) gli agenti raggiungono gli obiettivi.

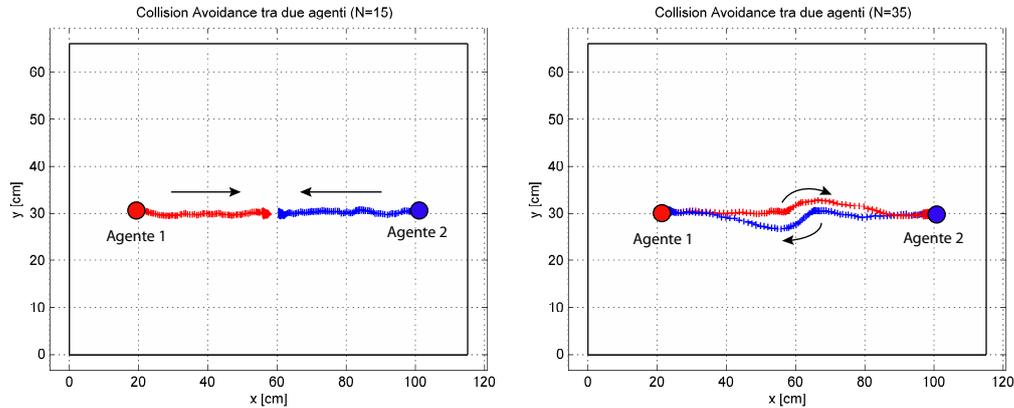


Figura 6.8: *Collision Avoidance*: confronto sulla scelta di N nel caso di simmetria del sistema

6.3 Violazione dei vincoli

In questa sezione saranno illustrate delle prove volte ad analizzare in dettaglio le proprietà dell'algorithmo di garantire il soddisfacimento di vincoli con probabilità predeterminata.

Si consideri il caso in cui un agente si muova lungo un traiettoria di riferimento rettilinea e orizzontale, che corrisponde anche alla retta che rappresenta un vincolo sulla posizione rispetto all'asse delle ordinate (in particolare si richiede $y \geq 50$). L'algorithmo è progettato in modo tale che l'agente violi il vincolo secondo una percentuale predefinita. In particolare, si fissa una probabilità di violazione e si riportano le traiettorie dello stato per le probabilità $p = 0.1$, $p = 0.2$ e $p = 0.3$.

Nelle Figure 6.9, 6.10 e 6.11, oltre alle traiettorie si riporta anche il tasso di violazione del vincolo e la strategia di inizializzazione utilizzata ad ogni passo. In particolare nella Figura 6.9 il numero dei punti al di sotto della retta in rosso sono 14; in Figura 6.10 sono 19 e infine in Figura 6.11 so-

no 32. Si nota inoltre che, ogni volta che il vincolo è violato, la strategia di inizializzazione usata dall'algoritmo è quella che prevede di impostare $\bar{x}_{t|t} = \bar{x}_{t|t-1}$; d'altro canto negli istanti in cui il vincolo non è violato la strategia utilizzata dipende, come detto, dal valore ottenuto dal funzionale di costo in corrispondenza delle due strategie di inizializzazione dello stato.

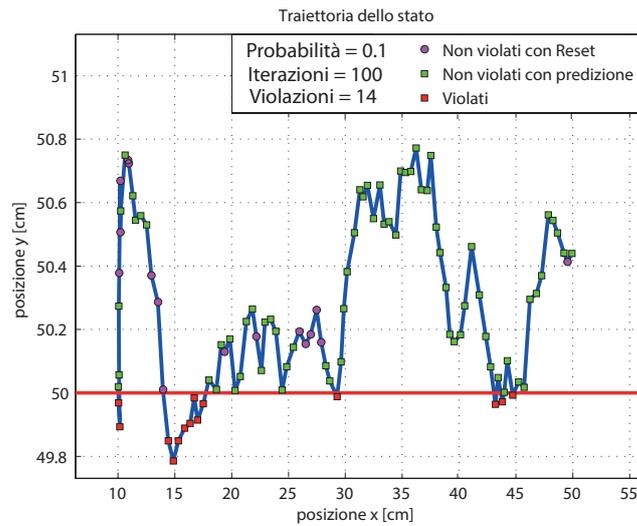


Figura 6.9: violazioni del vincolo con probabilità del 10%

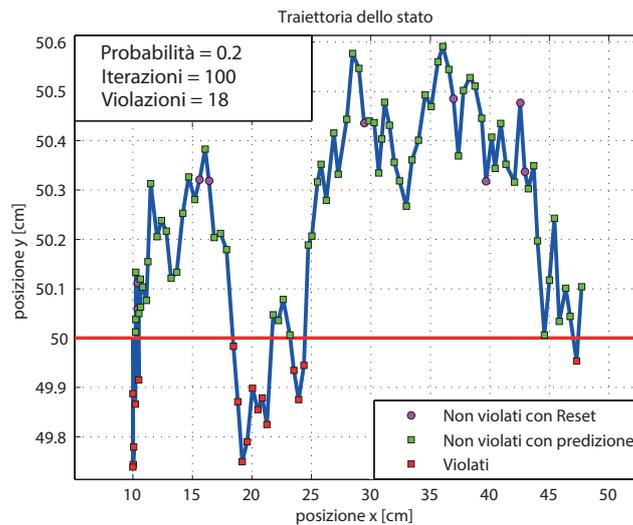


Figura 6.10: violazione del vincolo con probabilità del 20%

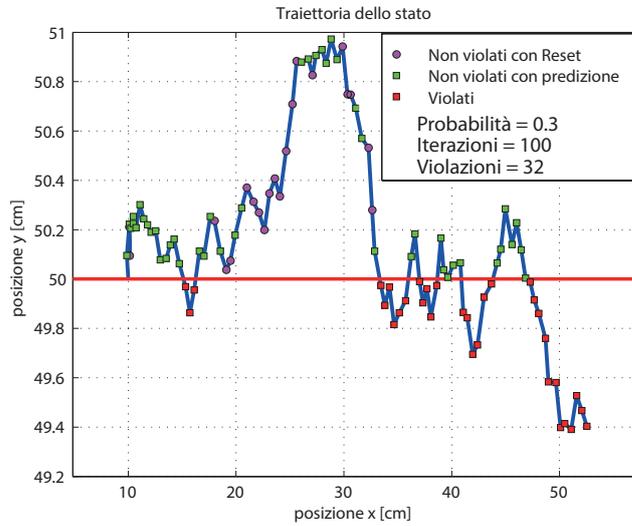


Figura 6.11: violazione del vincolo con probabilità del 30%

Si supponga ora di far partire un agente dal punto di coordinate $(20, 60)$ e di farlo muovere lungo un vincolo definito dalla retta di equazione $y = 40$. Questo è ottenuto semplicemente ponendo il *goal* nel punto di coordinate $x = 70, y = 10$. La stessa simulazione viene ripetuta 100 volte in modo da ottenere un numero considerevole di prove. In Figura 6.12 si riportano le traiettorie e si consideri in particolare la zona maggiormente densa di punti (intorno di $(70 - 40)$). È possibile verificare che l'insieme di punti che si trovano al di sotto della retta rappresentata in rosso è all'incirca il 30% dei punti totali che interessano la zona. Da ultimo ci si aspetta che le traiettorie in questa fascia abbiano una distribuzione gaussiana; si ricorda, infatti, che i disturbi sono stati simulati come gaussiani. La Figura 6.13 mostra quanto detto.

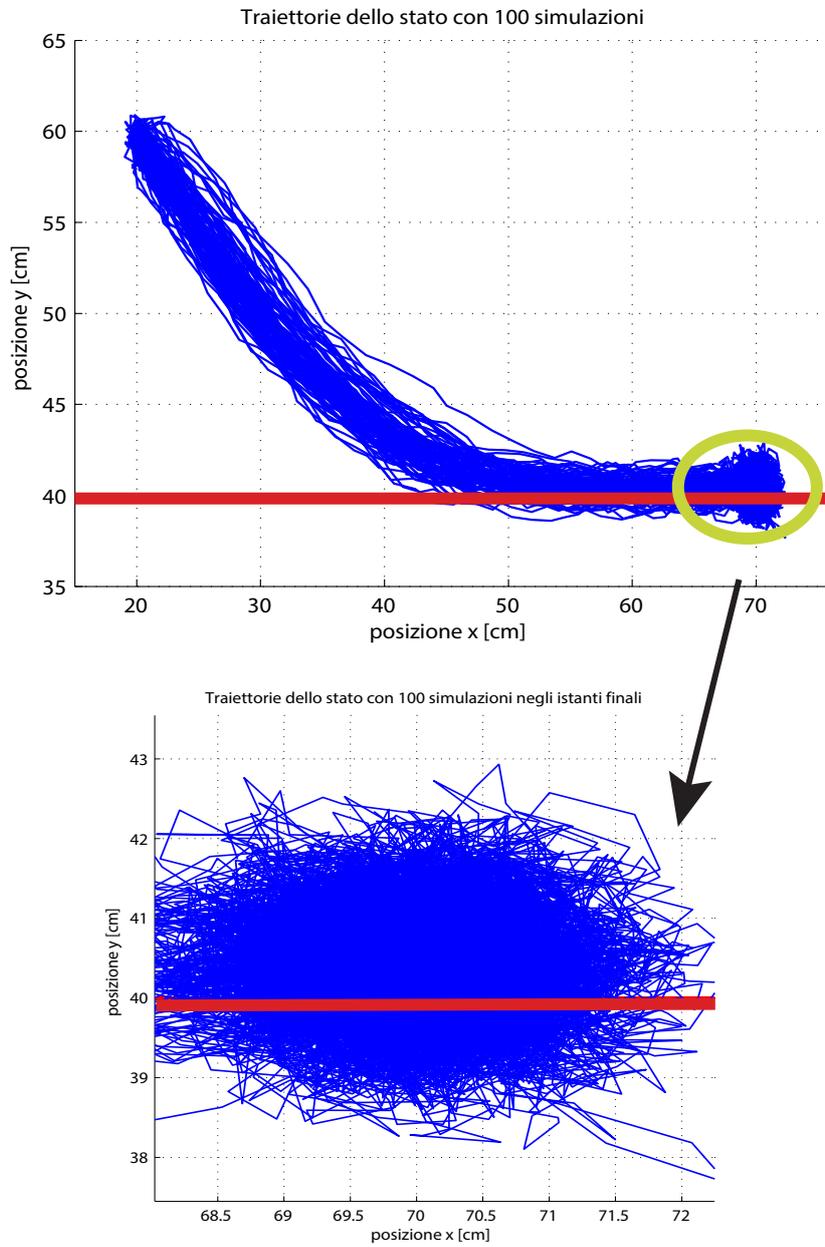


Figura 6.12: Traiettorie dello stato in 100 simulazioni

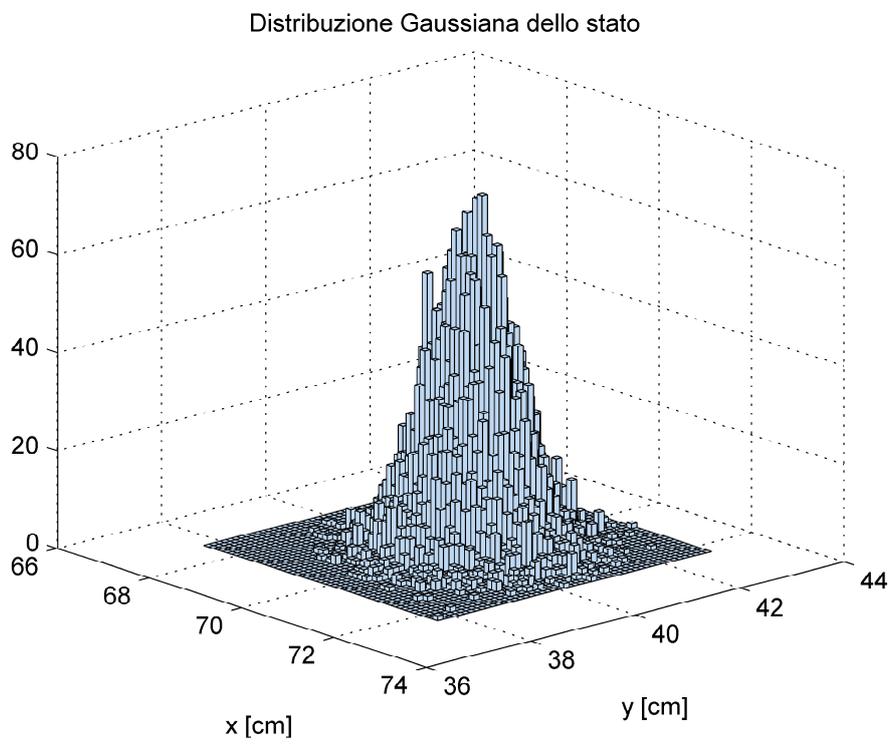


Figura 6.13: Distribuzione normale delle traiettorie dello stato

Capitolo 7

Risultati Sperimentali

In questo capitolo saranno mostrati i risultati di varie prove effettuate sull'apparato sperimentale descritto nei Capitoli 2 e 3. In particolare saranno effettuate prove simili a quelle analizzate in simulazione così da poter confrontare i risultati e validare l'algoritmo descritto nel capitolo 2. Inizialmente saranno effettuate prove con un singolo robot *e-puck*, per poi passare a prove in cui sono presenti tutti e tre i robot. Infine sarà mostrata una prova in cui la varianza del disturbo sarà stimata *on-line*.

