

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Dipartimento di Scienze Fisiche, Informatiche e Matematiche

CORSO DI LAUREA TRIENNALE IN MATEMATICA

---

**Metodi di collocazione diretta per problemi di  
ottimizzazione della traiettoria**

---

*Relatore*

**Prof. Marco PRATO**

*Candidato*

**Alessandro PIERRO**

Anno Accademico 2020-2021



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Il problema dell'ottimizzazione di una traiettoria . . . . .	1
1.2	Un problema test . . . . .	3
1.3	L'approccio del calcolo variazionale . . . . .	4
1.4	Panoramica dei metodi numerici . . . . .	7
<b>2</b>	<b>Metodi risolutivi</b>	<b>9</b>
2.1	Metodi di collocazione diretta . . . . .	9
2.1.1	Metodo del punto medio . . . . .	11
2.1.2	Metodo del trapezio . . . . .	13
2.1.3	Metodo di Hermite-Simpson . . . . .	14
2.2	Tecniche di miglioramento . . . . .	17
2.2.1	Scelta della guess iniziale . . . . .	17
2.2.2	Strategie per sistemi dinamici stiff . . . . .	18
2.2.3	Analisi dell'accuratezza . . . . .	19
<b>3</b>	<b>Applicazioni ed esempi</b>	<b>23</b>
3.1	Esempio 1: Controllo con lavoro minimo . . . . .	23
3.2	Esempio 2: La catenaria come problema di controllo . . . . .	25
3.3	Esempio 3: Model Predictive Contouring Control . . . . .	27
<b>4</b>	<b>Riflessioni conclusive</b>	<b>31</b>
	<b>Bibliografia</b>	<b>33</b>



## Capitolo 1

# Introduzione

Nel giugno 1696, Jhoann Bernoulli pubblicò un problema che, per quanto apparentemente semplice, non era ancora di facile soluzione:

Dati due punti A e B in un piano verticale, si determini la traiettoria del punto mobile M che, partendo da A e sotto la sola influenza del proprio peso, arriva in B nel minor tempo possibile.

Lo studio di questo problema, detto **problema della brachistocrona**, attirò l'interesse di numerosi matematici dell'epoca (tra gli altri, Newton, Leibniz, de l'Hôpital e Jakob Bernoulli, il fratello di Jhoann) che, negli anni seguenti, continuarono a dedicarsi a problemi simili. Si diede così avvio ad una nuova branca della matematica: il controllo ottimo. La teoria del controllo ottimo si concentra su problemi di ottimizzazione di funzionali, soggetti a vincoli sia differenziali che algebrici. [16]

L'obiettivo di questa Tesi è dare un'introduzione ai problemi di ottimizzazione della traiettoria, mostrare come questi possano essere risolti per via numerica con i metodi di collocazione diretta, ed infine valutare le proprietà dei metodi presentati applicandoli ad alcuni problemi di esempio.

Tutto il codice utilizzato nella trattazione di questa Tesi è accessibile online all'indirizzo <https://github.com/AlessandroPierro/trajectory-optimization>.

## 1.1 Il problema dell'ottimizzazione di una traiettoria

Si definiscono ora in maniera più formale i problemi di cui si occupa la teoria del controllo ottimo. Si consideri un sistema dinamico soggetto all'equazione differenziale vettoriale

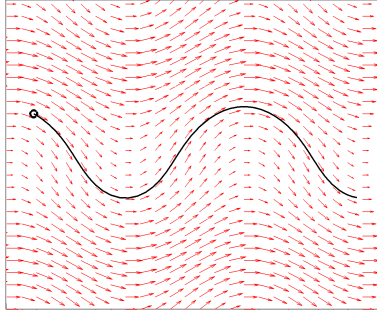
$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t); t) \quad (1.1)$$

dove  $\mathbf{x} : \mathbb{R} \rightarrow \mathbb{R}^n$  denota lo **stato del sistema** e  $\mathbf{u} : \mathbb{R} \rightarrow \mathbb{R}^m$  il **controllo** applicabile al sistema. Sotto le opportune ipotesi di regolarità, e fissate le condizioni al contorno  $\mathbf{x}(t_I) = \mathbf{x}_I$  e  $\mathbf{x}(t_F) = \mathbf{x}_F$ , il sistema ammetterà una, nessuna o molteplici soluzioni, al variare del controllo  $\mathbf{u}(t)$ . Si introduce il seguente **funzionale di costo**:

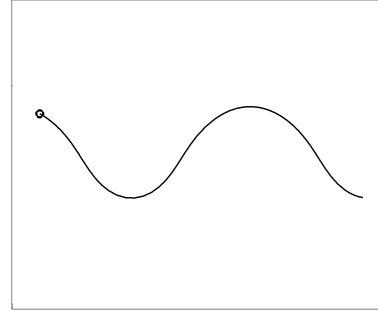
$$\mathcal{E}(t_I, t_F, \mathbf{x}(t), \mathbf{u}(t)) = J(t_I, t_F, \mathbf{x}(t_I), \mathbf{x}(t_F)) + \int_{t_I}^{t_F} w(t, \mathbf{x}(t), \mathbf{u}(t)) dt. \quad (1.2)$$

dove si dirà **termine di Mayer** il primo addendo  $J$  e **termine di Lagrange** l'addendo integrale. Fissato il controllo  $\mathbf{u}(t)$  in un intervallo  $[t_I, t_F]$ , e detta  $\bar{\mathbf{x}} : [t_I, t_F] \rightarrow \mathbb{R}^n$  l'associata soluzione del sistema (1.1), allora il funzionale  $\mathcal{E}$  associa alla traiettoria  $(\bar{\mathbf{x}}(t), \mathbf{u}(t))$  un valore numerico  $\mathcal{E}(t_I, t_F, \bar{\mathbf{x}}(t), \mathbf{u}(t)) \in \mathbb{R}$ .

La teoria del controllo ottimo si pone l'obiettivo di determinare un controllo  $\mathbf{u}$  che minimizza il funzionale  $\mathcal{E}$  soddisfacendo opportuni vincoli, sia differenziali (come il sistema (1.1)) che algebrici (come le condizioni al contorno). Tale controllo  $\mathbf{u}$



(A) Closed-loop solution



(B) Open-loop solution

potrà essere espresso in due diverse forme, che danno vita ad altrettante classi di problemi. In particolare, si parlerà di **soluzione closed-loop** o **policy** nel caso in cui si ricerchi il controllo  $\mathbf{u}(\mathbf{x})$  in funzione dello stato del sistema  $\mathbf{x}$ . Nel caso in cui si ricerchi  $\mathbf{u}(t)$ , ossia un controllo dipendente esclusivamente dal tempo, si parlerà di **soluzione open-loop**. In quest'ultimo caso rientra la classe di problemi oggetto di questa Tesi, i problemi di ottimizzazione della traiettoria.

**Definizione 1.** Si dirà **problema di ottimizzazione della traiettoria** il seguente problema di ricerca di minimo vincolato:

$$\min_{t_I, t_F, \mathbf{x}(t), \mathbf{u}(t)} J(t_I, t_F, \mathbf{x}(t_I), \mathbf{x}(t_F)) + \int_{t_I}^{t_F} w(t, \mathbf{x}(t), \mathbf{u}(t)) dt \quad (1.3)$$

eventualmente soggetto ad uno o più dei seguenti vincoli:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); t) \quad \text{dinamica del sistema} \quad (1.4)$$

$$\mathbf{x}(t_I) = \mathbf{x}_I \quad \mathbf{x}(t_F) = \mathbf{x}_F \quad \text{condizioni al contorno} \quad (1.5)$$

$$\mathbf{g}(t_I, t_F, \mathbf{x}(t_I), \mathbf{x}(t_F)) \leq 0 \quad \text{vincoli al contorno} \quad (1.6)$$

$$\mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t)) \leq 0 \quad \text{vincoli sulla traiettoria} \quad (1.7)$$

$$t_{min} \leq t_I \leq t_F \leq t_{max} \quad \text{limite sul tempo} \quad (1.8)$$

$$\mathbf{x}_{min} \leq \mathbf{x}(t) \leq \mathbf{x}_{max} \quad \text{limite sullo stato} \quad (1.9)$$

$$\mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max} \quad \text{limite sul controllo} \quad (1.10)$$

*Osservazione.* Nonostante la definizione di problema di ottimizzazione della traiettoria sia applicabile anche ai casi discontinui, nel seguito si supporrà che tutte le funzioni coinvolte siano di classe almeno  $\mathcal{C}^2$  sul loro dominio.

Dal punto di vista formale, il problema di ottimizzazione della traiettoria è un problema di minimizzazione in uno spazio funzionale, soggetto a vincoli sia algebrici che differenziali. La difficoltà nella risoluzione di tali problemi è dovuta al fatto che, senza ulteriori ipotesi, lo spazio di ricerca (tipicamente lo spazio delle funzioni vettoriali di classe  $\mathcal{C}^2$ ) risulta essere  $\infty$ -dimensionale.

I metodi descritti nel Capitolo 2.1, permetteranno di mappare tali problemi in problemi di programmazione non-lineare, ossia i problemi trattati classicamente dall'ottimizzazione numerica. Questo ne semplifica notevolmente la trattazione in

quanto si passa da un dominio  $\infty$ -dimensionale (quello delle funzioni) a un dominio di dimensione finita, ossia  $\mathbb{R}^N$ .

Prima di proseguire con la trattazione dei metodi di collocazione diretta, principale argomento di questa Tesi, si propone un problema di esempio e si danno i principali risultati teorici nell'ambito del controllo ottimo.

## 1.2 Un problema test

Si consideri un sistema fisico costituito da una massa puntiforme  $m$  vincolata a muoversi su una retta con attrito trascurabile. Fissata un'origine ed una parametrizzazione con ascissa curvilinea  $s$ , la dinamica del sistema è esprimibile come:

$$\begin{cases} \dot{s} = v \\ \dot{v} = u \end{cases} \quad (1.11)$$

dove il termine  $u$  denota un controllo esercitabile sulla massa puntiforme. Si suppongano inoltre fissate le seguenti condizioni al contorno:

$$s(t_I) = s_I \quad s(t_F) = s_F \quad (1.12)$$

$$v(t_I) = v_I \quad v(t_F) = v_F \quad (1.13)$$

e il seguente vincolo sul controllo  $u$ :

$$u_{\min} \leq u(t) \leq u_{\max} \quad \forall t \in [t_I, t_F]. \quad (1.14)$$

Si vuole determinare il controllo  $u^*$  che, soddisfacendo i vincoli appena definiti, minimizza l'utilizzo del controllo, ossia il funzionale:

$$\mathcal{W}(u(t)) = \int_{t_I}^{t_F} u^2(t) dt \quad (1.15)$$

Non entrando ancora nel dettaglio di come affrontare la soluzione di tale problema, si mostra il confronto tra la soluzione analitica<sup>1</sup> e la soluzione determinata sperimentalmente (il codice è contenuto in `problema_test.py`) mediante il metodo di collocazione dei trapezi, trattato nella Sezione 2.1.2. Come è possibile vedere dal grafico in Figura 1.2, la traiettoria della posizione  $s^*(t)$  fornita dalla soluzione discretizzata segue fedelmente la soluzione esatta  $s(t) = -2t^3 + 3t^2$ .<sup>2</sup>

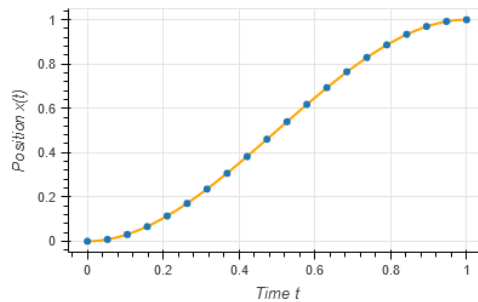


FIGURA 1.2: Posizione  $s(t)$

<sup>1</sup>La derivazione completa della soluzione analitica è al di fuori dello scopo di questa Tesi e può essere trovata, tra gli altri testi, in [12].

