

TESI DI LAUREA

Calcolo delle Traiettorie Ottime per Quadricotteri con Metodo Indiretto in presenza di ostacoli intermedi

Candidato:

Adam Seewald

Matricola VR407856

Relatore:

Prof. Paolo Fiorini

Correlatore:

Dott. Fabrizio Boriero

Mame a Starým Rodičom
Moji učitelia životom a môj nekonečný vzor.

(A mia Madre e ai miei Nonni
Miei maestri di vita e mia eterna guida.)

Indice

1	Introduzione	1
1.1	Descrizione del problema	1
1.2	Motivazioni	5
1.2.1	Quadricotteri	6
1.2.2	Generazione della traiettoria per quadricotteri	6
1.3	Struttura e contributi	8
2	Stato dell'arte	9
2.1	Ottimizzazione della traiettoria	10
2.1.1	Soluzione analitica	12
2.1.2	Metodi diretti	18
2.1.3	Metodi indiretti	19
3	Dinamica del quadricottero	23
3.1	Introduzione	24
3.2	Sottoattuazione	26
3.3	Modello dinamico	28
3.3.1	Equazioni del moto	30
3.4	Input di controllo	30
3.4.1	Input di controllo per una data traiettoria	30
3.4.2	Condizioni di fattibilità	32
3.5	Modello dinamico vincolato	32
3.6	Implementazione	35
4	Metodi diretti	37
4.1	Collocazione trapezoidale	38

4.1.1	Inizializzazione	39
4.1.2	Programma non lineare	40
4.1.3	Formulazione generale	41
4.2	Collocazione di Hermite-Simpson	44
4.2.1	Formulazione generale	46
5	Metodi indiretti	49
5.1	Introduzione	51
5.1.1	Principio Minimo di Pontryagin	52
5.2	Formulazione del problema	54
5.3	Condizione necessaria per l'ottimalità	57
5.4	Ricerca della traiettoria ottima	59
5.4.1	Funzionale costo aumentato	61
5.4.2	Varianza prima del funzionale costo	62
5.4.3	Algoritmo di convergenza ottima	64
5.5	Riassunto	64
6	Risultati delle simulazioni	67
6.1	Simulazione in assenza di ostacoli	68
6.1.1	Osservazioni	69
6.2	Simulazione in presenza di ostacoli	73
6.2.1	Simulazione del braccio robotico	73
6.2.2	Simulazione del quadricottero	73
6.2.3	Osservazioni	74
7	Conclusioni e lavoro futuro	79
7.1	Presentazione del lavoro	79
7.2	Lavoro futuro	82
	Bibliography	84

Elenco delle figure

1.1	Schema della generazione di traiettoria ottima con metodo indiretto	2
1.2	Il controllo di un quadricottero	7
3.1	Schema della dinamica del quadricottero	23
3.2	Sistemi di coordinate di un quadricottero	28
5.1	Schema della ricerca della traiettoria ottima	49
5.2	Schema dell'algoritmo della ricerca della condizione necessaria	60
5.3	Schema dell'algoritmo della discesa del gradiente	64
6.1	Controllo $\mathbf{u}(t)$ sub-ottimo	70
6.2	Controllo $\mathbf{u}_1^*(t)$ ottimo e stato \mathbf{q} senza ostacoli	71
6.3	Controllo $\mathbf{u}_2^*(t)$ ottimo e stato \mathbf{q} senza ostacoli	72
6.4	Stato della traiettoria ottima del braccio robotico	75
6.5	Controllo $\mathbf{u}^*(t)$ ottimo e stato \mathbf{q} in presenza ostacoli	76
6.6	Controlli in presenza di ostacoli con vincoli diversi	77

Ringraziamenti

La mia più sincera gratitudine va all’instancabile Correlatore di questa tesi Dott. Fabrizio Boriero, per la sua guida, il suo costante aiuto e la sua fiducia in me. Non secondo per importanza ringrazio di tutto il Relatore Prof. Paolo Fiorini, per il supporto e per aver dato un titolo, una struttura e un contenuto alla mia tesi. Ringrazio anche Prof. Nicola Sansonetto, per le sue dritte con le teorie matematiche alla base di questa tesi.

Questo lavoro è stato reso possibile anche grazie a tutto lo staff del Laboratorio di Robotica Altair al Dipartimento di Informatica. Ai miei cari colleghi, Davide Costanzi, Giacomo De Rossi, Andrea Roberti, Enrico Sartori, Chiara Tenga e Carlo Tadiello insieme a tutti gli altri va un sentito grazie. In particolare ringrazio Michela De Piccoli e Enrico Magnabosco per la loro sincera amicizia.

I miei compagni di Università presenti e passati, con cui ho condiviso questi cinque anni, sono stati tutti essenziali per me.

Il più sentito grazie per essermi stati vicino va ai miei pazienti amici Sergio Alberti e Alessia Bozzini. Alessia, mia compagna di banco e complice di tante avventure e sventure. Le sono grato di tutto.

Abstract

La generazione autonoma della traiettoria per quadricotteri è una disciplina relativamente nuova della quale teoria è stata sviluppata solo recentemente. In questa tesi viene presentato lo stato dell'arte degli algoritmi presenti per la ricerca di un controllo ottimo mediante metodi diretti e indiretti. I metodi indiretti forniscono una soluzione matematica più accurata anche se computazionalmente più onerosa. Il recente sviluppo di controllori ad alte prestazioni a bordo dei veicoli aerei autonomi ha permesso l'implementazione di questi metodi anche durante le normali operazioni di volo. I metodi indiretti vengono considerati alla luce del Principio Minimo di Pontryagin, e i parametri di controllo vengono aggiustati con l'algoritmo della discesa del gradiente. Il problema della traiettoria ottima viene esteso al caso di un quadricottero limitato con determinati vincoli intermedi.

Capitolo 1

Introduzione

Questa tesi si pone come obiettivo la generazione di traiettorie ottime per veicoli aerei autonomi, comunemente noti come droni, mediante una libreria di ottimizzazione che viene sviluppata in parallelo con la tesi. La generazione autonoma di una traiettoria per veicoli aerei è una disciplina relativamente nuova che ha avuto il picco nel suo sviluppo solo recentemente. Gli algoritmi proposti in letteratura si basano soprattutto sull'uso di metodi matematici per la risoluzione del problema dinamico, noti con il nome di metodi diretti di collocazione. Questi sono veloci e efficienti in robotica, dove è necessario un controllo preciso e accurato su un sottoinsieme dello spazio globale in cui è posizionato il robot. Molto spesso i veicoli aerei autonomi, invece, devono gestire un spazio molto più vasto ma molto meno denso di ostacoli e vincoli intermedi. Per questa classe di problemi si vuole, invece di adoperare i metodi diretti già ampiamente usati, implementare i metodi indiretti, che sono matematicamente più accurati. Si userà un funzionale costo che rappresenta il fenomeno che si vuole minimizzare nella ricerca della traiettoria ottima, e si cercherà di trovare un minimo locale mediante il soddisfacimento di una condizione necessaria.

1.1 Descrizione del problema

In un problema di ricerca di una traiettoria ottima, come quello trattato dalla libreria implementata in questa tesi, è necessario prima di tutto

implementare un modo per la definizione del sistema fisico che si intende adoperare. Sono state implementate delle classi, l'estensione delle quali permette di rappresentare il moto di un quadricottero nello spazio, nonché la presenza di vincoli intermedi di vario tipo. Nei paragrafi successivi viene spiegato in modo dettagliato cosa si intende per questo approccio, e in particolare, come la libreria può essere personalizzata ad una grande varietà di problemi dove è necessario trovare un controllo ottimo.

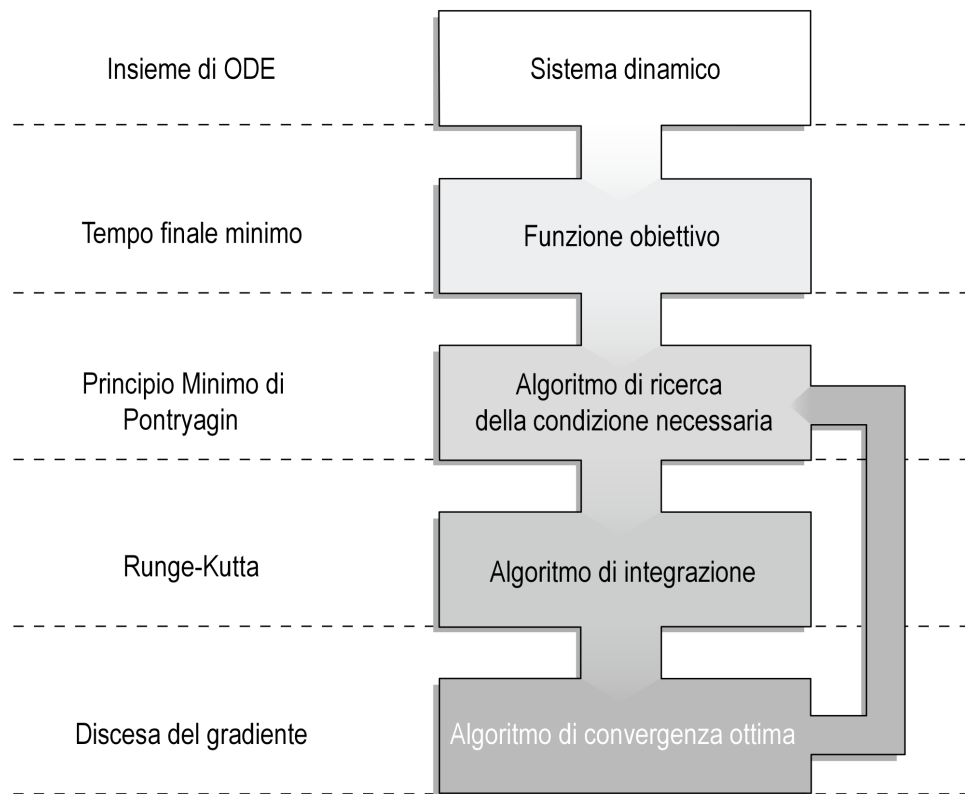


Figura 1.1: Schema concettuale dell'algoritmo della generazione di una traiettoria ottima mediante un metodo indiretto

Il problema che vogliamo trattare è quindi composto da una parte che rappresenta il *sistema dinamico* in analisi, da una *funzione obiettivo* che deve essere minimizzata, da un *algoritmo di integrazione* che permette di effettuare un passo di minimizzazione a partire da un *algoritmo di ricerca della condizione necessaria*, e un *algoritmo di convergenza ottima*: ovvero un algoritmo di raffinamento del controllo ottimo in modo che ogni iterazione del

algoritmo di ottimizzazione di partenza apporti dei miglioramenti e converga alla traiettoria ottima desiderata.

In particolare in questa tesi il sistema dinamico viene rappresentato mediante un insieme di equazioni differenziali ordinarie. La funzione obiettivo è un funzionale che può rappresentare il controllo del sistema o il tempo, in questo modo è possibile effettuare un'ottimizzazione sul controllo o sul tempo minimo. L'algoritmo di integrazione è implementato mediante integratori di Runge-Kutta di vari ordini, che vengono scelti automaticamente dalla libreria in base al sistema fisico sotto analisi (ovvero in base al tipo di derivata che rappresenta il sistema). L'algoritmo di ricerca della condizione necessaria è implementato con il Principio Minimo di Pontryagin e permette di valutare la funzione costo aumentata e di trovarne un minimo (locale). Per raffinare la soluzione con nuovi parametri iniziali viene usato come algoritmo di convergenza ottima la discesa del gradiente.

Partiamo con il problema di ottimizzazione della traiettoria di un quadricottero con un esempio. Immaginiamo un satellite che si sposta tra due pianeti. Ci riferiamo al termine *traiettoria* per descrivere il cammino che fa il satellite tra i due pianeti. Normalmente, questo cammino contiene sia lo stato (i.e. posizione e velocità) che il controllo (i.e. la spinta) come funzione del tempo. Il termine *ottimizzazione della traiettoria* si riferisce ad un insieme di metodi che sono usati per trovare la migliore traiettoria tra un insieme di traiettorie ammissibili. Questo avviene normalmente raffinando gli ingressi al sistema, noti come controlli, in funzione del tempo.

Prima di descrivere il problema dell'ottimizzazione della traiettoria matematicamente, dobbiamo chiarire la terminologia standard usata da qui in avanti. Adoperiamo la seguente notazione:

- t_k è il tempo al punto k
- N è il numero di segmenti della traiettoria (spline)
- $h_k = t_{k+1} - t_k$ è la durata della spline k
- $x_k = x(t_k)$ è lo stato al punto k
- $u_k = u(t_k)$ è il controllo al punto k

- $w_k = w(t_k, x_k, u_k)$ è l'integrando della funzione obiettivo al punto k
- $\dot{q} = \frac{d}{dt}q$, $\ddot{q} = \frac{d^2}{dt^2}q$ è la derivata prima e seconda di q in funzione del tempo

Per semplicità successivamente adoperiamo la notazione $k + \frac{1}{2}$ per indicare il punto intermedio della spline k . Di conseguenza se u_k indica il controllo all'inizio del segmento k , $u_{k+\frac{1}{2}}$ indica il controllo del punto intermedio.

Per spiegare la formulazione matematica alla base dei problemi di ottimizzazione della traiettoria, vediamo un semplice esempio che descrive il movimento di un blocco tra due punti. Questo esempio verrà ulteriormente rielaborato nella sezione 2.1.1. Il punto di inizio e fine è un punto di quiete in cui il blocco non si muove e il tempo che il blocco impiega è prefissato. Prima di tutto dobbiamo scrivere la *dinamica*, che descrive come il sistema si muove. In questo caso modelleremo il blocco come un punto di massa che viaggia in una dimensione, e il controllo (input) al sistema è semplicemente la forza che viene applicata al blocco. Usiamo x per posizione, v per velocità e u per controllo; quindi possiamo descrivere la dinamica del sistema come segue.

$$\dot{x} = v \quad \dot{v} = u \quad (1.1)$$

In questo caso, vogliamo che il blocco si muova di un'unità di distanza in un'unità di tempo, e vogliamo che sia stazionario sia al punto di inizio che al punto di fine. Queste condizioni sono note anche con il nome di **condizioni alla frontiera** o boundary conditions, e possono essere riassunte come segue.

$$\begin{aligned} x(0) &= 0 & x(1) &= 1 \\ v(0) &= 0 & v(1) &= 0 \end{aligned} \quad (1.2)$$

Una soluzione al problema di ottimizzazione della traiettoria è corretta e fattibile se soddisfa tutte le condizioni del problema, dette anche **vincoli**. In genere esistono molti tipi diversi di vincoli. Per il semplice problema di movimento del blocco, abbiamo solo due tipi di vincoli: la dinamica del sistema e le condizioni alla frontiera. L'insieme di controlli che producono

una traiettoria corretta sono detti controlli ammissibili. Il problema dell'ottimizzazione della traiettoria si occupa di trovare la migliore traiettoria ammissibile tra tutto l'insieme, detta anche **traiettoria ottima**. Usiamo la **funzione obiettivo** per descrivere in modo matematico cosa intendiamo con la traiettoria ottima. Per esempio, il problema del movimento del blocco può essere risolto mediante la forza minima quadrata (equazione 1.3) oppure con il minimo lavoro assoluto (equazione 1.4).

$$\min_{u(t), x(t), v(t)} \int_0^1 u^2(\tau) d\tau \quad (1.3)$$

$$\min_{u(t), x(t), v(t)} \int_0^1 |u(\tau)v(\tau)| d\tau \quad (1.4)$$

1.2 Motivazioni

Con questa tesi vogliamo implementare una libreria che permette la generazione di traiettorie ottime per quadricotteri mediante metodi indiretti. In letteratura la maggior parte dei generatori di traiettorie usano metodi diretti, come collocation method. Nonostante i metodi indiretti forniscono una soluzione matematicamente precisa al problema di generazione della traiettoria ottima, richiedono più tempo e risorse computazionali delle loro controparti dirette.

Con questa tesi vogliamo generare in modo automatico una traiettoria per quadricotteri, che utilizza metodi indiretti, e consente di introdurre dei vincoli intermedi che devono essere evitati dalla traiettoria ottima risultante dall'operazione di ottimizzazione.

Comunemente noti come droni, i veicoli aerei senza pilota o unmanned aerial vehicles UAV sono veicoli che possono compiere missioni di volo autonomamente senza un pilota umano a bordo, o possono essere tele-operati da un pilota situato su una stazione a terra. Il grado di autonomia degli UAV varia, ma normalmente hanno delle funzioni autonome di base: auto livellamento implementato mediante un'unità IMU, mantenimento della posizione usando un sistema di navigazione satellitare GNSS, e mantenimento dell'altezza usando un barometro. I veicoli più autonomi offrono anche funzioni come il decollo e atterraggio automatico, la pianificazione della traiettoria, e

l'evitamento degli ostacoli [1]. In generale un UAV può essere visto come un robot volante. In letteratura gli UAV hanno molti nomi diversi come micro aerial vehicle MAV, unmanned aerial system UAS, vertical take-off and landing aircraft VTOL, o multicottero. In questa tesi ci concentriamo sui UAV con quattro propulsori noti anche come quadricotteri.

1.2.1 Quadricotteri

Nell'ultimo decennio i quadricotteri sono stati adoperati per molte applicazioni. Si tratta di veicoli aerei relativamente semplici da sviluppare ma allo stesso tempo di essere molto affidabili con un elevato grado di manovrabilità.

Data la loro eccezionale agilità, i quadricotteri sono usati come popolari veicoli aerei miniaturizzati. Grazie alla configurazione dei propulsori decentralizzata è possibile ottenere una dinamica rotazionale molto veloce ed efficiente. Normalmente un rapporto spinta-peso alto permette grande accelerazione transazionale quando il quadricottero è in assenza di carico [2].

I quadricotteri vengono adoperati in molti ambienti e stanno cambiando il modo in cui molti lavori vengono fatti, i.e. riprese foto e video, mappatura aerea, ispezione degli edifici e operazioni di ricerca e salvataggio o SAR. Durante questi compiti, il quadricottero viene controllato da un pilota umano. In particolare in SAR e nella ispezione degli edifici, i quadricotteri sono un essenziale supporto quando le aree interessate sono pericolose o non percorribili a causa di contaminazioni, instabilità strutturali o spazi troppo ristretti.

1.2.2 Generazione della traiettoria per quadricotteri

Anche quando un quadricottero può essere tele operato da un pilota esperto, se sono presenti degli ostacoli, non è sempre facile trovare la traiettoria corretta: in questi casi si può incorrere in un allungamento dei tempi operazionali, in una mancata esplorazione di zone interessate, o talvolta, nella perdita stessa del quadricottero. E' quindi opportuno supportare le abilità di guida del pilota con algoritmi di ottimizzazione e sensori presenti sui circuiti del veicolo. Inoltre i quadricotteri spesso necessitano, in condizioni descritte sopra, di evitare ostacoli mediante svolte brusche o mediante il passaggio in

zone strette. Le traiettorie derivanti da simili operazioni non convenzionali sono difficili da calcolare da un algoritmo di ottimizzazione convenzionale. Questa difficoltà è dovuta soprattutto al fatto che un quadricottero è un sistema sottoattuato: esso ha solo quattro controlli di input per muoversi in uno spazio di lavoro di sei dimensioni. Per esempio, per muoversi in avanti, il quadricottero deve inclinarsi nella direzione del moto, modificando l'angolo di pitch, per generare sufficiente spinta. Il dispositivo non è in grado di muoversi in una configurazione orizzontale (con l'angolo di pitch nullo), e di conseguenza non può percorrere un'apertura lunga e stretta che invece risulterebbe ad esempio percorribile in una configurazione perfettamente orizzontale.

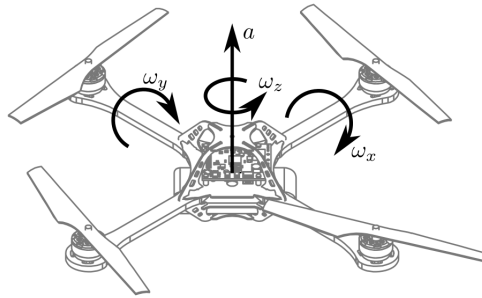


Figura 1.2: Il controllo di un quadricottero: i tassi di rotazione ω_x , ω_y e ω_z vengono monitorati da un controllore a bordo usando il feedback del giroscopio

L'impostazione UAV con quattro rotori è la più comune configurazione multi rotore studiata in letteratura per la semplicità della modellazione e del controllo rispetto ad altre piattaforme. Consiste di quattro rotori elettrici con propulsori montati in fondo ad una struttura a croce [3]. La configurazione dinamica permette il decollo e l'atterraggio in spazi ridotti, il mantenimento di quota sopra obiettivi target, nonché il movimento omnidirezionale nello spazio. Tuttavia, il maggior svantaggio di un quadricottero è il consumo di corrente dovuto alla rotazione ininterrotta degli attuatori, che riduce il tempo di volo. Inoltre, i multi rotori sono noti per essere instabili, in quanto soggetti a effetti delle vibrazioni e del rumore: il controllo dinamico diventa

molto complesso in presenza di turbolenza, deflessione dei propulsori e altre perturbazioni che sono incorporate nel modello dinamico sottoattuato.

1.3 Struttura e contributi

Nei capitoli successivi di questa tesi viene ampliato il problema trattato, quello della ottimizzazione delle traiettorie per quadricotteri, e vengono proposte soluzioni, anche alternative a quelle che si pone questa tesi.

Il presente capitolo introduce brevemente il problema e i concetti base necessari per la comprensione dei capitoli successivi. Viene introdotto il problema dell'ottimizzazione della traiettoria e vengono poste basi teoriche per i principi adoperati nei capitoli successivi. Vengono date anche definizioni iniziali sui UAV, in particolare sui quadricotteri, su cui si concentra questa tesi.

Nel capitolo 2 ci occupiamo di definire il problema di ottimizzazione della traiettoria mediante metodi diretti così come è presente nella letteratura mediante l'implementazione teorica di una soluzione analitica, alla base delle soluzioni in letteratura. Quindi viene proposto un esempio che viene risolto con due metodi analitici. In letteratura questi metodi vengono implementati per il calcolo numerico della soluzione della traiettoria ottima qualora la soluzione analitica risulta troppo complessa. La successiva implementazione della soluzione analitica per via numerica viene trattata nel capitolo 5.

Nel capitolo 3 definiamo la cinematica e la dinamica del problema trattato. Quindi viene presentata una variante vincolata di un quadricottero in movimento che viene poi rielaborata nel capitolo successivo.

Nel capitolo 4 introduciamo i metodi diretti per l'ottimizzazione della traiettoria. Questi metodi sono il diretto concorrente dei metodi indiretti che sono trattati in questa tesi. Per una comprensione approfondita del problema è quindi essenziale definire anche l'approccio con questi metodi.

Nel capitolo 5 descriviamo il cuore di questa tesi, ovvero la risoluzione della traiettoria ottima con metodi indiretti.

Capitolo 2

Stato dell'arte

La maggior parte della letteratura sull'ottimizzazione della traiettoria mediante metodi indiretti si basa sugli approcci presentati nel libro *Applied Optimal Control* di Bryson e Ho nel '75 [4].

Gli avanzamenti nei metodi numerici per l'ottimizzazione della traiettoria hanno portato a sviluppo nell'esplorazione spaziale [5] e in molti altri settori quali l'aviazione e la robotica. Molti schemi che sono stati proposti in letteratura solitamente formulano il problema del controllo ottimo in un problema ai limiti [6], che ha delle richieste piuttosto elevate della potenza computazionale: la soluzione del controllo diventa ad anello aperto. Di conseguenza, la teoria del controllo ottimo viene raramente adoperata in applicazioni real-time.