Nel capitolo 4 si è detto che la posizione corrente assunta dal robot è ricavata dalla telecamera posizionata sopra il banco di lavoro. Per riferire le posizioni assunte dai robot si è segnalato ogni robot con un triangolo colorato; a questo punto la telecamera sfrutta dapprima un algoritmo di *colour detection* in *RGB* per individuare i robot, dopodiché mediante un algoritmo di *corner detection* vengono individuati i vertici del triangolo e mediante semplici passaggi di geometria basilare viene ricavato il punto corrispondente alla posizione corrente del robot. Per maggiori dettagli su come avviene l'algoritmo si rimanda la trattazione a [5] e [6].

7.1 Prove di navigazione con Obstacle Avoidance e Collision Avoidance

In questa prima prova si considera il problema di far inseguire ad un robot *e-puck* una traiettoria di riferimento. In particolare la traiettoria da seguire è di tipo sinusoidale e di equazione $y = 30 + 20 \sin(0.1x)$, con x che varia tra 20 cm e 110 cm. Sono inoltre presenti due ostacoli nel piano di lavoro, in coordinate corrispondenti a dei punti sulla traiettoria di riferimento. In Figura 7.1 è riportato l'andamento della traiettoria del robot. Si può notare come il riferimento sia sempre ben seguito, tranne in quei punti in cui, dovendo rispettare i vincoli di *obstacle avoidance*, l'agente è in grado di aggirare l'ostacolo. Una volta evitati gli ostacoli, l'*e-puck* torna a seguire il riferimento.

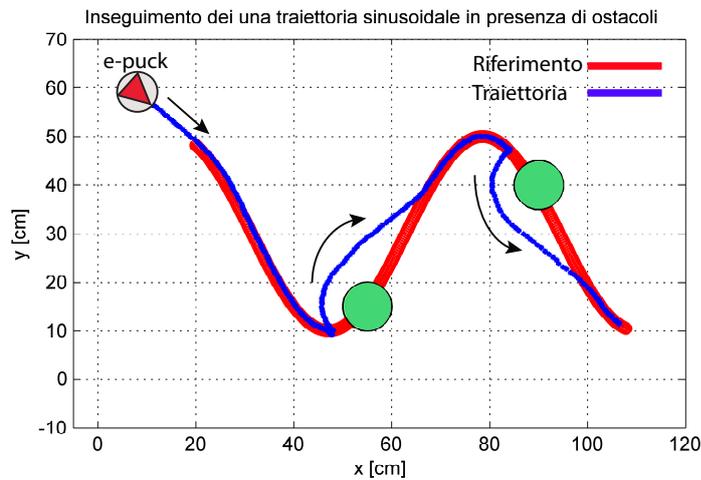


Figura 7.1: Inseguimento di una traiettoria sinusoidale in presenza di ostacoli

Nelle prove successive si consideri il caso in cui un robot *e-puck* deve raggiungere un *goal* senza collidere con gli ostacoli fissi presenti nel piano di lavoro e con gli altri robot presenti. Nella prima prova un solo robot *e-puck* deve raggiungere il *goal* senza collidere con gli ostacoli. In Figura 7.2 si riporta la traiettoria eseguita dal robot *e-puck*, mentre in Figura 7.3 vi è

una sequenza di foto che ne mostra l'andamento sul piano di lavoro. Gli ostacoli, come già detto sono in posizioni note e supposti circolari; inoltre sono stati approssimati con poligoni formati da un numero di lati pari a 20. L'orizzonte di predizione scelto è pari a $N = 15$. In entrambe le figure si nota come effettivamente il robot eviti correttamente le collisioni.

Si noti anche che al fine di generare traiettorie che effettivamente evitino collisioni si è dovuto aumentare il livello di confidenza di violazione dei vincoli, rispetto al caso in simulazione, in modo tale da garantire una corretta *obstacle avoidance*. Infine vengono anche mostrati gli andamenti del tempo di calcolo della soluzione del problema di MPC per l'agente e della velocità, rispettivamente nelle Figure 7.4 e 7.5. In particolare si può notare in Figura 7.4 come vi sia un campione, precisamente il primo (superiore a 0.25 s), più elevato degli altri. Questo è dovuto all'inizializzazione della telecamera, che richiede tempo maggiore, mentre tutti gli altri sono ampiamente inferiori a 0.2 s. Per quanto riguarda la velocità, l'andamento dipende ovviamente dallo spazio di lavoro; la velocità massima raggiunta è in questo caso poco superiore a 4 cm/s. Si nota che tale limite è legato ai limiti imposti sulla accelerazione del robot, i quali possono essere ulteriormente rilassati rispetto ai valori utilizzati attualmente.

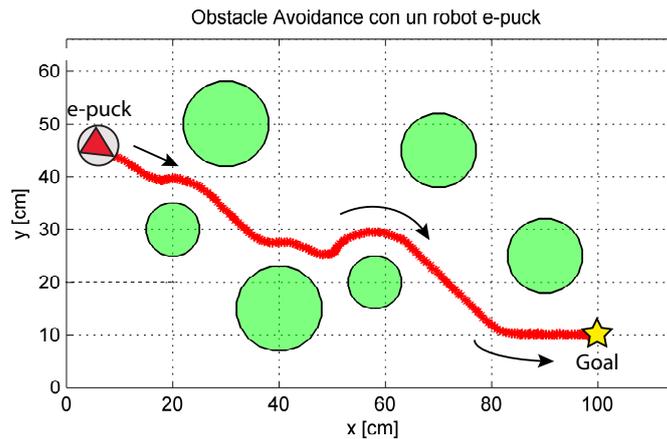


Figura 7.2: Navigazione con *Obstacle Avoidance*: traiettoria del robot *e-puck*

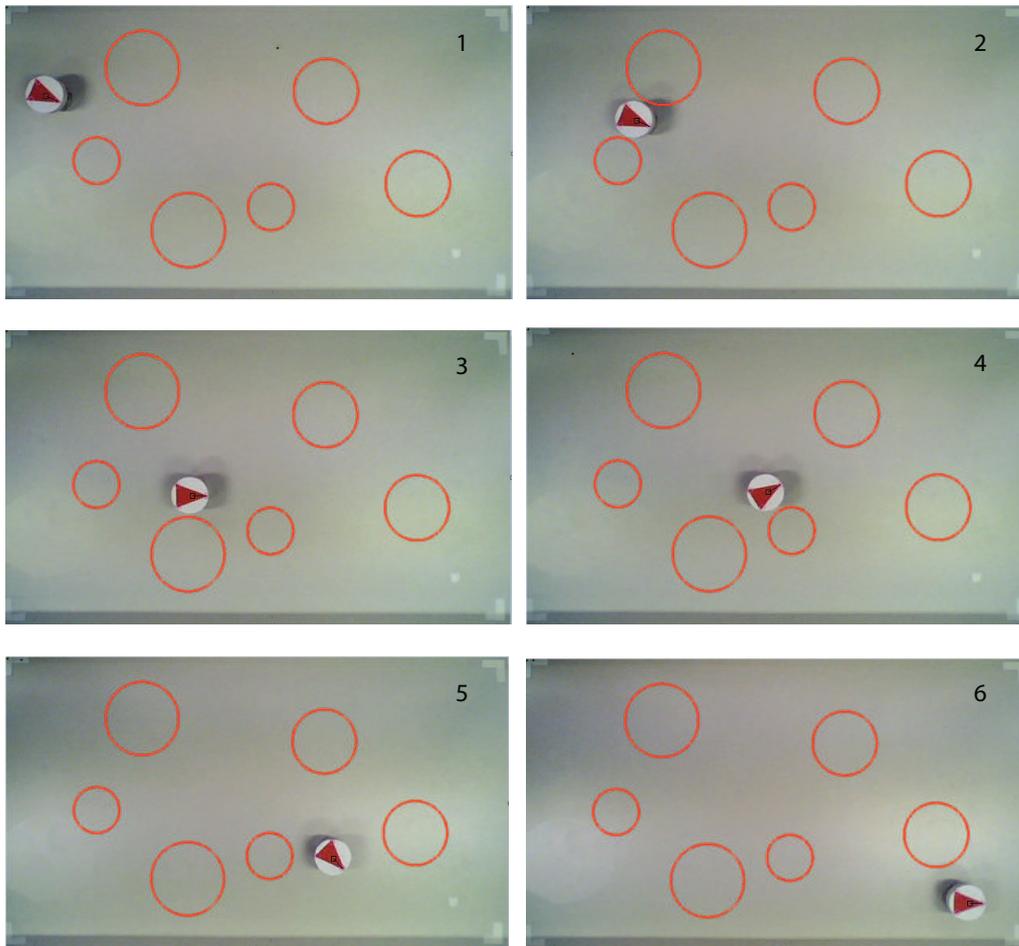


Figura 7.3: Navigazione di un *e-puck* verso un obiettivo con *obstacle avoidance*

Si nota che è possibile aumentare la velocità del robot aumentando opportunamente anche l'orizzonte di predizione. In particolare la taratura deve essere tale da non generare un tempo di calcolo troppo elevato. Le prove seguenti chiariranno maggiormente questo aspetto.

Nelle prossime due prove si considerano due robot che, in uno spazio privo di ostacoli fissi, devono raggiungere i propri obiettivi senza collidere fra loro. Si riportano le traiettorie (Figure 7.6 e 7.8) e le sequenze di foto (Figure 7.7 e 7.9), in cui vengono mostrati i risultati.