<sup>2</sup>Nei grafici, è riportata in giallo la soluzione esatta e in blu la soluzione numerica.

Analogamente, la soluzione numerica per il controllo  $u^* = (u_1, \dots, u_{20})$ , fatta eccezione per una leggera deviazione agli estremi del dominio, coincide con la soluzione esatta  $u(t) = -12t + 6$ .

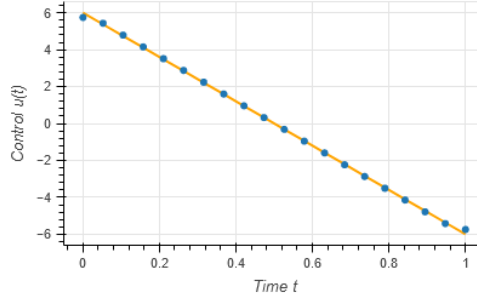


FIGURA 1.3: Controllo  $u(t)$

### 1.3 L'approccio del calcolo variazionale

L'approccio variazionale ai problemi di controllo ottimo si è sviluppato prevalentemente nell'ultimo secolo con i contributi di Lev Pontryagin (1908-1988) e Richard E. Bellman (1920-1984). In questo paragrafo si mostrerà come il **principio del massimo di Pontryagin** possa essere utilizzato per formulare delle condizioni sufficienti di ottimalità per il controllo di un sistema. [13]

Si consideri un problema di ottimizzazione della traiettoria nella seguente forma:

$$\min_{t_I, t_F, \mathbf{x}(t), \mathbf{u}(t)} \mathcal{E} = \mathcal{W}(t_F, \mathbf{x}(t_F)) + \int_{t_i}^{t_f} w(t, \mathbf{x}(t), \mathbf{u}(t)) dt \quad (1.16)$$

soggetto a

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); t) \quad \text{dinamica del sistema} \quad (1.17)$$

$$\mathbf{g}(\mathbf{x}(t_F), t_F) = 0 \quad \text{vincoli al contorno} \quad (1.18)$$

$$\mathbf{u}_{\min} \leq \mathbf{u}(t) \leq \mathbf{u}_{\max} \quad \text{limite sul controllo} \quad (1.19)$$

Una traiettoria ammissibile  $\bar{\mathbf{x}}(t)$  associata al controllo  $\bar{\mathbf{u}}(t)$  sarà ottima se corrisponde ad un minimo del funzionale di costo, ossia se

$$\mathcal{W}(t_F, \bar{\mathbf{x}}(t_F)) + \int_{t_i}^{t_f} w(t, \bar{\mathbf{x}}, \bar{\mathbf{u}}) dt \leq \mathcal{W}(t_F, \mathbf{x}(t_F)) + \int_{t_i}^{t_f} w(t, \mathbf{x}, \mathbf{u}) dt \quad (1.20)$$

per ogni altra coppia  $(\mathbf{x}(t), \mathbf{u}(t))$  soddisfacente alla dinamica del sistema e agli altri vincoli. Si introduce la variabile ausiliaria  $x_0$  e si aggiunge alle equazioni dinamiche del sistema l'equazione ausiliaria

$$\dot{x}_0 = w(\mathbf{x}(t), \mathbf{u}(t); t) \quad (1.21)$$

con condizione iniziale

$$x_0(t_i) = 0. \quad (1.22)$$



Si definisce dunque lo **spazio degli stati ampliato** come:

$$X = \begin{pmatrix} x_0 \\ \mathbf{x} \end{pmatrix} \quad (1.23)$$

Con questa notazione, la dinamica del sistema può essere riscritta nella forma:

$$\dot{X} = \begin{pmatrix} w(\mathbf{x}(t), \mathbf{u}(t); t) \\ f(\mathbf{x}(t), \mathbf{u}(t); t) \end{pmatrix}. \quad (1.24)$$

La matrice Jacobiana del campo del sistema 1.24, rispetto alle variabili dello spazio ampliato  $x_0, \dots, x_n$  e calcolata lungo una traiettoria  $(X(t), \mathbf{u}(t))$ , è definita come:

$$\mathbf{J}_{(X(t), \mathbf{u}(t))} \doteq \begin{pmatrix} \frac{\partial w}{\partial x_0} & \cdots & \frac{\partial w}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_0} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}_{(X(t), \mathbf{u}(t))} \quad (1.25)$$

*Osservazione.* La prima colonna di  $\mathbf{J}$  risulterà dunque nulla, non essendoci nessuna dipendenza dalla variabile ausiliaria  $x_0$ .

Con la notazione adottata, si enuncia il principio del massimo di Pontryagin (la dimostrazione, al di fuori dello scopo di questa Tesi, è svolta nel dettaglio in [5]).

**Teorema 1.1** (Principio del massimo di Pontryagin). *Condizione necessaria affinché una traiettoria ammissibile  $X^*(t)$  associata al controllo  $\mathbf{u}^*(t)$  sia ottimale rispetto al funzionale 1.16 è che esistano  $n + 1$  funzioni del tempo  $\lambda^*(t) = (\lambda_0^*(t), \dots, \lambda_n^*(t))$ , dette **funzioni aggiunte**, soddisfacenti al **sistema aggiunto***

$$\dot{\lambda}^*(t) = -(\mathbf{J}_{(X^*(t), \mathbf{u}^*(t))})^T \lambda^*(t) \quad (1.26)$$

con:

$$\lambda^*(t_f) = -\text{grad } \mathcal{W}_{X^*(t_f)} + \sum_{k=1}^l \mu_k \text{grad } g_{kX^*(t_f)} \quad (1.27)$$

e alle condizioni

$$\lambda^*(t) \neq (0, \dots, 0) \quad \forall t \in [t_I, t_F] \quad (1.28)$$

$$\lambda_0^*(t) \equiv \text{costante} \leq 0 \quad \forall t \in [t_I, t_F] \quad (1.29)$$

tali che la **funzione Hamiltoniana**

$$H(\mathbf{x}, \lambda, \mathbf{u}, t) \doteq \sum_{k=0}^n \lambda_k f_k(\mathbf{x}, \mathbf{u}, t) = \lambda^T \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (1.30)$$

goda della proprietà

$$H(X^*, \lambda^*, \bar{\mathbf{u}}, t) \leq H(X^*, \lambda^*, \mathbf{u}^*, t) \quad \forall t \in [t_I, t_F], \forall \bar{\mathbf{u}} \text{ ammissibile} \quad (1.31)$$

*Osservazione.* La condizione 1.31 esprime il *principio del massimo*: l'Hamiltoniana del sistema, valutata lungo la traiettoria ottimale con le relative funzioni aggiunte  $\lambda^*$ , deve assumere valore massimo, istante per istante, in corrispondenza del controllo ottimo.

Il principio del massimo di Pontryagin ha, nella teoria del controllo ottimo, lo stesso ruolo che le condizioni di Karush-Kuhn-Tucker hanno nell'ambito della programmazione non-lineare<sup>3</sup>: entrambe forniscono delle condizioni solo necessarie per l'ottimalità ma non suggeriscono un metodo per costruire una soluzione ottimale. Analogamente a quanto visto in Ottimizzazione Numerica però, aggiungendo opportune ipotesi, è possibile, in alcuni casi, ottenere delle condizioni sufficienti di ottimalità. Si presentano ora due Corollari del principio del massimo di Pontryagin che forniscono delle condizioni sufficienti di ottimalità. [5]

**Corollario 1.1.1** (Insieme finito di controlli ammissibili). *Siano  $\mathbf{u}_1, \dots, \mathbf{u}_k$  gli unici controlli ammissibili che soddisfano le ipotesi del Teorema 1.1. Allora, se esiste, il controllo ottimo  $\mathbf{u}^*$  sarà il controllo che, tra i  $\mathbf{u}_1, \dots, \mathbf{u}_k$ , rende minimo il valore del funzionale  $\mathcal{E}$ .*

Prima di presentare il prossimo Corollario, si danno alcune definizioni preliminari sul concetto di convessità di un sottoinsieme e di un funzionale.

**Definizione 2** (Convessità di un sottoinsieme). Dato uno spazio vettoriale  $\mathcal{X}$ , un sottoinsieme  $\mathcal{A} \in \mathcal{X}$  si dirà **convesso** se:

$$\forall x, y \in \mathcal{A} \Rightarrow \alpha x + (1 - \alpha)y \in \mathcal{A} \quad \forall \alpha \in [0, 1] \quad (1.32)$$

**Definizione 3** (Convessità di un funzionale). Sia  $\mathcal{A}$  un sottoinsieme convesso di uno spazio vettoriale  $\mathcal{X}$ , allora un funzionale  $f : \mathcal{A} \rightarrow \mathbb{R}$  si dirà **convesso** se:

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) \quad \forall x, y \in \mathcal{A} \quad \forall \alpha \in [0, 1] \quad (1.33)$$

Dato il vincolo sullo stato finale 1.18, si definisce l'insieme di tutti i controlli ammissibili che soddisfano tale vincolo come:

$$\mathcal{U} \doteq \{ \bar{\mathbf{u}} : [t_I, t_F] \rightarrow \mathbb{R}^m \mid \mathbf{g}(\bar{\mathbf{x}}(t_F), t_F) = 0 \} \quad (1.34)$$

**Corollario 1.1.2.** *Si supponga che:*

- *il campo  $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); t)$  sia derivabile con continuità rispetto ad  $\mathbf{u}$*
- *l'insieme dei controlli ammissibili  $\mathcal{U}$  sia convesso*
- *il funzionale di costo sia convesso rispetto ad  $\mathbf{u}$  su  $\mathcal{U}$*

*Allora, ogni traiettoria  $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$  soddisfacente le ipotesi del principio del massimo di Pontryagin è ottimale.*

Dai risultati presentati finora, risulta evidente che l'utilizzo del principio del massimo di Pontryagin per risolvere problemi di controllo ottimo è di difficile trattazione, in particolare in mancanza di ipotesi abbastanza restrittive come quelle sfruttate nei Corollari 1.1.1 e 1.1.2. Inoltre, il procedimento di risoluzione per via analitica è spesso di difficile implementazione al calcolatore; come si mostrerà con il Model Predictive Controller nell'esempio 3.3, la possibilità di automatizzare il processo di risoluzione di problemi di controllo ottimo è molto utile per il controllo di sistemi in *real-time* o *near real-time*, come veicoli a guida autonoma o controllori industriali, in cui si richiede la soluzione di centinaia di questi problemi ogni secondo.

<sup>3</sup>Come evidenziato in [3], tale analogia è sostanziale. Infatti, è possibile provare che le condizioni KKT per i problemi NLP costruiti come trascrizione di problemi di controllo ottimo, diventano equivalenti alle condizioni di Pontryagin quando il numero di punti di discretizzazione tende a  $+\infty$ .

Queste limitazioni, unite alla crescente disponibilità di capacità di calcolo e alla nascita di robuste librerie per la programmazione non-lineare, hanno portato nell'ultimo mezzo secolo ad un grande sviluppo dei metodi numerici per il controllo ottimo.

## 1.4 Panoramica dei metodi numerici

Una prima classe di metodi per la soluzione di problemi di controllo ottimo sono i cosiddetti **metodi diretti**, di cui fanno parte anche i metodi di collocazione trattati in questa Tesi. I metodi diretti sono basati sul paradigma *discretize-then-optimize*, ossia prima riscrivono il problema di controllo ottimo come un problema di programmazione non-lineare che, mediante i metodi di ottimizzazione numerica, permette di costruire una sequenza di soluzioni (eventualmente) convergente alla soluzione ottimale. Oltre ai metodi di collocazione diretta, sono ampiamente usati i metodi di shooting (usati anche per problemi di valori al contorno), i metodi di collocazione ortogonale (che discretizzano il dominio usando le radici di una famiglia di polinomi ortogonali) e i metodi spettrali (che utilizzano un'insieme di funzioni di base per approssimare la dinamica e il controllo). Nella classe dei **metodi indiretti** ricadono invece tutti quei metodi che prima costruiscono per via analitica le condizioni necessarie e sufficienti di ottimalità e, successivamente, le risolvono per via numerica. In generale, i metodi indiretti forniscono soluzioni più accurate rispetto ai metodi diretti ma richiedono un più elevato sforzo analitico ed una migliore *guess* iniziale, avendo solitamente una regione di convergenza ridotta. [3] [10]

Una diversa classe di algoritmi per ottenere soluzioni closed-loop di problemi di controllo ottimo sono basati sulla **programmazione dinamica differenziale (DDP)**. Introdotti nel 1966 da David Mayne, i metodi DDP sfruttano le proprietà di sottostruttura ottima per costruire in maniera incrementale una traiettoria ottimale. Una trattazione completa di questi algoritmi è contenuta in [11].

