



SAPIENZA  
UNIVERSITÀ DI ROMA

## Controllo di formazione di robot mobili

Facoltà di Ingegneria dell'informazione, Informatica e Statistica  
Corso di Laurea in Ingegneria Informatica e automatica

Candidato  
Sofia Santilli  
Matricola 1813509

Relatore  
Prof. Giuseppe Oriolo

Anno Accademico 2019/2020

Tesi discussa il 3 Dicembre 2020  
di fronte a una commissione esaminatrice composta da:  
Prof. Stefano Battilotti (presidente)  
Prof. Claudia Califano

---

**Controllo di formazione di robot mobili**  
Tesi di Laurea. Sapienza – Università di Roma

© 2020 Sofia Santilli. Tutti i diritti riservati

Questa tesi è stata composta con L<sup>A</sup>T<sub>E</sub>X e la classe Sapthesis.

Versione: 7 febbraio 2022

Email dell'autore: santilli.1813509@studenti.uniroma1.it

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Uniciclo</b>	<b>3</b>
2.1	Cos'è un robot mobile? . . . . .	3
2.2	Cos'è un uniciclo? . . . . .	4
2.3	Modello cinematico di un uniciclo . . . . .	4
<b>3</b>	<b>Controllo di formazione</b>	<b>6</b>
3.1	Sistemi multiagente . . . . .	6
3.2	Modello cinematico di una formazione di unicicli . . . . .	7
3.3	Legge di controllo di formazione . . . . .	8
3.3.1	Robustezza della legge di controllo rispetto alle singolarità . . . . .	9
<b>4</b>	<b>Simulazioni</b>	<b>12</b>
4.1	Centroide . . . . .	12
4.1.1	Singolarità nel caso del centroide . . . . .	17
4.2	Varianza . . . . .	19
4.2.1	Singolarità nel caso della varianza . . . . .	24
4.3	Centroide e Varianza . . . . .	26
<b>5</b>	<b>Conclusioni</b>	<b>33</b>

# Capitolo 1

## Introduzione

Sono sempre maggiori i contesti di applicazione della robotica nella vita di tutti i giorni: dai casi più semplici come i vacuum cleaner che puliscono le nostre case, alle applicazioni in campo medico, militare, all'esplorazione di nuovi pianeti. A volte in queste applicazioni, si trovano più robot che devono lavorare in uno stesso ambiente, collaborando o competendo tra loro: è questo il caso di una formazione di robot, che necessita di una legge di controllo che regoli il comportamento di ogni singolo robot.

La seguente trattazione si propone di analizzare la legge di controllo di una formazione di robot mobili. In questo contesto la formazione è intesa come un sistema multiagente caratterizzato da agenti cooperativi: ogni robot si muove ed orienta dando il suo contributo alla traiettoria finale desiderata. Quindi nel seguito i concetti di 'robot' o di 'agente' saranno usati in modo interscambiabile.

Si sottolinea come usare un sistema multi robot, per svolgere un certo compito, porti notevoli vantaggi rispetto all'uso di un singolo robot. Infatti il sistema ha una maggiore robustezza rispetto ai guasti: se un robot della formazione presenta un problema, questo può essere risolto mentre gli altri agenti continuano a svolgere il proprio lavoro. È inoltre possibile progettare agenti specializzati (più semplici ed economici rispetto ad un singolo agente complesso) in grado di risolvere sotto-problemi in parallelo.

In questo lavoro è stata considerata una classe particolare di robot mobili: gli unicicli. Un uniciclo in realtà non è una vera e propria entità fisica, ma solo un'astrazione matematica basata sul vincolo di puro rotolamento (vincolo che impone alla ruota di rotolare sul piano, senza traslare). Tuttavia, in questo elaborato, ci riferiremo agli unicicli semplicemente con il termine di 'robot', assumendoli come veicoli con una singola ruota orientabile.

Il nostro scopo è quello di controllare due entità della formazione, affinché seguano delle traiettorie desiderate. Le due entità in questione sono:

- il centroide, dato dal valore medio delle posizioni degli unicicli;
- la varianza, data dalla deviazione quadratica delle posizioni degli unicicli rispetto al centroide. Questa misura formalmente quanto la formazione è lontana dal suo valore medio.

Nell'implementazione della legge si è ricorsi ad un controllore feedback, basato sul controllo cinematico; l'errore è valutato solamente tramite la Jacobiana del task. Sono riportati i risultati (ottenuti da simulazioni Simulink) dell'applicazione di questa legge di controllo, considerata in tre varianti: con e senza termine di proiezione nel null space, e con la presenza di un termine di smorzamento. Si vedrà come queste varianti influiscono sulla robustezza, rispetto alle singolarità, del controllo implementato.

## Capitolo 2

# Uniciclo

In questo capitolo è presente una breve introduzione ai robot mobili in generale, per poi focalizzarsi sulla classe degli unicicli.

### 2.1 Cos'è un robot mobile?

La caratteristica principale dei robot mobili è la presenza di una base mobile che permette al robot di muoversi liberamente nell'ambiente circostante.

Ogni robot di questo tipo deve essere in grado di capire dove si trova e come muoversi al fine di raggiungere un obiettivo desiderato.

Sempre maggiore è il loro uso in applicazioni avanzate che richiedono ampie capacità di moto autonomo. Basandosi sul tipo di applicazione, si distinguono due categorie:

- i *robot da campo (field robots)*, sono applicati in ambienti critici per gli operatori umani: zone di difficile accesso, aree nelle quali non potrebbero sopravvivere o dove potrebbero essere esposti a rischi insostenibili. Questi robot possono svolgere esplorazioni e riportare dati utili riguardanti l'ambiente ad un operatore remoto, usando sensori di bordo adeguati.

Scenari tipici sono le esplorazioni di fondali marini, dell'interno di vulcani, gli interventi in aree contaminate da gas tossici o radiazioni. Importanti risultati sono stati conseguiti anche nelle esplorazioni spaziali: ricordiamo il successo del robot ‘Curiosity Mars Rover’, inviato dalla NASA sul suolo di Marte.

Altri contesti di applicazione sono i disastri causati dagli incendi nei tunnel oppure dai terremoti. In questi casi c’è un gran rischio di ulteriori esplosioni, fughe di gas, crolli; quindi le squadre di soccorso sono affiancate da robot di salvataggio.

- i *robot di servizio (service robots)* sono impiegati invece per applicazioni civili, ad esempio i sistemi di trasporto intelligente, che contribuirebbero ad una riduzione dell'inquinamento.

Questo tipo di robot è stato adottato anche nel settore della sorveglianza e della pulizia automatizzate.

Molti paesi stanno investendo sempre più risorse in un nuovo mercato di robot di servizio che vivano e collaborino con gli esseri umani nella loro

vita quotidiana. Queste tecnologie hanno infatti iniziato, ormai da anni, a trasformarsi in prodotti commerciali: dalla domotica, ai robot giocattolo, ai sistemi automatici per facilitare la vita quotidiana di anziani e di persone con disabilità (sedie a rotelle automatizzate, sollevatori ausiliari, robot riabilitativi). Sempre per quanto riguarda il campo medico poi, questi robot sono sfruttati per la grande accuratezza nel posizionare uno strumento e per la possibilità di effettuare interventi micro-invasivi e chirurgia endoscopica (per mezzo di piccoli robot teleoperati, ossia comandati a distanza).

In questo elaborato siamo interessati alla classe dei *robot mobili su ruote*.

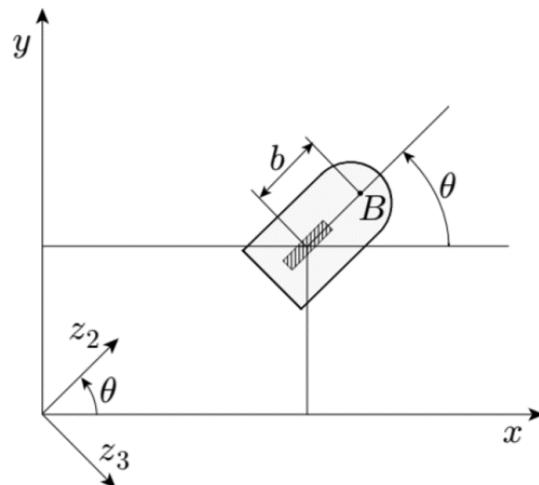
Tipicamente, da un punto di vista meccanico, un robot di questo tipo consiste in un corpo rigido (base o telaio) equipaggiato con un sistema di locomozione (ruote). Altri corpi rigidi (sempre basi o telai), anch'essi equipaggiati di ruote, possono essere connessi alla base mediante dei giunti rotanti.

In particolare avremo a che fare con gli *unicicli*.

## 2.2 Cos'è un uniciclo?

L'uniciclo è uno dei più semplici e compatti modelli di robot mobile. È composto da una singola ruota orientabile, il cui asse è parallelo al piano su cui si muove. La ruota ha un unico punto di contatto con il suolo ed è caratterizzata dal vincolo di puro rotolamento (approfondito nel paragrafo successivo) che impone alla ruota di rotolare sul piano, senza traslare ('*driftless*').

La configurazione dell'uniciclo è descritta da  $q = [x \ y \ \vartheta]^T$ , in cui  $(x,y)$  sono le coordinate cartesiane del punto di contatto della ruota con il suolo (oppure, equivalentemente, del centro della ruota);  $\vartheta$  rappresenta l'orientamento della ruota rispetto all'asse  $x$ .



## 2.3 Modello cinematico di un uniciclo

Come anticipato, una definizione più analitica di uniciclo è: “un'astrazione matematica che si basa sul vincolo di puro rotolamento”.

Questo vincolo, applicato al caso dell'uniciclo, è espresso come segue

$$\dot{x} \sin \vartheta - \dot{y} \cos \vartheta = [\sin \vartheta \quad -\cos \vartheta \quad 0] \dot{q} = 0 \quad (2.1)$$

e comporta che la velocità del punto di contatto sia zero nella direzione ortogonale all'asse sagittale<sup>1</sup> del veicolo. Per questo motivo la linea che passa attraverso il punto di contatto ed ha tale direzione è detta ‘zero motion line’.