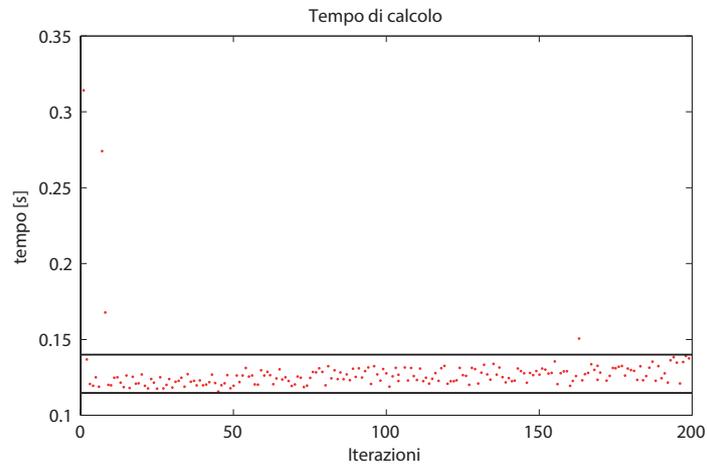


Figura 7.4: Navigazione con *Obstacle Avoidance*: tempo di calcolo

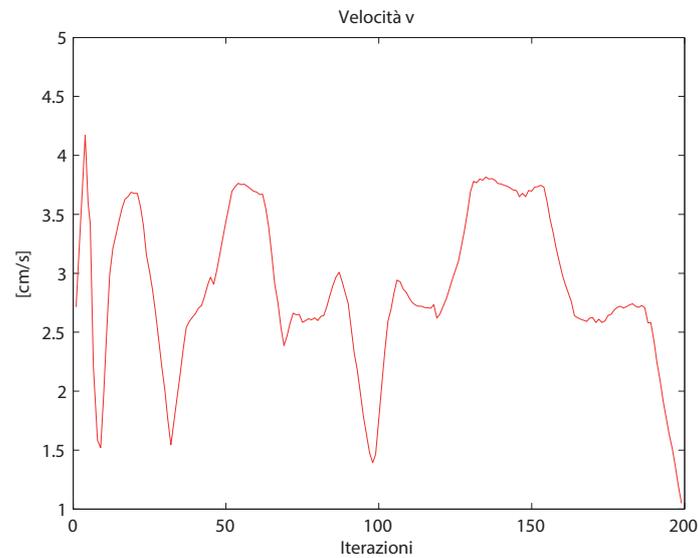


Figura 7.5: Navigazione con *Obstacle Avoidance*: velocità

In entrambe le prove si è utilizzato un orizzonte di predizione $N = 15$; inoltre nella prova in Figura 7.9 si è impostato un *goal* tempo variante al fine di evitare una situazione di stallo. La scelta di un *goal* tempo variante è stata

utilizzata poiché il rumore agente non era sufficiente a rendere asimmetrico il sistema, come descritto nel capitolo 6.

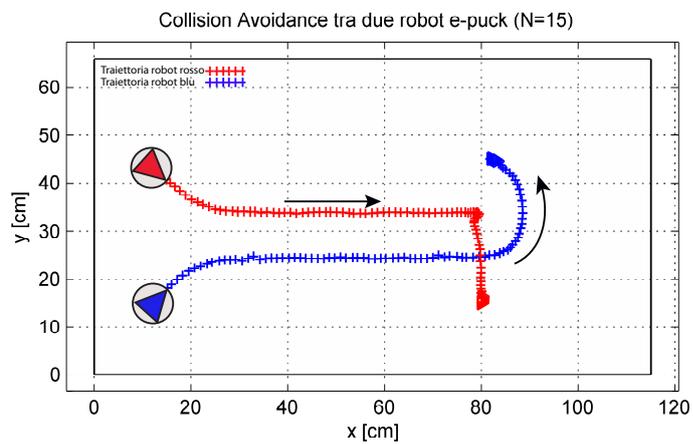


Figura 7.6: Traiettorie: Collision Avoidance tra due robot e-puck

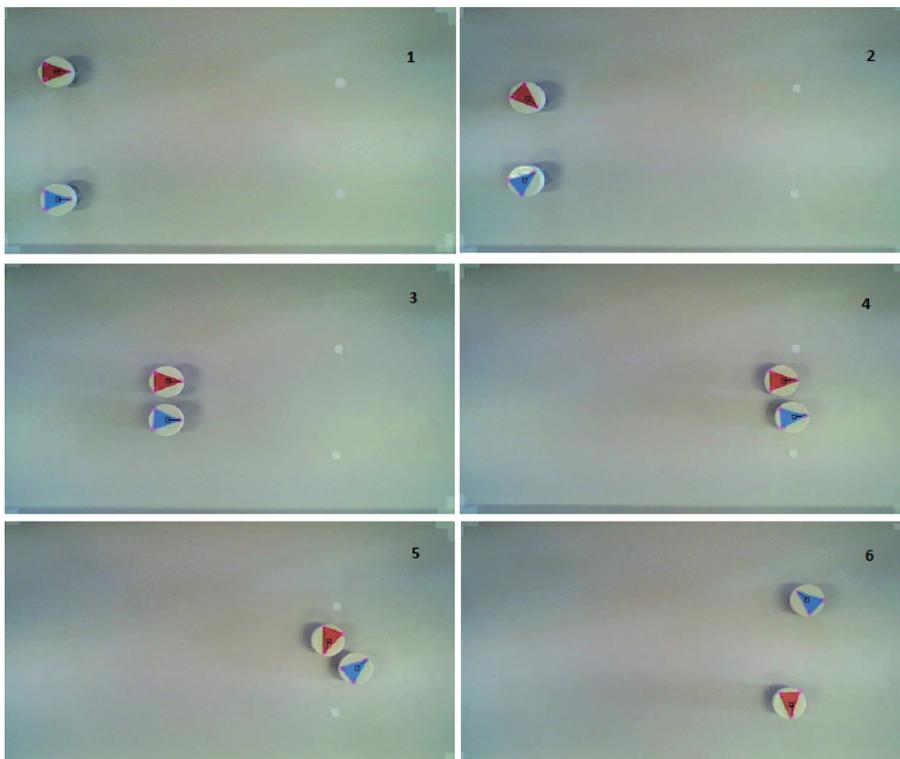


Figura 7.7: Collision Avoidance tra due robot e-puck

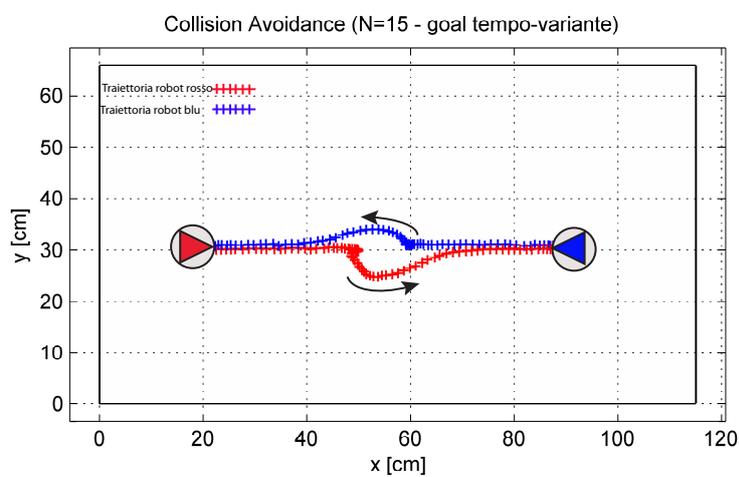


Figura 7.8: Traiettorie: Collision Avoidance tra 2 robot e-puck

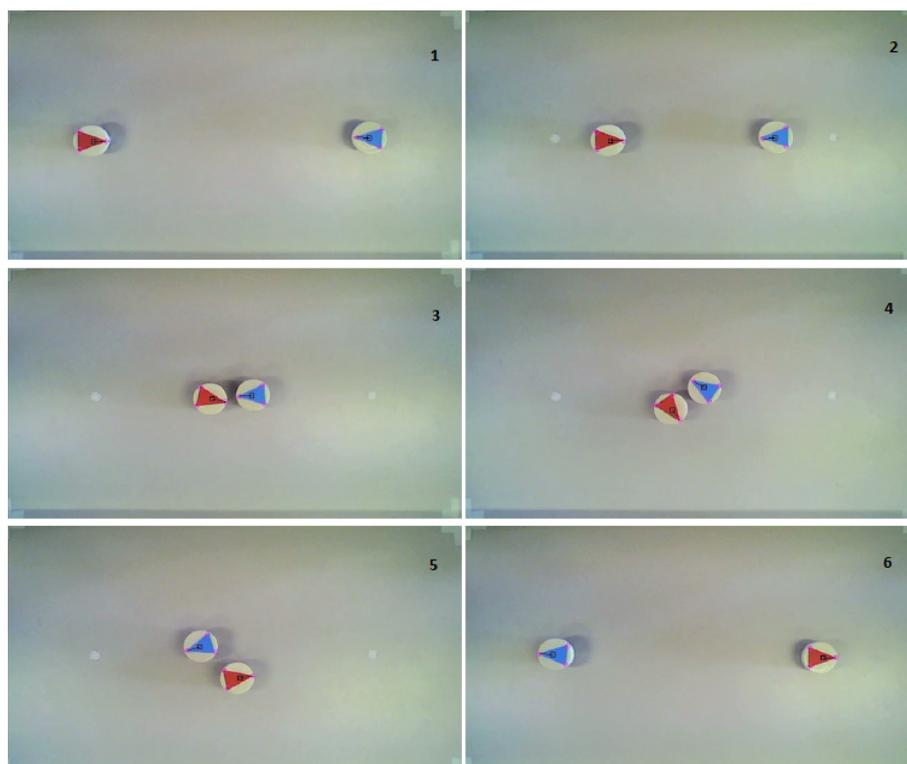


Figura 7.9: Collision Avoidance tra 2 robot e-puck

Come detto, è possibile verificare che la velocità dei robot durante la

navigazione può essere aumentata tarando opportunamente l'orizzonte di predizione N . In particolare si faccia riferimento alla Figura 7.10 nella quale si riporta l'andamento delle velocità dei due robot. In particolare viene messo a confronto l'andamento delle velocità svolgendo la prova anche con un orizzonte di predizione pari a $N = 35$. Si noti che per la prova con $N = 35$ non è stata riportata né la traiettoria né la sequenza di foto del robot poiché non si riscontrano, nella traiettoria seguita dagli agenti, significative differenze rispetto al caso $N = 15$. Come è possibile notare, una significativa differenza si riscontra nella velocità e nel corrispondente tempo massimo impiegato per raggiungere il *goal*.

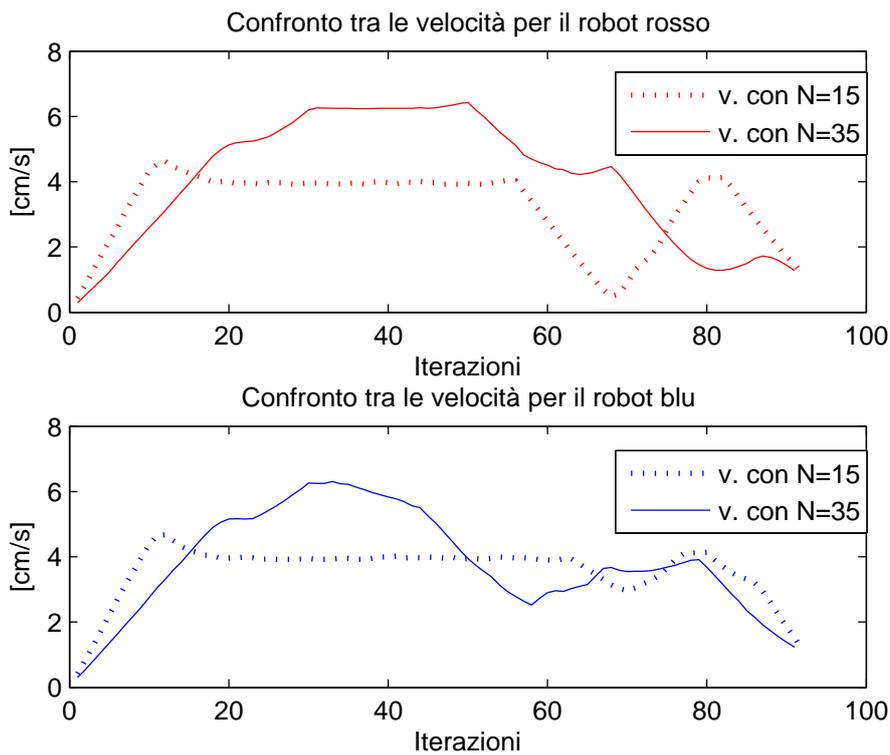


Figura 7.10: Confronto tra le velocità dei robot con due orizzonti di predizione differenti

Si riportano a questo punto i tempi di calcolo dell'algoritmo (Figura 7.11). Come detto, nel caso si utilizzi un orizzonte di predizione più elevato ($N = 35$) il tempo di calcolo complessivo risulta essere maggiore. Si noti che i tempi di calcolo sono abbastanza costanti durante l'intera prova; in particolare sono state identificate delle fasce in cui esso rimane limitato. D'altra parte una continua crescita del tempo di calcolo genererebbe dei ritardi inaccettabili nell'applicazione della legge di controllo.