Un ulteriore approccio che si è sviluppato soprattutto negli ultimi tre decenni è quello del **machine learning**. Sfruttando i dati sperimentali sul comportamento del sistema o la possibilità di simularli in maniera accurata, gli algoritmi di machine learning permettono di apprendere automaticamente un modello dinamico del sistema. Tale modello può essere poi utilizzato per impostare uno dei metodi sopra citati oppure, se è disponibile un valido ambiente di simulazione, può essere utilizzato per addestrare un controller basato sul **reinforcement learning**, ossia una tecnica di apprendimento *trial-and-error*. Tali tecniche sono trattate in [9], [2] e [8].



## Capitolo 2

# Metodi risolutivi

### 2.1 Metodi di collocazione diretta

I metodi di collocazione permettono di risolvere problemi di controllo ottimo mappandoli in opportuni problemi di programmazione non-lineare (NLP) vincolati e risolvendo quest'ultimi con i classici metodi di ottimizzazione numerica, come i metodi del punto interno o i metodi iterativi. Nonostante ogni metodo di collocazione presenti delle diverse scelte sulle modalità di implementazione, ciascuno di essi può essere scomposto nei seguenti 3 passi:

1. **Discretizzazione:** la variabile tempo  $t$  viene discretizzata su una partizione dell'intervallo di interesse. Come **variabile di ottimizzazione**, si assume dunque il valore dello stato del sistema e dei controlli in corrispondenza dei punti di discretizzazione. I **vincoli**, sia algebrici che differenziali, vengono imposti nei punti di discretizzazione. Come **funzione obiettivo** del problema NLP si assume un'opportuna discretizzazione del funzionale di costo: il termine di Mayer viene scritto come funzione dello stato e del controllo nell'ultimo punto di discretizzazione, mentre il termine di Lagrange viene approssimato con un'opportuna formula di quadratura, basata anch'essa sui punti di discretizzazione.
2. **Ottimizzazione:** il problema NLP vincolato impostato al punto precedente viene risolto mediante le usuali tecniche di ottimizzazione numerica. Come si vedrà, risultano particolarmente efficaci i metodi di programmazione quadratica sequenziale (**metodi SQP**).
3. **Interpolazione:** data la soluzione del problema NLP, i valori ottimali dello stato e del controllo nei punti di discretizzazione vengono utilizzati per costruire delle **spline di interpolazione** che approssimano la traiettoria ottimale in tutto il dominio di interesse.

Si introduce ora un'opportuna notazione per il procedimento appena descritto. Successivamente, si utilizzerà questa notazione generale per mostrare nel dettaglio 3 diversi metodi di collocazione diretta.

Si consideri un problema di ottimizzazione della traiettoria, così come descritto nella Definizione 1. Fissato l'intervallo di interesse  $[t_I, t_F]$ , si definiscono:

**Definizione 4.** Si diranno **punti di collocazione** un insieme di punti  $\Delta = \{t_0, t_1, \dots, t_{n-1}\}$  tali che:

$$t_I = t_0 < t_1 < \dots < t_{n-2} < t_{n-1} = t_F. \quad (2.1)$$

Si dirà inoltre **passo di discretizzazione** la lunghezza:

$$h_i = t_{i+1} - t_i \quad \forall i = 0, 1, \dots, n-2 \quad (2.2)$$

Per semplicità, si assumeranno i punti in  $\Delta$  come equispaziati, ossia il passo di discretizzazione sarà costante:

$$h = h_i = \frac{t_F - t_I}{n - 1} \quad \forall i = 0, 1, \dots, n - 2 \quad (2.3)$$

Si considerano ora lo stato del sistema  $\mathbf{x}(t)$  ed il controllo  $\mathbf{u}(t)$  nei punti di collocazione e si definiscono, per comodità di notazione, le quantità:

$$\mathbf{x}_k \doteq \mathbf{x}(t_k)$$

$$\mathbf{u}_k \doteq \mathbf{u}(t_k)$$

La **variabile di ottimizzazione** per il problema NLP sarà dunque definita come:

$$\mathcal{X} \doteq (\mathbf{x}_0, \dots, \mathbf{x}_{n-1}, \mathbf{u}_0, \dots, \mathbf{u}_{n-1}) \quad (2.4)$$

I vincoli non differenziali del problema di ottimizzazione della traiettoria, possono ora essere trascritti in vincoli sulla variabile di ottimizzazione. In particolare, si ha:

$$\mathbf{x}_0 = \mathbf{x}_I \quad \mathbf{x}_{n-1} = \mathbf{x}_F \quad \text{condizioni al contorno} \quad (2.5)$$

$$\mathbf{g}(t_0, t_{n-1}, \mathbf{x}_0, \mathbf{x}_{n-1}) \leq \mathbf{0} \quad \text{vincoli al contorno} \quad (2.6)$$

$$\mathbf{h}(t_k, \mathbf{x}_k, \mathbf{u}_k) \leq \mathbf{0} \quad \forall k = 0, \dots, n - 1 \quad \text{vincoli sulla traiettoria} \quad (2.7)$$

$$\mathbf{x}_{\min} \leq \mathbf{x}_k \leq \mathbf{x}_{\max} \quad \forall k = 0, \dots, n - 1 \quad \text{limite sullo stato} \quad (2.8)$$

$$\mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max} \quad \forall k = 0, \dots, n - 1 \quad \text{limite sul controllo} \quad (2.9)$$

Mentre i vincoli di partenza, fatta eccezione per le condizioni al contorno, erano dei vincoli imposti sull'intero intervallo  $[t_I, t_F]$  e dunque rappresentavano un insieme infinito di vincoli algebrici, i vincoli trascritti sui punti di discretizzazione costituiscono un insieme finito di vincoli di uguaglianza e disuguaglianza per le componenti della variabile di ottimizzazione  $\mathcal{X}$ .

Ricordando che il funzionale di costo del problema di ottimizzazione della traiettoria è scrivibile come:

$$\mathcal{E}(t_I, t_F, \mathbf{x}(t), \mathbf{u}(t)) = J(t_I, t_F, \mathbf{x}(t_I), \mathbf{x}(t_F)) + \int_{t_I}^{t_F} w(t, \mathbf{x}(t), \mathbf{u}(t)) dt$$

la **funzione obiettivo**  $\mathcal{C}$  del problema NLP, supponendo fissati gli istanti  $t_I$  e  $t_F$ , sarà definita da:

$$\mathcal{C}(\mathcal{X}) = J(t_I, t_F, \mathbf{x}_0, \mathbf{x}_{n-1}) + Q(\mathcal{X}; t_I, t_F) \quad (2.10)$$

La funzione  $Q$  denota un'approssimazione del termine di Lagrange del funzionale di costo mediante una formula di quadratura basata sui punti di collocazione. La scelta di tale formula, unita alla scelta dello schema per la discretizzazione dei vincoli differenziali (che si sta per introdurre), porta alla definizione dei vari metodi di collocazione diretta.

Restano ora da trascrivere i vincoli sulla dinamica del sistema come relazioni in termini finiti tra le componenti di  $\mathcal{X}$ . Integrando ambo i membri il sistema di ODEs che fornisce il vincolo sulla dinamica su ciascuno dei sotto-intervalli  $[t_k, t_{k+1}]$ , si ottiene

$$\int_{t_k}^{t_{k+1}} \dot{\mathbf{x}} dt = \int_{t_k}^{t_{k+1}} \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); t) dt \quad (2.11)$$

da cui segue subito

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \int_{t_k}^{t_{k+1}} \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); t) dt . \quad (2.12)$$

L'integrale del campo  $\mathbf{f}$  presente al membro destro verrà dunque approssimato mediante una formula di quadratura basata nei punti di collocazione. Scelte diverse sulla formula di quadratura portano a diversi metodi di collocazione diretta.

*Osservazione* (Accuratezza e Consistenza). Nella scelta dello schema di quadratura, è opportuno tener presente che i metodi di collocazione richiedono che tale schema sia **accurato**, ossia fornisca un'approssimazione dell'integrale reale con un errore trascurabile, e **consistente**, ossia che esegua la stessa sequenza di operazioni aritmetiche su qualunque input. Le formule di quadratura adattive dunque, seppur forniscano spesso un'accuratezza maggiore, non risultano adatte a questo contesto. Come verrà evidenziato più nel dettaglio nella Sezione 2.2.3, è opportuno gestire l'analisi dell'accuratezza al di fuori del problema di ottimizzazione NLP in quanto, l'utilizzo di schemi di quadratura adattiva introduce del *rumore* nel processo di discesa del gradiente e potrebbe causare fenomeni di non-convergenza.

### 2.1.1 Metodo del punto medio

Un primo metodo di discretizzazione è il **metodo del punto medio**. Tale metodo si basa sull'idea di approssimare la dinamica del sistema  $\mathbf{f}$ , in ogni intervallo  $[t_k, t_{k+1}]$ , con una stima del valore che essa assume nel punto medio dell'intervallo, ossia in:

$$t_{k+\frac{1}{2}} = \frac{t_k + t_{k+1}}{2} = t_k + \frac{h_k}{2} . \quad (2.13)$$

In particolare,  $\mathbf{f}$  sarà approssimata come:

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); t) \approx \mathbf{f}_{k+\frac{1}{2}} \quad (2.14)$$

Dal momento che i vincoli differenziali devono essere trascritti in funzione delle sole componenti della variabile di ottimizzazione, ossia dei valori dello stato del sistema e del controllo in corrispondenza dei punti di collocazione, il termine  $\mathbf{f}_{k+\frac{1}{2}}$  andrà approssimato. Esso potrà essere scritto in **forma esplicita** utilizzando i valori dove si può usare:

$$\mathbf{f}_{k+\frac{1}{2}} \doteq \mathbf{f} \left( \mathbf{x}_k + \frac{h}{2} \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k; t_k), \mathbf{u}_k; t_k + \frac{h}{2} \right) \quad (2.15)$$

Alternativamente, l'approssimazione in **forma esplicita** si basa sull'utilizzo di una media dei valori che lo stato del sistema assume agli estremi di ciascun intervallo di discretizzazione:

$$\mathbf{f}_{k+\frac{1}{2}} \doteq \mathbf{f} \left( \frac{1}{2}(\mathbf{x}_k + \mathbf{x}_{k+1}), \frac{1}{2}(\mathbf{u}_k + \mathbf{u}_{k+1}); t_k + \frac{h}{2} \right) \quad (2.16)$$

Utilizzando una delle due forme, l'integrale della dinamica su ciascun intervallo di discretizzazione può essere approssimato come:

$$\int_{t_k}^{t_{k+1}} \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); t) dt \approx \int_{t_k}^{t_{k+1}} \mathbf{f}_{k+\frac{1}{2}} dt = h \mathbf{f}_{k+\frac{1}{2}} \quad (2.17)$$

Sostituendo tale stima nell'equazione 2.12, si ottengono i seguenti **vincoli di collocazione**:

$$\mathbf{x}_{k+1} - \mathbf{x}_k = h\mathbf{f}_{k+\frac{1}{2}} \quad \forall k = 0, \dots, n-2 \quad (2.18)$$

Data tale discretizzazione dello stato del sistema e del controllo, il vincolo sulla dinamica del sistema, ossia un sistema di EDO, è stato dunque trascritto in  $n-1$  vincoli di uguaglianza sulle componenti della variabile di ottimizzazione.