L'idea dell'uso delle tecniche del controllo ottimo per la guida di veicoli aerei non è nuova. Con l'aumento della potenza di calcolo, numerosi ricercatori hanno cercato una soluzione numerica adoperabile con vincoli real-time. Nelle parti successive di questo capitolo introduciamo la teoria alla base della quasi totalità degli approcci indiretti per la ricerca della traiettoria del controllo ottima attualmente presenti in letteratura, ovvero quella introdotta da Bryson e Ho. Una trattazione più approfondita del problema viene posticipata ai capitoli successivi. Prima di definire l'approccio usato effettivamente in questa tesi faremo un resoconto anche degli altri metodi adoperati nella letteratura, "concorrenziali" ai metodi indiretti, le loro controparti dirette.

Per riassumere l'approccio di Bryson e Ho, e di conseguenza la maggior

parte degli approcci che vengono implementati se si vuole ricavare la traiettoria del controllo ottimo con metodi indiretti, usiamo l'esempio del blocco in movimento tra due punti di frontiera stabili che abbiamo introdotto nella sezione 1.1.

2.1 Ottimizzazione della traiettoria

Vediamo come viene trattato il problema di ottimizzazione della traiettoria da Bryson e Ho. È innanzitutto essenziale formulare il problema che vogliamo trattare. Successivamente il problema viene risolto per via analitica, da cui derivano i metodi indiretti.

Esistono molti modi per formulare il problema di ottimizzazione della traiettoria [7–9]. In questo capitolo ci restringiamo al problema d'esempio presentato nel capitolo 1.1. In questo modo possiamo semplificare la formulazione al problema di ottimizzazione della traiettoria a singola fase e a tempo continuo: ovvero ci restringiamo ad un problema in cui la dinamica del sistema è continua lungo tutta la traiettoria. Una formulazione più generale è descritta in [9].

In genere, una funzione obiettivo può includere due termini: un obiettivo di frontiera $J(\cdot)$ e un integrale di cammino lungo tutta la traiettoria, con l'integrando $w(\cdot)$. Un problema con entrambi i termini è detto essere nella **forma di Bolza**. Un problema con solo la parte integrale è detto nella **forma di Lagrange**, mentre un problema con solo la parte obiettivo è detto nella **forma di Mayer** [7].

$$\min_{t_0, t_F, x(t), u(t)} J(t_0, t_F, x(t_0), x(t_F)) + \int_{t_0}^{t_F} w(\tau, x(\tau), u(\tau)) d\tau \quad (2.1)$$

In ottimizzazione, usiamo il termine **variabile di decisione** per descrivere le variabili che il risolutore raffina per minimizzare la funzione obiettivo. Per il problema del movimento del blocco, le variabili di decisione sono il tempo iniziale e finale (t_0, t_F) , così come lo stato $x(t)$ e il controllo $u(t)$ rispettivamente. L'ottimizzazione è soggetta a molti limiti e vincoli, che vengono descritti nelle seguenti equazioni. La prima, e probabilmente la più

importante, delle equazioni è la dinamica del sistema. Questa è tipicamente non lineare e descrive come il sistema cambia nel tempo.

$$\dot{x}(t) = f(t, x(t), u(t)) \quad (2.2)$$

La seguente equazione descrive i vincoli sul cammino, che rinforzano le restrizioni lungo la traiettoria. Un vincolo di cammino può essere usato, per esempio, per tenere il piede di un robot che cammina sopra la terra mentre esegue un passo.

$$h(t, x(t), u(t)) \leq 0 \quad (2.3)$$

Un'altra importante equazione rappresenta il vincolo non lineare sulla frontiera, che pone delle restrizioni sullo stato iniziale e finale del sistema. Un tale vincolo può essere adoperato, per esempio, per assicurarsi che il passo di un robot che cammina è periodico.

$$g(t_0, t_F, x(t_0), x(t_F)) \leq 0 \quad (2.4)$$

Spesso ci sono dei limiti costanti sullo stato o sul controllo. Per esempio, un braccio robotico potrebbe avere dei limiti sull'angolo, sul coefficiente angolare, oppure sul momento che potrebbe essere applicato lungo tutta la traiettoria. L'equazione 2.5 rappresenta le restrizioni di cammino sullo stato, mentre l'equazione 2.6 le restrizioni di cammino sul controllo.

$$x_{\text{low}} \leq x(t) \leq x_{\text{upp}} \quad (2.5)$$

$$u_{\text{low}} \leq u(t) \leq u_{\text{upp}} \quad (2.6)$$

Infine, è spesso fondamentale includere dei limiti specifici sul tempo e stato iniziale e finale. Questi possono essere usati per assicurare che la soluzione al problema della pianificazione della traiettoria raggiunge il goal entro un determinato lasso temporale. L'equazione 2.7 introduce dei limiti per il tempo iniziale e finale, l'equazione 2.8 limiti sullo stato iniziale, mentre l'equazione 2.9 sullo stato finale.

$$t_{\text{low}} \leq t_0 < t_F \leq t_{\text{upp}} \quad (2.7)$$

$$x_{0,\text{low}} \leq x(t_0) \leq x_{0,\text{upp}} \quad (2.8)$$

$$x_{F,\text{low}} \leq x(t_F) \leq x_{F,\text{upp}} \quad (2.9)$$

2.1.1 Soluzione analitica

In questa sezione vediamo come viene trovata una soluzione al problema di ottimizzazione della traiettoria per via analitica. La soluzione analitica è il modo in cui viene trovata ottimizzata la traiettoria dai metodi indiretti presenti in letteratura e introdotti in [4]. Come vedremo, essi cercano le condizioni necessaria per minimizzare la funzione costo aumentata

Torniamo brevemente all'esempio del blocco in movimento tra due punti di cui vogliamo calcolare la traiettoria ottima. Modelleremo il blocco come punto di massa unitario che scorre senza attrito in una dimensione. Lo *stato* del blocco è rappresentato dalla sua posizione x e dalla velocità v , mentre il *controllo* del blocco è rappresentato dalla forza applicata al blocco come già visto nell'equazione 1.1. Successivamente dobbiamo definire i vincoli di frontiera che descrivono lo stato iniziale e finale del blocco. Decidiamo che vogliamo far muovere il blocco da $x = 0$ al tempo $t = 0$ a $x = 1$ al tempo $t = 1$. Poiché vogliamo che il blocco parta e arrivi in quiete, imponiamo la velocità iniziale e finale entrambe a $v(0) = v(1) = 0$ così come descritto nell'equazione 1.2. Una traiettoria che soddisfa sia la dinamica del sistema che le condizioni di frontiera è detta fattibile, e il corrispondente controllo ammissibile. Una traiettoria è ottima se minimizza una funzione obiettivo. In genere, siamo interessati a trovare una soluzione al problema dell'ottimizzazione della traiettoria che è sia fattibile che ottima. Per questo esempio adoperiamo una funzione obiettivo relativamente comune, ovvero l'integrale del controllo quadrato (equazione 1.3). Viene scelto solitamente questa funzione obbiettivo per sistemi semplici poiché tende a produrre una soluzione alla traiettoria continuamente derivabile.

Il problema può essere trattato in due modalità differenti; la prima tratta il problema come un vero problema di controllo ottimo, dove lo stato e il controllo sono separati e la dinamica del sistema è gestita con funzioni moltiplicatore. La seconda soluzione semplifica il problema prima sostituendo la dinamica nella funzione costo.

Soluzione completa

Vogliamo minimizzare la funzione costo $J(\cdot)$, data sopra, dove u è la forza (ovvero il controllo) applicata al blocco.

$$J(t, \mathbf{q}, u) = \int_0^1 u^2(\tau) d\tau \quad (2.10)$$

La dinamica del sistema $\mathbf{f}(\cdot)$ è data di sotto, con x che rappresenta la posizione, v la velocità, e $\mathbf{q} = \begin{bmatrix} x & v \end{bmatrix}^T$ è il vettore stato.

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \mathbf{f}(\mathbf{q}, u) = \begin{bmatrix} v \\ u \end{bmatrix} \quad (2.11)$$

Applicheremo anche le seguenti condizioni di frontiera, dove i pedici indicano il valore della frontiera della traiettoria.

$$\mathbf{q}_0 = \mathbf{q}(t)|_{t=0} = \begin{bmatrix} x_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{q}_1 = \mathbf{q}(t)|_{t=1} = \begin{bmatrix} x_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (2.12)$$

Dobbiamo soddisfare la dinamica per assicurare che la soluzione sia fattibile. Questo avviene modificando il funzionale costo in modo che includa la dinamica del sistema e un vettore delle funzioni moltiplicatore $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_x & \lambda_v \end{bmatrix}^T$. Possiamo notare che quando la dinamica è soddisfatta $\mathbf{f} - \dot{\mathbf{q}} = 0$ e quindi $\bar{J} = J$ a prescindere delle funzioni moltiplicatore.

$$\bar{J} = \int_0^1 (u^2(\tau) + \boldsymbol{\lambda}^T (\mathbf{f} - \dot{\mathbf{q}})) d\tau \quad (2.13)$$

Ora possiamo usare l'integrazione per parti per riscrivere la funzione costo così modificata. Usiamo la notazione con il pedice per indicare il valore della frontiera della traiettoria, per esempio: $\boldsymbol{\lambda}_0 = \boldsymbol{\lambda}(t)|_{t=0}$.

$$\bar{J} = \boldsymbol{\lambda}_0^T \mathbf{q}_0 - \boldsymbol{\lambda}_1^T \mathbf{q}_1 + \int_0^1 \left((u^2(\tau) + \boldsymbol{\lambda}^T \mathbf{f}) + (\dot{\boldsymbol{\lambda}}^T \mathbf{q}) \right) d\tau \quad (2.14)$$

A questo punto, è utile definire due quantità che saranno comodo durante il resto della derivazione. La prima è la **Lagrangiana** \mathcal{L} (equazione 2.15), ed indica il termine all'interno dell'integrale della funzione costo originale

J . La seconda è l'**Hamiltoniana** \mathcal{H} (equazione 2.16), ed indica la somma della Lagrangiana e il prodotto del vettore delle funzioni moltiplicatore con la dinamica del sistema [4].

$$\mathcal{L} = u^2 \quad (2.15)$$

$$\mathcal{H} = \mathcal{L} + \boldsymbol{\lambda}^T \mathbf{f} = u^2 + \lambda_x v + \lambda_v u \quad (2.16)$$

Consideriamo un semplice problema di ottimizzazione: trovare il minimo di una funzione scalare. Avremo il minimo quando la derivata prima di questa funzione è nulla e la derivata seconda è positiva. Un principio simile può essere usato per le traiettorie, sebbene usiamo il termine **variazione** invece di derivata. Una traiettoria ottima deve avere una variazione prima nulla, e una variazione seconda non negativa. Ci concentriamo sulla condizione necessaria, ovvero che la variazione prima sia nulla.

Supponiamo che la traiettoria ottima è data da \mathbf{q}^* e \mathbf{u}^* . Una traiettoria sub-ottima (equazione 2.17) può essere scritta come somma di una traiettoria ottima e una piccola perturbazione dalla traiettoria, come mostrato sotto dove ε è un parametro e $\delta\mathbf{q}$ e $\delta\mathbf{u}$ sono piccole (arbitrarie) variazioni nello stato e nel controllo.

$$\mathbf{q} = \mathbf{q}^* + \varepsilon\delta\mathbf{q} \quad \mathbf{u} = \mathbf{u}^* + \varepsilon\delta\mathbf{u} \quad (2.17)$$

La variazione prima della funzione costo è la derivata parziale rispetto al parametro ε .

$$\delta\bar{J} \equiv \left. \frac{\partial}{\partial \varepsilon} \bar{J} \right|_{\varepsilon=0} \quad (2.18)$$

Usando le regole di derivazione, possiamo scrivere un'espressione per la variazione prima della funzione costo [4].

$$\begin{aligned} \delta\bar{J} &= \boldsymbol{\lambda}_0^T \left. \frac{\partial \mathbf{q}_0}{\partial \varepsilon} \right|_{\varepsilon=0} - \boldsymbol{\lambda}_1^T \left. \frac{\partial \mathbf{q}_1}{\partial \varepsilon} \right|_{\varepsilon=0} + \int_0^1 \left[\left. \frac{\partial \mathcal{H}}{\partial \varepsilon} \right|_{\varepsilon=0} + \dot{\boldsymbol{\lambda}}^T \left. \frac{\partial \mathbf{q}}{\partial \varepsilon} \right|_{\varepsilon=0} \right] d\tau \\ &= \boldsymbol{\lambda}_0^T \delta\mathbf{q}_0 - \boldsymbol{\lambda}_1^T \delta\mathbf{q}_1 + \int_0^1 \left[\left(\frac{\partial \mathcal{H}}{\partial \mathbf{q}} + \dot{\boldsymbol{\lambda}}^T \right) \delta\mathbf{q} + \frac{\partial \mathcal{H}}{\partial \mathbf{u}} \delta\mathbf{u} \right] d\tau \end{aligned} \quad (2.19)$$

La variazione prima della funzione costo $\delta\bar{J}$ (equazione 2.19) deve essere nulla lungo la traiettoria ottima. Le variazioni sullo stato e sui punti iniziali

e finali della traiettoria sono nulle, poiché le condizioni di frontiera sono prefissate ($\delta \mathbf{q}_0 = 0$, $\delta \mathbf{q}_1 = 0$). Dunque i primi due termini dell'equazione 2.19 sono entrambi nulli. Le variazioni sugli stati $\delta \mathbf{q}$ e sui controlli $\delta \mathbf{u}$ lungo la traiettoria sono arbitrarie, quindi ognuno dei loro coefficienti deve essere nullo in modo da avere un termine integrale nullo.

$$\frac{\partial \mathcal{H}}{\partial \mathbf{q}} + \dot{\boldsymbol{\lambda}}^T = 0 \quad \frac{\partial \mathcal{H}}{\partial \mathbf{u}} = 0 \quad (2.20)$$

Le equazioni 2.20 sono la condizione necessaria per l'ottimalità. In un metodo indiretto la seconda equazione di 2.20 viene integrata in avanti per trovare i valori della funzione moltiplicatore, mentre la prima viene integrata all'indietro per trovare di quanto devono essere aggiustate le condizioni iniziali e il controllo in modo da raffinare la soluzione sub-ottima che è stata fornita a priori dell'esecuzione dell'algoritmo.

Una soluzione che le soddisfa sarà in un punto stazionario. Per essere rigorosi, dobbiamo anche mostrare che la variazione seconda sia non negativa, che implica che la soluzione è ad un minimo (ovvero è all'opposto ad un massimo o ad un punto di sella). Un simile calcolo va oltre gli scopi di questa tesi, ma viene trattato nel libro di Bryson e Ho [4]. Non lo trattiamo perché la maggior parte della letteratura disponibile si ferma alla valutazione della condizione necessaria, e quindi trova un possibile ottimo (ovvero un ottimo locale).

Il prossimo passo è di calcolare il valore degli elementi del vettore delle funzioni moltiplicatore, modificando la prima equazione di 2.20.

$$-\dot{\boldsymbol{\lambda}}^T = \frac{\delta \mathcal{H}}{\delta \mathbf{q}} \quad (2.21)$$

$$\dot{\boldsymbol{\lambda}} = - \left(\frac{\partial \mathcal{L}}{\partial \mathbf{q}} \right)^T - \left(\frac{\partial \mathbf{f}}{\partial \mathbf{q}} \right)^T \boldsymbol{\lambda} \quad (2.22)$$

Ora possiamo valutare l'equazione 2.22 per il nostro problema.

$$\begin{bmatrix} \dot{\lambda}_x \\ \dot{\lambda}_v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_x \\ \lambda_v \end{bmatrix} \quad (2.23)$$

Il sistema di equazioni rappresentato da 2.23 è lineare, e pertanto possiamo ottenere facilmente una soluzione. Nella seguente espressione, c_0 e c_1 sono costanti di integrazione e il tempo è dato da t .

$$\lambda_x = c_0 \quad \lambda_v = c_1 - c_0 t \quad (2.24)$$

Una volta che sappiamo il valore delle funzioni moltiplicatore, possiamo tornare indietro e risolvere la funzione di controllo usando la seconda equazione di 2.20.

$$\begin{aligned} 0 &= \frac{\partial \mathcal{H}}{\partial \mathbf{u}} \\ &= \frac{\partial}{\partial \mathbf{u}} (u^2 + \lambda_x v + \lambda_v u) \\ &= 2u + 0 + (c_1 - c_0 t) \end{aligned} \quad (2.25)$$

Da cui otteniamo la funzione di controllo

$$u = \frac{c_0 t - c_1}{2} \quad (2.26)$$

Ora possiamo usare la dinamica del sistema per ottenere le espressioni per la posizione e velocità come funzioni di tempo.

$$\begin{aligned} v &= \int u(\tau) d\tau = \frac{1}{4} c_0 t^2 - \frac{1}{2} c_1 t + c_2 \\ x &= \int v(\tau) d\tau = \frac{1}{12} c_0 t^3 - \frac{1}{4} c_1 t^2 + c_2 t + c_3 \end{aligned} \quad (2.27)$$

Per trovare le costanti di integrazione c_i , costruiamo un sistema lineare a partire dalle condizioni di frontiera.

$$\begin{bmatrix} x(0) \\ v(0) \\ x(1) \\ v(1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \frac{1}{12} & -\frac{1}{4} & 1 & 1 \\ \frac{1}{4} & -\frac{1}{2} & 1 & 0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad (2.28)$$

Risolvendo il sistema lineare e sostituendo i coefficienti trovati, otteniamo le espressioni per lo stato e il controllo.

$$\begin{aligned}
x(t) &= -2t^3 + 3t^2 \\
v(t) &= -6t^2 + 6t \\
u(t) &= -12t + 6
\end{aligned} \tag{2.29}$$

Soluzione semplificata

In letteratura vengono spesso fatte delle semplificazioni al problema della traiettoria ottima. Queste sono generalmente dettate dalla natura del problema, qualora questo concede trattazioni più facili del metodo analitico visto sopra. Un esempio di un approccio semplificato al calcolo della traiettoria ottima viene dato in questa sezione.

Per il problema del blocco in movimento può essere ottenuta anche una soluzione più breve, poiché in questo caso abbiamo che il controllo u è la derivata seconda della posizione x .

$$J = \int_0^1 u^2(\tau) d\tau = \int_0^1 \ddot{x}^2(\tau) d\tau \tag{2.30}$$

In questo caso otteniamo la Lagrangiana:

$$\mathcal{L}(t, x, \dot{x}, \ddot{x}) = \mathcal{L}(\ddot{x}) = \ddot{x}^2 \tag{2.31}$$

Per una soluzione rigorosa, dovremmo dimostrare che la variazione prima della funzione obiettivo è nulla, mentre la variazione seconda è non negativa. Ci concentriamo sulla variazione prima, che è la condizione necessaria per la soluzione ottima x^* . La successiva equazione è costruita usando integrazione per parti.

$$\frac{\partial \mathcal{L}}{\partial x^*} - \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{x}^*} - \frac{d^2}{dt^2} \frac{\partial \mathcal{L}}{\partial \ddot{x}^*} = 0 \tag{2.32}$$

I primi due termini sono nulli, poiché \mathcal{L} dipende solo da \ddot{x} , il termine finale può essere semplificato e scritto come:

$$\frac{d^2}{dt^2} (2\ddot{x}^*) = \frac{d^4}{dt^4} x^* = 0 \tag{2.33}$$

La soluzione a questa equazione è un polinomiale cubico con quattro coefficienti sconosciuti, identici a quelli dell'equazione 2.27. Troviamo i coefficienti usando le condizioni di frontiera 2.12.

$$x(t) = -2t^3 + 3t^2 \quad (2.34)$$

2.1.2 Metodi diretti

La maggior parte dei metodi per la risoluzione dei problemi di ottimizzazione della traiettoria possono essere classificati in **diretti** e **indiretti**. In questa tesi ci concentriamo sull'analisi dei metodi indiretti per ottenere una soluzione matematicamente esatta anche se computazionalmente onerosa. Decidiamo di seguire questo approccio nonostante nella maggior parte della letteratura che tratta l'ottimizzazione della traiettoria usa metodi diretti. Le architetture moderne dispongono di una potenza di calcolo che, a differenza di quanto avveniva nel passato, è in grado di ottenere un risultato in tempo soddisfacente. Inoltre un approccio simile per i quadricotteri risulta molto vantaggioso, in quanto il sistema dinamico che li descrive, è simile al sistema dinamico per cui sono nati i metodi indiretti: il volo di un aeromobile. L'aspetto chiave di un metodo diretto è che effettua una discretizzazione del problema di ottimizzazione della traiettoria stesso, normalmente con la trasformazione del problema di ottimizzazione della traiettoria originale in un programma non lineare. Questo processo di conversione è noto con il termine **trascrizione**, ed è anche il motivo per cui in letteratura troviamo spesso metodi diretti di trascrizione invece di metodi diretti di collocazione.

In generale, i metodi diretti di collocazione sono in grado di discretizzare un problema di ottimizzazione della traiettoria continuo mediante approssimazione di tutte le funzioni continue del problema in delle spline polinomiali. Una **spline** è una funzione fatta di una sequenza di segmenti polinomiali. Vengono usati i polinomi poiché presentano due importanti proprietà: possono essere rappresentati da un piccolo (e finito) insieme di coefficienti, ed è facile calcolare integrali e derivate di polinomiali in termini di questi coefficienti.

La maggior parte dei metodi di collocazione diretti trascrive un problema di ottimizzazione della traiettoria a tempo-continuo in un programma non lineare. Un programma non lineare è il nome speciale dato ad un problema di ottimizzazione con parametri vincolati che ha termini non lineari sia nella funzione obiettivo che nella funzione dei vincoli. Una formulazione tipica per un programma non lineare è data di seguito.

$$\min_z J(z) \quad (2.35)$$

Soggetto a seguenti vincoli

$$\begin{aligned} f(z) &= 0 \\ g(z) &\leq 0 \\ z_{\text{low}} &\leq z \leq z_{\text{upp}} \end{aligned} \quad (2.36)$$

In alcuni casi, un metodo di collocazione diretto produrrà un programma lineare o un programma quadratico invece di uno non lineare. Questo accade quando i vincoli (che includono anche la dinamica del sistema) sono lineari e la funzione obiettivo è lineare (si avrà un programma lineare) o quadratica (si avrà un programma quadratico). Sia i programmi lineari che quadratici sono molto più facili da risolvere che i programmi non-lineari. Molto spesso si tende ad approssimare dei programmi non-lineari trovando un intervallo di funzionamento in cui questi hanno un andamento quasi lineare. Questo accade specialmente in robotica per applicazioni di tipo real-time.

2.1.3 Metodi indiretti

I metodi diretti discretizzano il problema di ottimizzazione della traiettoria, convertendolo in un problema non lineare. C'è un altro insieme di metodi di risoluzione in letteratura, noti come metodi indiretti. I metodi indiretti costruiscono analiticamente la condizione necessaria e sufficiente per l'ottimalità. Solo successivamente discretizzano le condizioni e le risolvono numericamente. Un modo comune per distinguere le principali classi per la risoluzione dei problemi di ottimizzazione è il seguente:

- metodi diretti prima discretizzano e poi ottimizzano

- metodi indiretti prima ottimizzano e poi discretizzano

Consideriamo un problema scalare di ottimizzazione per mostrare come funzionano i metodi indiretti: minimizzazione di $y = f(t)$. Avremo il valore minimo $y^* = f(t^*)$ quando la derivata è nulla $\dot{y}(t^*) = 0$. Inoltre, dobbiamo verificare che la curvatura sia positiva $\ddot{y}(t^*) > 0$, in modo da assicurarci che abbiamo un minimo locale invece che un massimo o un punto di sella. Quando sono vere entrambe le condizioni, sappiamo che $y^* = f(t^*)$ è un minimo locale. Un'ottimizzazione indiretta funziona con lo stesso principio, ma le condizioni sono leggermente più difficili da costruire e da risolvere. Al contrario, un metodo diretto minimizza $y(t)$ costruendo una sequenza di prove tali che ogni nuova prova è migliore della precedente: $y(t_0) > y(t_1) > \dots > y(t^*)$ [10].