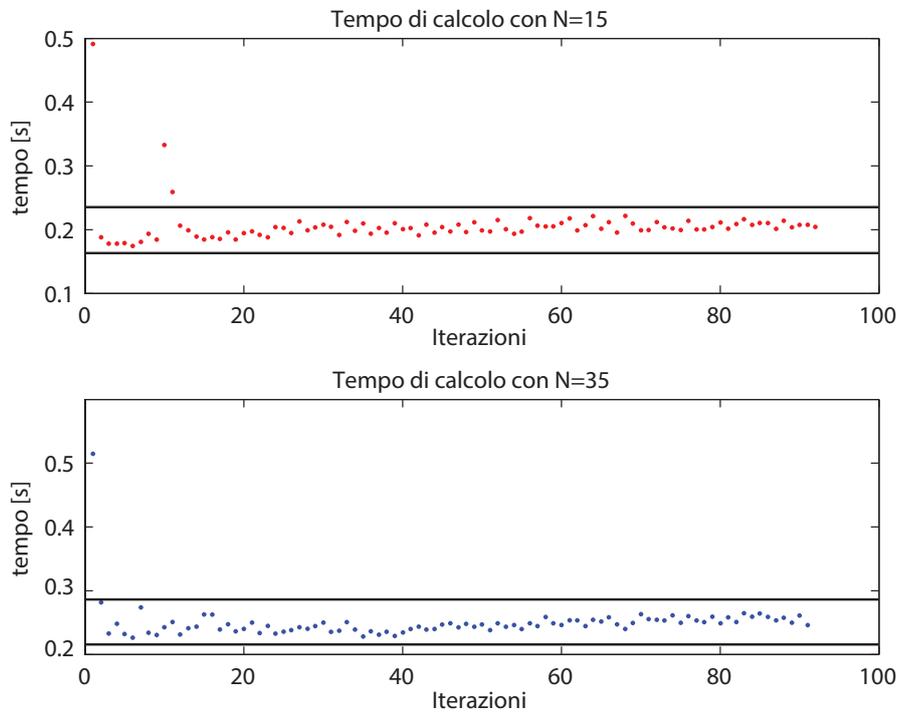


Figura 7.11: Confronto tra i tempi di campionamento con due orizzonti di calcolo differenti

Nella seguente prova si considerano tre robot in uno spazio in cui sono presenti 3 ostacoli e con un orizzonte pari a $N = 15$. La prova mostra i risultati dell'algoritmo congiunto di navigazione con *obstacle* e *collision avoidance*. Nelle Figure 7.12 e 7.13 sono riportate le traiettorie seguite dai

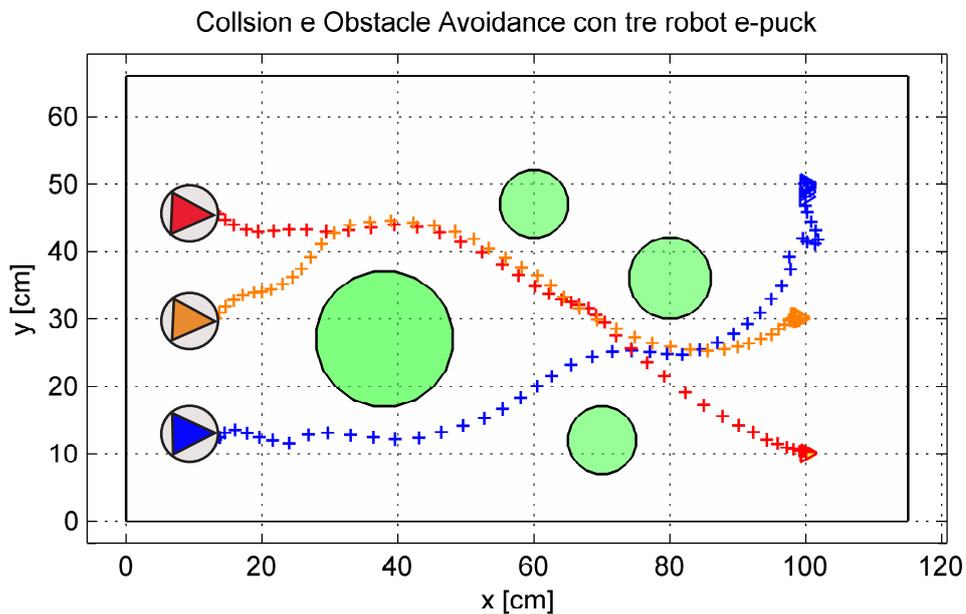


Figura 7.12: Traiettorie: *Collision e Obstacle Avoidance* tra tre robot e-puck

robot e la sequenza di foto che mostra l'andamento sul piano piano di lavoro. Si può notare che i tre *e-puck* durante il loro moto riescono perfettamente a coordinarsi al fine di non collidere tra loro e con gli ostacoli presenti.

7.2 Prove sulla stima adattativa

La presente prova mostra i risultati ottenuti sulla stima adattativa della varianza, descritta nel capitolo 5. Nello specifico la prova consiste nel far inseguire ad un robot *e-puck* una traiettoria rettangolare che interseca un vincolo come mostrato in Figura 7.14. In particolare il vincolo imposto è $y > 10$ e si richiede al robot di percorrere la traiettoria rettangolare tre volte, così da ricavare un numero di dati sufficiente.

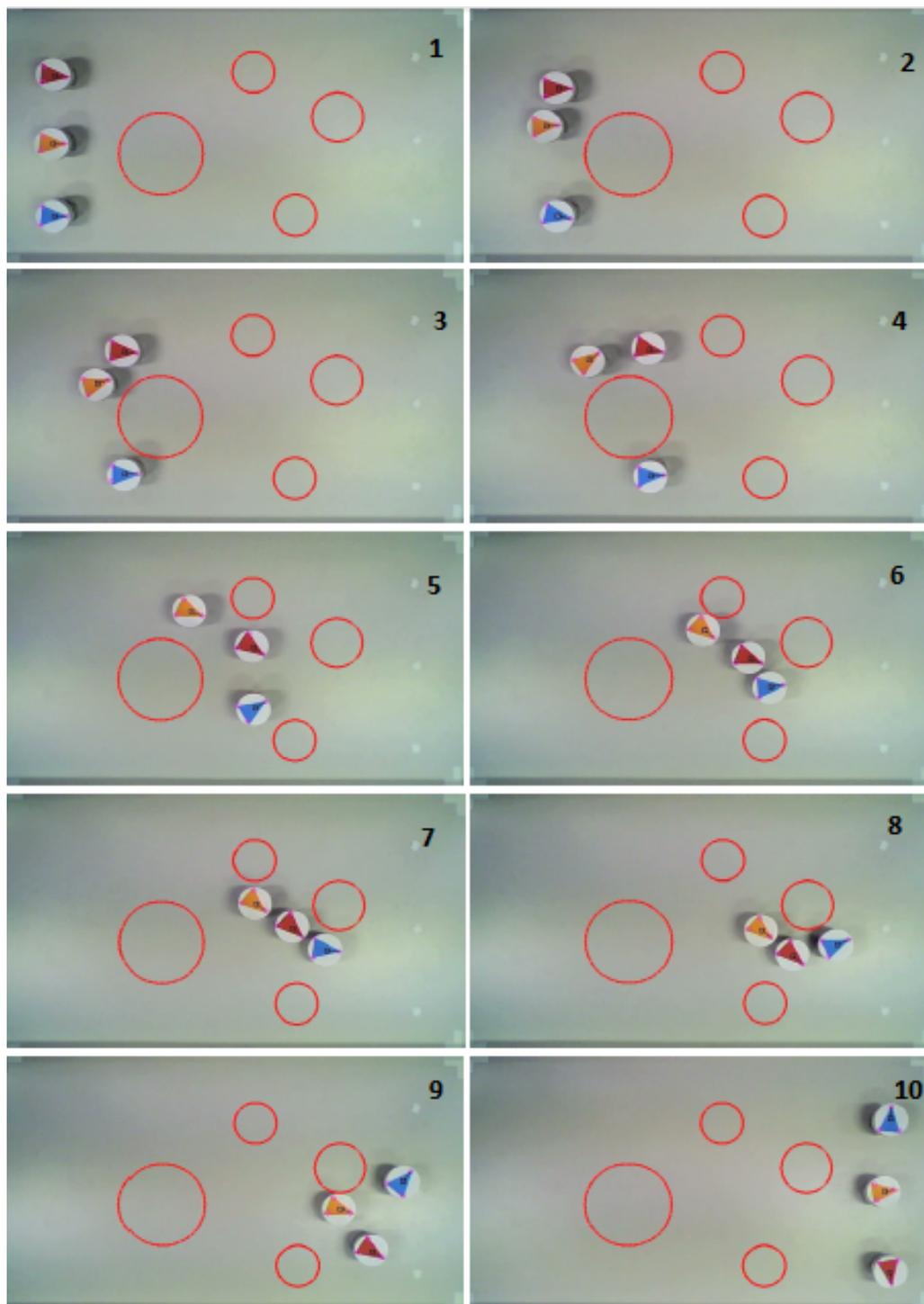


Figura 7.13: *Collision e Obstacle Avoidance* tra tre robot e-puck

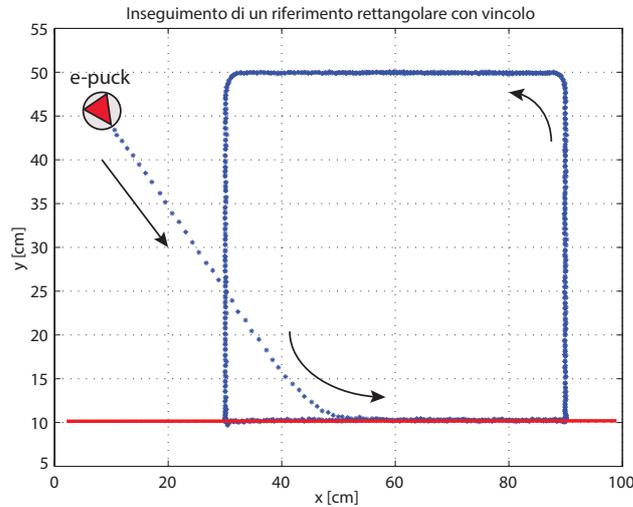


Figura 7.14: Traiettoria rettangolare seguita dall'*e-puck*

A questo punto si è inizializzata la varianza del rumore iniziale ad un valore pari a tre volte il valore stimato nel capitolo 4. Si è scelto inoltre di aggiornare la stima ad ogni giro, per un totale di tre volte.

I prossimi grafici mostrano i seguenti risultati: in Figura 7.15 vengono riportati contemporaneamente gli andamenti della coordinata y e l'andamento della stima. In particolare si può notare come la varianza venga stimata correttamente, i valori singolari massimo e minimo della matrice di covarianza utilizzata (W) convergono, infatti, a quelli reali calcolati *off-line* (matrice indicata in figura con W^*). Dalla Figura 7.15 si nota che la stima della varianza viene aggiornata non ad ogni passo, ma solamente prima che il robot raggiunga il vincolo in modo tale da non generare dinamiche indesiderate. È, quindi, possibile dimostrare la correttezza della stima, infatti facendo un ingrandimento della traiettoria della coordinata y (Figura 7.16 in alto) è possibile verificare che, quando è utilizzata una varianza maggiore la restrizione del vincolo è maggiore, ottenendo una soluzione più conservativa, mantenendo il robot più distante dal vincolo; d'altra parte man mano che la varianza diminuisce e si avvicina al valore vero la restrizione del vincolo diminuisce, permettendo all'agente di passare molto più vicino al vincolo e

di violarlo secondo la probabilità definita.