Si introduca ora la matrice

$$G(q) = \begin{bmatrix} \cos \vartheta & 0 \\ \sin \vartheta & 0 \\ 0 & 1 \end{bmatrix} \quad (2.2)$$

le cui colonne  $g_1(q)$  e  $g_2(q)$  sono, per ogni  $q$ , una base del null space della matrice associata ai vincoli Pfaffiani. Tutte le velocità generalizzate ammissibili sono ottenute come combinazione lineare di  $g_1(q)$  e  $g_2(q)$  e il modello cinematico dell'uniciclo è quindi rappresentabile come:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\vartheta} \end{bmatrix} = \begin{bmatrix} \cos \vartheta \\ \sin \vartheta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (2.3)$$

in cui gli input  $v$  e  $\omega$  hanno significato fisico:  $v^2$  è la ‘*driving velocity*’ (ad esempio il modulo, con segno, del vettore velocità del punto di contatto);  $\omega$  è la ‘*steering velocity*’, la velocità angolare della ruota intorno al suo asse verticale.

Il modello cinematico, riscritto in forma compatta, è il seguente:

$$\dot{q} = G(q)u \quad (2.4)$$

---

<sup>1</sup>L'asse sagittale dell'uniciclo è l'asse passante per il punto di contatto della ruota con il piano e parallelo alla direzione dell'uniciclo

<sup>2</sup>la velocità  $v$  è data da  $v = \omega r$ , con  $r$  raggio della ruota.

## Capitolo 3

# Controllo di formazione

Una formazione di robot è un gruppo di N robot che mirano a raggiungere un certo scopo, individuale o condiviso.

### 3.1 Sistemi multiagente

Consideriamo un problema di controllo di formazione nel frangente dei sistemi multiagente. Gli agenti, che nel nostro caso sono gli unicicli, possono interagire tra loro in modi differenti. La distinzione maggiore si basa sulla tipologia di obiettivo:

- i '*competitive agents*', agenti caratterizzati da un comportamento egoista: con le loro azioni persegono unicamente il proprio obiettivo locale, anche andando a condizionare negativamente gli obiettivi altrui;
- i '*cooperative agents*', agenti che lavorano insieme, ognuno con le proprie capacità, per raggiungere un obiettivo globale per il sistema.

Come già anticipato, in questa trattazione siamo interessati a quest'ultimo tipo di agenti. I vantaggi dei sistemi multiagente collaborativi sono molteplici:

- sono in grado di risolvere sotto-problemi in parallelo, riducendo il tempo per il completamento del compito dell'intero sistema e riducendo il costo computazionale (i sotto-problemi sono più semplici del problema principale);
- la presenza di N agenti rende il sistema complesso maggiormente robusto e affidabile rispetto ai guasti (nel caso un agente presenti dei guasti, è possibile isolarlo e provvedere alle riparazioni, mentre gli altri continuano a fungere);
- è possibile progettare agenti diversi e specializzati, possibilmente più semplici e meno costosi di un singolo agente complesso.

Possiamo trovare vari esempi di sistemi multiagente collaborativi intorno a noi: in natura abbiamo stormi di uccelli, branchi di pesci, colonie di formiche. Anche le frecce tricolori possono essere viste come un esempio di tale sistema, in quanto ogni singolo aereo esegue determinati movimenti e acrobazie al fine di rimanere in formazione e dare spettacolo.

### 3.2 Modello cinematico di una formazione di unicicli

In questa trattazione è considerata una formazione di sei unicicli; dunque nel seguito sarà sempre considerato  $N=6$ . A partire dalle equazioni viste nel caso del singolo uniciclo, ora siamo interessati al modello cinematico dell'intera formazione. Per far ciò, introduciamo un nuovo vettore di stato

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} \quad (3.1)$$

vettore  $3N$ -dimensionale, che considera tutti i singoli stati dei sei unicicli. Allo stesso modo, è costruito il vettore degli ingressi:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} \quad (3.2)$$

vettore  $2N$ -dimensionale.

La matrice  $G(q)$  della formazione (di dimensioni  $3N \times 2N$ ) presenta lungo la diagonale le matrici  $G_i(q)$  con  $i=1,\dots,6$  dei sei unicicli considerati; è poi completata con zeri

$$G(q) = \begin{bmatrix} \cos \vartheta_1 & 0 & 0 & \dots & \dots & \dots & 0 \\ \sin \vartheta_2 & 0 & \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & 0 & \dots & \dots & \dots & \dots \\ 0 & 0 & \cos \vartheta_2 & 0 & \dots & \dots & \dots \\ \dots & \dots & \sin \vartheta_2 & 0 & \dots & \dots & \dots \\ \dots & \dots & 0 & 1 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \cos \vartheta_N & 0 \\ \dots & \dots & \dots & \dots & \dots & \sin \vartheta_N & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 \end{bmatrix} \quad (3.3)$$

Il modello cinematico, in forma compatta, della formazione si scrive come

$$\dot{q} = G(q)u \quad (3.4)$$

### 3.3 Legge di controllo di formazione

Data una formazione di unicicli, definito il suo modello cinematico ed uno o più task (compiti) che devono essere eseguiti, si possono derivare delle strategie di controllo. Dato un task desiderato per la formazione, lo possiamo scrivere come una funzione di output, che dipende solo dallo stato:

$$y = f(q) \quad (3.5)$$

Questa relazione è tipicamente non lineare<sup>1</sup>. Si procede derivando la funzione di uscita, fino a che l'input del sistema non appare nell'equazione:

$$\dot{y} = \frac{dy}{dt} = \frac{\delta f}{\delta q}(q) = \frac{\delta f}{\delta q}Gu = JGu \quad (3.6)$$

dove  $\frac{\partial f}{\partial q}$  è la Jacobiana, G è la matrice definita nella sezione precedente e u è l'ingresso del sistema.

La relazione può essere riscritta come segue

$$\dot{y} = \bar{J}u \quad (3.7)$$

ponendo  $\bar{J} = JG$ , matrice di trasformazione che mette in relazione l'output con l'input del sistema. Si tratta di una Jacobiana che, per costruzione, tiene conto della cinematica dei robot e che, di conseguenza, permette di generare solo velocità ammissibili.

Data una traiettoria desiderata  $y_D(t)$ , possiamo definire l'errore del task:

$$e(t) = y_D(t) - y(t) \quad (3.8)$$

Derivando l'errore e scegliendo un ingresso per cui la dinamica di esso converge esponenzialmente a zero, si ottiene:

$$\dot{e} = \dot{y}_D - \dot{y} = \dot{y}_D - \bar{J}u = -Ke \quad (3.9)$$

Dalla (3.9), riarrangiando i termini, si ottiene la '*tracking equation*'

$$\bar{J}u = \dot{y}_D + Ke \quad (3.10)$$

A partire dal task desiderato, è possibile ottenere tutti i termini della precedente equazione, a parte il vettore degli ingressi, definito da noi al fine di imporre il controllo alla formazione.

Quindi, andando ad isolare u, si ottiene la *legge di controllo*:

$$u = \bar{J}^*(\dot{y}_D + Ke) \quad (3.11)$$

dove la matrice  $\bar{J}^*$  è una inversa generalizzata di  $\bar{J}$ ; una scelta tipica è la pseudoinversa di Moore-Penrose

$$\bar{J}^* = \bar{J}^T(\bar{J}\bar{J}^T)^{-1} \quad (3.12)$$

---

<sup>1</sup>si tratta di un *Driftless system*

che consente di invertire matrici rettangolari.

È possibile scrivere una *formulazione generale del vettore degli ingressi u*, che permette di rappresentare ogni soluzione:

$$u = \bar{J}^*(\dot{y}_D + Ke) + (I - \bar{J}^*\bar{J})u_0 \quad (3.13)$$

Quest'ultima si differenzia dalla (3.11) per la presenza di un secondo termine

$$(I - \bar{J}^*\bar{J})u_0 \quad (3.14)$$

definito come *termine di proiezione nel null space*<sup>2</sup> della Jacobiana.

Riguardo a quest'ultimo si distinguono due casi:

- $\bar{J}$  è invertibile. Allora la pseudoinversa  $\bar{J}^*$  coincide con l'inversa di  $\bar{J}$  e il termine di proiezione nel null space è nullo;
- $\bar{J}$  non è invertibile. La pseudoinversa non coincide con l'inversa e il termine di proiezione nel null space è non nullo (in quanto  $P \neq 0$ ). Si definisce quindi la matrice  $P$ , detta *matrice di proiezione*,  $P = I - \bar{J}^*\bar{J}$ .

Qualunque vettore (in questo caso  $u_0$ ) sia moltiplicato per  $P$ , viene proiettato nel null space di  $\bar{J}$ . Si può quindi sommare senza problemi il secondo termine  $Pu_0$  al controllo della (3.11) senza perturbare il task originale, ottenendo la (3.13). Infatti l'uscita del sistema controllato rimarrà la stessa, sia con che senza proiezione nel null space. Ciò che varia sono solo le configurazioni interne del sistema.

Questa variante della legge di controllo è molto rilevante, in quanto rende il controllore più robusto quando ci si avvicina (senza raggiungerle) alle singolarità. Segue una spiegazione più accurata.

### 3.3.1 Robustezza della legge di controllo rispetto alle singolarità

Le singolarità si presentano nel caso in cui  $\bar{J}$  non abbia rango pieno; di conseguenza non è possibile calcolarne la pseudoinversa. In corrispondenza delle configurazioni singolari si ha un deterioramento della performance, quantificabile in un aumento dell'errore: le traiettorie desiderabili non sono ottenibili, a causa di aumenti insostenibili nelle velocità degli unicicli. È dunque preferibile evitare situazioni di singolarità.

In questa sezione sono considerati due metodi per rendere la legge di controllo più robusta alle singolarità: la *proiezione nel null space dello Jacobiano* (già introdotta nel paragrafo precedente) e la *Robust kinematic inversion*.