Il maggior vantaggio di un metodo indiretto, quando confrontato con uno diretto, è che generalmente un metodo indiretto sarà più accurato ed avrà una stima d'errore più affidabile. Entrambi i vantaggi derivano dall'espressione analitica per la condizione necessaria e sufficiente che viene derivata mentre si costruisce il problema. Ci sono molte difficoltà associate con metodi indiretti quando questi sono comparati alla loro controparte diretta. Per esempio, la regione di convergenza tende ad essere più piccola per i metodi indiretti che per quelli diretti, il che significa che un metodo indiretto necessita di una inizializzazione migliore [7]. L'inizializzazione è ancora più difficile tenendo a mente il fatto che i metodi indiretti necessitano di inizializzare anche le variabili aggiunte che non vengono usate in un metodo diretto [10]. Infine, per ottenere una soluzione accurata per un metodo indiretto, è solitamente necessario costruire la condizione necessaria e sufficiente analiticamente, che può risultare molto complesso [7].

Metodo di shooting singolo

Come avviene con il metodo diretto di collocazione, il metodo diretto di shooting singolo (noto anche come shooting singolo), risolve il problema di ottimizzazione della traiettoria trasformandolo in un programma non lineare. La differenza fondamentale è che shooting singolo approssima la traiettoria usando una simulazione. Le variabili di decisione nel programma non lineare

sono una parametrizzazione ad anello aperto del controllo lungo la traiettoria, così come lo stato iniziale. È un metodo molto utile per applicazioni dove il controllo è semplice e ci sono pochi vincoli sul cammino. Così come avviene per la traiettoria di un quadricottero analizzata in questa tesi. Qualora il controllo risulti complicato e/o siano presenti vincoli sul cammino in grande numero, un approccio come quello che usa questo metodo potrebbe risultare computazionalmente inoperabile. Pertanto per alcune classi di problemi specifici, è molto meno oneroso adoperare un metodo diretto di collazione invece del shooting singolo.

Metodo di shooting multiplo

Un'estensione comune del metodo diretto di shooting singolo è il metodo diretto di shooting multiplo (chiamato anche shooting parallelo). Questo metodo, invece di rappresentare tutta la traiettoria come una singola simulazione, la divide in segmenti, e ogni segmento è rappresentato da una simulazione. Shooting multiplo tende ad essere molto più robusto che quello singolo, e dunque è usato per problemi più difficili.

Se confrontiamo shooting multiplo ai metodi di collocazione, esso tende a creare programmi non lineari piccoli ma densi, con poche variabili di decisione che sono più accoppiate. La difficoltà maggiore con i metodi di shooting sta nel fatto che è difficile implementare vincoli di cammino, poiché le variabili di stato intermedie non sono delle variabili di decisione nel programma non lineare [7]. Inoltre in alcuni casi si può avere che la relazione tra le variabili di decisione e i vincoli è altamente non lineare, che può causare una bassa convergenza [7, 10].

Capitolo 3

Dinamica del quadricottero

In questo capitolo introduciamo le nozioni base che servono a comprendere la struttura dinamica di un quadricottero e come avviene il suo spostamento nello spazio. La posizione e i gradi di libertà dei propulsori in un veicolo aereo determinano la dinamica e la cinematica dello stesso. Per definire i quadricotteri e analizzare i problemi che li concernono, nonché come questi vengono gestiti in questa tesi, analizzeremo brevemente il caso dei loro parenti generici, gli elicotteri.

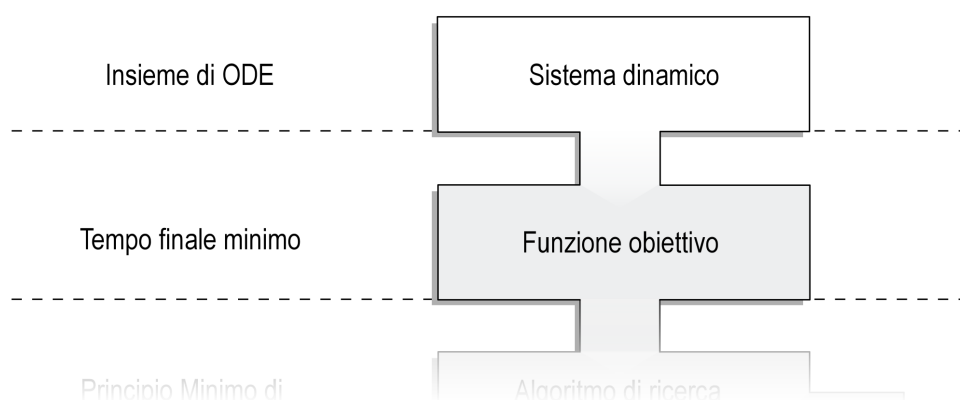


Figura 3.1: Schema concettuale della rappresentazione dinamica del quadricottero per la libreria della generazione della traiettoria ottima

3.1 Introduzione

Un elicottero è un veicolo aereo che usa rotori che girano velocemente per muovere l'aria verso il basso, e che quindi creano una spinta tale da mantenere l'elicottero in aria. Gli elicotteri convenzionali hanno due rotori. Essi vengono posizionati in modalità *complanare* dando entrambi la spinta verso l'alto, ruotando in direzioni opposte (in modo da bilanciare i momenti angolari esercitati sul corpo del elicottero). I due rotori possono anche essere posizionati con un rotore principale che fornisce la spinta e un rotore più piccolo laterale che controbilancia il momento del rotore principale. Tuttavia, queste configurazioni hanno bisogno di un controllo complicato per variare la direzione del moto; un **piatto oscillante** è usato per cambiare l'angolo di attacco sui rotori principali. Il modello con cui sono implementati i rotori, e il meccanismo del piatto oscillante, presentano alcuni problemi, incrementando il costo di produzione e la complessità dello sviluppo.

Un elicottero a quattro eliche, o un quadricottero, è un elicottero che ha quattro rotori piazzati allo stesso modo, normalmente posizionate agli angoli di una struttura a corpo squadrato. Con quattro rotori indipendenti non c'è il bisogno di usare un piatto oscillante: esso era necessario per consentire all'elicottero di utilizzare più gradi di libertà, ma lo stesso livello di controllo può essere ottenuto aggiungendo altri due rotori.

Lo sviluppo dei quadricotteri è rimasto bloccato fino a non molto tempo fa, poiché controllare quattro rotori indipendenti si è dimostrato molto difficile nonché impossibile senza l'ausilio di elettronica a bordo. Il costo decrescente dei microprocessori ha consentito, negli ultimi anni, la reintroduzione dello sviluppo dei quadricotteri per scopi commerciali, militari e anche privati. Le applicazioni military rappresentano attualmente la parte fondamentale per il mercato degli UAV, e questo settore sta crescendo considerevolmente [11].

Il controllo dei quadricotteri è un problema difficile e interessante. Con sei gradi di libertà (tre traslazionali e tre rotazionali) e solo quattro input indipendenti (la velocità dei rotori), i quadricotteri sono dei sistemi molto sottoattuati. In modo da raggiungere i sei gradi di libertà, i movimenti rotazionali e traslazionali sono accoppiati. La dinamica risultante è altamente

non lineare, soprattutto tenendo a mente gli effetti aerodinamici in gioco. Infine, a differenza dei veicoli a terra, gli elicotteri hanno poco attrito che li impedisce il moto, quindi devono introdurre un meccanismo di frenata autonomo in modo da fermare il movimento e rimanere in posizione stabile. Insieme questi fattori creano un problema di controllo molto interessante.

Il sistema di riferimento di un UAV può essere definito con l'asse x allineata con la fusoliera, asse y che punta attraverso l'ala destra e l'asse z che punta verso il basso, completando un sistema ortonormale. L'assetto (o orientazione) del sistema di riferimento può essere relativo ad un frame inerziale usando una matrice di rotazione. La matrice di rotazione può essere parametrizzata usando ad esempio i quaternioni, i parametri di Rodrigues o gli angoli di Eulero. Per veicoli aerei, l'uso degli **angoli di Eulero** per gli angoli di roll, pitch e yaw è già ben consolidato e facile da visualizzare [12]:

1. l'angolo di **roll** è definito come una rotazione attorno all'asse x
2. l'angolo di **pitch** come una rotazione attorno all'asse y
3. l'angolo di **yaw** come una rotazione attorno all'asse z

Attraverso queste tre rotazioni, l'assetto del sistema di riferimento può essere messo in relazione con un frame inerziale.

Come già accennato, un quadricottero ha quattro rotori usati sia per controllo rotazionale che traslazionale. Può quindi cambiare l'assetto variando la velocità di rotazione di rotori. Una delle principali sfide con i quadricotteri è che hanno solo quattro attuatori: ovvero hanno meno attuatori che gradi di libertà. Con meno attuatori che gradi di libertà, abbiamo un problema di controllo sottoattuato (che verrà trattato in sezione 3.2). Molti approcci diversi sono stati implementati nel tempo, come la linearizzazione del feedback, backstepping, modalità di sliding e il controllo del PID. Quello che accomuna tutti, è che deve essere trovata una relazione tra la dinamica rotazionale e traslazionale in modo da trovare il vettore posizionale. Questo può essere fatto direttamente nel assetto desiderato, che viene solitamente fatto per schemi di tip way-point guidance, o può essere fatto come parte del controllore.

3.2 Sottoattuazione

In generale, la dinamica di un corpo rigido può essere decomposta in dinamica traslazionale e dinamica rotazionale. La dinamica rotazionale descrive i cambiamenti in orientamento e nella velocità angolare, mentre la dinamica traslazionale descrive i cambiamenti in posizione e velocità relativi ad un frame inerziale. Un corpo rigido libero in volo ha sei gradi di libertà. Può traslare e ruotare lungo i tre assi. In caso di un corpo rigido pienamente attuato, il numero di attuatori è pari ai gradi di libertà, in modo che il problema di controllo possa essere risolto attraverso l'utilizzo delle tecniche basate sulla soluzione delle equazioni di Eulero-Lagrange [13–20]. Nel caso di un corpo rigido con meno attuatori che gradi di libertà, il corpo rigido è detto sottoattuato [21]. Questo significa che il veicolo non può seguire una traiettoria arbitraria, ma deve muoversi secondo i vincoli dei propri attuatori. Per esempio, un aereo deve volare in moto circolare per raggiungere una posizione che inizialmente era alla sua sinistra. Non può spostarsi direttamente nei lati ¹.

Il fatto che un UAV con configurazione a eliche fisse, e un aereo convenzionale, sono sottoattuati è menzionato raramente quando vengono sviluppate le leggi di controllo per la dinamica del volo. Essi condividono vincoli simili insieme a veicoli autonomi subacquatici o AUV (e che invece comunemente sono definiti come sottoattuati). Per sottolineare questo punto, consideriamo la configurazione degli attuatori di un UAV a eliche fisse. Ha quattro attuatori: spinta per il controllo traslazionale e tre controlli per il controllo rotazionale; e ha sei gradi di libertà. Con un numero inferiore di attuatori che gradi di libertà, un UAV a eliche fisse è sottoattuato. Questo comporta una difficoltà in più nella progettazione del controllo. A causa di questo sottoattuamento, non è possibile controllare tutti i gradi di libertà direttamente. Invece, è possibile mappare gli stati desiderati da sottoattuati ad attuati. Ciò può essere fatto usando dei metodi noti: è lo scopo del sistema di guida, che è il componente standard in qualsiasi sistema di volo. Consideriamo il problema del controllo di altitudine. L'unico modo per generare una

¹Un aereo può presentare piccoli cambiamenti nelle direzioni laterali attraverso le forze aerodinamiche, ma queste forze sono troppo piccole per essere usate per controllo laterale

spinta in alto è mediante il cambio degli angoli di attacco e di conseguenza l'orientazione. Per questa ragione, l'errore sull'altezza può essere mappato ad un angolo di attacco desiderato che può essere gestito usando l'equilibratore (parte finale dell'ala di un aeromobile) per generare momenti aerodinamici che allineano l'angolo di attacco ad un valore desiderato. Se il veicolo fosse stato pienamente attuato, sarebbe stato sufficiente generare una forza verso l'alto per ottenere l'altitudine desiderata, invece di usare l'angolo di attacco come variabile di stato.

Tenendo a mente il fatto che veicolo UAV ad elica fissa sono sottoattuati, possiamo applicare i risultati da altri sistemi a corpo rigido. Il controllo di sistemi sottoattuati è stato studiato per molti sistemi dinamici diversi, i.e. navi, AUV, quadricotteri, aerei a decollo e atterraggio verticali, veicoli spaziali, aeroscafi, e altri sistemi [22–42,42,43]. Le soluzioni per il controllo di questi sistemi hanno permesso di studiare la sottoattuazione, nonché di usarle come punti di riferimento per la progettazione dei rispettivi controllori. La formulazione generale per un sistema dinamico di secondo ordine può essere scritta come:

$$\ddot{\mathbf{q}} = \mathbf{f}_1(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{f}_2(\mathbf{q}, \dot{\mathbf{q}}, t)\mathbf{u} \quad (3.1)$$

Dove \mathbf{q} è un vettore delle posizioni, $\dot{\mathbf{q}} := \frac{d}{dt}\mathbf{q}$ è un vettore delle velocità, t è il tempo e \mathbf{u} è il controllo.

Definition 1. *Un sistema di controllo descritto dall'equazione 3.1 è **pienamente attuato** nella configurazione $(\mathbf{q}, \dot{\mathbf{q}}, t)$ se è in grado di imporre istantaneamente un'accelerazione in una direzione arbitraria in \mathbf{q} :*

$$\text{rank} f_2(\mathbf{q}, \dot{\mathbf{q}}, t) = \dim(\mathbf{q}) \quad (3.2)$$

□

Definition 2. *Un sistema di controllo descritto dall'equazione 3.1 è **sottoattuato** nella configurazione $(\mathbf{q}, \dot{\mathbf{q}}, t)$ se non è in grado di imporre un'accelerazione istantanea in una direzione arbitraria in \mathbf{q} :*

$$\text{rank} f_2(\mathbf{q}, \dot{\mathbf{q}}, t) < \dim(\mathbf{q}) \quad (3.3)$$

□

Dal punto di vista di controllo, questo significa che un veicolo pienamente attuato è in grado di eseguire qualsiasi traiettoria, anche una traiettoria che non è possibile per un veicolo sottoattuato. È possibile far seguire una traiettoria "non convenzionale" anche ad un veicolo sottoattuato imponendo certi vincoli sulla traiettoria stessa; ma per l'esecuzione della traiettoria, l'errore degli stati non attuati deve essere mappato a quelli attuati. Per un veicolo aereo questo significa che gli errori sulla posizione desiderata e la velocità relativa ad una traiettoria devono essere mappati ad una velocità desiderata, orientazione, velocità angolare e accelerazione. Questi stati desiderati possono essere eseguiti usando gli attuatori disponibili.

3.3 Modello dinamico

Il quadricottero è descritto con sei gradi di libertà. La posizione traslazionale (x, y, z) è misurata nel sistema di coordinate inerziale O come mostrato in figura 3.2. L'assetto del veicolo V è definita con la matrice di rotazione ${}^O_V R$. La matrice di rotazione è definita in modo tale che, quando viene moltiplicato un vettore \mathbf{v} nel sistema delle coordinate del veicolo V con essa, lo stesso vettore, descritto nel sistema di coordinate inerziale O , viene ottenuto come segue.

$${}^O_V \mathbf{v} = {}^O_V R {}^V_V \mathbf{v} \quad (3.4)$$

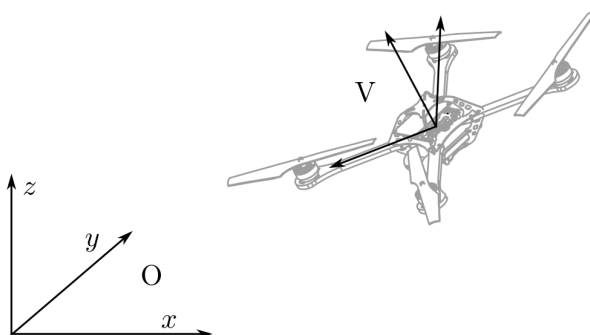


Figura 3.2: Il sistema di coordinate inerziale O , e il sistema di coordinate del veicolo V

Gli input di controllo di un quadricottero sono i tassi rotazionali desiderati sulle assi del sistema di riferimento del veicolo ω_x , ω_y e ω_z e la spinta a normalizzata per la massa m , così come mostrato nella figura 1.2.

Controllori ad alta banda a bordo del veicolo tracciano i tassi rotazionali desiderati usando il feedback proveniente da un giroscopio. Il quadricottero ha un'inerzia rotazionale molto bassa, e può produrre momenti alti a causa della configurazione esterna dei propulsori. Questo ha come risultato una possibile accelerazione rotazionale molto alta, $\dot{\omega}_x$ e $\dot{\omega}_y$, dell'ordine di 200 rad/s². Il veicolo ha una risposta molto rapida a variazioni nel tasso rotazionale (i risultati sperimentali hanno mostrato costanti di tempo dell'ordine di 20 ms per variazioni che non saturano i motori). Assumiamo quindi che possiamo controllare direttamente i tassi rotazionali ignorando la dinamica dell'accelerazione di cui una rotazione è composta. L'accelerazione rotazionale $\dot{\omega}_z$ è prodotta da una differenza di attrito tra i propulsori che ruotano in direzioni opposte. L'accelerazione raggiungibile da $\dot{\omega}_z$ è considerevolmente inferiore alle accelerazioni $\dot{\omega}_x$ e $\dot{\omega}_y$, ed è dell'ordine di 19 rad/s². Tuttavia mostreremo che ω_z non influenza di molto la dinamica del veicolo.

Così come accade per i tassi rotazionali del veicolo, assumiamolo che la spinta può essere variata istantaneamente. I risultati sperimentali hanno mostrato che la dinamica della spinta è all'incirca uguale alla dinamica rotazionale: i propulsori ruotano verso l'alto con una velocità maggiore rispetto alla rotazione verso il basso.

Assumiamo anche che tutti gli input di controllo sono soggetti a saturazione. La quantità dei tassi rotazionali del veicolo è limitata (tale limitazione può essere causata, ad esempio, dai limiti del giroscopio, o dalle limitazioni dei tracker dei tassi rotazionali). La spinta complessiva è limitata da un valore massimo e da uno minimo.

$$a_{\min} \leq a \leq a_{\max} \quad (3.5)$$

Con $a_{\min} > 0$. Questa limitazione è motivata dal fatto che tipicamente i quadricottero hanno propulsori di tipo fixed-pitch, ovvero non può essere variata l'imbardata: i propulsori non sono in grado di variare la direzione della rotazione in volo.

3.3.1 Equazioni del moto

L'accelerazione traslazionale del veicolo è dettata dall'assetto del veicolo e dalla spinta totale prodotta dai quattro propulsori. L'accelerazione traslazionale nel frame inerziale può essere espressa con la seguente equazione.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = {}^O_V R \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad (3.6)$$

Inoltre, il cambiamento nell'assetto del veicolo è in relazione agli input di controllo rotazionali [44]:

$${}^O_V \dot{R} = {}^O_V R \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} + \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (3.7)$$

3.4 Input di controllo

Calcoliamo gli input di controllo per una data traiettoria tenendo a mente gli effetti dell'input di controllo sulla fattibilità delle traiettorie. La traiettoria generata da un algoritmo di generazione deve soddisfare le equazioni del moto 3.6 e 3.7: gli input di controllo non possono eccedere i valori ammissibili.

3.4.1 Input di controllo per una data traiettoria

Sia $(x(t), y(t), z(t))$ una possibile traiettoria del veicolo. Per comodità di notazione, omettiamo le dipendenze temporali da qui in poi in questo paragrafo. Prendendo la seconda derivata della traiettoria e combinandola con le equazioni traslazionali del moto 3.6, introduciamo il vettore \mathbf{f} che rappresenta le forze totali normalizzate per la massa richieste dal quadricottero per eseguire una data traiettoria.

$$\mathbf{f} := \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = {}^O_V R \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} \quad (3.8)$$

Usando la norma euclidea (denotata da $\|\cdot\|$), la spinta a richiesta per seguire la traiettoria può essere calcolata come segue.

$$\|\mathbf{f}\| := \left\| {}^O_V R \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} \right\| = a \quad (3.9)$$

Poiché la norma euclidea di un vettore non viene variata dalla matrice di rotazione. La direzione della spinta può essere valutata come un semplice prodotto tra il vettore delle forze totali normalizzate per la massa \mathbf{f} e la sua norma.

$$\bar{\mathbf{f}} := \frac{\mathbf{f}}{\|\mathbf{f}\|} \quad (3.10)$$

Di conseguenza possiamo trovare la direzione semplicemente sostituendo le espressioni per \mathbf{f} e $\|\mathbf{f}\|$ e ottenendo:

$${}^O_V R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \bar{\mathbf{f}} \quad (3.11)$$

Svolgendo la derivata dell'espressione sopra e combinandola con l'equazione 3.7, otteniamo due tassi rotazionali del veicolo in funzione dell'assetto corrente e di $\dot{\bar{\mathbf{f}}}$.

$$\begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} = {}^V_O R \dot{\bar{\mathbf{f}}} \quad (3.12)$$

L'equazione 3.9 e le prime due righe dell'equazione 3.12 forniscono tre equazioni per i quattro controlli di input sconosciuti. Per determinare le traiettorie associate ai controlli di input è tuttavia necessario specificare anche una quarta equazione, che specifica il tasso di rotazione ω_z . Questo, per quanto visto nel paragrafo 3.3, può essere definito come una costante, per esempio può essere definito come $\omega_z = 0$.

3.4.2 Condizioni di fattibilità

Possiamo calcolare i vincoli di fattibilità per la traiettoria dai vincoli sugli input di controllo.

La **spinta complessiva** viene ottenuta dalle equazioni 3.8 e 3.9: chiaramente anch'essa deve sottostare ad un intervallo massimo e minimo. Per il calcolo di questo intervallo è sufficiente fare riferimento alle tre componenti che descrivono il vettore del moto traslazionale.

$$a_{\min} \leq \|\mathbf{f}\| = \sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} + g)^2} \leq a_{\max} \quad (3.13)$$

I **tassi rotazionali** ω_x e ω_y possono essere eseguiti come input del controllo se è presente un vincolo su ω_z in aggiunta alla traiettoria. Tuttavia possono essere vincolati usando la proprietà della norma unitaria della matrice di rotazione con l'equazione [45]:

$$\omega_{x,y} \leq \left\| \dot{\mathbf{f}} \right\| \quad (3.14)$$

3.5 Modello dinamico vincolato

Abbiamo modellato le equazioni di un quadricottero come un corpo rigido libero di muoversi rispetto ad ognuno dei suoi sei gradi di libertà. sotto l'azione della gravità e delle forze esterne (del controllo), e successivamente gli abbiamo impostato un'altezza prefissata costante a h mantenendo fisso il valore ω_z .