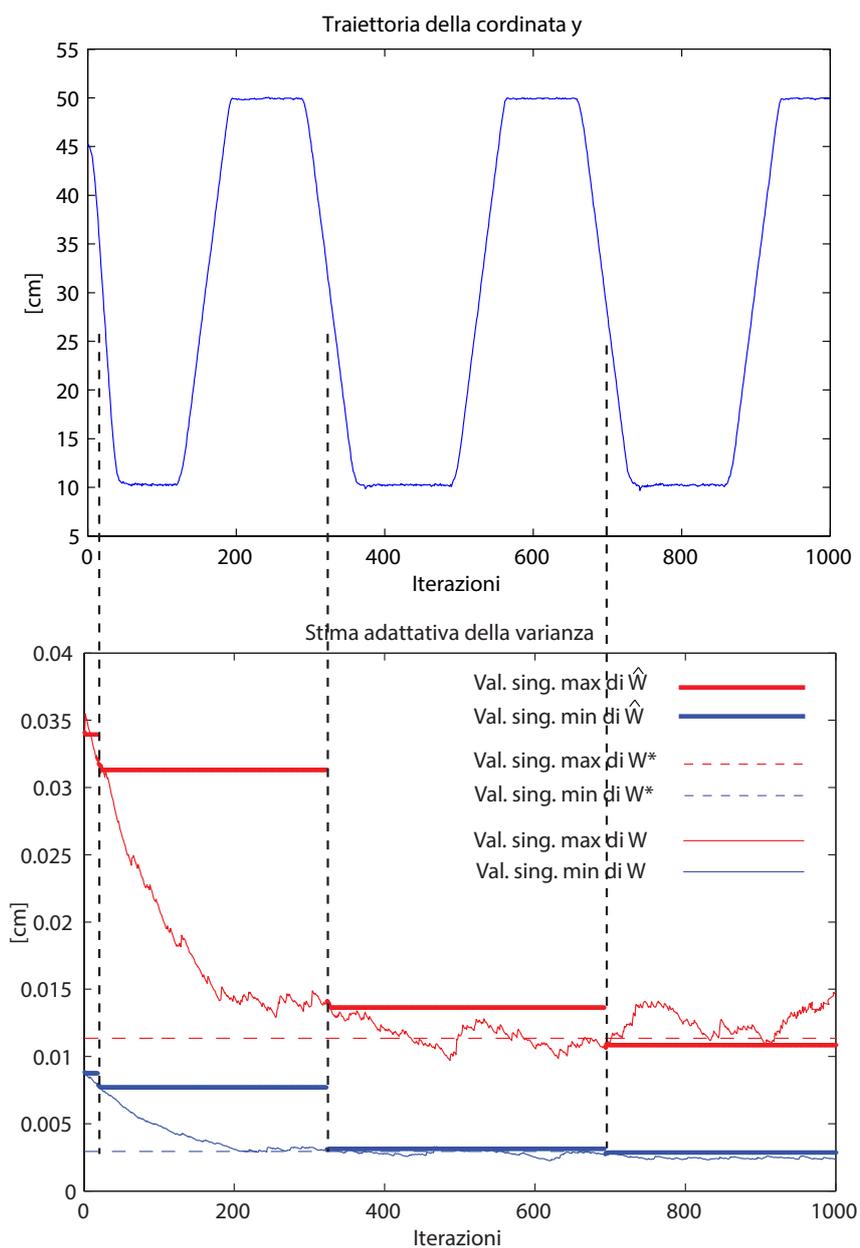


Figura 7.15: In alto: traiettoria della coordinata y . In basso: stima adattativa della varianza

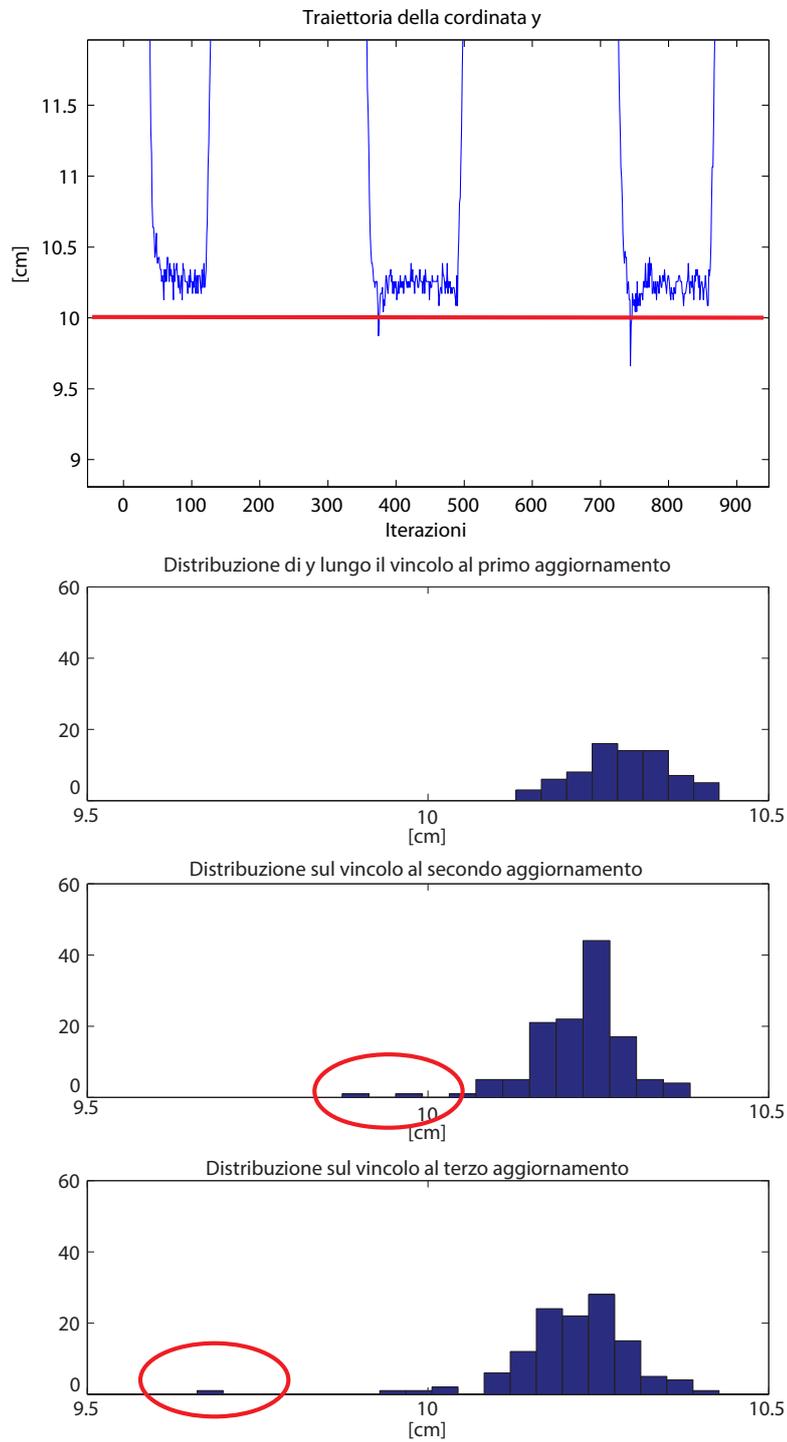


Figura 7.16: In alto: ingrandimento della traiettoria di y nei tre aggiornamenti. In basso: distribuzione di y lungo il vincolo nei tre aggiornamenti della varianza

Capitolo 8

Conclusioni e sviluppi futuri

In questo elaborato si è presentato uno schema di controllo distribuito stocastico per la gestione cooperativa di sistemi interagenti ma dinamicamente disaccoppiati. L'algoritmo descritto propone una soluzione per il controllo di sistemi di larga scala affetti da disturbi additivi modellizzabili come processi stocastici.

Inizialmente è descritta una metodologia di controllo centralizzato per la regolazione di un sistema lineare. In particolare la formulazione del problema di ottimizzazione prevede la minimizzazione di una funzione di costo dipendente dai valori attesi e dalle varianze dello stato lungo l'orizzonte di predizione. I vincoli, formulati inizialmente in termini probabilistici, sono stati riformulati in termini deterministici sfruttando la disuguaglianza di Cantelli.

Dopodiché l'algoritmo centralizzato è stato esteso al caso del *tracking*. Infine si è proposta una soluzione distribuita *ad-hoc* per la gestione cooperativa di sotto-sistemi.

L'algoritmo descritto porta con sé notevoli vantaggi, tra cui la possibilità di includere disturbi con supporto illimitato. In particolare, questo vantaggio è di fondamentale importanza se si pensa a molti algoritmi robusti già noti in letteratura, i quali, nonostante garantiscano ottime prestazioni, pongono talvolta forti limitazioni alla caratterizzazione dei disturbi stessi. Inoltre

l'algoritmo proposto non comporta il calcolo di *robust invariant set*.

L'algoritmo è stato quindi utilizzato per risolvere uno specifico problema di moto e coordinamento di robot mobili, tra cui, *obstacle* e *collision avoidance*. L'applicazione ad un apparato sperimentale ha mostrato le buone prestazioni dello schema proposto.

I vantaggi appena discussi aprono la strada ad una serie di possibili sviluppi futuri volti a migliorare ulteriormente prestazioni ed efficienza dell'algoritmo presentato. In particolare si ritiene interessante lo sviluppo nelle seguenti direzioni:

- La caratterizzazione del disturbo svolta in questo lavoro di Tesi è stata sufficiente per gli scopi teorici che ci si era proposti. Tuttavia è possibile svolgere un'analisi di identificazione dello stesso più accurata, ad esempio mediante l'utilizzo di modelli ARMA, ARMAX, ARIMAX, ecc.
- La velocità dei robot è stimata mediante un filtro di Kalman integrato nel software usato dagli stessi robot mobili. In questo lavoro l'incertezza della stima è considerata una componente del disturbo agente sullo stato. A questo punto, l'uso di tecniche di controllo *output-feedback*, che prevedono il progetto di uno stimatore dello stato integrato nello schema di controllo, potrebbero permettere di ottimizzare ulteriormente le prestazioni del controllore.
- Da un punto di vista implementativo, in questa Tesi si è utilizzato il software MATLAB sia per risolvere il problema di ottimizzazione, sia per comunicare con i robot. Tuttavia, tale software, molto potente in ambito simulativo, non è particolarmente adatto alla comunicazione con l'hardware. In vista di ciò, per render più efficiente e veloce la comunicazione con i robot, si prevede di sviluppare dei codici (ad

esempio scritti in C o in Phyton) dedicati alla sola gestione comunicativa, in particolare connessione via *bluetooth*, connessione con la webcam e rilevamento delle posizioni.

Appendice A

Implementazione

In questa appendice verranno analizzati gli aspetti implementativi dell'algoritmo presentato. In particolare tale algoritmo è stato sviluppato in ambiente MATLAB, mentre la scelta del solutore del problema di problemi di ottimizzazione quadratica è ricaduta su `cplexqp`, sviluppato da IBM. In particolare `cplexqp` trova il minimo di un problema specificato come segue:

$$\begin{cases} \min_{\Xi} \frac{1}{2} \Xi^T H \Xi + f^T \Xi \\ A_{ineq} \Xi \leq b_{ineq} \\ A_{eq} \Xi = b_{eq} \end{cases} \quad (\text{A.1})$$

in cui H e f sono funzioni note, Ξ è il vettore rispetto al quale è risolta l'ottimizzazione, mentre A_{ineq} , b_{ineq} , A_{eq} e b_{eq} sono matrici costanti che definiscono i vincoli. Per definire in modo più compatto possibile il problema di ottimizzazione, dove possibile, i vincoli di uguaglianza presenti (ad esempio l'equazione che definisce l'evoluzione del sistema) verranno usati direttamente per definire opportunamente la funzione di costo.

A partire dalla funzione di costo e dai vincoli descritti nei capitoli 3 e 5, in questa sezione, ci si propone di ricavare le matrici da fornire al solutore (A.1) necessarie all'ottimizzazione.