#### Proiezione nel Null Space

Riguardo questo metodo è fondamentale ricordare che si basa sulla matrice pseudoinversa (non calcolabile nel caso di singolarità vera e propria). Dunque questo metodo è applicabile solo nel caso in cui ci si trovi *vicino* a delle singolarità.

---

<sup>2</sup>Il null space di una matrice A è un set di vettori che soddisfano l'equazione omogenea  $Ax=0$ . La nullità di una matrice è la dimensione del sottospazio vettoriale delle soluzioni del sistema lineare omogeneo associato alla matrice

Quando  $\bar{J}$  non è invertibile abbiamo detto che la pseudoinversa non coincide con l'inversa. La Jacobiana è una matrice rettangolare che presenta più colonne che righe. Una matrice del genere è caratterizzata da un null space non vuoto; tutti i vettori che vi si trovano risulteranno in self-motion (cambiamenti nella configurazione non modificano le variabili di output). In queste condizioni il sistema risulta ridondante: non ci sarà un'unica soluzione inversa, ma  $\infty^{N-M}$  soluzioni, con N numero dei gradi di libertà del sistema e M dimensione del task. Tra le infinite soluzioni, noi sceglieremo quella che massimizza una data funzione obiettivo. La funzione obiettivo considerata in questo elaborato è

$$H(q) = \sqrt{\det(\bar{J}\bar{J}^T)} \quad (3.15)$$

detta *indice di manipolabilità*. Si sceglie questa funzione in quanto permette di rendere il controllo con proiezione nel null space più robusto alle singolarità. Infatti l'indice di manipolabilità descrive quanto si è lontani dalle singolarità: questo indice tende a zero tanto più si è prossimi ad una singolarità. Per questo motivo lo si vuole massimizzare.

Per massimizzare H è possibile applicare la seguente formula, che ci permette di stabilire il vettore  $u_0$  che sarà usato nel controllore:

$$Pu_0 = P(\alpha G \nabla_q H) \quad (3.16)$$

dove  $\alpha$  è la *step size* e G è la matrice usata per derivare il modello cinematico<sup>3</sup>.

Bisogna però ricordare che massimizzare l'indice di manipolabilità non da garanzie su cosa accadrà: la massimizzazione può dare i risultati desiderati, portare qualche miglioramento o anche non servire a nulla, visto che non ci sono vincoli esplicativi sulle configurazioni dei robot. Come la proiezione nel null space influisce nelle simulazioni che presentano condizioni di singolarità sarà mostrato successivamente con degli esempi.

### Robust Kinematic Inversion

Nel caso  $\bar{J}$  non abbia rango pieno, non è possibile calcolare la pseudoinversa della Jacobiana come visto precedentemente, in quanto quando la traiettoria di riferimento attraversa una configurazione singolare le velocità calcolate attraverso la pseudoinversa esploderebbero in valori infiniti. Per evitare questa situazione si ricorre alla *Damped Least-Squares inverse*:

$$\bar{J}_{DLS} = \bar{J}^T (\bar{J}\bar{J}^T + \mu_{DLS}^2 I)^{-1} \quad (3.17)$$

dove  $\mu_{DLS}^2$  è il *termine di smorzamento (damping factor)*, elevato al quadrato per garantire che questo termine additivo sia positivo.

Guardando questa equazione si nota come la pseudoinversa di Moore-Penrose si differenzi dalla (3.17) solo per il termine di smorzamento.

Per questa ragione, è possibile definire una regione di singolarità ed attivare il termine

---

<sup>3</sup>la presenza di G(q) garantisce di generare solo velocità ammissibili

di smorzamento solo quando la configurazione entra in questa regione. Si può definire questa regione come fosse una variabile, al fine di assicurarsi un comportamento regolare:

$$\mu_{\text{DLS}}^2 = \begin{cases} 0 & \text{if } \theta_{\min} \geq \epsilon \\ (1 - (\frac{\theta_{\min}}{\epsilon})^2) \mu_{\text{DLS}_{\max}}^2 I^{-1} & \text{if } \theta_{\min} < \epsilon \end{cases} \quad (3.18)$$

$\theta_{\min}$  rappresenta il valore singolare minimo della matrice Jacobiana. È calcolabile tramite la Singular Value Decomposition (SVD), andando a considerare il valore minimo della matrice  $\Sigma$  ottenuta dalla decomposizione rispetto alla configurazione iniziale;  $\epsilon$  è il limite superiore della regione singolare. Questa formulazione di  $\mu_{\text{DLS}}^2$  risulta in un termine di smorzamento massimo  $\mu_{\text{DLS}_{\max}}^2$  quando ci si trova in singolarità.

Da notare che, al fine di utilizzare questa espressione, è necessario valutare ad ogni passo  $\theta_{\min}$ , il che può essere abbastanza costoso da un punto di vista computazionale; per questo si tende ad adottare una stima iterativa.

Il metodo presentato può essere combinato con tutte le altre tecniche di controllo illustrate precedentemente. In particolare bisogna ricordare che, avendo modificato la pseudoinversa, ora nella matrice di proiezione è necessario utilizzare  $\bar{J}_{\text{DLS}}$  per calcolare la matrice di proiezione e il vettore degli ingressi nella formula (3.13):

$$P_{\text{DLS}} = I - \bar{J}_{\text{DLS}} \bar{J} \quad (3.19)$$

$$u = \bar{J}_{\text{DLS}}(\dot{y}_{\text{D}} + K e) + P_{\text{DLS}} \dot{u}_0 \quad (3.20)$$

# Capitolo 4

## Simulazioni

In questo capitolo si mostrano i risultati a cui porta l'applicazione della legge di controllo precedentemente mostrata, nelle diverse varianti descritte. Questa legge è applicata a due entità della formazione di robot, il *centroide* e la *varianza*, affinchè queste seguano le traiettorie desiderate.

### 4.1 Centroide

Il *centroide* è dato dal valore medio della posizione degli unicicli:

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \sum_{i=1}^N x_i \\ \sum_{i=1}^N y_i \end{bmatrix} \quad (4.1)$$

Come obiettivi del controllo, si vuole che il centroide tracci tre tipi di traiettorie (dipendenti dal tempo):

- *traiettoria costante*: il centroide deve rimanere fermo in un punto desiderato ( $x_0, y_0$ ). Quindi si considera una traiettoria del tipo  $f(t)=\text{cost}$ , costante nel tempo. Si tratta di un problema di *set-point regulation*.
- *traiettoria rettilinea*, con equazione  $f(t)=mx+q$ ;
- *traiettoria circolare*, con equazione

$$\begin{aligned} x &= x_0 + r \cos \omega \\ y &= y_0 + r \sin \omega \end{aligned}$$

Nel seguito si farà riferimento ai seguenti valori per i parametri delle equazioni precedenti:

- nella traiettoria costante: cost=10;
- nella traiettoria rettilinea: m=2, q=4;
- nella traiettoria circolare:  $x_0=4, y_0=4, r=3, \omega=2$ .

Si consideri ovviamente che si può scegliere qualsiasi valore da assegnare a questi parametri.

Per quanto riguarda invece il *tempo di simulazione* è stato considerato  $T = 7$  secondi per tutte le simulazioni riguardanti questa sezione sul centroide.

Nel seguito sono riportati i risultati ottenuti dalle simulazioni fatte sulle tre traiettorie desiderate, a partire dalle *condizioni iniziali*:

$$\begin{cases} \mathbf{x}_0 = [5 \ 4 \ 3 \ 2 \ 1 \ 1] \\ \mathbf{y}_0 = [1 \ 1 \ 2 \ 0 \ 0 \ 0] \\ \theta_0 = [30^\circ \ 0^\circ \ 45^\circ \ 90^\circ \ 90^\circ \ 130^\circ] \end{cases}$$

Le condizioni iniziali qui considerate sono lontane dalla singolarità (come sarà spiegato meglio a breve) e per questo i risultati ottenuti saranno gli stessi sia nel caso venga considerato il termine di proiezione nel null space, sia in caso contrario.

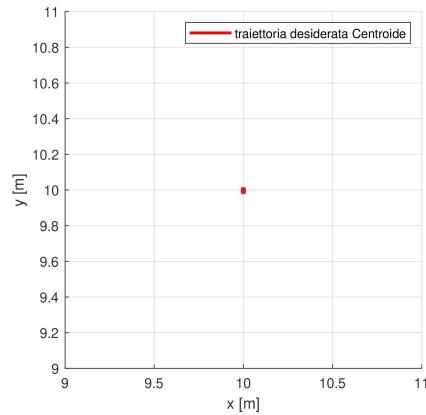
Nelle simulazioni riguardanti il centroide non si è ricorsi all'uso del termine di smorzamento.

Avendo considerato un sistema ideale, non sottoposto a vincoli di saturazione per gli ingressi, dopo diverse prove e simulazioni, si è giunti alla conclusione che il *guadagno* migliore da considerare fosse  $K=50$ , in quanto questo fa sì che l'errore tenda a zero quasi immediatamente, senza destabilizzare il sistema.

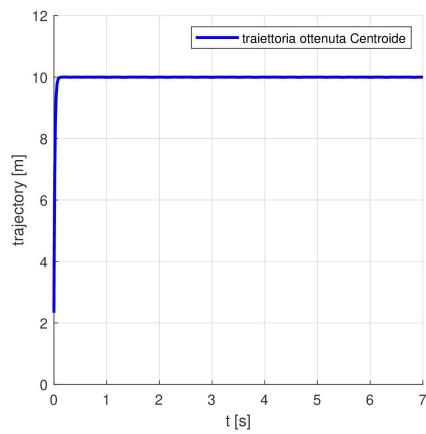
La prima simulazione, in Figura 4.2, riguarda il problema della set-point regulation. Solo in questo caso il grafico della traiettoria ottenuta presenta sull'asse delle x il tempo, in modo da essere più esplicativo.

In Figura 4.5 si riportano i risultati ottenuti nel caso in cui la traiettoria desiderata sia quella rettilinea.