Sia $T = t_x, t_y, t_z$ un sistema di coordinate inerziale orientato in accordo alla convenzione North-East-Down (NED), e sia $E = (e_1, e_2, e_3)$ il sistema di coordinate del corpo, orientato in accordo con le assi inerziali del quadricottero. Usiamo la convenzione NED degli angoli di Eulero per l'assetto del quadricottero [46] con (ψ, θ, ϕ) gli angoli di yaw, pitch e roll ottenuti da una rotazione di $e_3 e_2 e_1$.

Lo spazio degli stati del quadricottero è definito con $\mathbb{R}^3 \times \text{SO}(3) \times \mathbb{R}^3 \times \mathbb{R}^3$, con le coordinate $(\mathbf{q}, \dot{\mathbf{q}}) = (x, y, z, \psi, \theta, \phi, \dot{x}, \dot{y}, \dot{z}, \Omega_1, \Omega_2, \Omega_3)$, dove $(x, y, z) \in \mathbb{R}^3$ sono le coordinate del centro di massa del quadricottero rispetto al sistema di riferimento T , $(\dot{x}, \dot{y}, \dot{z})$ sono le componenti della velocità del centro di

massa, e $(\Omega_1, \Omega_2, \Omega_3) \in \mathbb{R}$ sono le componenti delle velocità angolari del quadricottero nel suo sistema di riferimento E. Con queste assunzioni la lagrangiana del sistema risulta come segue.

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + \frac{1}{2}\Omega \mathbb{I}_C \Omega + mgz \quad (3.15)$$

Dove g è l'accelerazione dovuta alla forza di gravità, m denota la massa e \mathbb{I}_C è il tensore dell'inerzia relativo al centro di massa C del quadricottero; i prodotti nell'equazione 3.15 sono semplici prodotti scalari in \mathbb{R}^3 . Nel sistema di riferimento del quadricottero, il tensore d'inerzia è diagonale $\mathbb{I}_C = \text{diag}(I_1, I_2, I_3)$, dove I_1 , I_2 e I_3 sono le componenti diagonali rispetto agli assi principali d'inerzia. Poiché il quadricottero è simmetrico, è richiesto che $I_1 = I_2$. Trasformiamo la Lagrangiana nella corrispettiva hamiltoniana mediante le **trasformate di Legendre**. L'hamiltoniano ha come stato lo stato originale \mathbf{q} e il momento coniugato \mathbf{p} relativo al vettore $\dot{\mathbf{q}}$ e può essere espresso con la seguente equazione.

$$\mathcal{H}(\mathbf{q}, \mathbf{p}) = \frac{1}{2m}(p_x^2 + p_y^2 + p_z^2) + \frac{1}{2}\mathbb{I}_C^{-1}MM - mgz \quad (3.16)$$

Dove M è il momento angolare nel sistema di riferimento del quadricottero, coniugato con Ω rispetto al centro di massa C . Siano $(F_x, F_y, F_z, \tau_1, \tau_2, \tau_3) \in \mathbb{R}^3 \times \mathbb{R}^3$ i controlli che agiscono sul quadricottero, dove vale:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = R \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix} \quad (3.17)$$

Con u una funzione di controllo e R la matrice di trasformazione che rappresenta l'assetto del quadricottero.

$$R = \begin{bmatrix} \cos \theta \cos \psi & \cos \psi \sin \theta \sin \varphi - \sin \psi \cos \varphi & \cos \psi \sin \theta \sin \varphi + \sin \psi \cos \varphi \\ \cos \theta \cos \psi & \sin \psi \sin \theta \sin \varphi + \cos \psi \cos \varphi & \cos \varphi \sin \theta \sin \psi + \cos \psi \sin \varphi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{bmatrix} \quad (3.18)$$

Per semplicità assumiamo di avere un modello lineare dell'attrito dell'aria, in questo modo la forza d'attrito è proporzionale alla velocità. Siano κ_x , κ_y

e κ_z i coefficienti d'attrito. Grazie alla simmetria planare del quadricottero, possiamo altrettanto assumere che κ_x e κ_y sono uguali.

Con queste assunzioni le equazioni hamiltoniane per i quadricottero possono essere espresse nella seguente forma.

$$\dot{x} = \frac{p_x}{m} \quad \dot{y} = \frac{p_y}{m} \quad \dot{z} = \frac{p_z}{m} \quad (3.19)$$

$$\dot{p}_x = \frac{\kappa_x}{m} p_x + F_x \quad \dot{p}_y = \frac{\kappa_x}{m} p_y + F_y \quad \dot{p}_z = -mg + \frac{\kappa_z}{m} p_z + F_z \quad (3.20)$$

$$\begin{aligned} M_1 &= I_1(\dot{\varphi} - \dot{\psi} \sin \theta) & \dot{M}_1 &= \frac{M_2 M_3}{I_1 I_3} (M_3 - M_2) = \tau_1 \\ M_2 &= I_1(\dot{\psi} \cos \theta \sin \varphi + \dot{\theta} \sin \varphi) & \dot{M}_2 &= I_1(\dot{\psi} \cos \theta \sin \varphi + \dot{\theta} \cos \varphi) = \tau_2 \\ M_3 &= I_3(\dot{\psi} \cos \theta \cos \varphi + \dot{\theta} \sin \varphi) & \dot{M}_3 &= \tau_3 \end{aligned} \quad (3.21)$$

Ora imponiamo il vincolo in modo che il quadricottero non possa ruotare attorno alla direzione z e che mantenga un'altezza fissa. Impostando un'altezza fissa, \dot{z} e \dot{p}_z spariscono portando a:

$$u = -\frac{mg}{\cos \theta \cos \varphi} \quad (3.22)$$

Imponendo che il quadricottero non possa ruotare attorno all'asse z , chiediamo che il termine \dot{M}_3 sparisca, quindi M_3 è costante. Le equazioni hamiltoniane per il sistema vincolato diventano:

$$\dot{x} = \frac{p_x}{m} \quad \dot{y} = \frac{p_y}{m} \quad (3.23)$$

$$\dot{p}_x = \frac{\kappa_x}{m} p_x + \frac{mg}{\cos \theta} \sin \theta \quad \dot{p}_y = \frac{\kappa_x}{m} p_y - \frac{mg}{\cos \varphi \cos \theta} \sin \varphi \quad (3.24)$$

$$\begin{aligned} M_1 &= I_1 \dot{\varphi} & \dot{M}_1 + \frac{M_2 M_3}{I_1 I_3} (M_3 - M_2) &= \tau_1 \\ M_2 &= I_1 \dot{\theta} \cos \varphi & \dot{M}_2 + \frac{M_1 M_3}{I_1 I_3} (M_1 - M_3) &= \tau_2 \end{aligned} \quad (3.25)$$

Se imponiamo un altro vincolo, ovvero che il quadricottero non possa ruotare attorno alla direzione y , avremo a che fare con un sistema con due

gradi di libertà $(x, \theta) \in \mathbb{R} \times]0, \pi[$, dove come sopra, x è la coordinata del centro di massa, e θ è l'angolo pitch. L'equazione Hamiltoniana per questo sistema diventa:

$$\dot{x} = \frac{p_x}{m} \quad \dot{\theta} = \frac{p_\theta}{I_3(1 + \cos^2 \theta)} \quad (3.26)$$

$$\dot{p}_x = -mg \tan \theta + \frac{\kappa_x}{m} p_x \quad \dot{p}_\theta = \frac{p_\theta^2 \sin \theta \cos \theta}{I_3(1 + \cos^2 \theta)^2} + \tau_\theta \quad (3.27)$$

Dove p_x e p_θ sono i momenti coniugati rispetto a x e θ .

3.6 Implementazione

L'implementazione della dinamica viene fatta usando le equazioni 3.26 e 3.27. Viene usato il caso semplice di un quadricottero in una dimensione, in quanto in questo modo possiamo simulare il comportamento del passaggio del quadricottero in una dimensione, e vederne il risultato diretto semplicemente analizzando l'output dell'algoritmo. Il sistema dinamico viene rappresentato attraverso una classe della libreria di ottimizzazione implementata in questa tesi. Avendo a che fare con un sistema che usa la derivata prima, deriviamo la classe che rappresenta il sistema dalla super-classe più adeguata (ovvero quella della derivata prima).

Nel capitolo successivo introduciamo i metodi indiretti per l'ottimizzazione della traiettoria. Quindi, possiamo utilizzare la rappresentazione della dinamica nella libreria così come l'abbiamo ottenuta con le equazioni differenziali prime. Varieremo il controllo in base a principi matematici, in particolare il Teorema di Pontryagin e la discesa del gradiente, in modo da minimizzare determinati funzionali costo. Troveremo una traiettoria ottima (locale) mediante la ricerca di condizioni necessarie della minimizzazione del funzionale costo aumentato. La traiettoria del controllo ottimo trovata con questo metodo potrà essere adoperata, dal controllore a bordo del quadricottero, per eseguire una traiettoria migliore (rispetto a determinati parametri desiderati) della traiettoria di partenza.

La libreria implementata in questa tesi è sufficientemente flessibile per consentire di variare il problema dinamico in qualunque modo. In questo

modo non è necessario variare l'algoritmo di ottimizzazione per trovare una traiettoria ottima (a partire da una traiettoria sub-ottima come si vedrà bene nel prossimo paragrafo), ma è sufficiente ridefinire una classe che meglio si adagia al problema fisico in questione. La libreria è pre-fornita con due possibili classi, con le quali derivate si può implementare una grande varietà di problemi fisici.

Se prendiamo in considerazione il problema di ottimizzazione della traiettoria per quadricotteri, possiamo rappresentare l'hamiltoniano della dinamica del sistema con la classe che esprime la derivata prima, mentre la lagrangiana del sistema con la classe che esprime la derivata seconda. Le motivazioni di questa scelta sono evidenti dall'equazione 3.15 e 3.16.

Capitolo 4

Metodi diretti

Nella sezione dell'introduzione di questa tesi abbiamo visto un'anteprima del funzionamento dei metodi di collocazione ampiamente adoperati per la risoluzione dei problemi di ottimizzazione. La principale caratteristica di un metodo di collocazione è il fatto che discretizza il problema di ottimizzazione della traiettoria, normalmente mediante la conversione del problema in un programma non lineare. La discretizzazione di una traiettoria continua avviene per approssimazione di tutte le funzioni continue nel problema in delle funzioni fatte di una sequenza di segmenti polinomiali, ovvero in delle spline. Nel paragrafo 2.1.1 abbiamo elaborato una soluzione analitica al problema di ottimizzazione della traiettoria. La maggioranza dei algoritmi presenti in letteratura utilizza un approccio mediante metodi di collocazione. Prima di introdurre l'approccio adoperato in questa tesi, ovvero un approccio mediante metodi indiretti, vediamo in questo capitolo in dettaglio come funzionano i metodi diretti di collocazione, e come vengono adoperati nel problema della ricerca di una traiettoria ottima che deve soddisfare determinati funzionali costo.

Come visto nel paragrafo 2.1.1, la soluzione analitica viene trovata adoperando il principio del calcolo delle variazioni. La variazione viene trovata a partire dall'Hamiltoniano della dinamica e della funzione costo aumentata con delle funzioni moltiplicatrici. Questi metodi convertono il problema di ottimizzazione originale in un sistema di equazioni differenziali, le quali possono avere una soluzione analitica. I metodi indiretti che sono stati

adoperati in questa tesi, a differenza di quelli diretti, risolvono il problema di ottimizzazione della traiettoria con un principio simile. Analiticamente costruiscono le condizioni necessarie e sufficienti per l'ottimalità, e dopo le risolvono numericamente.

4.1 Collocazione trapezoidale

Vediamo come risolvere il problema di ottimizzazione della traiettoria, rappresentato da un blocco che si sposta tra due punti introdotto nella sezione 1.1, mediante il metodo di collocazione trapezoidale. Dobbiamo prima di tutto convertire il problema a tempo continuo originale in un programma non-lineare. Innanzitutto dobbiamo discretizzare la traiettoria. Questa operazione ci fornirà un insieme finito di variabili di decisione, e viene fatta rappresentando la posizione continua $x(t)$ e velocità $v(t)$ con i loro valori in specifici punti temporali, noti come **punti di collocazione**.

$$\begin{aligned} t &\rightarrow t_0 \dots t_k \dots t_N \\ x &\rightarrow x_0 \dots x_k \dots x_N \\ v &\rightarrow v_0 \dots v_k \dots v_N \end{aligned} \tag{4.1}$$

Successivamente dobbiamo convertire la dinamica del sistema continua in un insieme di vincoli che possiamo applicare allo stato e al controllo nei punti di collocazione. Per questa operazione viene adoperata la **regola del trapezio**. L'idea chiave è che il cambiamento nello stato tra due punti di collocazione è uguale all'integrale della dinamica del sistema. L'integrale viene approssimato con la regola del trapezio, come mostrato di seguito, dove $h_k \equiv (t_{k+1} - t_k)$.

$$\begin{aligned} \dot{x} &= v \\ \int_{t_k}^{t_{k+1}} \dot{x} dt &= \int_{t_k}^{t_{k+1}} v dt \\ x_{k+1} - x_k &\approx \frac{h_k}{2} (v_{k+1} + v_k) \end{aligned} \tag{4.2}$$

Semplificando e applicando il procedimento all'equazione della velocità, arriviamo ad un insieme di equazioni che ci permettono di approssimare la dinamica tra ogni coppia di punti di collocazione. Queste equazioni sono note come **vincoli di collocazione**. Le equazioni valgono in ogni segmento $k = 0 \dots (N - 1)$ della traiettoria.

$$x_{k+1} - x_k = \frac{h_k}{2} (v_{k+1} + v_k) \quad (4.3)$$

$$v_{k+1} - v_k = \frac{h_k}{2} (u_{k+1} + u_k) \quad (4.4)$$

Le condizioni di frontiera sono facili da gestire in questo modo: le applichiamo semplicemente allo stato nei punti di collocazione iniziali e finali.

$$\begin{aligned} x_0 &= 0 & x_N &= 1 \\ v_0 &= 0 & v_N &= 0 \end{aligned} \quad (4.5)$$

Infine, dobbiamo approssimare la funzione obiettivo usando la regola del trapezio, convertendola in una sommatoria sulla forza del controllo in ogni punto di collocazione. Qualora la funzione obiettivo è rappresentata da $u^2(t)$, la sua approssimazione collocazionale può essere espressa come segue.

$$\min_{u(t)} \int_{t_0}^{t_N} u^2(\tau) d\tau \approx \min_{u_0 \dots u_N} \sum_{k=0}^{N-1} \frac{h_k}{2} (u_k^2 + u_{k+1}^2) \quad (4.6)$$

4.1.1 Inizializzazione

La maggior parte dei risolutori non lineari richiede una stima iniziale. Per problemi semplici, come il problema d'esempio presentato, si ottiene un risultato corretto a partire da un grande range di possibili stime iniziali. Per quelli più complessi, una stima iniziale scarsa può portare il risolutore a rimanere bloccato su una soluzione sbagliata, o può portare al fallimento completo della convergenza ad una soluzione.

Tornando all'esempio del blocco in movimento trattato, assumiamo semplicemente che la posizione del blocco (x) cambia linearmente tra la posizione

iniziale e finale. Deriviamo successivamente la posizione iniziale della traiettoria per calcolare le traiettorie della velocità (v) e della forza (u). Questa opzione delle traiettorie iniziali soddisfa la dinamica del sistema e le condizioni di frontiera sulle posizioni, ma viola le condizioni di frontiera sulle velocità.

$$\begin{aligned}x^{\text{init}}(t) &= t \\v^{\text{init}}(t) &= \frac{d}{dt}x^{\text{init}}(t) = 1 \\u^{\text{init}}(t) &= \frac{d}{dt}v^{\text{init}}(t) = 0\end{aligned}\tag{4.7}$$

Una volta che abbiamo ottenuto una traiettoria iniziale, la possiamo valutare in ogni punto di collocazione per ottenere valori da passare al risolutore non-lineare.

$$x_k^{\text{init}} = t_k \quad v_k^{\text{init}} = 1 \quad u_k^{\text{init}} = 0\tag{4.8}$$

4.1.2 Programma non lineare

Abbiamo usato la collocazione trapezoidale per trascrivere il problema di ottimizzazione della traiettoria a tempo continuo in un programma non-lineare (ovvero in un problema di ottimizzazione non vincolato), ora dobbiamo risolverlo. Generalmente dopo aver fatto la trascrizione diretta, un problema di ottimizzazione della traiettoria viene convertito in un problema di programmazione non-lineare. Scopriamo che, per questo esempio, otteniamo un problema quadratico: i vincoli 4.3, 4.4 e 4.5 sono ambedue lineari, e la funzione obiettivo 4.6 è quadratica. Risolvere un problema quadratico è normalmente molto più facile che uno non lineare.

Assumiamo di aver risolto il programma non lineare. Abbiamo un insieme di posizioni x_k , velocità v_k e controlli u_k che soddisfano la dinamica e i vincoli di frontiera e che minimizzano il funzionale costo. Tutto quello che rimane è la costruzione di una spline che interpola la traiettoria della soluzione tra i vari punti di collocazione. Per la collocazione trapezoidale, risulta comodo

l'utilizzo di una spline lineare per il controllo, una quadratica per lo stato (la scelta dipende dal tipo di problema fisico in questione).

Una volta che abbiamo visto come applicare la collocazione trapezoidale ad un semplice esempio, vediamo più in dettaglio come risolvere un problema generico con questo metodo.

4.1.3 Formulazione generale

Riassumiamo brevemente ciò visto fino a questo punto mediante l'esempio del blocco in movimento tra due punti prestabiliti che rappresentano la frontiera di un problema di ottimizzazione della traiettoria. Il metodo di collocazione trapezoidale converte un sistema di ottimizzazione della traiettoria in un programma non lineare. Questo viene fatto usando la regola del trapezio per l'integrazione, per convertire ogni aspetto continuo del problema in una sua approssimazione discreta. In questa sezione vedremo come questa trasformazione viene fatta per ogni aspetto di un problema di ottimizzazione della traiettoria.

Solitamente ci sono delle espressioni integrali nell'ottimizzazione della traiettoria. Esse sono presenti nel funzionale obiettivo, e talvolta possono essere presenti anche nei vincoli. Il nostro obiettivo qui è di approssimare gli integrali continui $\int w(\cdot)dt$ mediante una sommatoria $\sum c_k w_k$. Il concetto chiave è che la sommatoria richiede solo il valore dell'integrando $w(t_k) = w_k$ nei punti di collocazione t_k lungo la traiettoria. Questa approssimazione viene fatta applicando la regola del trapezio per l'integrazione tra ogni coppia di punti di collocazione, che porta ad esprimere il rapporto tra sommatoria e integrale con la seguente espressione, dove $h_k = t_{k+1} - t_k$ [10].

$$\int_{t_0}^{t_F} w(\tau, \mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau \approx \sum_{k=0}^{N-1} \frac{h_k}{2} (w_k + w_{k+1}) \quad (4.9)$$

Una delle funzioni chiave dei metodi di collocazione è che rappresentano la dinamica del sistema come un insieme di vincoli, noti come **vincoli di collocazione**. Per la collocazione trapezoidale, i vincoli di collocazione sono costruiti scrivendo la dinamica nella forma integrale e approssimando l'integrale con la regola del trapezio [10].

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{f} \\
\int_{t_k}^{t_{k+1}} \dot{\mathbf{x}} \, dt &= \int_{t_k}^{t_{k+1}} \mathbf{f} \, dt \\
\mathbf{x}_{k+1} - \mathbf{x}_k &\approx \frac{h_k}{2} (\mathbf{f}_{k+1} + \mathbf{f}_k)
\end{aligned} \tag{4.10}$$

Questa approssimazione viene poi applicata tra ogni coppia di punti di collocazione:

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \frac{h_k}{2} (\mathbf{f}_{k+1} + \mathbf{f}_k) \quad k \in 0 \dots (N-1) \tag{4.11}$$

Possiamo notare che \mathbf{x}_k è una variabile di decisione in un programma non lineare, mentre $\mathbf{f}_k = \mathbf{f}(t_k, \mathbf{x}_k, \mathbf{u}_k)$ è il risultato della valutazione della dinamica del sistema in ogni punto di collocazione.

Oltre ai vincoli di collocazione, che rinforzano la dinamica del sistema, ci possono essere anche dei vincoli sullo stato, sul controllo, sul cammino e vincoli di frontiera. Questi vincoli sono tutti gestiti rinforzandoli a specifici punti collocazionali. Ad esempio, semplici vincoli sullo stato e sul controllo possono essere approssimati:

$$\mathbf{x} < \mathbf{0} \rightarrow \mathbf{x}_k < \mathbf{0} \quad \forall k \tag{4.12}$$

$$\mathbf{u} < \mathbf{0} \rightarrow \mathbf{u}_k < \mathbf{0} \quad \forall k \tag{4.13}$$

I vincoli di frontiera vengono rinforzati sul punto di collocazione iniziale e finale.

$$\begin{aligned}
\mathbf{h}(t_0, \mathbf{x}(t_0), \mathbf{u}(t_0)) < \mathbf{0} &\rightarrow \mathbf{h}(t_0, \mathbf{x}_0, \mathbf{u}_0) < \mathbf{0} \\
\mathbf{h}(t_N, \mathbf{x}(t_N), \mathbf{u}(t_N)) < \mathbf{0} &\rightarrow \mathbf{h}(t_N, \mathbf{x}_N, \mathbf{u}_N) < \mathbf{0}
\end{aligned} \tag{4.14}$$

Infine, ci sono due accortezze da tenere a mente riguardo ai vincoli. In primo luogo, i problemi di ottimizzazione della traiettoria con vincoli sul cammino tendono ad essere molto più difficili da risolvere della loro controparte senza. I dettagli su questo comportamento vanno oltre i scopi di questa tesi. Inoltre, nella collocazione trapezoidale i vincoli della traiettoria sono sempre dei punti di collocazione. Ci sono alcuni metodi, per i quali questi vincoli non

sono dei punti di collocazione. Per questi metodi è necessaria una trattazione diversa dei vincoli.

La collocazione trapezoidale approssima la traiettoria del controllo e la dinamica del sistema in funzioni lineari tra due coppie di punti fissati. Questa approssimazione è nota anche come **spline lineare**. Nel paragrafo precedente, dov'è stato presentato un esempio di questo metodo non abbiamo detto in modo approfondito come trovare le varie spline che compongono queste traiettorie. Quando costruiamo una spline, il termine punto di contatto si riferisce a qualsiasi punto che congiunge due segmenti polinomiali. Per la collocazione trapezoidale, i punti di contatto delle spline sono coincidenti con i punti di collocazione.

Consideriamo una traiettoria di controllo che è una semplice spline lineare. Sappiamo entrambi il tempo e il controllo ad ogni punto di contatto, quindi è immediato derivarne l'espressione per \mathbf{u} nell'intervallo $t \in [t_k, t_{k+1}]$. Per semplificare le equazioni successive definiamo $\tau = t - t_k$ e $h_k = t_{k+1} - t_k$.

$$\mathbf{u}(t) \approx \mathbf{u}_k + \frac{\tau}{h_k} (\mathbf{u}_{k+1} - \mathbf{u}_k) \quad (4.15)$$

La traiettoria dello stato è rappresentata da una spline quadratica. Questo fatto deriva direttamente dalle equazioni di collocazione 4.11. Le equazioni di collocazione trapezoidali sono esatte quando la dinamica del sistema varia linearmente tra due punti di contatto, un fatto che usiamo per approssimare la dinamica su un singolo segmento $t \in [t_k, t_{k+1}]$ come mostrato qui sotto.

$$\mathbf{f}(t) = \dot{\mathbf{x}}(t) \approx \mathbf{f}_k + \frac{\tau}{h_k} (\mathbf{f}_{k+1} - \mathbf{f}_k) \quad (4.16)$$

Tuttavia siamo interessati allo studio dello spazio e non della sua derivata prima (ovvero volgiamo l'equazione di \mathbf{x} invece che di $\dot{\mathbf{x}}$), quindi integriamo entrambi i lati dell'equazione 4.16 per ottenere un'espressione quadratica per lo stato, come avviene di seguito.

$$\mathbf{x}(t) = \int \dot{\mathbf{x}}(t) d\tau \approx c + \mathbf{f}_k \tau + \frac{\tau^2}{2h_k} (\mathbf{f}_{k+1} - \mathbf{f}_k) \quad (4.17)$$

Possiamo risolvere l'equazione 4.17 per la costante di integrazione c usando il valore dello stato alla frontiera $\tau = 0$ per ottenere l'equazione finale dello stato.

$$\mathbf{x}(t) \approx \mathbf{x}_k + \mathbf{f}_k \tau + \frac{\tau^2}{2h_k} (\mathbf{f}_{k+1} - \mathbf{f}_k) \quad (4.18)$$

Le equazioni delle spline 4.15 e 4.18 sono specifiche per la collocazione trapezoidale, poiché c'è una corrispondenza uno a uno tra le equazioni di collocazione e la spline interpolante. In generale, se il controllo è una spline di ordine n , allora lo stato è rappresentato da una spline di ordine $n + 1$.