Per approssimare il termine di Lagrange del funzionale di costo, ossia definire la funzione  $Q$  introdotta in 2.10, il metodo del punto medio utilizza una formula di quadratura trapezoidale. Tale formula consiste nell'approssimare la funzione integranda, in questo caso  $w$ , con una spline lineare che avrà nodi coincidenti con i punti di collocazione  $t_0, \dots, t_{n-1}$ . Detta  $w_K \doteq w(t_k, \mathbf{x}_k, \mathbf{u}_k)$ , in ciascun intervallo  $[t_k, t_{k+1}]$  la funzione  $w(t, \mathbf{x}(t), \mathbf{u}(t))$  sarà approssimata dal segmento:

$$\begin{aligned} w(t, \mathbf{x}(t), \mathbf{u}(t)) &\approx \frac{t - t_{k+1}}{t_k - t_{k+1}} w_k + \frac{t - t_k}{t_{k+1} - t_k} w_{k+1} \\ &= \frac{t_{k+1} - t}{t_{k+1} - t_k} w_k + \frac{t - t_k}{t_{k+1} - t_k} w_{k+1} \\ &= \left( \frac{w_{k+1} - w_k}{t_{k+1} - t_k} \right) t + \frac{t_{k+1} w_k - t_k w_{k+1}}{t_{k+1} - t_k} \end{aligned}$$

Integrando tale segmento, in ogni intervallo  $[t_k, t_{k+1}]$  si ottiene:

$$\begin{aligned} &\int_{t_k}^{t_{k+1}} \left[ \left( \frac{w_{k+1} - w_k}{t_{k+1} - t_k} \right) t + \frac{t_{k+1} w_k - t_k w_{k+1}}{t_{k+1} - t_k} \right] dt \\ &= \left[ \left( \frac{w_{k+1} - w_k}{t_{k+1} - t_k} \right) \frac{t^2}{2} + \frac{t_{k+1} w_k - t_k w_{k+1}}{t_{k+1} - t_k} t \right]_{t_k}^{t_{k+1}} \\ &= \left( \frac{w_{k+1} - w_k}{t_{k+1} - t_k} \right) \frac{t_{k+1}^2 - t_k^2}{2} + \frac{t_{k+1} w_k - t_k w_{k+1}}{t_{k+1} - t_k} (t_{k+1} - t_k) \\ &= (w_{k+1} - w_k) \frac{t_{k+1} - t_k}{2} + (t_{k+1} w_k - t_k w_{k+1}) \\ &= \frac{1}{2} w_{k+1} t_{k+1} - \frac{1}{2} w_{k+1} t_k - \frac{1}{2} w_k t_{k+1} + \frac{1}{2} w_k t_k + w_k t_{k+1} - w_{k+1} t_k \\ &= \frac{h}{2} (w_k + w_{k+1}) \end{aligned}$$

Dall'addittività dell'integrale rispetto all'unione, si ottiene che l'intero termine di Lagrange sarà approssimato come:

$$\int_{t_I}^{t_F} w(t, \mathbf{x}(t), \mathbf{u}(t)) dt \approx \frac{h}{2} \left[ w_0 + 2 \sum_{k=1}^{n-2} w_k + w_{n-1} \right] . \quad (2.19)$$

Essendo la dinamica del sistema ed il controllo approssimati con delle funzioni costanti in ciascun intervallo  $[t_k, t_{k+1}]$ , lo stato del sistema sarà approssimato con una spline lineare con nodi coincidenti ai punti di collocazione. In particolare, detta  $\mathcal{X}^*$  la soluzione del problema di programmazione non-lineare

$$\mathcal{X}^* = (\mathbf{x}_0^*, \dots, \mathbf{x}_{n-1}^*, \mathbf{u}_0^*, \dots, \mathbf{u}_{n-1}^*) , \quad (2.20)$$



la traiettoria ottimale per lo stato sarà espressa, per ciascun intervallo, come:

$$\mathbf{x}^*(t) = \mathbf{x}_k^* + \frac{t - t_k}{h}(\mathbf{x}_{k+1}^* - \mathbf{x}_k^*) \quad \forall t \in [t_k, t_{k+1}] \quad (2.21)$$

Il controllo, d'altra parte, è assunto costante in ogni intervallo di discretizzazione, per cui si ha:

$$\mathbf{u}^*(t) = \mathbf{u}_k^* \quad \forall t \in [t_k, t_{k+1}[ \quad (2.22)$$

Come si vedrà nel dettaglio nella Sezione 2.2.3, tali rappresentazioni sono particolarmente utili per valutare l'accuratezza della soluzione ottenuta.

*Osservazione* (Sparsità del problema NLP). È interessante osservare che i problemi NLP che si ottengono trascrivendo problemi di controllo ottimo con i metodi di collocazione diretta hanno il vantaggio di essere estremamente sparsi. A titolo di esempio, in Figura 2.1 è riportata la struttura della matrice Jacobiana della funzione obiettivo e dei vincoli di un problema trascritto con il metodo del punto medio e  $n = 10$  punti di collocazione.

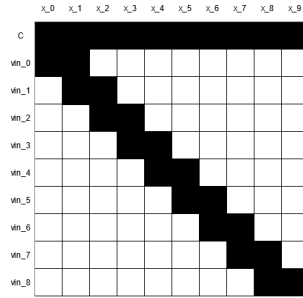


FIGURA 2.1: Struttura della matrice Jacobiana - le celle nere indicano i coefficienti non nulli

Come emerge dalla Figura, la matrice risulta estremamente sparsa, con una percentuale di coefficienti non nulli pari solo al 28% del numero totale.

### 2.1.2 Metodo del trapezio

Un metodo simile a quello del punto medio, ma che fornisce una soluzione di ordine superiore, è il **metodo del trapezio**. Tale metodo utilizza la formula di quadratura trapezoidale, introdotta nelle pagine precedenti, per approssimare l'integrale della dinamica del sistema. Analogamente a quanto fatto per lo stato e per il controllo, si definisce:

$$\mathbf{f}_k \doteq \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k); t_k) = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k; t_k) \quad (2.23)$$

Dunque, applicando la formula di quadratura sviluppata a Pagina 12, la dinamica  $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); t)$  può essere approssimata nell'intervallo  $[t_k, t_{k+1}]$  con il segmento:

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); t) \approx \left( \frac{\mathbf{f}_{k+1} - \mathbf{f}_k}{t_{k+1} - t_k} \right) t + \frac{t_{k+1}\mathbf{f}_k - t_k\mathbf{f}_{k+1}}{t_{k+1} - t_k} \quad (2.24)$$

Integrando sull'intero intervallo di discretizzazione si ottiene:

$$\int_{t_k}^{t_F} \mathbf{f} dt \approx \frac{h}{2}(\mathbf{f}_k + \mathbf{f}_{k+1}) \quad (2.25)$$

Sostituendo tale approssimazione in 2.12, per ciascuna coppia di punti di collocazione  $t_k$  e  $t_{k+1}$  si ottiene il **vincolo di collocazione**:

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \frac{h}{2}(\mathbf{f}_k + \mathbf{f}_{k+1}) \quad (2.26)$$

Per la costruzione della funzione obiettivo, il metodo del trapezio utilizza la stessa formula di quadratura del metodo del punto medio sviluppata in 2.19.

Analogamente a quanto fatto per il metodo precedente, si costruiscono le interpolazioni dello stato del sistema e del controllo. Il controllo ottimale sarà esprimibile con una spline lineare come:

$$\mathbf{u}^*(t) = \mathbf{u}_k^* + \frac{t - t_k}{h}(\mathbf{u}_{k+1}^* - \mathbf{u}_k^*) \quad (2.27)$$

Lo stato del sistema, in quanto integrale della dinamica, si esprimerà mediante una spline quadratica con i nodi coincidenti ai punti di collocazione. Infatti, la dinamica del sistema valutata lungo la traiettoria ottimale nei punti di collocazione è:

$$\mathbf{f}^*(t) = \dot{\mathbf{x}}^*(t) \approx \mathbf{f}_k^* + \frac{t - t_k}{h}(\mathbf{f}_{k+1}^* - \mathbf{f}_k^*) \quad (2.28)$$

Introducendo per semplicità la variabile  $\tau \doteq t - t_k$  e integrando ambo i membri, si ottiene:

$$\mathbf{x}^*(\tau) = \int_0^\tau \dot{\mathbf{x}}^*(\tau) d\tau \approx \mathbf{f}_k^* \tau + \frac{\tau^2}{2h}(\mathbf{f}_{k+1}^* - \mathbf{f}_k^*) + \mathbf{c} \quad (2.29)$$

dove  $\mathbf{c}$  denota la costante di integrazione. Ricordando che  $\mathbf{x}(\tau = 0) = \mathbf{x}_k$  e sostituendo nuovamente la variabile  $\tau$ , si giunge a:

$$\mathbf{x}^*(t) = \mathbf{x}_k^* + \mathbf{f}_k^*(t - t_k) + \frac{(t - t_k)^2}{2h}(\mathbf{f}_{k+1}^* - \mathbf{f}_k^*) \quad \forall t \in [t_k, t_{k+1}] \quad (2.30)$$

### 2.1.3 Metodo di Hermite-Simpson

Il metodo di Hermite-Simpson si basa su due approssimazioni interpolatorie: la prima, basata sulla formula di Simpson, approssima la dinamica del sistema con un polinomio di secondo grado; la seconda, utilizza un polinomio cubico di Hermite per stimare il valore dello stato del sistema nel punto medio dell'intervallo di discretizzazione. In particolare, la dinamica del sistema  $\mathbf{f}$  può essere approssimata in ciascuno degli intervalli  $[t_k, t_{k+1}]$  con una funzione quadratica  $\mathbf{q}(t)$ , tale che:

$$\mathbf{q}(t_k) = \mathbf{f}_k \quad \mathbf{q}(t_{k+\frac{1}{2}}) = \mathbf{f}_{k+\frac{1}{2}} \quad \mathbf{q}(t_{k+1}) = \mathbf{f}_{k+1} \quad (2.31)$$

Date queste tre condizioni, la funzione quadratica interpolante si costruisce facilmente utilizzando i polinomi di Lagrange nella forma:

$$\begin{aligned} \mathbf{q}(t) = & + \frac{(t - t_{k+\frac{1}{2}})(t - t_{k+1})}{(t_k - t_{k+\frac{1}{2}})(t_k - t_{k+1})} \mathbf{f}_k + \\ & + \frac{(t - t_k)(t - t_{k+1})}{(t_{k+\frac{1}{2}} - t_k)(t_{k+\frac{1}{2}} - t_{k+1})} \mathbf{f}_{k+\frac{1}{2}} + \\ & + \frac{(t - t_k)(t - t_{k+\frac{1}{2}})}{(t_{k+1} - t_k)(t_{k+1} - t_{k+\frac{1}{2}})} \mathbf{f}_{k+1} \end{aligned}$$

Come mostrato in [1], l'integrale di questa funzione risulta essere:

$$\int_{t_k}^{t_{k+1}} \mathbf{q}(t) dt = \frac{h}{6} (\mathbf{f}_k + 4\mathbf{f}_{k+\frac{1}{2}} + \mathbf{f}_{k+1}) \quad (2.32)$$

Questa formula è detta **formula di Simpson**. Sostituendo tale approssimazione in 2.12, si ottengono i seguenti vincoli di collocazione:

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \frac{h}{6} (\mathbf{f}_k + 4\mathbf{f}_{k+\frac{1}{2}} + \mathbf{f}_{k+1}) \quad \forall k = 0, \dots, n-2 \quad (2.33)$$

Essendo il valore della dinamica  $\mathbf{f}_{k+\frac{1}{2}}$  a priori ignoto, è necessario esprimere tale valore in funzione delle componenti della variabile di ottimizzazione  $\mathcal{X}$ . A tal fine, si costruisce un'interpolazione di Hermite. Prima di enunciare il teorema su tale tecnica di interpolazione, si introduce la base dei polinomi di Lagrange.