Funzione di costo

La funzione di costo è:

$$J_t = J_m(\bar{x}_t, \bar{u}_{t\dots t+N-1}) + J_v(X_t) \quad (\text{A.2})$$

dove si ricorda che:

$$J_m = \sum_{k=t}^{t+N-1} (\|\bar{x}_k - \tilde{x}_t\|_Q^2 + \|\bar{u}_k - \tilde{u}_t\|_R^2) + \|\bar{x}_{t+N} - \tilde{x}_t\|_P^2 + \|\tilde{z}_t - \tilde{z}_G\|_T \quad (\text{A.3})$$

$$J_v = \sum_{k=t}^{k+N-1} tr \left\{ \left(Q + (\bar{K})^T R \bar{K} \right) X_t \right\} + tr \{ P X_{t+N} \} \quad (\text{A.4})$$

Si ricorda che verrà rielaborato solamente il termine J_m e non J_v , dato che quest'ultimo può...dipende univocamente dalla strategia di inizializzazione adottata, dove, si ricorda, esistono solamente due possibili valori possibili. Si è scelto infatti di non ottimizzare rispetto alla sequenza di guadagni K_k , e di porre $K_{k|t} = \bar{K}$, $\forall k$. D'altra parte, J_m dipende dai valori futuri dello stato nominale, i quali vengono predetti mediante l'equazione dinamica (2.22). Il problema viene quindi riformulato in funzione di quantità note e quantità dipendenti dal vettore di ottimizzazione. Si noti, inoltre, che il peso sul goal T è nel caso specifico uno scalare. In particolare sarà indicato con γ e posto uguale a 10^5 . Si vuole calcolare la predizione dello stato a partire dall'istante t . Si consideri l'equazione per lo stato nominale:

$$\bar{x}_{t+1} = A\bar{x}_t + B\bar{u}_t \quad (\text{A.5})$$

Iterando tale equazione, si calcola

$$\begin{aligned}
 \bar{x}_t &= I\bar{x}_t \\
 \bar{x}_{t+1} &= A\bar{x}_t + B\bar{u}_t \\
 \bar{x}_{t+2} &= A^2\bar{x}_t + AB\bar{u}_t + B\bar{u}_{t+1} \\
 &\vdots \\
 \bar{x}_{t+N} &= A^N\bar{x}_t + A^{N-1}B\bar{u}_t + \dots + B\bar{u}_{t+N-1}
 \end{aligned}$$

che riscritta in forma matriciale è:

$$\underbrace{\begin{bmatrix} \bar{x}_t \\ \bar{x}_{t+1} \\ \bar{x}_{t+2} \\ \vdots \\ \bar{x}_{t+N} \end{bmatrix}}_{\tilde{\mathcal{X}}_t} = \underbrace{\begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{N+1} \end{bmatrix}}_{\mathcal{A}} \bar{x}_t + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}}_{\mathcal{B}} \underbrace{\begin{bmatrix} \bar{u}_t \\ \bar{u}_{t+1} \\ \bar{u}_{t+2} \\ \vdots \\ \bar{u}_{N+1} \end{bmatrix}}_{\tilde{\mathcal{U}}_t} \quad (\text{A.6})$$

A questo punto è possibile riscrivere la funzione come segue:

$$\begin{aligned}
 J_m &= \|\tilde{\mathcal{X}}_t - \tilde{\mathcal{X}}_t\|_{\mathcal{Q}}^2 + \|\tilde{\mathcal{U}}_t - \tilde{\mathcal{U}}_t\|_{\mathcal{R}}^2 + \gamma\|\tilde{z}_t - z_G\|^2 = \\
 &= \|\mathcal{A}\bar{x}_t + \mathcal{B}\bar{\mathcal{U}}_t - \tilde{\mathcal{X}}_t\|_{\mathcal{Q}}^2 + \|\bar{\mathcal{U}}_t - \tilde{\mathcal{U}}_t\|_{\mathcal{R}}^2 + \gamma\|\tilde{z}_t - z_G\|^2
 \end{aligned}$$

dove

$$\tilde{\mathcal{X}}_t = \begin{bmatrix} \tilde{x}_t \\ \vdots \\ \tilde{x}_t \end{bmatrix}, \quad \tilde{\mathcal{U}}_t = \begin{bmatrix} \tilde{u}_t \\ \vdots \\ \tilde{u}_t \end{bmatrix}$$

Le matrici di peso sono diagonali

$$\mathcal{Q} = \begin{bmatrix} Q & & & \\ & \ddots & & \\ & & Q & \\ & & & P \end{bmatrix}, \quad \mathcal{R} = \begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix}$$

Si definisce quindi il vettore di ottimizzazione:

$$\Xi = \begin{bmatrix} \bar{\mathcal{U}}_t \\ \tilde{z}_t \end{bmatrix} \quad (\text{A.7})$$

e si ricorda che:

$$\begin{bmatrix} \tilde{x}_t \\ \tilde{u}_t \end{bmatrix} = M \tilde{z}_t = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \tilde{z}_t \quad (\text{A.8})$$

e quindi:

$$\begin{aligned} J_m &= \|\mathcal{A}\bar{x}_t + \mathcal{B}[I \ 0]\Xi - M_1[0 \ I]\Xi\|_{\mathcal{Q}}^2 + \|[I \ 0]\Xi - M_2[0 \ I]\Xi\|_{\mathcal{R}}^2 + \\ &\quad + \gamma\|[0 \ I]\Xi - z_G\|^2 \\ &= \|\mathcal{A}\bar{x}_t + \underbrace{[\mathcal{B} - M_1]}_{\tilde{\mathcal{B}}}\Xi\|_{\mathcal{Q}}^2 + \|\underbrace{[I \ M_2]}_{\tilde{\mathcal{C}}}\Xi\|_{\mathcal{R}}^2 + \gamma\|\underbrace{[0 \ I]}_{\tilde{\mathcal{D}}}\Xi - z_G\|^2 \\ &= \|\mathcal{A}\bar{x}_t + \tilde{\mathcal{B}}\Xi\|_{\mathcal{Q}}^2 + \|\tilde{\mathcal{C}}\Xi\|_{\mathcal{R}}^2 + \gamma\|\tilde{\mathcal{D}}\Xi - z_G\|^2 \\ &= \bar{x}_t^T \mathcal{A}^T \mathcal{Q} \mathcal{A} \bar{x}_t + \Xi^T \tilde{\mathcal{B}}^T \mathcal{Q} \tilde{\mathcal{B}} \Xi + \Xi^T \tilde{\mathcal{C}}^T \mathcal{R} \tilde{\mathcal{C}} \Xi + \gamma \Xi^T \tilde{\mathcal{D}}^T \tilde{\mathcal{D}} \Xi + \gamma z_G^T z_G + \\ &\quad - 2(\bar{x}_t^T \mathcal{A}^T \mathcal{Q} \tilde{\mathcal{B}} \Xi) - 2(z_G^T \tilde{\mathcal{D}} \Xi) \end{aligned}$$

A questo punto è possibile separare i termini dipendenti dal vettore di ottimizzazione Ξ dai termini noti. In particolare:

$$\begin{aligned} J_m &= \Xi^T \left(\tilde{\mathcal{B}}^T \mathcal{Q} \tilde{\mathcal{B}} + \tilde{\mathcal{C}}^T \mathcal{R} \tilde{\mathcal{C}} + \gamma \tilde{\mathcal{D}}^T \tilde{\mathcal{D}} \right) \Xi - 2 \left(\bar{x}_t^T \mathcal{A}^T \mathcal{Q} \tilde{\mathcal{B}} + \gamma z_G^T \tilde{\mathcal{D}} \right) \Xi + \\ &\quad + \bar{x}_t^T \mathcal{A}^T \mathcal{Q} \mathcal{A} \bar{x}_t + \gamma z_G^T z_G \end{aligned}$$

A questo punto è possibile definire le matrici H e f che compaiono in (A.1) come segue

$$\begin{cases} H = 2 \left(\tilde{\mathcal{B}}^T \mathcal{Q} \tilde{\mathcal{B}} + \tilde{\mathcal{C}}^T \mathcal{R} \tilde{\mathcal{C}} + \gamma \tilde{\mathcal{D}}^T \tilde{\mathcal{D}} \right) \\ f^T = 2 \left(\bar{x}_t^T \mathcal{A}^T \mathcal{Q} \tilde{\mathcal{B}} - \gamma z_G^T \tilde{\mathcal{D}} \right) \end{cases} \quad (\text{A.9})$$

Si noti che gli ultimi due termini del funzionale di costo, essendo noti, non dipendono dal vettore di ottimizzazione Ξ e quindi non influiscono sul valore ottimo di J_m , possono quindi essere ignorati.

Vincoli

I vincoli da inserire nel problema di ottimizzazione devono essere espressi rispetto al vettore di ottimizzazione Ξ . In particolare ci si propone di ricavare le matrici A_{ineq} e b_{ineq} necessarie al solutore come definito in (A.1). In particolare si consideri il vincolo sullo stato, relativo all' i -esimo robot

$$(b_r)^T \bar{x}_{t+k} \leq 1 - \sqrt{(b_r)^T X_{t+k} b_r} f(p_r) \quad \text{con } k = 0, \dots, N-1 \quad (\text{A.10})$$

Tale vincolo deve essere rispettato lungo tutto l'orizzonte di predizione. Riscrivendolo in termini matriciali si ha:

$$\underbrace{\begin{bmatrix} (b_r)^T & & \\ & \ddots & \\ & & (b_r)^T \end{bmatrix}}_{\bar{B}_r} \underbrace{\begin{bmatrix} \bar{x}_t \\ \vdots \\ \bar{x}_{t+N} \end{bmatrix}}_{\bar{\mathcal{X}}_t} \leq \underbrace{\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}}_{\mathcal{H}_1} - \underbrace{\begin{bmatrix} \sqrt{(b_r)^T X_t b_r} f(p_r) \\ \vdots \\ \sqrt{(b_r)^T X_{t+N} b_r} f(p_r) \end{bmatrix}}_{\mathcal{H}_1}$$

Si riscrive la disuguaglianza rispetto al vettore Ξ :

$$\begin{aligned} \bar{B}_r (\mathcal{A} \bar{x}_t + \mathcal{B} \bar{u}_t) \leq \mathcal{H}_1 &\Rightarrow \bar{B}_r (\mathcal{A} \bar{x}_t + \mathcal{B} [I \ 0] \Xi) \leq \mathcal{H}_1 \\ &\Rightarrow \bar{B}_r [\mathcal{B} \ 0] \Xi \leq \mathcal{H}_1 - \bar{B}_r \mathcal{A} \bar{x}_t \end{aligned}$$

che può essere riscritta come:

$$A_{ineq1} \Xi \leq b_{ineq1} \quad \text{con} \quad \begin{cases} A_{ineq1} = \bar{B}_r [\mathcal{B} \ 0] \\ b_{ineq1} = \mathcal{H}_1 - \bar{B}_r \mathcal{A} \bar{x}_t \end{cases}$$

Si consideri ora il vincolo sulla variabile di ingresso

$$(c_s)^T \bar{u}_{t+k} \leq 1 - \sqrt{(c_s)^T U_{t+k} c_s} f(p_s) \quad \text{con } k = 0, \dots, N-1 \quad (\text{A.11})$$

Allo stesso modo di quanto fatto sopra viene rielaborato come segue:

$$\underbrace{\begin{bmatrix} (c_s)^T & & \\ & \ddots & \\ & & (c_s)^T \end{bmatrix}}_{\bar{C}_s} \underbrace{\begin{bmatrix} \bar{u}_t \\ \vdots \\ \bar{u}_{t+N} \end{bmatrix}}_{\bar{U}_t} \leq \underbrace{\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}}_{\mathcal{H}_2} - \underbrace{\begin{bmatrix} \sqrt{(c_s)^T U_t c_s} f(p_s^u) \\ \vdots \\ \sqrt{(c_s)^T U_{t+N} c_s} f(p_s^u) \end{bmatrix}}_{\mathcal{H}_2}$$

che riscritta è:

$$\bar{C}_s \bar{U}_t \leq \mathcal{H}_2 \Rightarrow \bar{C}_s [I \ 0] \Xi \leq \mathcal{H}_2$$

quindi:

$$A_{ineq2} \Xi \leq b_{ineq2} \quad \text{con} \quad \begin{cases} A_{ineq2} = \bar{C}_s [I \ 0] \\ b_{ineq2} = \mathcal{H}_2 \end{cases}$$