4.2 Collocazione di Hermite-Simpson

La collocazione di Hermite-Simpson è simile alla collocazione trapezoidale, ma fornisce una soluzione che è di ordine maggiore ed è più accurata. Questo succede perché la collocazione trapezoidale approssima la funzione obiettivo e la dinamica del sistema con funzioni lineari, mentre la collocazione di Hermite-Simpson li approssima come funzioni quadratiche. Un vantaggio ulteriore del metodo di collocazione è che la traiettoria dello stato è una spline cubica hermitiana, ed ha una derivata prima continua.

Le espressioni integrali sono comuni in problemi di ottimizzazione della traiettoria, soprattutto nella funzione obiettivo. Il metodo di collocazione di Hermite-Simpson approssima questi integrali usando la quadratura di Simpson, nota anche come regola di Simpson per l'integrazione. La regola funziona approssimando l'integrando del integrale come una funzione quadratica.

$$\int_{t_0}^{t_F} w(\tau) d\tau \approx \sum_{k=0}^{N-1} \frac{h_k}{6} (w_k + 4w_{k+1/2} + w_{k+1}) \quad (4.19)$$

In qualsiasi metodo di collocazione, i vincoli di collocazione sono costruiti per approssimare la dinamica del sistema. Nel metodo di collocazione di Hermite-Simpson costruiamo questi vincoli riscrivendo la dinamica del sistema in forma integrale: il cambiamento nello stato tra due punti di contatto t_k dovrebbe essere uguale all'integrale della dinamica del sistema $\mathbf{f}(\cdot)$ tra questi punti.

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f} \\ \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}} \, dt &= \int_{t_k}^{t_{k+1}} \mathbf{f} \, dt \end{aligned} \quad (4.20)$$

La trascrizione dalla dinamica continua ad un insieme di equazioni di collocazione si ha quando approssimiamo l'integrale continuo in 4.20 con la quadratura di Simpson e la applichiamo tra ogni coppia di punti di contatto.

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \frac{h_k}{6} (\mathbf{f}_k + 4\mathbf{f}_{k+1/2} + \mathbf{f}_{k+1}) \quad (4.21)$$

Per la collocazione di Hermite-Simpson abbiamo bisogno anche di un'altra equazione di collocazione, oltre a 4.21, per rinforzare la dinamica. Questo accade poiché la dinamica del punto intermedio del segmento $\mathbf{f}_{k+1/2}$ è una funzione dello stato $\mathbf{x}_{k+1/2}$, il quale non è noto a priori. Possiamo calcolare lo stato del punto intermedio costruendo un'interpolazione per la traiettoria dello stato e valutandola nel punto intermedio dell'intervallo.

$$\mathbf{x}_{k+1/2} = \frac{1}{2} (\mathbf{x}_k + \mathbf{x}_{k+1}) \frac{h_k}{8} (\mathbf{f}_k - \mathbf{f}_{k+1}) \quad (4.22)$$

L'equazione 4.22 è particolare in quanto può essere valutata esplicitamente in termini di stato nei punti di contatto. Quindi, è possibile combinare ambedue le equazioni 4.21 e 4.22 in un singolo e complicato vincolo di collocazione. Quando la trascrizione del sistema dinamico avviene usando un singolo vincolo di collocazione, la formulazione che ne risulta è detta in forma compressa. Una formulazione alternativa è quella di creare una variabile di decisione addizionale per lo stato nel punto intermedio $\mathbf{x}_{k+1/2}$, e di usare entrambe le equazioni 4.21 e 4.22 come equazioni dei vincoli. Quando le equazioni di collocazione sono formulate usando questa coppia di vincoli, esse sono dette in forma separata.

Esiste molte varianti tra la forma separata e compressa del metodo di collocazione di Hermite-Simpson. La regola generale è che la forma separata è meglio quando si lavora con un numero piccolo di segmenti, mentre quella compressa quando il numero di segmenti è grande.

4.2.1 Formulazione generale

Oltre ai vincoli di collocazione, che rinforzano la dinamica del sistema, possiamo anche avere dei limiti sullo stato e sul controllo, vincoli sul cammino e vincoli sulla frontiera. Questi vincoli vengono gestiti valutandoli a specifici punti di contatto. Ad esempio, dei limiti sullo stato e sul controllo possono essere approssimati con le seguenti equazioni.

$$\mathbf{x} < \mathbf{0} \rightarrow \mathbf{x}_k < \mathbf{0}, \quad \mathbf{x}_{k+1/2} < \mathbf{0} \quad (4.23)$$

$$\mathbf{u} < \mathbf{0} \rightarrow \mathbf{u}_k < \mathbf{0}, \quad \mathbf{u}_{k+1/2} < \mathbf{0} \quad (4.24)$$

I vincoli di cammino possono essere gestiti in un modo analogo; vengono applicati su tutti i punti di collocazione, come mostrato di seguito.

$$\mathbf{g}(t, \mathbf{x}, \mathbf{u}) < \mathbf{0} \rightarrow \mathbf{g}(t_k, \mathbf{x}_k, \mathbf{u}_k) < \mathbf{0}, \quad \mathbf{g}(t_{k+1/2}, \mathbf{x}_{k+1/2}, \mathbf{u}_{k+1/2}) < \mathbf{0} \quad (4.25)$$

Mentre i vincoli di frontiera sono valutati ai punti di contatto iniziali e finali.

$$\begin{aligned} \mathbf{h}(t_0, \mathbf{x}(t_0), \mathbf{u}(t_0)) < \mathbf{0} &\rightarrow \mathbf{h}(t_0, \mathbf{x}_0, \mathbf{u}_0) < \mathbf{0} \\ \mathbf{h}(t_N, \mathbf{x}(t_N), \mathbf{u}(t_N)) < \mathbf{0} &\rightarrow \mathbf{h}(t_N, \mathbf{x}_N, \mathbf{u}_N) < \mathbf{0} \end{aligned} \quad (4.26)$$

Così come nel metodo di collocazione trapezoidale, i problemi di ottimizzazione della traiettoria con vincoli sul cammino tendono a essere molto più difficili che i problemi senza. In più, nella collocazione con Hermite-Simpson le frontiere della traiettorie sono sempre punti di contatto. Ci sono dei metodi dove questo non accade ed i vincoli di frontiera devono essere gestite con metodologie più complicate. L'analisi di queste metodologie esula dagli scopi di questa tesi.

Una volta che abbiamo risolto il programma non-lineare, sappiamo i valori delle traiettorie dello stato e del controllo per ogni punto di contatto. Il prossimo passo è quello di costruire una traiettoria continua per interpolare la soluzione tra i punti di collocazione. Così come avviene per la collocazione trapezoidale, useremo dei interpolanti polinomiali che vengono derivati dalle equazioni di collocazione.

La collocazione mediante Hermite-Simpson funziona usando la quadratura di Simpson per approssimare ogni segmento della traiettoria. Essa usa dei segmenti quadratici, che vengono posizionati attraverso tre punti uniformemente spazati, per approssimare l'integrando. In questo caso stiamo approssimando entrambi il controllo e la dinamica del sistema come quadratiche sopra ogni segmento della traiettoria. L'equazione generale per interpolazione quadratica per una curva $\mathbf{u}(t)$ che passa tra tre punti (t_A, \mathbf{u}_A) , (t_B, \mathbf{u}_B) , e (t_C, \mathbf{u}_C) viene riprodotta di seguito [47].

$$\begin{aligned} \mathbf{u}(t) = & \frac{(t - t_B)(t - t_C)}{(t_A - t_B)(t_A - t_C)} \mathbf{u}_A + \frac{(t - t_A)(t - t_C)}{(t_B - t_A)(t_B - t_C)} \mathbf{u}_B \\ & + \frac{(t - t_A)(t - t_B)}{(t_C - t_A)(t_C - t_B)} \mathbf{u}_C \end{aligned} \quad (4.27)$$

Per il nostro caso, possiamo semplificare questa equazione poiché i nostri punti sono spazati uniformemente. Usiamo k , $k + 1/2$ e $k + 1$ invece di A , B e C rispettivamente. Inoltre, come nella sezione precedente, definiamo $h_k = t_{k+1} - t_k$, $t_{k+1/2} = \frac{t_k + t_{k+1}}{2}$ e $\tau = t - t_k$. Dopo aver svolto le sostituzioni insieme a operazioni algebriche di base, possiamo arrivare alla seguente equazione semplificata per l'interpolazione della traiettoria di controllo.

$$\begin{aligned} \mathbf{u}(t) = & \frac{2}{h_k^2} \left(\tau - \frac{h_k}{2} \right) (\tau - h_k) \mathbf{u}_k \\ & - \frac{4\tau}{h_k^2} (\tau - h_k) \mathbf{u}_{k+1/2} + \frac{2\tau}{h_k^2} \left(\tau - \frac{h_k}{2} \right) \mathbf{u}_{k+1} \end{aligned} \quad (4.28)$$

La collocazione di Hermite-Simpson rappresenta la dinamica del sistema $\mathbf{f}(\cdot) = \dot{x}$ usando polinomiali quadratici per ogni segmento. Di conseguenza, la formulazione adoperata per la traiettoria del controllo, può essere anche applicata alla dinamica del sistema.

$$\begin{aligned} \dot{\mathbf{x}}(t) = & \frac{2}{h_k^2} \left(\tau - \frac{h_k}{2} \right) (\tau - h_k) \mathbf{f}_k \\ & - \frac{4\tau}{h_k^2} (\tau - h_k) \mathbf{f}_{k+1/2} + \frac{2\tau}{h_k^2} \left(\tau - \frac{h_k}{2} \right) \mathbf{f}_{k+1} \end{aligned} \quad (4.29)$$

Anche in questo caso ci interessa l'espressione per la traiettoria dello stato $\mathbf{x}(t)$ invece della sua derivata $\dot{\mathbf{x}}(t)$. Per ottenere la traiettoria dello stato, integriamo l'equazione 4.29 dopo averla sistemata in modalità polinomiale.

$$\begin{aligned} \mathbf{x}(t) = \int \mathbf{f}_k + (-3\mathbf{f}_k + 4\mathbf{f}_{k+1/2} - \mathbf{f}_{k+1}) \left(\frac{\tau}{h_k} \right) \\ + (2\mathbf{f}_k - 4\mathbf{f}_{k+1/2} + 2\mathbf{f}_{k+1}) \left(\frac{\tau}{h_k} \right)^2 dt \end{aligned} \quad (4.30)$$

Possiamo calcolare l'integrale usando l'algebra di base, e successivamente usare come costante di integrazione la condizione di frontiera $\mathbf{x}(t_k) = \mathbf{x}_k$. Riportiamo di seguito l'equazione che ne risulta. Essa ci permette di interpolare la traiettoria dello stato.

$$\begin{aligned} \mathbf{x}(t) = x_k + \mathbf{f}_k \left(\frac{\tau}{h_k} \right) + \frac{1}{2} (-3\mathbf{f}_k + 4\mathbf{f}_{k+1/2} - \mathbf{f}_{k+1}) \left(\frac{\tau}{h_k} \right)^2 \\ + \frac{1}{3} (2\mathbf{f}_k - 4\mathbf{f}_{k+1/2} + 2\mathbf{f}_{k+1}) \left(\frac{\tau}{h_k} \right)^3 \end{aligned} \quad (4.31)$$

Capitolo 5

Metodi indiretti

Nel paragrafo 3.3 abbiamo visto come rappresentare un quadricottero con sei gradi di libertà. Abbiamo visto l'assetto del veicolo con la matrice di rotazione e gli input di controllo rappresentati dai tassi rotazionali desiderati sulle assi del sistema di riferimento del veicolo. Quindi abbiamo visto l'evoluzione del sistema che può essere calcolata mediante le equazioni del moto (l'accelerazione traslazionale è data dall'assetto e dalla forza dei propulsori).

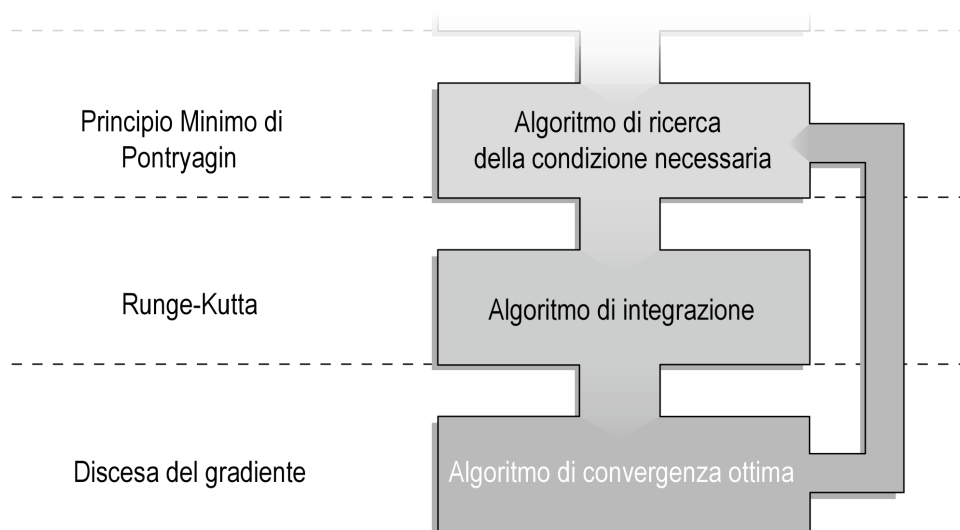


Figura 5.1: Schema concettuale della ricerca della traiettoria ottima con l'uso di un metodo indiretto

Siamo passati all'analisi di un caso più semplice nel paragrafo 3.5, ovvero il caso di un quadricottero vincolato che non può ruotare attorno alla direzione z e y . Abbiamo quindi ottenuto un sistema con due gradi di libertà (x, θ) . Per implementare la dinamica di un sistema che rappresenta l'evoluzione di un quadricottero vincolato, abbiamo usato le equazioni 3.26 e 3.27, che sono espresse in un sistema Hamiltoniano. In questo modo nello stato del sistema risultante viene espresso: lo stato originale \mathbf{q} , che incontriamo anche nella Lagrangiana del sistema $(\mathbf{q}, \dot{\mathbf{q}})$, e il momento coniugato \mathbf{p} relativo al vettore $\dot{\mathbf{q}}$. Lo stato nel nostro sistema può essere espresso con la coppia di vettori (\mathbf{q}, \mathbf{p}) , ognuno dei quali composto di due elementi, uno per grado di libertà.

In questo capitolo il nostro obiettivo è quello di definire il problema dinamico, ovvero trovare una soluzione, per via analitica al problema della traiettoria ottima. Iniziamo introducendo in dettaglio come questi problemi vengono risolti. Viene rielaborata la parte introdotta nel capitolo 2.1.1 in cui abbiamo risolto analiticamente un problema di ottimizzazione della traiettoria con la potenza minima. Il problema che avevamo risolto era relativamente semplice, e pertanto non abbiamo riscontrato particolari problemi a ottenere una soluzione analitica. Nel caso di un quadricottero vincolato questo non avviene, e abbiamo bisogno di un meccanismo più complesso. Le basi matematiche che abbiamo introdotto valgono in entrambi i casi, ma la soluzione ottenuta con il metodo presentato in questo caso permette una formulazione più generale e può essere applicata a una vasta classe di problemi di ottimizzazione della traiettoria, dalla robotica, all'automazione industriale, fino all'aviazione. L'utilizzo di questo meccanismo per quadricotteri permette di sfruttare direttamente le proprietà dell'hamiltoniano. In particolare, grazie a questo approccio, possiamo vincolare i momenti coniugati a determinate condizioni.

Qualora il nostro obiettivo è quello di far passare il quadricottero attraverso spazi ristretti, o assicurarci che il suo assetto verticale sia rispettato in determinati istanti temporali, il momento coniugato può tornarci d'aiuto. Se abbiamo che un assetto inclinato del quadricottero non permette il passaggio attraverso uno spazio ridotto, possiamo vincolare il momento coniugato ad un valore costante positivo $c_p > 0$, con un angolo $\bar{\theta} \approx 0$. In questo modo, con una spinta sufficiente e senza ulteriore azione dei propulsori, il quadricottero

manterrà un assetto orizzontale e permetterà il passaggio attraverso la zona interessata, in cui in condizioni normali il passaggio non sarebbe possibile.

5.1 Introduzione

Come abbiamo visto nella prima parte di questa tesi, in una situazione standard, e a meno che le equazioni di sistema, l'indice delle performance, e i vincoli siano troppo semplici, dobbiamo adoperare metodi numerici per la risoluzione dei problemi di ottimizzazione. La quantità di calcolo richiesta per problemi anche relativamente semplici è proibitiva se il calcolo deve essere fatto senza metodi automatici. Questo è il motivo principale per cui il calcolo delle variazioni ha trovato un utilizzo concreto nell'ingegneria e nelle scienze applicate solo a partire dalla metà degli anni '50. Lo sviluppo negli ultimi anni di microprocessori economici ad alta velocità, ha permesso un'ulteriore evoluzione e ha concesso l'utilizzo di queste metodologie anche negli ambienti real-time, dove è necessario fornire una soluzione entro una deadline. La mancanza di una soluzione entro la deadline in sistemi real-time può portare a condizioni pericolose, e per alcuni sistemi anche a danni fallimentari che possono portare al malfunzionamento generale del sistema.

Nella sezione 2.1 abbiamo analizzato il problema di ottimizzazione della traiettoria restringendoci ad un problema a singola fase e a tempo continuo. Abbiamo quindi visto che possiamo ottenere una soluzione mediante metodi diretti o indiretti. I primi, denominati anche metodi diretti di collocazione, sono stati trattati ampiamente nella sezione 4 a loro dedicata. I metodi indiretti invece, trovano la traiettoria ottima con un principio analogo alla soluzione analitica adoperata nella sezione 2.1.1. Costruiscono analiticamente le condizioni necessarie e sufficienti per l'ottimalità e le risolvono numericamente.

Gli algoritmi indiretti devono necessariamente partire da una traiettoria sub-ottima che deve essere elaborata a priori dell'algoritmo di ottimizzazione stesso. Il processo di definizione di una traiettoria sub-ottima iniziale può essere comparato al processo di ottenimento di stime iniziali per metodo indiretto di shooting. Il metodo di shooting, a partire da una stima iniziale rappresentata dall'angolo di tiro, effettua la dinamica del sistema, controlla i

risultati e ridefinisce le condizioni iniziali di lancio. In questo modo è in grado di ottenere, alla seconda interazione del metodo, un risultato più accurato (a patto di non cadere in una condizione di over-shooting, in cui il metodo diverge invece di convergere ad ogni step ad una soluzione più accurata). Allo stesso modo, per metodi complessi come quello presentato qui, per trovare la traiettoria del controllo ottima, è innanzitutto essenziale fornire una traiettoria sub-ottima. Se questa traiettoria risulta ottima, non sarà necessario fare altre iterazioni del metodo. Qualora essa non lo è, la traiettoria dovrà essere raffinata. Appena viene raggiunto un range di errore ritenuto accettabile, ulteriori iterazioni del metodo non si rendono più necessarie. Questa metodologia ricorsiva è comune alla quasi totalità degli algoritmi indiretti.

In questo capitolo vogliamo ottenere la traiettoria ottima mediante la minimizzazione del tempo di volo. Il tempo ottimo è una scelta appropriata per il problema di un quadricottero in moto, ma non è l'unica. Possiamo allo stesso modo definire il controllo minimo, o la potenza minima, come avveniva nella soluzione analitica che abbiamo calcolato nel paragrafo 2.1.1. La soluzione del problema dinamico è la traiettoria a tempo minimo per una data traiettoria sub-ottima. La minimizzazione del tempo del moto è uno degli obiettivi nella gerarchia dei problemi di pianificazione in un ambiente temporale. La soluzione del problema dinamico raggiunge quindi uno di questi obiettivi gerarchici. La traiettoria ottima è altrettanto utile nello sviluppo e valutazione di euristiche per problemi cinematici, in quanto le traiettorie sub-ottime possono essere comparata con quelle ottime e la loro euristica può venire variata di conseguenza. Il capitolo usa la teoria e le basi matematiche elaborate nella tesi di dottorato del Prof. Paolo Fiorini [48]. L'ottimizzazione avviene sulla condizione necessaria derivata dal Principio di Pontryagin [49].

5.1.1 Principio Minimo di Pontryagin

Il principio Minimo (o l'analogo massimo) di Pontryagin è una metodologia sviluppata a partire dagli anni '50 e '60 dal matematico russo Pontryagin. Si basa sul principio del minimo (o massimo). È stata adoperata in relazione ai problemi di massimizzazione di funzionali di costo, e porta alla formula-

zione di condizioni necessarie, a differenza della maggior parte delle teorie antecedenti (che invece fornivano solo delle condizioni sufficienti). La teoria complementare a quella del principio si basa sugli studi di Hamilton-Jacobi e porta al calcolo di equazioni differenziali alle derivate parziali. Questi calcoli sono ben più onerosi di quelli introdotti da Pontryagin.

L'uso di condizioni necessarie permette di trovare abbastanza semplicemente una soluzione, che se le soddisfa, è una buona candidata ad essere una soluzione ottima. Ma non è detto che questa soluzione sia ottima (altrimenti sarebbero anche sufficienti). Per affermare che una soluzione trovata imponendo le condizioni necessarie fornite dal teorema di Pontryagin è effettivamente una soluzione ottima, serve un'analisi supplementare. Esistono dei risultati che sfruttano le proprietà delle funzioni coinvolte, come concavità o convessità e linearità. Oppure si possono trovare tutte le soluzioni ottime e verificare che quella trovata è l'unica che soddisfa le condizioni necessarie, ed è quindi per forza ottima.

Possiamo riassumere il Principio Minimo di Pontryagin con il seguente teorema.