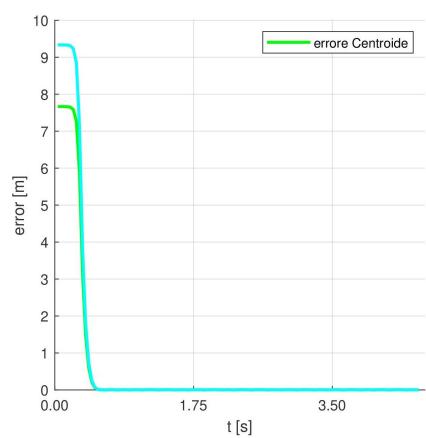
Il caso della traiettoria circolare desiderata è riportato invece in Figura 4.8.



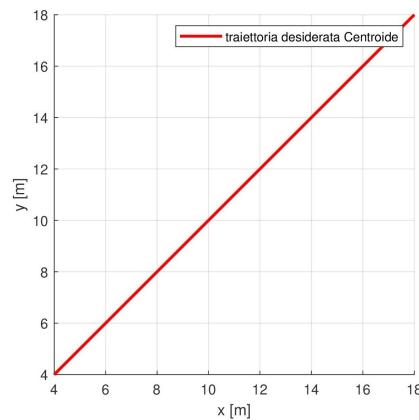
**Figura 4.1** Centroide, set-point regulation. Traiettoria desiderata.



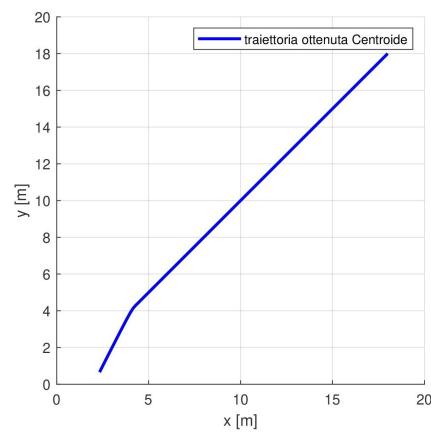
**Figura 4.2** Centroide, set-point regulation. Traiettoria risultante.  
In questo caso la traiettoria è riportata rispetto al tempo sull'asse delle ascisse.



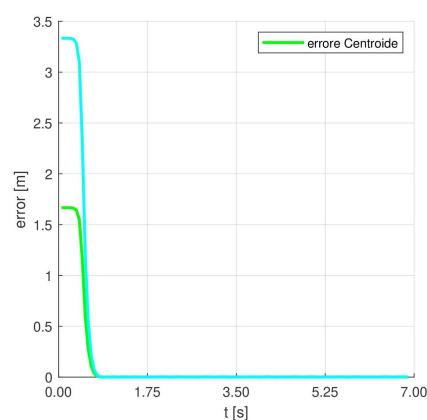
**Figura 4.3** Centroide, set-point regulation. Errore in uscita.



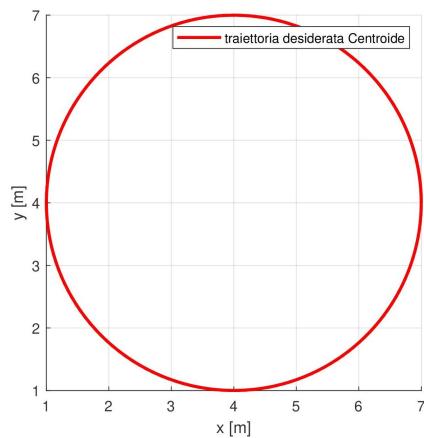
**Figura 4.4** Centroide, traiettoria rettilinea. Traiettoria desiderata.



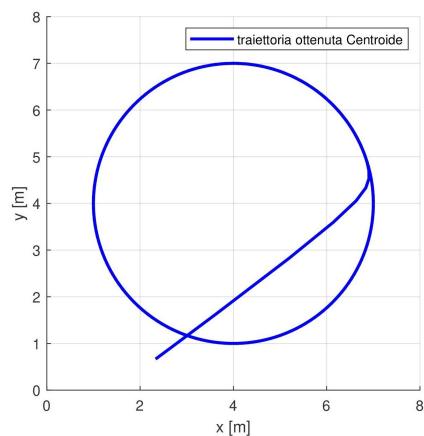
**Figura 4.5** Centroide, traiettoria rettilinea. Traiettoria risultante.



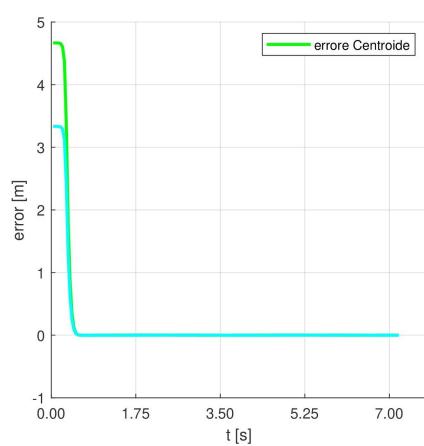
**Figura 4.6** Centroide, traiettoria rettilinea. Errore in uscita.



**Figura 4.7** Centroide, traiettoria circolare. Traiettoria desiderata.



**Figura 4.8** Centroide, traiettoria circolare. Traiettoria risultante.



**Figura 4.9** Centroide, traiettoria circolare. Errore in uscita.

### 4.1.1 Singolarità nel caso del centroide

La matrice Jacobiana  $J$ , nel caso del centroide, è la seguente:

$$J = \frac{\delta f}{\delta q} = \frac{1}{N} \begin{bmatrix} 1 & 0 & 1 & 0 & \dots & 1 & 0 \\ 0 & 1 & 0 & 1 & \dots & 0 & 1 \end{bmatrix} \quad (4.2)$$

con  $N=6$ .

Quindi la matrice  $\bar{J} = JG(q)$  è

$$\bar{J} = \frac{1}{6} \begin{bmatrix} \cos \theta_1 & 0 & \cos \theta_2 & 0 & \dots & \cos \theta_6 & 0 \\ \sin \theta_1 & 0 & \sin \theta_2 & 0 & \dots & \sin \theta_6 & 0 \end{bmatrix} \quad (4.3)$$

con  $N=6$ .

Si può notare come  $\bar{J}$  dipenda solo dall'orientamento degli unicicli, cioè dall'angolo  $\theta$ . Quindi una configurazione risulta singolare quando tutti i coseni o tutti i seni degli angoli sono linearmente dipendenti tra loro, ovvero quando tutti gli unicicli hanno stesso angolo  $\theta$ .

Nei prossimi grafici è mostrato il caso di una configurazione iniziale di unicicli prossima alla singolarità:

$$\left\{ \begin{array}{l} x_0 = [5 \ 4 \ 3 \ 2 \ 1 \ 1] \\ y_0 = [1 \ 1 \ 2 \ 0 \ 0 \ 0] \\ \theta_0 = [0^\circ \ 0^\circ \ 0^\circ \ 0.1^\circ \ 0^\circ \ 0^\circ] \end{array} \right.$$

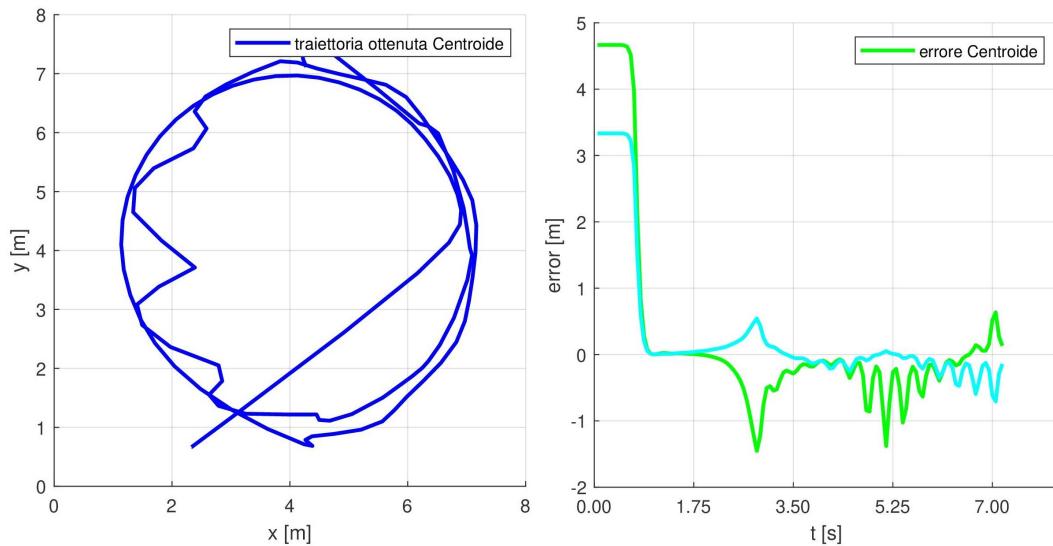
Tutti gli angoli sono posti a zero, tranne  $\theta_4 = 0.1$ . È necessario avere almeno un angolo che si discosti leggermente dagli altri, altrimenti non sarebbe possibile calcolare la pseudoinversa della Jacobiana.

Essendo vicini alla singolarità, le simulazioni con e senza proiezione nel null space si discostano, come si può vedere dai grafici alla pagina successiva. Infatti per quanto riguarda il caso senza proiezione nello spazio nullo di  $\bar{J}$ , il controllo applicato (con matrice di proiezione  $P=0$ ) non riesce a far sì che il centroide segua la traiettoria desiderata (Figura 4.10); invece la proiezione nello spazio nullo porta dei miglioramenti, in quanto infatti la traiettoria che il centroide segue torna ad essere esattamente quella desiderata (Figura 4.11). Questo esempio permette quindi di mostrare come, in alcuni casi, la proiezione nel null space riesca a mantenere il nostro sistema lontano dalla singolarità, evitando che le velocità degli unicicli esplodano e permettendo di ottenere i risultati desiderati.