**Definizione 5** (Base dei polinomi di Lagrange). Dato un insieme di  $n-1$  punti  $(t_0, x_0), \dots, (t_{n-1}, x_{n-1})$  tali che  $t_0 < \dots < t_{n-1}$ , si definiscono **polinomi di Lagrange** i seguenti  $n-1$  polinomi:

$$L_i(t) = \prod_{j=0, j \neq i}^{n-1} \frac{(t - t_j)}{(t_i - t_j)} \quad (2.34)$$

Sfruttando la definizione di polinomi di Lagrange, si costruisce il polinomio di interpolazione di Hermite che permette di porre, oltre alle usuali condizioni di interpolazione, anche condizioni sulla derivata prima del polinomio interpolante.

**Teorema 2.1** (Polinomio di Hermite<sup>1</sup>). Dato un intervallo  $[a, b]$  e un insieme di  $n$  punti  $a = t_0 < \dots < t_{n-1} = b$ , allora esiste ed è unico in  $[a, b]$  il polinomio di interpolazione  $\mathbf{p}_M(t)$  di grado al più  $M = 2n + 1$  che soddisfi, per assegnati valori  $\mathbf{x}_k$  e  $\mathbf{x}'_k$  con  $k = 0, \dots, n-1$ , le condizioni:

$$\mathbf{p}_M(t_k) = \mathbf{x}_k \quad \mathbf{p}'_M(t_k) = \mathbf{x}'_k \quad \forall k = 0, \dots, n-1 \quad (2.35)$$

In particolare, tale polinomio si dirà **polinomio di Hermite** e sarà scrivibile nella forma

$$\mathbf{p}_M(t) = \sum_{k=0}^{n-1} \mathbf{x}_k H_i(t) + \sum_{k=0}^{n-1} \mathbf{x}'_k K_i(t) \quad (2.36)$$

dove le funzioni  $H_i$  e  $K_i$  sono scrivibili in funzione dei polinomi di Lagrange come

$$H_i(t) = L_i(t)^2 (1 - 2(t - t_i) L'_i(t_i))$$

$$K_i(t) = L_i(t)^2 (t - t_i)$$

Si utilizza ora l'interpolazione di Hermite per approssimare la dinamica del sistema. In particolare, si costruisce un polinomio cubico  $\mathbf{p}_3$  su ciascun intervallo  $[t_k, t_{k+1}]$  imponendo:

$$\mathbf{p}_3(t_k) = \mathbf{x}_k \quad \mathbf{p}_3(t_{k+1}) = \mathbf{x}_{k+1} \quad (2.37)$$

$$\mathbf{p}'_3(t_k) = \mathbf{f}_k \quad \mathbf{p}'_3(t_{k+1}) = \mathbf{f}_{k+1} \quad (2.38)$$

<sup>1</sup>Una trattazione più completa del polinomio di interpolazione di Hermite può essere trovata in [4] ed è al di fuori dello scopo di questa Tesi.

I due polinomi della base di Lagrange  $L_0$  e  $L_1$  saranno:

$$L_0(t) = \frac{t - t_{k+1}}{t_k - t_{k+1}}$$

$$L_1(t) = \frac{t - t_k}{t_{k+1} - t_k}$$

Conseguentemente, le funzioni  $H_0$ ,  $H_1$ ,  $K_0$  e  $K_1$  saranno scrivibili come:

$$H_0(t) = \frac{(t - t_{k+1})^2}{(t_k - t_{k+1})^2} \left( 1 - 2 \frac{(t - t_k)}{t_k - t_{k+1}} \right)$$

$$H_1(t) = \frac{(t - t_k)^2}{(t_{k+1} - t_k)^2} \left( 1 - 2 \frac{(t - t_{k+1})}{t_{k+1} - t_k} \right)$$

$$K_0(t) = \frac{(t - t_{k+1})^2}{(t_k - t_{k+1})^2} (t - t_k)$$

$$K_1(t) = \frac{(t - t_k)^2}{(t_{k+1} - t_k)^2} (t - t_{k+1})$$

Dunque, il polinomio cercato avrà forma:

$$\mathbf{p}_3(t) \doteq \mathbf{x}_k H_0(t) + \mathbf{x}_{k+1} H_1(t) + \mathbf{f}_k K_0(t) + \mathbf{f}_{k+1} K_1(t) \quad (2.39)$$

Valutando tale polinomio in  $t_{k+\frac{1}{2}}$  e ricordando  $h = t_{k+\frac{1}{2}} - t_k = t_{k+1} - t_{k+\frac{1}{2}}$ , si arriva alla stima cercata:

$$\begin{aligned} \mathbf{x}_{k+\frac{1}{2}} &\approx p(t_{k+\frac{1}{2}}) = \mathbf{x}_k H_0(t_{k+\frac{1}{2}}) + \mathbf{x}_{k+1} H_1(t_{k+\frac{1}{2}}) + \mathbf{f}_k K_0(t_{k+\frac{1}{2}}) + \mathbf{f}_{k+1} K_1(t_{k+\frac{1}{2}}) \\ &= \frac{1}{4} \left( 2\mathbf{x}_k + 2\mathbf{x}_{k+1} + \frac{h_k}{2} \mathbf{f}_k - \frac{h_k}{2} \mathbf{f}_{k+1} \right) \\ &= \frac{1}{2} (\mathbf{x}_k + \mathbf{x}_{k+1}) + \frac{h_k}{8} (\mathbf{f}_k - \mathbf{f}_{k+1}) \end{aligned}$$

Oltre ai vincoli di collocazione, dovranno quindi essere imposti anche le seguenti **condizioni di interpolazione**:

$$\mathbf{x}_{k+\frac{1}{2}} = \frac{1}{2} (\mathbf{x}_k + \mathbf{x}_{k+1}) + \frac{h}{8} (\mathbf{f}_k - \mathbf{f}_{k+1}) \quad \forall k = 0, \dots, n-2 \quad (2.40)$$

Tali condizioni possono essere imposte in due diversi modi. Un primo metodo, consiste nell'ampliare la variabile di ottimizzazione per ottenere anche il valore dello stato e del controllo nei punti medi degli intervalli di collocazione, permettendo così di imporre le condizioni direttamente sui  $\mathbf{x}_{k+\frac{1}{2}}$ . In questo caso di **forma separata**. Alternativamente, è possibile sostituire l'espressione trovata per  $\mathbf{x}_{k+\frac{1}{2}}$  nei vincoli di collocazione, eliminando così la dipendenza dal valore dello stato nei punti medi. Questa forma prende invece il nome di **forma compressa**.

*Osservazione* (Forma compressa vs Forma separata). In generale, stimare quale delle due forme sia più conveniente usare non è immediato e bisogna effettuare alcune considerazioni di carattere implementativo. Come mostrato da John T. Betts in [3], il costo computazionale della risoluzione di un problema NLP può essere scomposto come:

$$T \approx c_1 n \gamma (\gamma + 3) + c_2 n^{1.9} \quad (2.41)$$

dove  $c_1$  e  $c_2$  sono delle costanti e  $\gamma$  è un indice non dipendente da  $n$  legato alla sparsità della matrice Jacobiana del problema, utilizzata per stimare con il metodo delle differenze finite il gradiente della funzione obiettivo e dei vincoli. In generale, si ha che  $\gamma_{separata} \leq \gamma_{compressa}$ , ossia il problema impostato mediante la forma separata è più sparso ma richiede il doppio di punti di collocazione rispetto alla forma compressa. Dunque, essendo nell'espressione di  $T$  il termine  $n^{1.0}$  dominante, esisterà un  $n^*$  tale che risulterà più conveniente utilizzare la forma di Hermite-Simpson separata per ogni  $n < n^*$  e la forma compressa per ogni  $n \geq n^*$ .

Infine, eventuali termini integrali nel funzionale di costo verranno approssimati mediante la formula di quadratura di Simpson composta. Applicando la formula 2.32 a ciascun intervallo di integrazione e ricordando l'addittività dell'integrale, si giunge a:

$$Q(w) = \frac{h}{6} \sum_{k=0}^{n-1} \left( w_k + 4w_{k+\frac{1}{2}} + w_{k+1} \right) \quad (2.42)$$

Non si entrerà nel dettaglio della derivazione completa delle formule di interpolazione utilizzate nel metodo di Hermite-Simpson per costruire la traiettoria ottimale  $(\mathbf{x}^*(t), \mathbf{u}^*(t))$ , tuttavia si evidenzia una proprietà importante delle soluzioni ottenute mediante tale metodo. Approssimando la dinamica del sistema con una funzione quadratica, il metodo di Hermite-Simpson permette di interpolare i valori ottimali dello stato  $\mathbf{x}_k^*$  con una spline cubica che, grazie alle condizioni di interpolazione, risulterà essere non solo continua, ma anche derivabile con continuità sull'intero intervallo  $[t_I, t_F]$ . Questa proprietà, unita alle buone proprietà di convergenza, ha reso il metodo di Hermite-Simpson ampiamente utilizzato.

## 2.2 Tecniche di miglioramento

Definiti i concetti fondamentali dei metodi di collocazione diretta e date le principali formule, si affrontano ora alcune tecniche di miglioramento applicabili a tali metodi e l'analisi dell'accuratezza delle soluzioni prodotte da ciascun metodo. Come si vedrà, alcune considerazioni emergeranno dai metodi di ottimizzazione numerica adottati e dagli schemi di integrazione numerica. Altre, saranno proprie del contesto dei problemi di controllo ottimo.

### 2.2.1 Scelta della guess iniziale

Uno dei problemi riscontrati più di frequente quando si utilizzano metodi di collocazione diretta è la convergenza in punti di minimo locale con scarse proprietà di ottimalità o la totale non convergenza. Questo problema è tipico dei problemi di programmazione non-lineare con funzione obiettivo o regione ammissibile non convessa, o nei quali la regione delle soluzioni ottimali non è limitata. Un metodo molto utilizzato per evitare questi fenomeni e favorire una convergenza più veloce e accurata, consiste nella costruzione di una *guess iniziale* per il problema NLP trascritto che, pur non rispettando tutti i vincoli, fornisca un'approssimazione della soluzione ottimale cercata. In generale, questo può non essere facile, specialmente se il problema non ha condizioni al contorno o presenta un funzionale di costo molto complesso. Tuttavia, come si vedrà, spesso anche inizializzazioni piuttosto semplici portano a notevoli miglioramenti delle proprietà di convergenza.

Un fenomeno di non convergenza si è incontrato nel corso della sperimentazione numerica riguardante l'Esempio 3.2 della curva catenaria. In una prima implementazione infatti, sono state testate due guess iniziali piuttosto semplici: la prima

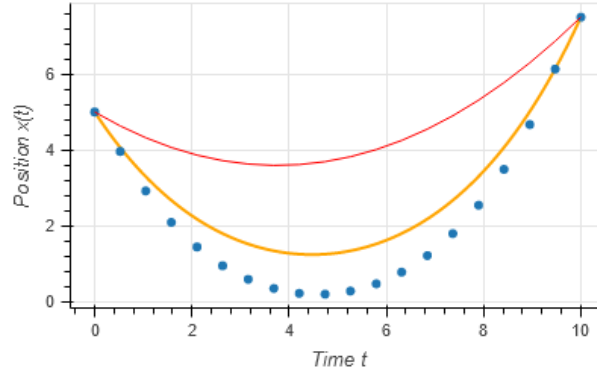


FIGURA 2.2: Problema della catenaria: in rosso, la guess iniziale parabolica; in giallo, la soluzione esatta; in blu, la soluzione numerica

corrispondente al vettore nullo  $\mathcal{X}_0 = (0, \dots, 0)$ , la seconda, utilizzando un vettore di punti che variano linearmente tra le due condizioni al contorno per la posizione  $x$  e il vettore nullo per il controllo  $u$ . Entrambe queste inizializzazioni tuttavia rendevano il problema NLP non convergente. Al fine di ovviare a questa problematica, si è costruita una traiettoria parabolica congiungente i punti definiti dalle condizioni al contorno, fornendo così una guess iniziale vicina alla reale traiettoria della catenaria. Come mostrato in Figura 2.2, tale inizializzazione ha permesso di ottenere una soluzione che, seppur non ancora ottimale, si avvicina alla soluzione esatta.