Theorem 1. *Sia $\dot{\mathbf{q}}$ l'evoluzione dello stato del sistema dinamico*

$$\dot{\mathbf{q}} = \mathbf{f}(q_1, q_2, \dots, q_n, u_1, u_2, \dots, u_m) \quad j = 1, 2, \dots, n$$

con gli ingressi $\mathbf{u} = u_1, u_2, \dots, u_m$ indipendenti dallo stato \mathbf{q} e dal tempo t , e sia J il funzionale costo

$$J(t, \mathbf{q}, \mathbf{u}) = \int_{t_0}^{t_f} \mathcal{L} dt$$

dove \mathcal{L} è il funzionale costo che si intende minimizzare, nel nostro caso $\mathcal{L} = 1$, siano inoltre le condizioni iniziali Γ al tempo t_0 e finali Ω al tempo t_f date

$$\mathcal{H} = \mathcal{L} + \boldsymbol{\lambda}^T \cdot \mathbf{f}(\mathbf{q}, \mathbf{u})$$

dove $\boldsymbol{\lambda}$ è il cosiddetto vettore delle funzioni moltiplicatore. Sia \mathbf{u}^ , un controllo ottimo tra gli istanti temporali di inizio e fine, allora esiste una funzione vettoriale continua non nulla, $\boldsymbol{\lambda}(t) \in \mathbb{R}^{n+1}$, che soddisfa le equazioni dei co-stati [48]:*

$$\frac{d\lambda_j}{dt} = - \sum_n^{i=0} \frac{\partial f_i(\mathbf{q}, \mathbf{u}^*)}{q_j} \Big|_{\mathbf{q}=\mathbf{q}^*(t)} \lambda_i \quad j = 0, 1, \dots, n$$

con q_0 definito come segue:

$$\frac{dq_0}{dt} = \mathcal{L}(\mathbf{q}, \mathbf{u})$$

per qualsiasi istante temporale entro i limiti temporali vale che esiste un minimo nell'insieme di tutti i possibili controlli del sistema \mathbf{U} [48]

$$\min_{\mathbf{u} \in \mathbf{U}} \mathcal{H}(\boldsymbol{\lambda}, \mathbf{q}^*(t), \mathbf{u}(t)) = \mathcal{H}(\boldsymbol{\lambda}(t), \mathbf{q}^*(t), \mathbf{u}^*(t)) \quad (5.1)$$

oppure che il controllo ottimo trovato è già minimo, e in tal caso

$$\mathcal{H}(\boldsymbol{\lambda}(t), \mathbf{q}^*(t), \mathbf{u}^*(t)) = 0 \quad (5.2)$$

o ancora che $\lambda_0(t) = c$ dove $c \geq 0$ è una costante positiva dipendente dalla funzione obiettivo \mathcal{L} . \square

Il teorema trova la condizine necessaria per l'ottimaità e vale se i controlli ammessi sono limitati. Per evitare problemi con la dipendenza temporale, aumentiamo la dimensione dello spazio degli stati nel seguente modo, definendo $q_{n+1} \equiv t$ (e di conseguenza la sua derivata diventa $\dot{q}_{n+1} = 1$).

Questo trasforma il problema della pianificazione dello spostamento nella forma tempo-invariante richiesta dal teorema 1.

5.2 Formulazione del problema

Il problema della ricerca del controllo ottimo per un quadricottero che si deve spostare tra due stati fissati (tra due vincoli di frontiera), può essere formulato matematicamente come segue. Vogliamo trovare il controllo \mathbf{u} tra tutti i possibili controlli \mathbf{U} che rispetta i vincoli e che minimizza la funzione costo J .

$$J(t, \mathbf{q}, \mathbf{u}) = \int_0^1 \mathcal{L}(\mathbf{q}, \mathbf{u}) dt \quad (5.3)$$

Dove non teniamo conto dell'errore sullo stato finale in quanto lo vogliamo tenere prossimo a zero. La dinamica del sistema $\mathbf{f}(\cdot)$, conforme alle equazioni 3.26 e 3.27, viene riportata di sotto, con x che rappresenta la posizione del quadricottero nello spazio, θ l'angolo di pitch, e p_x con p_θ rappresentato rispettivamente i momenti coniugati relativi al vettore $\dot{\mathbf{q}}$. Il vettore stato è $\mathbf{q} = \begin{bmatrix} x & \theta & p_x & p_\theta \end{bmatrix}^T$.

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \dot{p}_x \\ \dot{p}_\theta \end{bmatrix} = \mathbf{f}(\mathbf{q}, \mathbf{u}) = \begin{bmatrix} \frac{p_x}{m} \\ \frac{p_\theta}{I_3(1+\cos^2 \theta)} \\ -mg \tan \theta + \frac{\kappa_x}{m} p_x \\ \frac{p_\theta^2 \sin \theta \cos \theta}{I_3(1+\cos^2 \theta)^2} + \tau_\theta \end{bmatrix} \quad (5.4)$$

Applicheremo anche le seguenti condizioni di frontiera, dove i pedici dei vettori indicano il valore della frontiera della traiettoria. Si tratta della stessa notazione che abbiamo adoperato nel paragrafo 2.1.1.

$$\begin{aligned} \mathbf{q}_0 = \mathbf{q}(t) \Big|_{t=0} &= \begin{bmatrix} x_0 \\ \theta_0 \\ p_{x_0} \\ p_{\theta_0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \Gamma(t_0, \mathbf{q}(t_0)) \\ \mathbf{q}_1 = \mathbf{q}(t) \Big|_{t=1} &= \begin{bmatrix} x_1 \\ \theta_1 \\ p_{x_1} \\ p_{\theta_1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2,025 \cdot 10^{-1} \\ 0 \end{bmatrix} = \Omega(t_f, \mathbf{q}(t_f)) \end{aligned} \quad (5.5)$$

Per problemi a tempo minimo, con specifiche condizioni di frontiera, la funzione costo \mathcal{L} può essere scritta.

$$\int_{t_0}^{t_f} \mathcal{L}(\mathbf{q}, \mathbf{u}) = t_f \quad \mathcal{L}(\mathbf{q}, \mathbf{u}) = 1 \quad (5.6)$$

Quindi possiamo riscrivere la funzione costo J usando la formulazione di Mayer (la funzione costo può includere due termini, obiettivo di frontiera e integrale di cammino; un problema con solo la parte obiettivo è detto nella forma di Mayer), come funzione di stato e tempo finale.

$$J = t_f = \phi(t_f, \mathbf{q}(t_f)) = \int_{t_0}^{t_f} \mathcal{L}(\mathbf{q}(t), \mathbf{u}(t)) dt \quad (5.7)$$

Per risolvere un problema di ottimizzazione vincolato, come quello del movimento di un quadricottero, dobbiamo aggiungere i vincoli alla funzione costo J . Questa operazione avviene a mezzo di un vettore di funzioni moltiplicatore dette anche **moltiplicatori di Lagrange**. Per fare questo esistono due opzioni. O i vincoli vengono aggiunti direttamente al funzionale costo dell'equazione 5.7 [50, 51], o vengono differenziati rispetto al tempo, finché i controlli \mathbf{u} non diventano espliciti. Le equazioni che ne risultano possono poi essere aggiunte come vincoli di controllo dipendenti dallo stato alla funzione obiettivo J [48]. Supponiamo di aver un vincolo tempo-variante di un ostacolo Ψ .

In un problema dove sono presenti i vincoli tempo-varianti, l'approccio di appendere i vincoli alla funzione obiettivo J non può essere realizzato. In questo caso dobbiamo fare un'operazione di trasformazione degli vincoli dello stato nel vincolo di controllo per ogni frontiera di un ostacolo mediante l'operazione di differenziazione. Il numero delle differenziazioni di ogni vincolo Ψ è detto l'**ordine del vincolo** [48].

Vediamo brevemente come funziona l'approccio presentato. Consideriamo il vincolo dovuto ad un singolo ostacolo:

$$\Psi(t, \mathbf{q}) = 0 \quad (5.8)$$

Derivando il vincolo n volte, otteniamo il controllo \mathbf{u} .

$$\Psi^{(n)}(\mathbf{q}, \mathbf{u}) = 0 \quad (5.9)$$

Assumiamo che t_1 sia il tempo di entrata nell'ostacolo, e t_2 il tempo di uscita. Per far in modo che il vincolo sullo stato sia valido, tutte le derivate di Ψ devono essere nulle (tranne che la derivata n -esima) al tempo t_1 .

Nel caso del quadricottero i controlli devono sottostare a determinati range operazionali, e di conseguenza abbiamo dei vincoli sul controllo. Il controllo all'istante i , denotato con u_i , deve sottostare ad un intervallo. Questi vincoli possono essere espressi con la seguente formulazione.

$$\mathbf{U} = \{\mathbf{u} \mid u_{\min} \leq u_i \leq u_{\max}\} \quad (5.10)$$

In quanto abbiamo ottenuto anche un altro vincolo sul controllo dovuto a causa della derivazione del vincolo sullo stato che non era applicabile, l'insieme dei vincoli sul controllo deve essere esteso. Sia \mathcal{U} un nuovo set di vincoli sul controllo definito come segue.

$$\mathcal{U} = \begin{cases} u_{\min} \leq u_i \leq u_{\max} \\ \Psi^{(n)}(\mathbf{q}, \mathbf{u}) = 0 \end{cases} \quad \text{se } t_1 \leq t \leq t_2 \quad (5.11)$$

In questo modo otteniamo un problema di ottimizzazione a tre punti, in quanto la soluzione deve soddisfare oltre i vincoli iniziali e finali anche il vincolo derivato dal vincolo originale dello stato sull'ostacolo.

5.3 Condizione necessaria per l'ottimalità

Nel paragrafo 5.1.1 abbiamo riassunto il Principio Minimo di Pontryagin, adesso vediamo come applicarlo per ottenerne la condizione necessaria per l'ottimalità.

Sia $\varphi(\mathbf{q}, \mathbf{u}) \subset \mathcal{U}$ l'insieme dei vincoli di controllo che abbiamo definito nell'equazione 5.11, e che tengono conto del vincolo sullo stato intermedio (che è diventato con l'operazione di differenziazione un vincolo sul controllo). L'hamiltoniano \mathcal{H} può essere esteso con la seguente formulazione.

$$\mathcal{H} = \boldsymbol{\lambda}^T \cdot \mathbf{f} + \boldsymbol{\mu}^T \cdot \varphi \quad (5.12)$$

La dinamica del sistema deve essere espressa nella seguente forma. Ovvero deve essere lineare o approssimabile a lineare per applicare il teorema 1.

$$\mathbf{f} = \tilde{\mathbf{f}} + \mathbf{g} \cdot \mathbf{u} \quad (5.13)$$

Nel nostro caso questo è vero, in quanto possiamo riscrivere il sistema come segue.

$$\mathbf{f}(\mathbf{q}, \mathbf{u}) = \begin{bmatrix} \frac{p_x}{m} \\ \frac{p_\theta}{I_3(1+\cos^2 \theta)} \\ -mg \tan \theta + \frac{\kappa_x}{m} p_x \\ \frac{p_\theta^2 \sin \theta \cos \theta}{I_3(1+\cos^2 \theta)^2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tau_\theta \quad (5.14)$$

Nell'equazione 5.12, $\boldsymbol{\mu}$ è un nuovo vettore di variabili aggiunte, che può essere trovato mediante la derivata parziale sui vincoli del controllo φ che abbiamo trovato con l'equazione 5.11 [48].

$$\boldsymbol{\mu} = -\boldsymbol{\lambda}^T \mathbf{g}(\mathbf{q}) \left(\frac{\partial \varphi(\mathbf{q}, \mathbf{u})}{\partial \mathbf{u}} \right)^{-1} \quad (5.15)$$

E i co-stati $\boldsymbol{\lambda}$ soddisfano l'equazione della funzione costo aumentata per $t \in [t_1, t_2]$:

$$\dot{\boldsymbol{\lambda}} = -\boldsymbol{\lambda} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{q}} - \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \left(\frac{\partial \varphi}{\partial \mathbf{u}} \right)^{-1} \frac{\partial \varphi}{\partial \mathbf{q}} \right) \quad (5.16)$$

Queste due equazioni per co-stati soddisfano i vincoli di controllo \mathcal{U} [48].

Le nozioni che abbiamo visto nel paragrafo 5.1.1 possono essere ora usate per giustificare la seguente condizione necessaria di ottimalità.

Dato il funzionale costo J , che vogliamo minimizzare, dell'equazione 5.7, e l'hamiltoniano dell'equazione 5.12 che abbiamo ottenuto aumentando il funzionale costo, dove $\boldsymbol{\mu}$ e $\boldsymbol{\lambda}$ sono vettori di variabili aggiunte, mentre $\varphi(\mathbf{q}, \mathbf{u}) \subset \mathcal{U}$ rappresentano gli insiemi dei vincoli di controllo incluso il vincolo sullo stato originale trasformato a mezzo dell'operazione di derivazione. Allora, il controllo ottimo, $\mathbf{u}^*(t)$ che genera la soluzione ottima $\mathbf{q}^*(t)$ nell'intervallo $t_0 \leq t \leq t_f$, minimizza J , e soddisfa le condizioni iniziali Γ e finali Ω , deve soddisfare le seguenti condizioni necessarie [48]:

1. le equazioni hamiltoniane (successivamente nel capitolo vedremo come ottenere queste espressioni appartire dalla funzione costo aumentata)

$$0 = \left(\frac{\partial \mathcal{H}}{\partial \mathbf{u}} \right) \quad \dot{\boldsymbol{\lambda}} = - \left(\frac{\partial \mathcal{H}}{\partial \mathbf{q}} \right)^T \quad (5.17)$$

2. l'hamiltoniano del controllo ottimo deve essere inferiore a quello del controllo sub-ottimo

$$\mathcal{H}^*(\boldsymbol{\lambda}, \mathbf{q}^*, \mathbf{u}^*) \leq \mathcal{H}(\boldsymbol{\lambda}, \mathbf{q}^*, \mathbf{u}) \quad (5.18)$$

3. l'hamiltoniano del controllo ottimo deve essere nullo

$$\mathcal{H}^*(\boldsymbol{\lambda}(t), \mathbf{q}^*(t), \mathbf{u}^*(t)) = 0 \quad (5.19)$$

La traiettoria ottima \mathbf{u}^* del controllo, che troviamo con l'operazione della ricerca della condizione necessaria, che abbiamo adoperato di sopra, ci permette di valutare lo stato e di ottenere un cammino ottimo per esso. Questo cammino ottimo soddisfa:

- le condizioni iniziali Γ al tempo t_0
- le condizioni finali Ω al tempo t_f
- il funzionale costo J
- le equazioni dei moltiplicatori di lagrange (che devono essere trovati con l'operazione di integrazione in avanti)
- le condizioni necessarie dell'hamiltoniano

La traiettoria ottima trovata mediante un algoritmo che usa le basi matematiche del Principio Minimo di Pontryagin deve soddisfare le condizioni 1-3. Per la raffinazione della soluzione della traiettoria ottima, qualora la traiettoria sub-ottima risultante da una prima iterazione del algoritmo non sia soddisfacente, usiamo l'algoritmo di discesa del gradiente che trova una soluzione che converge alla traiettoria ottima, che nel nostro problema è rappresentata dalla traiettoria del quadricottero compiuta nel minor tempo possibile.

5.4 Ricerca della traiettoria ottima

Abbiamo visto che possiamo scrivere il problema del quadricottero in movimento tra due punti come un problema di valori ai limiti a tre punti.

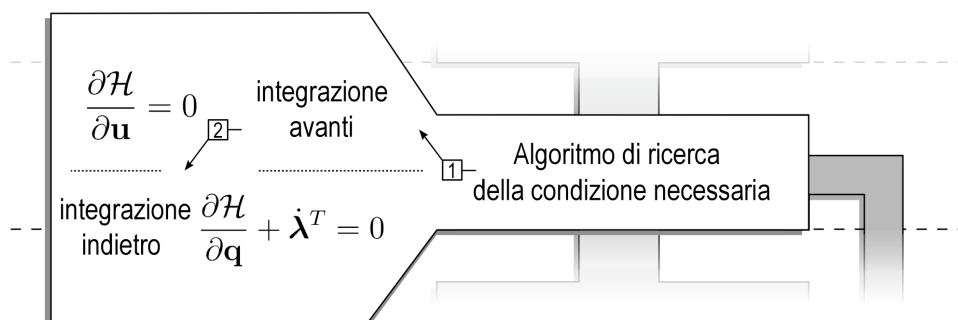


Figura 5.2: Schema concettuale dell'algoritmo della ricerca della condizione necessaria per trovare i valori delle funzioni moltiplicatore con cui ottimizzare la traiettoria del controllo

Con l'analisi del Principio Minimo di Pontryagin abbiamo trovato un insieme di condizioni necessarie per l'ottimalità. Si tratta di un approccio standard introdotto da Bryson e Ho [4]. Questo problema è un problema difficile da risolvere analiticamente, tranne che per pochi semplici casi, e pertanto può essere solo risolto numericamente [48].

Una volta che abbiamo trovato i valori delle funzioni moltiplicatore tali che le condizioni necessarie sono soddisfatte, necessitiamo di un ulteriore meccanismo per la raffinazione del controllo e di conseguenza per ottenere una soluzione ottima a partire da una stima iniziale sub-ottima. In questa tesi usiamo il **metodo della discesa del gradiente** per la ricerca dei controlli ottimi \mathbf{u}^* , a partire da una stima di controllo sub-ottima \mathbf{u} che viene aggiustata passo per passo ad ogni iterazione dell'algoritmo, finché il valore d'errore tra la soluzione ottima idealmente ottenuta, e quella sub-ottima restituita dall'algoritmo non è inferiore di un valore d'errore massimo prefissato.

Tutte le procedure che si basano sulla discesa del gradiente sono delle procedure di minimizzazione locale, e quindi possono portare solo ad un minimo locale (e non globale) in prossimità delle stime iniziali. Nel nostro caso l'algoritmo troverà una soluzione migliore, $\tilde{\mathbf{u}}$, alla soluzione sub-ottima \mathbf{u} fornita come stima iniziale all'algoritmo. Questo ci fornisce l'indicazione del fatto che una stima iniziale appropriata è essenziale, poiché la traiettoria \mathbf{u} verrà direttamente usata per ricavarne un minimo locale.

Assumiamo come funzionale costo il tempo totale $J = 1$, un artificio che abbiamo usato ampiamente in questo capitolo. In questo modo vogliamo trovare una traiettoria del controllo in modo che la traiettoria trovata abbia un tempo di completamento migliore della traiettoria stimata in partenza.

Dalla condizione necessaria del Principio di Pontryagin abbiamo che la varianza prima della funzione costo (o funzionale obiettivo) aumentata deve essere prossima a zero. Per ottenere la varianza prima, dobbiamo innanzitutto ottenere la funzione costo aumentata \bar{J} , da cui poi possiamo valutare la varianza prima analiticamente, e ottenere le espressioni delle funzioni moltiplicatore in modo che le condizioni necessarie siano soddisfatte.

5.4.1 Funzionale costo aumentato

Prima di tutto, per procedere con il calcolo della traiettoria ottima a partire da una stima sub-ottima fornita in ingresso \mathbf{u} , esprimiamo analiticamente come trovare la funzione costo aumentata. Per assicurarci della fattibilità della soluzione, dobbiamo soddisfare la dinamica del sistema. Innanzitutto modifichiamo il funzionale costo in modo che includa la dinamica del sistema e un vettore delle funzioni moltiplicatore $\boldsymbol{\lambda}$.

$$\bar{J} = \int_0^1 (\boldsymbol{\lambda}^T \cdot (\mathbf{f} - \dot{\mathbf{q}}) + \boldsymbol{\mu}^T \cdot \varphi) d\tau \quad (5.20)$$

La dinamica è soddisfatta qualora accade che $\mathbf{f} - \dot{\mathbf{q}} = 0$, $\varphi = 0$ e quindi $\bar{J} = J$ a prescindere delle funzioni moltiplicatore.

Usiamo la formula dell'integrazione per parti per riscrivere la funzione costo e ottenere la seguente espressione. Come prima, usiamo la notazione a pedice per indicare i limiti della traiettoria $\boldsymbol{\lambda}_0 = \boldsymbol{\lambda}(t) \Big|_{t=0}$.

$$\bar{J} = \boldsymbol{\lambda}_0^T \mathbf{q}_0 - \boldsymbol{\lambda}_1^T \mathbf{q}_1 + \int_0^1 \left((\boldsymbol{\lambda}^T \cdot \mathbf{f}) + (\dot{\boldsymbol{\lambda}}^T \cdot \mathbf{q}) + (\boldsymbol{\mu}^T \cdot \varphi) \right) d\tau \quad (5.21)$$

A questo punto possiamo notare che l'hamiltoniano del nostro sistema diventa della seguente forma, poiché la funzione che vogliamo minimizzare e che abbiamo espresso con il funzionale costo rappresenta il tempo minimo, e di conseguenza abbiamo che $\mathcal{L} = 1$.

$$\mathcal{H} = \boldsymbol{\lambda}^T \cdot \mathbf{f} + \boldsymbol{\mu}^T \cdot \varphi \quad (5.22)$$

Otteniamo il minimo di una funzione scalare semplice quando la derivata prima di questa funzione è nulla e la derivata seconda è positiva (per semplici regole di derivazione). Analogamente possiamo adoperare lo stesso principio anche per le traiettorie, sebbene usiamo il termine **variazione** invece di derivata. Una traiettoria ottima deve avere una variazione prima nulla, e una variazione seconda non negativa. Vediamo di fare in modo che la variazione prima sia nulla, in modo da trovare una traiettoria migliore che raffina la traiettoria sub-ottima di partenza.

5.4.2 Varianza prima del funzionale costo

Sia \mathbf{q}^* la traiettoria ottima dello stato, e sia \mathbf{u}^* la traiettoria ottima del controllo. La traiettoria sub-ottima fornita come stima iniziale \mathbf{u} e \mathbf{q} (equazione 5.23) può essere scritta come somma di una traiettoria ottima e una piccola perturbazione dalla traiettoria, come abbiamo già visto. Sia ε un parametro che dipende dal tempo finale t_f , poiché vogliamo ottenere la traiettoria a tempo minimo, con $\delta\mathbf{q}$ e $\delta\mathbf{u}$ arbitrarie variazioni nello stato e nel controllo rispettivamente.

$$\mathbf{q} = \mathbf{q}^* + \varepsilon\delta\mathbf{q} \quad \mathbf{u} = \mathbf{u}^* + \varepsilon\delta\mathbf{u} \quad (5.23)$$

La variazione prima della funzione costo è la derivata parziale rispetto al tempo.

$$\delta\bar{J} \equiv \left. \frac{\partial}{\partial t} \bar{J} \right|_{t=t_f} \quad (5.24)$$

Mediante l'uso delle regole di derivazione, scriviamo un'espressione per la variazione prima della funzione costo [4].

$$\begin{aligned}
\delta \bar{J} &= \lambda_0^T \frac{\partial \mathbf{q}_0}{\partial t} \Big|_{t=t_f} - \lambda_1^T \frac{\partial \mathbf{q}_1}{\partial t} \Big|_{t=t_f} + \varphi(\mathbf{q}, \mathbf{u}) \Big|_{t=t_f} dt_f \\
&\quad + \int_0^1 \left[\frac{\partial \mathcal{H}}{\partial t} \Big|_{t=t_f} + \dot{\lambda}^T \frac{\partial \mathbf{q}}{\partial t} \Big|_{t=t_f} \right] d\tau \\
&= \lambda_0^T \delta \mathbf{q}_0 - \lambda_1^T \delta \mathbf{q}_1 + \varphi(\mathbf{q}, \mathbf{u}) \Big|_{t=t_f} dt_f \\
&\quad + \int_0^1 \left[\left(\frac{\partial \mathcal{H}}{\partial \mathbf{q}} + \dot{\lambda}^T \right) \delta \mathbf{q} + \frac{\partial \mathcal{H}}{\partial \mathbf{u}} \delta \mathbf{u} \right] d\tau
\end{aligned} \tag{5.25}$$

Poiché per una traiettoria ottima la condizione necessaria deve essere soddisfatta, la variazione prima della funzione costo $\delta \bar{J}$ (equazione 5.25) deve essere nulla tra gli istanti temporali t_0 e t_f . Le variazioni sullo stato e sui punti iniziali e finali della traiettoria sono nulle, poiché le condizioni di frontiera sono prefissate, e di conseguenza ($\delta \mathbf{q}_0 = 0$, $\delta \mathbf{q}_1 = 0$). Dunque i primi due termini dell'equazione 5.25 sono entrambi nulli. Il valore dei vincoli nel tempo finale devono essere nulli. Inoltre, come avveniva nel esempio che abbiamo usato nel capitolo 2.1.1, le variazioni sugli stati $\delta \mathbf{q}$ e sui controlli $\delta \mathbf{u}$ lungo la traiettoria sono arbitrarie, quindi ognuno dei loro coefficienti deve essere nullo in modo da avere un termine integrale nullo, e di conseguenza l'intero termine della varianza prima della funzione costo aumentata nulla.

$$\varphi(\mathbf{q}, \mathbf{u}) \Big|_{t=t_f} dt_f = 0 \quad \frac{\partial \mathcal{H}}{\partial \mathbf{q}} + \dot{\lambda}^T = 0 \quad \frac{\partial \mathcal{H}}{\partial \mathbf{u}} = 0 \tag{5.26}$$

Le equazioni 5.26 sono la condizione necessaria per l'ottimalità per il nostro esempio. Ricapitoliamo le nozioni già visto e di conseguenza come adoperare l'equazione 5.26 per trovare la traiettoria ottima. La terza equazione di 5.26 viene integrata in avanti per trovare i valori della funzione moltiplicatore, mentre la seconda viene integrata all'indietro per trovare di quanto devono essere aggiustate le condizioni iniziale e il controllo in modo da raffinare la soluzione sub-ottima che è stata fornita a priori dell'esecuzione dell'algoritmo. Inoltre i valori delle funzioni moltiplicatore per i vincoli sullo stato derivati e trasformati in vincoli sul controllo devono altresì essere trovati con la prima equazione.