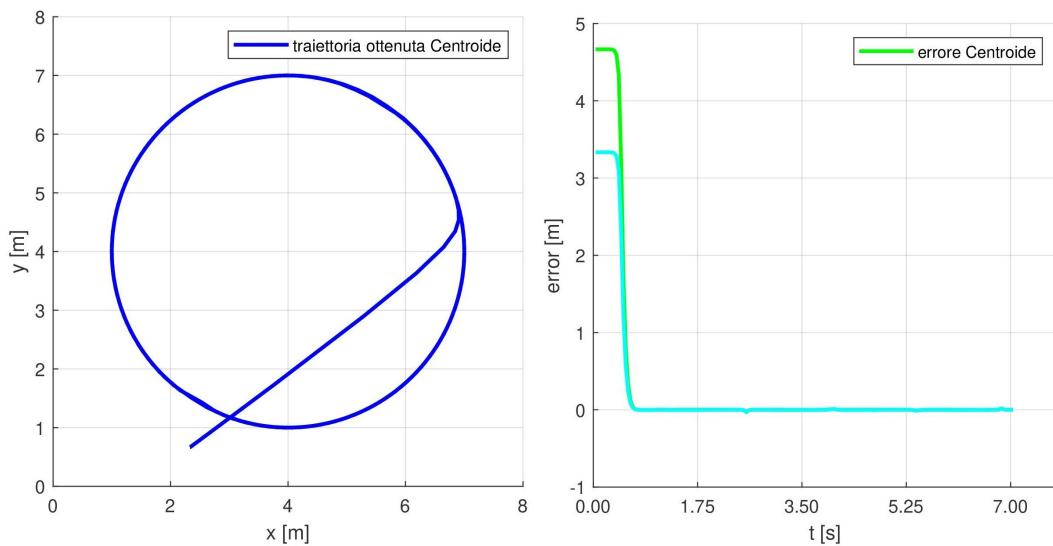
La traiettoria desiderata presa in esame nella seguente simulazione è la traiettoria circolare, caratterizzata dagli stessi parametri considerati precedentemente.

Il tempo di simulazione considerato è di 7 secondi.

Il guadagno preso in considerazione è sempre  $K=50$ .



**Figura 4.10** Traiettoria ed errore del centroide sottoposto a legge di controllo senza proiezione nel null space.



**Figura 4.11** Traiettoria ed errore del centroide sottoposto a legge di controllo con proiezione nel null space.

## 4.2 Varianza

La *varianza* è la deviazione quadratica della posizione dell'uniciclo rispetto al centroide:

$$\begin{bmatrix} \rho_x \\ \rho_y \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \sum_{i=1}^N (x_c - x_i)^2 \\ \sum_{i=1}^N (y_c - y_i)^2 \end{bmatrix} \quad (4.4)$$

con N=6. Misura formalmente quanto è lontana la formazione dal suo valore medio. Come obiettivi del controllo, si vuole che la varianza tracci tre tipi di traiettorie (dipendenti dal tempo):

- *traiettoria rettilinea crescente*, con equazione  $f(t) = m_1 x + q_1$ ,  $m_1 > 0$ ;
- *traiettoria rettilinea decrescente*, con equazione  $f(t) = m_2 x + q_2$ ,  $m_2 < 0$ ;
- *traiettoria sinusoidale*, con equazione  $f(t) = q_0 + b \sin(\frac{2\pi}{k} t)$ , che compie un'oscillazione ogni  $k$  secondi;  $b$  è l'ampiezza del moto.

Nel seguito si farà riferimento ai seguenti valori per i parametri delle equazioni precedenti:

- nella traiettoria rettilinea crescente:  $m_1 = 1$  ,  $q_1 = 3$  ;
- nella traiettoria rettilinea decrescente:  $m_2 = -0.4$  ,  $q_2 = 30$  ;
- nella traiettoria sinusoidale:  $q_0 = 4$  ,  $b = 0.5$  ,  $k = 4$  .

Nello stabilire i parametri delle equazioni delle traiettorie si faccia attenzione: la varianza infatti non può assumere valori negativi, quindi bisogna assegnare valori che facciano sì che la traiettoria della varianza rimanga sempre al di sopra dell'ascissa. Per quanto riguarda invece il *tempo di simulazione*, è stato considerato  $T = 14$  secondi per tutte le simulazioni riguardanti questa sezione sulla varianza.

Nel seguito sono riportati i risultati ottenuti dalle simulazioni fatte sulle tre traiettorie desiderate, a partire dalle *condizioni iniziali*:

$$\begin{cases} x_0 = [5 \ 4 \ 0 \ 2 \ 1 \ 0] \\ y_0 = [1 \ 1 \ 2 \ 0 \ 0 \ 0] \\ \theta_0 = [90^\circ \ 90^\circ \ 30^\circ \ 45^\circ \ 0^\circ \ 90^\circ] \end{cases}$$

Le condizioni iniziali qui considerate sono lontane dalla singolarità (se ne parlerà meglio nel prossimo paragrafo) e per questo i risultati ottenuti sono gli stessi sia nel caso venga considerato il termine di proiezione nel null space, sia in caso contrario.

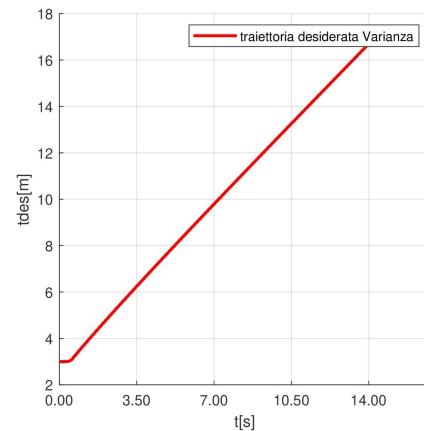
Nelle simulazioni riguardanti la varianza non si è ricorsi all'uso del termine di smorzamento.

Avendo considerato un sistema ideale, non sottoposto a vincoli di saturazione per gli ingressi, dopo diverse prove e simulazioni, si è giunti alla conclusione che il *guadagno* migliore da considerare fosse  $K=55$ , in quanto questo fa sì che l'errore tenda a zero quasi immediatamente, senza destabilizzare il sistema.

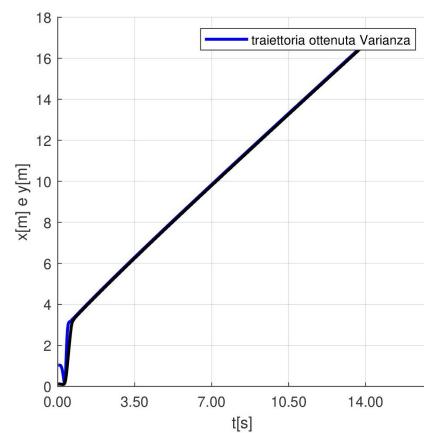
In Figura 4.13 sono riportati i risultati ottenuti nel caso in cui la traiettoria desiderata fosse quella rettilinea crescente; l'errore del sistema, nel tempo, è riportato in Figura 4.14.

Il caso in cui la traiettoria desiderata sia rettilinea decrescente è presentato nelle figure successive: la traiettoria ottenuta in Figura 4.16, l'errore del task in Figura 4.17.

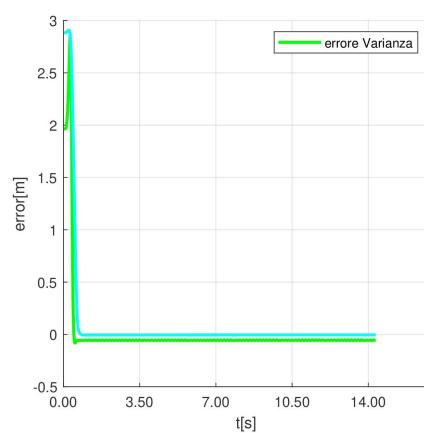
Infine, in Figura 4.19 sono riportati i risultati ottenuti nel caso di traiettoria sinusoidale, in Figura 4.20 l'errore in uscita al sistema.



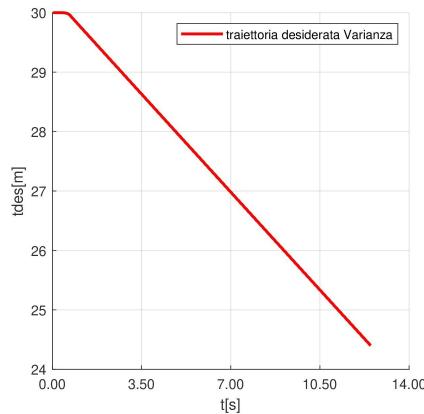
**Figura 4.12** Varianza, traiettoria rettilinea crescente. Traiettoria desiderata.



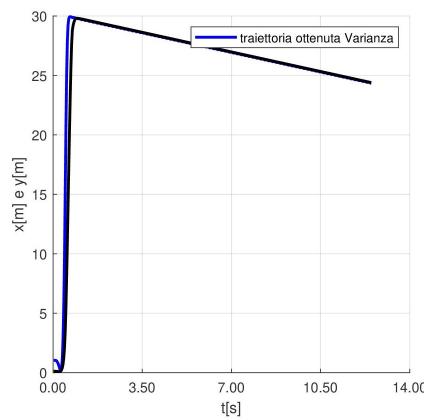
**Figura 4.13** Varianza, traiettoria rettilinea crescente. Traiettoria risultante.



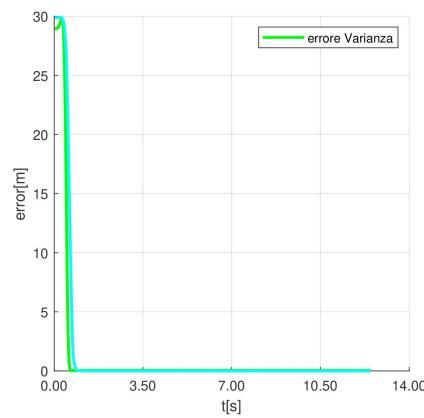
**Figura 4.14** Varianza, traiettoria rettilinea crescente. Errore in uscita.