Vincolo di Obstacle Avoidance

Si consideri il vincolo (5.9). Ad ogni istante di tempo si calcola il vettore $\delta_{hi,l}^o$:

$$\delta_{hi,l}^o(k|t-1) = \left(h_l^{[o,hi]} \right)^T \left(\bar{z}_{k|t-1}^{[i]} - z_h^o \right) - d_{hi}^o - \sqrt{\left(h_l^{[o,hi]} \right)^T Z_{k|t-1}^{[i]} h_l^{[o,hi]} f(p_{hi}^o)}$$

A questo punto per ogni vettore $\delta_{hi,l}^o$ si calcola il vincolo l -esimo maggiormente violato:

$$\bar{l}_{hi}^o(k|t-1) = \underset{l \in \{1, \dots, r_{hi}^o\}}{\operatorname{argmax}} \delta_{hi,l}^o(k|t-1)$$

Il vincolo da inserire al tempo t è

$$\left(h_{\bar{l}_{hi}^o}^{[o,hi]} \right)^T \left(\bar{z}_t^{[i]} - \bar{z}_{k|t-1}^{[i]} \right) \geq -\delta_{hi, \bar{l}_{hi}^o}^o(t|t-1)(k|t-1) \quad (\text{A.12})$$

Si riscriva il vincolo in termini matriciali

$$\underbrace{\begin{bmatrix} \left(h_{\bar{l}_o^{[o,hi]}(t|t-1)}\right)^T & & & \\ & \ddots & & \\ & & \left(h_{\bar{l}_o^{[o,hi]}(t+N-1|t-1)}\right)^T & \end{bmatrix}}_{H_{\bar{l}_o^{[o,hi]}(k|t-1)}^o} \underbrace{\begin{bmatrix} \bar{z}_t - \bar{z}_{t|t-1} \\ \vdots \\ \bar{z}_{t+N-1} - \bar{z}_{t+N-1|t-1} \end{bmatrix}}_{\bar{Z}_t - \bar{Z}_{t|t-1}} \geq$$

$$\underbrace{\begin{bmatrix} -\delta_{\bar{h}_i, \bar{l}_o^{[o,hi]}(t|t-1)}^o(t|t-1) \\ \vdots \\ -\delta_{\bar{h}_i, \bar{l}_o^{[o,hi]}(t+N-1|t-1)}^o(t+N-1|t-1) \end{bmatrix}}_{\Delta_{\bar{l}_o^{[o,hi]}(t-1)}^o}$$

che in modo più compatto equivale a:

$$H_{\bar{l}_o^{[o,hi]}(k|t-1)}^o (\bar{Z}_t - \bar{Z}_{t|t-1}) \geq \Delta_{\bar{l}_o^{[o,hi]}(t-1)}^o(t-1)$$

Si pone

$$\mathcal{C} = \begin{bmatrix} C & & \\ & \ddots & \\ & & C \end{bmatrix}$$

e risulta

$$H_{\bar{l}_o^{[o,hi]}(k|t-1)}^o \mathcal{C} (\bar{\mathcal{X}}_t - \bar{\mathcal{X}}_{t|t-1}) \geq \Delta_{\bar{l}_o^{[o,hi]}(t-1)}^o(t-1) \Rightarrow$$

$$\Rightarrow H_{\bar{l}_o^{[o,hi]}(k|t-1)}^o \mathcal{C} (\mathcal{A}\bar{x}_t + \mathcal{B} [I \ 0] \Xi - \bar{\mathcal{X}}_{t|t-1}) \geq \Delta_{\bar{l}_o^{[o,hi]}(t-1)}^o(t-1)$$

e si separino i termini dipendenti da Ξ da quelli non dipendenti,

$$-H_{\bar{l}_o^{[o,hi]}(k|t-1)}^o \mathcal{C} \mathcal{B} [I \ 0] \Xi \leq H_{\bar{l}_o^{[o,hi]}(k|t-1)}^o \mathcal{C} (\mathcal{A}\bar{x}_t - \bar{\mathcal{X}}_{t|t-1}) - \Delta_{\bar{l}_o^{[o,hi]}(t-1)}^o(t-1) \quad (\text{A.13})$$

quindi:

$$A_{ineq3} \Xi \leq b_{ineq3} \quad \text{con} \quad \begin{cases} A_{ineq3} = -H_{l^o(k|t-1)}^o \mathcal{CB} [I \quad 0] \\ b_{ineq3} = H_{l^o(k|t-1)}^o \mathcal{C}(\mathcal{A}\bar{x}_t - \bar{\mathcal{X}}_{t|t-1}) - \Delta_{l^o(t-1)}^o(t-1) \end{cases}$$

Vincolo di Collision Avoidance

Il vincolo da includere è

$$\left(h_t^{[ji]}\right)^T \left(\bar{z}_t^{[i]} - \bar{z}_t^{[j]}\right) \geq d_{ij} + \sqrt{\left(h_t^{[hi]}\right)^T \left(Z_t^{[i]} - Z_t^{[j]}\right) h_l^{[ij]} f(p_{ij})}$$

Si definisce per ogni $k = t, \dots, t + N - 1$

$$\delta_{ij,l}(k|t-1) = \left(h_t^{[ji]}\right)^T \left(\bar{z}_t^{[i]} - \bar{z}_t^{[j]}\right) - d_{ij} - \sqrt{\left(h_t^{[hi]}\right)^T \left(Z_t^{[i]} - Z_t^{[j]}\right) h_l^{[ij]} f(p_{ij})}$$

e l'indice

$$\bar{l}_{ij}(k|t-1) = \underset{l \in \{1, \dots, r_{ij}\}}{\operatorname{argmax}} \delta_{ij,l}(k|t-1)$$

A questo punto i vincoli da garantire lungo tutto l'orizzonte sono:

- robot i

$$\left(h_{\bar{l}_{ij}(k|t-1)}\right)^T \left(\bar{z}_k^{[i]} - \bar{z}_{k|t-1}^{[i]}\right) \geq \frac{1}{2} \delta_{ij, \bar{l}_{ij}(k|t-1)}(k|t-1) \quad (\text{A.14})$$

che in termini matriciali è

$$\underbrace{\begin{bmatrix} \left(h_{\bar{l}_{ij}(t|t-1)}\right)^T \\ \vdots \\ \left(h_{\bar{l}_{ij}(t+N-1|t-1)}\right)^T \end{bmatrix}}_{H_{\bar{l}(k|t-1)}} \underbrace{\begin{bmatrix} \bar{z}_t^{[i]} - \bar{z}_{t|t-1}^{[i]} \\ \vdots \\ \bar{z}_{t+N-1}^{[i]} - \bar{z}_{t+N-1|t-1}^{[i]} \end{bmatrix}}_{\bar{z}_t^{[i]} - \bar{z}_{t|t-1}^{[i]}} \geq$$

$$\underbrace{\begin{bmatrix} -\frac{1}{2}\delta_{ij,\bar{l}_{ij}}(t|t-1)(t|t-1) \\ \vdots \\ -\frac{1}{2}\delta_{ij,\bar{l}_{ij}}(t+N-1|t-1)(t+N-1|t-1) \end{bmatrix}}_{\Delta_{\bar{l}(k|t-1)}}$$

Si esprime il vincolo rispetto al vettore Ξ :

$$\begin{aligned} H_{\bar{l}(k|t-1)} \left(\bar{\mathcal{Z}}_t^{[i]} - \bar{\mathcal{Z}}_{t|t-1}^{[i]} \right) &\geq \Delta_{\bar{l}(k|t-1)} \Rightarrow \\ H_{\bar{l}(k|t-1)} \mathcal{C} \left(\bar{\mathcal{X}}_t^{[i]} - \bar{\mathcal{X}}_{t|t-1}^{[i]} \right) &\geq \Delta_{\bar{l}(k|t-1)} \Rightarrow \\ H_{\bar{l}(k|t-1)} \mathcal{C} \left(\mathcal{A} \bar{x}_t^{[i]} - \mathcal{B} [I \quad 0] \Xi^{[i]} - \bar{\mathcal{X}}_{t|t-1}^{[i]} \right) &\geq \Delta_{\bar{l}(k|t-1)} \end{aligned}$$

- robot j

$$\left(h_{\bar{l}_{ij}(k|t-1)} \right)^T \left(\bar{z}_{k|t-1}^{[j]} - \bar{z}_k^{[j]} \right) \geq \frac{1}{2} \delta_{ij,\bar{l}_{ij}}(k|t-1)(k|t-1) \quad (\text{A.15})$$

che in termini matriciali è

$$\underbrace{\begin{bmatrix} \left(h_{\bar{l}_{ij}(t|t-1)} \right)^T \\ \vdots \\ \left(h_{\bar{l}_{ij}(t+N-1|t-1)} \right)^T \end{bmatrix}}_{H_{\bar{l}(k|t-1)}} \underbrace{\begin{bmatrix} \bar{z}_{t|t-1}^{[j]} - \bar{z}_t^{[j]} \\ \vdots \\ \bar{z}_{t+N-1}^{[i]} - \bar{z}_{t+N-1|t-1}^{[i]} \end{bmatrix}}_{\bar{\mathcal{Z}}_{t|t-1}^{[i]} - \bar{\mathcal{Z}}_t^{[j]}} \geq$$

$$\underbrace{\begin{bmatrix} -\frac{1}{2}\delta_{ij,\bar{l}_{ij}}(t|t-1)(t|t-1) \\ \vdots \\ -\frac{1}{2}\delta_{ij,\bar{l}_{ij}}(t+N-1|t-1)(t+N-1|t-1) \end{bmatrix}}_{\Delta_{\bar{l}(k|t-1)}}$$

Come si può notare i vincoli per i due robot sono simmetrici.

A questo punto si esprimono i vincoli rispetto al vettore Ξ

$$\begin{aligned} H_{\bar{l}(k|t-1)} \left(\bar{\mathcal{Z}}_t^{[j]} - \bar{\mathcal{Z}}_{t|t-1}^{[j]} \right) &\geq \Delta_{\bar{l}(k|t-1)} \Rightarrow \\ H_{\bar{l}(k|t-1)} \mathcal{C} \left(\bar{\mathcal{X}}_t^{[j]} - \bar{\mathcal{X}}_{t|t-1}^{[j]} \right) &\geq \Delta_{\bar{l}(k|t-1)} \Rightarrow \\ H_{\bar{l}(k|t-1)} \mathcal{C} \left(\mathcal{A}\bar{x}_t^{[j]} - \mathcal{B} \begin{bmatrix} I & 0 \end{bmatrix} \Xi^{[j]} - \bar{\mathcal{X}}_{t|t-1}^{[j]} \right) &\geq \Delta_{\bar{l}(k|t-1)} \end{aligned}$$

quindi il vincolo per il robot i è:

$$A_{ineq4} \Xi^{[i]} \leq b_{ineq4}^{[i]} \quad \text{con} \begin{cases} A_{ineq4} = H_{\bar{l}(k|t-1)} \mathcal{C} \mathcal{B} \begin{bmatrix} I & 0 \end{bmatrix} \\ b_{ineq4}^{[i]} = H_{\bar{l}(k|t-1)} \mathcal{C} \left(\mathcal{A}\bar{x}_t^{[i]} - \bar{\mathcal{X}}_{t|t-1}^{[i]} \right) - \Delta_{\bar{l}(k|t-1)} \end{cases}$$

mentre quello per il robot j è:

$$A_{ineq4} \Xi^{[j]} \leq b_{ineq4}^{[j]} \quad \text{con} \begin{cases} A_{ineq4} = H_{\bar{l}(k|t-1)} \mathcal{C} \mathcal{B} \begin{bmatrix} I & 0 \end{bmatrix} \\ b_{ineq4}^{[j]} = H_{\bar{l}(k|t-1)} \mathcal{C} \left(\mathcal{A}\bar{x}_t^{[j]} - \bar{\mathcal{X}}_{t|t-1}^{[j]} \right) - \Delta_{\bar{l}(k|t-1)} \end{cases}$$

Vincolo Terminale

Ad ogni passo di ottimizzazione è necessario garantire che lo stato raggiunga il suo *target* e in modo congiunto, risulti in una posizione dello spazio di lavoro che sia priva di collisioni. Il vincolo terminale si traduce quindi in vincoli di uguaglianza e di disequaglianza, impostati secondo una logica di tipo *zero terminal constraint*.