5.4.3 Algoritmo di convergenza ottima

Una volta trovate le condizioni necessarie e di conseguenza le varie componenti delle funzioni moltiplicatore in modo che queste siano soddisfatte, è necessario compiere un ultimo passo qualora il controllo trovato non è soddisfacente.

Sia ε una costante arbitraria scelta a priori dell'esecuzione dell'algoritmo. Vogliamo ottenere un controllo \mathbf{u}^* , a partire da un controllo sub-ottimo \mathbf{u} , in modo che la differenza delle varie componenti temporali non ecceda per nessun $t \in [t_0, t_f]$ il valore ε . Ovvero deve valere la seguente espressione.

$$\mathbf{u}^*(t) - \mathbf{u}(t) \leq \varepsilon \quad \forall t \in [t_0, t_f] \quad (5.27)$$

Mediante la raffinazione del controllo, quest'ultimo può essere gradualmente migliorato, fino a tendere ad un valore oltre il quale lo supponiamo ottimo. In questa tesi usiamo l'approccio che prevede l'uso della discesa del gradiente per la raffinazione della traiettoria sub-ottima.

Per effettuare l'operazione della discesa del gradiente viene direttamente effettuato il calcolo mediante l'algoritmo fornito dalla libreria Matlab.

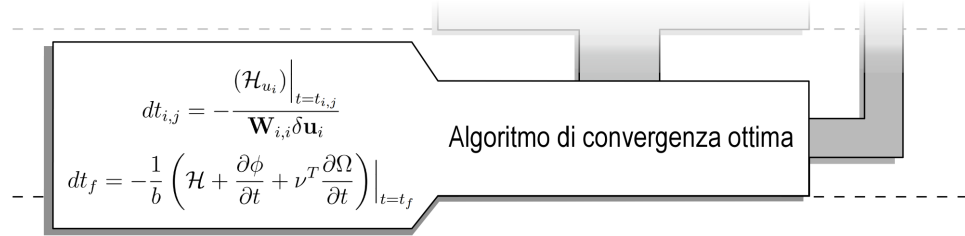


Figura 5.3: Schema concettuale dell'algoritmo della discesa del gradiente alternativo elaborato nel lavoro di Prof. Paolo Fiorini [48]

5.5 Riassunto

In questo capitolo abbiamo trovato un metodo numerico che converge ad una soluzione che soddisfa le condizioni necessarie delle equazioni 5.17

e 5.19. Abbiamo che la definizione di hamiltoniano \mathcal{H} soddisfa una delle condizioni necessarie 5.17. Le equazioni aggiunte sono soddisfatte poiché λ è una combinazione di λ singoli.

La traiettoria ottenuta con questo metodo numerico soddisfa la condizione necessarie di ottimalità; abbiamo quindi trovato un minimo locale al problema della traiettoria ottima con funzione costo corrispondente al tempo finale. Per ottenere una variante della soluzione, dove viene ad esempio adoperata meno potenza, possiamo utilizzare come funzione costo J il valore del controllo stesso u^2 . Un esempio che riporta questa situazione è stato trattato nello stato dell'arte di questa tesi 2.1.1.

Capitolo 6

Risultati delle simulazioni

In questo capitolo analizziamo i risultati della traiettoria generata nel caso di un quadricottero vincolato descritto dalle equazioni 3.26 e 3.27 del paragrafo 3.5. Il quadricottero vincolato rappresenta un caso particolare del quadricottero libero di muoversi nello spazio. In questo modo possiamo imporre determinati vincoli intermedi al quadricottero e analizzare il suo comportamento. Il comportamento del quadricottero in una dimensione può poi essere riportato al caso più generale di un quadricottero nelle tre dimensioni reali in un'analisi successiva.

Rivediamo brevemente la notazione che abbiamo adoperato. Nel caso di un quadricottero non vincolato, questo ha sei gradi di libertà: tre per la variazione dei tassi rotazionali, tre per traslazionali. Lo stato del quadricottero può di conseguenza essere rappresentato con le coordinate $(\mathbf{q}, \dot{\mathbf{q}}) = (x, y, z, \psi, \theta, \phi, \dot{x}, \dot{y}, \dot{z}, \Omega_1, \Omega_2, \Omega_3)$, dove $(x, y, z) \in \mathbb{R}^3$ sono le coordinate del centro di massa del quadricottero rispetto al sistema di riferimento T, $(\dot{x}, \dot{y}, \dot{z})$ sono le componenti della velocità del centro di massa, e $(\Omega_1, \Omega_2, \Omega_3) \in \mathbb{R}^3$ sono le componenti delle velocità angolari del quadricottero nel suo sistema di riferimento E.

La notazione che abbiamo usato per lo stato può essere adoperata per scrivere la lagrangiana del sistema, $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$, come avveniva nell'equazione 3.15. Trasformiamo la lagrangiana nell'hamiltoniana mediante la trasformata di Legendre. L'hamiltoniana ha come stato lo stato originale insieme al momento coniugato, $\mathcal{H}(\mathbf{q}, \mathbf{p})$, come avveniva nell'equazione 3.16.

Per la simulazione viene adoperata la libreria Matlab [52] che usa un principio matematicamente equivalente al principio di Pontryagin discusso nel capitolo 5.

6.1 Simulazione in assenza di ostacoli

Nella simulazione consideriamo lo stato composto di due variabili (x, θ) , la traslazione attorno all'asse x , e la rotazione del angolo pitch θ . In questo modo, nella prima simulazione, possiamo vedere l'evoluzione del modello vincolato in assenza di ostacoli intermedi. Partiamo da vincoli di frontiera prestabiliti così come li abbiamo adoperati nell'equazione 5.5.

$$\Gamma(t_0, \mathbf{q}(t_0)) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \Omega(t_f, \mathbf{q}(t_f)) = \begin{bmatrix} 1 \\ 0 \\ 2,025 \cdot 10^{-1} \\ 0 \end{bmatrix} \quad (6.1)$$

E con l'evoluzione del sistema $\mathbf{f}(\mathbf{q}, \mathbf{u})$ così come l'abbiamo definita nel capitolo 5.2 nell'equazione 5.4.

$$\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \mathbf{u}) = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \dot{p}_x \\ \dot{p}_\theta \end{bmatrix} = \begin{bmatrix} \frac{p_x}{m} \\ \frac{p_\theta}{I_3(1+\cos^2 \theta)} \\ -mg \tan \theta + \frac{\kappa_x}{m} p_x \\ \frac{p_\theta^2 \sin \theta \cos \theta}{I_3(1+\cos^2 \theta)^2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tau_\theta \quad (6.2)$$

Nella simulazione partiamo da una traiettoria sub-ottima generata con il metodo euristico da un insieme di prove sperimentali. La traiettoria viene mostrata in figura 6.1. Raffiniamo la traiettoria di partenza mediante l'uso del Principio Minimo di Pontryagin che abbiamo descritto nel teorema 1. Per applicare il principio dobbiamo trovare la funzione costo aumentata \bar{J} e la sua varianza prima $\delta \bar{J}$. Quest'ultima per i scopi della minimizzazione dovrà essere prossima a zero $\delta \bar{J} \approx 0$.

Per raffinare il controllo, e trovare il controllo in modo che la varianza prima della funzione costo aumentata tenda a zero, facciamo l'integrazione in avanti e troviamo i valori delle varie funzioni moltiplicatore in modo che rispettino l'equazione $\frac{\partial \mathcal{H}}{\partial \mathbf{u}} = 0$. In questo modo troviamo dei valori delle

funzioni moltiplicatore, dei quali possiamo infine trovare l'evoluzione $\dot{\lambda}^T = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}$.

Allo successiva iterazione usiamo le funzioni moltiplicatore aggiustate per l'integrazione in avanti e continuiamo con l'algoritmo finché non otteniamo un valore $\delta \bar{J}$ soddisfacente $\delta \bar{J} < \epsilon$.

6.1.1 Osservazioni

Possiamo osservare che otteniamo dei controlli ottimi diversi a seconda di stime iniziali diverse. Questo è un comportamento standard che incontriamo con quasi tutti gli algoritmi di ottimizzazione della traiettoria che si basano sulla risoluzione della condizione necessaria per l'ottimalità.

Nella figura 6.1 viene rappresentato il controllo che usiamo per generare la traiettoria del controllo ottimo $\mathbf{u}_1^*(t)$ rappresentata nella parte bassa della figura 6.2. Nella parte alta della figura vengono rappresentati i vari stati del sistema e la loro evoluzione temporale. Possiamo notare che affinché il quadricottero raggiunga la posizione $\mathbf{q}(1) = x = 1$, le variazioni degli angoli di θ sono molto piccole, e di conseguenza sono molto piccole anche le variazioni del momento coniugato. Il controllo che abbiamo disegnato rappresenta la funzione costo a tempo minimo, ovvero la funzione definita con la seguente espressione.

$$J = \int_0^{t_f} 1 \, dt = t_f \quad (6.3)$$

Nella figura 6.3 è rappresentato il controllo ottimo $\mathbf{u}_2^*(t)$ generato con il controllo sub-ottimo nullo. La funzione costo è la stessa dell'equazione 6.3. Possiamo notare che il controllo generato, nonostante l'evoluzione dello stato sia circa identica, è leggermente diverso dalla figura 6.2. Questo accade perché il controllo ottimo \mathbf{u}_1^* è stato generato a partire da un controllo sub-ottimo migliore del controllo nullo usato per generare \mathbf{u}_2^* , in quanto quest'ultimo ha una struttura (quasi) a bang-bang.

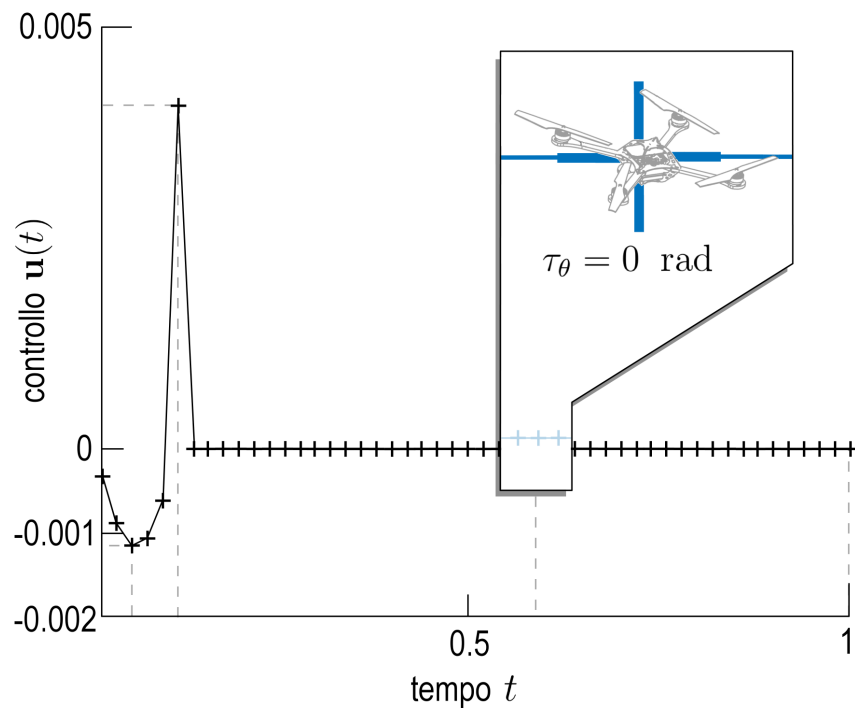


Figura 6.1: Plot del controllo sub-ottimo $\mathbf{u}(t)$ usato per l'algoritmo di ottimizzazione come stima iniziale

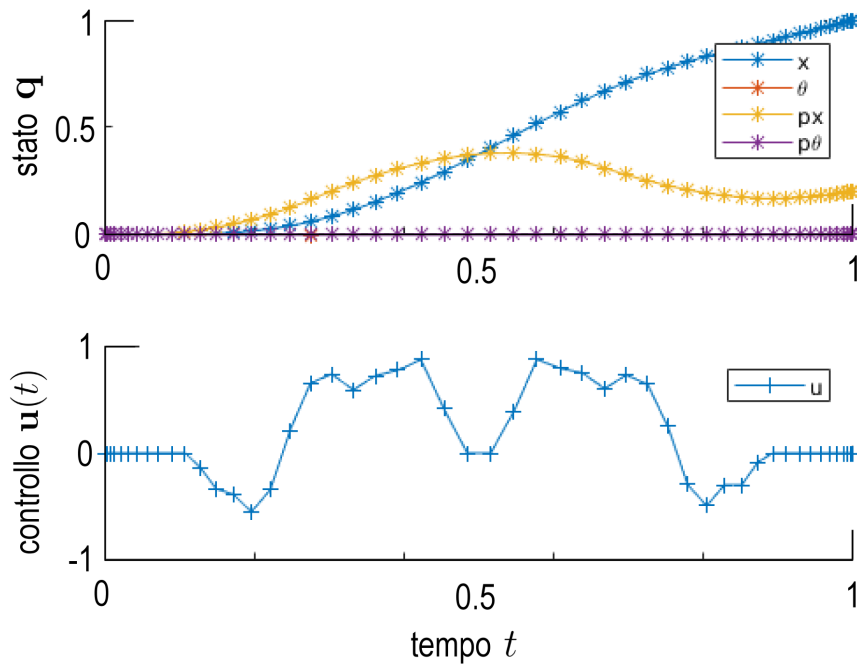


Figura 6.2: Plot dello stato \mathbf{q} e del controllo ottimo $\mathbf{u}_1^*(t)$ senza ostacoli trovato con l'algoritmo di minimizzazione che si basa sul principio di Pontryagin e sulla discesa del gradiente

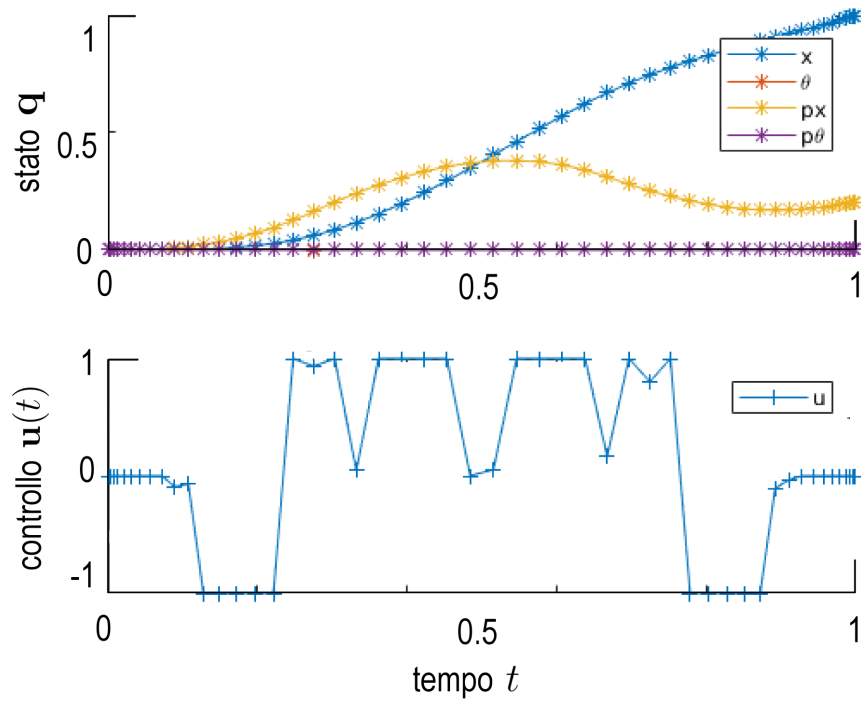


Figura 6.3: Plot dello stato \mathbf{q} e del controllo ottimo $\mathbf{u}_2^*(t)$ senza ostacoli trovato con l'algoritmo di minimizzazione che si basa sul principio di Pontryagin e sulla discesa del gradiente

6.2 Simulazione in presenza di ostacoli

6.2.1 Simulazione del braccio robotico

Prima di simulare l'assetto del quadricottero in presenza di ostacoli, è utile simulare un sistema dove è più facile notare il comportamento del sistema in prossimità di determinati ostacoli. Per la simulazione è stata adoperata la libreria Matlab.

Vogliamo minimizzare il problema del braccio robotico in due dimensioni con due vincoli circolari nel cammino descritto in [53]. Vengono riportate di seguito in questo ordine le condizioni di frontiera, l'evoluzione de sistema dinamico, e il funzionale costo da minimizzare.

$$\Gamma(t_0, \mathbf{q}(t_0)) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Omega(t_f, \mathbf{q}(t_f)) = \begin{bmatrix} 12 \cdot 10^{-1} \\ 16 \cdot 10^{-1} \end{bmatrix} \quad (6.4)$$

$$\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \mathbf{u}) = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} V \cdot \cos \theta \\ V \cdot \sin \theta \end{bmatrix} \quad (6.5)$$

$$J = \int_0^1 \left(\left(\frac{d^2 x}{dt^2} \right)^2 + \left(\frac{d^2 y}{dt^2} \right)^2 \right) dt \quad (6.6)$$

Inoltre vengono definiti i seguenti due vincoli intermedi.

$$(x - 4 \cdot 10^{-1})^2 + (y - 5 \cdot 10^{-1})^2 \geq 1 \cdot 10^{-1} \quad (6.7)$$

$$(x - 8 \cdot 10^{-1})^2 + (y - 15 \cdot 10^{-1})^2 \geq 1 \cdot 10^{-1} \quad (6.8)$$

La traiettoria ottima dello stato x e y che risulta dalla valutazione del sistema e raffinamento del controllo sub-ottimo $\theta = \pi/4$ costante è mostrata in figura 6.4.

6.2.2 Simulazione del quadricottero

In questa simulazione riprendiamo lo stesso problema descritto nel paragrafo 6.1, ma in questo caso aggiungiamo un ostacolo descritto dalla seguente equazione.

$$(t - 8279 \cdot 10^{-4})^2 + (x - 7525 \cdot 10^{-4})^2 \geq 1 \cdot 10^{-1} \quad (6.9)$$

Questo vincolo indica che lo stato x del sistema non può evolvere all'interno di una circonferenza con raggio 0,10 che si trova spostata dall'asse delle x all'altezza 0,75 dall'asse del tempo t alla larghezza 0,83.

In questa simulazione, come nella precedente, usiamo la funzione costo che rappresenta il tempo minimo $J = t_f$.

Il controllo sub-ottimo da cui viene generato il controllo ottimo è il controllo che abbiamo già adoperato nella figura 6.1.

6.2.3 Osservazioni

Possiamo notare che la traiettoria generata permette al sistema di evolvere in modo da evitare l'ostacolo. Infatti, il parametro dello stato x , che rappresenta lo spostamento nello spazio rispetto al tempo, evolve al di sopra dell'ostacolo cercando di evitarlo.

Possiamo notare che ci sono certe traiettorie che non possono essere valutate e di cui l'evoluzione non è possibile in presenza di determinati ostacoli. Questo succede soprattutto qualora abbiamo a che fare con dei vincoli sul controllo. Diffatti la traiettoria generata in modo da evitare l'ostacolo descritto dall'equazione 6.9, è ai limiti degli estremi dei vincoli sul controllo ($-1 \leq u \leq 1$), e presenta una struttura quasi a bang-bang.

La struttura del controllo che otteniamo con dei vincoli sullo stato più rilassati, ha una struttura molto più smusata. La prima parte della figura 6.6 mostra il controllo con dei vincoli $-10 \leq u \leq 10$, la seconda invece mostra il controllo con dei vincoli $-50 \leq u \leq 100$. Notiamo che il comportamento è esattamente come ci attendiamo. L'evoluzione in questi casi non viene rappresentata, in quanto è all'incirca uguale all'evoluzione della figura 6.5.