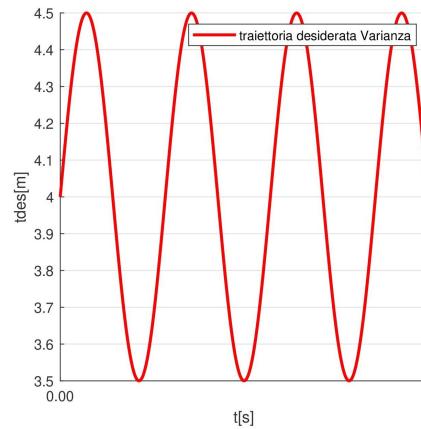
**Figura 4.15** Varianza, traiettoria rettilinea decrescente. Traiettoria desiderata.



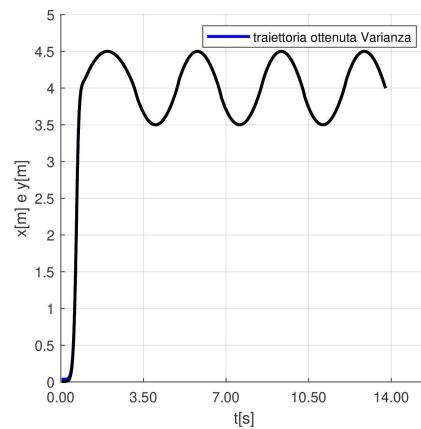
**Figura 4.16** Varianza, traiettoria rettilinea decrescente. Traiettoria risultante.



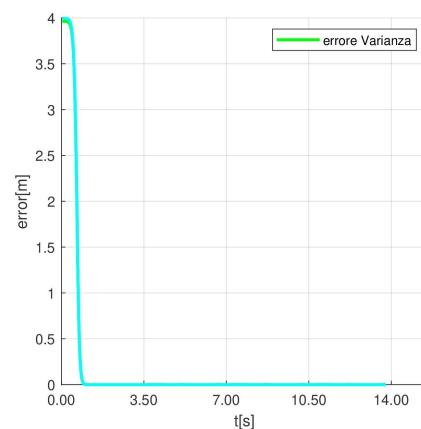
**Figura 4.17** Varianza, traiettoria rettilinea decrescente. Errore in uscita.



**Figura 4.18** Varianza, traiettoria sinusoidale. Traiettoria desiderata.



**Figura 4.19** Varianza, traiettoria sinusoidale. Traiettoria risultante.



**Figura 4.20** Varianza, traiettoria sinusoidale. Errore in uscita.

### 4.2.1 Singolarità nel caso della varianza

La matrice Jacobiana  $J$ , nel caso della varianza, è la seguente:

$$J = \frac{\delta f}{\delta q} = \frac{2}{N^2} \begin{bmatrix} ((N-1)x_1 - x_2 \dots - x_N) & 0 & 0 & \dots \\ 0 & ((N-1)y_1 - y_2 \dots - y_N) & 0 & \dots \end{bmatrix} \quad (4.5)$$

con  $N=6$ .

Quindi la matrice  $\bar{J} = JG(q)$  è

$$\bar{J} = \frac{2}{6^2} \begin{bmatrix} \cos \theta_1 (5x_1 - x_2 \dots - x_6) & 0 & 0 & \dots \\ 0 & \sin \theta_1 (5y_1 - y_2 \dots - y_6) & 0 & \dots \end{bmatrix} \quad (4.6)$$

con  $N=6$ .

In questo caso  $\bar{J}$  non dipende solo dall'orientamento degli unicicli, ma anche dalle coordinate cartesiane degli unicicli. Quindi una configurazione risulta singolare quando tutti gli unicicli sono caratterizzati da uno stesso  $\theta$ , oppure se hanno tutte stessa coordinata  $x$  o stessa coordinata  $y$ <sup>1</sup>.

Nei prossimi grafici è mostrato il caso di una *configurazione iniziale* di unicicli prossima alla singolarità:

$$\begin{cases} x_0 = [5 \ 4 \ 0 \ 2 \ 1 \ 0] \\ y_0 = [0.1 \ 0 \ 0 \ 0 \ 0 \ 0] \\ \theta_0 = [90^\circ \ 90^\circ \ 30^\circ \ 45^\circ \ 0^\circ \ 90^\circ] \end{cases}$$

Tutte le coordinate  $y$  degli unicicli sono poste a zero, tranne  $y_1 = 0.1$ <sup>2</sup>.

In questo esempio, in cui siamo quindi molto vicini alla singolarità, si può vedere come la proiezione nel null space porti ad un miglioramento, ma non istantaneo: infatti solo dopo un errore iniziale di alcuni secondi (Figura 4.21), la traiettoria della varianza si instaura sulla traiettoria desiderata.

La *traiettoria desiderata* presa in esame nella seguente simulazione è quella sinusoidale, caratterizzata dagli stessi parametri presi in esame precedentemente.

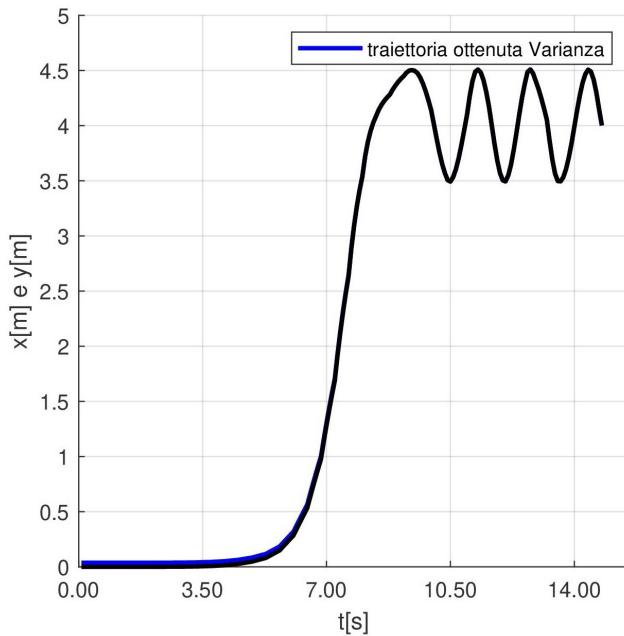
Il *tempo di simulazione* considerato è di 14 secondi.

Il *guadagno* preso in considerazione è sempre  $K=55$ .

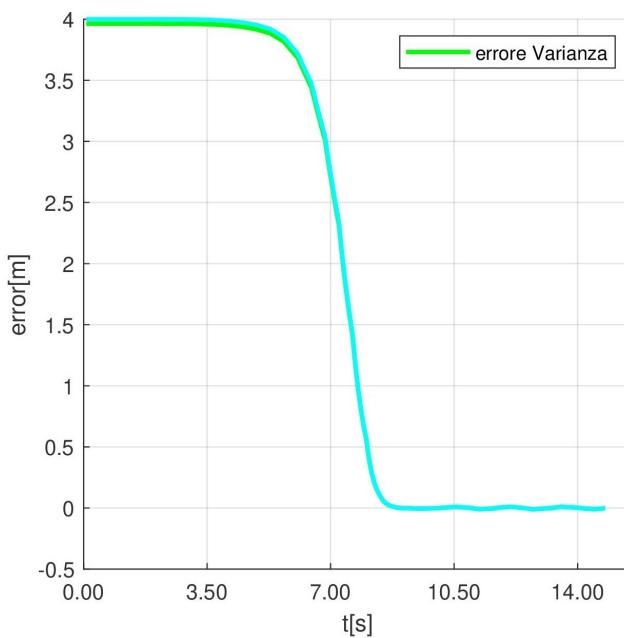
---

<sup>1</sup>Esistono varie configurazioni che portano alla singolarità, ma questi sono i casi più scontati.

<sup>2</sup>È necessario avere almeno una coordinata che si discosti leggermente dalle altre, altrimenti non sarebbe possibile calcolare la pseudoinversa della Jacobiana.



**Figura 4.21** Traiettoria della varianza sottoposta a legge di controllo con proiezione nel null space.



**Figura 4.22** Errore della varianza sottoposta a legge di controllo con proiezione nel null space.

### 4.3 Centroide e Varianza

In questo paragrafo si considera sempre la stessa legge di controllo, che però va a regolare contemporaneamente le traiettorie di centroide e varianza, combinandole tra loro (centroide costante e varianza rettilinea crescente, centroide rettilineo e varianza sinusoidale, ...).

La Jacobiana in questo caso è così fatta:

$$J = \begin{bmatrix} J_c \\ J_v \end{bmatrix} \quad (4.7)$$

Di conseguenza

$$\bar{J} = \begin{bmatrix} \bar{J}_c \\ \bar{J}_v \end{bmatrix} \quad (4.8)$$

Per quanto riguarda le singolarità, in questo caso valgono le stesse configurazioni singolari del caso precedente. Inoltre, le singolarità algoritmiche diventano una questione rilevante<sup>3</sup>. Proprio per questo motivo, per gestire al meglio le singolarità, oltre al termine di proiezione nello spazio nullo, nelle simulazioni si è ricorsi anche all'applicazione della Robust Kinematic Inversion. Il parametro  $\epsilon$  è stato posto pari a 0.01, mentre  $\sigma_{\min}$  è stato trovato andando a prendere il valore minimo sulla diagonale della matrice  $\Sigma$ .  $\Sigma$  è stata calcolata su matlab tramite la *singular value decomposition (svd)*.  $\sigma_{\min}$  è quindi inizialmente preso pari a 0.4082.

Vediamo subito un esempio di come l'applicazione della Robust Kinematic Inversion porti miglioramenti nell'inseguimento di traiettoria. Si considerino le seguenti *condizioni iniziali*:

$$\begin{cases} x_0 = [1 \ 2 \ 3 \ 2 \ 1 \ 1] \\ y_0 = [0 \ 2 \ 3 \ 5 \ 6 \ 0] \\ \theta_0 = [30^\circ \ 45^\circ \ 15^\circ \ 90^\circ \ 40^\circ \ 1^\circ] \end{cases}$$

Sono riportati i risultati di una simulazione fatta rispetto alle traiettorie circolare per il centroide e rettilinea decrescente per la varianza. Si mostra come le singolarità algoritmiche siano rilevanti in questo caso e come i risultati ottenuti dalle simulazioni si differenzino a seconda che si usi il termine di smorzamento nel calcolo della pseudoinversa o meno. In Figura 4.24 si vede come la traiettoria, dopo un certo intervallo temporale, peggiori significativamente, a causa della singolarità: in questo caso è stata usata la classica pseudoinversa di Moore-Penrose nella legge di controllo.