In particolare il vincolo di uguaglianza da rispettare è:

$$\bar{x}_{t+N} = M_1 \tilde{z}_t \tag{A.16}$$

A questo punto si esprime il vincolo (A.16) rispetto al vettore di ottimizzazione

$$\begin{aligned} \bar{x}_{t+N} = M_1 \tilde{z}_t &\Rightarrow \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} \bar{\mathcal{X}}_t = M_1 \tilde{D} \Xi \Rightarrow \\ &\Rightarrow \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} \left(\mathcal{A}\bar{x}_t + \mathcal{B} \begin{bmatrix} I & 0 \end{bmatrix} \Xi \right) = M_1 \tilde{D} \Xi \Rightarrow \end{aligned}$$

quindi:

$$A_{eq}^{[i]} \Xi = b_{eq}^{[i]} \quad \text{con} \quad \begin{cases} A_{eq}^{[i]} = \left(\begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} \mathcal{B} \begin{bmatrix} I & 0 \end{bmatrix} - M_1 \tilde{D} \right) \\ b_{eq}^{[i]} = - \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} \mathcal{A} \bar{x}_t \end{cases}$$

I vincoli di disuguaglianza che garantiscono che il robot sia in regioni *safe* sono definiti in termini di *obstacle avoidance* e di *collision avoidance*.

Per quanto riguarda l'*obstacle avoidance* tra il robot i -esimo e l'ostacolo h -esimo, si consideri la funzione:

$$\tilde{\delta}_{hi,l}^o(t-1) = (h_l^{[o,hi]})^T (\tilde{z}_{t-1|t-1}^{[i]} - z_h^o) - d_{hi}^o - \sqrt{(h_l^{[o,hi]})^T \bar{Z}^{[i]} h_l^{[o,hi]}} f(p_{hi}^o)$$

e l'indice corrispondente al vincolo maggiormente "violato"

$$\tilde{l}_{hi}^o(t-1) = \underset{l \in \{1, \dots, r_{hi}^o\}}{\text{argmax}} \tilde{\delta}_{hi,l}^o(t-1)$$

A questo punto il vincolo da inserire è:

$$(h_{\tilde{l}_{hi}(t-1)}^{[o,hi]})^T (\tilde{z}_t^{[i]} - \tilde{z}_{t-1|t-1}^{[i]}) \geq -\tilde{\delta}_{hi, \tilde{l}_{hi}(t-1)}^o(t-1) \quad (\text{A.17})$$

Esprimendolo rispetto Ξ si ottiene:

$$(h_{\tilde{l}_{hi}(t-1)}^{[o,hi]})^T (\tilde{D} \Xi - \tilde{z}_{t-1|t-1}^{[i]}) \geq -\tilde{\delta}_{hi, \tilde{l}_{hi}(t-1)}^o(t-1)$$

quindi:

$$A_{ineq5} \Xi \leq b_{ineq5} \quad \text{con} \quad \begin{cases} A_{ineq5} = -(h_{\tilde{l}_{hi}(t-1)}^{[o,hi]})^T \tilde{D} \\ b_{ineq5} = \tilde{\delta}_{hi, \tilde{l}_{hi}(t-1)}^o(t-1) - (h_{\tilde{l}_{hi}(t-1)}^{[o,hi]})^T \tilde{z}_{t-1|t-1}^{[i]} \end{cases}$$

Per quanto riguarda la *collision avoidance* si richiede di calcolare

$$\tilde{\delta}_{ij,l}^o(t-1) = (h_l^{[ij]})^T (\tilde{z}_{t-1|t-1}^{[i]} - \tilde{z}_{t-1|t-1}^{[j]}) - d_{ij} - \sqrt{(h_l^{[ij]})^T \bar{Z}^{[i]} h_l^{[ij]}} f(p_{ij})$$

essendo noti al tempo t i valori di $\tilde{z}_{t-1|t-1}^{[i]}$ e $\tilde{z}_{t-1|t-1}^{[j]}$. Si calcola quindi l'indice corrispondente al vincolo più violato

$$\tilde{l}_{ij}(t-1) = \operatorname{argmax}_{l \in \{1, \dots, r_{hi}^o\}} \tilde{\delta}_{ij,l}(t-1)$$

A questo punto si consideri il vincolo che il robot i deve garantire:

$$(h_{\tilde{l}_{ij}(t-1)}^{[ij]})^T (\tilde{z}_t^{[i]} - \tilde{z}_{t-1|t-1}^{[i]}) \geq -\frac{1}{2} \tilde{\delta}_{ij, \tilde{l}_{ij}(t-1)}(t-1) \quad (\text{A.18})$$

e lo si riscriva rispetto al vettore di ottimizzazione Ξ

$$(h_{\tilde{l}_{ij}(t-1)}^{[ij]})^T (\tilde{D} \Xi - \tilde{z}_{t-1|t-1}^{[i]}) \geq -\frac{1}{2} \tilde{\delta}_{ij, \tilde{l}_{ij}(t-1)}(t-1)$$

quindi:

$$A_{ineq6} \Xi \leq b_{ineq6}^{[i]} \quad \text{con} \begin{cases} A_{ineq6} = -(h_{\tilde{l}_{ij}(t-1)}^{[ij]})^T \tilde{D} \\ b_{ineq6}^{[i]} = \frac{1}{2} \tilde{\delta}_{ij, \tilde{l}_{ij}(t-1)}(t-1) - (h_{\tilde{l}_{ij}(t-1)}^{[ij]})^T \tilde{z}_{t-1|t-1}^{[i]} \end{cases}$$

Per quanto riguarda il robot j il vincolo è

$$(h_{\tilde{l}_{ij}(t-1)}^{[ij]})^T (\tilde{z}_{t-1|t-1}^{[j]} - \tilde{z}_t^{[j]}) \geq -\frac{1}{2} \tilde{\delta}_{ij, \tilde{l}_{ij}(t-1)}(t-1) \quad (\text{A.19})$$

che espresso rispetto Ξ è:

$$(h_{\tilde{l}_{ij}(t-1)}^{[ij]})^T (\tilde{z}_{t-1|t-1}^{[j]} - \tilde{D} \Xi) \geq -\frac{1}{2} \tilde{\delta}_{ij, \tilde{l}_{ij}(t-1)}(t-1)$$

quindi:

$$A_{ineq6} \Xi \leq b_{ineq6}^{[j]} \quad \text{con} \begin{cases} A_{ineq6} = (h_{\tilde{l}_{ij}(t-1)}^{[ij]})^T \tilde{D} \\ b_{ineq6}^{[j]} = \frac{1}{2} \tilde{\delta}_{ij, \tilde{l}_{ij}(t-1)}(t-1) + (h_{\tilde{l}_{ij}(t-1)}^{[ij]})^T \tilde{z}_{t-1|t-1}^{[j]} \end{cases}$$

A questo punto per ogni $i = 1, \dots, M$ le matrici A_{ineq} e b_{ineq} si calcolano come:

$$A_{ineq} = \begin{bmatrix} A_{ineq1} \\ A_{ineq2} \\ A_{ineq3} \\ A_{ineq4} \\ A_{ineq5} \\ A_{ineq6} \end{bmatrix}, \quad b_{ineq} = \begin{bmatrix} b_{ineq1} \\ b_{ineq2} \\ b_{ineq3} \\ b_{ineq4} \\ b_{ineq5} \\ b_{ineq6} \end{bmatrix}$$

Bibliografia

- [1] L. Magni, R. Scattolini. *Complementi di Controlli Automatici*, Pitagora Editrice Bologna, 2006.
- [2] D. Limon, I. Alvarado, T. Alamo, E. F. Camacho. MPC for tracking piecewise constant references for constrained linear system, *Automatica*, Vol. 44, 2008, pp. 2382-2387.
- [3] M. Farina, L. Giulioni, R. Scattolini. A Probabilistic Approach to Model Predictive Control, *52nd IEEE Conference on Decision and Control*, 2013, pp. 7734-7739.
- [4] G. C. Calafiore, L. Fagiano. Robust Model Predictive Control via Scenario Optimization, *IEEE Transactions on Automatic Control*, Vol. 58, no. (1), 2013, pp. 219-224.
- [5] A. Perizzato. *Navigazione e Coordinamento di agenti mobili mediante tecniche di controllo predittivo distribuito*. Tesi di laurea mag. Politecnico di Milano, 2013.
- [6] M. Vergani. *Controllo predittivo distribuito per la navigazione e il sensing di sistemi multiagente*. Tesi di laurea mag. Politecnico di Milano, 2014.
- [7] E.G. Gilbert, K. T. Tan, Linear systems with state and control constraints: the theory and application of maximal output admissible sets, *IEEE Transactions on Automatic Control*, Vol. 36, no. (9), 1991, pp. 1008-1020.

- [8] F. Oldewurtel, A. Parisio, C. N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, K. Wirth. Energy Efficient Building Climate Control using Stochastic Model Predictive Control and Weather Predictions, *2010 American Control Conference*, 2010, pp. 5100-5105.
- [9] Y. Ma, S. Vichik, and F. Borrelli. Fast Stochastic MPC with Optimal Risk Allocation Applied to Building Control Systems, *51st IEEE Conference on Decision and Control*, 2012, pp. 7559-7564.
- [10] F. Oldewurtel, C. N. Jones, A. Parisio and M. Morari. Stochastic Model Predictive Control for Building Climate Control, *IEEE Transaction on Control Systems Technology*, Vol. 22 (3), 2014, pp. 1998-1205.
- [11] S. Di Cairano, D. Bernardini, A. Bemporad and I. V. Kolmanovskiy. Stochastic MPC With Learning for Driver-Predictive Vehicle Control and its Application to HEV Energy Management, *IEEE Transactions on Control Systems Technology*, Vol. 22 (3), 2014, pp. 1118-1031.
- [12] P. Trodden, A. Richards. Distributed model predictive control of linear systems with persistent disturbances, *International Journal of Control*, Vol. 83, No. 8, 2010, 1653–1663.
- [13] H. Fukushima, K. Kon and F. Matsuno, Distributed Model Predictive Control for Multi-Vehicle Formation with Collision Avoidance Constraints, *44th IEEE Conference on Decision and Control, and the European Control Conference*, 2005, pp. 5480-5485.
- [14] A. Parisio e L. Glielmo, Stochastic Model Predictive Control for Economic/Environmental Operation Management of Microgrids, *European Control Conference*, 2013, pp. 2014-2019.
- [15] S.V. Rakovic e D.Q. Mayne. Set Robust Control Invariance for Linear Discrete Time Systems, *44th IEEE Conference on Decision and Control, and the European Control Conference*, 2005, pp. 975-980.

-
- [16] J.M. Maestre, R.R. Negenborn *Distributed Model Predictive Control Made Easy*, Springer, 2014.
- [17] B. Picasso, D. Desiderio and R. Scattolini. Robust Stability Analysis of Nonlinear Discrete-Time Systems With Application to MPC, *IEEE Transactions on Automatic Control*, Vol. 57, No. 1, 2010, pp. 185-191.
- [18] L. Magni, G. De Nicolao, R. Scattolini and F. Allgöwer. Robust model predictive control for nonlinear discrete-time systems, *International Journal of Robust and Nonlinear Control*, Vol. 13, No. 3-4, 2003, pp. 229-246.
- [19] D.Q. Mayne, M.M. Seron, S.V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances, *Automatica*, Vol. 41, No. 2, 2005, pp. 219-224.