## 2.2.2 Strategie per sistemi dinamici stiff

Il processo di trascrizione dei vincoli dinamici in vincoli di collocazione equivale sostanzialmente all'applicazione di un metodo numerico per la risoluzione di equazioni differenziali ordinarie. Come studiato in "Analisi Numerica A" e trattato anche in [17] nel caso del problema 3.1, l'applicazione di tali metodi a sistemi anche apparentemente semplici può talvolta dare luogo a comportamenti oscillatori della soluzione. Tali fenomeni, seppur amplificati da alcuni metodi numerici, sono una caratteristica intrinseca di alcuni sistemi dinamici che vengono per questo definiti **sistemi stiff** (nell'ambito del controllo ottimo e nella letteratura in merito, si fa spesso riferimento a questo fenomeno come **ringing**). Un esempio di tale fenomeno è stato rilevato applicando il metodo di Hermite-Simpson al problema presentato nella Sezione 1.2

Una tecnica per limitare l'effetto di tali fenomeni, propria dei metodi di collocazione diretta, consiste nella **regolarizzazione della funzione obiettivo**. Tale strategia si basa sull'aggiunta nella funzione obiettivo di un termine di *smoothing* che quantifica la variazione nell'uso del controllo in istanti successivi. Minimizzando la funzione obiettivo, e dunque anche il termine di smoothing, si ottiene una soluzione con una ridotta variazione del controllo, riducendo così il fenomeno oscillatorio. In particolare, si parlerà di *smoothing del primo ordine* se si utilizza un termine che stima la derivata prima del controllo, come:

$$\frac{1}{h} \sum_{k=0}^{n-2} [\mathbf{u}_{k+1} - \mathbf{u}_k]^2 \quad (2.43)$$

Analogamente, si parlerà di *smoothing del secondo ordine* se si minimizza una stima

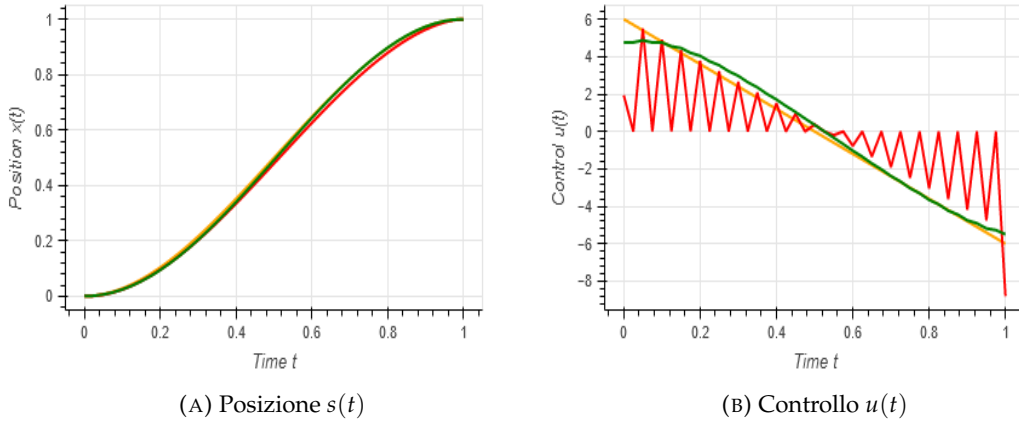


FIGURA 2.3: In giallo, la soluzione esatta; in rosso, la soluzione numerica instabile; in verde, la soluzione numerica con smoothing.

della derivata seconda del controllo:

$$\frac{1}{h^3} \sum_{k=1}^{n-2} [\mathbf{u}_{k+1} - 2\mathbf{u}_k + \mathbf{u}_{k-1}]^2. \quad (2.44)$$

Come mostrato in Figura 2.3b, l'applicazione di uno smoothing del primo ordine alla soluzione del problema 3.1 ha permesso di ottenere un controllo ottimale più liscio.

*Osservazione.* Nei sistemi nominali che non tengono conto della dinamica reale degli attuatori, se il passo di discretizzazione è abbastanza piccolo, un controllo di tipo *bang-bang* è, in termini della traiettoria predetta, indistinguibile da un controllo che mantiene costantemente il valore medio (ciò è anche evidente in Figura 2.3a: la traiettoria generata dal controllo instabile è quasi coincidente a quella generata dal controllo normalizzato). Tuttavia, specialmente nel caso del controllo di veicoli con passeggeri a bordo, questo può tradursi in uno stile di guida piuttosto brusco e sgradevole.

### 2.2.3 Analisi dell'accuratezza

Come per ogni metodo numerico, è opportuno stimare l'accuratezza delle soluzioni ottenute mediante i metodi di collocazione. In generale, non si ha a disposizione la soluzione esatta dei problemi che si trattano e bisognerà basarsi su metriche diverse per valutare se la soluzione numerica soddisfa le necessità di accuratezza imposte dal contesto. In particolare, analogamente al procedimento , una stima dell'accuratezza della soluzione numerica può essere ottenuta valutando quanto fedelmente essa rispetta le equazioni dinamiche del sistema. Nel seguito, si denoterà con  $(\mathbf{x}(t), \mathbf{u}(t))$  la soluzione esatta, con  $(\mathbf{x}_k^*, \mathbf{u}_k^*)$  la soluzione numerica nei punti di collocazione e con  $(\bar{\mathbf{x}}^*(t), \bar{\mathbf{u}}^*(t))$  la sua interpolazione (concorde con il metodo adottato) sul resto del dominio.

Per definizione, la soluzione esatta sarà tale che:

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} \mathbf{f}(\bar{\mathbf{x}}^*(t), \bar{\mathbf{u}}^*(t); t) dt \quad (2.45)$$

Introducendo una tecnica spesso utilizzata nel valutare l'accuratezza di schemi di

integrazione numerica, si definisce l'approssimazione dello stato del sistema all'istante  $t_{k+1}$  ottenuta a partire dalla soluzione esatta all'istante  $t_k$  più l'integrale della dinamica valutata lungo l'interpolazione della soluzione numerica:

$$\hat{\mathbf{x}}(t_{k+1}) \doteq \mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} \mathbf{f}(\bar{\mathbf{x}}^*(t), \bar{\mathbf{u}}^*(t); t) dt \quad (2.46)$$

Analogamente, applicando lo stesso procedimento alla soluzione numerica nei punti di collocazione si ottiene:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1}^* &= \mathbf{x}_k^* + \int_{t_k}^{t_{k+1}} \mathbf{f}(\bar{\mathbf{x}}^*(t), \bar{\mathbf{u}}^*(t); t) dt \\ &= \mathbf{x}_k^* + \int_{t_k}^{t_{k+1}} \frac{d\bar{\mathbf{x}}^*(t)}{dt} dt - \int_{t_k}^{t_{k+1}} \frac{d\bar{\mathbf{x}}^*(t)}{dt} dt + \int_{t_k}^{t_{k+1}} \mathbf{f}(\mathbf{x}^*(t), \mathbf{u}^*(t); t) dt \\ &= \mathbf{x}_k^* + \bar{\mathbf{x}}^*(t_{k+1}) - \bar{\mathbf{x}}^*(t_k) + \int_{t_k}^{t_{k+1}} \left[ \mathbf{f}(\mathbf{x}^*(t), \mathbf{u}^*(t); t) - \frac{d\bar{\mathbf{x}}^*(t)}{dt} \right] dt \end{aligned}$$

ma, essendo  $\bar{\mathbf{x}}^*(t_k) = \mathbf{x}_k^*$  per le condizioni di interpolazione, si ha:

$$= \mathbf{x}_{k+1}^* + \int_{t_k}^{t_{k+1}} \left[ \mathbf{f}(\mathbf{x}^*(t), \mathbf{u}^*(t); t) - \frac{d\bar{\mathbf{x}}^*(t)}{dt} \right] dt$$

Ponendo  $\mathbf{f}^* = \mathbf{f}(\mathbf{x}^*(t), \mathbf{u}^*(t); t)$ , spostando  $\mathbf{x}_{k+1}^*$  al primo termine e prendendo il valore assoluto ambo i membri, si ottiene:

$$|\hat{\mathbf{x}}_{k+1}^* - \mathbf{x}_{k+1}^*| = \left| \int_{t_k}^{t_{k+1}} \left| \mathbf{f}^* - \frac{d\bar{\mathbf{x}}^*(t)}{dt} \right| dt \right| \leq \int_{t_k}^{t_{k+1}} \left| \mathbf{f}^* - \frac{d\bar{\mathbf{x}}^*(t)}{dt} \right| dt \quad (2.47)$$

dove si dirà:

$$\epsilon(t) \doteq \left| \mathbf{f}^* - \frac{d\bar{\mathbf{x}}^*(t)}{dt} \right|. \quad (2.48)$$

**Definizione 6** (Errore locale di discretizzazione). Con la notazione finora adottata, si definisce **errore assoluto locale** nell'intervallo  $[t_k, t_{k+1}]$ , per ciascuna delle componenti  $x_i$  dello stato  $\mathbf{x}$ , la quantità:

$$\eta_i \doteq \int_{t_k}^{t_{k+1}} \epsilon_i(t) dt. \quad (2.49)$$

Tali espressioni integrali possono essere valutate per ciascun intervallo di discretizzazione e per ciascuna componente dello stato del sistema utilizzando un metodo di integrazione numerica. A tale scopo, un metodo molto diffuso per le sue proprietà di convergenza è il metodo di **quadratura di Rhomberg**.

*Osservazione* (Formula di quadratura di Rhomberg). Non entrando nel dettaglio della definizione analitica delle formule di quadratura di Rhomberg, si rimanda per maggiori dettagli sull'implementazione alla documentazione della libreria utilizzata (<https://docs.scipy.org/doc/scipy/reference/>).

La stima introdotta dell'errore di discretizzazione risulta particolarmente utile nel cosiddetto processo di **mesh refinement**. Come accennato all'inizio del Capitolo 2.1, la gestione dell'accuratezza è da effettuare al di fuori del problema NLP modificando la finezza dei punti di collocazione o l'ordine delle formule di collocazione utilizzate. In particolare, data per ciascun intervallo la stima dell'errore locale, è possibile calcolare quanti punti di collocazione o quanti nodi di interpolazione è



necessario aggiungere per ottenere la precisione desiderata. Si dicono **h-methods** i metodi che raggiungono l'accuratezza richiesta rendendo più densi i punti di collocazione, ossia rendendo il passo di discretizzazione  $h$  più piccolo. Sono detti invece **p-methods** i metodi che migliorano l'accuratezza aumentando il grado dello schema di discretizzazione utilizzato, ossia aumentando il numero dei nodi di interpolazione per ciascun intervallo. [3]

*Osservazione* (Curse of dimensionality). È interessante evidenziare che le tecniche di ottimizzazione introdotte soffrono del fenomeno che Richard E. Bellman chiamava *curse of dimensionality* o "maledizione della dimensionalità". Infatti, i tempi di esecuzione di tali programmi crescono notevolmente al crescere del numero di punti di collocazione (e quindi della dimensione della variabile di ottimizzazione e del numero di vincoli). Al fine di verificare sperimentalmente questo fenomeno, si è eseguito il programma per il problema della Sezione 1.2 con un numero crescente di punti di collocazione e se ne è misurato il tempo di esecuzione.

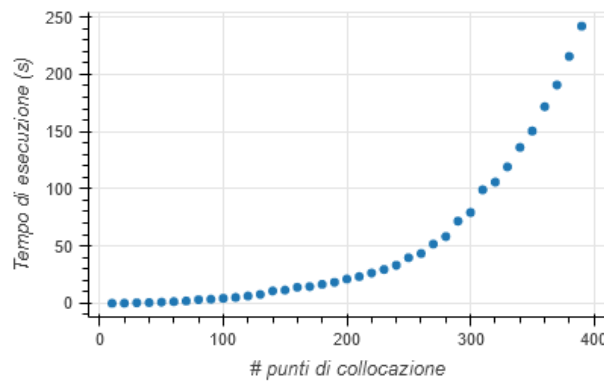


FIGURA 2.4: Sull'asse delle ascisse il numero di punti di collocazione utilizzati, sull'asse delle ordinate il tempo di esecuzione in s.