---

<sup>3</sup>Ad esempio si consideri una formazione di due unicicli: questa presenta 4 ingressi disponibili (due velocità lineari e due angolari). È immediato notare che la matrice  $\bar{J}$  della (4.8) sarà sempre singolare: ha quattro colonne, due delle quali fatte solo di zeri, cosicché il rango non sarà mai maggiore di due. Nessuno dei due task dipende dalla velocità angolare; il che significa che non importa quanti ( $N$ ) unicicli consideriamo, tra le  $2N$  colonne metà di queste saranno nulle (quelle relative alle velocità angolari). Per questo motivo, abbiamo bisogno di almeno quattro unicicli per avere la possibilità di avere quattro colonne indipendenti, così da evitare le singolarità algoritmiche.

Ricorrendo invece al termine di smorzamento, attivato nelle regioni di singolarità, le entità della formazione conservano il comportamento desiderato per tutto il tempo della simulazione, come è possibile riscontrare in Figura 4.27.

Nelle restanti simulazioni, è stata presa in esame la stessa configurazione iniziale precedentemente adottata, ma singolare rispetto all'orientamento degli unicicli. Per ottenere i risultati desiderati si è ricorsi contemporaneamente al termine di smorzamento ed al termine di proiezione nello spazio nullo della Jacobiana.

In Figura 4.30 la traiettoria considerata per il centroide è quella costante, mentre la varianza segue una traiettoria sinusoidale.

In Figura 4.33 il centroide segue una traiettoria circolare, la varianza una traiettoria rettilinea crescente.

In Figura 4.36 il centroide segue una traiettoria rettilinea crescente, la varianza una traiettoria sinusoidale.

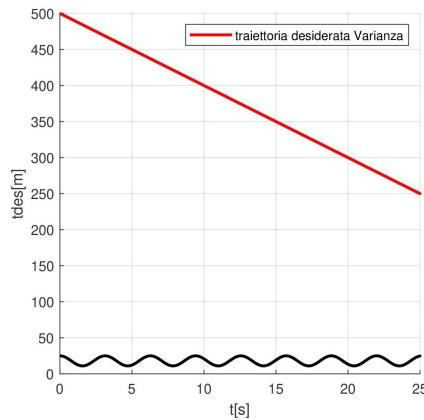
Il *tempo di simulazione* considerato è sempre  $T = 25$  secondi; per quanto riguarda invece il *guadagno*, il sistema è stato strutturato in modo da poter assegnare due guadagni diversi, in modo da ottenere i migliori risultati possibili: il guadagno del centroide  $K_c = 10$  e il guadagno della varianza  $K_v = 5$ .

I *parametri* delle traiettorie desiderate presi in esame nelle seguenti simulazioni sono:

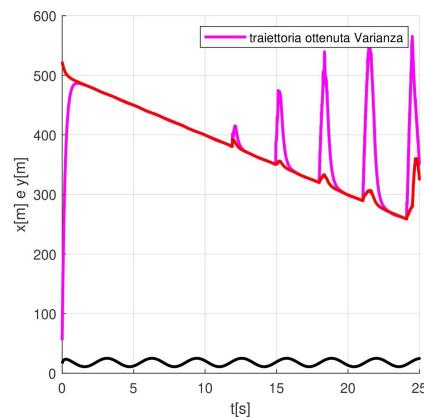
- nella traiettoria costante del centroide: cost=30;
- nella traiettoria rettilinea del centroide: m=1, q=6;
- nella traiettoria circolare del centroide:  $x_0=18$ ,  $y_0=18$ ,  $r=9$ ,  $\omega=2$ ;
- nella traiettoria rettilinea crescente della varianza:  $m_1=5$  ,  $q_1=500$  ;
- nella traiettoria rettilinea decrescente della varianza:  $m_2= -10$  ,  $q_2=500$  ;
- nella traiettoria sinusoidale della varianza:  $q_0=300$  ,  $b=40$  ,  $k=4$  .

Da notare che, a differenza di prima, anche la traiettoria del centroide è rappresentata su grafici che sull'asse delle ascisse presentano il tempo, al fine di poterla riportare sullo stesso grafico della varianza: quindi, quello che su un grafico xy era una circonferenza, qui sarà equivalente a due sinusoidi, una per la componente x del centroide ed una per la componente y.

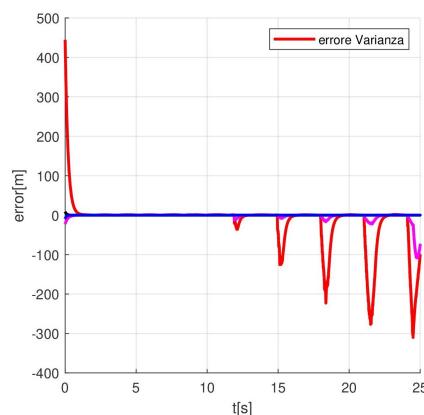
Nei grafici le linee blu e nere si riferiscono alle componenti (x,y) del centroide, mentre le linee rosse e magenta si riferiscono alle componenti (x,y) della varianza.



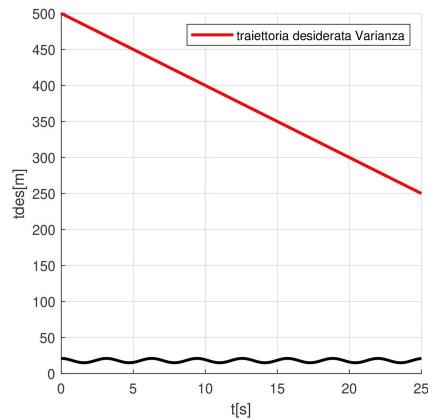
**Figura 4.23** Traiettorie desiderate. Centroide in nero e blu, traiettoria sinusoidale; varianza in rosso e magenta, traiettoria rettilinea decrescente.



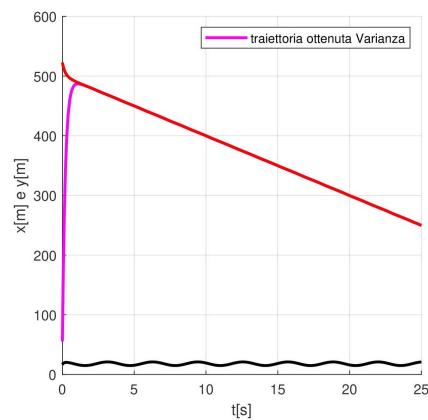
**Figura 4.24** Traiettorie ottenute. Centroide in nero e blu, traiettoria sinusoidale; varianza in rosso e magenta, traiettoria rettilinea decrescente. No *damping factor*.



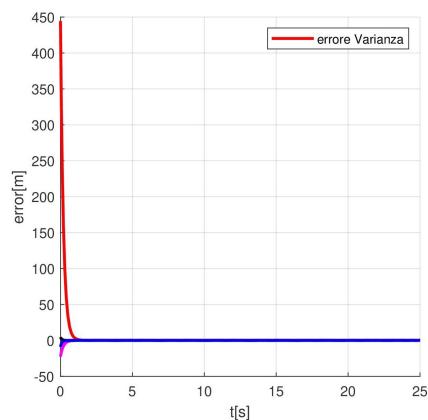
**Figura 4.25** Errore in uscita. Centroide in nero e blu, traiettoria sinusoidale; varianza in rosso e magenta, traiettoria rettilinea decrescente.



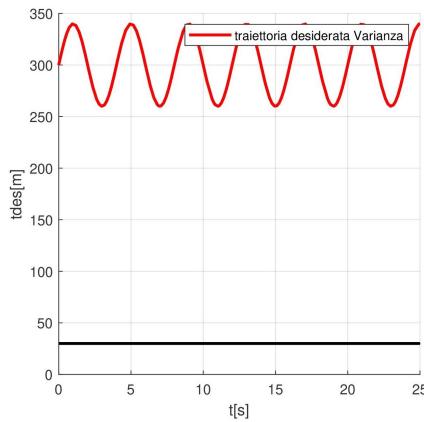
**Figura 4.26** Traiettorie desiderate. Centroide in nero e blu, traiettoria sinusoidale; varianza in rosso e magenta, traiettoria rettilinea decrescente.



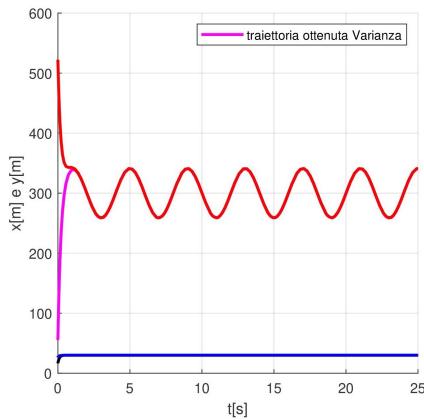
**Figura 4.27** Traiettorie ottenute. Centroide in nero e blu, traiettoria sinusoidale; varianza in rosso e magenta, traiettoria rettilinea decrescente. Si *damping factor*.



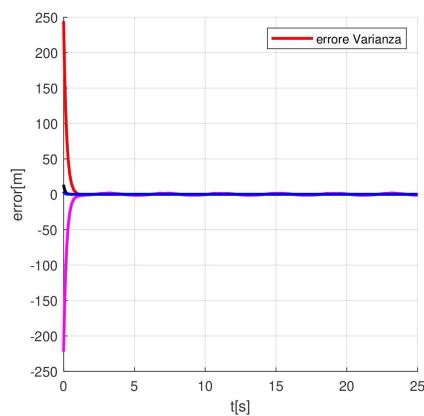
**Figura 4.28** Errore in uscita. Centroide in nero e blu, traiettoria sinusoidale; varianza in rosso e magenta, traiettoria rettilinea decrescente.