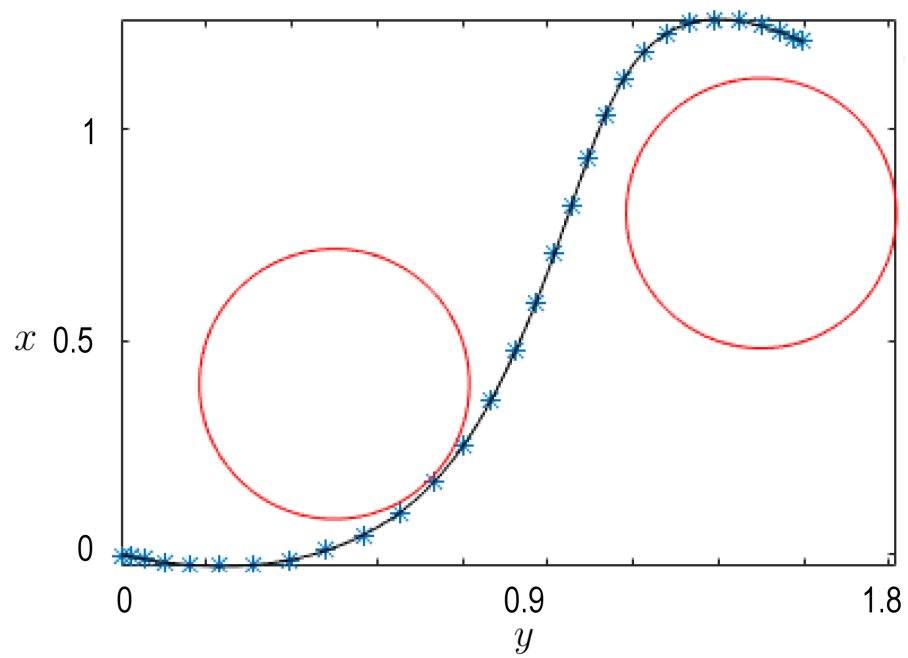


Figura 6.4: Plot dello stato \mathbf{q} della traiettoria ottima del braccio robotico

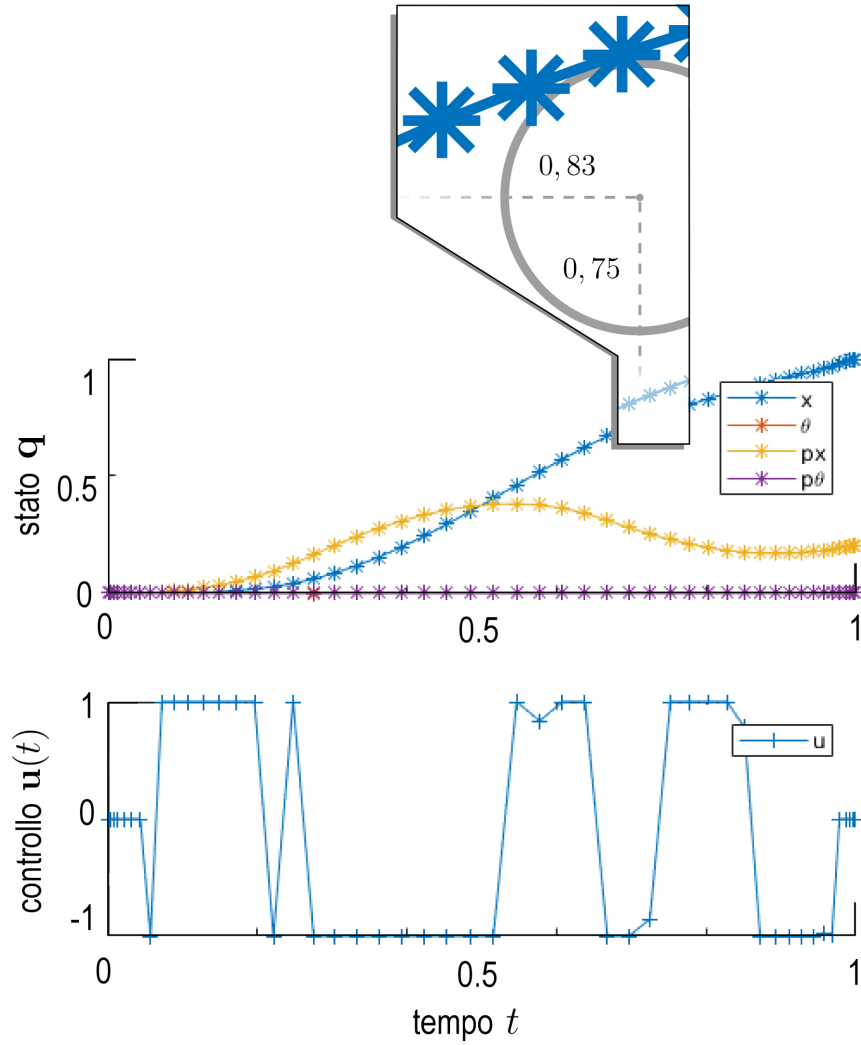


Figura 6.5: Plot dello stato \mathbf{q} e del controllo ottimo $\mathbf{u}(t)$ in presenza di un ostacolo (circonferenza di raggio 0,1) trovato con l'algoritmo di minimizzazione che si basa sul principio di Pontryagin e sulla discesa del gradiente

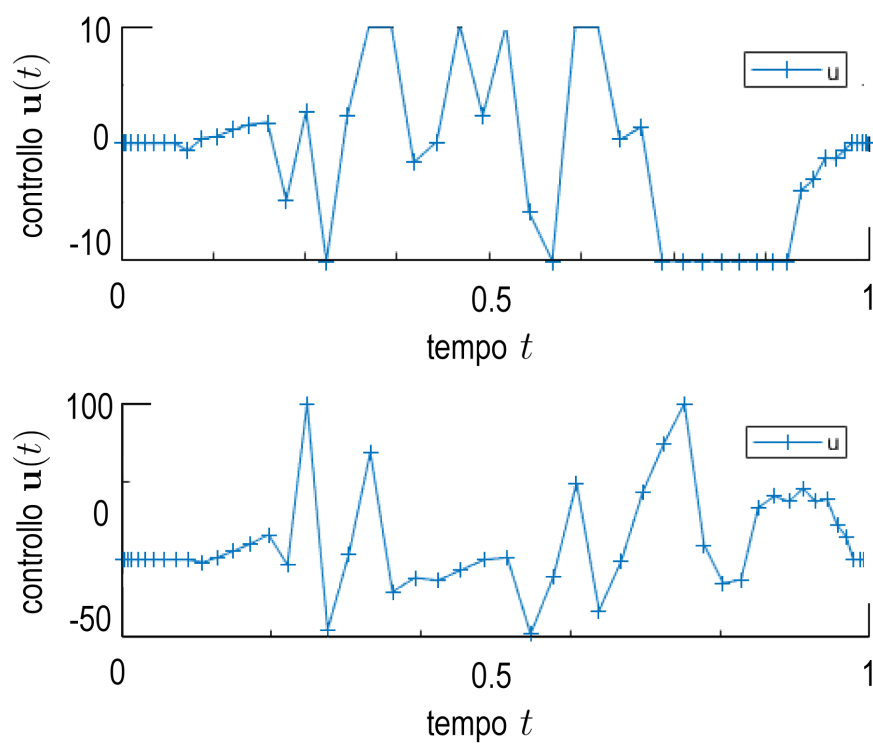


Figura 6.6: Plot dei controlli al variare dei vincoli. Con vincoli sul controllo più larghi, si ha un controllo più rilassato

Capitolo 7

Conclusioni e lavoro futuro

In questo capitolo rivediamo brevemente cosa abbiamo ottenuto con questa tesi, e quali possibili sviluppi futuri ha il lavoro che abbiamo trattato. Vedremo in particolare che alcuni aspetti matematici e implementativi sono rimasti aperti, e meritano una trattazione approfondita successiva.

7.1 Presentazione del lavoro

Con questa tesi abbiamo generato delle traiettorie di controllo ottime per quadricotteri, mediante una libreria di ottimizzazione che è stata sviluppata in C++ per consentire la portabilità sull'elettronica a bordo del quadricottero. L'uso dei quadricotteri come veicoli aerei miniaturizzati ha avuto il picco nello sviluppo solo recentemente. Gli algoritmi efficienti che sono disponibili in letteratura per la generazione di una traiettoria ottima si basano principalmente sui cosiddetti metodi diretti di collocazione. Tuttavia la letteratura attualmente disponibile per la generazione di traiettorie per quadricotteri usa prevalentemente metodi diretti, detti anche metodi di collocazione (o metodi diretti di collocazione). In questa tesi abbiamo implementati i metodi indiretti. Quest'ultimi sono matematicamente molto più accurati dei metodi diretti.

Per la trattazione dei metodi indiretti, abbiamo implementato una funzione costo che rappresenta il fenomeno che si vuole minimizzare nella ricerca della traiettoria ottima. Nel paragrafo 2, abbiamo definito lo stato dell'arte

dei metodi indiretti per la ricerca della traiettoria ottima. Abbiamo visto che la stragrande maggioranza degli algoritmi oggi presenti in letteratura, che permettono di risolvere i problemi di ottimizzazione con metodi indiretti, sono basati sul approccio adoperato nel libro di Bryson e Ho [4]. Abbiamo quindi introdotto un problema d'esempio, il problema del blocco in movimento da un punto ad un altro e abbiamo descritto come funziona la ricerca della traiettoria ottima mediante la funzione costo aumentata, la sua varianza prima, e la ricerca della condizione necessaria. Abbiamo quindi visto che per la ricerca di un minimo è sufficiente soddisfare la condizione necessaria per l'ottimalità, ovvero è sufficiente trovare i valori delle funzioni moltiplicatore, che abbiamo definito come λ , in modo che la varianza prima della funzione costo aumentata sia nulla $\delta \bar{J} \approx 0$. Nell'esempio che abbiamo usato per la spiegazione dello stato dell'arte abbiamo visto una possibile soluzione analitica, che è alla base della risoluzione dei problemi della traiettoria ottima presenti in letteratura.

Nel capitolo 3 abbiamo introdotto le nozioni base che servono a comprendere la struttura dinamica di un quadricottero e come avviene il suo spostamento nello spazio. Abbiamo visto che la posizione e i gradi di libertà dei propulsori in un veicolo aereo determinano la dinamica e la cinematica dello stesso. Abbiamo usato gli angoli di Eulero per gli angoli di roll, pitch e yaw già ben consolidati e facili da visualizzare. L'angolo di roll è definito come una rotazione attorno all'asse x . L'angolo di pitch come una rotazione attorno all'asse y , mentre l'angolo di yaw come una rotazione all'asse z . Abbiamo visto che attraverso queste tre rotazioni, l'assetto del sistema di riferimento può essere messo in relazione con un frame inerziale. Siamo passati all'analisi della sottoattuazione e abbiamo visto che un corpo rigido libero in volo ha sei gradi di libertà. Può traslare e ruotare lungo i tre assi. Tuttavia con un quadricottero, nonostante abbiamo sei gradi di libertà, abbiamo a che fare con un sistema sottoattuato in quanto esso ha solo quattro attuatori. I quattro propulsori che si compongono di eliche posizionate in modo complementare agli estremi degli assi del quadricottero. Abbiamo definito l'input di controllo e come questo deve venire vincolato in modo da ottenere una traiettoria ottima del controllo. Siamo passati all'analisi di un modello vincolato del quadricottero, in quanto abbiamo visto che per i scopi della tesi

è sufficiente vedere un numero limitato degli stati per poi successivamente estendere il problema ad un numero di stati maggiore. Abbiamo visto come trasformare il sistema da una rappresentazione lagrangiana mediante la trasformata di Legendre nella corrispondente rappresentazione hamiltoniana. In questo modo possiamo rappresentare anche i momenti coniugati relativi ai vari stati e possiamo, ad esempio, imporre dei vincoli sui momenti. Questo ci consente di valutare l'energia acquisita dal quadricottero, con l'analisi del momento rispetto all'asse x in determinati istanti temporali. Ci permette anche l'analisi del momento angolare e quindi dell'energia angolare acquisita dal sistema in determinati istanti temporali.

Nel capitolo 4 descriviamo i metodi diretti di collocazione, che sono il diretto concorrente dei metodi indiretti descritti in questa tesi e vengono spesso adoperati nella letteratura. Abbiamo visto che la principale caratteristica di un metodo di collocazione è il fatto che discretizza il problema di ottimizzazione della traiettoria, normalmente mediante la conversione del problema in un programma non lineare con la trascrizione. Abbiamo ripreso il problema del blocco in movimento che abbiamo descritto nel paragrafo 2.1.1 del capitolo 2, e lo abbiamo adattato a questo metodo. L'operazione di discretizzazione ci ha fornito un insieme di punti di collocazione; successivamente abbiamo convertito la dinamica del sistema continua in un insieme di vincoli che possiamo applicare allo stato e al controllo nei punti di collocazione. Abbiamo quindi visto due possibili metodi, un metodo più semplice e più approssimato che si basa sulla regola del trapezio, la collocazione trapezoidale, e il metodo di Hermite-Simpson, simile alla collocazione trapezoidale, ma fornisce una soluzione che è di ordine maggiore ed è più accurata.

Nel capitolo 5 abbiamo visto il cuore di questa tesi, i metodi indiretti per la risoluzione del problema della traiettoria del controllo ottima, a partire da una stima iniziale sub-ottima. La ricerca della soluzione ottima si basa sul Principio Minimo di Pontryagin, che abbiamo visto con il teorema 1: troviamo la funzione costo aumentata \bar{J} e la sua varianza prima $\delta\bar{J}$. Per i scopi della minimizzazione dovrà essere prossima a zero $\delta\bar{J} \approx 0$.

Per raffinare il controllo, dobbiamo soddisfare l'equazione $\frac{\partial \mathcal{H}}{\partial \mathbf{u}} = 0$. Ovvero dobbiamo trovare il controllo in modo che la varianza prima della funzione costo aumentata tenda a zero, facciamo l'integrazione in avanti e troviamo i

valori delle varie funzioni moltiplicatore λ . In questo modo troviamo dei valori delle funzioni moltiplicatore, dei quali possiamo infine trovare l'evoluzione $\dot{\lambda}^T = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}$.

Alla successiva iterazione dell'algoritmo di ottimizzazione, usiamo le funzioni moltiplicatore aggiustate per l'integrazione in avanti. continuiamo con l'algoritmo finché non otteniamo un valore $\delta \bar{J}$ soddisfacente $\delta \bar{J} < \epsilon$, dove ϵ è un valore arbitrario definito a priori dell'esecuzione dell'algoritmo di ottimizzazione.

7.2 Lavoro futuro

La trattazione della generazione della traiettoria ottima mediante i metodi indiretti in questa tesi può essere estesa al caso generale che include tutti i stati di cui si compone il quadricottero. La libreria sviluppata è abbastanza flessibile per consentire una simile modifica. Tuttavia non abbiamo implementato un metodo facile per la trasformazione del sistema tra una rappresentazione lagrangiana ad una hamiltoniana, e attualmente è richiesto che questi dati vengano forniti all'interno della libreria, che pertanto non è in grado di fare la trasformazione a sé. Inoltre non sono state fatte analisi approfondite su come il momento coniugato può tornare utile per la risoluzione di problemi relativi al passaggio del quadricottero attraverso finestre limitate. Abbiamo semplicemente visto il caso generale di un vincolo sullo stato x e ne abbiamo trovato la traiettoria del controllo con l'approccio del Principio Minimo di Pontryagin e della discesa del gradiente per aggiustare i parametri iniziali.

In primo luogo può quindi essere fatta un'analisi approfondita sulla fattibilità della traiettoria in presenza di ostacoli, che in questa tesi non viene gestita. L'algoritmo semplicemente calcola la traiettoria, anche se questa è in disaccordo con l'ostacolo, qualora non esiste un controllo che è in grado di soddisfare il vincolo. Una trattazione matematica approfondita è necessaria per risolvere anche questo punto, ma per i scopi di questa tesi è stato omissso.

Una possibile evoluzione di questa tesi è quella della generalizzazione della libreria sul controllore del quadricottero e sulla ricerca dinamica della traiettoria ottima. In questo modo una traiettoria sub-ottima trovata con meto-

di computazionalmente meno onerosi può essere notevolmente migliorata e usata come la traiettoria sub-ottima in input a questo metodo.

C'è un'altra problematica che concerne i principi matematici stessi adoperati in questa tesi. Questa problematica, descritta nelle parti successive di questo paragrafo, può essere ulteriormente elaborata nelle evoluzioni successive del lavoro di questa tesi: con il Principio Minimo di Pontryagin, che abbiamo descritto in dettaglio nel capitolo 5, abbiamo trovato un ottimo locale, mediante la ricerca della condizione necessaria. Tuttavia non abbiamo verificato la condizione sufficiente e pertanto non sappiamo rispondere alla domanda di quale tipologia di minimo si tratta. Inoltre non abbiamo trattato il problema del minimo globale. Abbiamo trovato un minimo, che potrebbe essere un minimo globale, e solitamente con i problemi che trattiamo lo è, ma anche non esserlo. È quindi necessario valutare tra tutti i possibili minimi e vedere quale di loro sia migliore.

Il codice che abbiamo presentato ha inoltre un ampio margine di possibili ottimizzazione e miglioramenti. In particolare manca un'interfaccia e delle versioni ottimizzate e precompilate che possono essere adoperate sull'elettronica a bordo.

Bibliografia

- [1] C. F. Liew, D. DeLatte, N. Takeishi, and T. Yairi, “Recent developments in aerial robotics: An survey and prototypes overview,” *arXiv preprint arXiv:1711.10085*, 2017.
- [2] M. Hehn and R. D’Andrea, “Quadrocopter trajectory generation and control,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1485–1491, 2011.
- [3] E. Hernandez-Martinez, G. Fernandez-Anaya, E. Ferreira, J. Flores-Godoy, and A. Lopez-Gonzalez, “Trajectory tracking of a quadcopter uav with optimal translational control,” *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 226–231, 2015.
- [4] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.
- [5] A. Pal, A. Pancholy, and P. Dwivedi, “Application of sub-optimal mpqc guidance for trajectory optimization of various aerospace vehicles with terminal angle constraint,” *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 58–63, 2018.
- [6] S. M. Roberts and J. S. Shipman, “Two-point boundary value problems: shooting methods,” 1972.
- [7] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.
- [8] M. A. Patterson and A. V. Rao, “Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian

- quadrature collocation methods and sparse nonlinear programming,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, no. 1, pp. 1–41, 2014.
- [9] A. V. Rao, “A survey of numerical methods for optimal control,” *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.
- [10] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*, vol. 19. Siam, 2010.
- [11] S. Bouabdallah, P. Murrieri, and R. Siegwart, “Design and control of an indoor micro quadrotor,” in *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*, vol. 5, pp. 4393–4398, IEEE, 2004.
- [12] O. Egeland and J. Gravdahl, “Modeling and simulation for automatic control. marine cybernetics, trondheim, norway,” tech. rep., ISBN 82-92356-01-0, 2002.
- [13] J.-J. Slotine and L. Weiping, “Adaptive manipulator control: A case study,” *IEEE transactions on automatic control*, vol. 33, no. 11, pp. 995–1003, 1988.
- [14] B. Paden and R. Panja, “Globally asymptotically stable ‘pd+’ controller for robot manipulators,” *International Journal of Control*, vol. 47, no. 6, pp. 1697–1712, 1988.
- [15] S. Ramirez *et al.*, “Passivity-based control of euler-lagrange systems,” 1998.
- [16] J. Mei, W. Ren, and G. Ma, “Distributed coordinated tracking with a dynamic leader for multiple euler-lagrange systems,” *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1415–1421, 2011.
- [17] L. Luyckx, M. Loccupier, and E. Noldus, “On the design of nonlinear controllers for euler-lagrange systems,” *NONLINEAR DYNAMICS AND SYSTEMS THEORY*, vol. 1, no. 1, pp. 97–107, 2001.

- [18] A. Loria, “Uniform global position feedback tracking control of mechanical systems without friction,” in *American Control Conference (ACC), 2013*, pp. 5722–5727, IEEE, 2013.
- [19] E. Kyrkjebo and K. Y. Pettersen, “Output synchronization control of euler-lagrange systems with nonlinear damping terms,” in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC’05. 44th IEEE Conference on*, pp. 4951–4957, IEEE, 2005.
- [20] A. M. Bloch, N. E. Leonard, and J. E. Marsden, “Controlled lagrangians and the stabilization of mechanical systems. i. the first matching theorem,” *IEEE Transactions on automatic control*, vol. 45, no. 12, pp. 2253–2270, 2000.
- [21] M. Reyhanoglu, A. van der Schaft, N. H. McClamroch, and I. Kolmanovsky, “Dynamics and control of a class of underactuated mechanical systems,” *IEEE Transactions on Automatic Control*, vol. 44, no. 9, pp. 1663–1671, 1999.
- [22] A. P. Aguiar and J. P. Hespanha, “Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty,” *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1362–1379, 2007.
- [23] A. P. Aguiar and A. M. Pascoal, “Dynamic positioning and way-point tracking of underactuated auvs in the presence of ocean currents,” in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 2, pp. 2105–2110, IEEE, 2002.
- [24] F. Arrichiello, S. Chiaverini, and T. I. Fossen, “Formation control of underactuated surface vessels using the null-space-based behavioral control,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 5942–5947, IEEE, 2006.
- [25] E. Borhaug and K. Y. Pettersen, “Adaptive way-point tracking control for underactuated autonomous vehicles,” in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC’05. 44th IEEE Conference on*, pp. 4028–4034, IEEE, 2005.

- [26] M. Breivik and T. I. Fossen, “Guidance-based path following for autonomous underwater vehicles,” in *OCEANS, 2005. Proceedings of MTS/IEEE*, pp. 2807–2814, IEEE, 2005.
- [27] F. Bullo and K. M. Lynch, “Kinematic controllability for decoupled trajectory planning in underactuated mechanical systems,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 402–412, 2001.
- [28] I. Fantoni and R. Lozano, *Non-linear control for underactuated mechanical systems*. Springer Science & Business Media, 2002.
- [29] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.
- [30] J.-M. Godhavn and O. Egeland, “Attitude control of an underactuated satellite,” in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 4, pp. 3986–3987, IEEE, 1995.
- [31] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, “A control approach for thrust-propelled underactuated vehicles and its application to vtol drones,” *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1837–1853, 2009.
- [32] D. Lee, T. C. Burg, B. Xian, and D. M. Dawson, “Output feedback tracking control of an underactuated quad-rotor uav,” in *American Control Conference, 2007. ACC’07*, pp. 1775–1780, IEEE, 2007.
- [33] T. Lee, M. Leoky, and N. H. McClamroch, “Geometric tracking control of a quadrotor uav on $se(3)$,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 5420–5425, IEEE, 2010.
- [34] N. E. Leonard, “Periodic forcing, dynamics and control of underactuated spacecraft and underwater vehicles,” in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 4, pp. 3980–3985, IEEE, 1995.

- [35] M. López-Martínez, J. Acosta, and J. Cano, “Non-linear sliding mode surfaces for a class of underactuated mechanical systems,” *IET Control Theory & Applications*, vol. 4, no. 10, pp. 2195–2204, 2010.
- [36] R. Olfati-Saber, *Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles*. PhD thesis, 2001.
- [37] K. Y. Pettersen and O. Egeland, “Time-varying exponential stabilization of the position and attitude of an underactuated autonomous underwater vehicle,” *IEEE Transactions on Automatic Control*, vol. 44, no. 1, pp. 112–115, 1999.
- [38] J. Refsnes, A. J. Sørensen, and K. Y. Pettersen, “Output feedback control of a slender body underactuated auv with experimental results,”
- [39] A. Roberts and A. Tayebi, “Adaptive position tracking of vtol uavs,” *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 129–142, 2011.
- [40] M. W. Spong, “Underactuated mechanical systems,” in *Control problems in robotics and automation*, pp. 135–150, Springer, 1998.
- [41] A. Tayebi and S. McGilvray, “Attitude stabilization of a vtol quadrotor aircraft,” *IEEE Transactions on control systems technology*, vol. 14, no. 3, pp. 562–571, 2006.
- [42] Y. Yoshimura, T. Matsuno, and S. Hokamoto, “Position and attitude control of an underactuated satellite with constant thrust,” in *AIAA Guidance, Navigation, and Control Conference*, p. 6352, 2011.
- [43] K. Wichlund, O. Sordalen, and O. Egeland, “Control properties of underactuated vehicles,” in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 2, pp. 2009–2014, IEEE, 1995.
- [44] P. C. Hughes, *Spacecraft attitude dynamics*. Courier Corporation, 2012.

- [45] D. S. Bernstein, *Matrix mathematics: Theory, facts, and formulas with application to linear systems theory*, vol. 41. Princeton University Press Princeton, 2005.
- [46] G. Cai, B. M. Chen, and T. H. Lee, *Unmanned rotorcraft systems*. Springer Science & Business Media, 2011.
- [47] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, “Numerical recipes in c++,” *The art of scientific computing*, 1992.
- [48] P. Fiorini, *Robot motion planning among moving obstacles*. PhD thesis, Los Angeles, CA, USA, 1995. AAI0803036.
- [49] G. Leitmann, “An introduction to optimal control,” 1966.
- [50] J. L. Speyer, *Optimization and control of nonlinear systems with inflight constraints*. Harvard University, 1968.
- [51] D. H. Jacobson, M. M. Lele, and J. L. Speyer, “New necessary conditions of optimality for control problems with state-variable inequality constraints,” *Journal of mathematical analysis and applications*, vol. 35, no. 2, pp. 255–284, 1971.
- [52] E. Rutquist, M. Edval, “PROPT - Matlab Optimal Control Software.” <https://tomopt.com/>.
- [53] R. Bhattacharya, “OPTRAGEN 1.0: A MATLAB Toolbox for Optimal Trajectory Generation,” College Station, TX 77843-3141, USA, 2006.