Tali dati sono riproducibili con `problema_test/dimensionality.py`.



## Capitolo 3

# Applicazioni ed esempi

Al fine di comprendere meglio e analizzare i fenomeni numerici che emergono dall'applicazione delle tecniche illustrate, si presentano ora tre esempi concreti, ad alcuni dei quali si è già fatto riferimento nel corso della discussione. Il primo esempio, tratto dalla letteratura, fornisce un'introduzione alle problematiche implementative dei metodi di collocazione e un'opportunità per illustrare alcune delle tecniche di miglioramento descritte finora. Il secondo riprende un noto problema del calcolo variazionale, il problema della catenaria, e mostra come questo possa essere scritto e risolto come problema di controllo ottimo. Il terzo esempio è frutto dell'esperienza acquisita nel corso del corrente Anno Accademico nell'ambito dell'insegnamento "Automotive Learning by Doing 1", durante il quale ho lavorato nel team di Ateneo per la Formula SAE Driverless per lo sviluppo di un veicolo da corsa autonomo.

### 3.1 Esempio 1: Controllo con lavoro minimo

Un problema di notevole interesse nel campo del controllo ottimo dei veicoli è determinare il controllo che permette di muovere il sistema da uno stato  $\mathbf{x}_0$  a uno stato  $\mathbf{x}_F$  minimizzando il consumo di energia, sia essa sotto forma di carburante o corrente elettrica. Tale problema, applicato al controllo di un treno, è stato ampiamente trattato nella letteratura scientifica in [3] e ripreso da [17].

Si consideri un treno che viaggia lungo una direzione prefissata  $\vec{e}$  su una traiettoria liscia ad altezza variabile. Lo stato del sistema sarà descritto dalla coppia  $(s(t), v(t))$  dove  $s$  e  $v$  denotano rispettivamente la posizione lungo la direzione  $\vec{e}$  e la velocità del treno all'istante  $t$ . Il sistema dinamico sarà inoltre soggetto al seguente sistema di equazioni differenziali:

$$\begin{cases} \dot{s} = v \\ \dot{v} = h(s) - (\alpha + \beta|v| + \gamma v^2) + u_a - u_f \end{cases} \quad (3.1)$$

La variazione di velocità  $\dot{v}$ , ossia l'accelerazione, è composta dai quattro termini:

- **L'accelerazione di gravità**  $h(s)$  che dipende dal profilo dell'altezza
- **L'attrito radente** con i binari  $-\alpha$ , con  $\alpha \in \mathcal{R}^+$  fissato
- **L'attrito viscoso** con l'aria  $-(\beta|v| + \gamma v^2)$ , con coefficienti  $\beta, \gamma \in \mathcal{R}^+$  fissati
- **Il contributo dei controlli**  $u_a$  e  $u_f$ , che denotano rispettivamente l'utilizzo dell'acceleratore e dei freni.

Fissate inoltre le condizioni al contorno

$$s(t_0) = s_I \quad s(t_F) = s_F \quad (3.2)$$

$$v(t_0) = v_I \quad v(t_F) = v_F \quad (3.3)$$

e tenendo conto delle limitazioni costruttive del veicolo, date da:

$$\begin{aligned} 0 &\leq u_a(t) \leq u_a^{\max} \\ u_f^{\min} &\leq u_f(t) \leq 0 \end{aligned}$$

si vuole minimizzare il consumo di carburante. Assumendo che esso sia proporzionale al lavoro totale svolto dall'acceleratore durante il moto, si assume come funzionale di costo

$$\mathcal{E} = \int_{t_0}^{t_F} u_a(t) v(t) dt. \quad (3.4)$$

Risulta evidente che la soluzione di tale problema dipenderà largamente dal profilo dell'altezza del percorso, rappresentato dal termine  $h(s)$ . In particolare, si assumerà:

$$h_2(s) = s^4 - 5s^3 + 10s \quad (3.5)$$

Per la sperimentazione numerica, reperibile nella cartella /train/, si sono utilizzati i seguenti valori:

Costante	Valore
$t_I$	0.0s
$t_F$	3.0s
$s_I$	0.0m
$s_F$	5.0m
$v_I$	0.0m/s
$v_F$	0.0m/s
$u_a^{\max}$	10.0m/s <sup>2</sup>
$u_f^{\min}$	-2.0m/s <sup>2</sup>
$\alpha$	0.21
$\beta$	0.16
$\gamma$	0.10

Si è scelto di utilizzare una discretizzazione con il metodo del trapezio con una partizione equispaziata del dominio  $[t_0, t_F]$ . Per semplificare la scrittura dei vincoli di collocazione, si introduce la variabile ausiliaria  $a$  dell'accelerazione, modificando quindi il sistema della dinamica in:

$$\begin{cases} \dot{s} = v \\ \dot{v} = a \\ a = h(s) - (\alpha + \beta|v| + \gamma v^2) + u_a - u_f \end{cases}. \quad (3.6)$$

La variabile di ottimizzazione  $\mathcal{X}$  è definita come:

$$\mathcal{X} = (s_0, \dots, s_{n-1}, v_0, \dots, v_{n-1}, a_0, \dots, a_{n-1}, u_0^a, \dots, u_{n-1}^a, u_0^b, \dots, u_{n-1}^b) \quad (3.7)$$

Le condizioni al contorno si traducono facilmente in condizioni sulla variabile  $\mathcal{X}$ :

$$s_0 = s_I \quad s_{n-1} = s_F \quad (3.8)$$

$$v_0 = v_I \quad v_{n-1} = v_F \quad (3.9)$$

Analogamente, i vincoli box sui controlli vengono imposti come:

$$\begin{aligned} 0 &\leq u_k^a \leq u_a^{\max} \\ u_f^{\min} &\leq u_k^f \leq 0 \end{aligned}$$

I vincoli di collocazione saranno, con  $k = 0, \dots, n-2$ :

$$\begin{aligned} s_{k+1} - s_k &= \frac{h}{2}(v_k + v_{k+1}) \\ v_{k+1} - v_k &= \frac{h}{2}(a_k + a_{k+1}) \end{aligned}$$

Unitamente a questi, è necessario imporre in ogni punto di collocazione il vincolo sull'accelerazione, ossia:

$$a_k = h(s_k) - (\alpha + \beta|v_k| + \gamma v_k^2) + u_k^a + u_k^f \quad \forall k = 0, \dots, n-2 \quad (3.10)$$

La funzione obiettivo si ottiene mediante una formula di quadratura trapezoidale composta del funzionale di costo:

$$\min_{\mathcal{X} \in \mathbb{R}^{5n}} J(\mathcal{X}) = \frac{h}{2} \left( u_0^a v_0 + 2 \sum_{k=1}^{n-2} u_k^a v_k + u_{n-1}^a v_{n-1} \right) \quad (3.11)$$

Il problema di ottimizzazione così ottenuto è stato risolto con l'algoritmo Sequential Quadratic Programming implementato nella libreria Scipy in Python.

## 3.2 Esempio 2: La catenaria come problema di controllo

Un problema classico nell'ambito della minimizzazione di funzionali, è il cosiddetto **problema della catenaria**. Studiato per la prima da Galileo nel 1638, tale problema consiste nel determinare la geometria che una catena di massa omogenea e lunghezza fissata assume se sospesa per i suoi due capi. Come è noto, tale problema è risolvibile con i metodi della meccanica analitica mediante le equazioni di Eulero-Lagrange che permettono di dimostrare la soluzione generale: il **coseno iperbolico**.

Come mostrato in [6] e ripreso nella suite COPS [7], il problema della catenaria può essere anche impostato come un problema di controllo ottimo. In particolare, detta  $x(t)$  l'equazione della curva incognita, si può considerare come controllo la derivata prima del sistema, giungendo quindi alla dinamica:

$$\dot{x} = u \quad (3.12)$$

Concordemente con le leggi fisiche, la curva catenaria sarà quella che minimizza l'energia potenziale totale del sistema, ossia il funzionale:

$$\mathcal{E} = \int_{t_I}^{t_F} x(t) \sqrt{1 + u^2(t)} dt \quad (3.13)$$

Inoltre, si ha che, fissata la lunghezza totale  $L$ , la curva  $x(t)$  deve rispettare:

$$\int_{t_I}^{t_F} \sqrt{1 + \dot{x}^2(t)} dt = L \quad (3.14)$$

Nelle implementazioni disponibili nella cartella /catenary/ si sono utilizzati i seguenti valori numerici:

Costante	Valore
$t_I$	$0.0m$
$t_F$	$10.0m$
$x_I$	$5.0m$
$x_F$	$7.5m$
$L_{tot}$	$15m$

Come accennato nella sezione riguardante la scelta della guess iniziale, si è iniziato applicando il metodo dei trapezi, ottenendo il seguente problema NLP:

$$\min_{\mathbf{x}, \mathbf{u}} \quad \frac{h}{2} \left( x_0 w_0 + 2 \sum_{k=1}^{n-2} x_k w_k + x_{n-1} w_{n-1} \right) \quad (3.15)$$

soggetto ai vincoli

$$x_0 = 5.0$$

$$x_{n-1} = 7.5$$

$$\frac{h}{2} \left( w_0 + 2 \sum_{k=1}^{n-2} w_k + w_{n-1} \right) = 15.0$$

$$x_{k+1} - x_k = h(u_k + u_{k+1}) \quad \forall k = 0, \dots, n-2$$

dove si è definito:

$$w_k \doteq \sqrt{1 + u_k^2} \quad (3.16)$$

Inizializzando l'algoritmo di ottimizzazione Sequential Quadratic Programming con delle scelte povere di guess iniziale ha portato alla non-convergenza. In particolare, il risolutore restituiva il messaggio:

Singular matrix C in LSQ subproblem

Si è quindi costruita una guess iniziale migliore, trovando una parabola passante per i due punti definiti dalle condizioni al contorno. In particolare, imponendo:

$$\begin{cases} 0^2 a + 0b + c = 5 \\ 10^2 a + 10b + c = 7.5 \end{cases} \quad (3.17)$$

si ottiene un sistema con un'incognita libera:

$$\begin{cases} c = 5 \\ b = 0.25 - 10a \end{cases} \quad (3.18)$$

Ponendo  $a = 0.1$ , si ottiene la parabola:

$$x = 0.1t^2 - 0.75t + 5 \quad (3.19)$$

Come mostrato in Figura 3.1, tale miglioramento ha permesso di ottenere un problema convergente che già fornisce un'approssimazione della soluzione esatta.

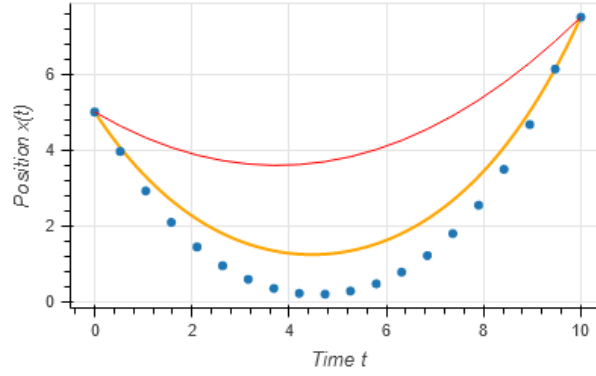


FIGURA 3.1: Problema della catenaria: in rosso, la guess iniziale parabolica; in giallo, la soluzione esatta; in blu, la soluzione numerica

### 3.3 Esempio 3: Model Predictive Contouring Control

I metodi di collocazione diretta possono essere utilizzati per eseguire il controllo ottimo di un sistema, anche se il modello nominale di cui si è in possesso non è particolarmente accurato. Questo è ad esempio il caso di sistemi controllati ad alte velocità e al limite del dominio di definizione del modello nominale, come un'auto da corsa. Per ovviare alla conoscenza parziale della dinamica reale del sistema e per ridurre la dimensione del problema di ottimizzazione, è possibile adottare un approccio iterativo in *real-time*: fissata una frequenza di esecuzione, ad ogni iterazione verrà risolto un nuovo problema di ottimizzazione della traiettoria i cui vincoli saranno scelti per tenere conto dello stato del sistema in quel istante. Il controllo ottimo ottenuto come soluzione, viene quindi applicato al sistema andando a determinare lo stato dello stesso alla prossima iterazione.