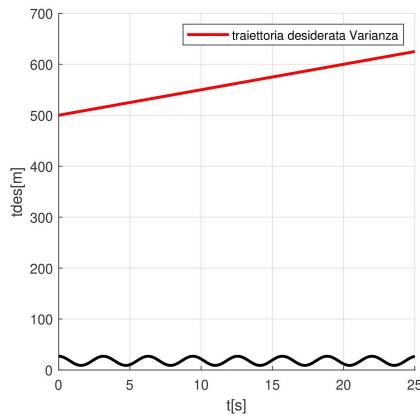
**Figure 4.29** Traiettorie desiderate. Centroide in nero e blu, set-point regulation; varianza in rosso e magenta, sinusoidale.



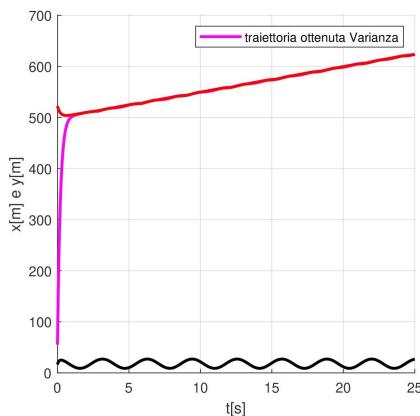
**Figure 4.30** Traiettorie ottenute. Centroide in nero e blu, set-point regulation; varianza in rosso e magenta, traiettoria sinusoidale.



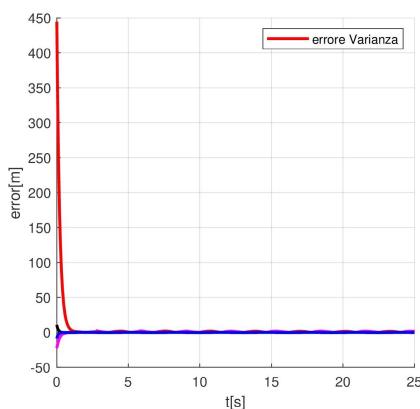
**Figure 4.31** Errore in uscita. Centroide in nero e blu, set-point regulation; varianza in rosso e magenta, traiettoria sinusoidale.



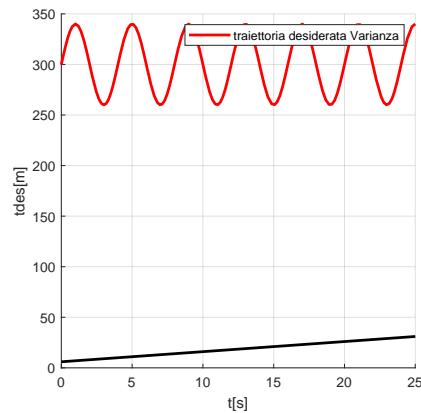
**Figure 4.32** Traiettorie desiderate. Centroide in nero e blu, traiettoria sinusoidale; varianza in rosso e magenta, traiettoria rettilinea crescente.



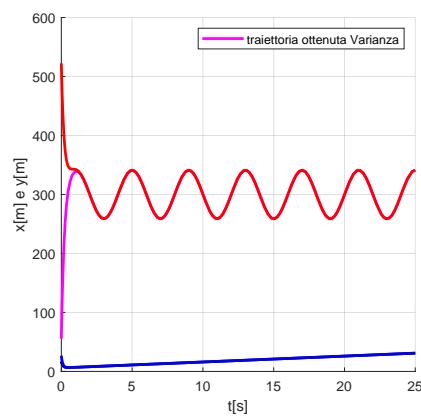
**Figure 4.33** Traiettorie ottenute. Centroide in nero e blu, traiettoria sinusoidale; varianza in rosso e magenta, traiettoria rettilinea crescente.



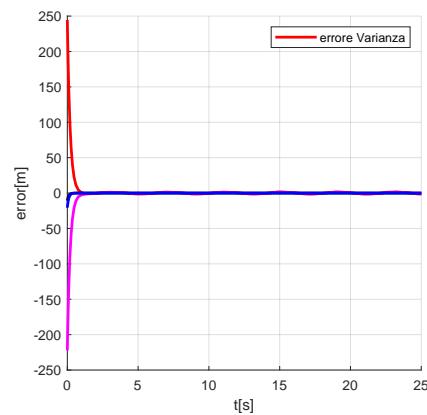
**Figure 4.34** Errore in uscita. Centroide in nero e blu, traiettoria sinusoidale; varianza in rosso e magenta, traiettoria rettilinea crescente.



**Figura 4.35** Traiettorie desiderate. Centroide in nero e blu, traiettoria rettilinea; varianza in rosso e magenta, traiettoria sinusoidale.



**Figura 4.36** Traiettorie ottenute. Centroide in nero e blu, traiettoria rettilinea; varianza in rosso e magenta, traiettoria sinusoidale.



**Figura 4.37** Errore in uscita. Centroide in nero e blu, traiettoria rettilinea; varianza in rosso e magenta, traiettoria sinusoidale.

# Capitolo 5

## Conclusioni

In questa trattazione è stata analizzata la legge di controllo di una formazione di sei unicicli. La legge è stata applicata a due entità della formazione: centroide e varianza. Sono stati riportati, tramite grafici MatLab, i risultati ottenuti dall'applicazione di questa legge in tre varianti diverse, andando a confrontare come ogni variante influisce sulla robustezza del controllo rispetto alle singolarità.

La prima possibilità consiste nell'uso di una legge di controllo come in Formula (3.11), senza termine di proiezione nel null space della Jacobiana del sistema. Abbiamo però visto che l'applicazione di una legge di controllo di questo tipo, nel caso di configurazioni iniziali degli unicicli vicine alla singolarità, porta ad un aumento insostenibile nelle velocità degli unicicli ed un conseguente deterioramento delle performance.

Per porre rimedio alla precedente situazione è stata introdotta una legge di controllo con termine di proiezione nel null space della Jacobiana del sistema (Formula (3.13)): questo termine è dato dalla moltiplicazione tra la matrice di proiezione  $P$  e il vettore  $u_0$  scelto in modo da massimizzare una funzione obiettivo. La funzione obiettivo qui considerata è l'indice di manipolabilità  $H(q)$ , il quale tende a zero tanto più si è prossimi ad una singolarità. Per questo motivo andare a massimizzare  $H(q)$  rende il controllo con proiezione nel null space più robusto alle singolarità. Questo non comporta però una garanzia: la massimizzazione può portare i risultati desiderati, solo dei miglioramenti o anche non servire a nulla, in quanto non ci sono vincoli espliciti sulle configurazioni dei robot. Si ricorda che, poiché questo controllo dipende sempre dal calcolo della pseudoinversa, può essere applicato solo nel caso in cui si è vicini ad una singolarità (perchè se il sistema fosse in singolarità ( $\bar{J}$  non ha rango pieno) non sarebbe possibile calcolare la pseudoinversa).

Infine la terza possibilità consiste nell'introduzione di un termine di smorzamento nel calcolo della pseudoinversa (Formula (3.17)). Questo approccio può essere applicato senza problemi nel caso la pseudoinversa non sia calcolabile (quando  $\bar{J}$  non ha rango pieno). Si definisce una regione di singolarità, in modo da attivare il termine di smorzamento solo quando la configurazione entra in questa regione. Questo metodo può poi essere combinato con le due tecniche di controllo precedenti al fine di cercare di ottenere i risultati desiderati.

# Bibliografia

- [1] "Robotics, Modeling, Planning and Control" - B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo
- [2] Wheeled mobile robots: modeling, planning and control - Oriolo Giuseppe, Luglio 2017
- [3] L. Pallottino, "Sistemi Robotici Distribuiti" - Versione del 4 Marzo 2015
- [4] "I sistemi multiagente, così governiamo l'intelligenza artificiale" - Agenda Digitale, 14 Febbraio 2020
- [5] "Control of unicycle type robots: tracking, path following and point stabilization" - Ricardo Carona, A. Pedro Aguiar, Jose Gaspar

## Ringraziamenti

*Questo spazio lo dedico alle persone che hanno contribuito, con il loro instancabile supporto, a questo importante traguardo.*

*Un ringraziamento speciale al mio Relatore Prof. Giuseppe Oriolo e a Filippo Smaldone per i loro consigli, la disponibilità e le conoscenze trasmesse durante tutto il percorso di stesura dell'elaborato.*

*Un grazie di cuore al mio collega Luca, con il quale ho condiviso questi tre anni così intensi, tra ansie, 'segmentation faults', lama e "No Sofia questa è una domanda stupida. Mi rifiuto di risponderti". Non sarebbe stato lo stesso senza di te.*

*Grazie alle mie amiche Giulia Di, Monda, Morgana e al mio Pancreas. Vi ho raggruppate in poche righe, ma non basterebbe un trattato per esprimere il bene che vi voglio. Grazie per essere sempre presenti, ormai da tanti (troppi) anni, e aver condiviso con me gioie, dolori e soprattutto pazzie e pettigolezzi. Sono certa che ancora molte ne seguiranno.*

*Questo percorso mi ha anche regalato una persona speciale, Gabriele. Grazie di aver ascoltato tutti gli sfoghi e le ansie di questo periodo e di essermi stato accanto in ogni momento. Ti ringrazio di avermi regalato, tra un polletto e l'altro, tutti quei momenti semplici ma perfetti, che porterò sempre con me.*

*Dulcis in fundo, ringrazio infinitamente Mamma e Papà, che mi hanno sempre sostenuto, appoggiando ogni mia decisione e sopportando tutti i miei sfoghi nei momenti di stress pre-esame e non solo. È solo grazie a loro se sono arrivata fin qui, grazie ai loro consigli (che mi pesa dirlo, ma alla fine si dimostrano essere Quasi sempre giusti) e al loro amore (un po anche grazie alla loro 'pesantezza'). Vi amo.*

*Infine, dedico questa tesi a me stessa, alle nottate di studio e alla mia tenacia, che mi hanno permesso di raggiungere questo risultato e che mi permetteranno di perseguire i miei obiettivi futuri.*