Tale tecnica è definita **Model Predictive Control** in quanto, durante ciascuna iterazione dell'algoritmo di ottimizzazione, viene scelto un controllo e viene predetta la corrispondente traiettoria dello stato del sistema, basandosi sul modello dinamico. Nell'ambito dell'insegnamento "Automotive Learning by Doing 1", si è applicato tale approccio per la realizzazione di un controller per un'auto da corsa a guida autonoma, basandosi sul lavoro già svolto negli anni scorsi da Andrea Montanari (ex-studente UNIMORE). La dinamica del veicolo è stata approssimata con il **modello a bicicletta**, un sistema basato su una geometria con un singolo asse longitudinale e due sole ruote. Le equazioni del modello utilizzate sono:

$$\begin{cases} \dot{x}(t) = v(t) \cos(\beta(t) + \theta(t)) \\ \dot{y}(t) = v(t) \sin(\beta(t) + \theta(t)) \\ \dot{\theta}(t) = \frac{v(t)}{L} \tan \delta(t) \cos \beta(t) \\ \dot{v}(t) = a_{throttle}(t) + a_{brake}(t) \\ \beta(t) = \arctan\left(\frac{L_{rear}}{L} \tan(\delta(t))\right) \end{cases} \quad (3.20)$$

dove si ha:

- Le componenti  $x$  e  $y$  denotano la posizione del centro di massa del veicolo nel sistema di riferimento globale
- La componente  $\theta$  denota l'angolo tra l'asse del veicolo e l'asse  $x$

- $v$  è il modulo della velocità del veicolo, riferita alla direzione dell'asse
- Il controllo di acceleratore e freno è espresso rispettivamente da  $a_{throttle}$  e  $a_{brake}$
- L'angolo di sterzo è espresso da  $\delta$
- Il termine  $\beta$  denota lo **slipping angle**, ossia la differenza in angolo tra la direzione della velocità del veicolo e la direzione dell'asse principale

Il software di *perception* del veicolo, basato su tecniche di computer vision e machine learning, restituisce al controller una sequenza ordinata di punti al centro della carreggiata. Tali punti, interpolati mediante una spline cubica parametrica, costituiscono la **traiettoria di riferimento** che il controller dovrebbe cercare di seguire. A tal fine, si definiscono i seguenti errori:

**Definizione 7** (Cross-Track Error). Dato un punto  $(x, y)$  nel sistema di riferimento globale, si dirà **cross-track error (cte)** la minima distanza tra il punto e la curva della traiettoria di riferimento.

**Definizione 8** (Heading Error). Dato un punto  $(x, y)$  ed una direzione di percorrenza  $\hat{u}$ , si dirà **heading error (epsi)** la differenza in angolo tra il versore  $\hat{u}$  ed il versore tangente alla traiettoria di riferimento, nel punto più vicino a  $(x, y)$ .

Inoltre, per ciascun punto della traiettoria di riferimento, è possibile stimare la velocità massima che il veicolo potrà tenere, utilizzando le informazioni sulla curvatura del tracciato e sull'attrito dei pneumatici con l'asfalto.

**Definizione 9** (Velocità target). Fissato un punto sulla traiettoria di riferimento, detto  $\mu$  il coefficiente di attrito del veicolo e  $r$  il raggio di curvatura della traiettoria di riferimento nel punto in considerazione, si dirà **velocità target** la seguente:

$$targetVelocity \doteq \sqrt{g\mu r} \quad (3.21)$$

dove  $g$  denota l'accelerazione di gravità.

Utilizzando la notazione finora introdotta, è possibile formulare il problema di *path following*, ossia il calcolo dei comandi necessari per seguire la traiettoria di riferimento con la velocità target desiderata, come un problema NLP. Si definisce la seguente variabile di ottimizzazione:

$$\mathcal{X} = (\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \mathbf{v}, \mathbf{a}, \delta, cte, \text{epsi}, \mathbf{x}) \quad (3.22)$$

I vincoli sulla dinamica del sistema, descritti nell'Equazione 3.20, vengono discretizzati applicando il metodo di collocazione dei trapezi. La funzione obiettivo è invece definita come:

$$\begin{aligned} J = & w_{cte} \sum [cte(t_i)]^2 \\ & + w_{epsi} \sum [epsi(t_i)]^2 \\ & + w_v \sum [v(t_i) - targetVelocity]^2 \\ & + w_{delta} \sum [\delta(t_i)]^2 \\ & + w_a \sum [a(t_i)]^2 \\ & + w_{delta_a} \sum [\delta(t_{i+1}) - \delta(t_i)]^2 \end{aligned}$$



$$+ w_{acc_d} \sum [a(t_{i+1}) - a(t_i)]^2$$

I vari addendi hanno rispettivamente lo scopo di minimizzare il cross-track error, l'heading error, la differenza tra la velocità effettiva e la velocità target, l'utilizzo dello sterzo e dell'acceleratore e la variazione del controllo in istanti successivi (tecnica di regolarizzazione utilizzata per ottenere un controllo più fluido e stabile). I vettori  $\mathbf{a}^*$  e  $\delta^*$ , soluzioni del problema NLP, forniranno il controllo cercato.

Con un orizzonte temporale di predizione di 2s e un intervallo di discretizzazione di 50ms, il problema NLP trascritto ha un totale di 320 variabili e più di 250 vincoli. Utilizzando il pacchetto di ottimizzazione numerica Interior Point Optimizer, ciascun problema viene risolto in media in 10ms, permettendo un controllo del veicolo in tempo reale con una frequenza di aggiornamento di 10Hz.

*Osservazione* (Complessità computazionale). Una lezione appresa nel corso dello sviluppo è stata l'importanza di mantenere il costo computazionale della valutazione della funzione obiettivo e dei vincoli ragionevolmente basso. In particolare, l'algoritmo implementato in partenza per determinare il punto più vicino sulla spline interpolante il tracciato rispetto alla posizione corrente utilizzava un metodo di discesa del gradiente che, seppur molto preciso, risultava piuttosto dispendioso dal punto di vista computazionale. Dal momento che tale funzione viene valutata, ad ogni iterazione del problema, una volta per ciascun punto di collocazione, il processo di ottimizzazione del problema NLP raggiungeva tempi di completamento inaccettabili, nell'ordine dei 5 – 10s. Per evitare tale problema, si è discretizzata la spline con una risoluzione prefissata e si sono organizzati i punti così ottenuti in una struttura *quad-tree* che permette di determinare il punto più vicino sulla spline in tempo logaritmico rispetto al numero di nodi. Tale cambiamento ha ridotto drasticamente il costo computazionale della valutazione dei vincoli e della funzione obiettivo ed ha permesso di raggiungere le attuali performance del controller.



## Capitolo 4

# Riflessioni conclusive

Obiettivo di questa Tesi è fornire un'introduzione ai problemi del controllo ottimo, ed in particolare ai problemi di ottimizzazione della traiettoria, e mostrare come possono essere risolti numericamente con i metodi di collocazione diretta. Dopo una prima presentazione teorica dei problemi in questione ed un'analisi delle principali difficoltà nel risolverli, si sono derivati tre dei principali schemi di collocazione diretta: il metodo del punto medio, il metodo del trapezio ed il metodo di Hermite-Simpson. In seguito, si sono trattati alcuni argomenti che emergono dall'utilizzo di queste tecniche, in particolare riguardanti l'analisi dell'accuratezza. Infine, nel Capitolo 3 si sono presentate tre applicazioni dei metodi discussi: le prime due tratte dalla letteratura scientifica e la terza frutto dell'esperienza acquisita nel team di Ateneo per la Formula SAE Driverless. Risolvendo passo passo questi tre problemi, dall'impostazione teorica fino alla fase implementativa, si sono affrontate alcune problematiche tipiche dei metodi numerici e che hanno portato ad alcune delle osservazioni contenute nella discussione precedente.

In conclusione, il percorso elaborato in questa Tesi si colloca all'intersezione tra diverse aree della matematica applicata, trattando temi dell'ottimizzazione numerica, del controllo ottimo e dell'analisi funzionale. Il lavoro svolto ha permesso di consolidare le competenze di base nell'ambito della trattazione numerica di problemi di controllo ottimo e di approfondire le competenze di programmazione in ambiente Python e C++.



# Bibliografia

- [1] Kendall E. Atkinson. *An introduction to numerical analysis*. en. 2nd ed. New York: Wiley, 1989. ISBN: 978-0-471-62489-9.
- [2] Andrew G. Barto. «Reinforcement learning control». en. In: *Current Opinion in Neurobiology* 4.6 (dic. 1994). ISSN: 0959-4388.
- [3] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. en. Second. Society for Industrial e Applied Mathematics, gen. 2010. ISBN: 978-0-89871-688-7 978-0-89871-857-7.
- [4] Silvia Bonettini e Simone Rebegoldi. *Dispense di Analisi Numerica A*.
- [5] Alberto Bressan e Benedetto Piccoli. *Introduction to the Mathematical Theory of Control*. Gen. 2007.
- [6] Lamberto Cesari. *Optimization - Theory and Applications*. A cura di Lamberto Cesari. Applications of Mathematics. New York, NY: Springer, 1983. ISBN: 978-1-4613-8165-5.
- [7] Elizabeth D Dolan, Jorge J More e Todd S Munson. «Benchmarking Optimization Software with COPS 3.0». en. In: (), p. 57.
- [8] Dimitris Dracopoulos. *Evolutionary Learning Algorithms for Neural Adaptive Control*. en. Perspectives in Neural Computing. London: Springer-Verlag, 1997. ISBN: 978-3-540-76161-7.
- [9] Thomas Duriez, Steven L. Brunton e Bernd R. Noack. *Machine Learning Control - Taming Nonlinear Dynamics and Turbulence*. en. A cura di Thomas Duriez, Steven L. Brunton e Bernd R. Noack. Fluid Mechanics and Its Applications. Cham: Springer International Publishing, 2017. ISBN: 978-3-319-40624-4.
- [10] Maurizio Falcone et al., cur. *Numerical Methods for Optimal Control Problems*. en. Springer INdAM Series. Springer International Publishing, 2018. ISBN: 978-3-030-01958-7.
- [11] David H. Jacobson e David Q. Mayne. *Differential dynamic programming*. Modern analytic and computational methods in science and mathematics no. 24. New York: American Elsevier Pub. Co, 1970.
- [12] Matthew Kelly. «An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation». en. In: *SIAM Review* 59.4 (gen. 2017), pp. 849–904. ISSN: 0036-1445, 1095-7200.
- [13] Giovanni Marro. «Introduzione al Controllo Ottimo». it. In: (), p. 40.
- [14] Diego Pardo et al. «Evaluating direct transcription and nonlinear optimization methods for robot motion planning». en. In: *IEEE Robotics and Automation Letters* 1.2 (lug. 2016). arXiv: 1504.05803, pp. 946–953. ISSN: 2377-3766, 2377-3774.
- [15] Diego Pardo et al. «Evaluating direct transcription and nonlinear optimization methods for robot motion planning». In: *IEEE Robotics and Automation Letters* 1.2 (lug. 2016). arXiv: 1504.05803. ISSN: 2377-3766, 2377-3774.

- [16] Hector J. Sussmann e Jan C. Willems. «300 Years of Optimal Control: From The Brachystochrone to the Maximum Principle». In: *IEEE Control Systems* ().
- [17] Robert J Vanderbei. «Case Studies in Trajectory Optimization: Trains, Planes, and Other Pastimes». en. In: (), p. 29.